BPMN Meets DMN:
Business Process and Decision Modeling

Week 6:
Business Decision Modeling

An openHPI Course

Prof. Dr. Mathias Weske

Hasso Plattner Institute at the University of Potsdam
Business Process Technology

Week 6
Business Decision Modeling

- 6.1 Motivation
- 6.2 DMN Overview
- 6.3 Decision Requirements Diagrams
- 6.4 Decision Logic
- 6.5 Semantics of Decision Tables
- 6.6 Analysis of Decision Tables
- 6.7 Consistency of Processes and Decisions

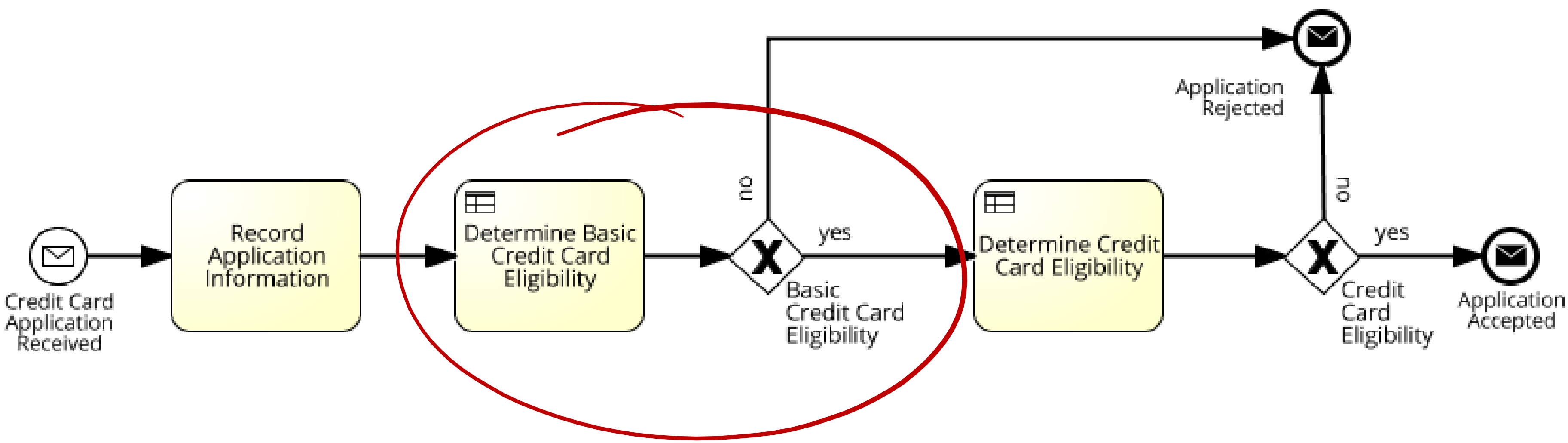
Video Clip 6.1

Motivation

Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

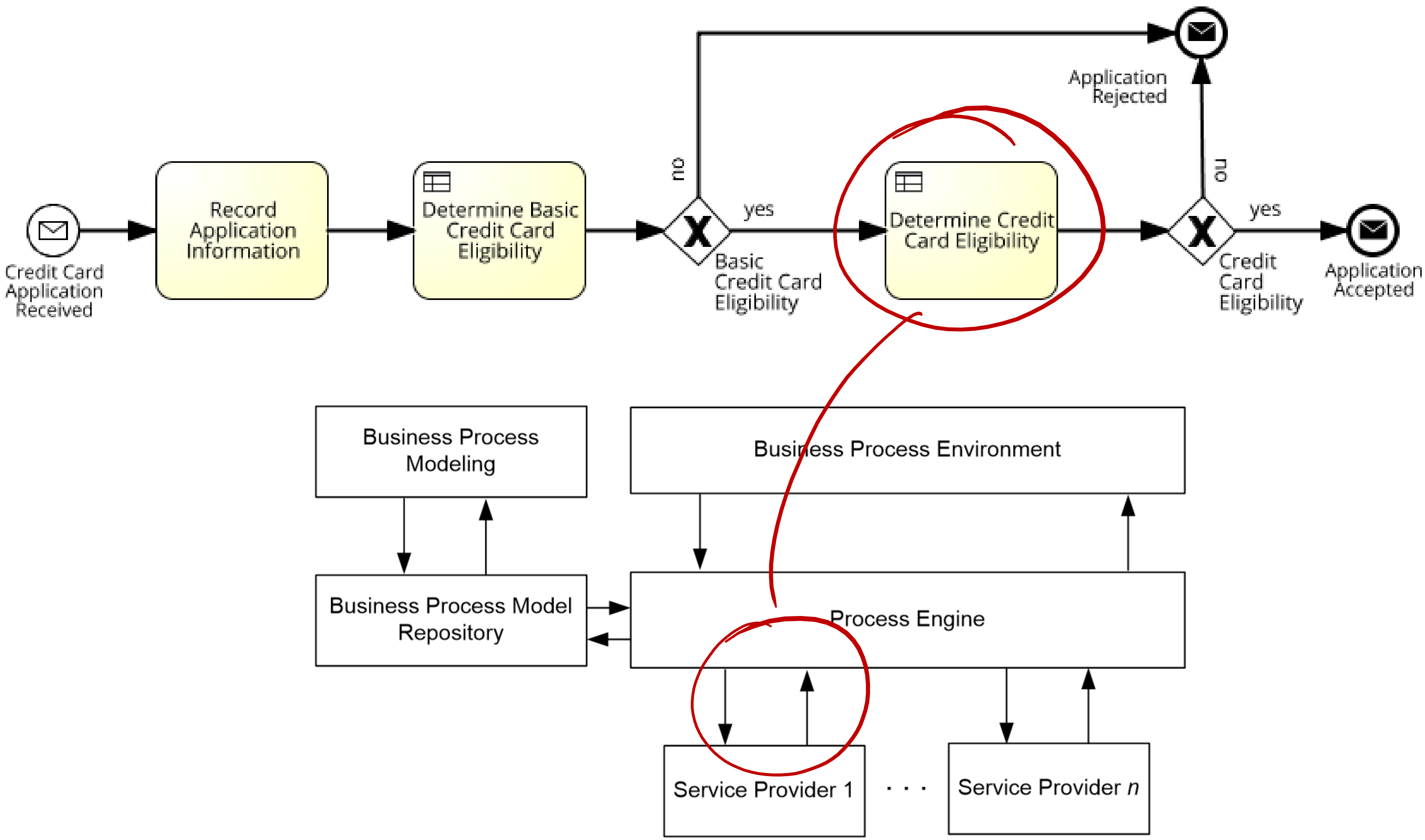
6.1 Motivation

- 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



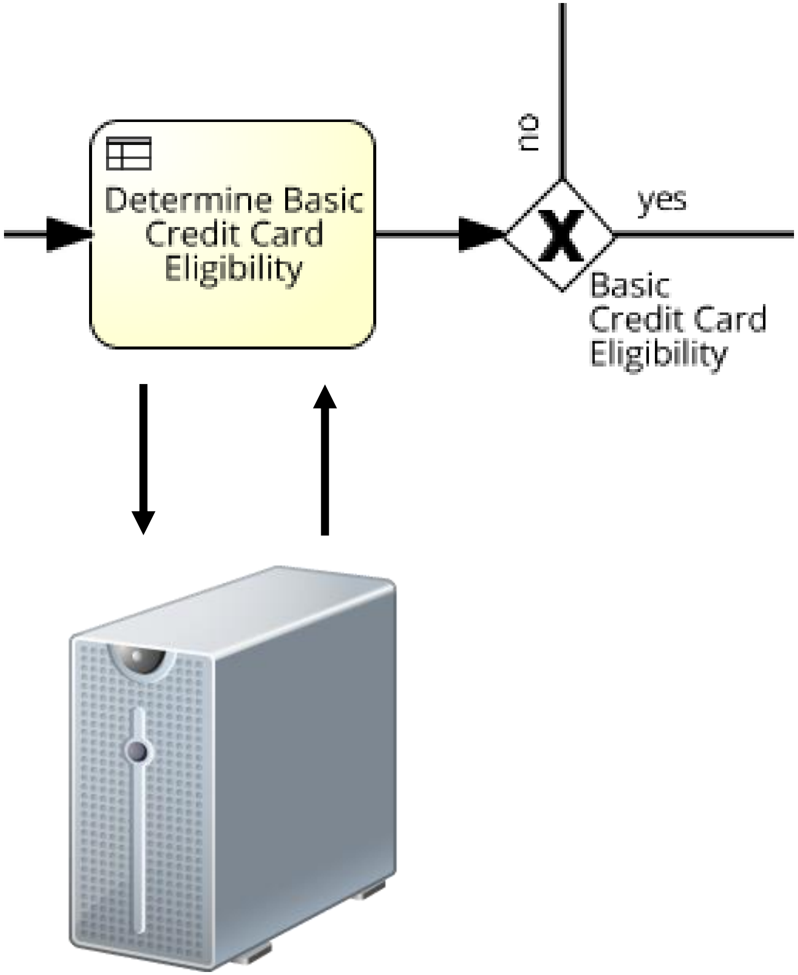
It‘s the decisions that make or break your company

Process Automation



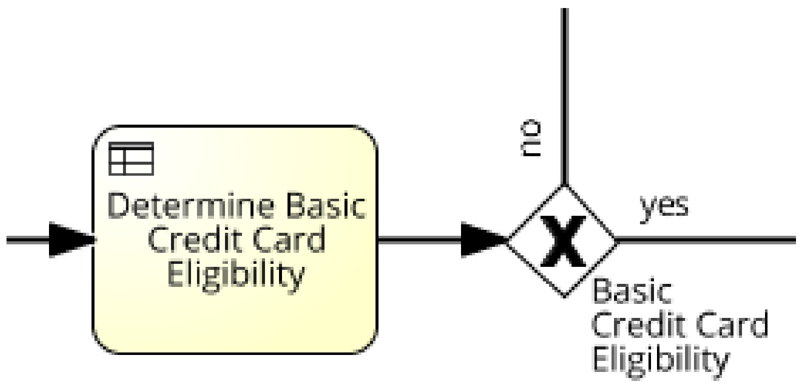
from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

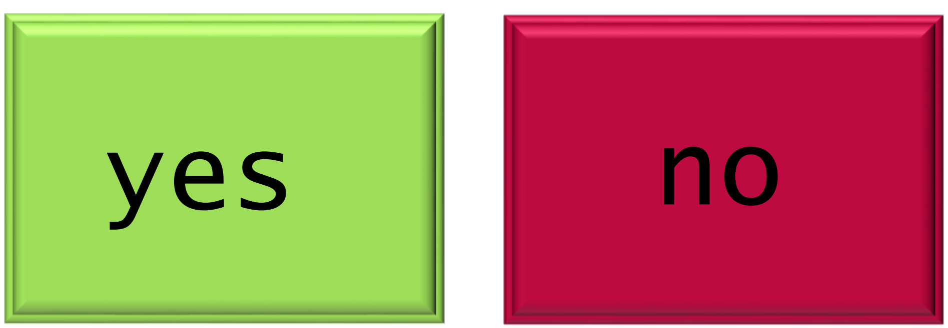
Implementation of Decisions (1/2)



- 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)

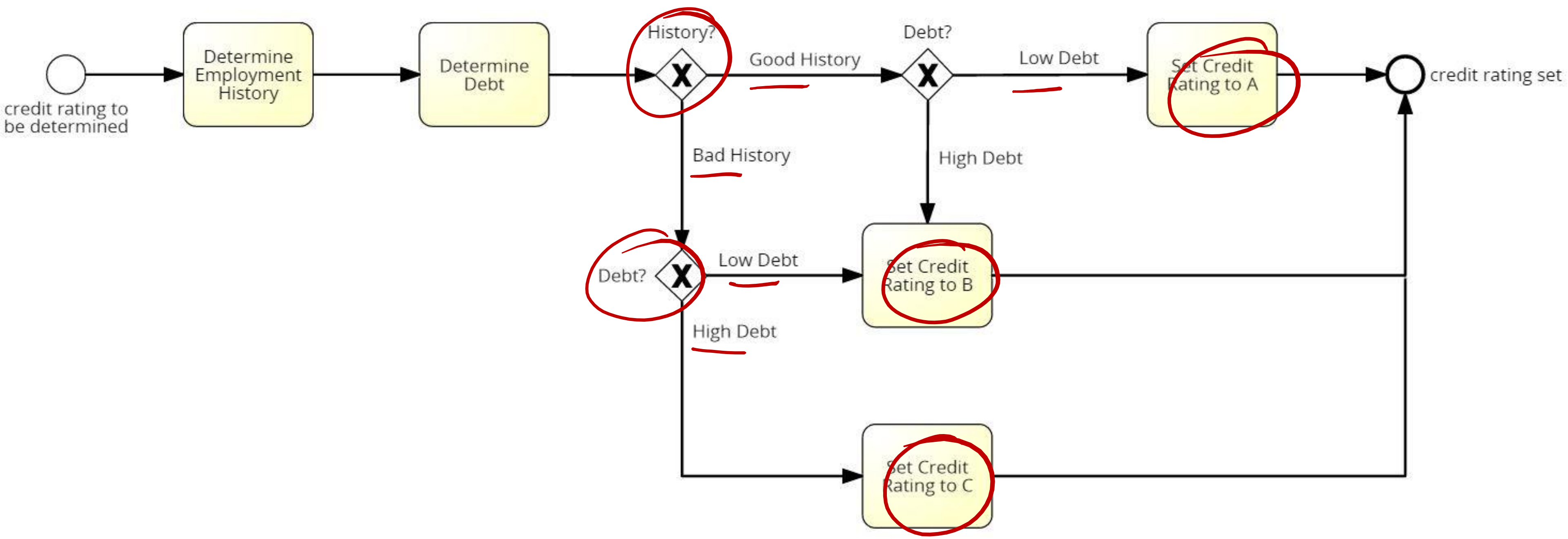




- Decision is taken by process participant
- Issues
  - How is the decision taken?
  - Which data is being used?
  - Which policies are taken into account?
  - Who is responsible for the decision?

Decisions encoded in Processes

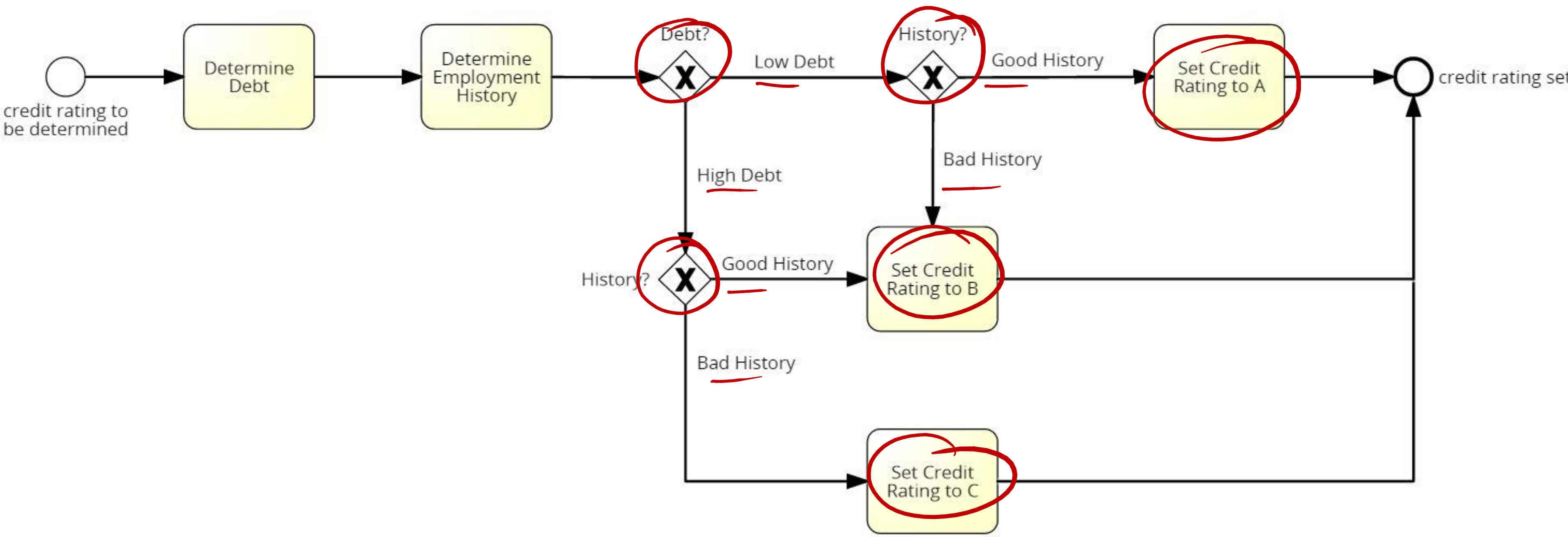
- Empirical analysis of 500+ process models from the financial industry
- Example, Version 1



Process models are being mis-used to model decisions

Decisions and Processes

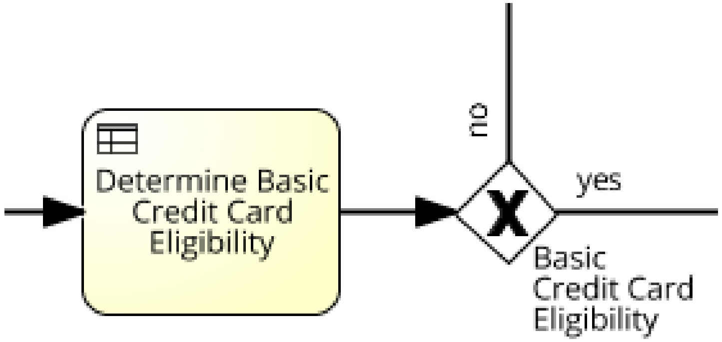
- Example, Version 2



- Process constrains execution sequence of activities, while not providing actual decision logic
  - Sequence flow does not reflect causality

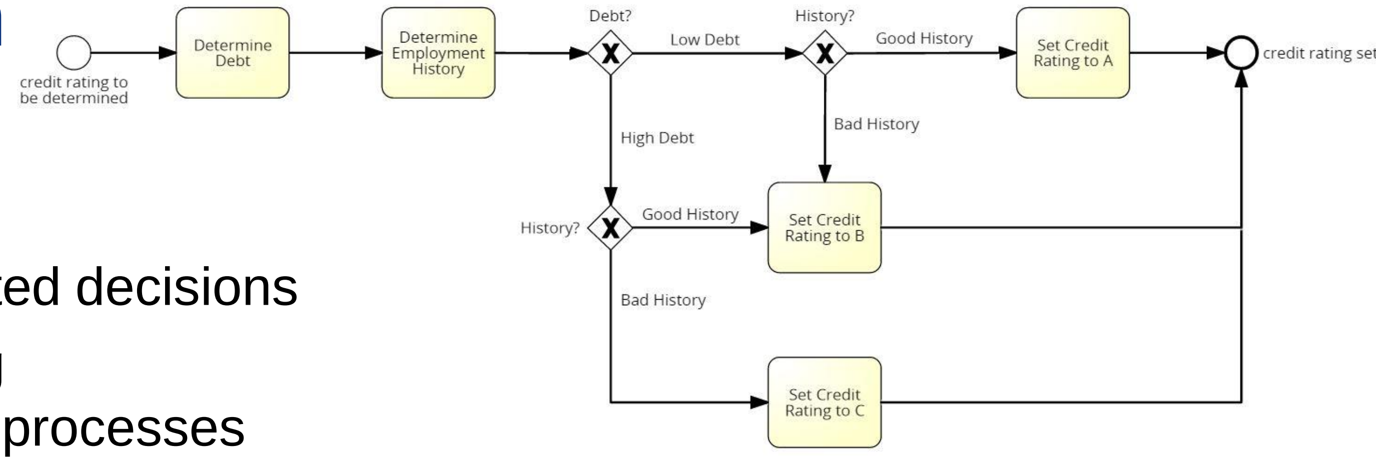
Decision Management

- 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

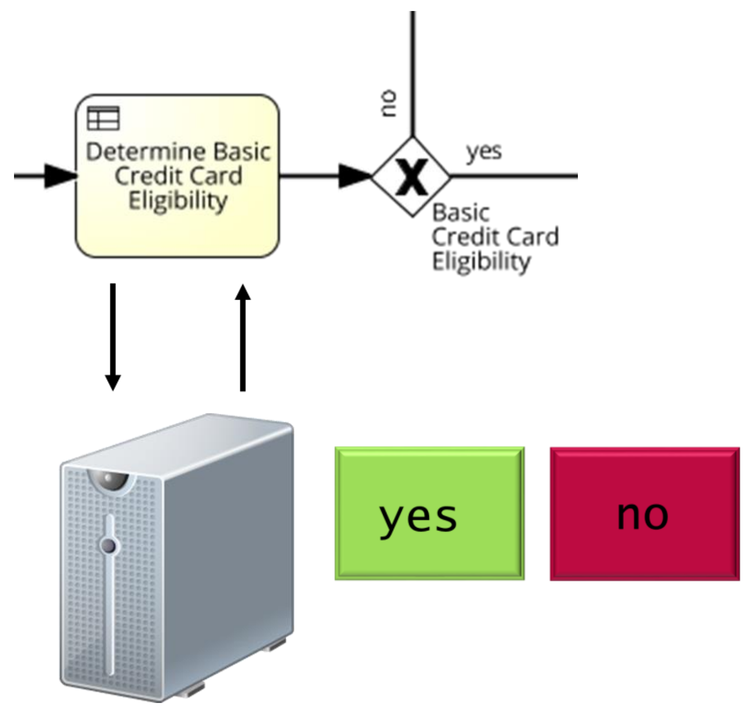


- Explicit representation of
  - decision structure and decision logic to provide a common language for domain experts and engineers

Video Clip 6.1 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



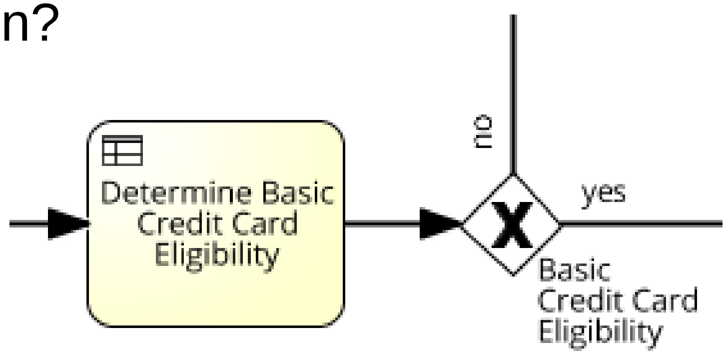
Video Clip 6.2

DMN Overview

Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Decision Management

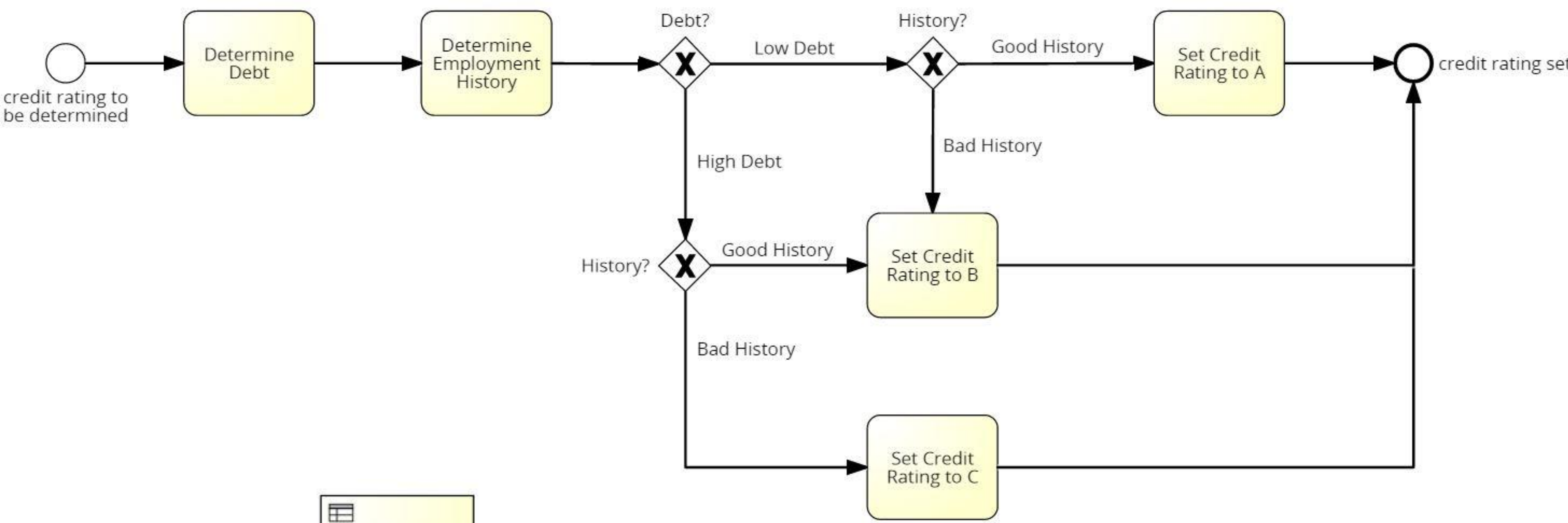
- Goal of decision management
  - Model well-defined, operational decisions, thereby improving transparency, maintainability, and supporting change and automation of decisions
- Decision management answers questions
  - How is the decision taken?
  - Which data is being used?
  - Which policies are taken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

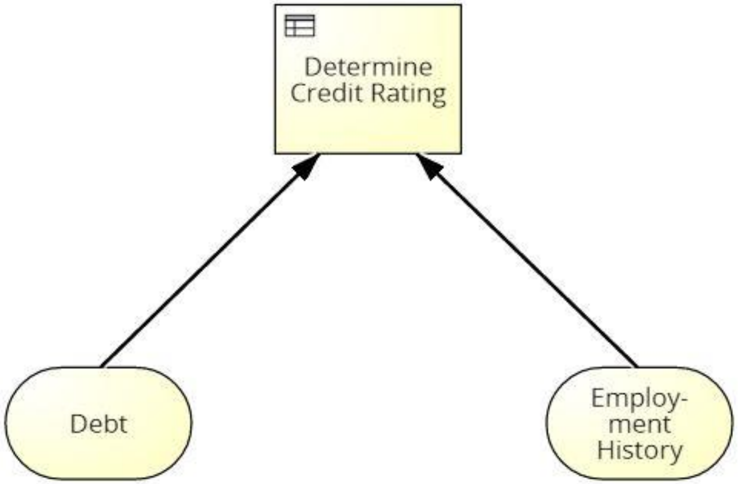


Decision Model and Notation

- DMN 1.0
  - OMG standard, Sept 2015
  - DMN considers two levels of abstraction
  - No major changes in the upcoming Version 1.1
- 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 or business rules
- Relationship to BPMN
  - DMN complements BPMN nicely
  - But DMN can also be used without BPMN!

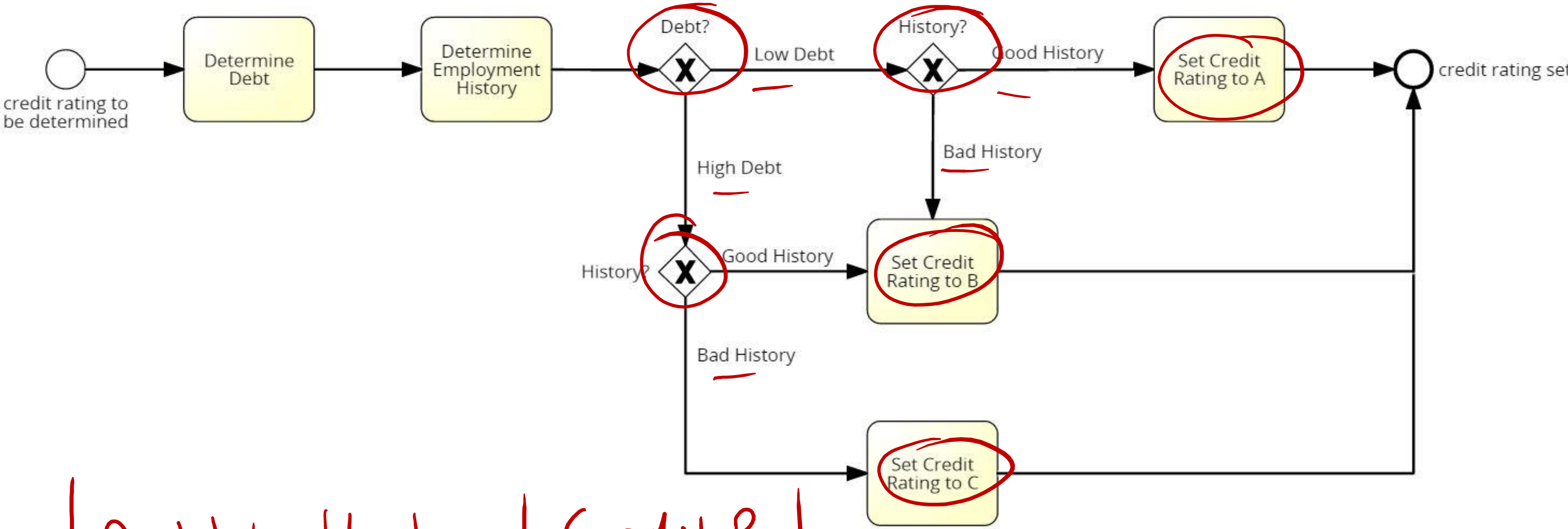
Example 1: Decision Requirements





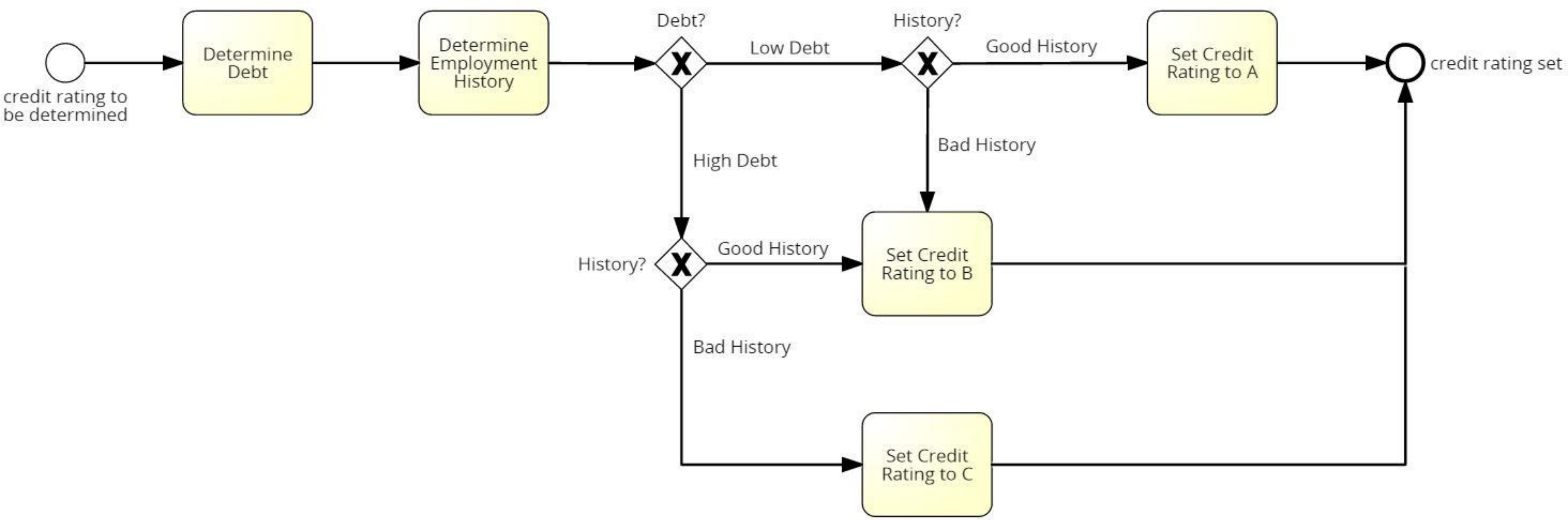
- Decision Requirement Diagrams
  - Model mainly the structure of decisions and their inputs

Example 1: Decision Logic



Debt	History	Credit R.
L	G	A
L	B	B
H	G	B
H	B	C

Example 1: Decision Logic

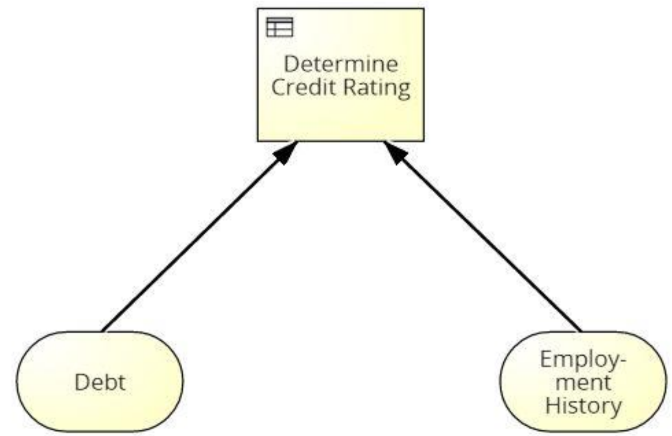


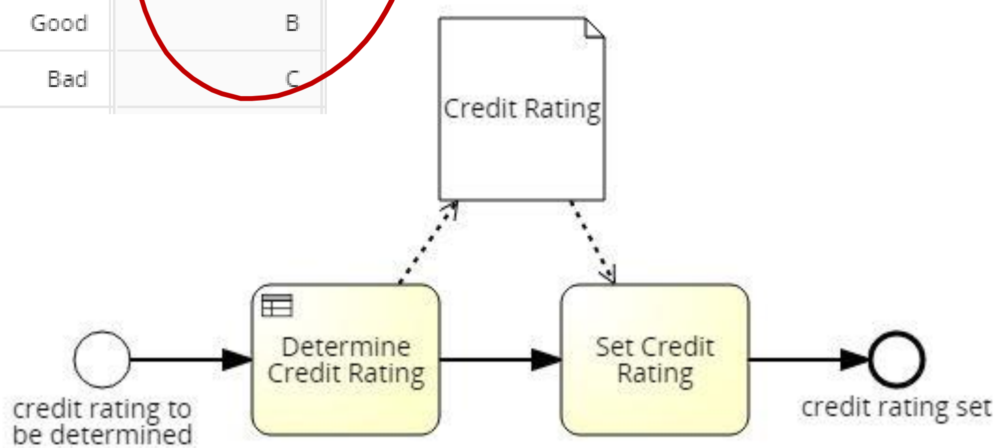
	Inputs		Outputs
UC	Debt	Employ ment History	Credit Rating
	{Low,High}	{Bad,Good}	{A,B,C}
1	= Low	= Good	A
2	= Low	= Bad	B
3	= High	= Good	B
4	= High	= Bad	C

Example 1: Decision Assigns Value

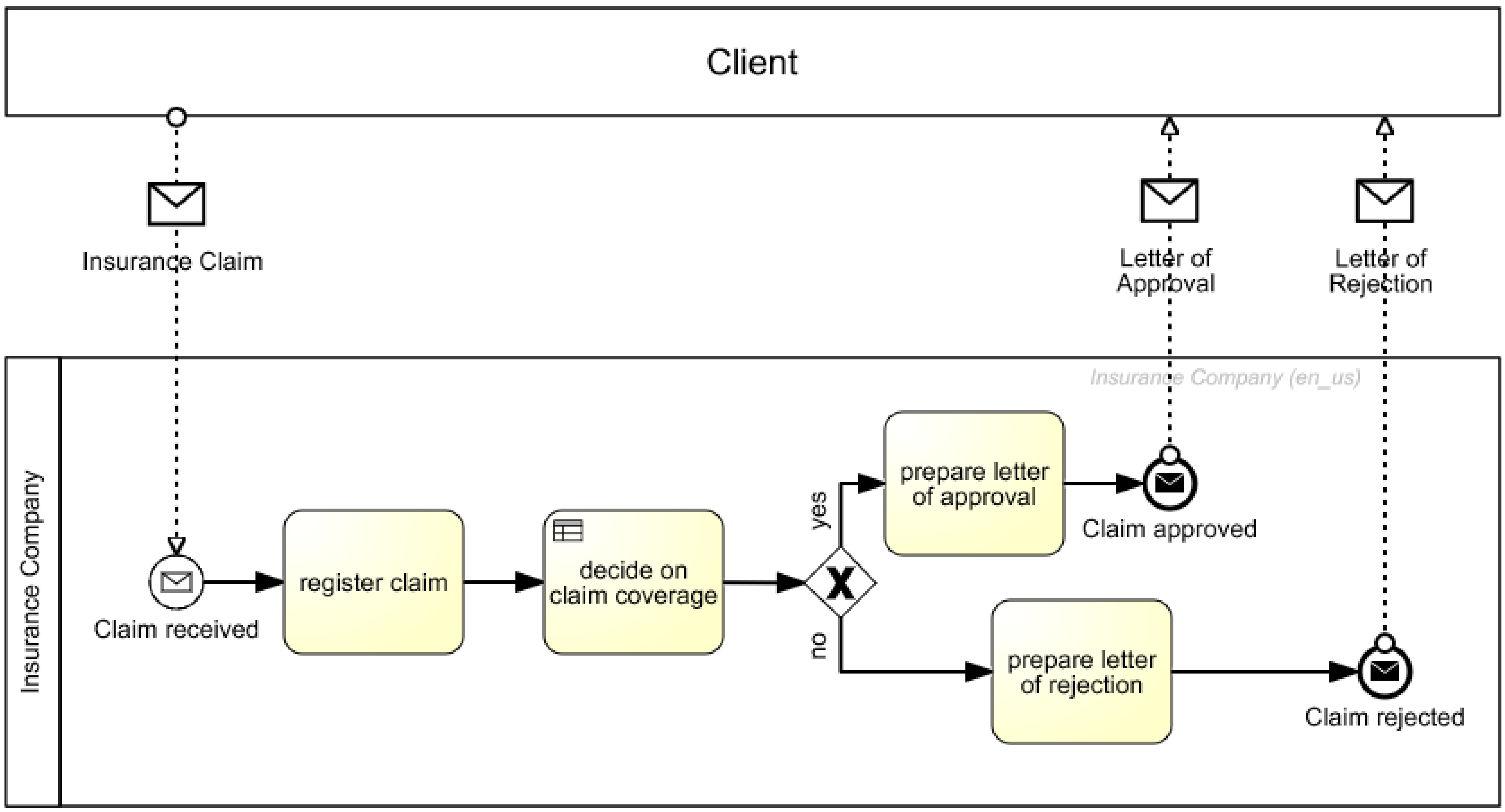
- Decision assigns a value to data object *Credit Rating*
  - This data object is used in a subsequent activity
  - The process can be simplified

	Inputs				Outputs
UC	Debt		Employ ment 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

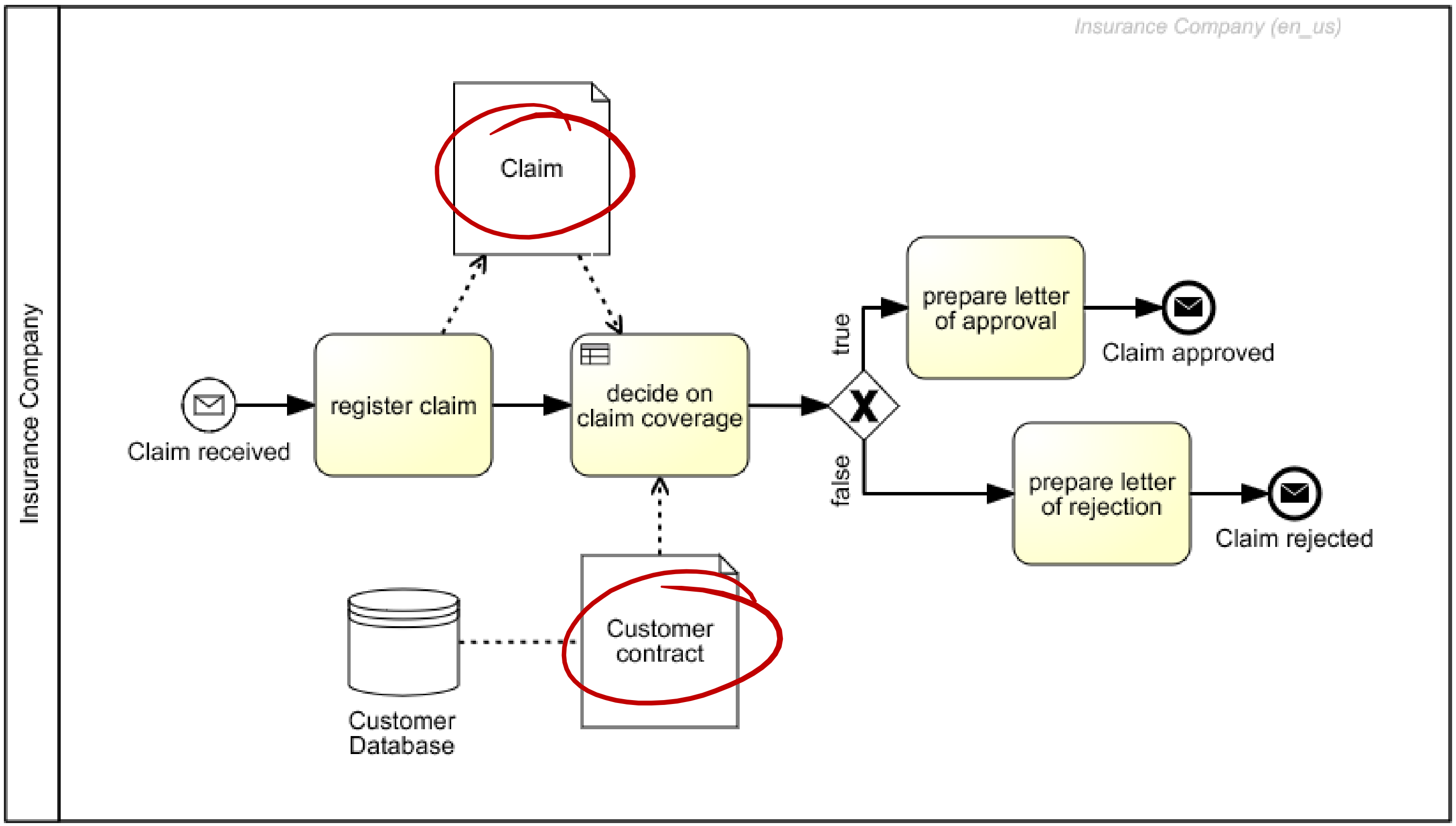




Example 2: Insurance Claim Process

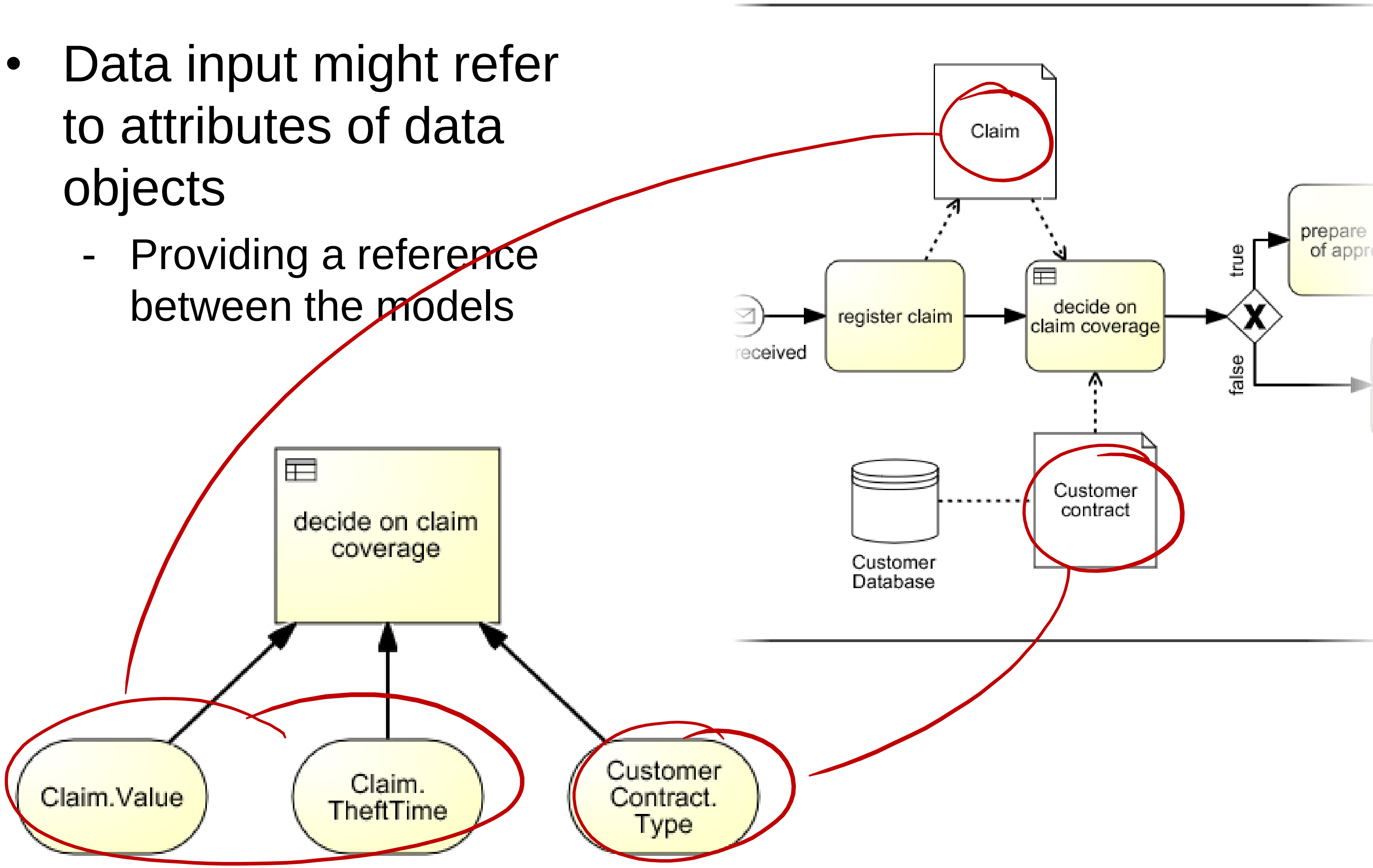


Example 2: Adding Data Objects

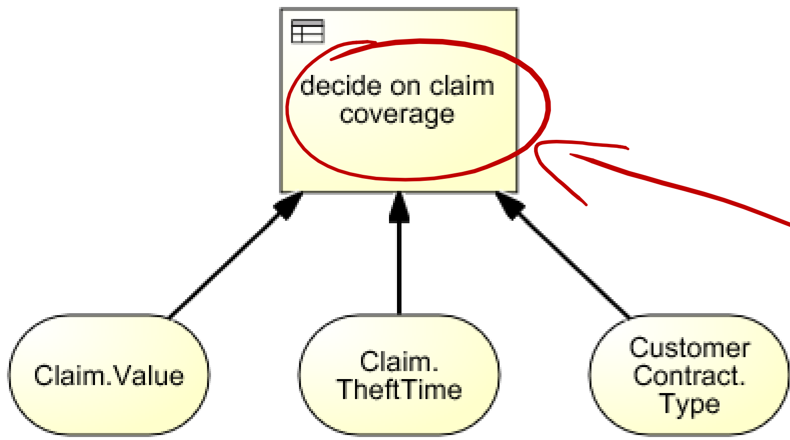


Example 2: Decision Requirements

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



Example 2: Decision Logic



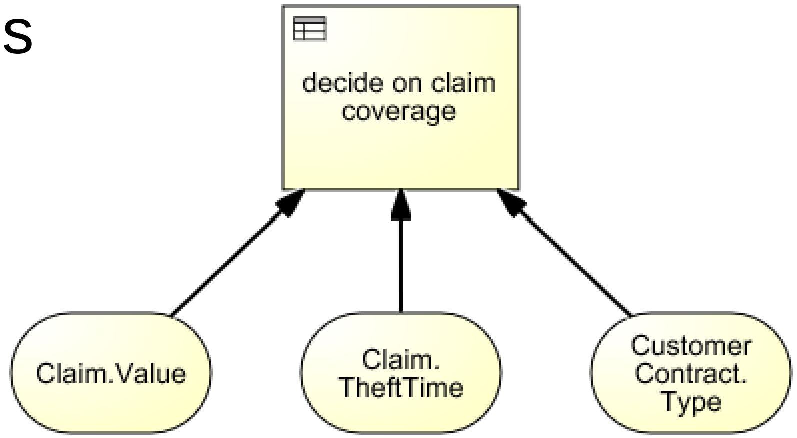
Add rule ▾ Remove rule | Add input ▾ Remove column

F	Claim.Value (Inputs)		Claim.TheftTime		CustomerContract.Type		Output (Outputs)
		[0€..1000€]		[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 6.2
DMN Overview

- Decision Requirements Diagrams model the structure of decisions

- Decision logic expresses how decisions are taken



Add rule ▾ | Remove rule | Add input ▾ | Remove column

F	Inputs						Outputs
	Claim.Value		Claim.TheftTime		CustomerContract.Type		Output
	[0€..1000€]		[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

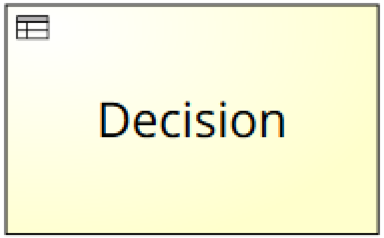
Video Clip 6.3

Decision Requirements Diagrams

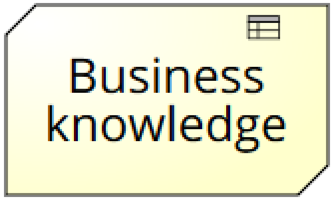
Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

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*

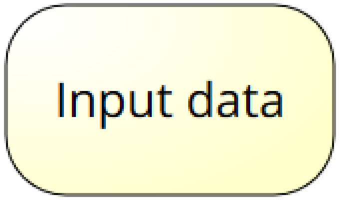


- Business knowledge
  - Contains knowledge on how a decision should be taken
  - Can be re-used by different decisions
  - Is invoked by decisions
    - Input data is transferred to the business knowledge
    - Business knowledge provides output to the decision

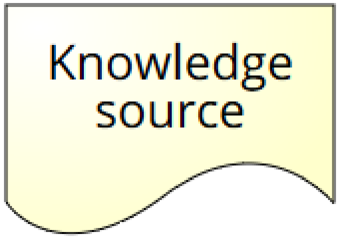


Further Model Elements

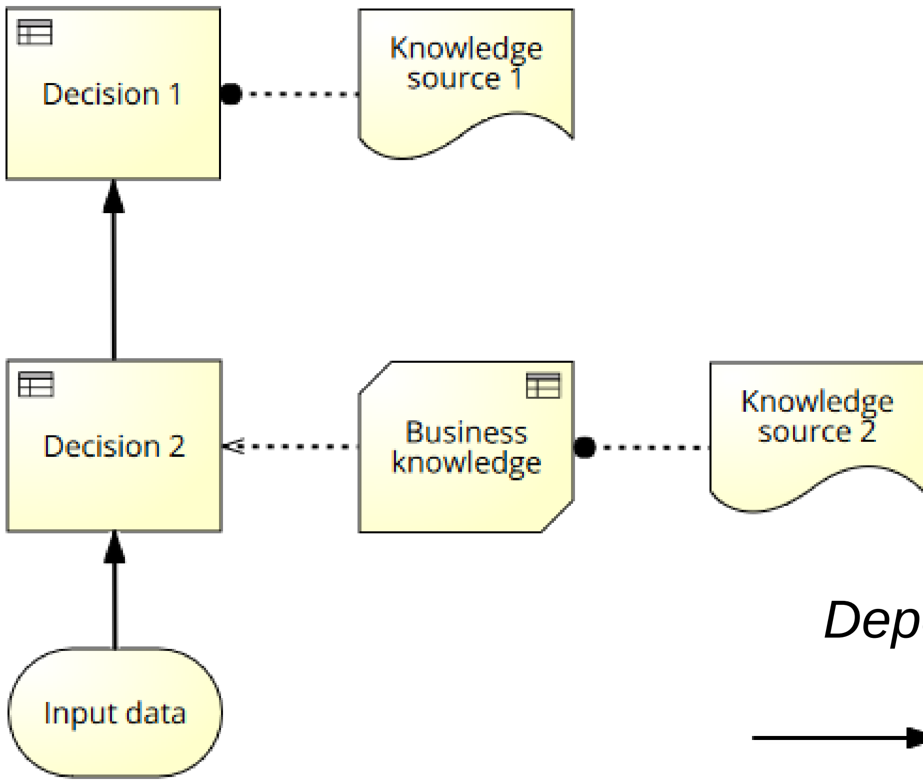
- Input data
  - Input data for decisions
  - To be used during decision making
  - Can be linked to a process model by label



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



Decision Requirements Graph



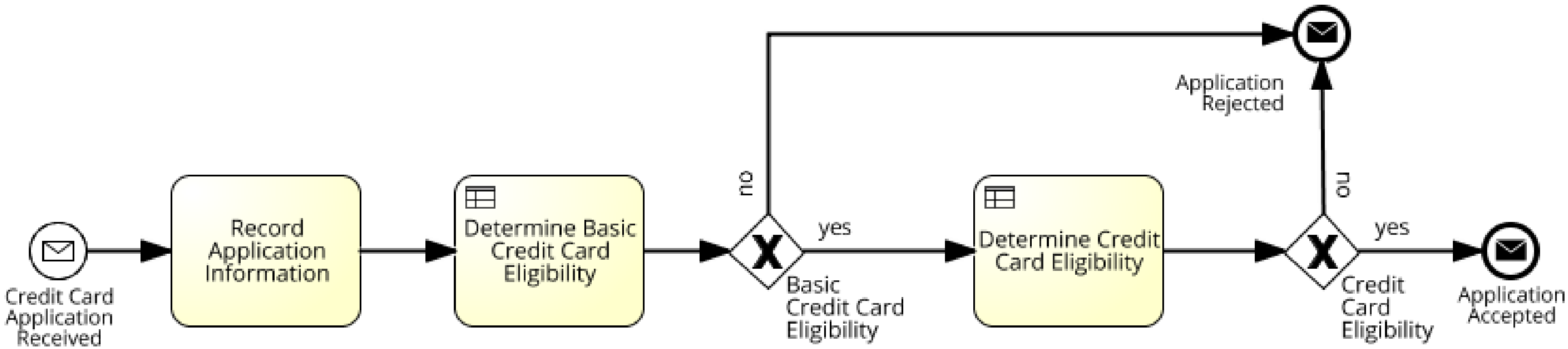
Dependencies

Information Requirement

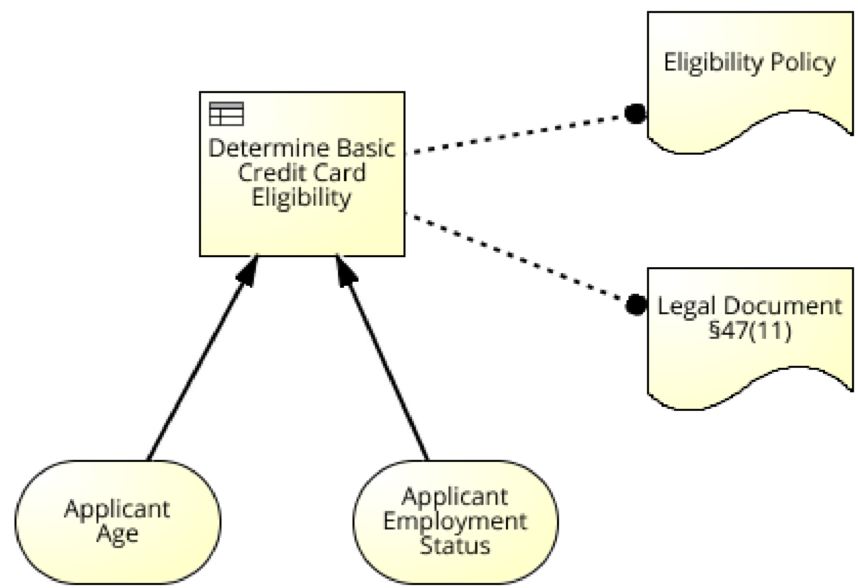
Knowledge Requirement

Authority Requirement

Credit Card Example (1/2)

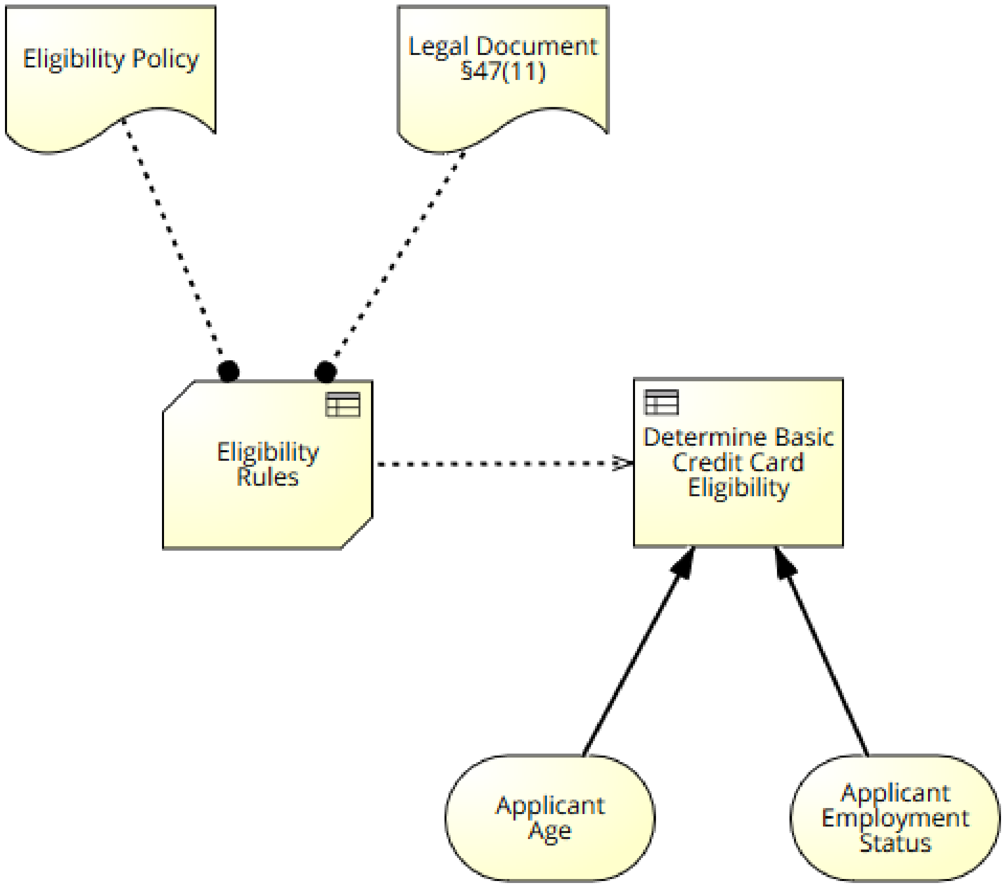


- The decision
  - uses applicant age and employment status as input
  - is based on eligibility policy of the institution and a legal document



Credit Card Example (2/2)

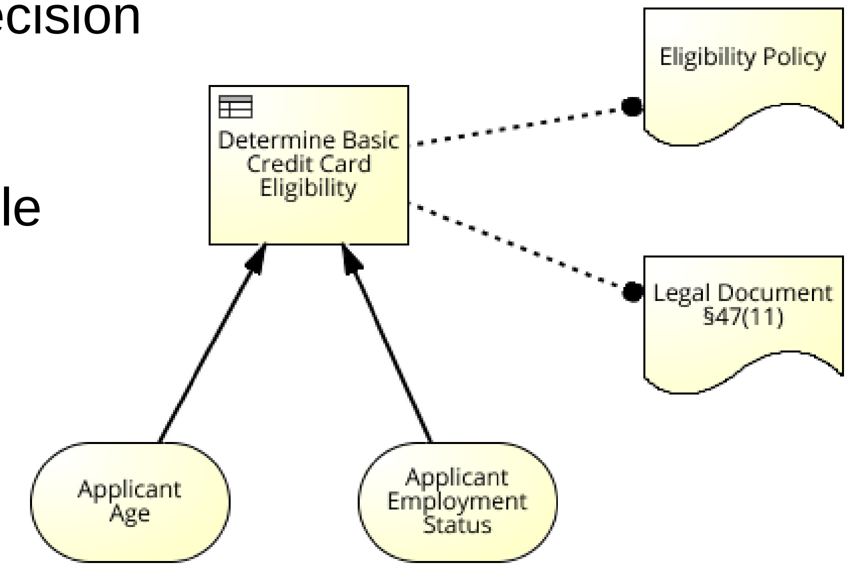
- Example
  - Decision invokes the business knowledge *Eligibility Rules*
- Notice
  - Knowledge sources are linked to business knowledges rather than decisions
  - Because knowledge source contains decision logic



Organizational Aspects

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

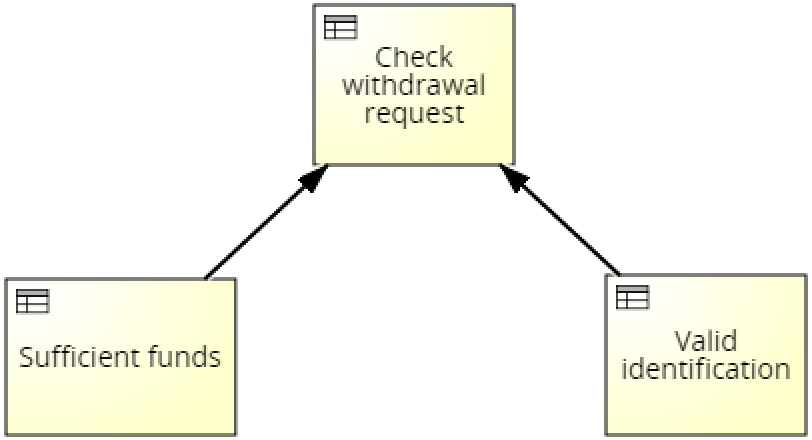
Scenario: Withdrawal of funds at a Bank

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.

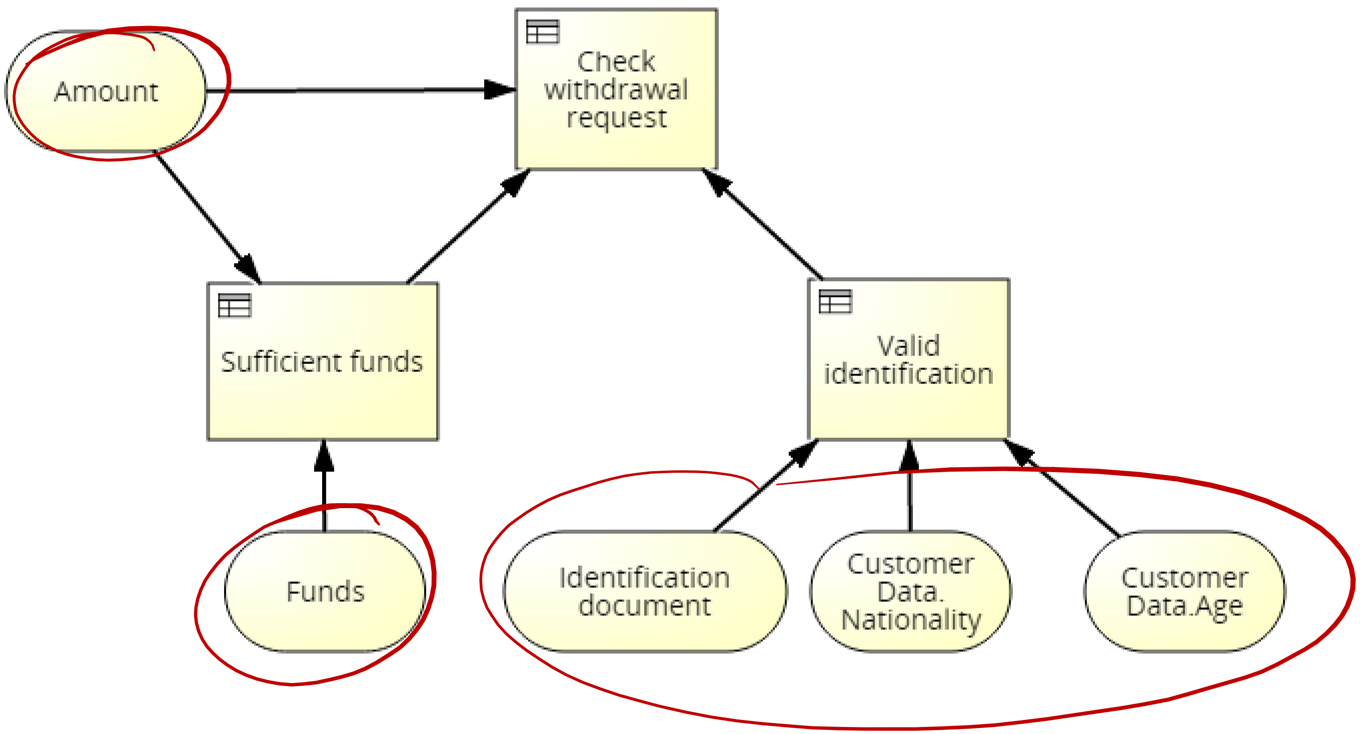
Scenario: Structure of Decisions

1. Define decisions and sub-decisions



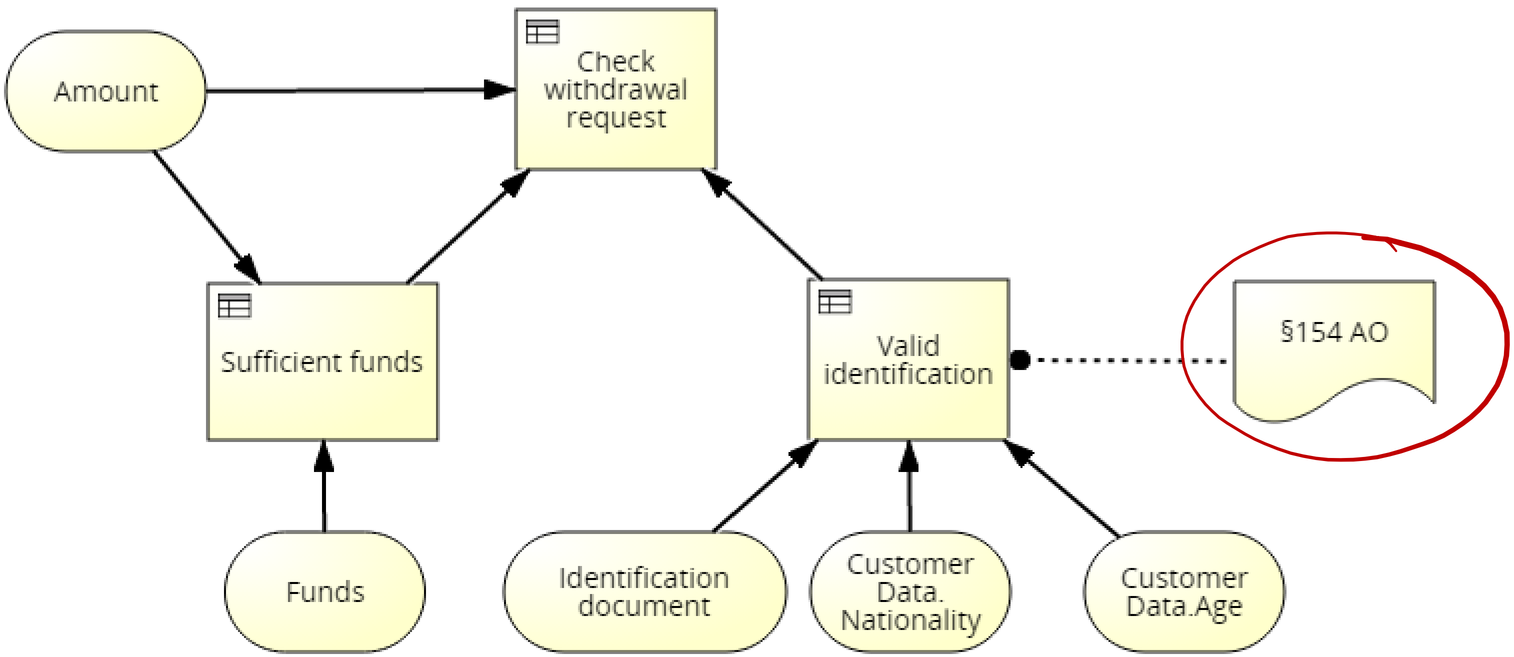
Scenario: Input Data

2. Define input data



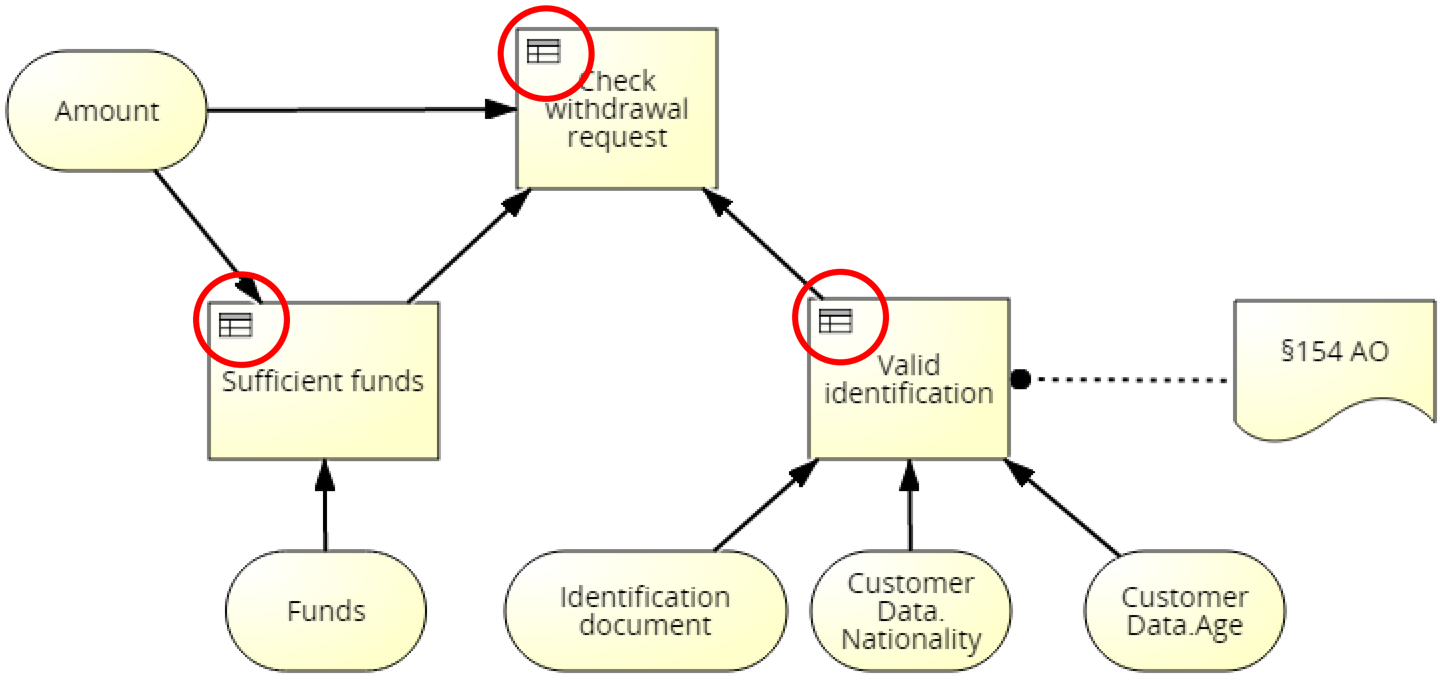
Scenario: Knowledge Sources

3. Define knowledge source



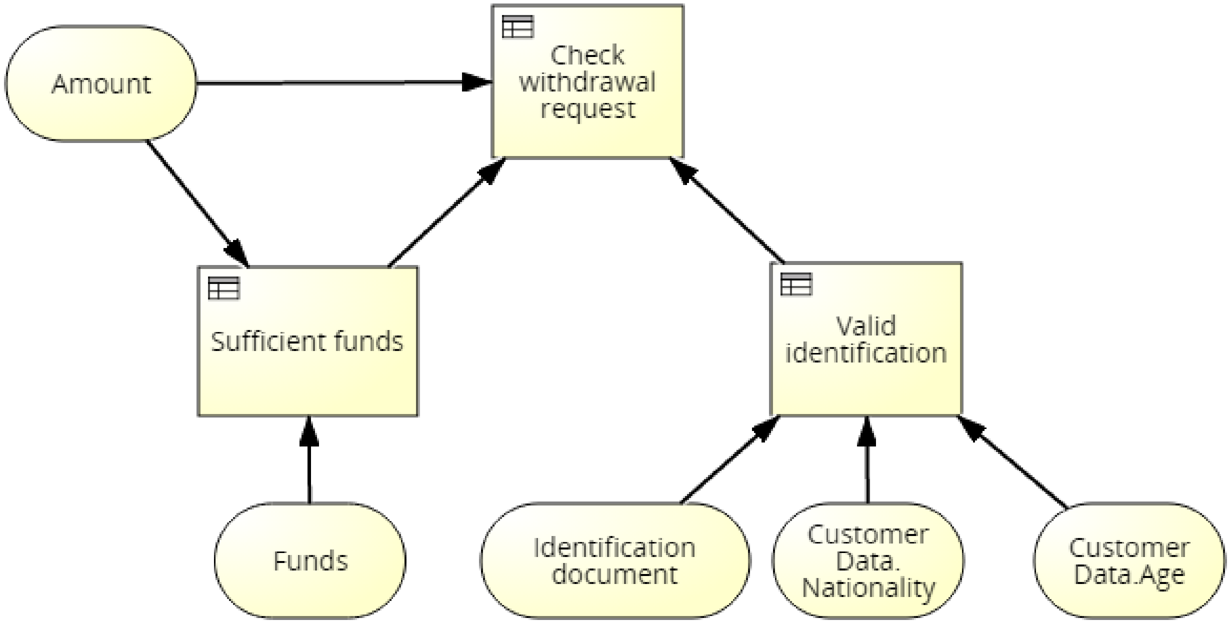
Scenario: Decision Logic

4. Define decision logic
   … will be done in the next video clip



Video Clip 6.3
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 6.4

Decision Logic

Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

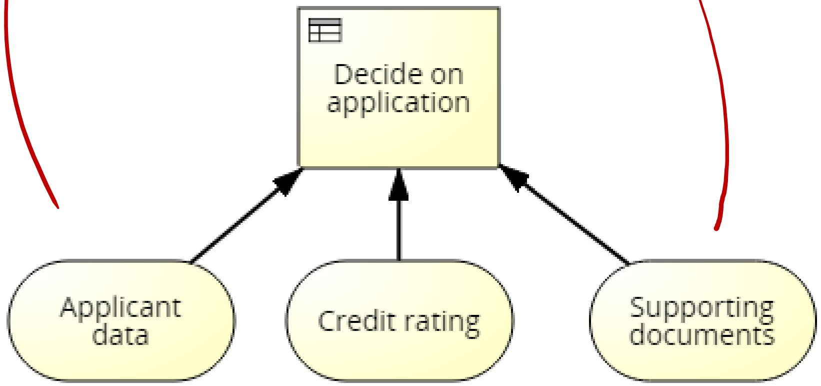
Introduction to Decision Logic

- 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."*

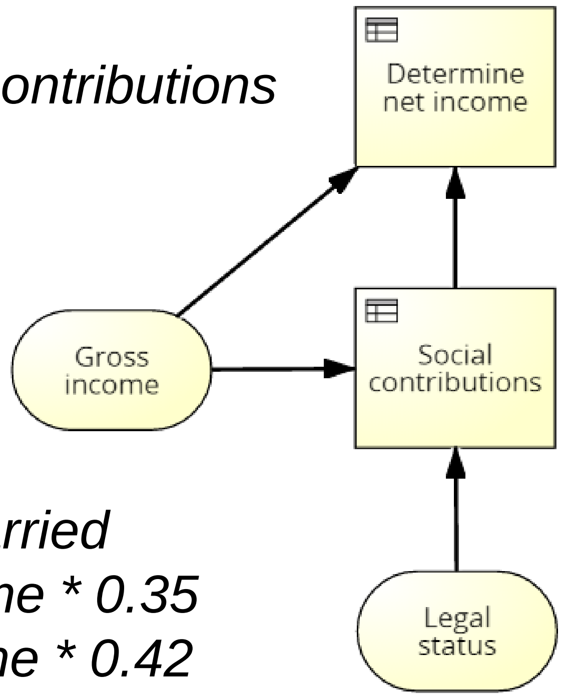


Literal expression

- 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 exactly one column)
  - Set of rules, the rows of the table
  - Hit policy of the table (explained below)
- Representations
  - FEEL
  - Decision table (“boxed expression” in DMN)

Decision Table Terminology

Hit policy

Completeness indicator

Set of inputs

Output column
(multiple output columns allowed)

Ordered list of output values

	Inputs				Outputs
U	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no

Rule number

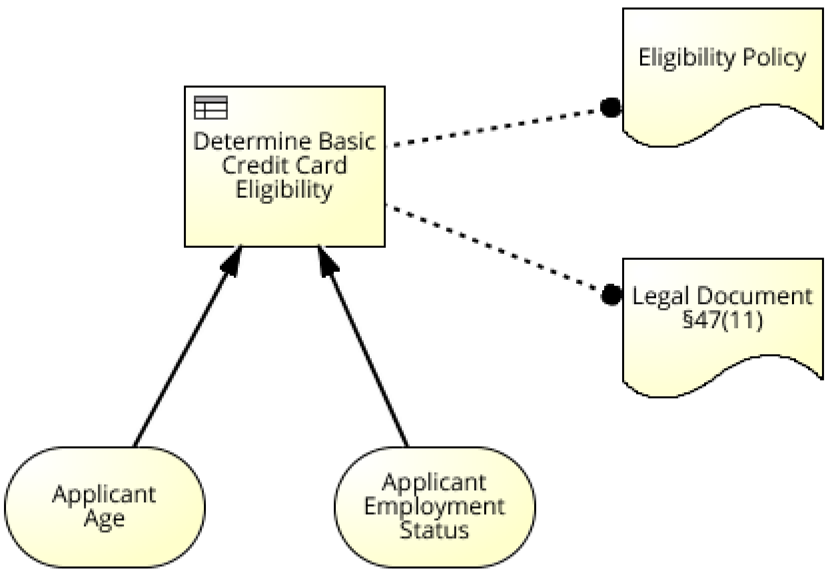
Input entries

Domains of input values

Output entries

Decision Tables

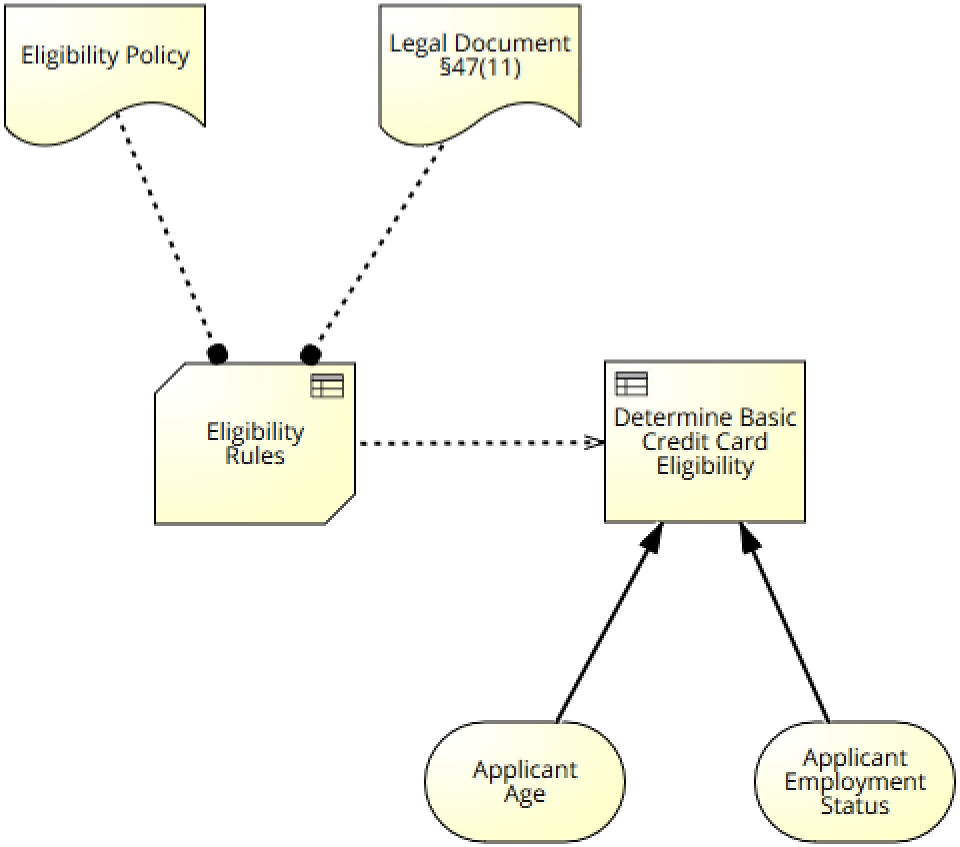
- Structure
  - One rule per row
  - Conjunctive expressions on input data
  - Disjunctions by multiple rows
  - Output column shaded



U	Inputs				Outputs
	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		*{employed,unemployed}*		*{yes,no}*
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no

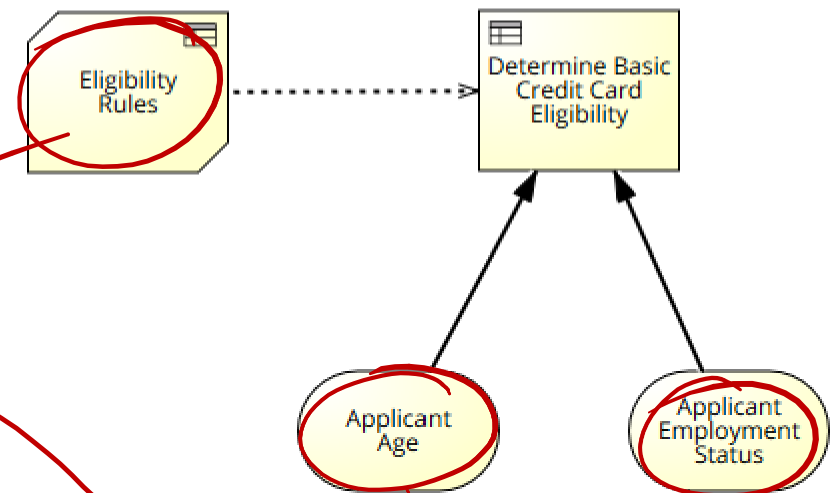
Invocation

- A decision can invoke a business knowledge
  - Input values of decision table are transferred as parameters
  - Return values are received (the decision output)



Invocation

- Example
  - Decision invokes business knowledge *Eligibility Rules*



Eligibility Rules

U	Inputs				Outputs
	AppAge		**EmpStatus**		**Output**
	[1..100]		*{employed,unemployed}*		*{yes,no}*
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no

Determine Basic Credit Card Eligibility

Eligibility Rules	
AppAge	**Applicant.Age**
EmpStatus	**Applicant.EmploymentStatus**

Video Clip 6.4
Decision Logic

- Define decision logic
  - by text,
  - literal expression,
  - decision table,
  - Invocation.

„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*

	Inputs				Outputs
U	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	<	65		-	no

Eligibility Rules

	Inputs				Outputs
U	AppAge		EmpStatus		Output
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no

Determine Basic Credit Card Eligibility

Eligibility Rules	
AppAge	Applicant.Age
EmpStatus	Applicant.EmploymentStatus

Video Clip 6.5

Semantics of Decision Tables

Business Process and Decision Modeling:
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

	Inputs				Outputs
U	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	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

	Inputs				Outputs
U	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
- Example: Credit policy compliance decision
  - For example, (not A, 10.000, 60.000) matches rules 2, 4
  - These rules have the same output

	Inputs						*Outputs*
A	**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
- Example: Health risk decision
  - Input combination (70, bad)

	Inputs				Outputs
P	Age		Medical history		Health risk level
	Number		*{good,bad}*		*{low,medium,high}*
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
  - These have different outputs
  - Priority policy makes sure that only output *medium* is returned, because *medium* appears in the output list before *high*

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

The priority policy should be avoided, because of its complex 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: Discount rate decision
  - Input combination (VIC, wholesale, web)

	Inputs						Outputs
F	Customer location		Customer type		Channel		Discount rate
	{VIC,NSW}		{wholesale,retail}		{web,phone}		Percentage
1	=	VIC	=	wholesale	=	web	15.00 %
2	=	NSW	=	wholesale	=	web	10.00 %
3		-	=	wholesale	=	web	5.00 %
4		-	=	retail	=	web	5.00 %
5		-		-	=	phone	0.00 %

Single Hit Policies

- Example: Discount rate decision
  - (VIC, wholesale, web) matches rules 1 and 3, which have different outputs
  - F policy makes sure that only rule 1 fires

	Inputs			Outputs
F	Customer location	Customer type	Channel	Discount rate
	{VIC,NSW}	{wholesale,retail}	{web,phone}	Percentage
1	= VIC	= wholesale	= web	15.00 %
2	= NSW	= wholesale	= web	10.00 %
3	-	= wholesale	= web	5.00 %
4	-	= retail	= web	5.00 %
5	-	-	= phone	0.00 %

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

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, 4, 5 fire
  - Output priority leads to (20,10,5)

	Inputs				Outputs
O	Age		Years of service		Eligible annual leave (days)
	Number		Number		{20,10,5,3}
1		-		-	20
2	<	20		-	3
3	>	60		-	5
4		-	≥	10	5
5		-	≥	20	10

Multi Hit Policies

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

R	Inputs				Outputs
	Age		Years of service		Eligible annual leave (days)
	Number		Number		{20,10,5,3}
1		-		-	20
2	<	20		-	3
3	>	60		-	5
4		-	≥	10	5
5		-	≥	20	10

- Input (50, 31)
- Rules 1, 4, 5 fire
  - Rule order leads to (20,5,10)

Multi Hit Policies

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

C+	Inputs				Outputs
	Age		Years of service		Eligible annual leave (days)
	Number		*Number*		*{20,10,5,3}*
1		-		-	20
2	<	20		-	3
3	>	60		-	5
4		-	≥	10	5
5		-	≥	20	10

- Input (50, 31)
- Rules 1, 4, 5 fire
  - Summation
    leads to result 35

Video Clip 6.5
Semantics of Decision Tables

- Single hit policies
  - Unique
  - Any
  - Priority
  - First

U	Inputs				Outputs
	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no

- Multi hit policies
  - Output order
  - Rule order
  - Collect
    - + < > #

R	Inputs				Outputs
	Age		Years of service		Eligible annual leave (days)
	Number		Number		{20,10,5,3}
1		-		-	20
2	<	20		-	3
3	>	60		-	5
4		-	≥	10	5
5		-	≥	20	10

Video Clip 6.6

Analysis of Decision Tables

Business Process and Decision Modeling:
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 x 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 x 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

(20, good)

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
			high, medium, low
1	< 20	good	low
2	> 20	good	medium
3	> 65	bad	high
5	–	bad	medium

- Remarks
  - Even though Completeness is not covered in DMN 1.1, it is good practice to check it!
  - 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	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 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 deterministically

Consistency of Decision Tables

- Example 3
  - Is Any property now violated?

	Inputs				Outputs
A	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

Any

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

Video Clip 6.6
Analysis of Decision Tables

- Completeness
  of Decision Tables

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

- Consistency of Decision Tables
  - Unique
  - Any
  - Priority / First

	Inputs				Outputs
U	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

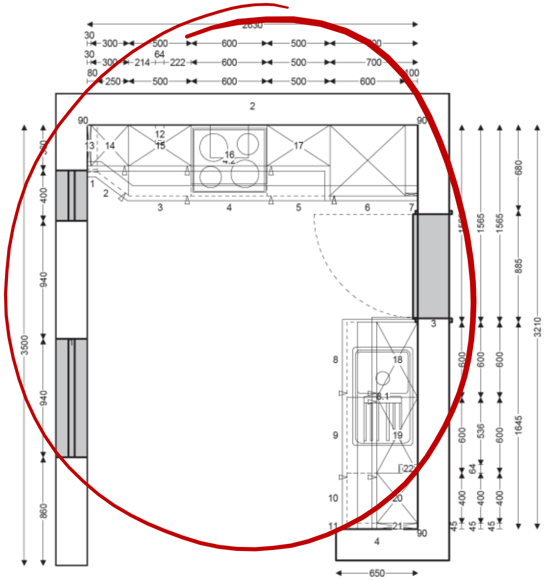
Video Clip 6.7

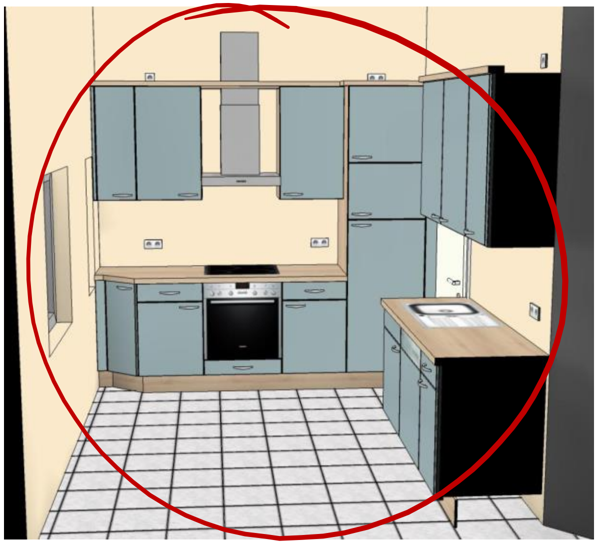
Consistency of Processes and Decisions

Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Introduction

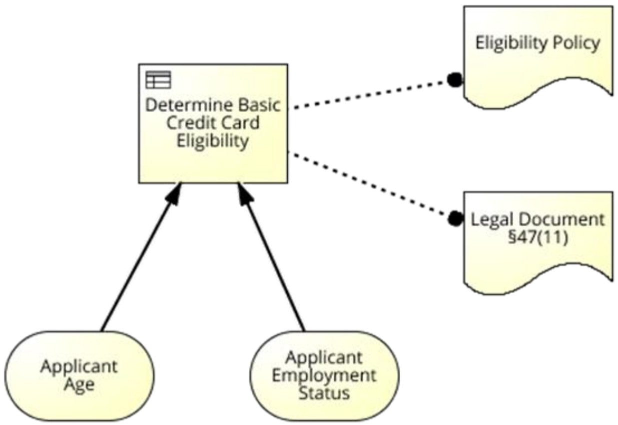
- A business process with decisions can be represented by
  - Process model (BPMN process diagram)
  - Decision requirements model (DMN DRG)
  - Decision logic model (DMN Decision Table)
- Modeling theory: One original – several models
  - These models need to be consistent
  - There must not be any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

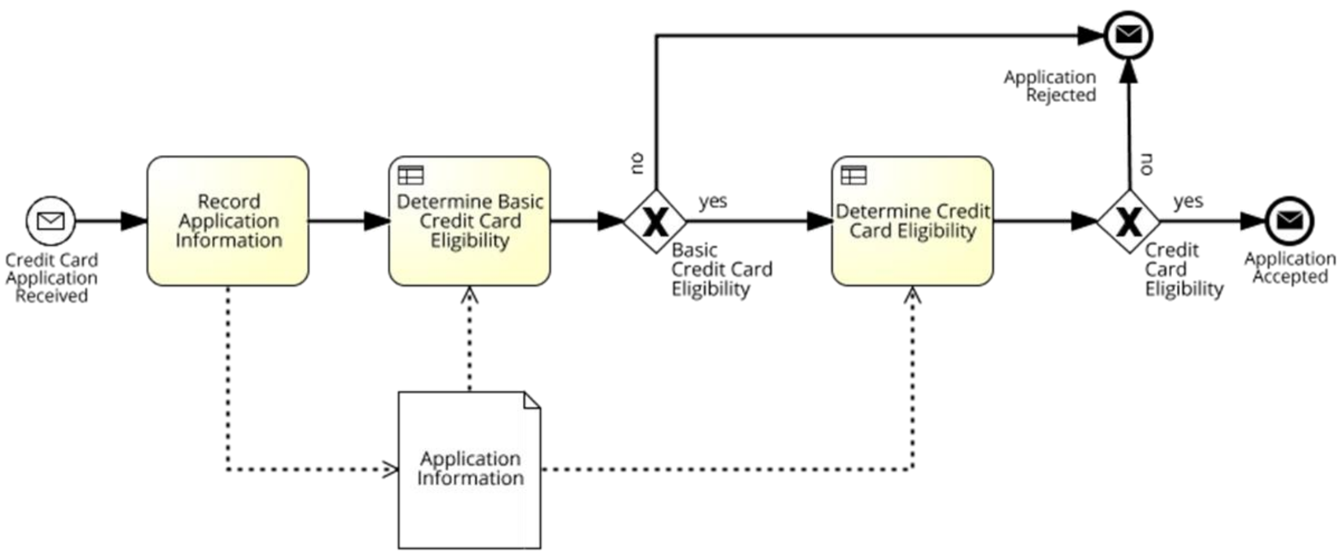




Structural Consistency

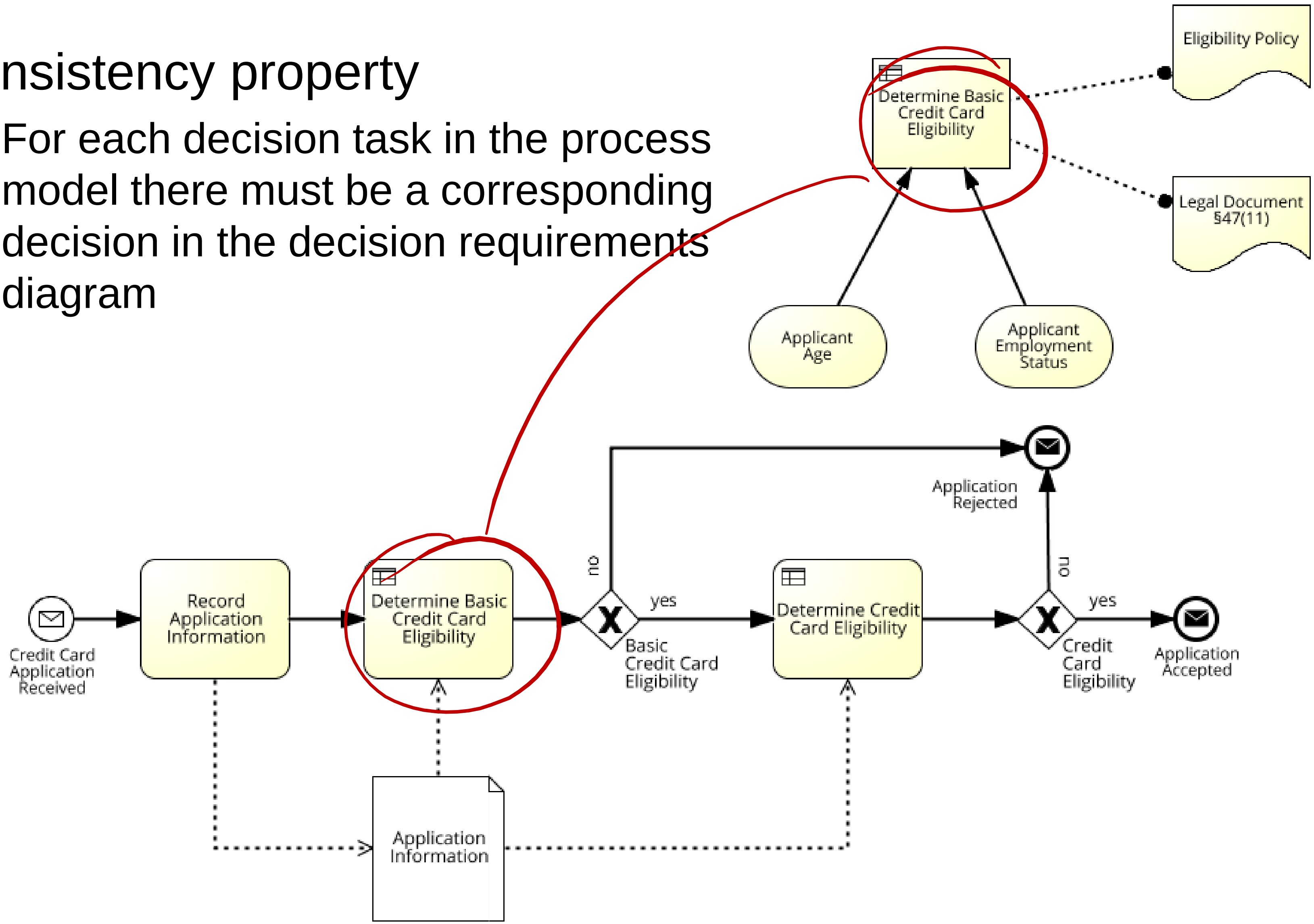
- We investigate consistency related to the structure of decisions, involving
  - Decision tasks and decisions
  - Data aspects
  - Organizational aspects





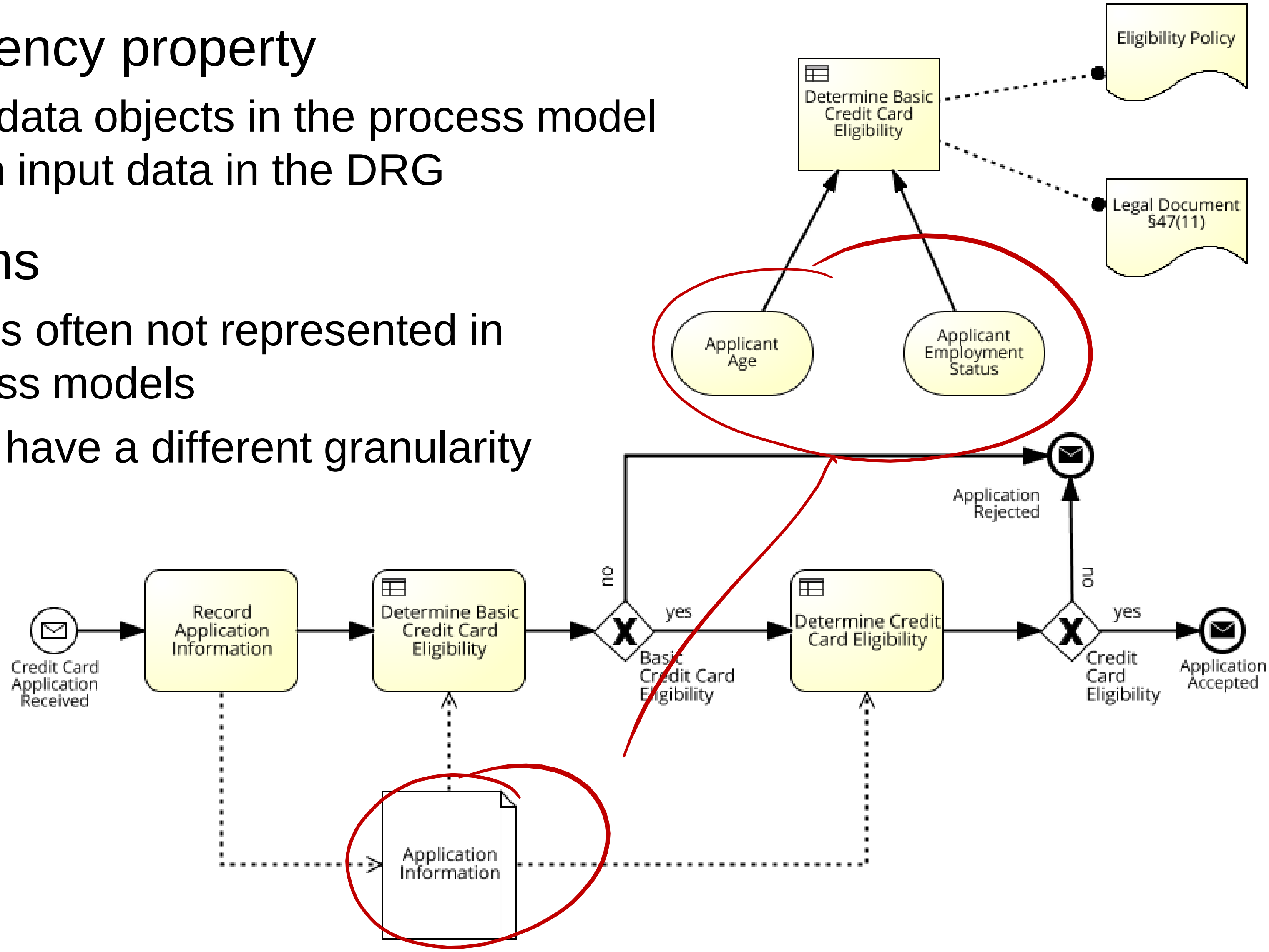
Decision Tasks and Decisions

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



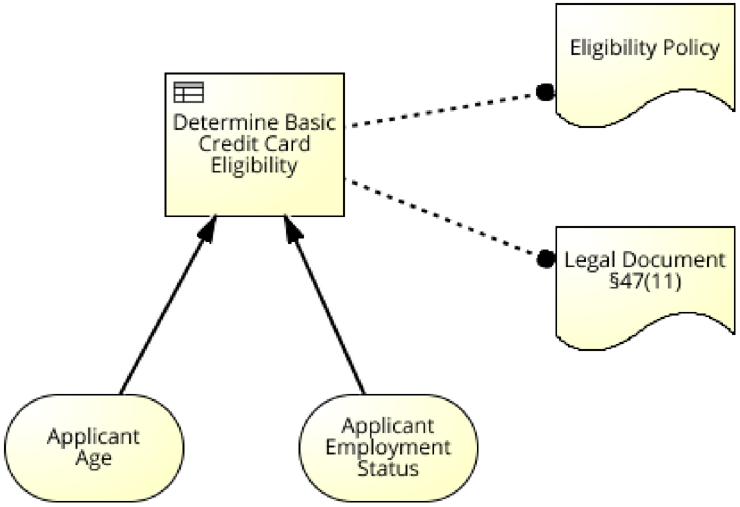
Input Data

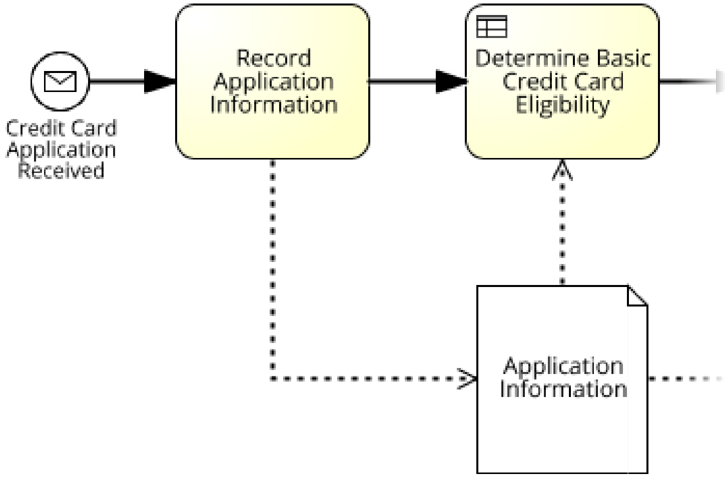
- Consistency property
  - Input data objects in the process model match input data in the DRG
- Problems
  - Data is often not represented in process models
  - Might have a different granularity



Input Data

- 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 and decision models expose the need to align data objects with decision inputs

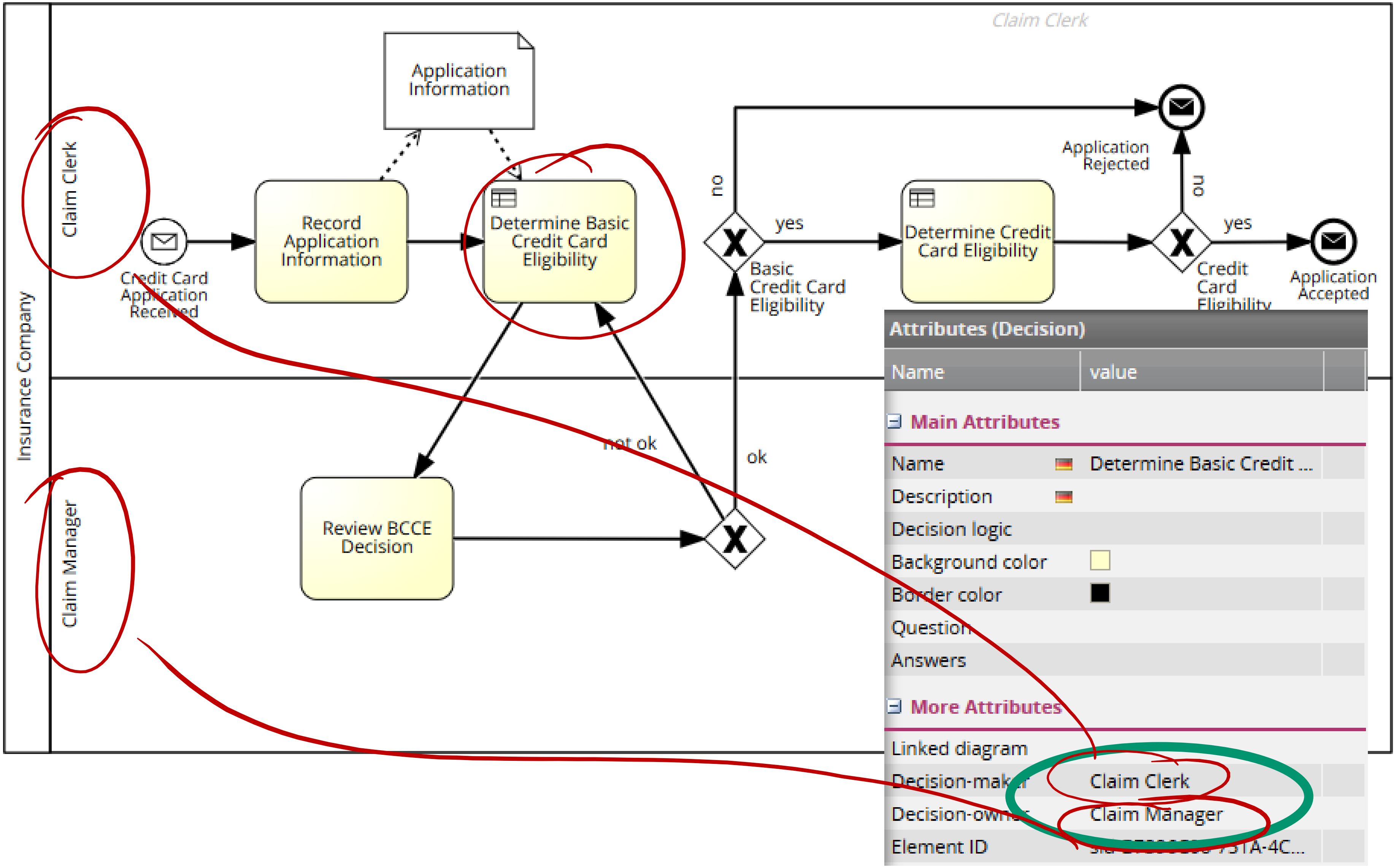




Organizational Aspects

- Defining responsibilities
  - We can specify lanes in business processes, assigning responsibilities to activities
  - We can also specify organizational aspects in decision models
- Consistency property
  - The resource identified by the lane of a decision task *D* in a process model matches the resource that is the decision maker of *D* in the decision model
  - In DMN, we can also specify a decision owner, who can be involved in a possible reviewing activity of the decision

Organizational Aspects



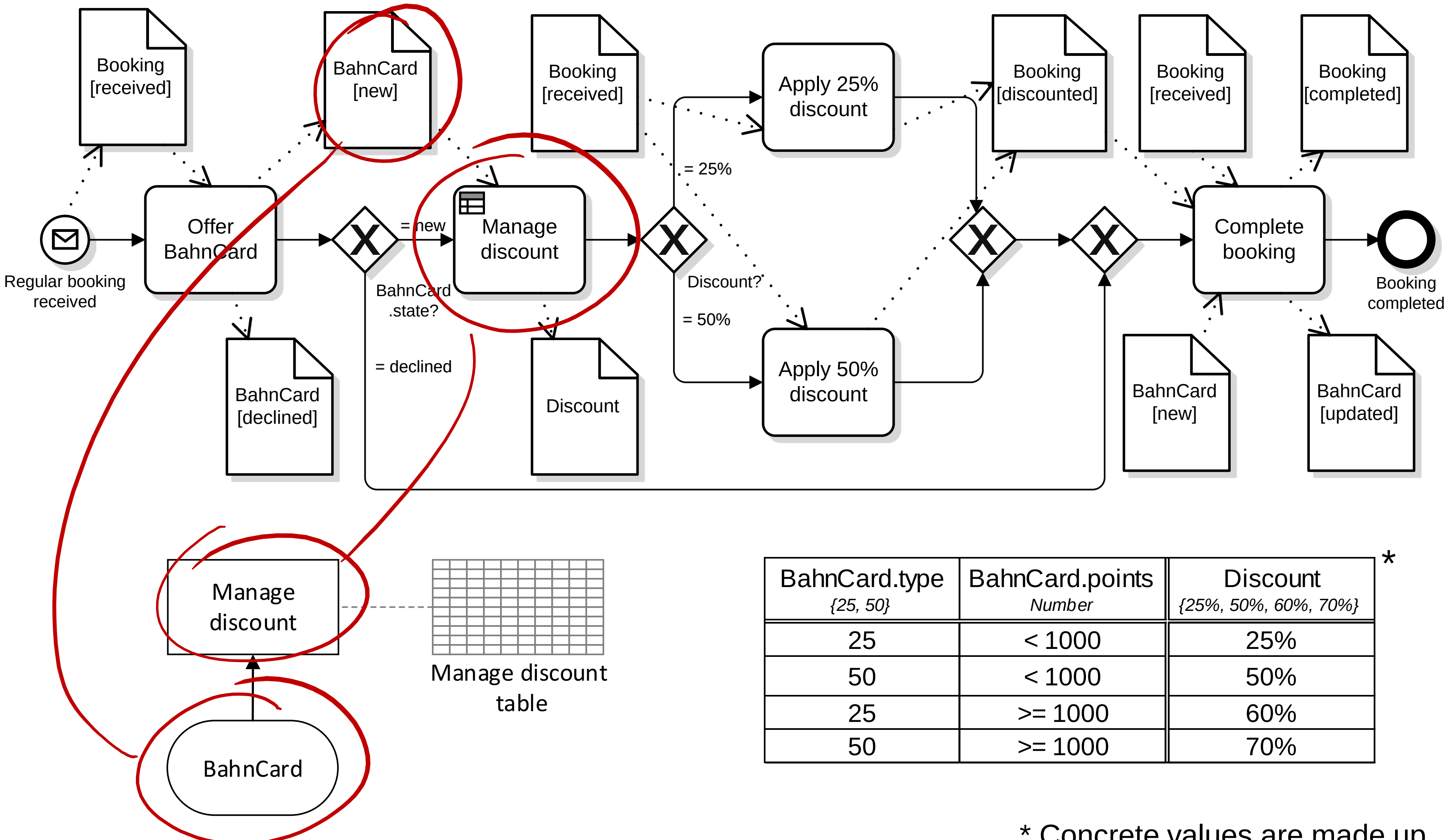
Behavioural Consistency

- Investigate if decision output is consistent with the process behaviour

- *Output coverage*
  - All outputs of the decision can be processed by the process
- *Branch coverage*
  - 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

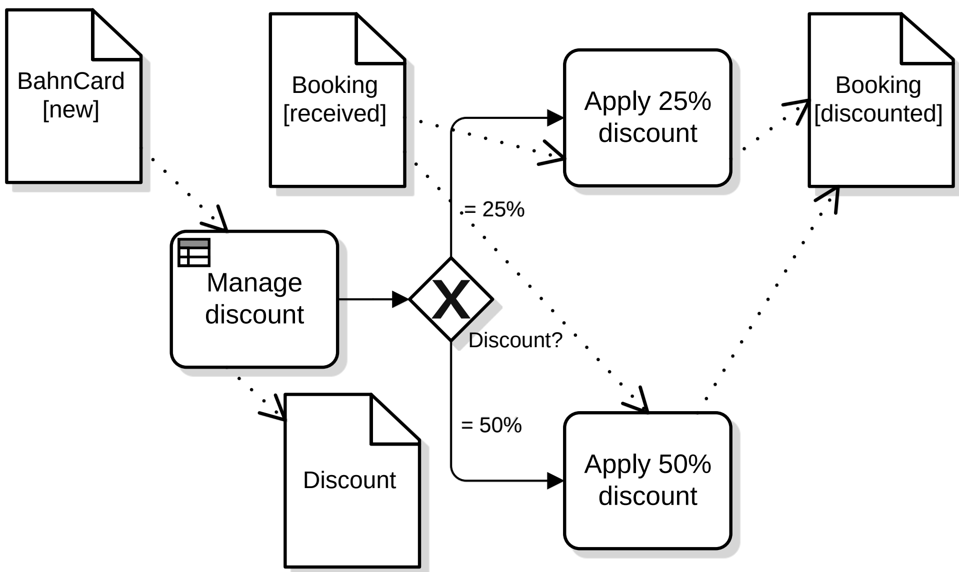


BahnCard.type *{25, 50}*	BahnCard.points *Number*	Discount *{25%, 50%, 60%, 70%}*
25	< 1000	25%
50	< 1000	50%
25	>= 1000	60%
50	>= 1000	70%

* Concrete values are made up

Output Coverage Criterion

- All outputs of the decision can be processed by the process
  - To illustrate that criterion, we use a modified decision table

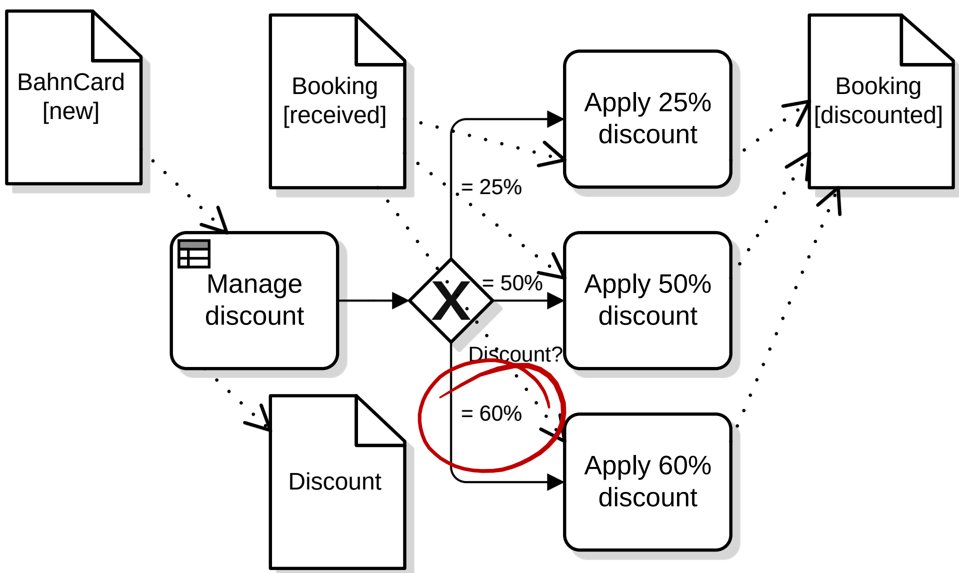


BahnCard.type *{25, 50}*	BahnCard.points *Number*	Discount *{25%, 50%, 60%, 70%}*
25	< 1000	25%
50	< 1000	50%
--	>= 1000	60%

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

Branch Coverage Criterion

- Each branch in the process model is represented by at least one row in the decision model
  - Again, the decision table is modified

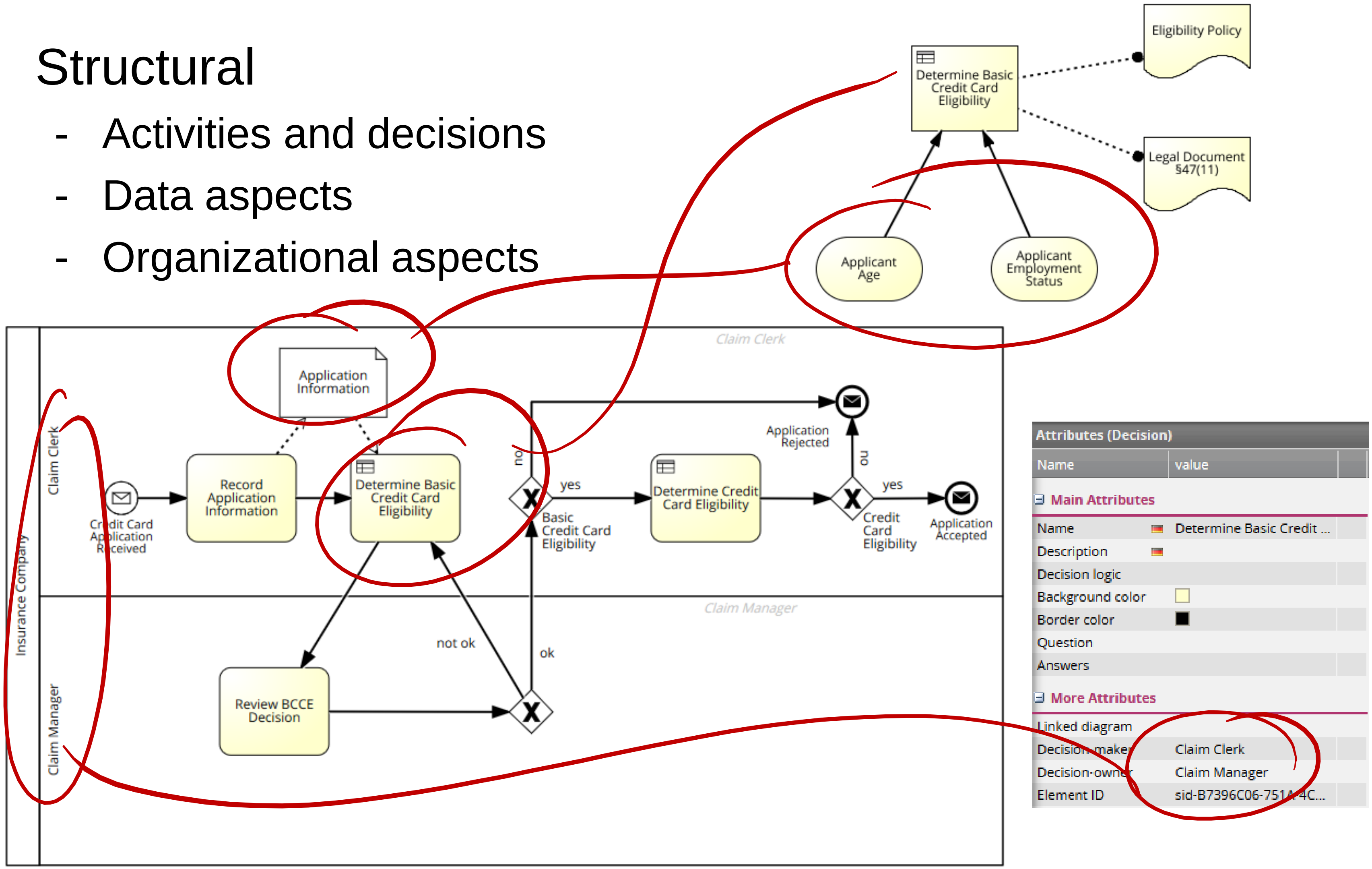


BahnCard.type *{25, 50}*	BahnCard.points *Number*	Discount *{25%, 50%}*
25	< 1000	25%
50	< 1000	50%
--	>= 1000	50%

Branch coverage criterion is violated, because
There is no row in the decision table for the branch 60%

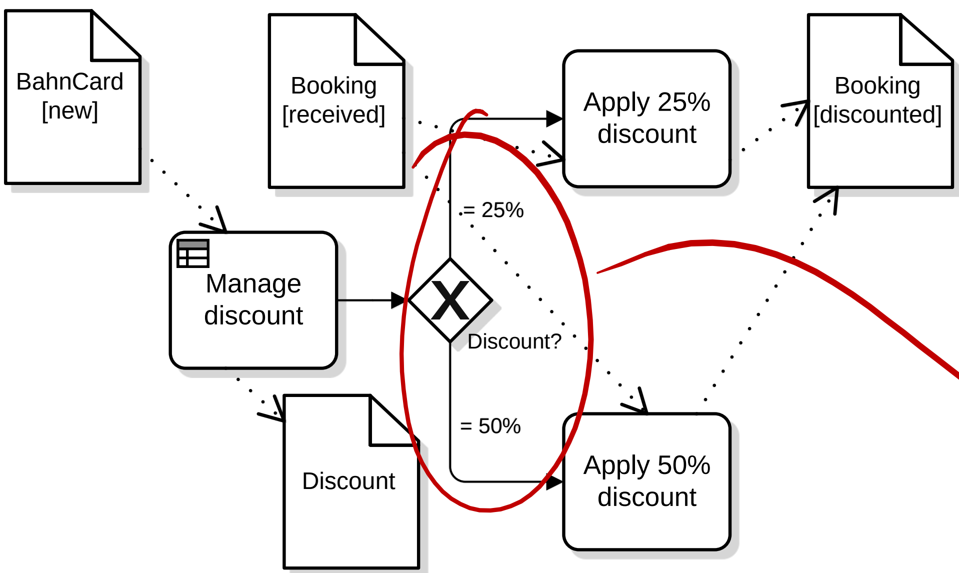
Video Clip 6.7
Consistency of Processes and Decisions

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



Video Clip 6.7
Consistency of Processes and Decisions

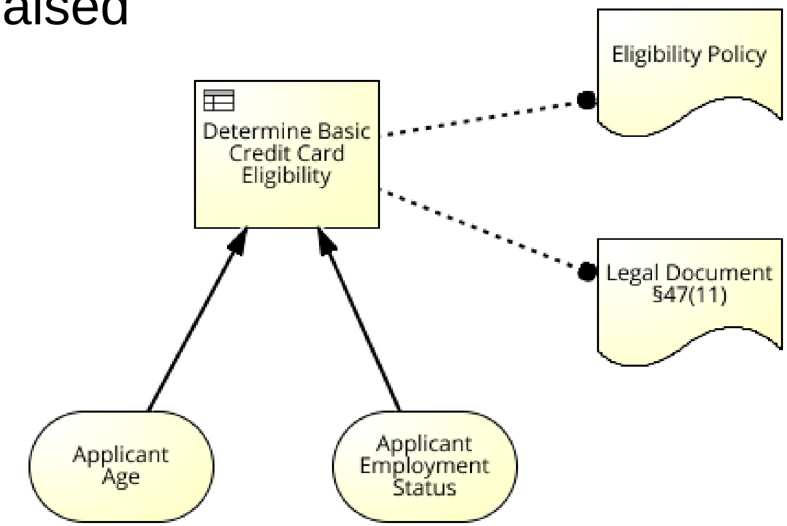
- Behavioural
  - Decisions and branches in the process
  - Output coverage
  - Branch coverage



BahnCard.type *{25, 50}*	BahnCard.points *Number*	Discount *{25%, 50%, 60%, 70%}*
25	< 1000	25%
50	< 1000	50%
--	>= 1000	60%

Summary Week 6
Business Decision Management

- 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“)

	Inputs				Outputs
U	Applicant Age		Applicant Employment Status		Basic Credit Card Eligibility
	[1..100]		{employed,unemployed}		{yes,no}
1	<	18		-	no
2	∈	[18..30]		-	yes
3	∈	[31..65]	=	employed	yes
4	∈	[31..65]	=	unemployed	no
5	>	65		-	no