

Business Decision Modeling

Prof. Dr. Mathias Weske

Hasso Plattner Institute, University of Potsdam
Business Process Technology

Business Decision Modeling

1. Motivation
2. DMN Overview
3. Decision Requirements Diagrams
4. Decision Logic
5. Semantics of Decision Tables
6. Analysis of Decision Tables
7. Consistency of Processes and Decisions

Video Clip 1

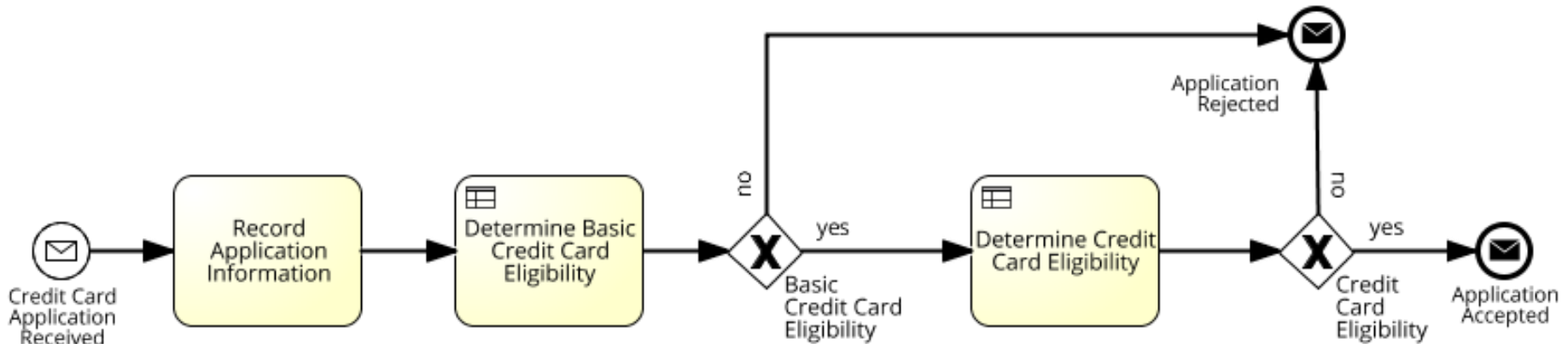
Motivation

(Business Decision Modeling)

Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

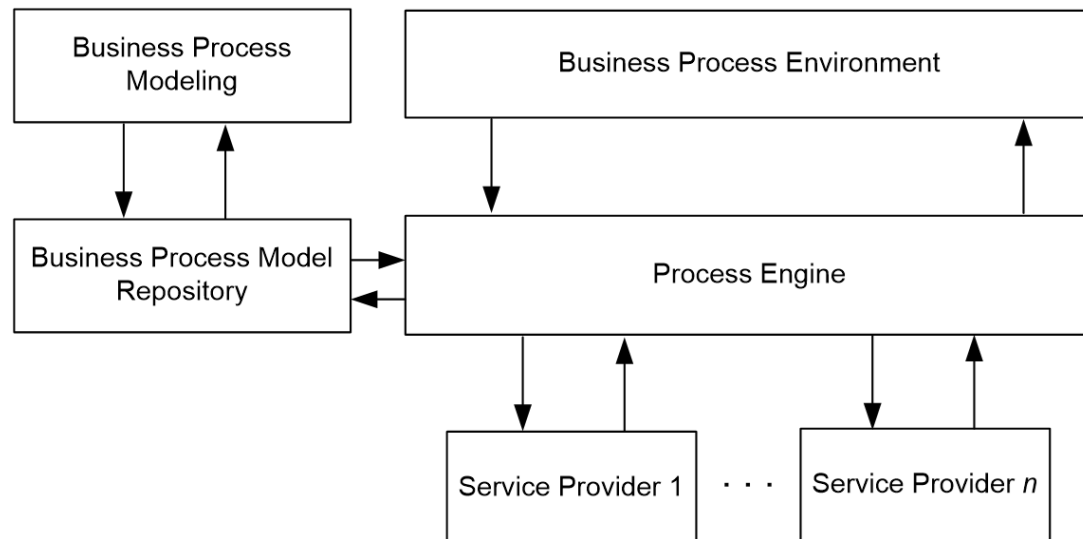
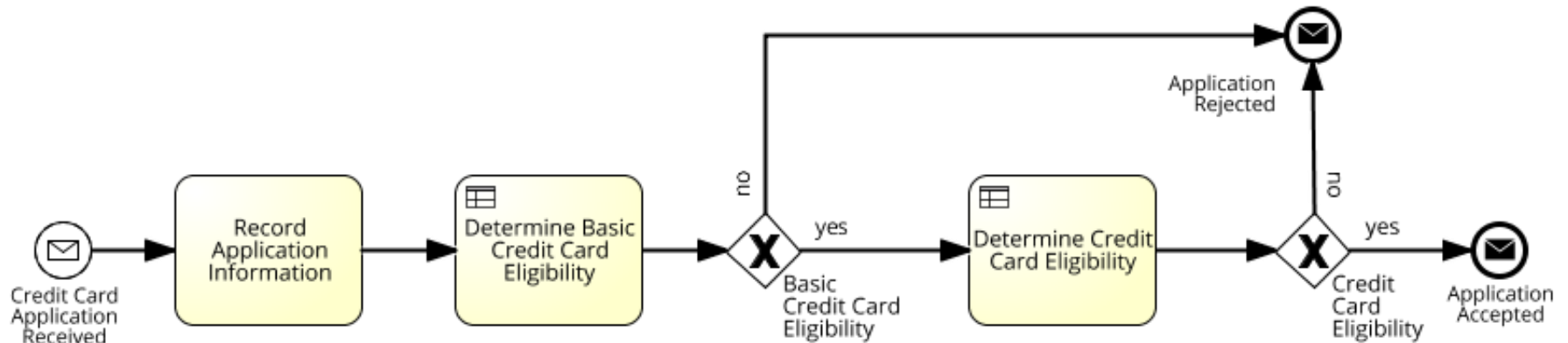
Motivation

- Many business processes contain decisions
 - Decisions might impact sequence flow (XOR gateway)
 - Decisions assign values (e.g., discount for specific customers)
- Example
 - Determine credit card eligibility



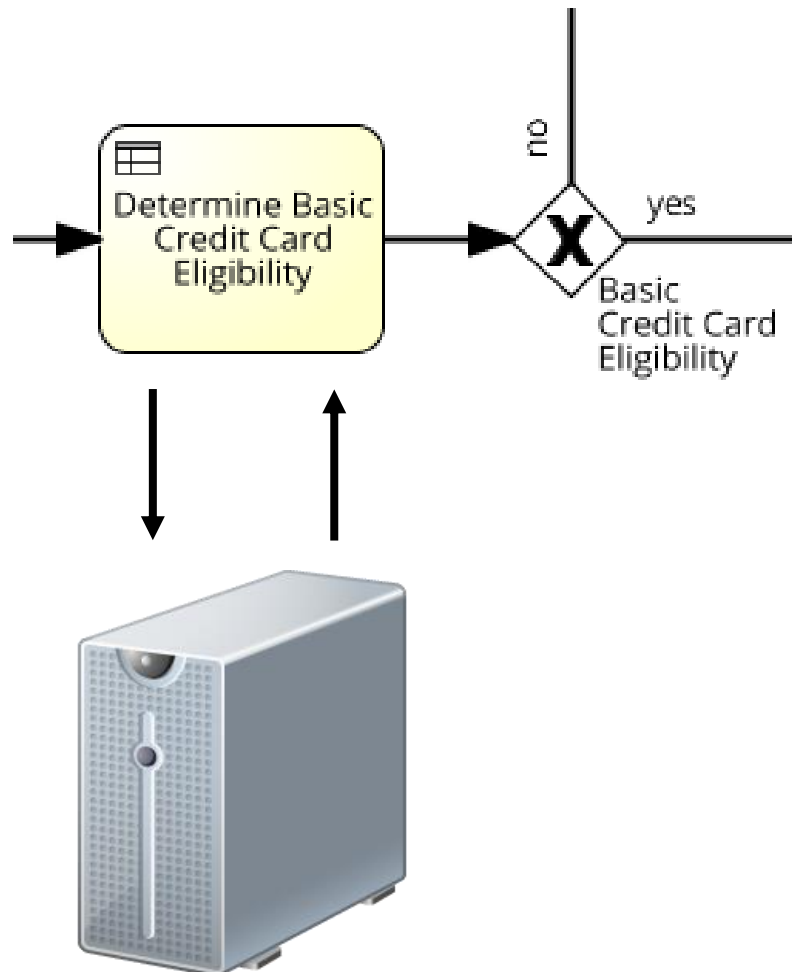
It's the decisions that make or break your company

Process Automation



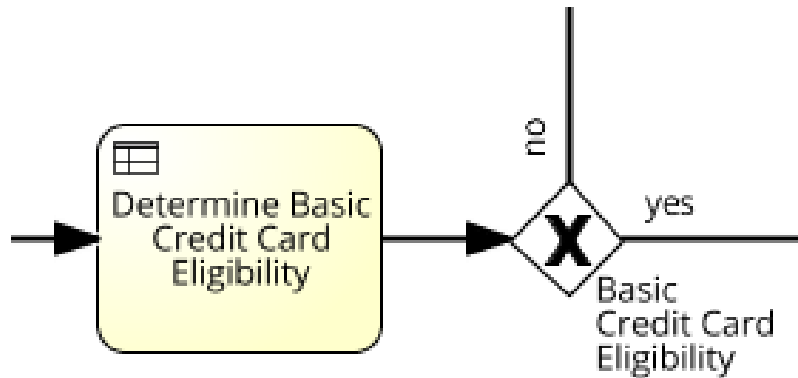
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Implementation of Decisions (1/2)



- Decision is implemented in software
- Issues
 - How is the decision taken?
 - Which data is being used?
 - Which policies are taken into account?
 - Who is responsible for the decision?

Implementation of Decisions (2/2)



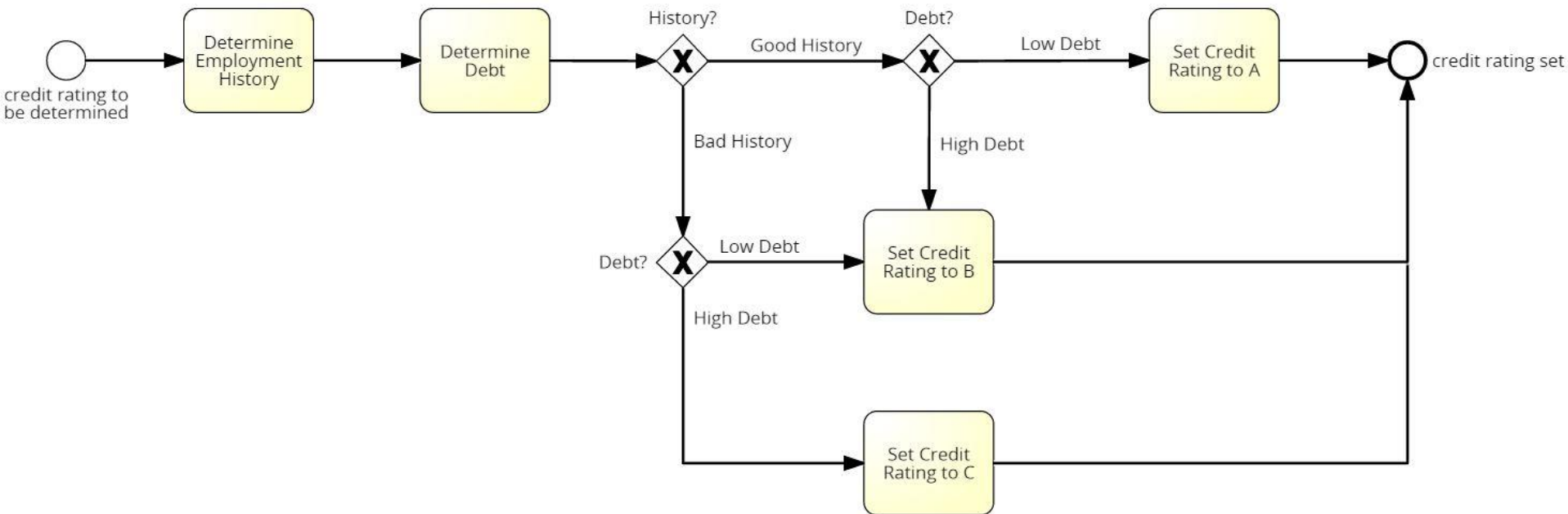
- Decision is taken by process participant
- Issues
 - How is the decision taken?
 - Which data is being used?
 - Which policies are taken into account?
 - Who is responsible for the decision?

yes

no

Decisions encoded in Processes

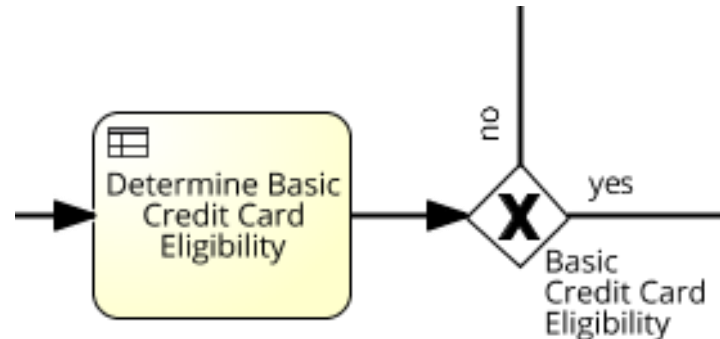
- Empirical analysis of 500+ process models from the financial industry
- Example



Process models are being mis-used to model decisions

Decision Modeling

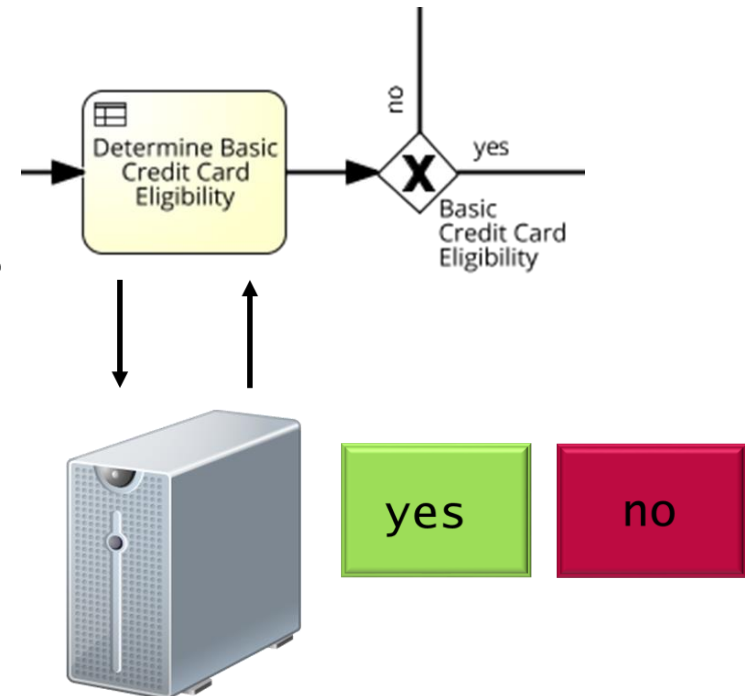
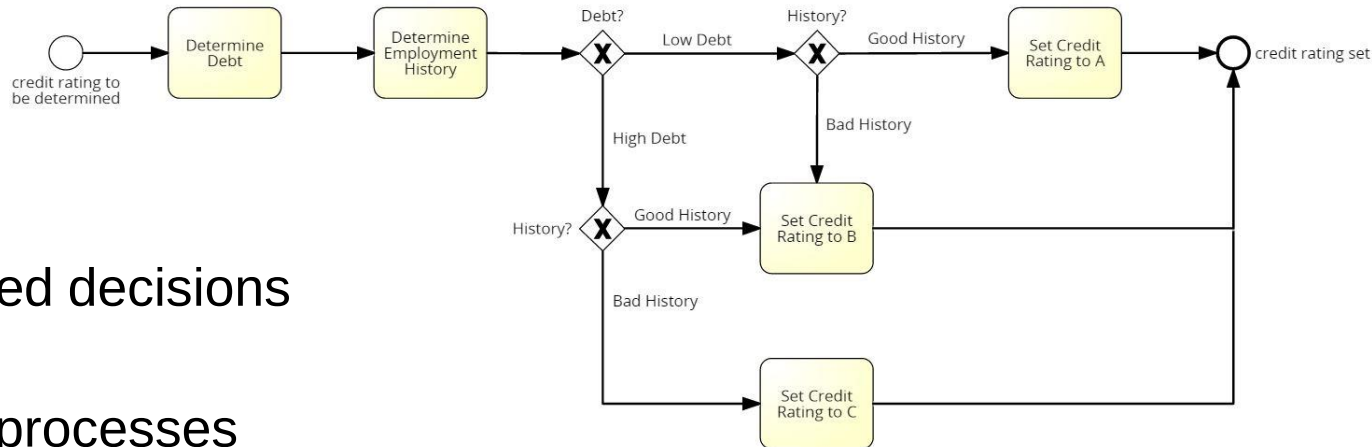
- Describe decisions in a dedicated decision model
- Focus is on well-defined, operational decisions
 - Like granting a credit or processing an insurance claim
 - Focus is not on strategic decisions like entering a new market
- Modeling goals
 - Improve transparency, maintainability, and change
 - Support automation of repeatable decisions
- Explicit representation of
 - decision structure and decision logic to provide a common language for domain experts and engineers



Video Clip 1

Motivation

- Avoid ...
 - undocumented decisions
 - representing decisions in processes
- Answer questions
 - How is the decision taken?
 - Which data is being used?
 - Which policies are taken into account?
 - Who is responsible for the decision?
- Provide a common language
 - for domain experts and engineers



Video Clip 2

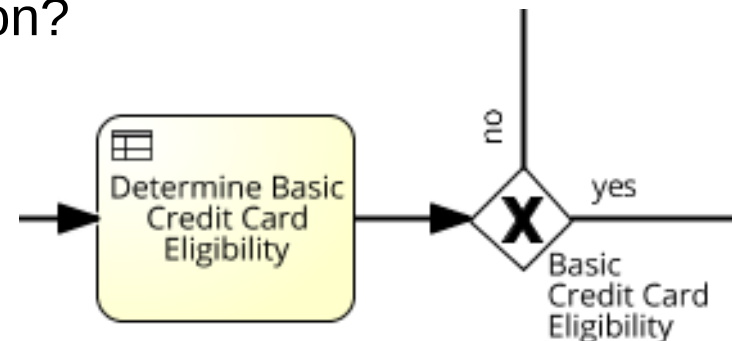
DMN Overview

(Business Decision Modeling)

Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

Decision Management

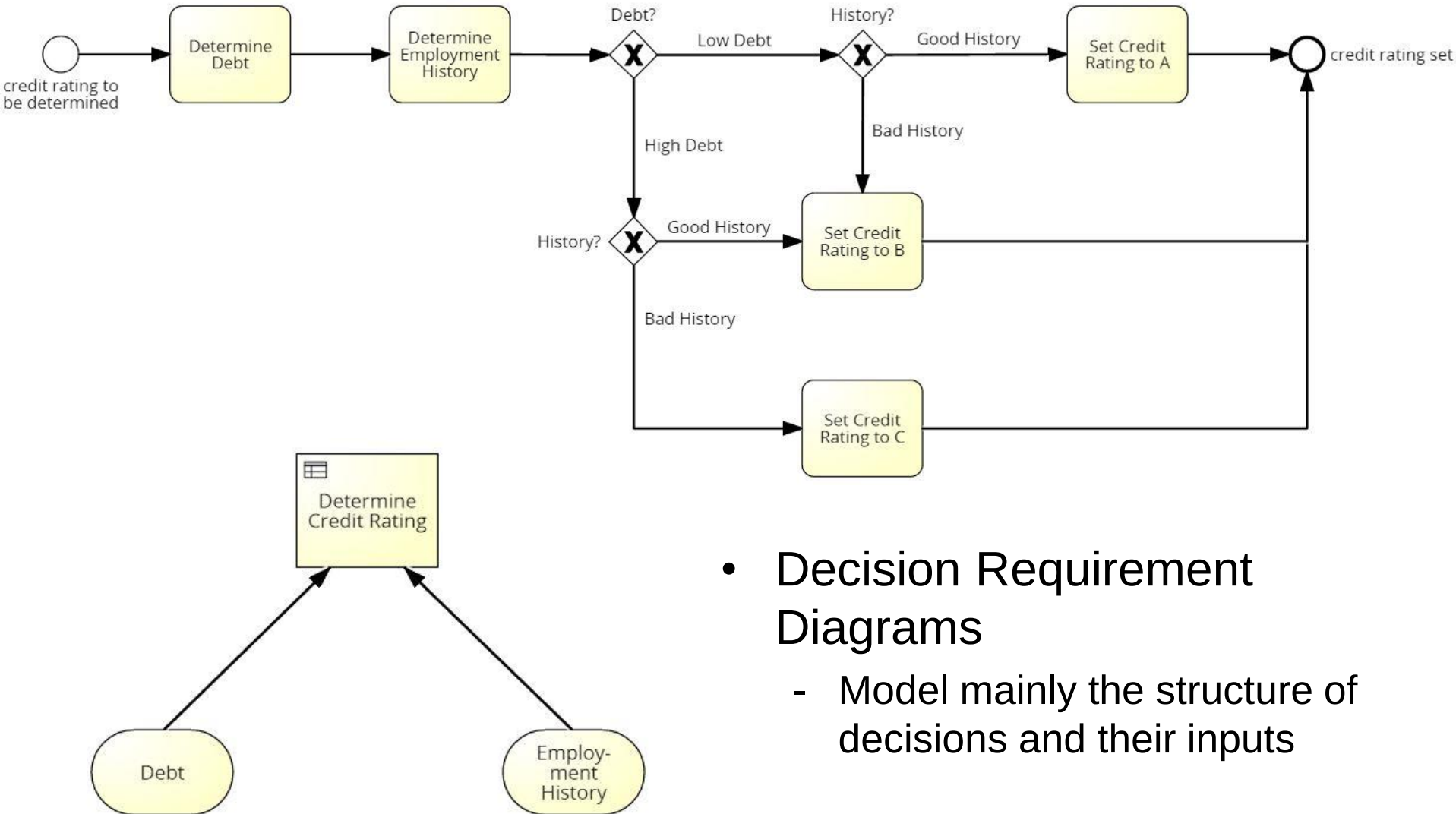
- Goal of decision management
 - Model well-defined, operational decisions, thereby improving transparency, maintainability, and supporting change and automation of decisions
- Decision management answers questions
 - How is the decision taken?
 - Which data is being used?
 - Which policies are taken into account?
 - Who is responsible for the decision?



Decision Model and Notation

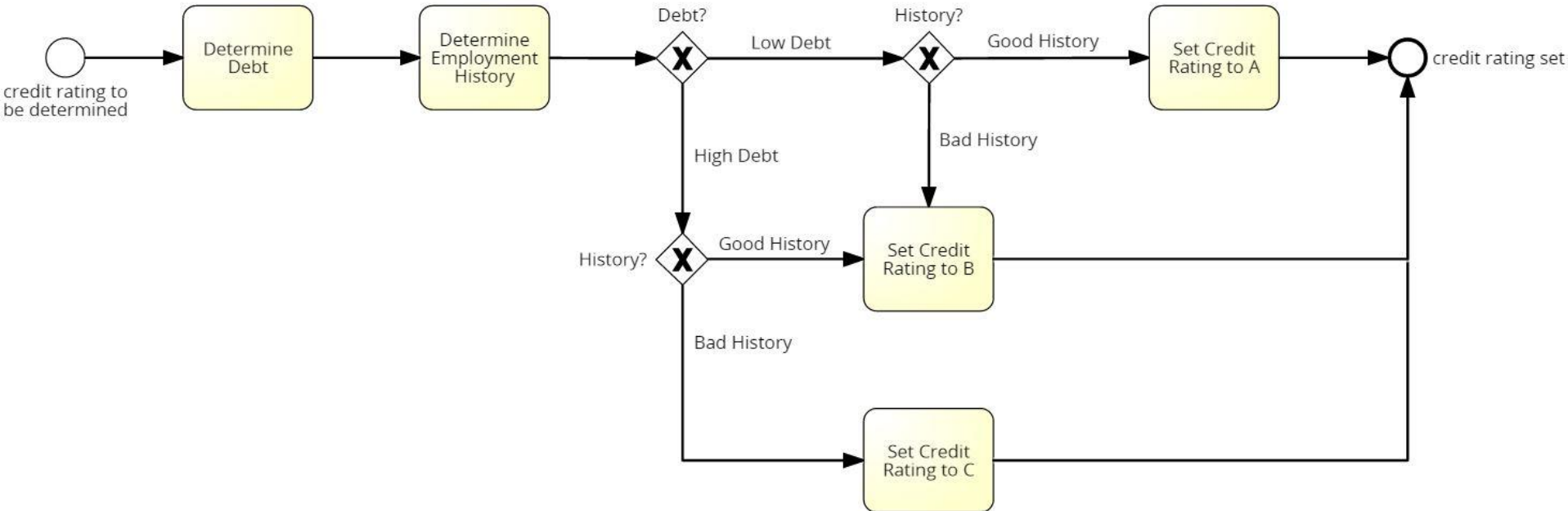
- DMN
 - OMG standard
 - DMN considers two levels of abstraction
- Decision requirements
 - Business level to define structure of decisions, their inputs, and relationships between them (and a few other concepts)
- Decision logic
 - Defines how decisions are taken, e.g., by decision tables
- Relationship to BPMN
 - DMN complements BPMN nicely
 - But DMN can also be used without BPMN!

Example 1: Decision Requirements



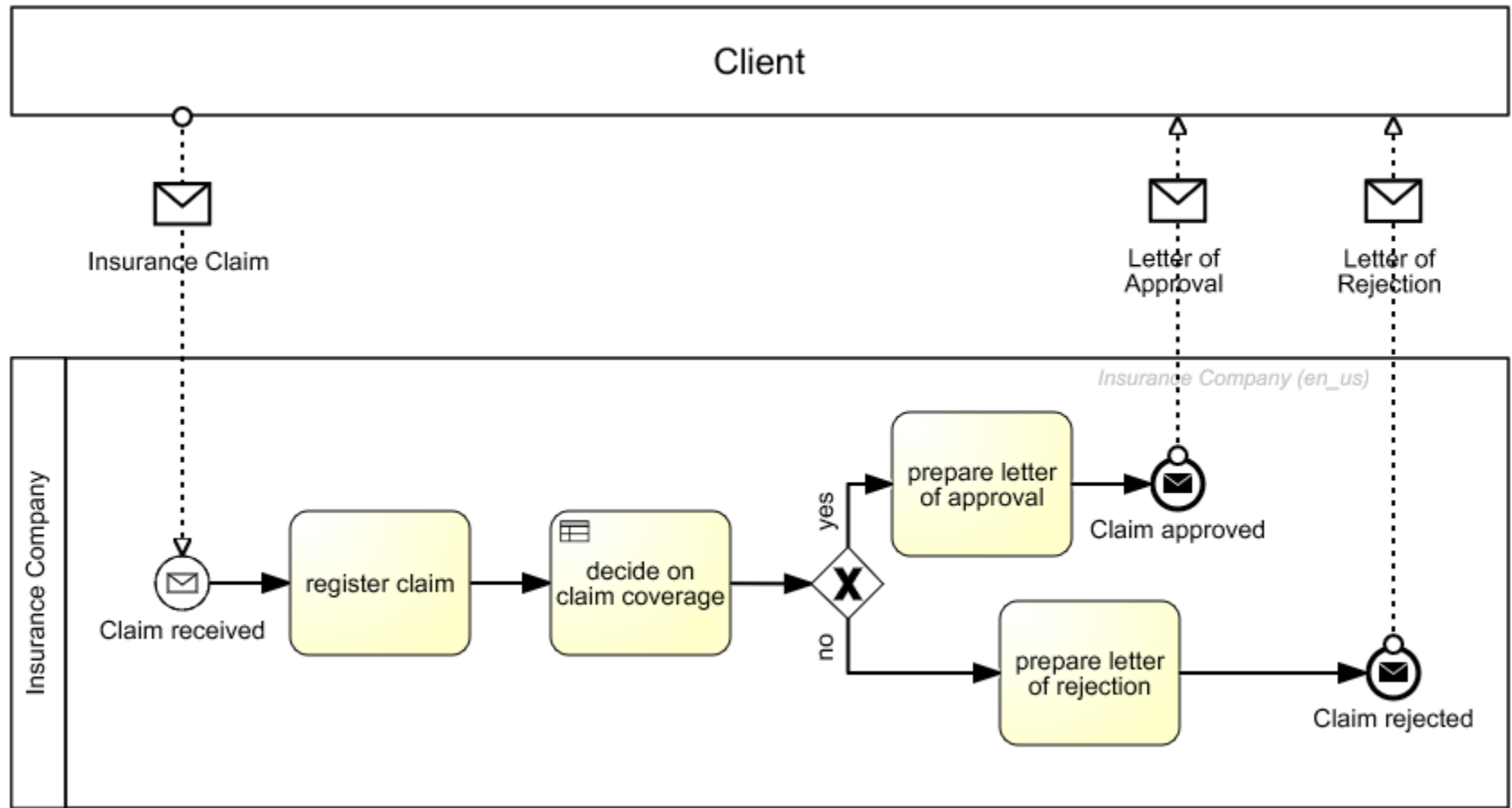
- Decision Requirement Diagrams
 - Model mainly the structure of decisions and their inputs

Example 1: Decision Logic



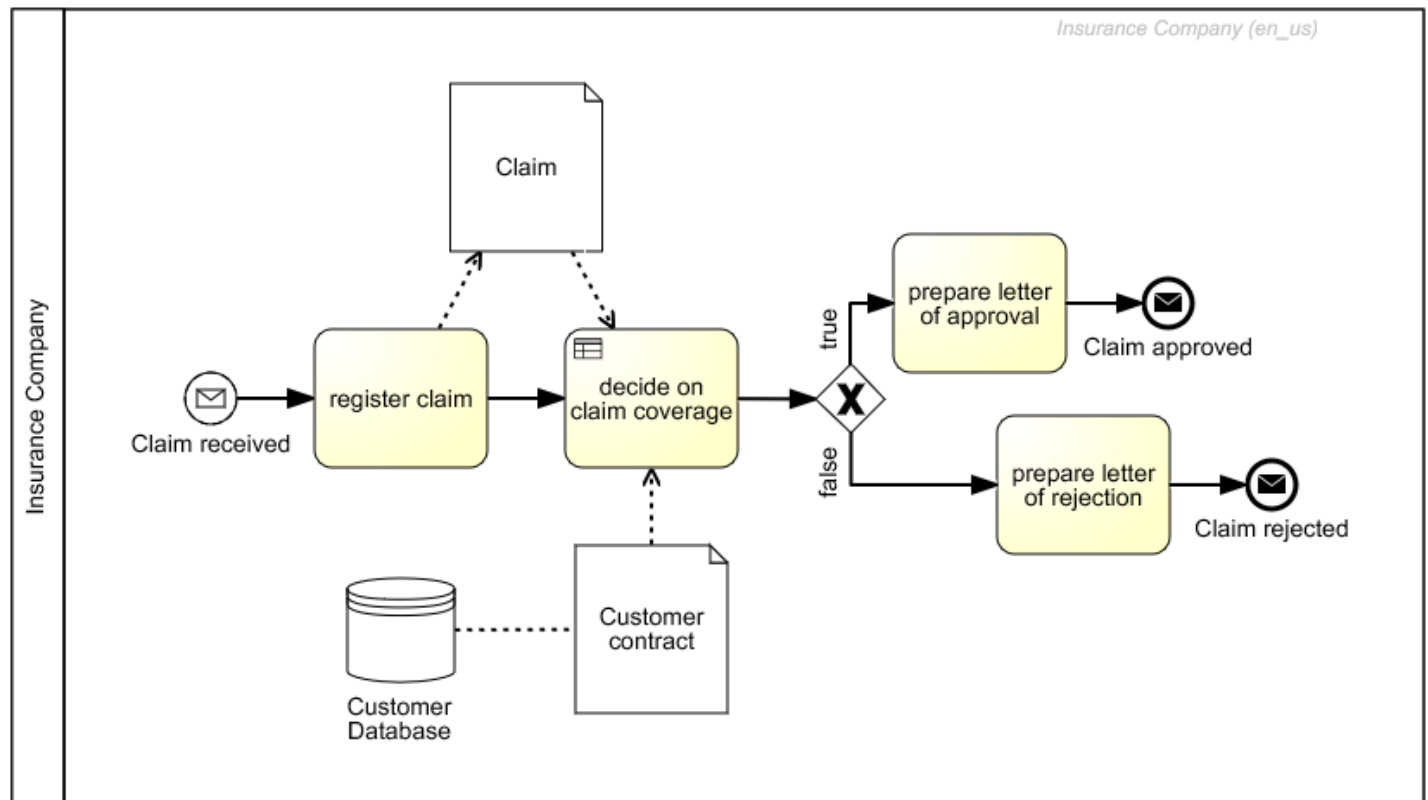
UC	Inputs		Outputs	
	Debt	Employment History	Credit Rating	
	{Low, High}	{Bad, Good}	{A, B, C}	
1	= Low	= Good	A	
2	= Low	= Bad	B	
3	= High	= Good	B	
4	= High	= Bad	C	

Example 2: Insurance Claim Process



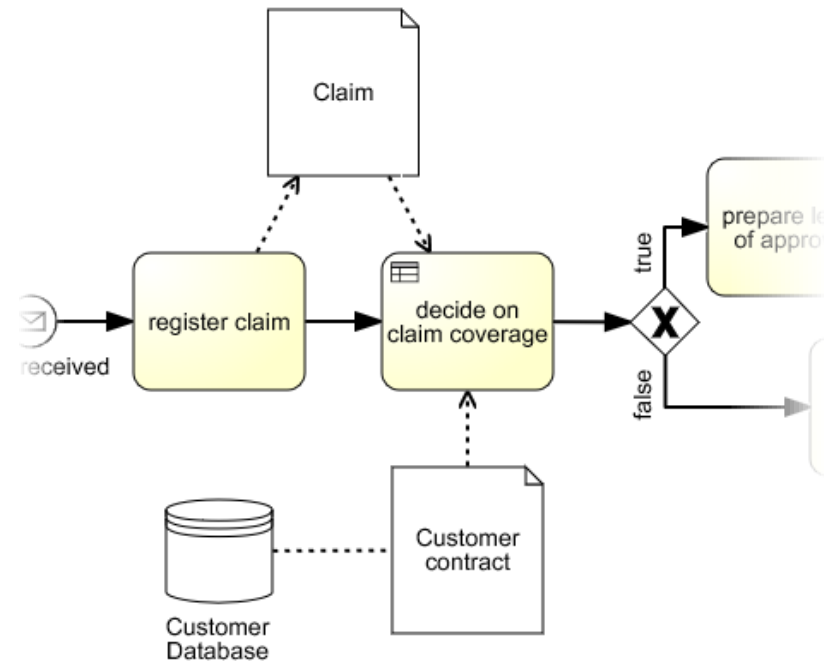
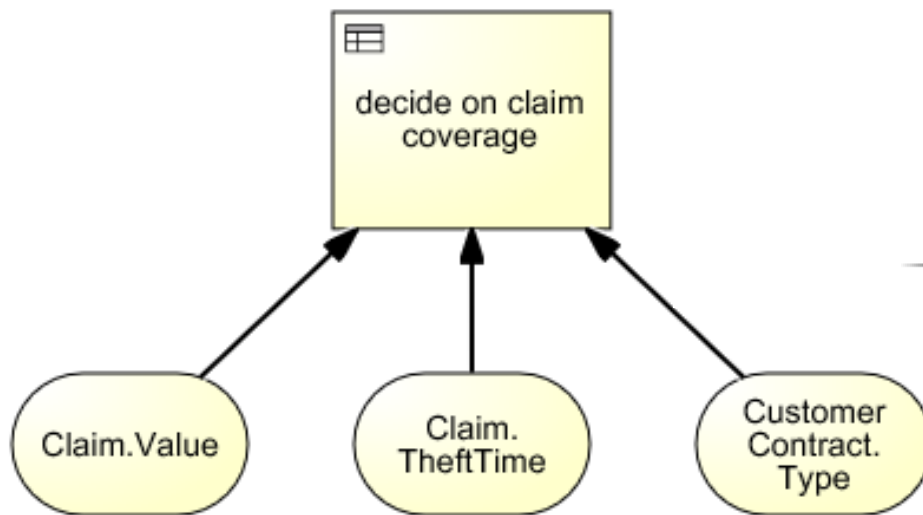
Example 2: Adding Data Objects

- In BPMN, data is represented by data objects
 - Dashed edges indicate the reading and writing of data objects
 - Associations with data stores ease implementation
 - Use data models to define data structures, e.g., attributes

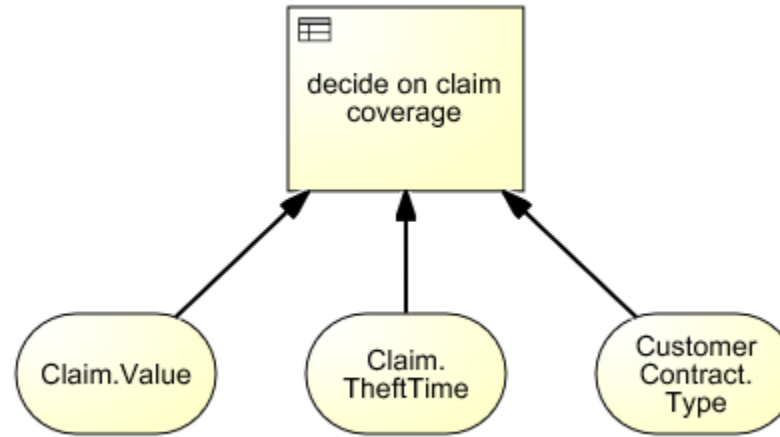


Example 2: Decision Requirements

- Data input might refer to attributes of data objects
 - Providing a reference between the models



Example 2: Decision Logic

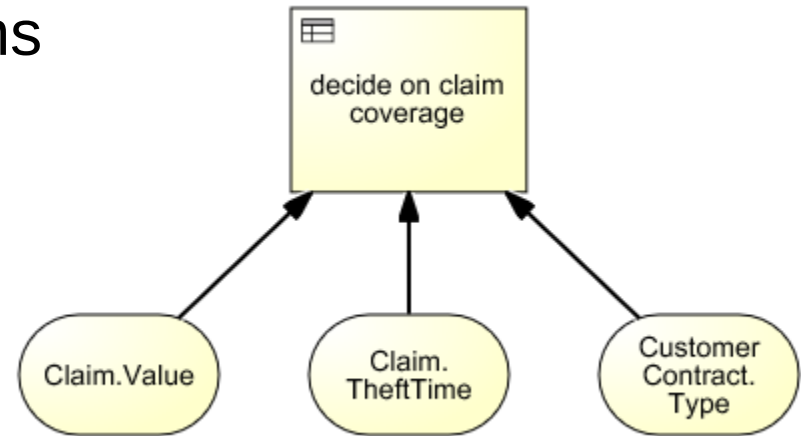


Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	<i>[0€;2000€]</i>	<i>[0;24]</i>	<i>{Gold, Silver, Bronze}</i>	<i>{yes, no}</i>
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

Video Clip 2

DMN Overview

- Decision Requirements Diagrams model the structure of decisions
- Decision logic expresses how decisions are taken



Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

Video Clip 3

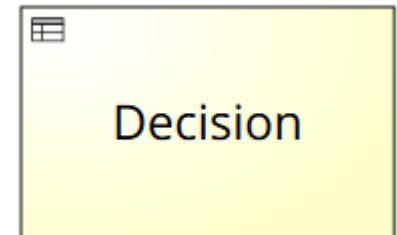
Decision Requirements Diagrams

(Business Decision Modeling)

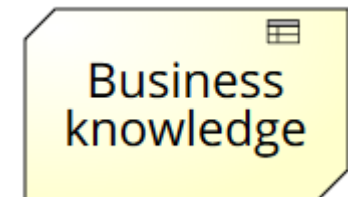
Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

Elements of Decision Requirements Diagrams

- Decision
 - Determines from a set of input values (typically one) output value
 - Is assigned a decision logic, but can also invoke a *business knowledge*



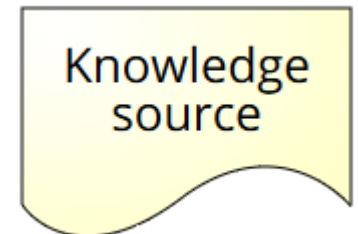
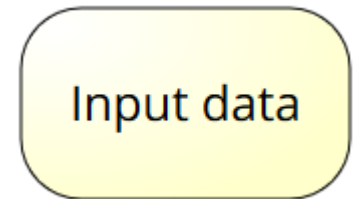
- Business knowledge
 - Contains knowledge on how a decision should be taken
 - Can be re-used by different decisions
 - Is invoked by decisions
 - Input data is transferred to the business knowledge
 - Business knowledge provides output to the decision



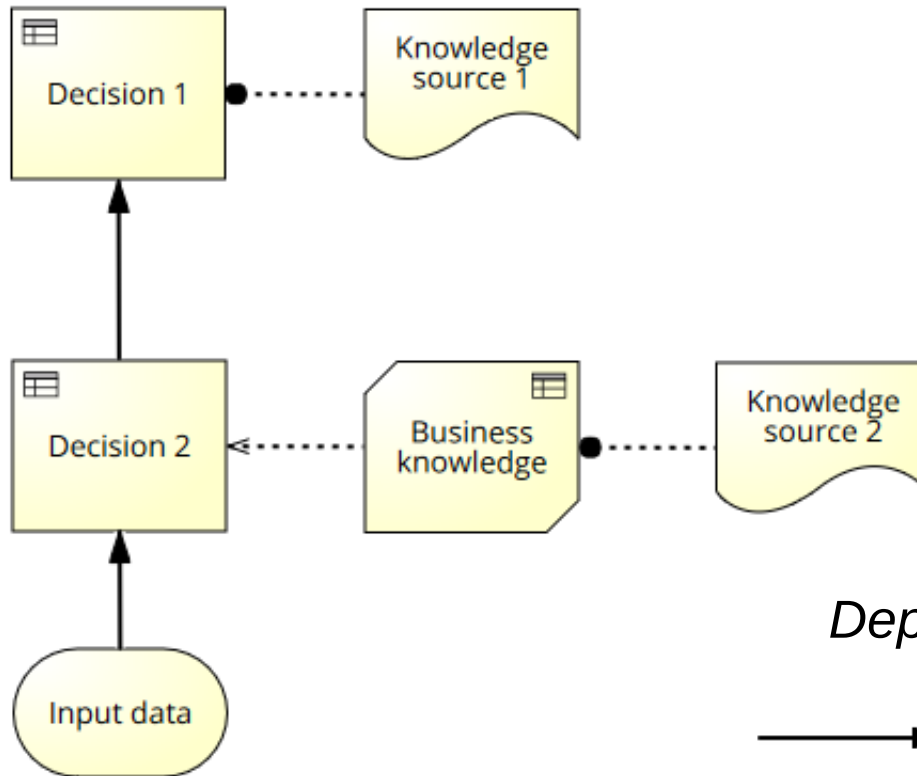
Further Model Elements

- Input data
 - Input data for decisions
 - To be used during decision making
 - Can be linked to a process model by label
 - Can represent attributes of data objects

- Knowledge source
 - Represents reasons for decisions, to document decisions, e.g.,
 - Legal regulations
 - Business policies
 - Human experts



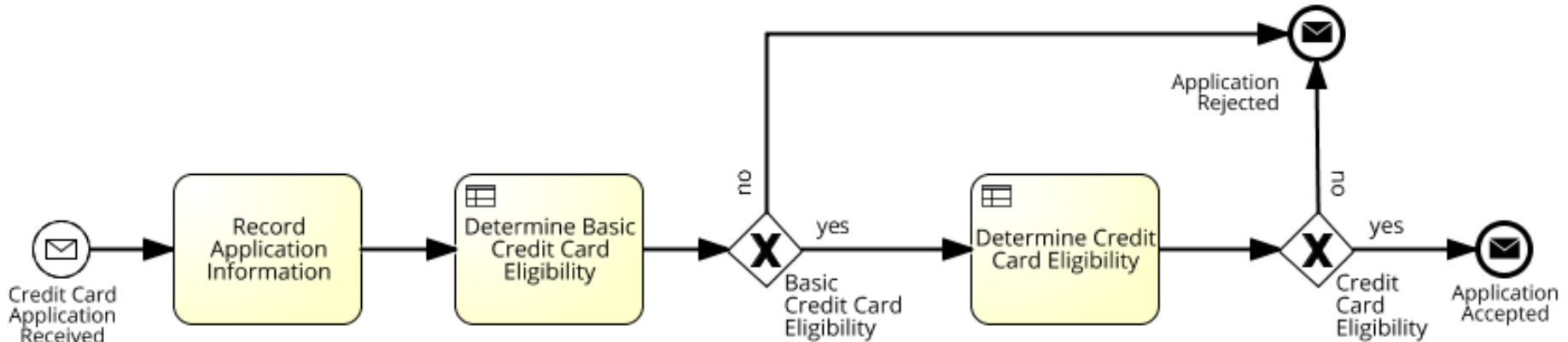
Decision Requirements Graph



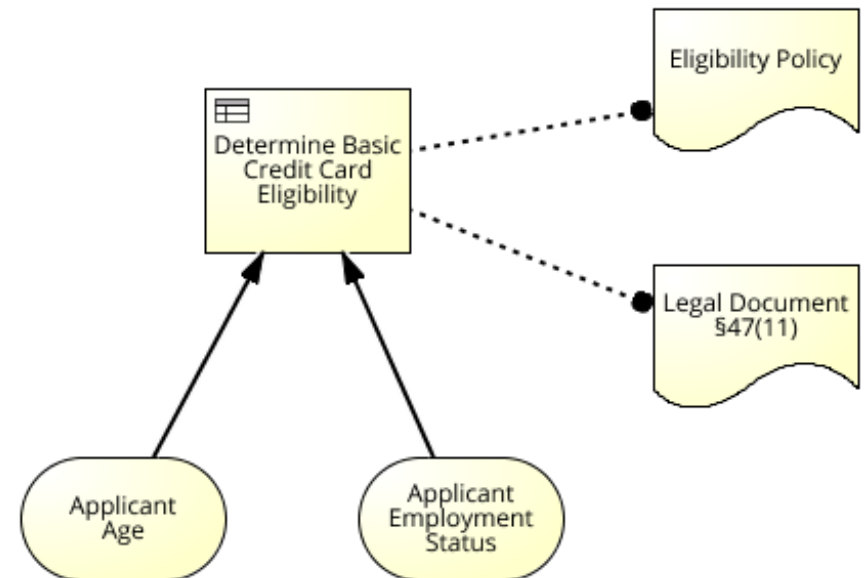
Dependencies

-  *Information Requirement*
-  *Knowledge Requirement*
-  *Authority Requirement*

Credit Card Example (1/2)

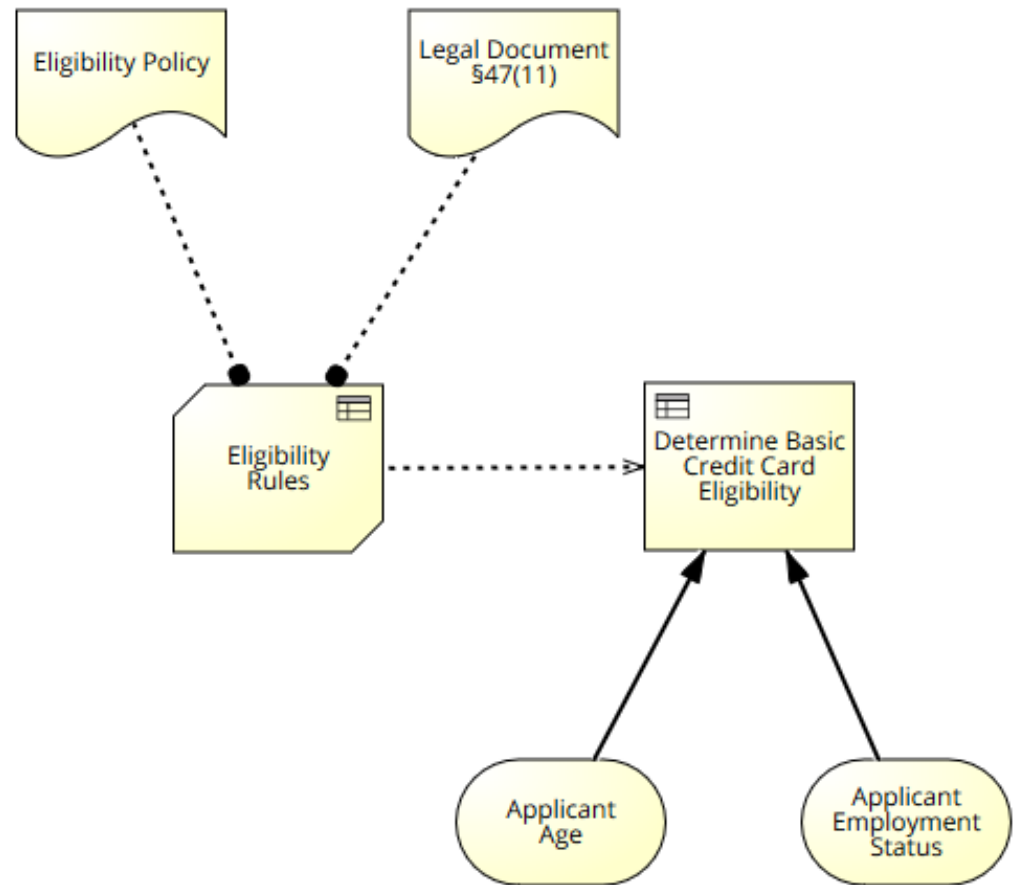


- The decision
 - uses applicant age and employment status as input
 - is based on eligibility policy of the institution and a legal document



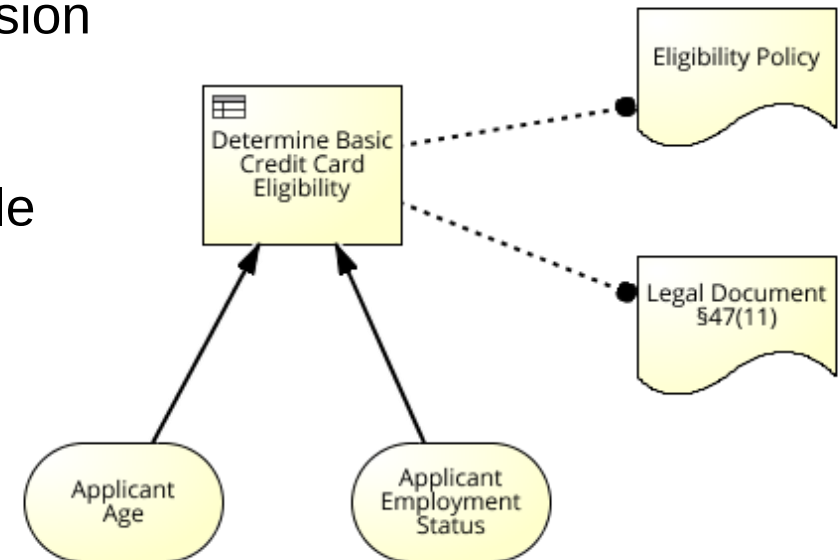
Credit Card Example (2/2)

- Example
 - Decision invokes the business knowledge *Eligibility Rules*
- Notice
 - Knowledge sources are linked to business knowledges rather than decisions
 - Because knowledge source contains decision logic



Organizational Aspects

- Decision maker
 - Person or role that takes a decision
- Decision owner
 - Person or role that is responsible for a decision



Name	value
Main Attributes	
Name	Determine Basic Credit Card Eligibility
Description	
Decision logic	
Background color	
Border color	
Question	
Answers	
More Attributes	
Linked diagram	
Decision-maker	Member CC Checking Dept
Decision-owner	Head of CC Checking Dept

Sample Scenario

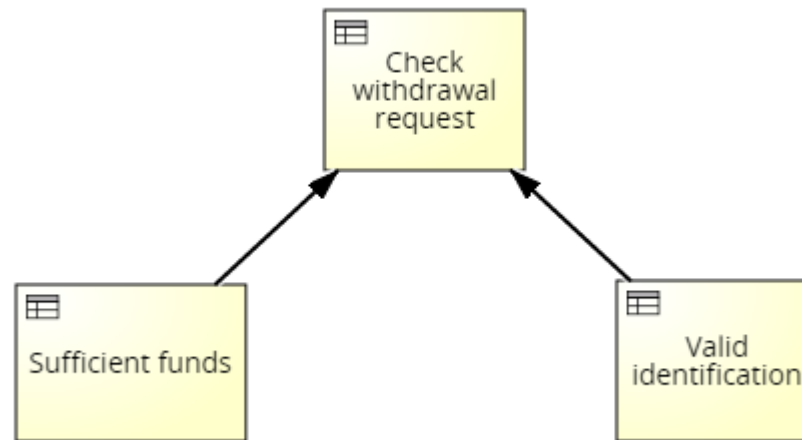
Scenario: Withdrawal of funds at a Bank

A client expresses the wish to withdraw 9000€ from his bank account.

The bank needs to take the decision whether or not the funds can be provided.

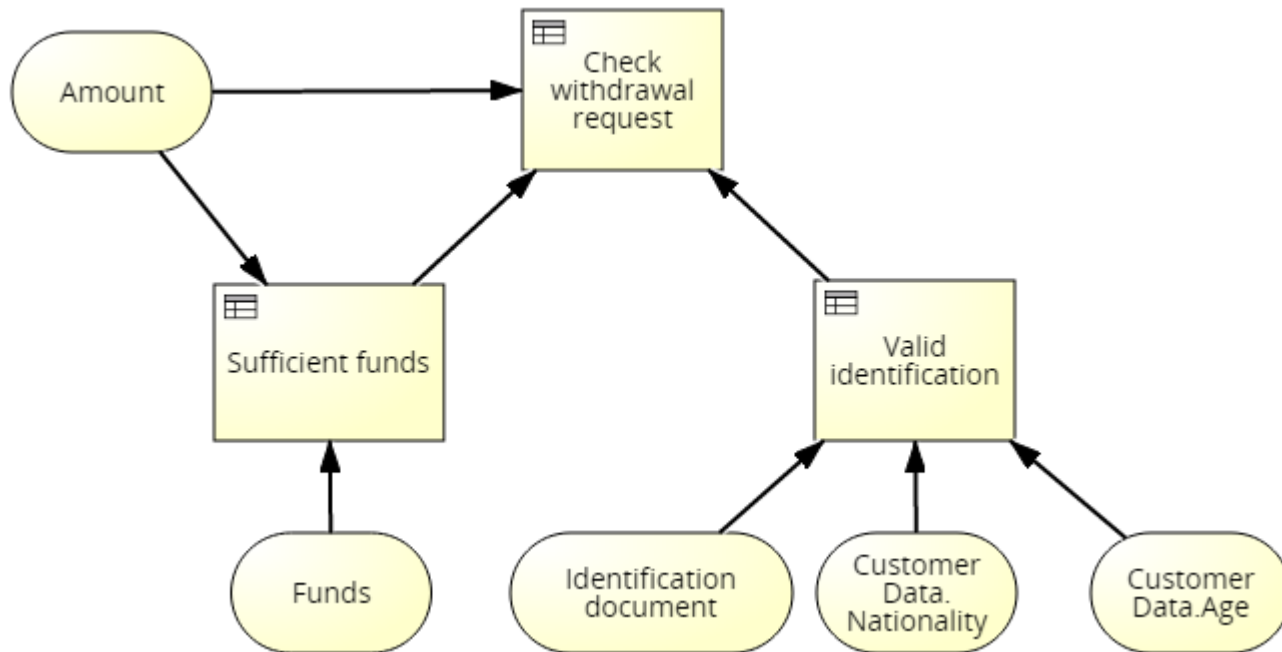
Scenario: Structure of Decisions

1. Define decisions and sub-decisions



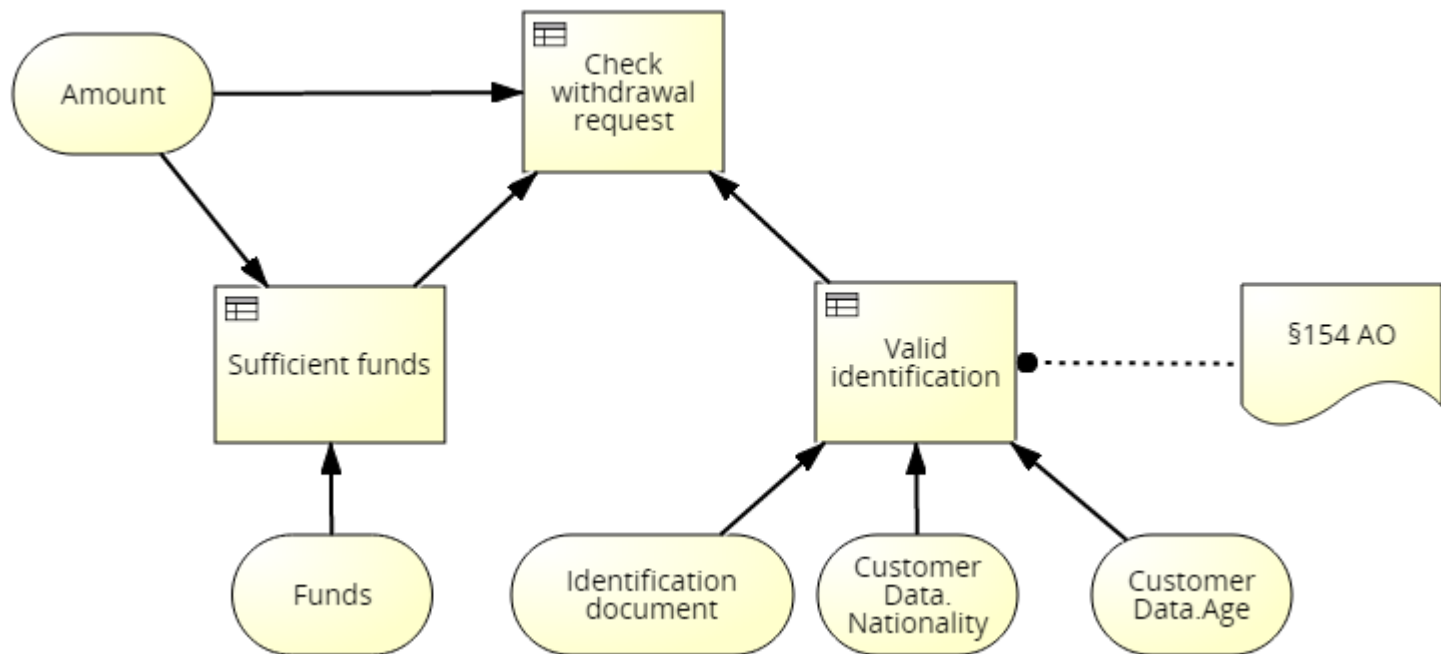
Scenario: Input Data

2. Define input data



Scenario: Knowledge Sources

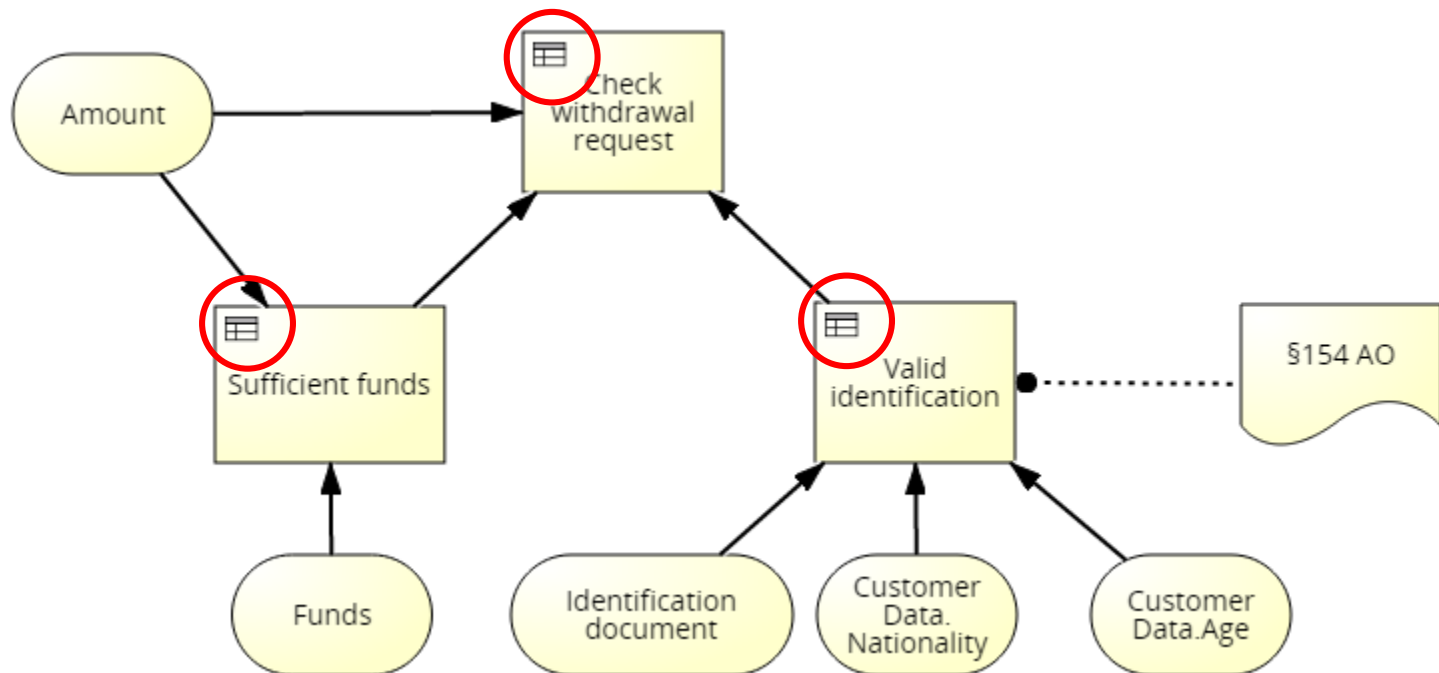
3. Define knowledge source



Scenario: Decision Logic

4. Define decision logic

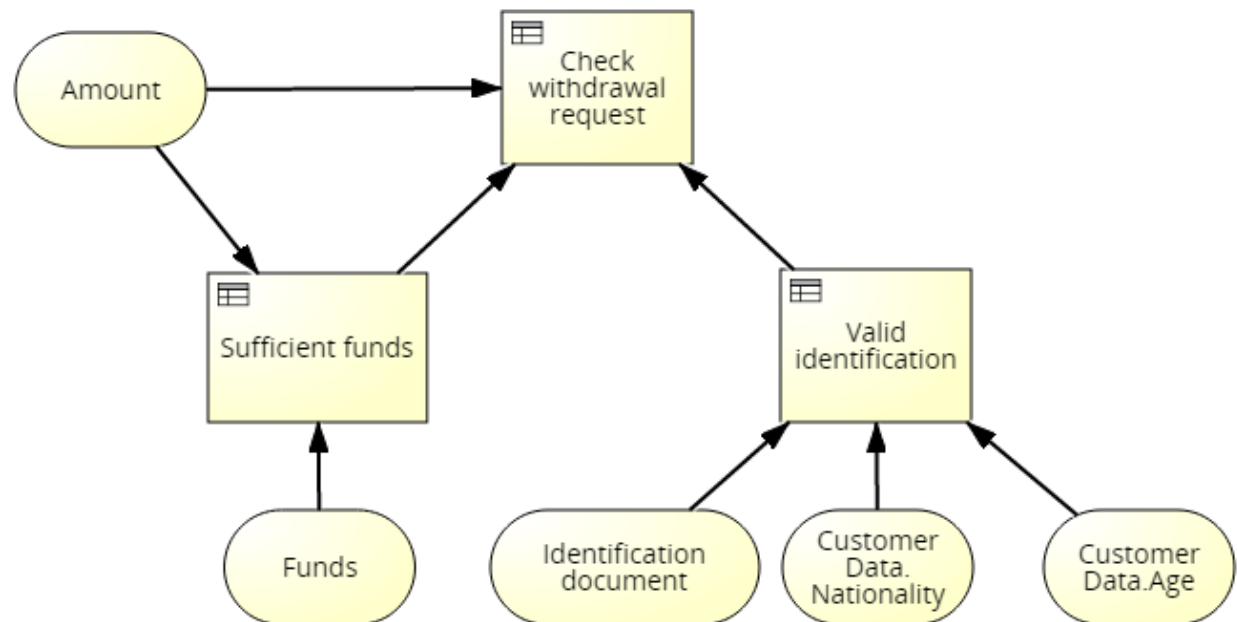
... will be done in the next video clip



Video Clip 3

Decision Requirements Diagrams

- Structure decisions and their context
 - Input data, knowledge sources, business knowledges
- Modularization of decisions
 - A decision used by another decision
 - Hierarchical structure of decisions



Video Clip 4

Decision Logic

(Business Decision Modeling)

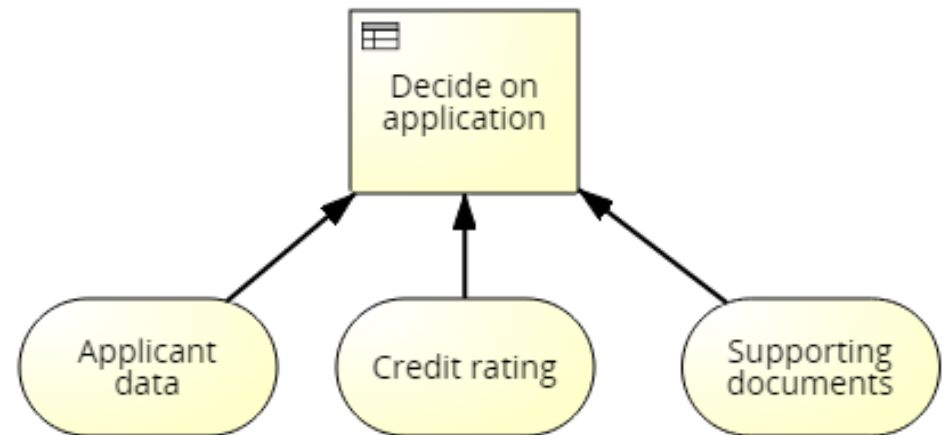
Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

Introduction to Decision Logic

- Decision Logic
 - Describes how decision output is derived from its input
 - Expressed through value expressions, which might be formal
 - Can be defined in DMN in different ways
- Literal expression
 - Textual representation
 - Program code
 - FEEL, Friendly Enough Expression Language (part of DMN)
- Decision Table
 - Serialized by FEEL
- Invocation
 - Decision calls a business knowledge by passing input values and receiving output

Textual expression

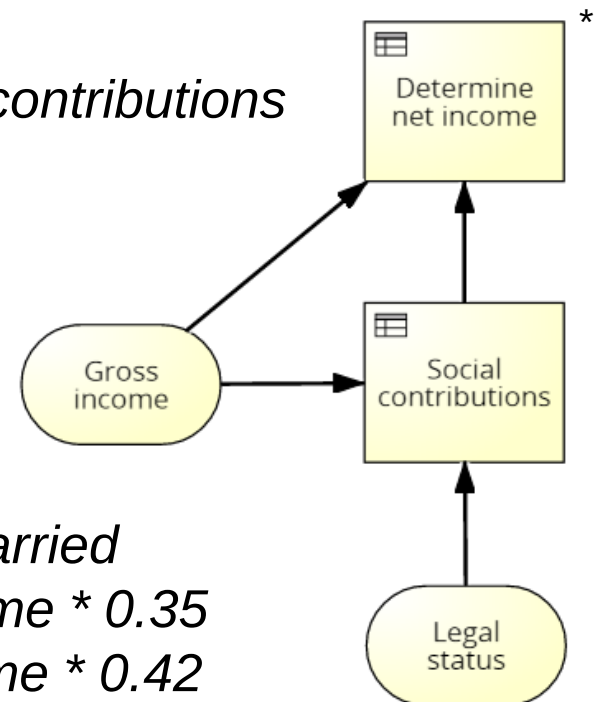
- Decision logic is provided informally by text
 - Leaves room for interpretation by human decision maker
 - DRG provides overview of the decision structure
- Example: *Application for renting a flat*
 - „*The applicant lives in a stable partnership and has a good credit rating. He does not plan to have any pets. All supporting documents are authentic. And he must be a nice person.*“



Literal expression

- Decision logic is specified in a formal manner
 - Decision is executable
 - Can be expressed in FEEL
- Example: *Calculation of net income*

Gross income – Social contributions



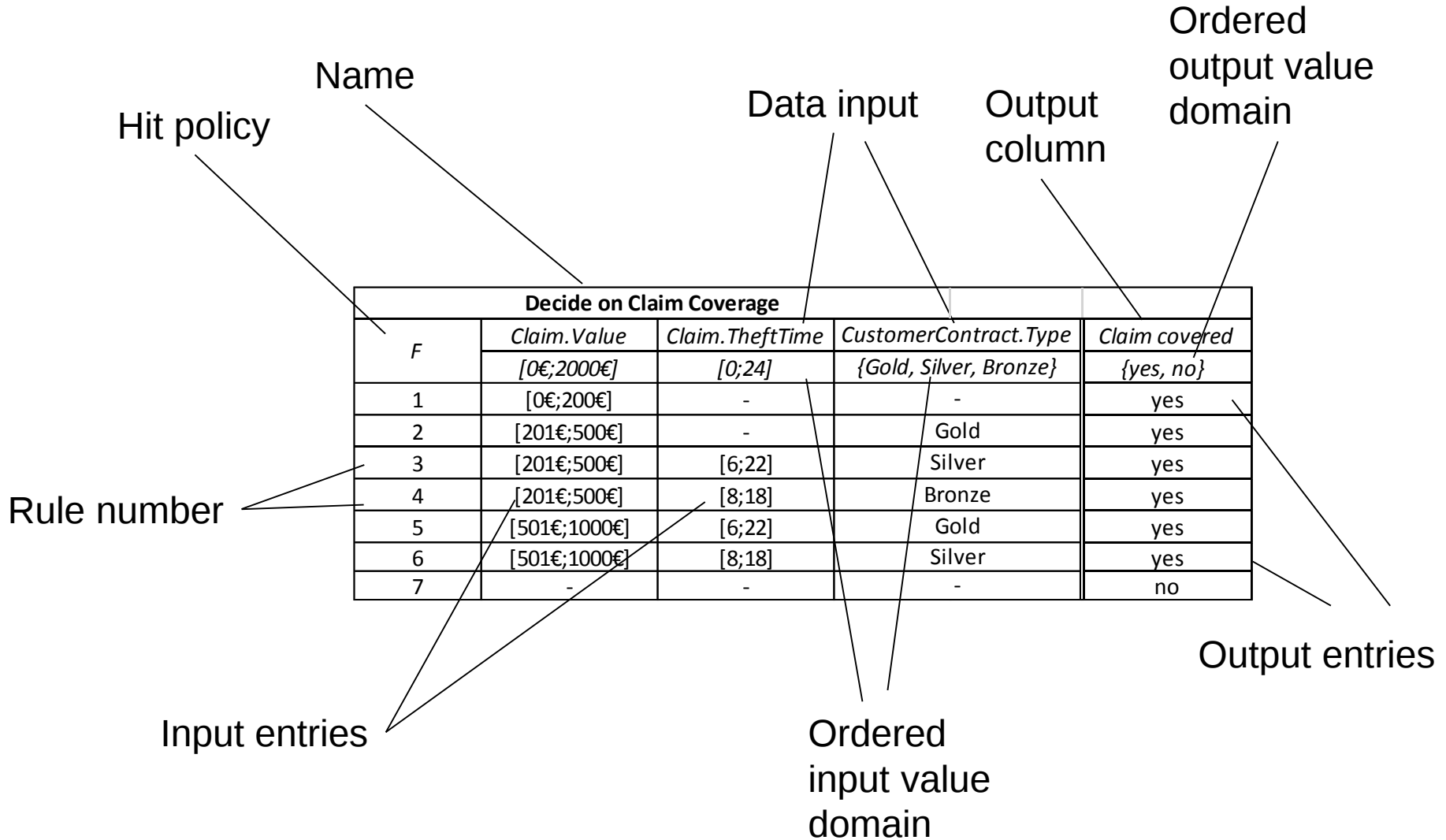
*if Legal status = married
then Gross income * 0.35
else Gross income * 0.42*

* Example is made up

Decision Tables

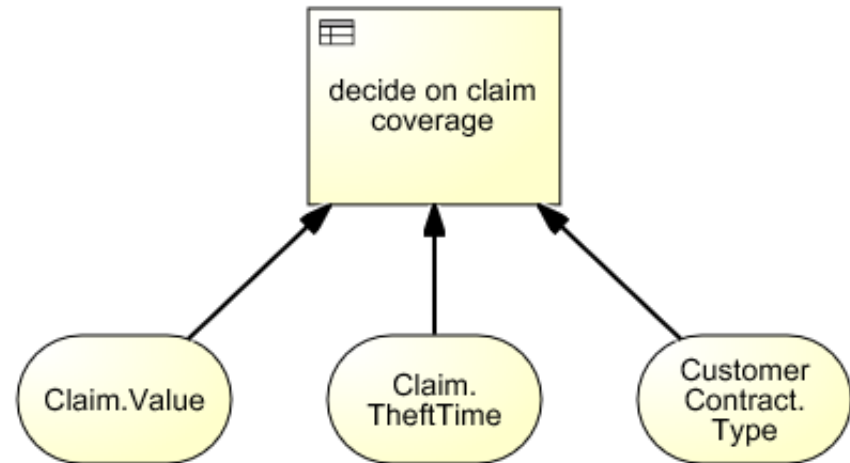
- Formal tabular specification of input/output expressions
 - Organized as rules
 - These specify how output is derived from input
- Elements of decision tables
 - Name of the table
 - Inputs (arbitrary input: „-“) with domains
 - Domains include open $]1,10[$ (or „(1,10)“) and closed $[1,10]$ intervals, simple data types and enumerations
 - Set of outputs (typically one column)
 - Set of rules, the rows of the table
 - Hit policy of the table (explained below)

Decision Table Terminology



Decision Tables

- Structure
 - One rule per row
 - Conjunctive expressions on input data
 - Disjunctions by multiple rows



Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	<i>[0€;2000€]</i>	<i>[0;24]</i>	<i>{Gold, Silver, Bronze}</i>	<i>{yes, no}</i>
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

Video Clip 4

Decision Logic

- Define decision logic
 - by text
 - literal expression
 - decision table

„The applicant lives in a stable partnership and has a good credit rating. He does not plan to have any pets. All supporting documents are authentic. And he must be a nice person.“

*if Legal status = married
 then Gross income * 0.35
 else Gross income * 0.42*

Decide on Claim Coverage				
F	Claim.Value	Claim.TheftTime	CustomerContract.Type	Claim covered
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

Video Clip 5

Semantics of Decision Tables

(Business Decision Modeling)

Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

Introduction

- Semantics of decision tables
 - How output is derived from input
 - Semantics determined by hit policy

- Single hit policies
 - Output of exactly one rule is returned

Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

- Multi hit policies
 - Output values of several rules are used
 - Several output values are returned or
 - An aggregate value is returned

Single Hit Policies

- Unique (U)
 - For any given input combination, exactly one rule matches
- Example: Health risk decision

U	Inputs		Outputs
	Age	Medical history	Health risk level
	Number	{good,bad}	{low,medium,high}
1	< 20	= good	low
2	< 20	= bad	medium
3	∈ [20..65]	-	medium
4	> 65	= good	medium
5	> 65	= bad	high

The unique policy should be used in most cases, because it guarantees that for each input combination exactly one rule matches

Single Hit Policies

- Any (A)
 - For a given combination of input values, several rules can match
 - All rules that match return the same output value
 - We can choose *any* matching rule
- Example: Credit policy compliance decision
 - For example, (not A, 10.000, 60.000) matches rules 2, 4

A	Inputs			Outputs
	Credit rating	Credit card balance	Personal loan balance	Credit policy compliance
	{A, not A}	Number	Number	{compliant, not compliant}
1	= A	< 20000	< 50000	compliant
2	= not A	-	-	not compliant
3	-	≥ 20000	-	not compliant
4	-	-	≥ 50000	not compliant

Single Hit Policies

- Priority (P)
 - For a given input, several rules can match
 - These rules might have different output values
 - Output with highest priority is returned
 - Priority determined by ordering in output domain
- Example: Health risk decision
 - Input combination (70, bad)

P	<i>Inputs</i>		<i>Outputs</i>	
	Age	Medical history	Health risk level	
	<i>Number</i>	<i>{good,bad}</i>	<i>{low,medium,high}</i>	
1	≤ 20	= good	low	
2	> 20	= good	medium	
3	> 65	= bad	high	
4	-	= bad	medium	

Single Hit Policies

- Example: Health risk decision
 - (70, bad) matches rules 3 and 4, with different outputs
 - Priority hit policy makes sure that only output *medium* is returned
 - Due to ordering of output domain

P	<i>Inputs</i>		<i>Outputs</i>	
	Age	Medical history	Health risk level	
	<i>Number</i>	<i>{good,bad}</i>	<i>{low,medium,high}</i>	
1	≤ 20	= good	low	
2	> 20	= good	medium	
3	> 65	= bad	high	
4	-	= bad	medium	

The priority policy should be avoided, because of its not very intuitive semantics

Single Hit Policies

- First (F)
 - Variant of Priority policy, in which rules are processed in order
 - The first rule that matches determines the output
- Example: Decide on Claim Coverage

Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	<i>[0€;2000€]</i>	<i>[0;24]</i>	<i>{Gold, Silver, Bronze}</i>	<i>{yes, no}</i>
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

Single Hit Policies

- Example: Claim coverage decision
 - (450, 18, Silver) matches rules 3 and 7, which have different outputs
 - Due to F policy, only rule 3 fires, and the claim is covered

Decide on Claim Coverage				
F	Claim.Value	Claim.TheftTime	CustomerContract.Type	Claim covered
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

*The first policy should only be used in cases, where
if-then-else logic is appropriate*

Single Hit Policies

- Transformation of decision tables
 - Any decision table can be transformed to the unique hit policy
 - Typically, the resulting table has more rows than the original one

Decide on Claim Coverage				
<i>U</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	]22;6[	Silver	no
5	[201€;500€]	[8;18]	Bronze	yes
6	[201€;500€]	]18;8[	Bronze	no
7	[501€;1000€]	[6;22]	Gold	yes
8	[501€;1000€]	]22;6[	Gold	no
9	[501€;1000€]	[8;18]	Silver	yes
10	[501€;1000€]	]18;8[	Silver	no
11	[501€;1000€]	-	Bronze	no
12	> 1000€	-	-	no

Multi Hit Policies

- Multi hit
 - Several matching rules can be used for determining the output
- Output order (O)
 - Returns all hits in decreasing output priority order
 - Output priorities are specified in the ordered list of output values in decreasing order of priority

- Input (50, 31)
 - Rules 1, 2, 4, 5, 6 fire
- Output priority leads to (2, 2, 3, 4, 26)

Decide on Leave Days			
<i>O</i>	<i>Age</i>	<i>Years of Service</i>	<i>LeaveDays</i>
	<i>Number</i>	<i>Number</i>	<i>Number</i>
1	-	-	26
2	> 30	-	2
3	> 60	-	3
4	-	> 10	2
5	-	> 20	3
6	-	> 30	4

Multi Hit Policies

- Rule order (R)
 - Outputs of all matching rules are returned as a sorted list, in the order of the rules

- Input (50, 31)
 - Rules 1, 2, 4, 5, 6 fire
- Rule order leads to (26, 2, 2, 3, 4)

Decide on Leave Days			
<i>R</i>	<i>Age</i>	<i>Years of Service</i>	<i>LeaveDays</i>
	<i>Number</i>	<i>Number</i>	<i>Number</i>
1	-	-	26
2	> 30	-	2
3	> 60	-	3
4	-	> 10	2
5	-	> 20	3
6	-	> 30	4

Multi Hit Policies

- Collect (C)
 - Outputs of all matching rules are returned as unsorted list
 - Aggregation can be applied
(*sum +, min <, max >, count #*)

- Input (50, 31)
 - Rules 1, 2, 4, 5, 6 fire
- C+ leads to 37

Decide on Leave Days			
C+	Age	Years of Service	LeaveDays
	Number	Number	Number
1	-	-	26
2	> 30	-	2
3	> 60	-	3
4	-	> 10	2
5	-	> 20	3
6	-	> 30	4

Video Clip 5

Semantics of Decision Tables

- Single hit policies

- Unique
- Any
- Priority
- First

Decide on Claim Coverage				
<i>F</i>	<i>Claim.Value</i>	<i>Claim.TheftTime</i>	<i>CustomerContract.Type</i>	<i>Claim covered</i>
	[0€;2000€]	[0;24]	{Gold, Silver, Bronze}	{yes, no}
1	[0€;200€]	-	-	yes
2	[201€;500€]	-	Gold	yes
3	[201€;500€]	[6;22]	Silver	yes
4	[201€;500€]	[8;18]	Bronze	yes
5	[501€;1000€]	[6;22]	Gold	yes
6	[501€;1000€]	[8;18]	Silver	yes
7	-	-	-	no

- Multi hit policies

- Output order
- Rule order
- Collect
 - + < > #

Decide on Leave Days			
<i>C+</i>	<i>Age</i>	<i>Years of Service</i>	<i>LeaveDays</i>
	<i>Number</i>	<i>Number</i>	<i>Number</i>
1	-	-	26
2	> 30	-	2
3	> 60	-	3
4	-	> 10	2
5	-	> 20	3
6	-	> 30	4

Video Clip 6

Analysis of Decision Tables

(Business Decision Modeling)

Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

Analysis of Decision Tables

- Decision tables are important assets
 - Their quality should be as high as possible
 - Therefore we investigate analysis techniques
- Completeness
 - Are all possible input combinations actually covered by the decision table?
- Consistency
 - Is the decision table in line with its hit policy?

Determine Health Risk Level			
U	Age	Medical History	Health Risk Level
1	< 20	good	low
2		bad	medium
3	[20 .. 65]	–	medium
4	> 65	good	medium
5		bad	high

Completeness of DecisionTables

- Completeness, more concrete
 - For all combinations of input values, is there a rule that covers it?
- Example 1
 - For $Age=\{1..100\}$ and $MedicalHistory=\{good,bad\}$, all combinations of the set $Age \times MedicalHistory$ have to be covered

Determine Health Risk Level			
U	Age	Medical History	Health Risk Level
1	< 20	good	low
2		bad	medium
3	[20 .. 65]	–	medium
4	> 65	good	medium
5		bad	high

Decision table is complete, because all combinations are covered

Completeness of DecisionTables

- Example 2
 - For $Age=\{1..100\}$ and $MedicalHistory=\{good,bad\}$, all combinations of the set $Age \times MedicalHistory$ have to be covered

Determine Health Risk Level			
p	Age	Medical History	Health Risk Level
			high, medium, low
1	< 20	good	low
2	> 20	good	medium
3	> 65	bad	high
5	–	bad	medium

Decision table is not complete, since the case $Age=20$ and $MedicalHistory=good$ is not covered

Completeness of DecisionTables

- Incomplete Decision Tables

- When a process with uncovered values is executed, the decision cannot be taken
- Completeness makes sure that a decision can always be taken

Determine Health Risk Level			
p	Age	Medical History	Health Risk Level
			<i>high, medium, low</i>
1	< 20	good	low
2	> 20	good	medium
3	> 65	bad	high
5	–	bad	medium

- Remarks

- It is good practice to check table completeness
- Tools provide automatic completeness checking

Consistency of Decision Tables

- Single hit properties can be analyzed
- Unique property
 - Violated if there is an input combination that matches two rows
- Example 1

U	<i>Inputs</i>		<i>Outputs</i>
	Age	Medical history	Health risk level
	<i>Number</i>	<i>{good,bad}</i>	<i>{low,medium,high}</i>
1	< 20	= good	low
2	< 20	= bad	medium
3	∈ [20..65]	-	medium high
4	> 65	= good	medium
5	> 65	= bad	high

Unique property is satisfied

Consistency of Decision Tables

- Example 2
 - Is the Unique property now violated?

U	Inputs		Outputs	
	Age	Medical history	Health risk level	
	Number	{good,bad}	{low,medium,high}	
1	< 20	= good	low	
2	< 20	= bad	medium	
3	∈ [20..65]	-	medium	
4	> 65	= good	medium	
5	> = 65	= bad	high	

Unique property is violated, because there is an input combination (65,bad) for which rules 3 and 5 match

Consistency of Decision Tables

- Any property
 - Violated if a given input combination matches two rows with different outputs
- Is the Any property violated?

A	Inputs		Outputs	
	Age	Medical history	Health risk level	
	Number	{good,bad}	{low,medium,high}	
1	< 20	= good	low	
2	< 20	= bad	medium	
3	∈ [20..65]	-	medium	
4	> 65	= good	medium	
5	> = 65	= bad	high	

Yes, since there are outputs (high, medium) for (65,bad), matching rules 3 and 5

Consistency of Decision Tables

- Can Priority be violated as well?

P	Inputs			Outputs
	Age	Medical history		Health risk level
	Number	{good,bad}		{low,medium,high}
1	<	20	= good	low
2	<	20	= bad	medium
3	∈	[20..65]	-	medium
4	>	65	= good	medium
5	>=	65	= bad	high

Actually not, since even if several rows match, a single one will be chosen:

For (65,bad), rules 3 will be chosen

Consistency of Decision Tables

- Example 3
 - Is Any property now violated?

A	<i>Inputs</i>		<i>Outputs</i>	
	Age	Medical history	Health risk level	
	<i>Number</i>	<i>{good,bad}</i>	<i>{low,medium,high}</i>	
1	< 20	= good	low	
2	< 20	= bad	medium	
3	∈ [20..65]	= -	medium	
4	> = 65	= good	medium	
5	> 65	= bad	high	

No, since there is just one output (medium) for (65,good), matching rules 3 and 4.

Video Clip 6

Analysis of Decision Tables

- Completeness of Decision Tables

Determine Health Risk Level			
p	Age	Medical History	Health Risk Level
			<i>high, medium, low</i>
1	< 20	good	low
2	> 20	good	medium
3	> 65	bad	high
5	–	bad	medium

- Consistency of Decision Tables

- Unique
- Any
- Priority / First

U	<i>Inputs</i>			<i>Outputs</i>
	Age	Medical history		Health risk level
	<i>Number</i>	<i>{good,bad}</i>		<i>{low,medium,high}</i>
1	< 20	=	good	low
2	< 20	=	bad	medium
3	∈ [20..65]		-	medium high
4	> 65	=	good	medium
5	> =	=	bad	high

Video Clip 7

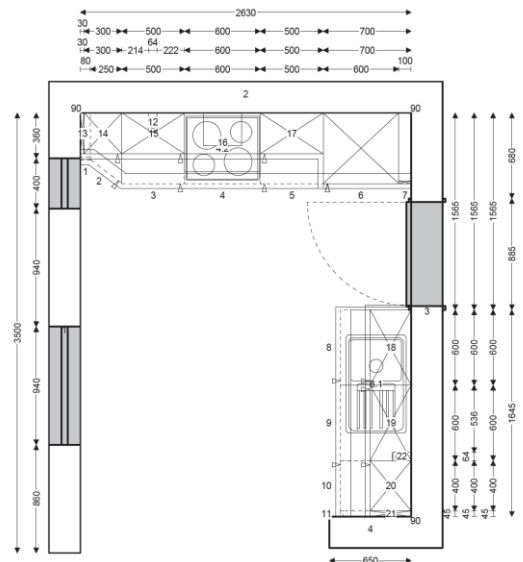
Consistency of Processes and Decisions

(Business Decision Modeling)

Business Processes: Modeling, Simulation, Execution
An openHPI Course by Mathias Weske

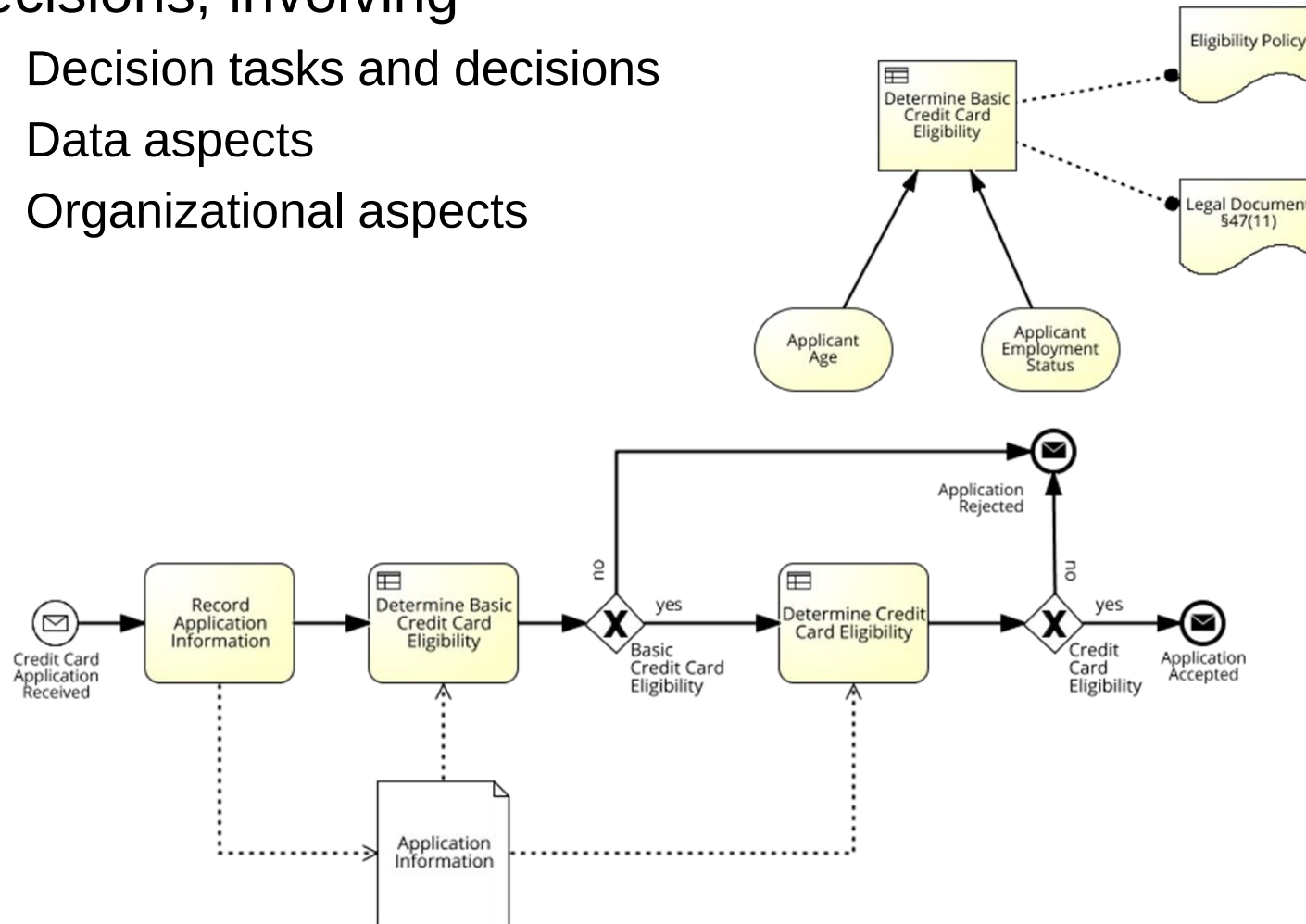
Introduction

- A business process with decisions can be represented by
 - Process model (BPMN process diagram)
 - Decision requirements model (DMN DRG)
 - Decision logic model (DMN Decision Table)
- Modeling theory: One original – several models
 - These models need to be consistent
 - There must not be any contradictions between them



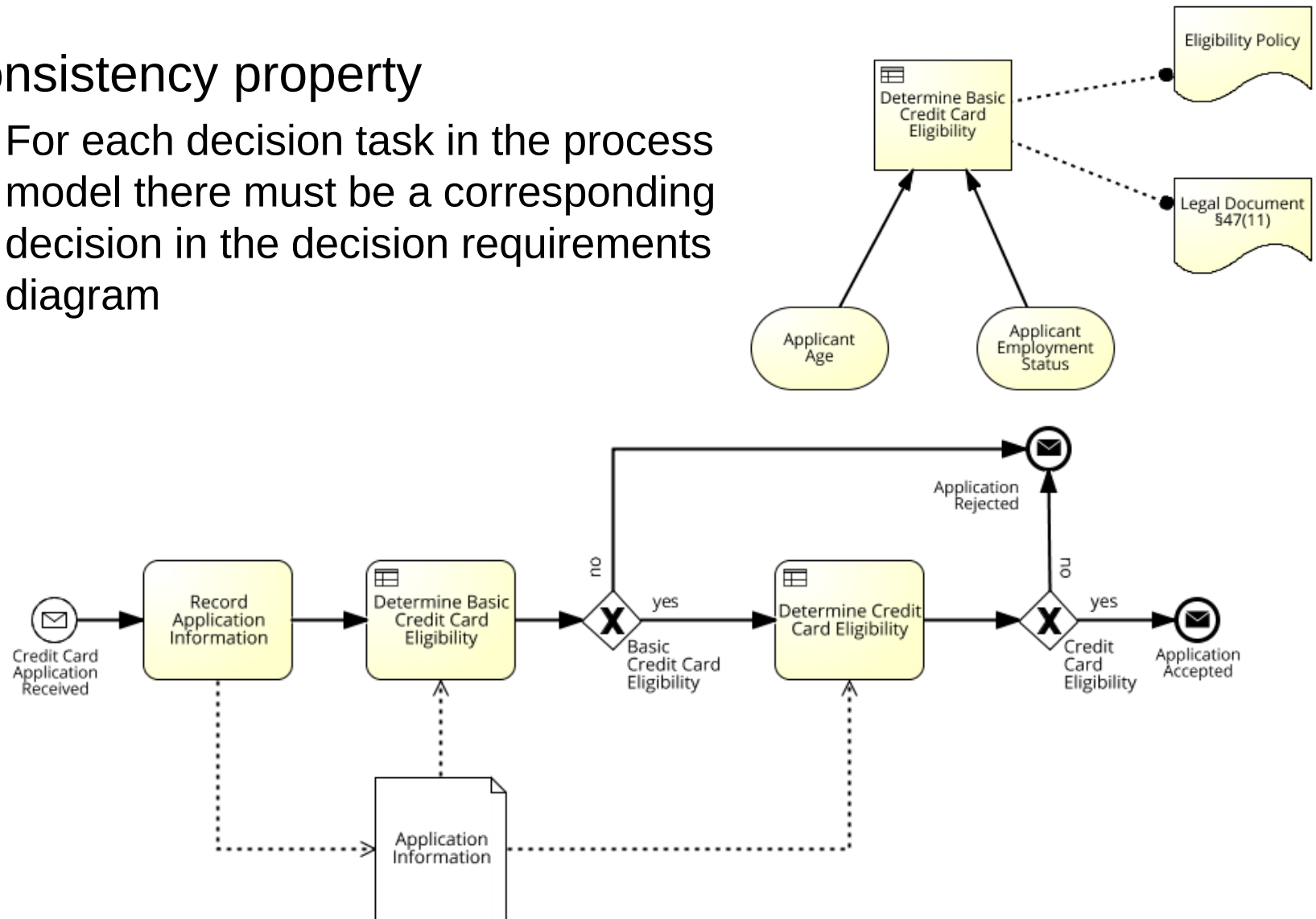
Structural Consistency

- We investigate consistency related to the structure of decisions, involving
 - Decision tasks and decisions
 - Data aspects
 - Organizational aspects



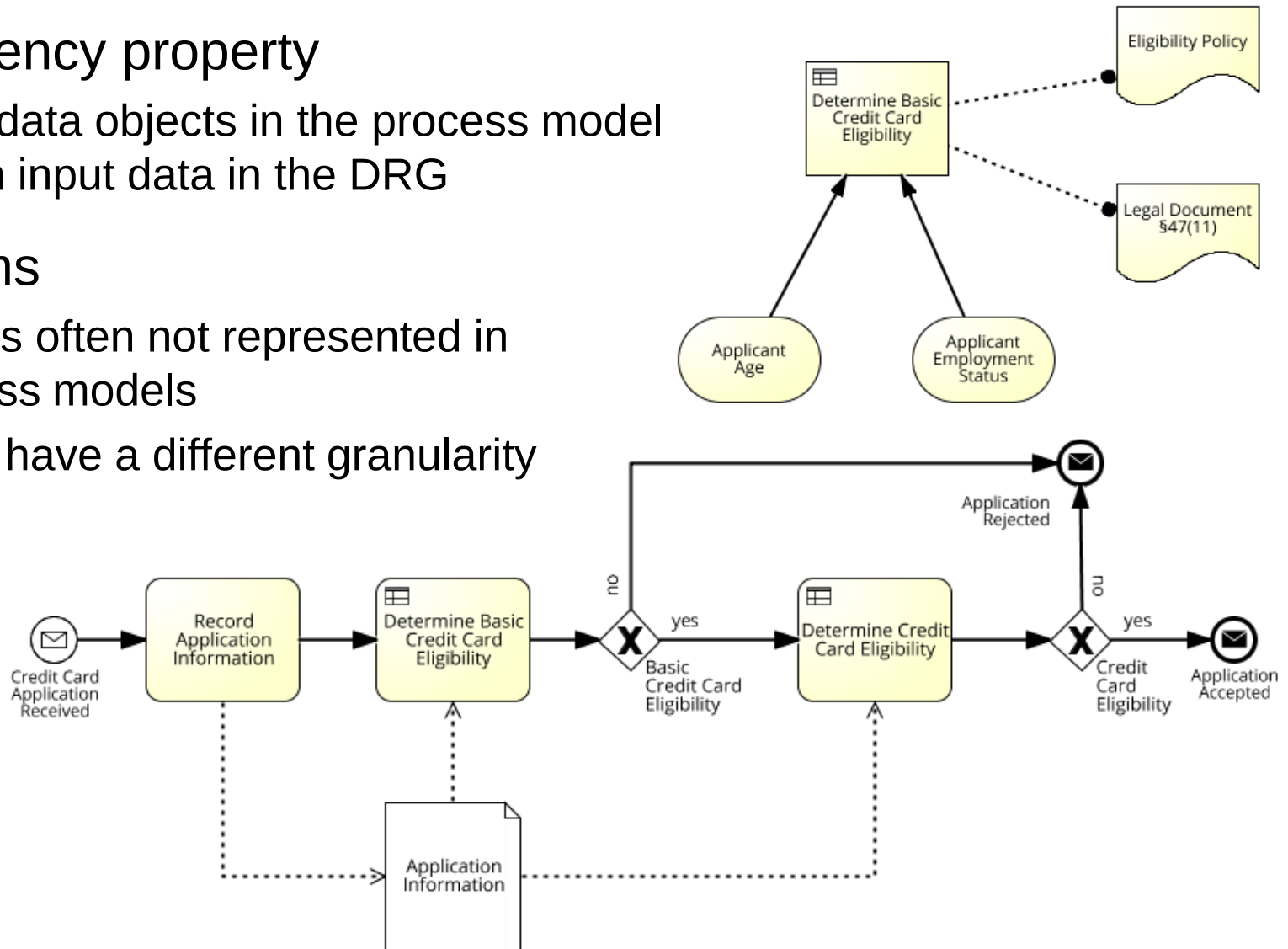
Decision Tasks and Decisions

- Consistency property
 - For each decision task in the process model there must be a corresponding decision in the decision requirements diagram



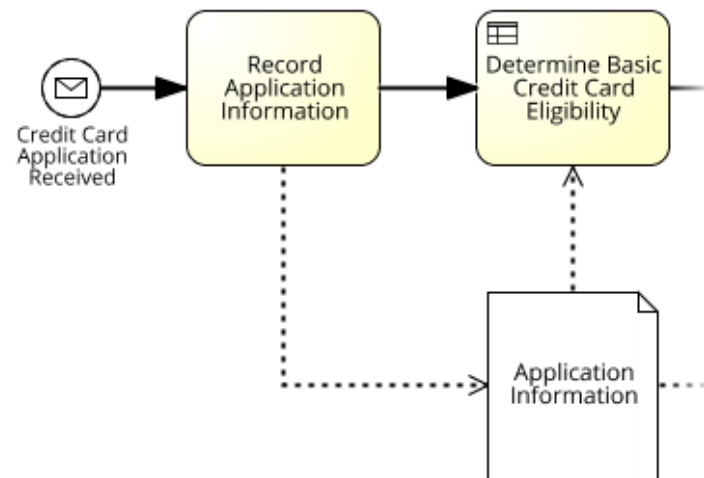
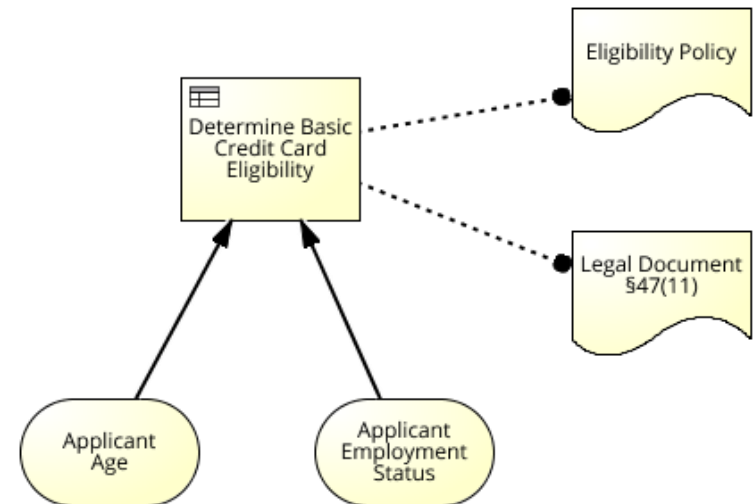
Data Aspects

- Consistency property
 - Input data objects in the process model match input data in the DRG
- Problems
 - Data is often not represented in process models
 - Might have a different granularity



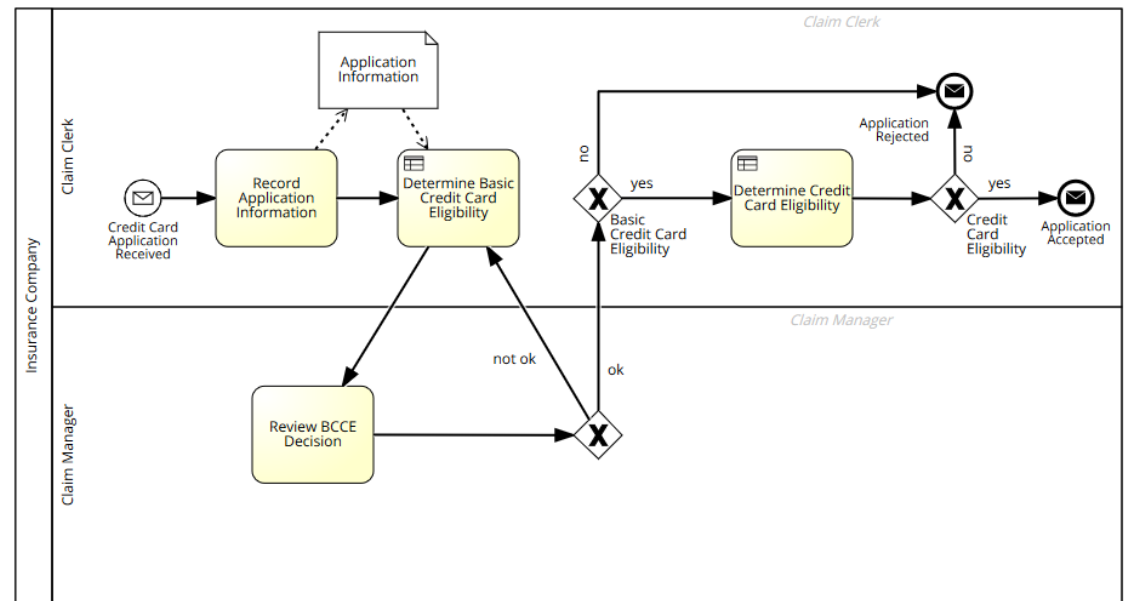
Data Aspects

- Approach
 - Use data models, like UML class diagrams, to align data objects with decision inputs
 - Make sure all data is available when decision task is enabled
- Models facilitate communication
 - Process models and decision models expose the need to align data objects with decision inputs



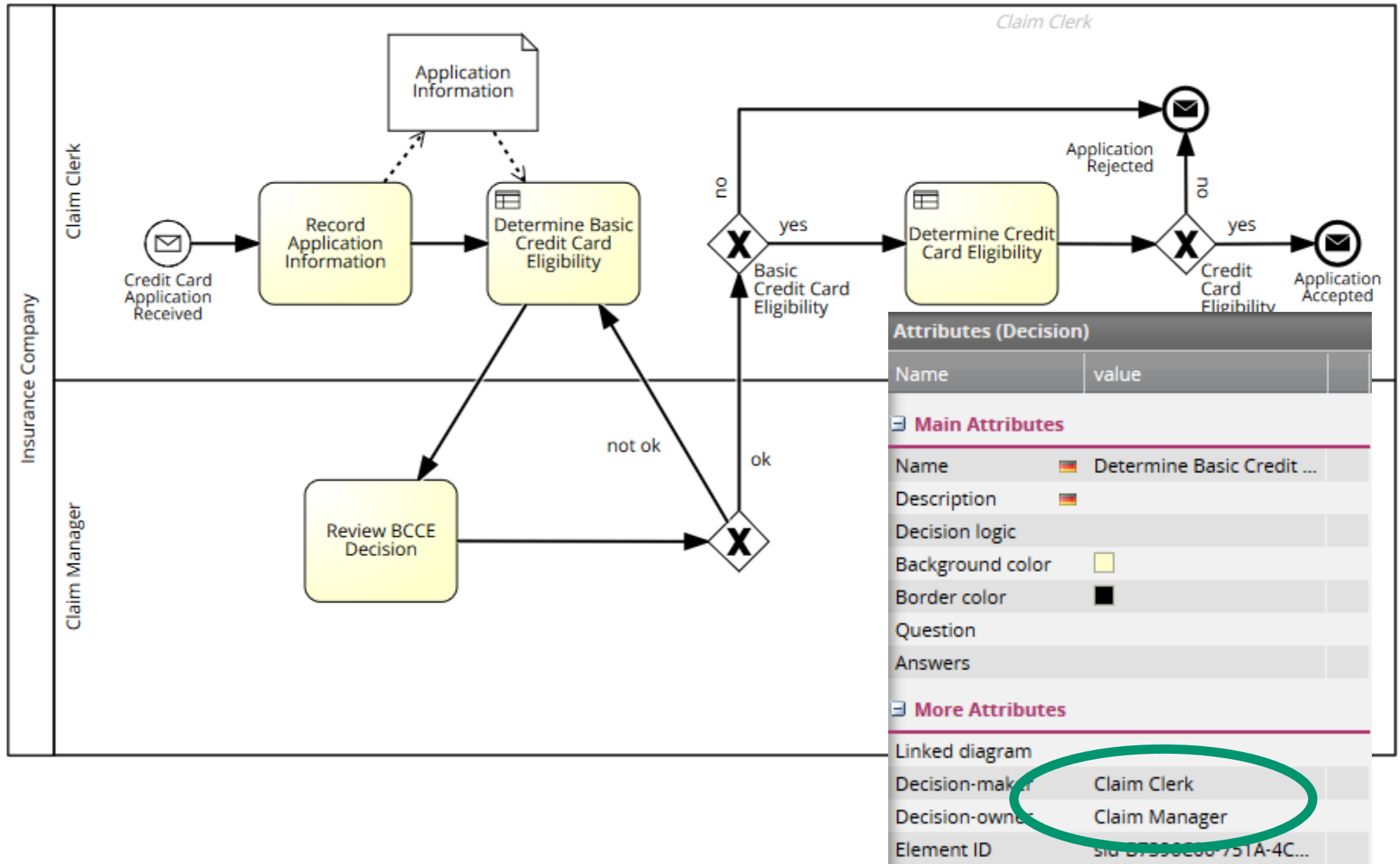
Organizational Aspects

- Defining responsibilities
 - Lanes in process models assign responsibilities
 - Organizational aspects can also be defined in decision models



- Consistency property
 - The resource identified by the lane of a decision task matches the decision maker

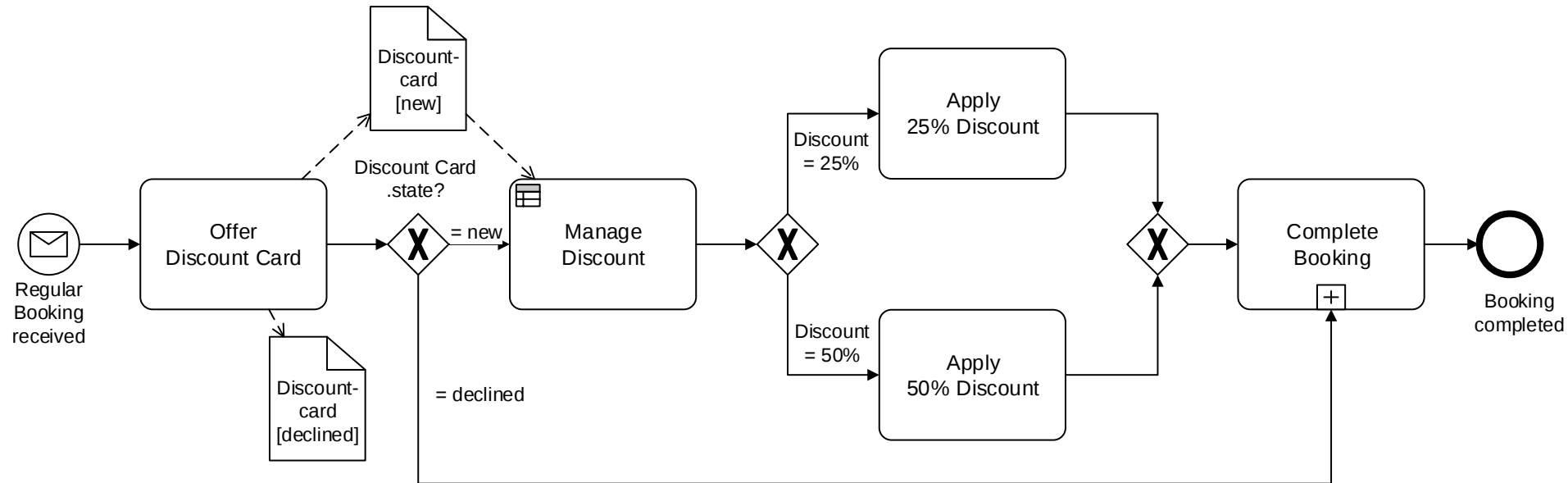
Organizational Aspects



Behavioural Consistency

- Investigate if decision output is consistent with the process behaviour
- *Output coverage*
 - All outputs of the decision can be processed by the process
- *Dead branch absence*
 - Each branch in the process is represented by at least one possible output in the decision model
- Remark
 - These concepts will be illustrated by a fictional train ticket purchasing scenario

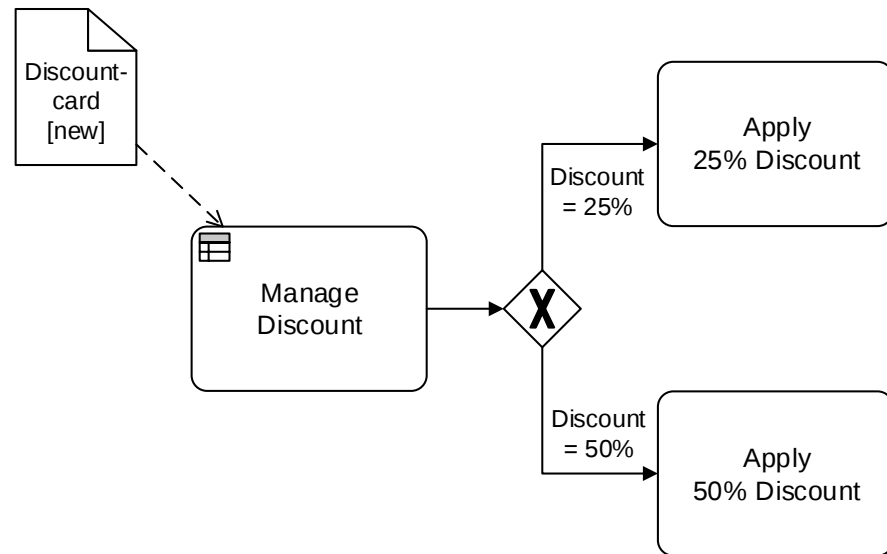
Booking a Train Ticket



Manage Discount			
U	<i>Discountcard.type</i>	<i>Discountcard.points</i>	<i>Discount</i>
	{25, 50}	Number	{25%, 30%, 50%, 60%}
1	25	< 1000	25%
2	50	< 1000	50%
3	25	>= 1000	30%
4	50	>= 1000	60%

Output Coverage Criterion

- All outputs of the decision can be processed

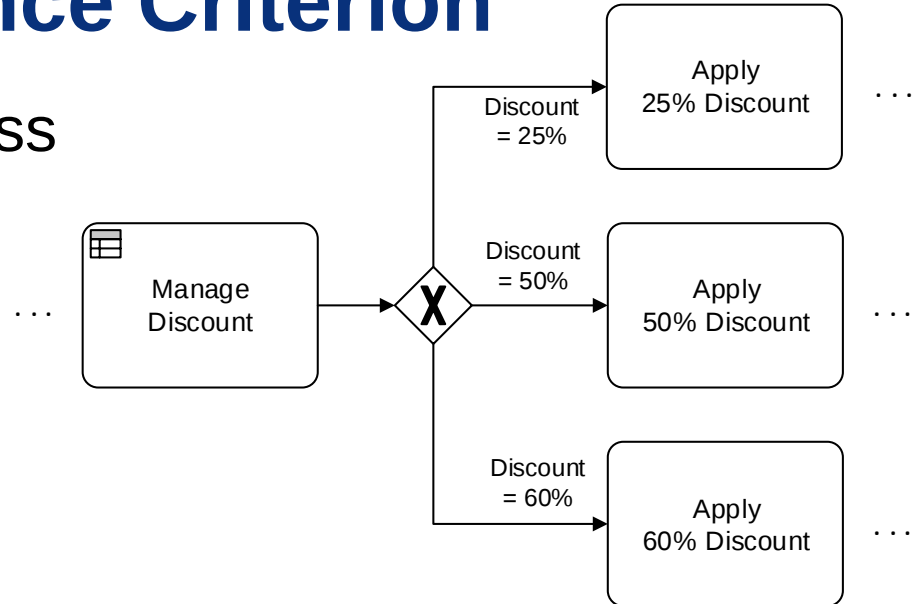


Manage Discount			
U	Discountcard.type	Discountcard.points	Discount
	{25, 50}	Number	{25%, 30%, 50%, 60%}
1	25	< 1000	25%
2	50	< 1000	50%
3	-	>= 1000	60%

Output coverage criterion violated, because there is no branch in the process model for output 60%

Dead Branch Absence Criterion

- Each branch in the process is represented by at least one row in the decision model



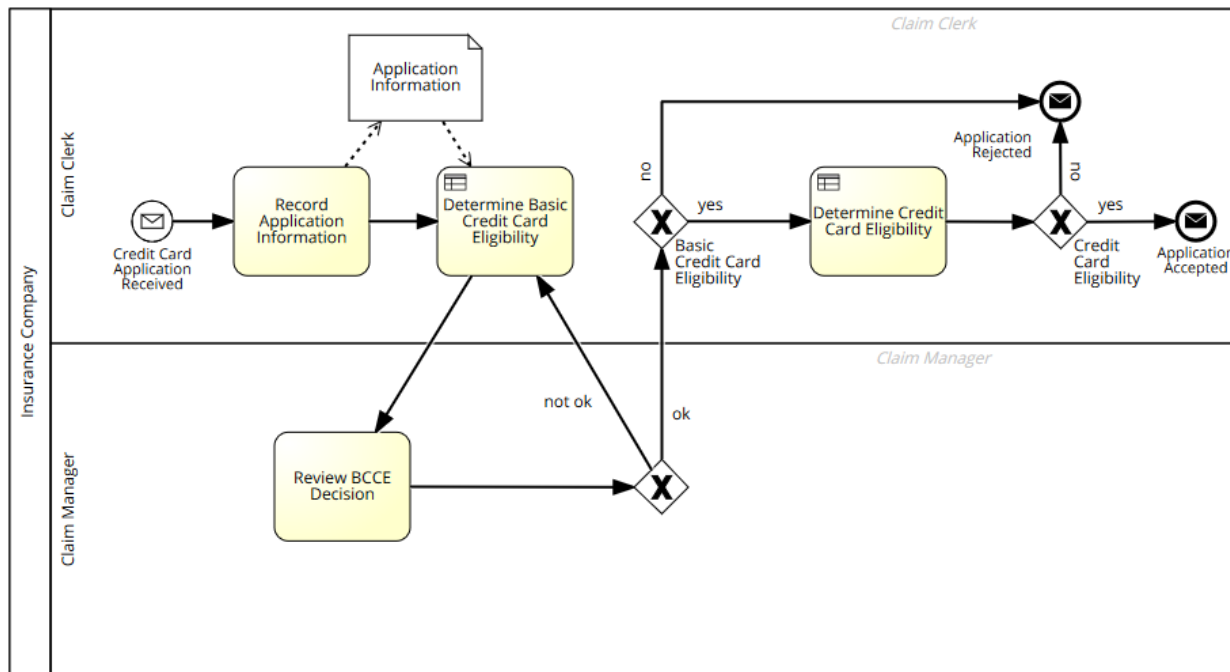
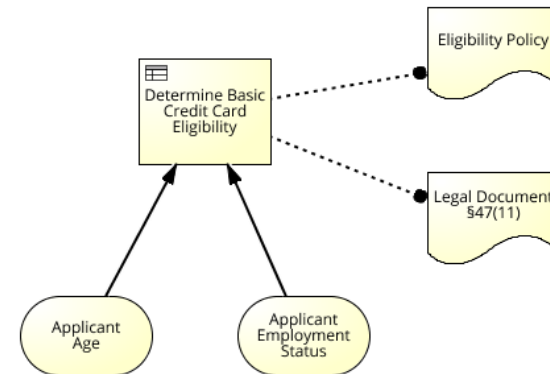
Manage Discount			
U	Discountcard.type	Discountcard.points	Discount
	{25, 50}	Number	{25%, 30%, 50%, 65%}
1	25	< 1000	25%
2	50	< 1000	50%
3	-	>= 1000	65%

*Dead branch absence criterion is violated, because
There is no row in the decision table for the branch 60%*

Video Clip 7

Consistency of Processes and Decisions

- Structural
 - Activities and decisions
 - Data aspects
 - Organizational aspects

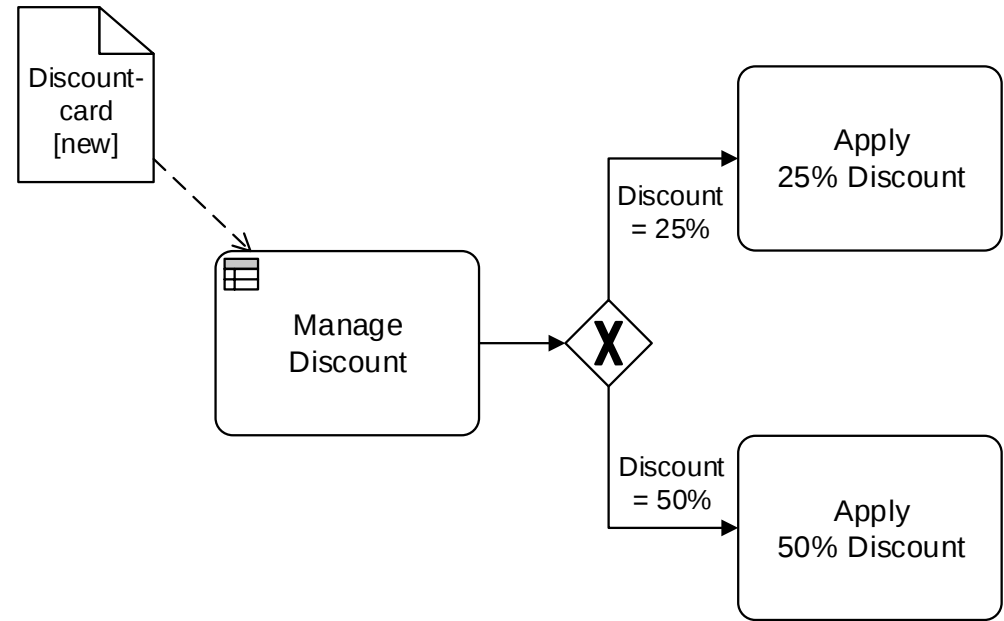


Attributes (Decision)	
Name	value
Main Attributes	
Name	Determine Basic Credit ...
Description	
Decision logic	
Background color	
Border color	
Question	
Answers	
More Attributes	
Linked diagram	
Decision-maker	Claim Clerk
Decision-owner	Claim Manager
Element ID	sid-B7396C06-751A-4C...

Video Clip 7

Consistency of Processes and Decisions

- Behavioural
 - Decisions and branches in the process
 - Output coverage
 - Dead branch absence

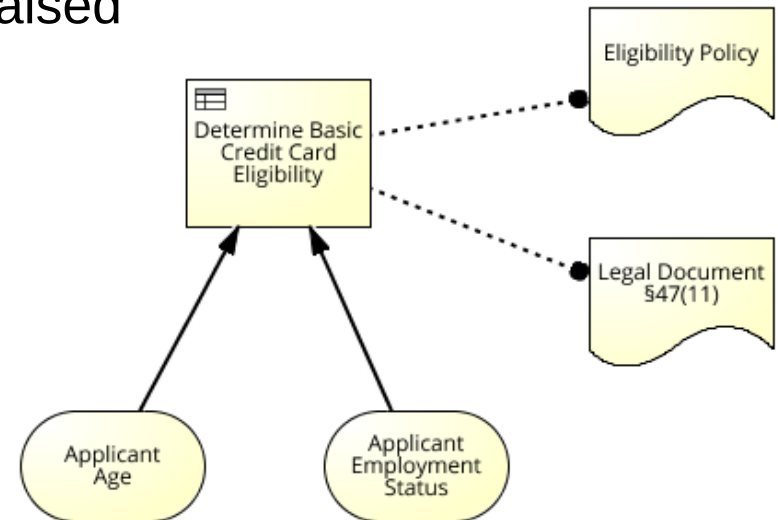


Manage Discount			
U	<i>Discountcard.type</i>	<i>Discountcard.points</i>	<i>Discount</i>
	{25, 50}	Number	{25%, 30%, 50%, 70%}
1	25	>1000	30%
2	50	< 1000	50%
3	50	>= 2000	70%

Summary

Business Decision Modeling

- DMN provides answers to the questions raised
 - How are decisions taken?
 - Which data is being used?
 - Which policies are taken into account?
 - Who is responsible for the decision?



- Massive interest by industry, e.g. finance
 - Main drivers: Auditing and compliance requirements
 - Moving processes closer to implementation („help reducing Business-IT Gap“)

Decide on Leave Days			
C+	Age	Years of Service	LeaveDays
	Number	Number	Number
1	-	-	26
2	> 30	-	2
3	> 60	-	3
4	-	> 10	2
5	-	> 20	3
6	-	> 30	4