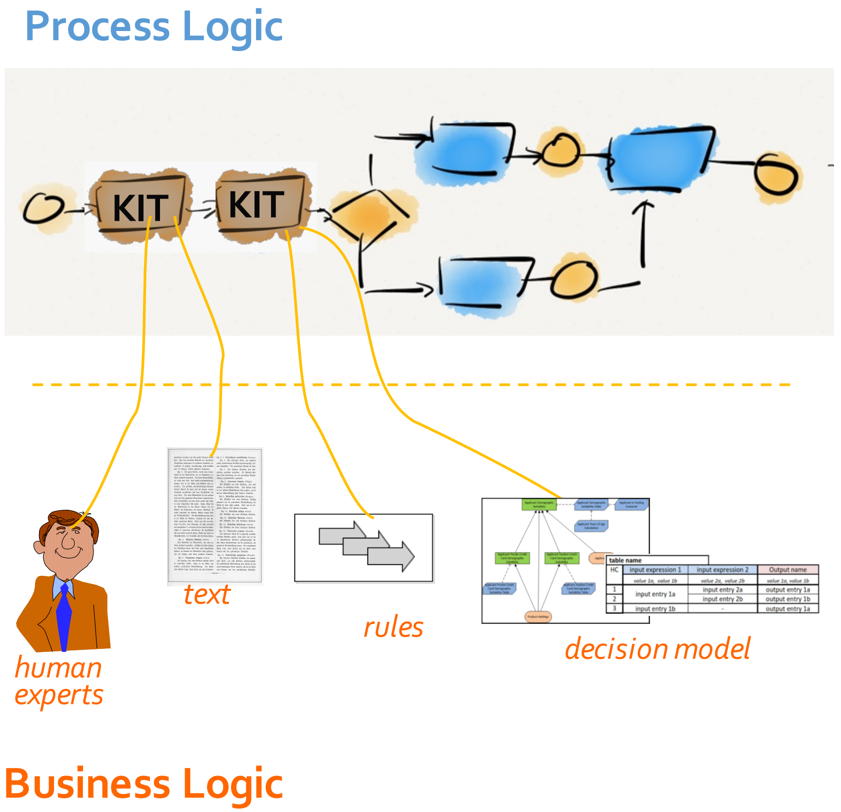
UNICAM
Università di Camerino
1336

Decision-aware Business Processes

Barbara Re

Distinguishing Process Logic and Business Logic



- The process model contains the process logic
- Business logic can be assigned to tasks in the process model:
  - knowledge-intensive tasks
- The business logic can occur in different forms
  - implicit in head of people
  - as text (e.g. guidelines)
  - as business rules
  - as decision model
  - coded in an application

Perspectives on Process Modeling

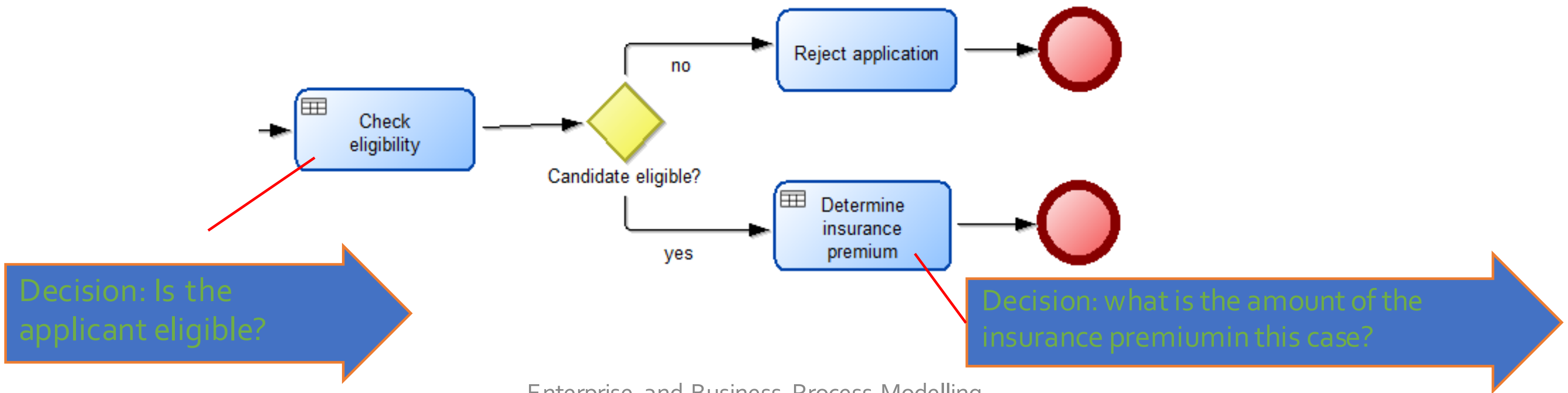
A new perspective on process modeling is reflected in the combination of three ideas

- **Process:** an organized, coordinated flow of activities, conducted by participants, acting on and deciding with data and knowledge, to achieve a business goal
- **Decision:** decisions are made by applying business knowledge in the form of business rules or other decision logic to process data. A decision model likewise reflects how a decision is made.
- **Event:** A process can also be considered a connected sequence of events that respond to states, causes, and conditions. In an event-based view, the process is a linkage of the transitions from one processing state to another.

(Debevoise & Taylor 2014)

Decision Tasks in Business Processes

- A **decision task** is a task in which some decision is made
- The business logic that is used for decision making is called *decision logic*
- Two kinds of decision tasks
  - Decision tasks deriving values for data
  - Decision tasks providing data for gateways
    - At the gateway only the result of the decision should be tested (for the selection of the path) not the criteria for the decision



Decision: Is the applicant eligible?

Decision: what is the amount of the insurance premiumin this case?

Basic Elements of Operational Business Decisions

- A decision is characterized by a *question*, for example:
  - Should the insurance claim be accepted, rejected or examined for fraud?
  - Which resource should be assigned to this task?
  - Which service should be used to ship this package?
- A *potential outcome* is some result, conclusion, or answer that might be deemed appropriate for a case. Examples:
  - some form of yes/no (e.g. eligible/non-eligible)
  - some quanitities (e.g. dollar amounts)
  - some categories (e.g. silver, gold, or platinum customer)
  - some real-world instances (e.g. software product to be purchased)
  - some course of action (e.g. on-site visit, teleconference, email)
- A *case* is some particular matter or situation arising in a day-to-day business activity and requiring consideration
- The *outcome* is the result, conclusion, or answer for a *given* case
- The business logic that is used for decision making is called *decision logic* (the set of all decision rules selecting a decision outcome)

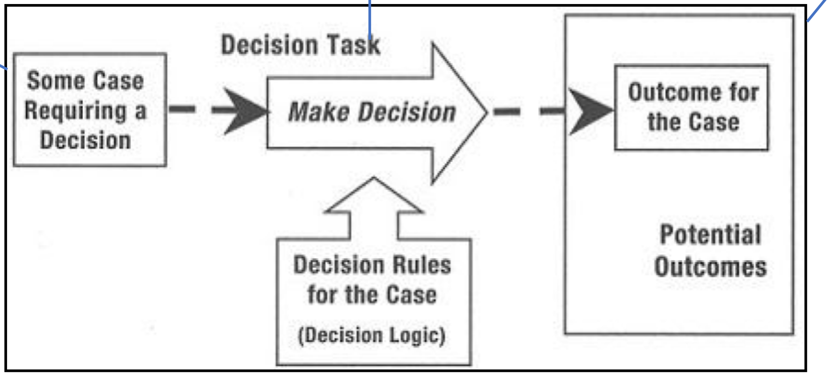
Example for a Business Decision: Data for Gateway

- Process: Handling auto insurance applications
- Decision Task: Check Eligibility of Applicant
- Potential outcomes: "yes" and "no" (i.e. eligible/non-eligible)
- Decision Logic: Terms of insurance

***Case:** John Smith applies for an auto insurance*

***Decision Task:** Check Eligibility*

***Outcome:** John Smith is eligible for auto insurance*



(Ross 2011, p. 152f)

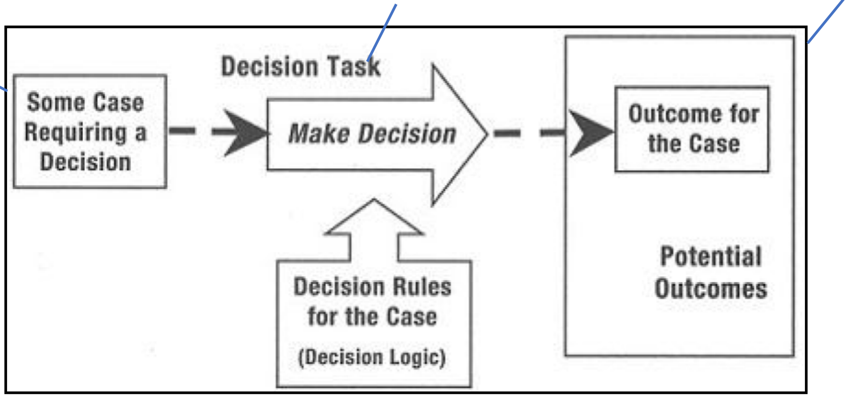
Example for a Business Decision: deriving values for data

- Process: Handling auto insurance applications
- Decision Task: Determine insurance premium
- Potential outcomes: amount of premium (i.e. amount)
- Decision Logic: Calculations for premiums

Case: *John Smith applies for an auto insurance*

Decision Task: *Determine insurance premium*

Outcome: *John Smith has to pay CHF 700 per year*



(Ross 2011, p. 152f)

Representation of Decision Rules

- There are a variety of ways to represent decision rules, e.g.
  - Semi-formal description
    - *The reimbursement is 90% if the patient visited a doctor's office and the physican was present*
  - Decision Table

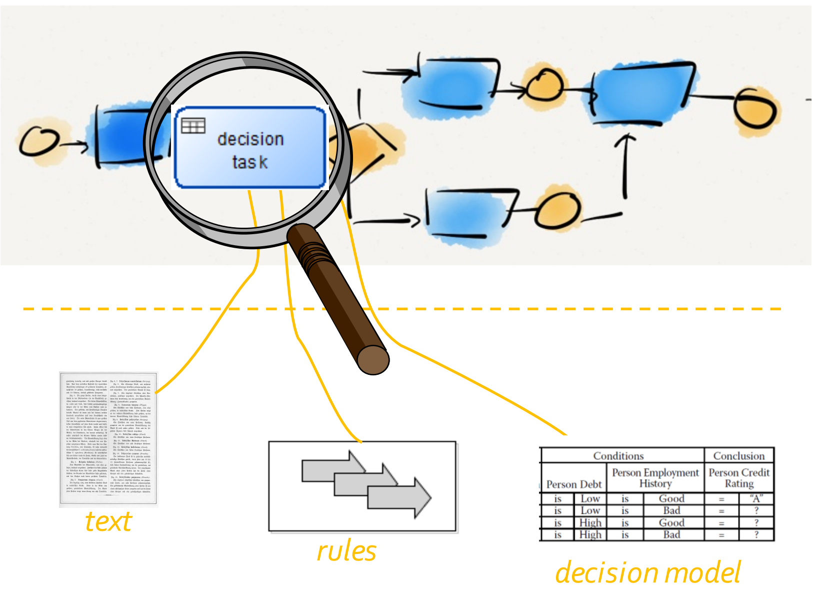
Conditions	*1*	*2*	*3*	*4*	*5*	*6*
1. Type of visit	*D*	*D*	*H*	*H*	*L*	*L*
2. Participating Physician?	*Y*	*N*	*Y*	*N*	*Y*	*N*
Effects						
1. Reimburse 50%		*X*				
2. Reimburse 70%				*X*		*X*
3. Reimburse 90%	*X*					
4. Impossible or no reimbursement			*X*		*X*	

Reimbursement depends on whether the patient visited the doctor's office (D), a hospital (H) or a lab (L) and whether the Doctor is a Participating Physician

Each column represents a rule.

Decision-Aware Process Models: Managing Process Logic and Decision Logic Separately

Process Logic



Conditions				Conclusion	
Person Debt		Person Employment History		Person Credit Rating	
is	Low	is	Good	=	"A"
is	Low	is	Bad	=	?
is	High	is	Good	=	?
is	High	is	Bad	=	?

Business Logic / Decision Logic

- The process model contains the process logic → procedural
- Decision logic is externalized from decision tasks and represented in a different kind of model → declarative
- Separating business decisions from business process tasks
  - simplifies the business process model
  - allows to manage business logic in a declarative form

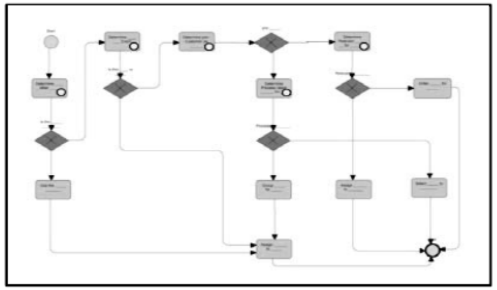
Distinguishing a Procedural Task from a Declarative Decision

- A procedural solution specifies how, in a step-by-step manner, something is to be done
  - A business process model is a procedural solution because it prescribes a set of tasks that are carried out in a particular sequence
- A declarative solution only specifies what needs to be done, with no details as to how, in a step-by-step manner, it is to be carried out, because sequence is irrelevant to arriving at the correct result
  - A Decision Table is a declarative solution because it is a set of unordered business logic, not a set of ordered tasks.

(von Halle & Goldberg 2010, p. 67)

Procedural versus Declarative

A procedural solution specifies how, in a step by step manner, something is to be done

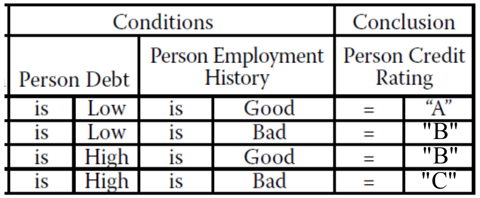


HOW

PROCESS LOGICS

Business process is a procedural solution of tasks to be performed in precise sequential order. The “How” of a unit of work.

A declarative solution is what needs to be done, with no details as to the methods to be used (no sequential information).



Conditions				Conclusion	
Person Debt		Person Employment History		Person Credit Rating	
is	Low	is	Good	=	“A”
is	Low	is	Bad	=	"B"
is	High	is	Good	=	"B"
is	High	is	Bad	=	"C"

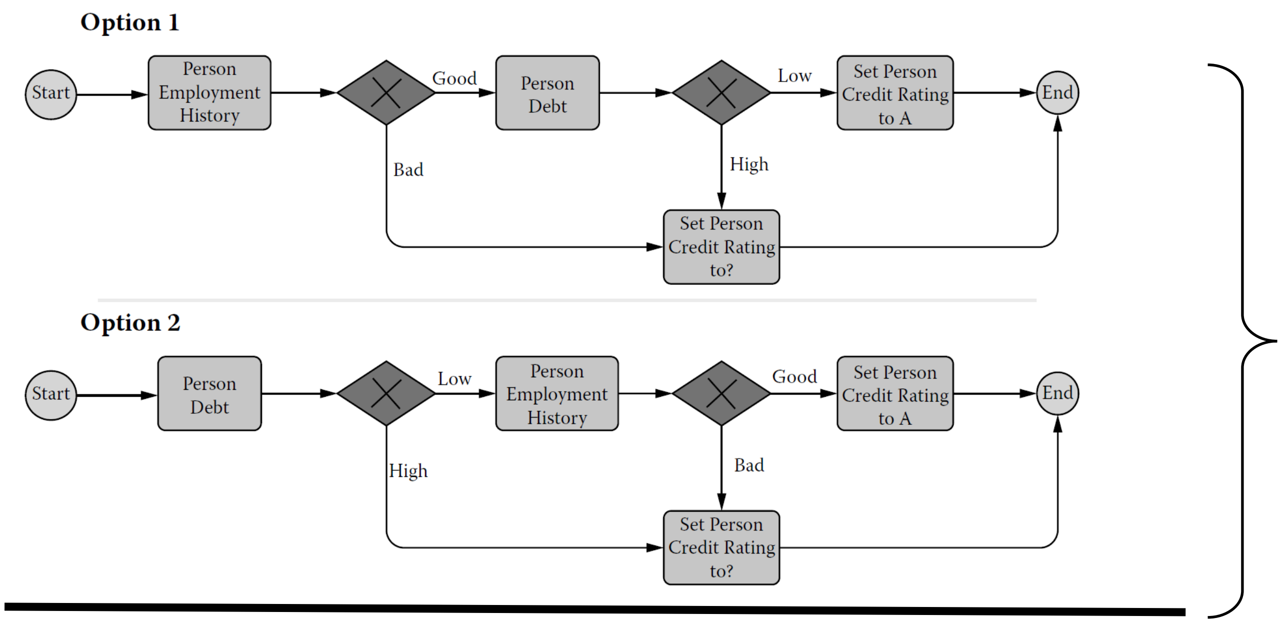
WHAT

BUSINESS LOGICS

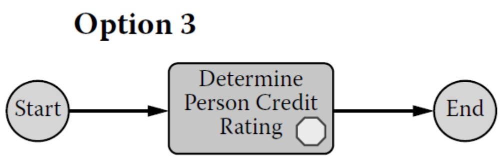
A declarative solution occurs when sequence is irrelevant to the result. The “What” of a unit of work.

(von Halle & Goldberg 2010, p. 67)

Example 1: Declarative vs. Procedural Solutions


Option 1
Start
Person Employment History
Good
Bad
Person Debt
Low
High
Set Person Credit Rating to A
Set Person Credit Rating to?
End
Option 2
Start
Person Debt
Low
High
Person Employment History
Good
Bad
Set Person Credit Rating to A
Set Person Credit Rating to?
End

Procedural



Process Model

	Conditions				Conclusion	
Rule Pattern	Person Debt		Person Employment History		Person Credit Rating	
1	is	Low	is	Good	=	"A"
1	is	Low	is	Bad	=	?
1	is	High	is	Good	=	?
1	is	High	is	Bad	=	?

Decision Table

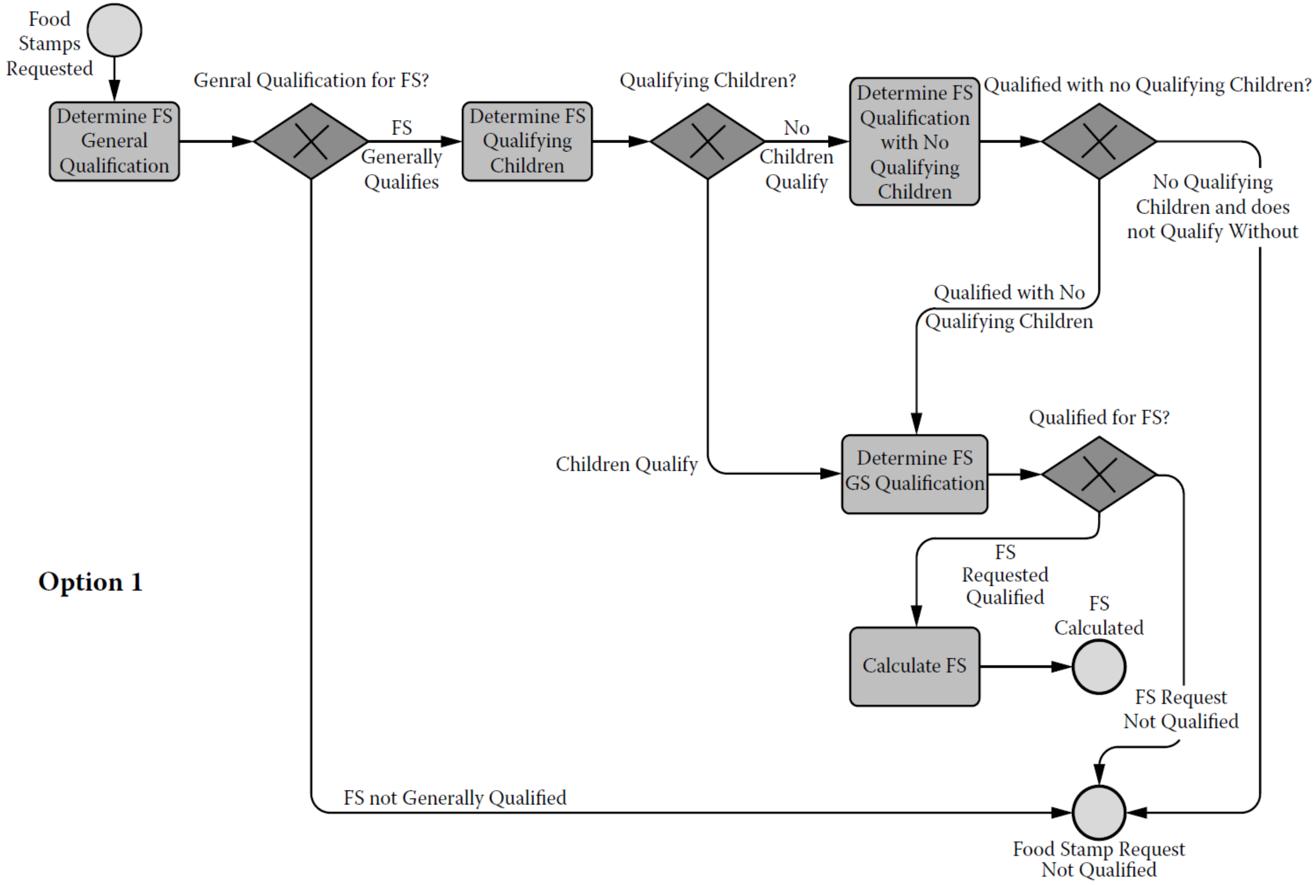
Declarative

(von Halle & Goldberg 2010, p. 69)

Advantages of Separating Business Processes and Business Logic in Option 3

- The Decision Table implies no particular sequence among the conditions to be tested
- The Decision Table easily highlights all possible combinations of conditions
- To change or add conditions in a business process model is much more cumbersome than doing so in a Decision table
    - If other conditions are needed, additional columns can be added to a decision table

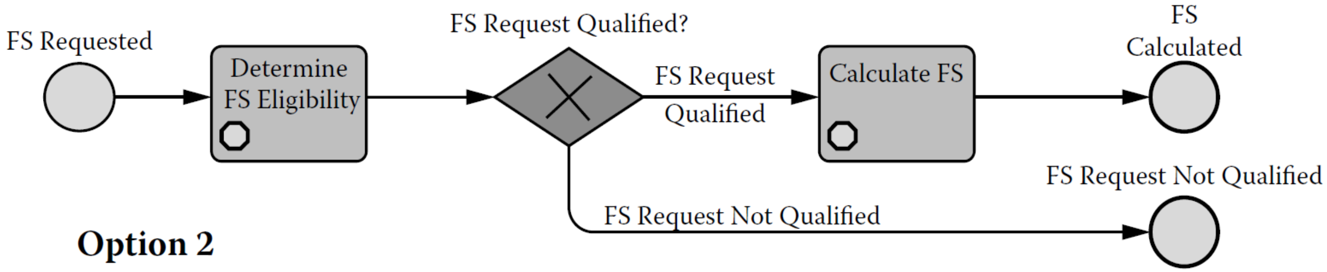
Example 2: Business Logic contained in a Process Model


Food Stamps Requested
Determine FS General Qualification
Genral Qualification for FS?
FS Generally Qualifies
Determine FS Qualifying Children
Qualifying Children?
No Children Qualify
Determine FS Qualification with No Qualifying Children
Qualified with no Qualifying Children?
No Qualifying Children and does not Qualify Without
Qualified with No Qualifying Children
Children Qualify
Determine FS GS Qualification
Qualified for FS?
FS Requested Qualified
FS Calculated
Calculate FS
FS Request Not Qualified
FS not Generally Qualified
Food Stamp Request Not Qualified

How many main decisions?

Option 1

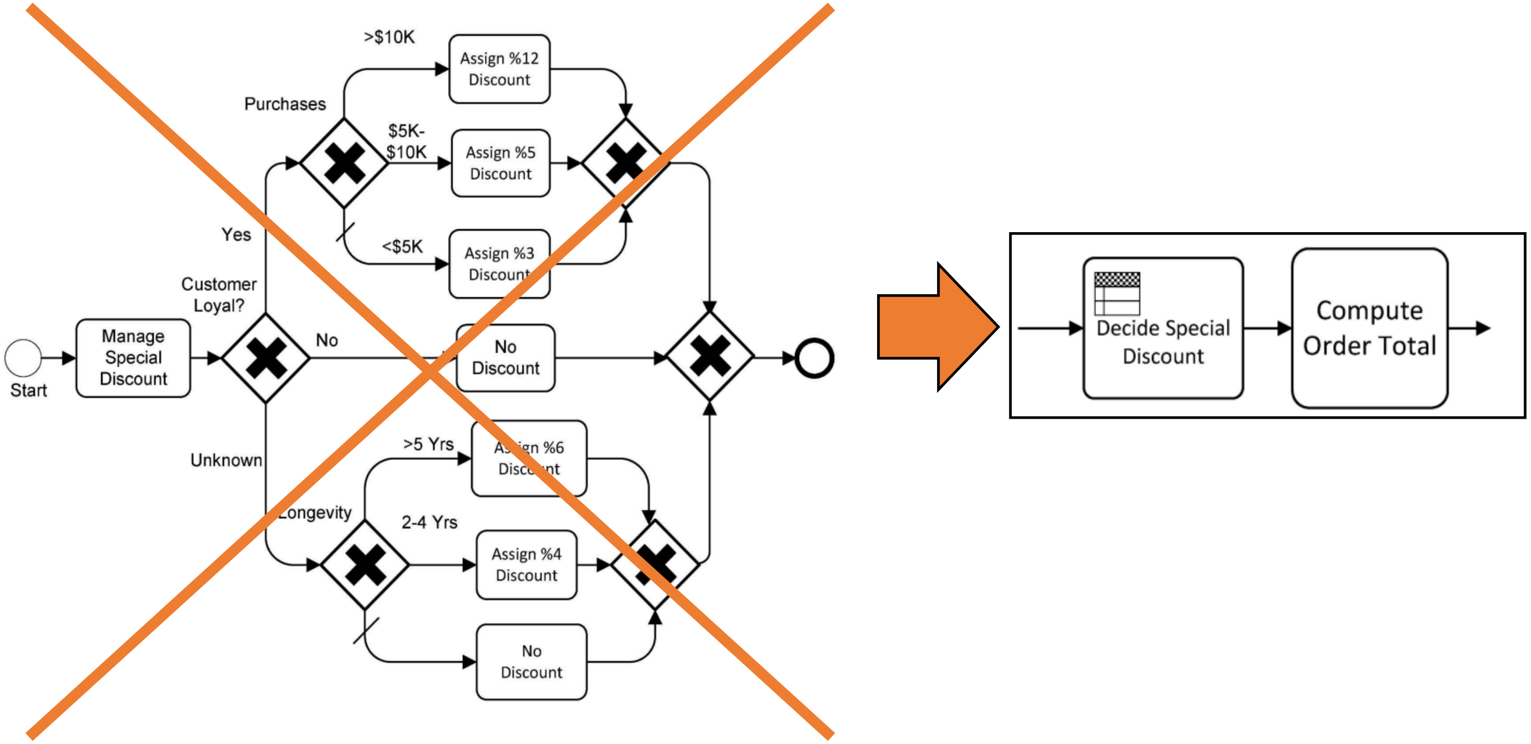
Managing Business Logic separately



- This solution has two tasks with their Decision Models
- The Decision Models can be viewed, managed, and executed as one whole set of business logic
- The process model is simplified
- The decision logic is a black box evaluating conditions and reaching a conclusion
- Business Logic can be reused: (i) the whole decision model and (ii) Individual decision tables/rules

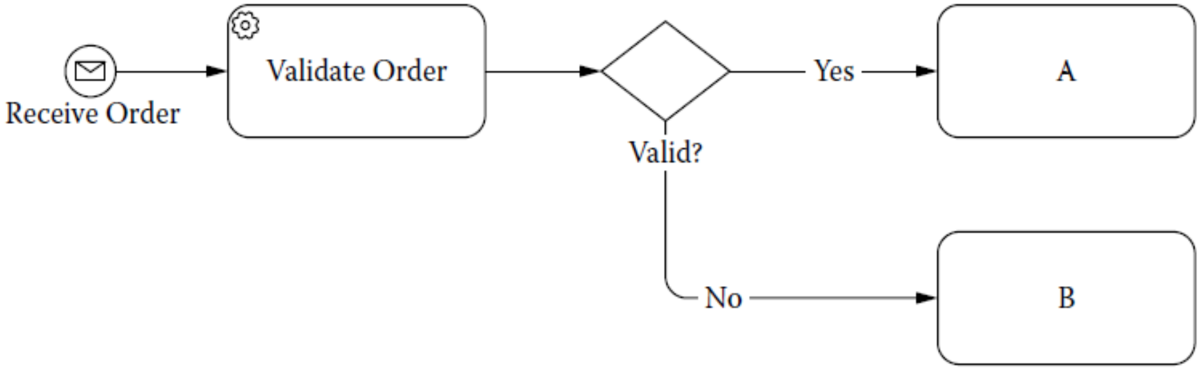
(von Halle & Goldberg 2010, p. 71f)

Example 3: Collapsing gateways for a complex discount decision into a decision

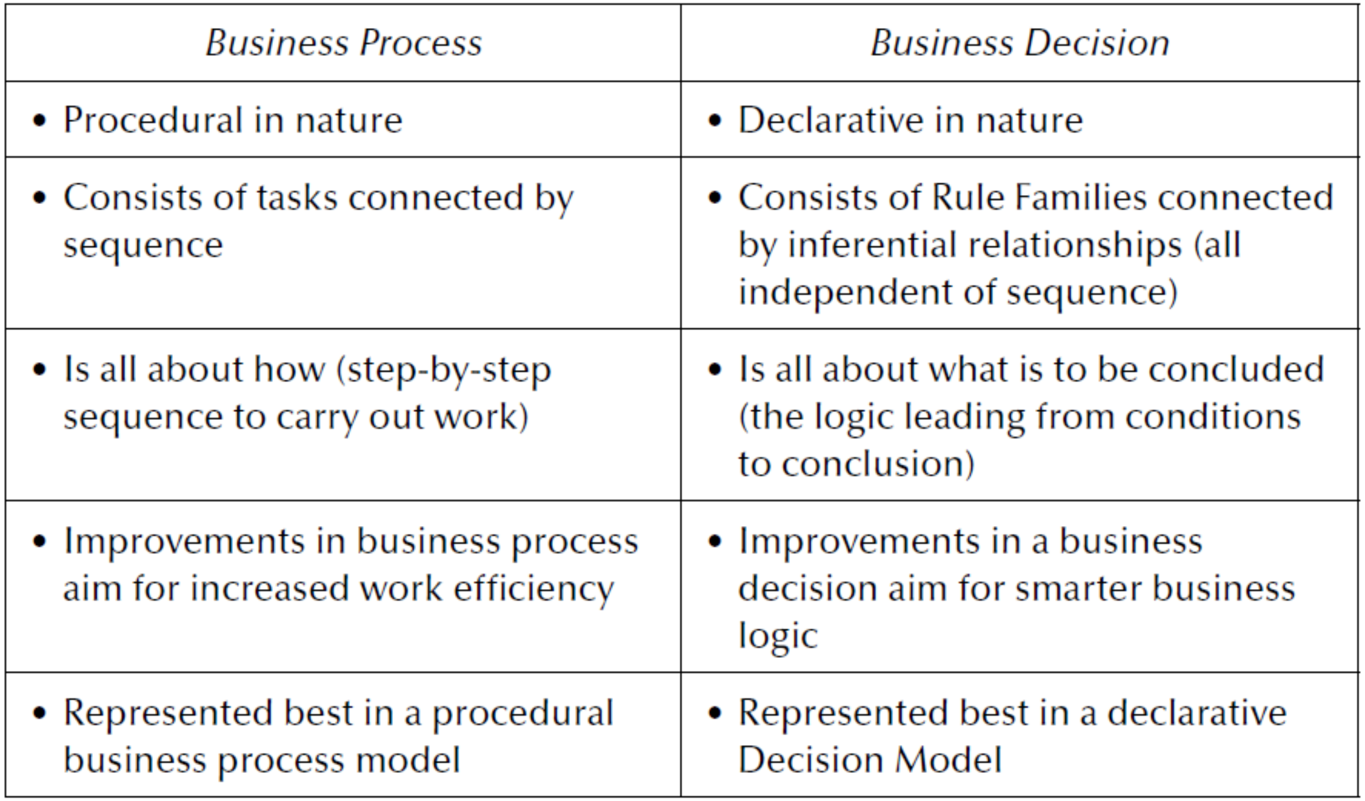


Integrating the Decision Model with BPMN

- Execution of a decision described by a Decision Model is a particular type of task in BPMN
- In BPMN 2.0 the corresponding task type is called a business rule task
- In the figure below, Validate Order is the decision task. Its logic is described by a Decision Model. The gateway simply tests the output of the decision and routes the flow either to A or B based on the result



Distinctions between Business Process and Business Decision



Business Process	*Business Decision*
• Procedural in nature	• Declarative in nature
• Consists of tasks connected by sequence	• Consists of Rule Families connected by inferential relationships (all independent of sequence)
• Is all about how (step-by-step sequence to carry out work)	• Is all about what is to be concluded (the logic leading from conditions to conclusion)
• Improvements in business process aim for increased work efficiency	• Improvements in a business decision aim for smarter business logic
• Represented best in a procedural business process model	• Represented best in a declarative Decision Model

(von Halle & Goldberg 2010, p.70)

Advantages of separating Business Logic from Business Process Model

- Allows a much simpler business process model
  - If a business process is too complicated, a reason might be that business rules are embedded in the flow
- Makes changes to business process and business logic easier
  - Permits changes in the Decision Model without changing the business process model and vice versa
- Makes governance of business processes and business logic easier to manage
- Decision Model can be reused in several processes
  - the whole decision model
  - individual decision tables and rules

Achieving Business Excellence by Managing Decision Logic Separately

- von Halle and Goldberg argue that operational excellence alone is insufficient for sustainable competitive advantage
- Key business processes must not only be efficient and consumer-friendly but also smart and agile
  - Business processes become agile when declarative business decisions are separated from procedural business process tasks
  - Business processes become smart when the business decisions are governed appropriately by business leaders
- When the business leadership clearly understands the business logic behind the business decisions, the impact of those decisions can be ascertained, and the business can quickly and easily make adjustments.

(von Halle & Goldberg 2010, p. 78)

Literatur

- Von Halle, B., & Goldberg, L. (2010). *The Decision Model: A Business Logic Framework Linking Business and Technology*. CRC Press Auerbach Publications.
- Tom Debevoise and James Taylor (2014) The Microguide to Process and Decision Modeling in BPMN/DMN.

UNICAM
Università di Camerino
1336

Decision Model and Notation (DMN)

Barbara Re

OBJECT MANAGEMENT GROUP

Two Different Perspectives

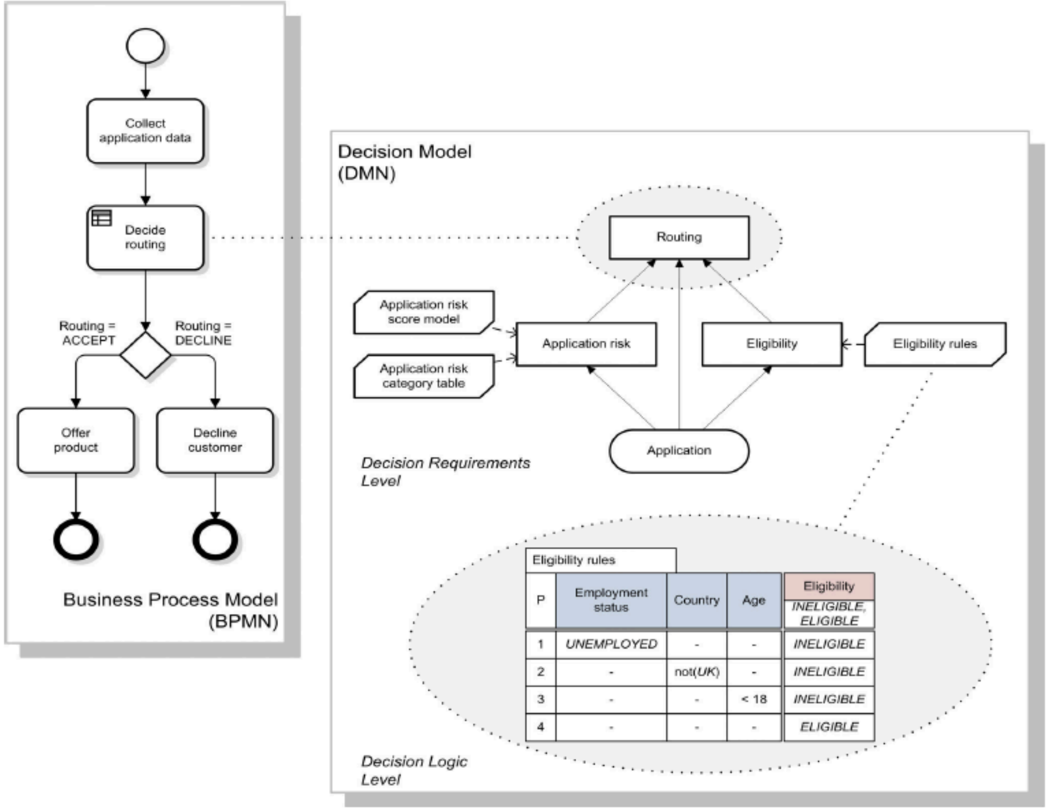
- Decision-making is addressed from two different perspectives by existing modeling standards
  - **Business process models** (e.g., **BPMN**) can describe the coordination of decision-making within business processes by defining specific tasks or activities within which the decision-making takes place
  - **Decision logic** (e.g., PRR, PMML) can define the specific logic used to make individual decisions, for example as business rules, decision tables, or executable analytic models

Decision Requirements Diagram

DMN will provide a third perspective – **the Decision Requirements Diagram** – forming a bridge between business process models and decision logic models:

- Business process models will define tasks within business processes where decision-making is required to occur
- Decision Requirements Diagrams will define the decisions to be made in those tasks, their interrelationships, and their requirements for decision logic
- Decision logic will define the required decisions in sufficient detail to allow validation and/or automation

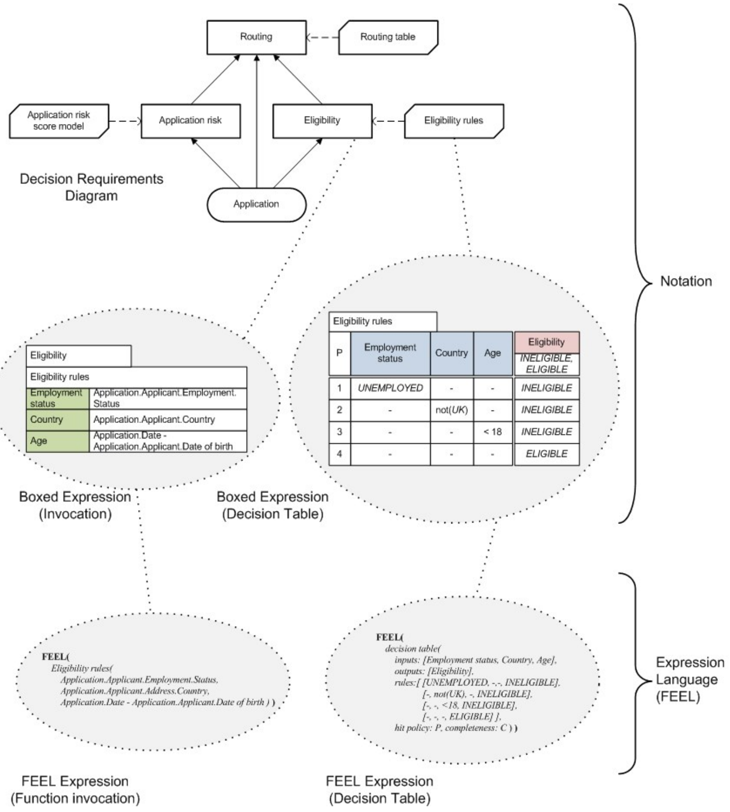
Aspects of Modelling



Eligibility rules

P	Employment status	Country	Age	Eligibility INELIGIBLE, ELIGIBLE
1	UNEMPLOYED	-	-	INELIGIBLE
2	-	not(UK)	-	INELIGIBLE
3	-	-	< 18	INELIGIBLE
4	-	-	-	ELIGIBLE

Decision Model and Notation (DMN)


Routing
Routing table
Application risk score model
Application risk
Eligibility
Eligibility rules
Application
Decision Requirements Diagram
Notation
Expression Language (FEEL)

Eligibility	
Eligibility rules	
Employment status	Application.Applicant.Employment.Status
Country	Application.Applicant.Country
Age	Application.Date - Application.Applicant.Date of birth

Boxed Expression (Invocation)

Eligibility rules

P	Employment status	Country	Age	Eligibility *INELIGIBLE, ELIGIBLE*
1	*UNEMPLOYED*	-	-	*INELIGIBLE*
2	-	not(*UK*)	-	*INELIGIBLE*
3	-	-	< 18	*INELIGIBLE*
4	-	-	-	*ELIGIBLE*

Boxed Expression (Decision Table)

```
FEEL(
  Eligibility rules(
    Application.Applicant.Employment.Status,
    Application.Applicant.Address.Country,
    Application.Date - Application.Applicant.Date of birth ) )
```

FEEL Expression (Function invocation)

```
FEEL(
  decision table(
    inputs: [Employment status, Country, Age],
    outputs: [Eligibility],
    rules:[ [UNEMPLOYED, -,-, INELIGIBLE],
            [-, not(UK), -, INELIGIBLE],
            [-, -, <18, INELIGIBLE],
            [-, -, -, ELIGIBLE] ],
    hit policy: P, completeness: C ) )
```

FEEL Expression (Decision Table)

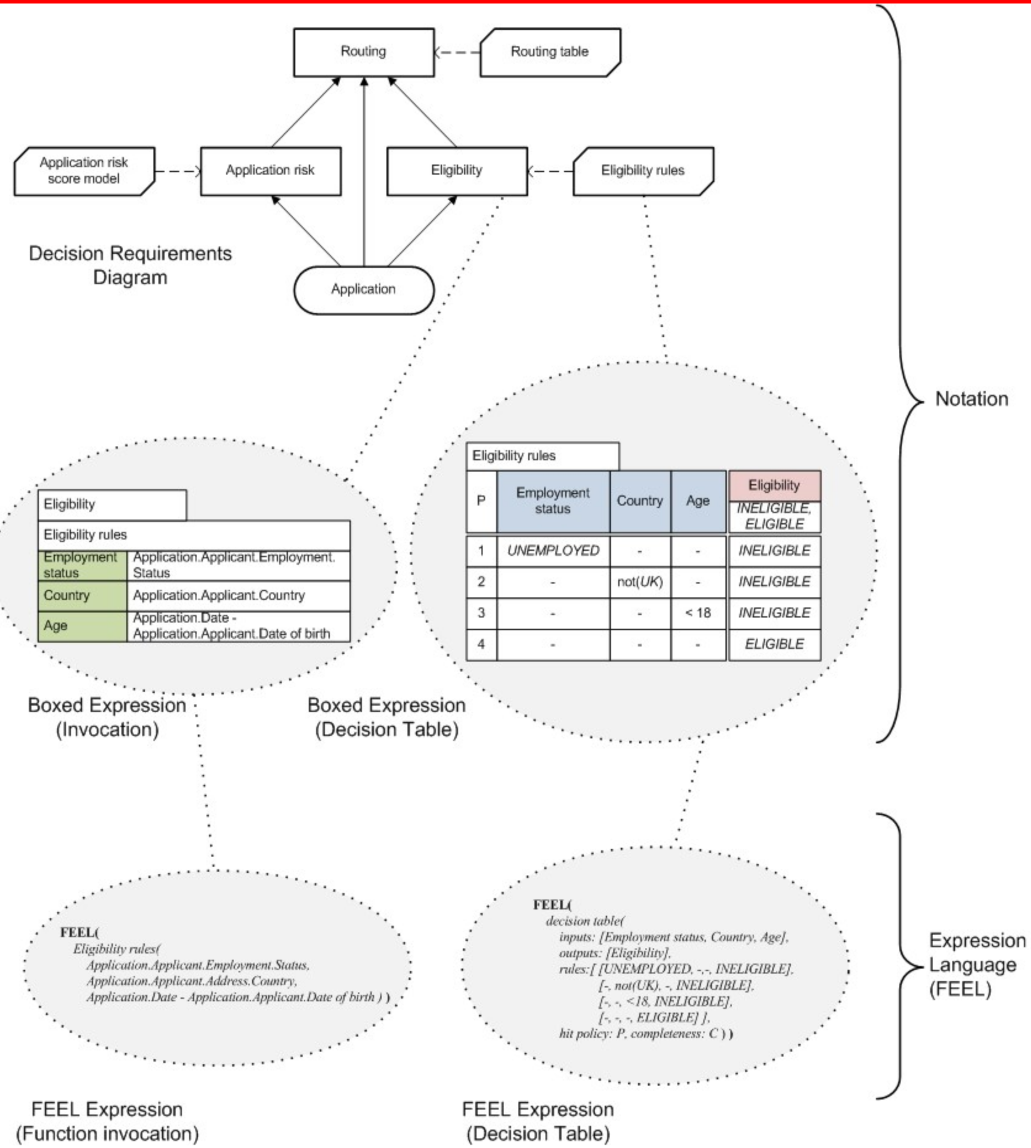
- The Decision Model and Notation is a new standard from the OMG
- It is currently published in its version 1.0
- Purpose of DMN: provide the constructs that are needed to model decision, so that organizational decision-making can be
  - readily depicted in diagrams
  - accurately defined by business analysts
  - (optionaly) automated

DMN

It defines the concept of a Decision Requirements Graph (DRG) comprising a set of elements and their connection rules, and a corresponding notation: the Decision Requirements Diagram (DRD)

It defines Decision Logic via Value expression and/or decision table

It provides a language called FEEL for defining and assembling decision tables, calculations, if/then/else logic, simple data structures, and externally defined logic from Java and PMML into executable expressions with formally defined semantic


Routing
Routing table
Application risk score model
Application risk
Eligibility
Eligibility rules
Application
Decision Requirements Diagram
Boxed Expression (Invocation)
Boxed Expression (Decision Table)
Notation
FEEL Expression (Function invocation)
FEEL Expression (Decision Table)
Expression Language (FEEL)

Eligibility	
Eligibility rules	
Employment status	Application.Applicant.Employment.Status
Country	Application.Applicant.Country
Age	Application.Date - Application.Applicant.Date of birth

Eligibility rules

P	Employment status	Country	Age	Eligibility *INELIGIBLE, ELIGIBLE*
1	*UNEMPLOYED*	-	-	*INELIGIBLE*
2	-	not(*UK*)	-	*INELIGIBLE*
3	-	-	< 18	*INELIGIBLE*
4	-	-	-	*ELIGIBLE*

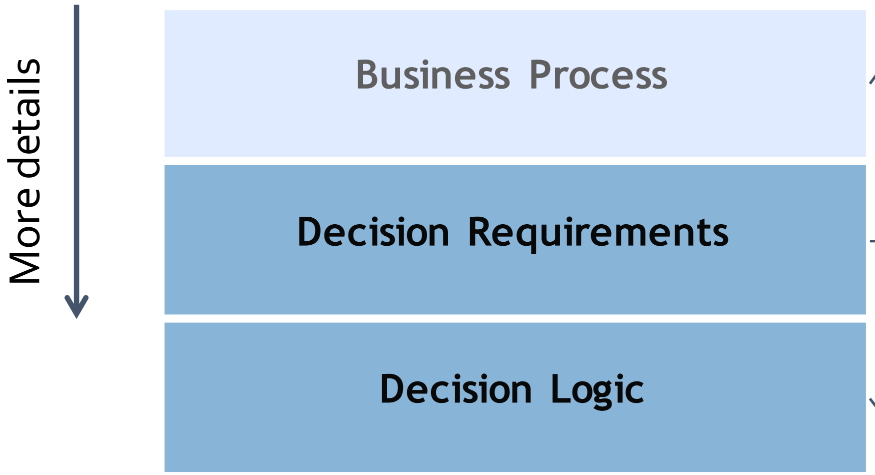
```
FEEL(
   Eligibility rules(
      Application.Applicant.Employment.Status,
      Application.Applicant.Address.Country,
      Application.Date - Application.Applicant.Date of birth ) )
```

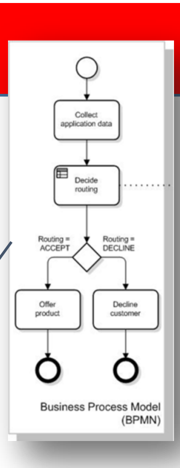
```
FEEL(
   decision table(
      inputs: [Employment status, Country, Age],
      outputs: [Eligibility],
      rules:[ [UNEMPLOYED, -,-, INELIGIBLE],
              [-, not(UK), -, INELIGIBLE],
              [-, -, <18, INELIGIBLE],
              [-, -, -, ELIGIBLE] ],
      hit policy: P, completeness: C ) )
```

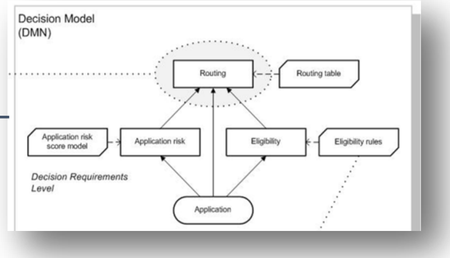
Scope and Uses of DMN

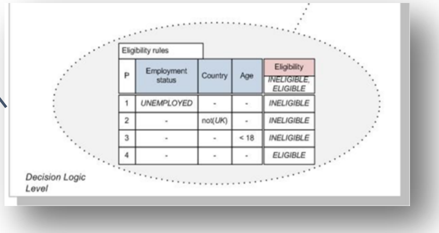
- Decision modeling is carried out by business analysts in order to understand and define the decisions used in a business or organization
- Such decisions are typically operational decisions made in day-to-day business processes, rather than the strategic decision-making for which fewer rules and representations exist.
- Three uses of **DMN** can be discerned in this context:
  - For modeling human decision-making
  - For modeling the requirements for automated decision-making
  - For implementing automated decision-making

Main Concepts of DMN









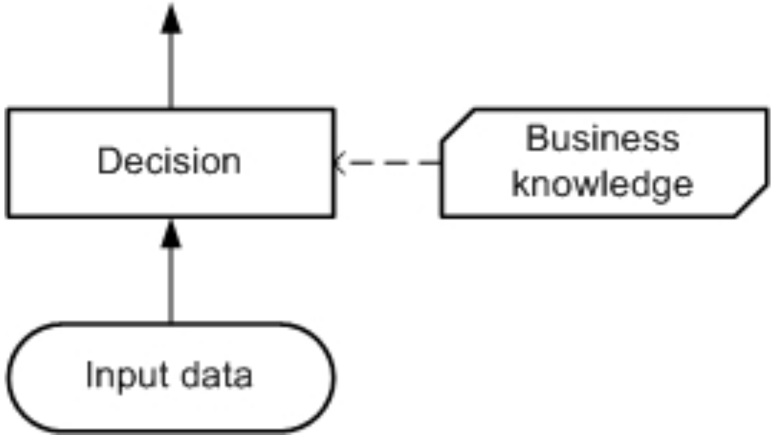
Eligibility rules

P	Employment status	Country	Age	Eligibility
				INELIGIBLE, ELIGIBLE
1	*UNEMPLOYED*	-	-	*INELIGIBLE*
2	-	not(*UK*)	-	*INELIGIBLE*
3	-	-	< 18	*INELIGIBLE*
4	-	-	-	*ELIGIBLE*

Decision Logic Level

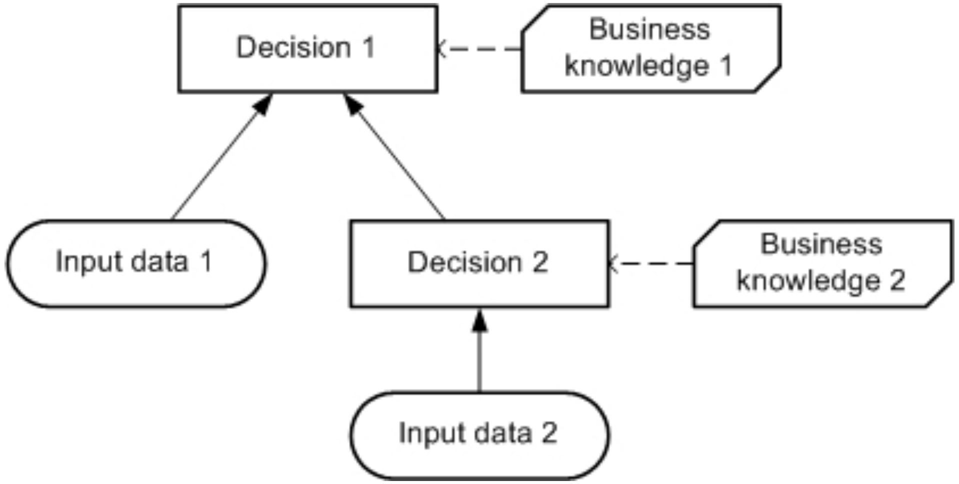
Basic Concepts – Decision Requirements Level

- A **decision** is the act of determining an **output** value (the chosen option), from a number of **input** values, using logic defining how the output is determined from the inputs
- **Decision logic** may include one or more **business knowledge models** which encapsulate business know-how



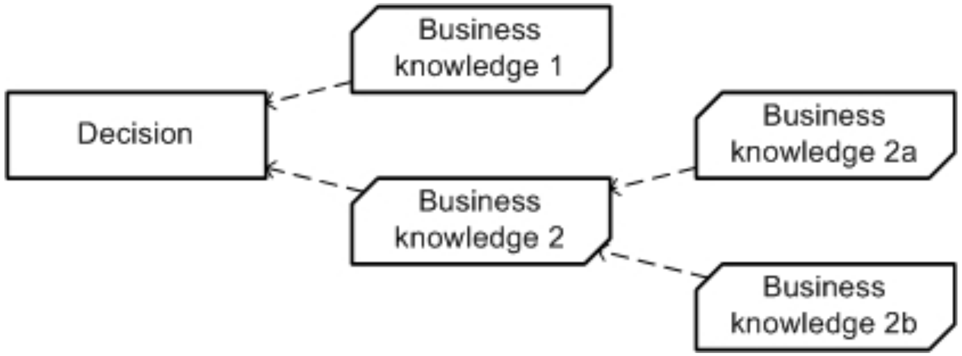
Basic Concepts – Decision Requirements Level

- Decisions can be decomposed into sub-decisions. Top level decisions can be thought of as selecting an answer from a range of possible answers. Lower level decisions often will simply provide input to other decisions
- Decisions may therefore be connected in a network called a **Decision Requirements Graph (DRG)**, which may be drawn as a **Decision Requirements Diagram** (**DRD**)



Basic Concepts – Decision Requirements Level

- A decision may require multiple business knowledge models, and a business knowledge model may require multiple other business knowledge models

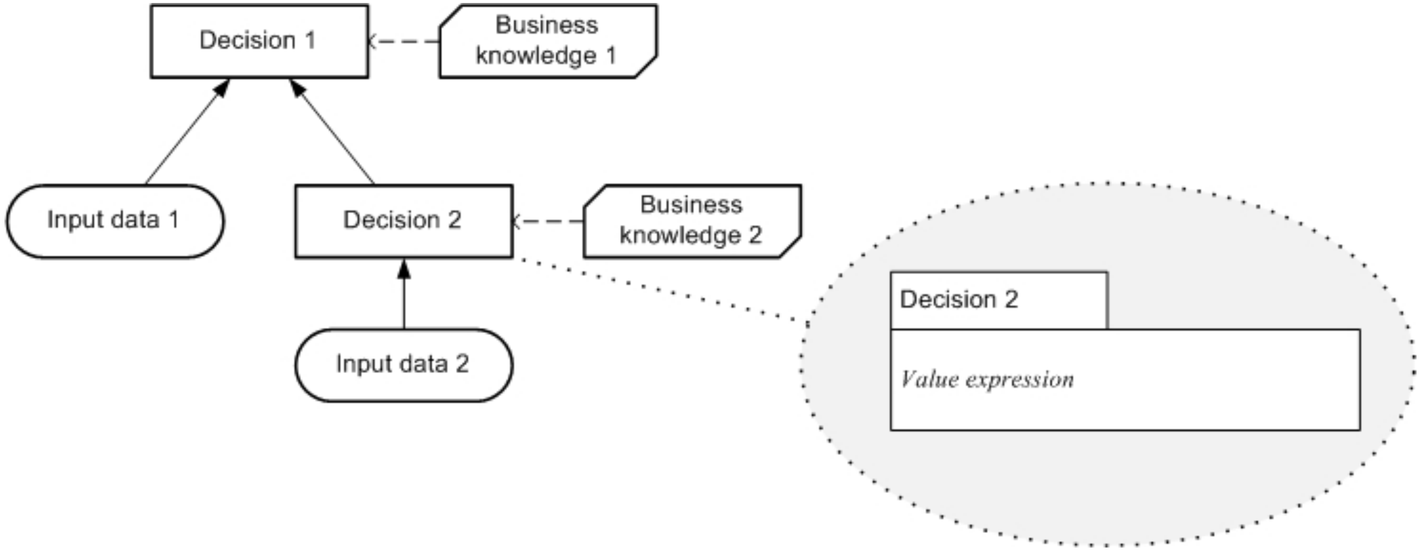


Basic Concepts – Decision Logic Level

- Using decision logic, the same components described at decision requirements level may be specified in greater detail, to capture:
  - a complete set of **business rules and calculations**
  - **(if desired) to allow the decision-making to be fully automated**

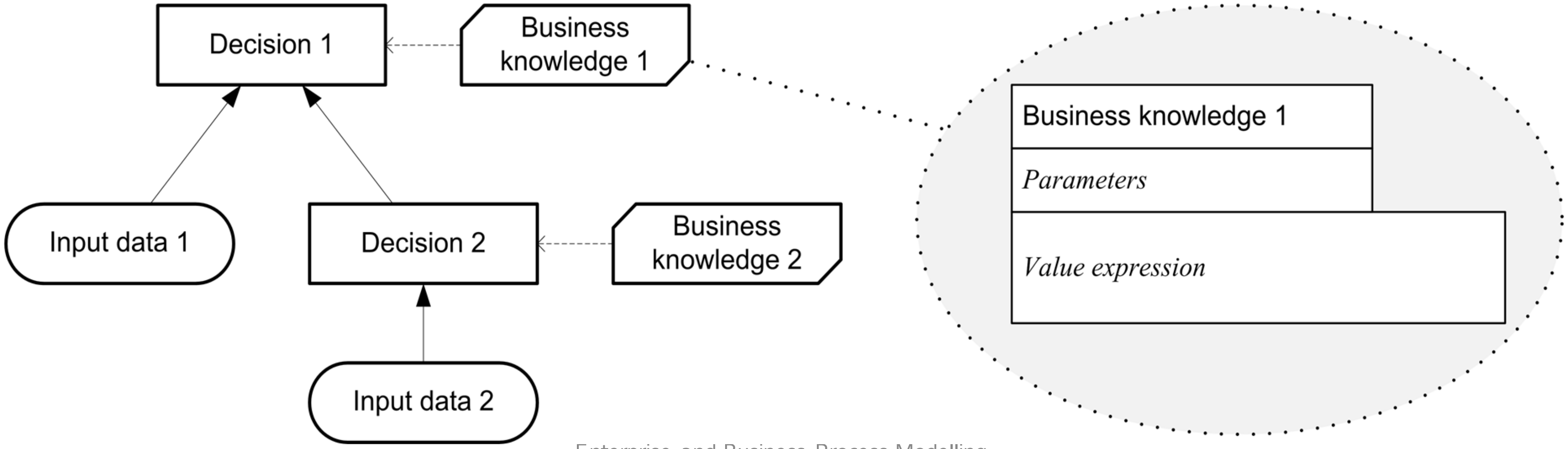
Decision and corresponding value expression

- At the decision logic level, every decision in a DRG is defined using a **value expression** which specifies how the decision's output is determined from its inputs
- The decision is considered to *be* the evaluation of the expression
- The value expression may be notated using a **boxed expression**



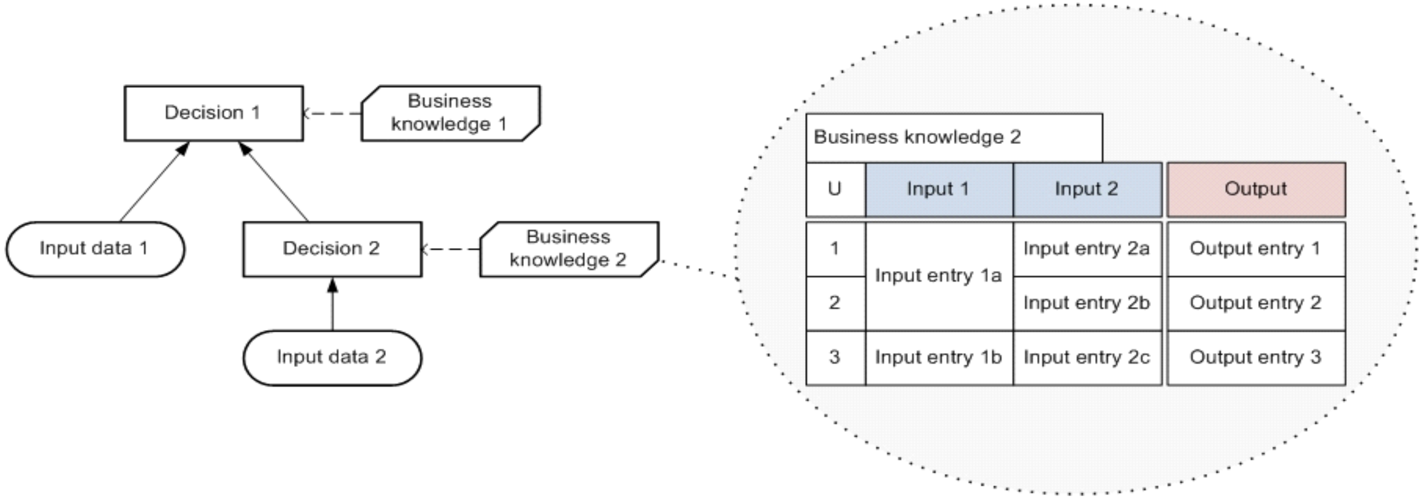
Business knowledge model and corresponding value expression

- At the decision logic level, a business knowledge model is defined using a value expression that specifies how an output is determined from a set of inputs
- Value expressions may be encapsulated as functions, which may be invoked from decisions' value expressions



Business knowledge model and corresponding decision table

- A business knowledge model may contain any decision logic which is capable of being represented as a function
- This will allow the import of many existing decision logic modeling standards (e.g., for business rules and analytic models) into **DMN**
- An important format of business knowledge, specifically supported in **DMN**, is the Decision Table
- Such a business knowledge model may be notated using a **Decision Table**



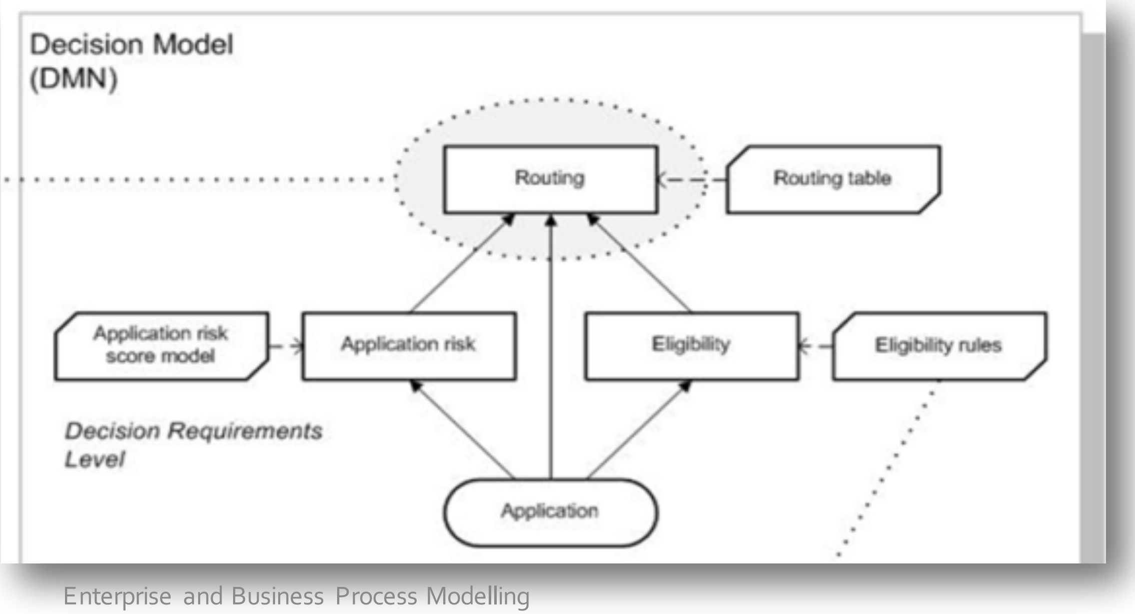
Business knowledge 2

U	Input 1	Input 2	Output
1	Input entry 1a	Input entry 2a	Output entry 1
2		Input entry 2b	Output entry 2
3	Input entry 1b	Input entry 2c	Output entry 3

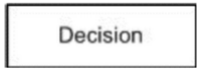
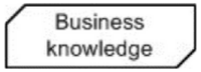
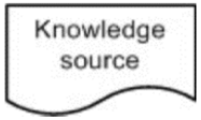
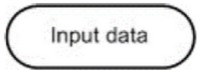
Main concepts – Decision Requirements Level

- Business concepts only
- Business decisions
- Areas of business knowledge
- Sources of business knowledge

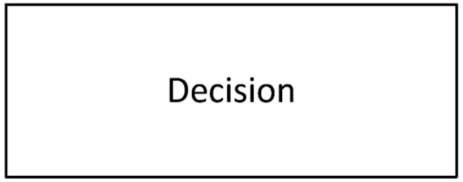
Decision Requirements



Constructs of a Decision Requirements Model

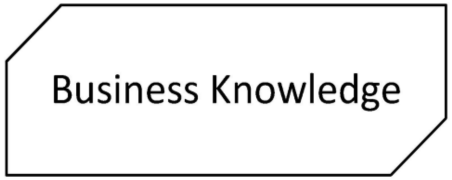
Construct	DMN Notation	Description
ELEMENTS		
Decision		The act of determining an output from a number of inputs, using decision logic which may reference one or more business knowledge models.
Business Knowledge Model		A function encapsulating business knowledge, in the form of business rules, decision table or analytic model. Some of the tool may not support this element. In such case the decision logic is directly linked to the Decision rather than the business knowledge model.
Knowledge Source		The authority for a business knowledge model or decision.
Input Data		Information used as an input by one or more decisions. It also denotes the parameters of a Business Knowledge Model.
REQUIREMENTS		
Information Requirement	⟶	Information - input data or decision output - required for a decision.
Knowledge Requirement	- - - - - - - - - - →	The invocation of a business knowledge model.
Authority Requirement	- - - - - - - - - - ●	Showing the knowledge source of an element or the dependency of a knowledge source on input data.

Decision



- A **decision** is the act of determining an **output** value (the chosen option), from a number of **input** values, using logic defining how the output is determined from the inputs
- Two properties should be captured for every decision:
  - Question: A natural language statement that represents the decision in the form of a question. This should be specific and detailed.
  - Allowed Answers: A natural language description of the possible answers to this question.
- For action-oriented decisions, the allowed answers represent the responses that the process must handle when the decision model is invoked by a business rule task

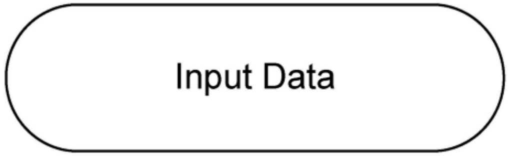
Business Knowledge Models



Business Knowledge

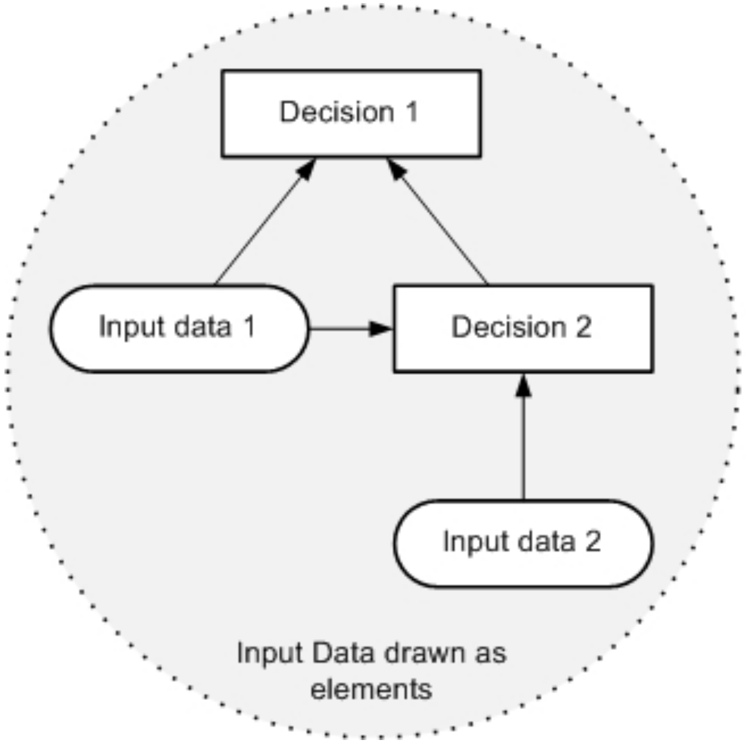
- Business knowledge models represent functions that encapsulate reusable decision making logic. The logic they encapsulate might be a set of business rules, a decision tree, a decision table, or an analytic model.
- The specifics of knowledge representation involved need not be displayed on the diagram but could be.
- The decision logic that can be specified in a business knowledge model can also be linked directly to a decision, but encapsulating it in a business knowledge model allows it to be reused, parameterized and displayed on a Decision Requirements Diagram

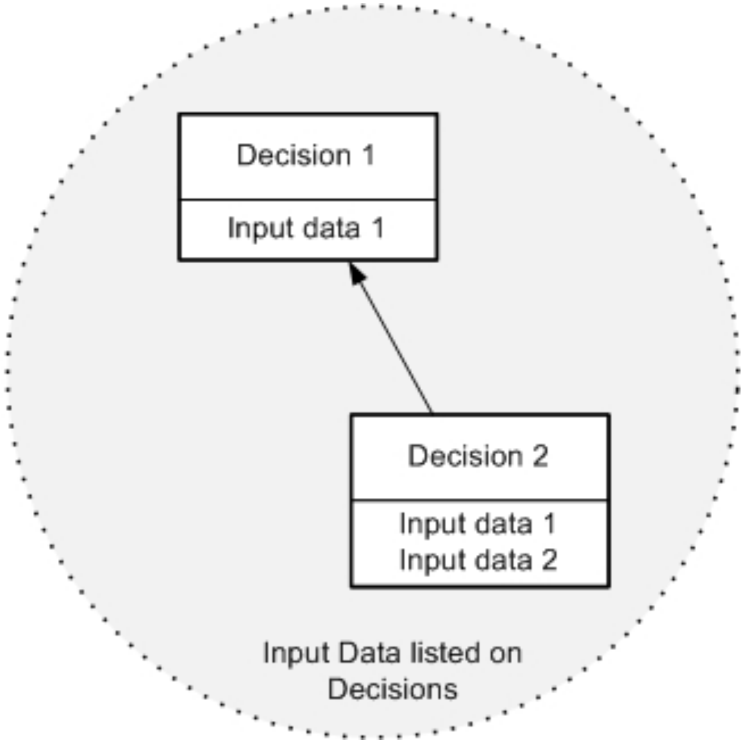
Input Data



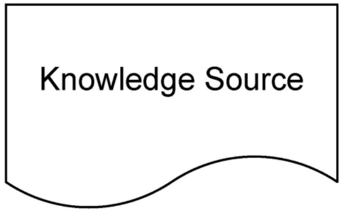
- Decisions require inputs, and many of these are input data, which is data that is input to the decision making from outside the decision context.
- Input data elements typically represent business entities that are being used in the decision making, such as Policy or Customer. However, sometimes they can represent any information element at any level of detail.
- Each input data element can be described in terms of a hierarchical information model that specifies exactly what information elements comprise the input data.

Decision with the Listed Input Data option





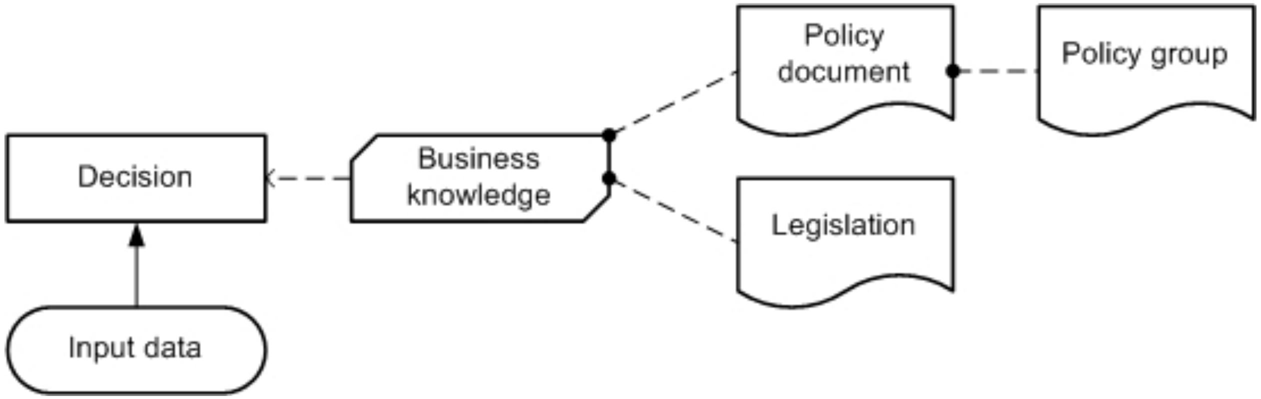
Knowledge Source



- Knowledge sources represent the source of know-how for making a decision
- This could be:
  - **Regulations or policies** about how a decision must be made, best practices or expertise on how it should be made
  - **Analytic knowledge** on how it might be made more accurate

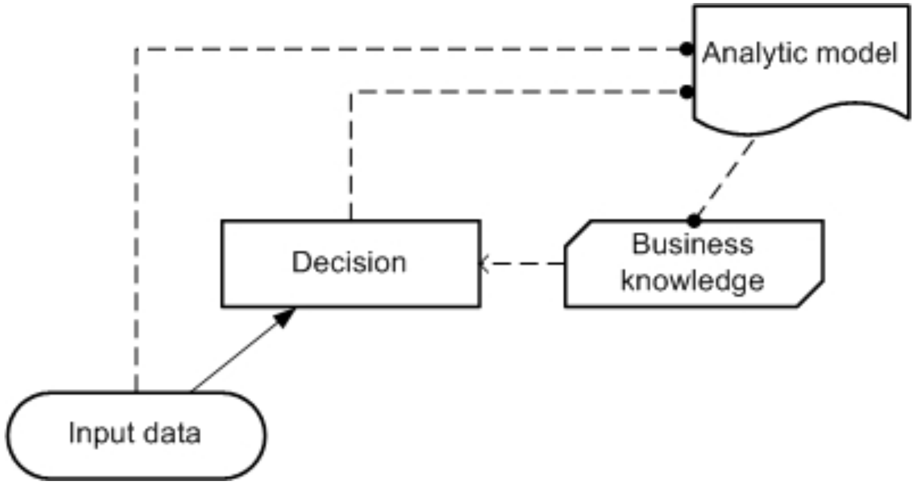
Knowledge Sources representing authorities

- Knowledge sources are the authorities for a decision and typically refer to some external document or source that contains detailed guidance

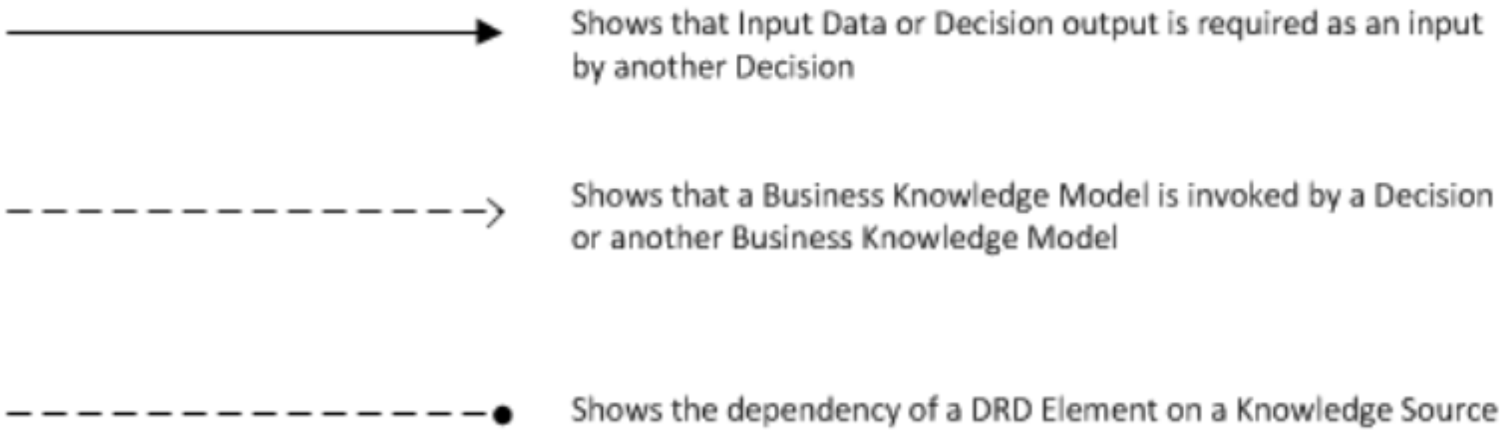


Knowledge source representing predictive analytics

- They may be drawn from **Input Data** and **Decisions to Knowledge Sources**, where, in conjunction with use (a), they represent the derivation of Business Knowledge Models from instances of Input Data and Decision results, using analytics



DRD Requirements

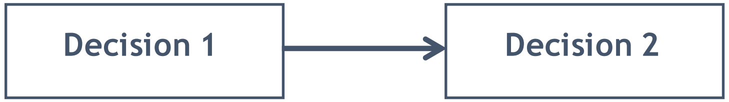


Shows that Input Data or Decision output is required as an input by another Decision

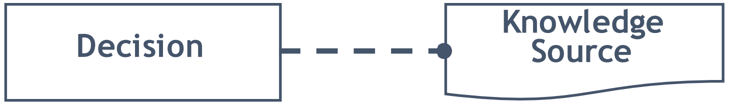
Shows that a Business Knowledge Model is invoked by a Decision or another Business Knowledge Model

Shows the dependency of a DRD Element on a Knowledge Source

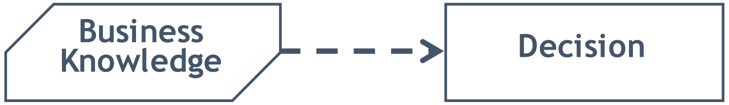
Elements and Allowed Relationships of the Requirements Graph



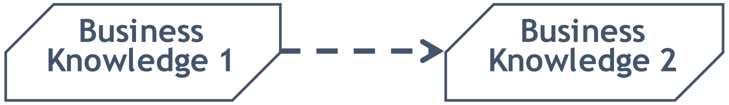
Decision 1 is **used as input** for decision 2



Decision **depends on** Knowledge Source

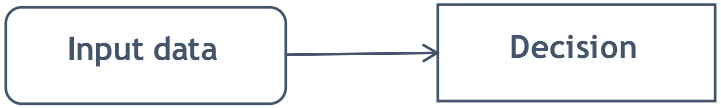


Business Knowledge **invokes** a Decision

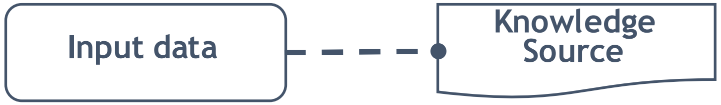


Business Knowledge 1 **invokes** Business Knowledge 2

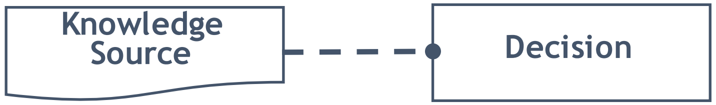
(Coenen 2013)



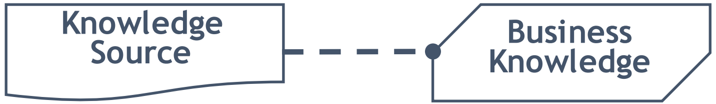
Input data is **used as input** for decision



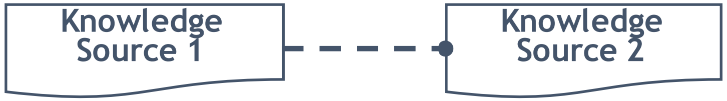
Input data **depends on** Knowledge Source



Knowledge Source **depends on** Decision

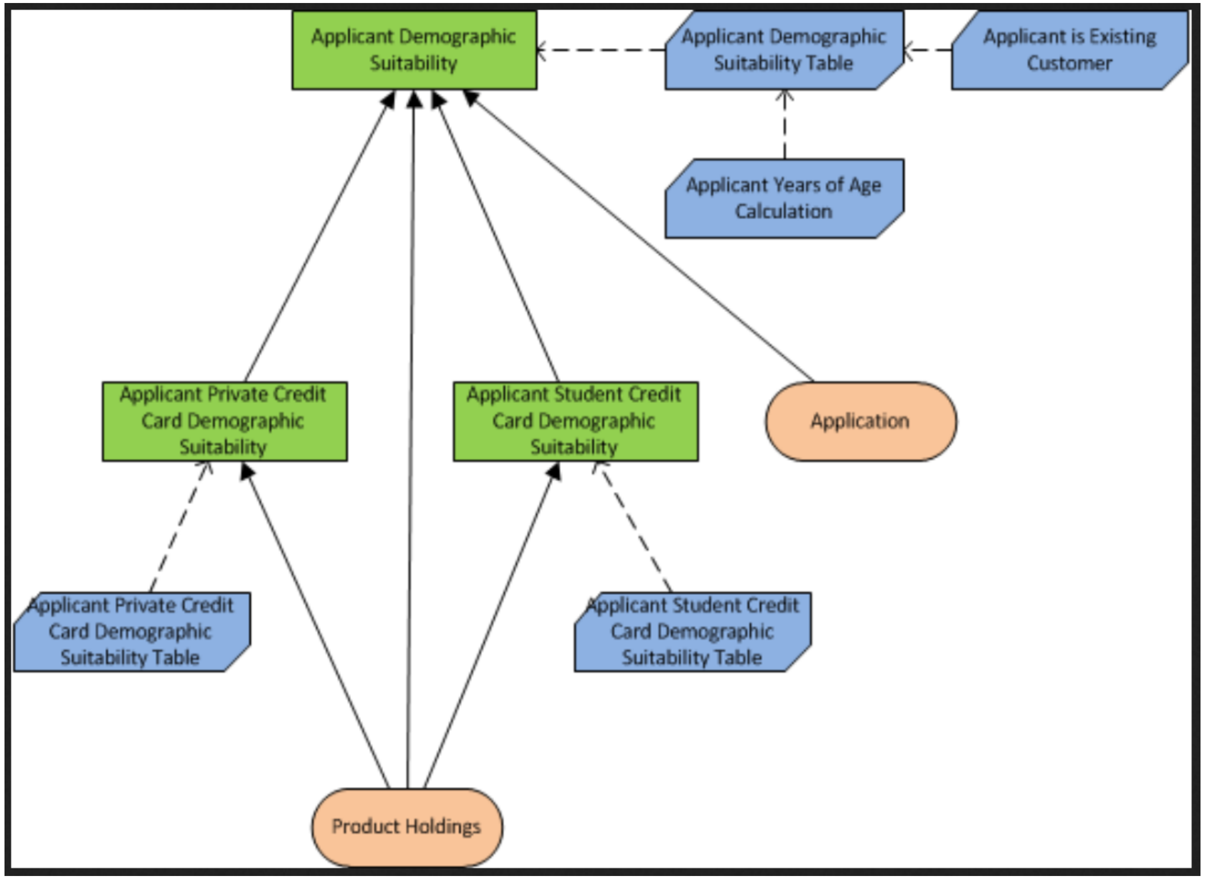


Knowledge Source **depends on** Business Knowledge



Knowledge Source1 **depends on** Knowledge Source2

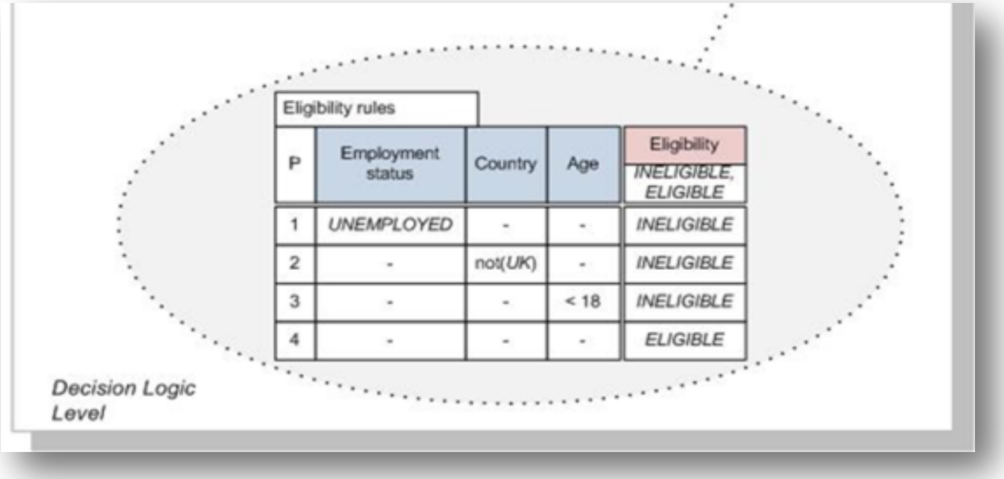
Example


Applicant Demographic Suitability
Applicant Demographic Suitability Table
Applicant is Existing Customer
Applicant Years of Age Calculation
Applicant Private Credit Card Demographic Suitability
Applicant Student Credit Card Demographic Suitability
Application
Applicant Private Credit Card Demographic Suitability Table
Applicant Student Credit Card Demographic Suitability Table
Product Holdings

Main concepts – Decision Logic

- Greater detail
- Business rules
- Calculations
- Automated
- Display

Decision Logic

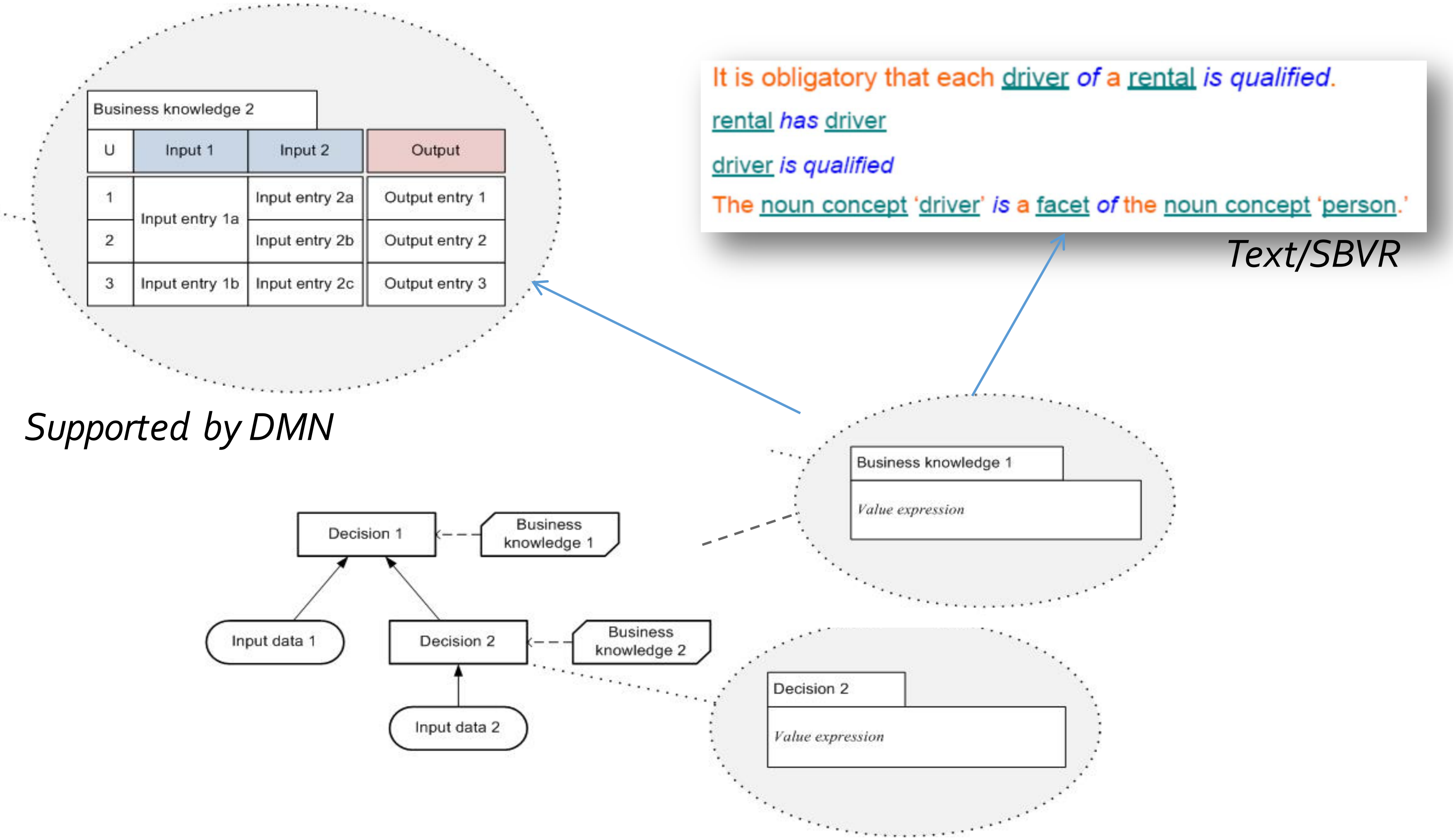


Eligibility rules

P	Employment status	Country	Age	Eligibility INELIGIBLE, ELIGIBLE
1	UNEMPLOYED	-	-	INELIGIBLE
2	-	not(UK)	-	INELIGIBLE
3	-	-	< 18	INELIGIBLE
4	-	-	-	ELIGIBLE

Decision Logic Level

Modeling Decision Logic



Business knowledge 2

U	Input 1	Input 2	Output
1	Input entry 1a	Input entry 2a	Output entry 1
2		Input entry 2b	Output entry 2
3	Input entry 1b	Input entry 2c	Output entry 3

Supported by DMN

It is obligatory that each driver of a rental is qualified.

rental has driver

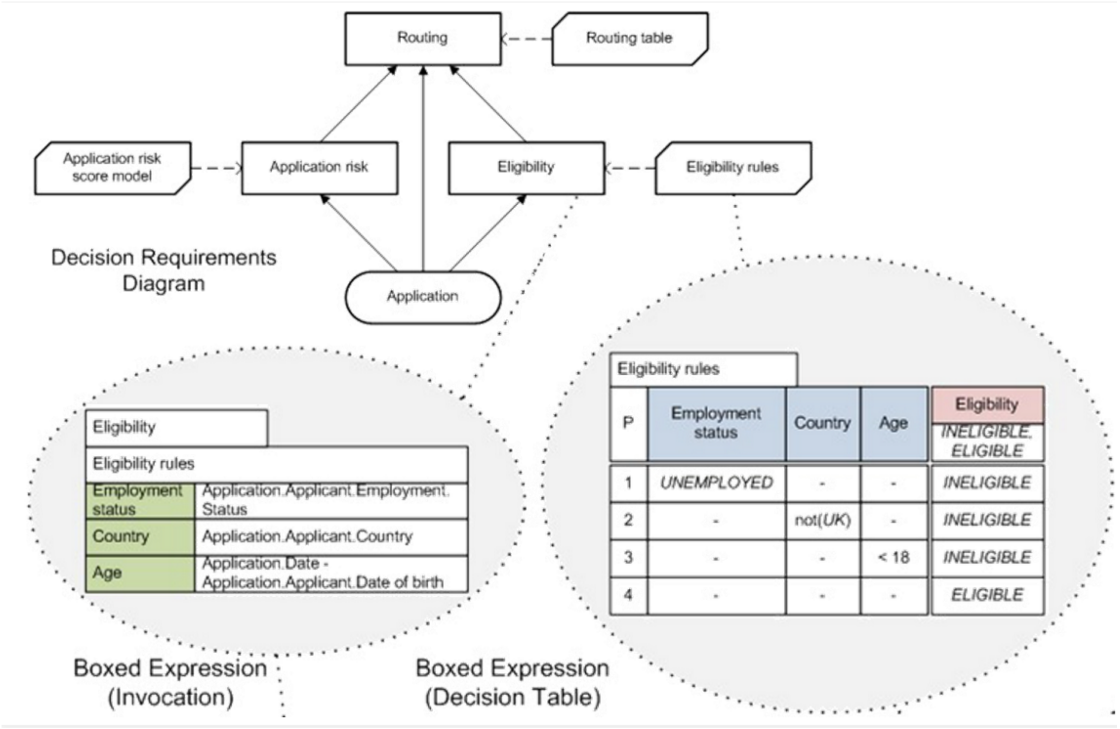
driver is qualified

The noun concept 'driver' is a facet of the noun concept 'person.'

Text/SBVR

Boxed Expression

The notation for decision logic is **boxed expressions** which decompose the decision logic model into small pieces that are associated with elements of Decision Requirements Diagram


Routing
Routing table
Application risk score model
Application risk
Eligibility
Eligibility rules
Decision Requirements Diagram
Application
Boxed Expression (Invocation)
Boxed Expression (Decision Table)

Eligibility

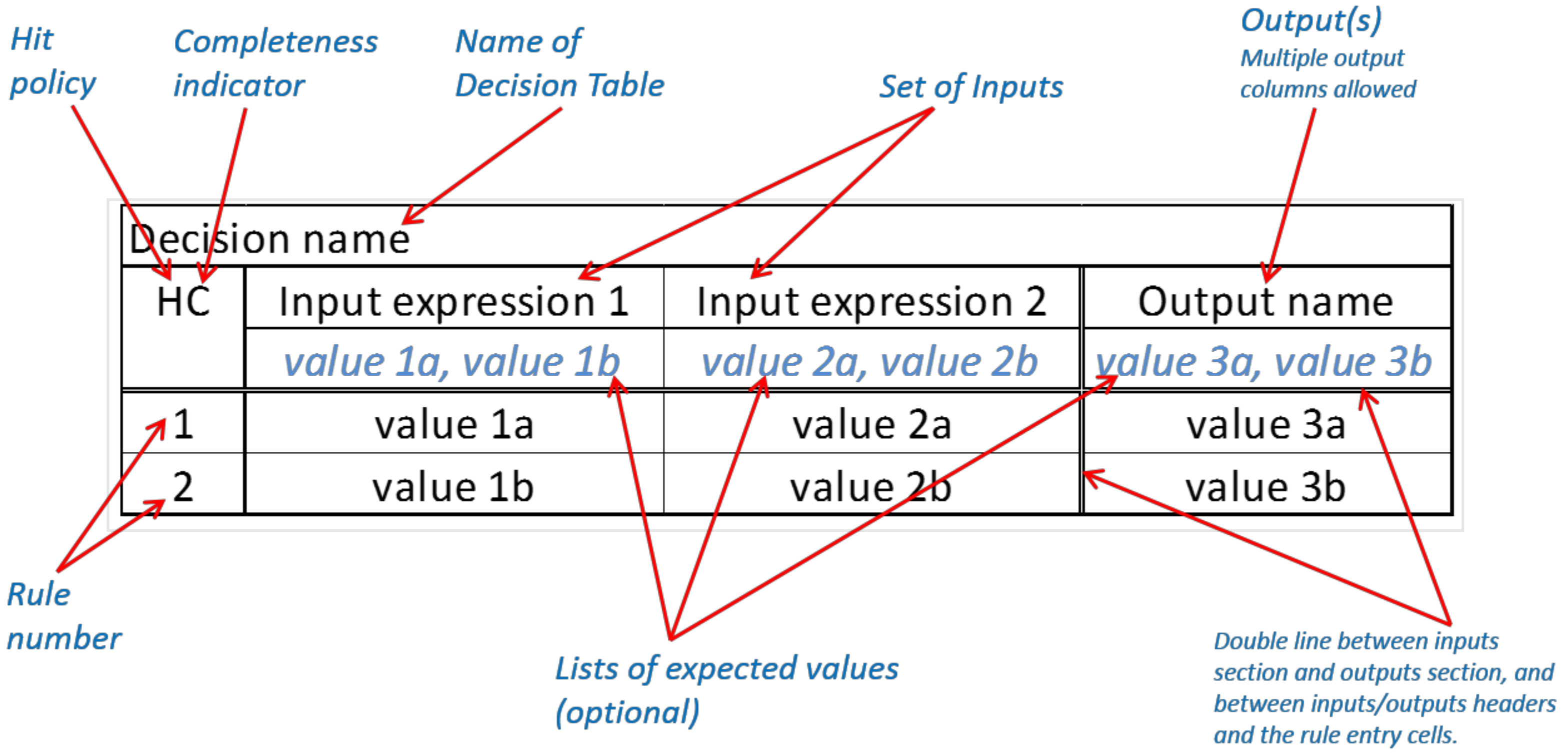
Eligibility rules

Employment status	Application.Applicant.Employment.Status
Country	Application.Applicant.Country
Age	Application.Date - Application.Applicant.Date of birth

Eligibility rules

P	Employment status	Country	Age	Eligibility INELIGIBLE, ELIGIBLE
1	UNEMPLOYED	-	-	INELIGIBLE
2	-	not(UK)	-	INELIGIBLE
3	-	-	< 18	INELIGIBLE
4	-	-	-	ELIGIBLE

Structure of a Decision Table in DMN



Hit policy

Completeness indicator

Name of Decision Table

Set of Inputs

Output(s)
Multiple output columns allowed

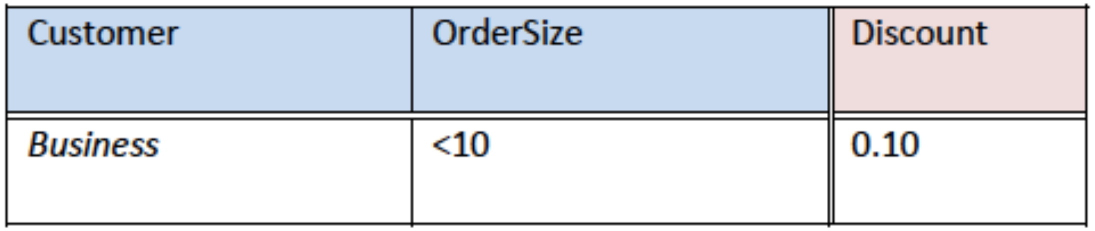
Decision name			
HC	Input expression 1	Input expression 2	Output name
	value 1a, value 1b	*value 2a, value 2b*	*value 3a, value 3b*
1	value 1a	value 2a	value 3a
2	value 1b	value 2b	value 3b

Rule number

Lists of expected values (optional)

Double line between inputs section and outputs section, and between inputs/outputs headers and the rule entry cells.

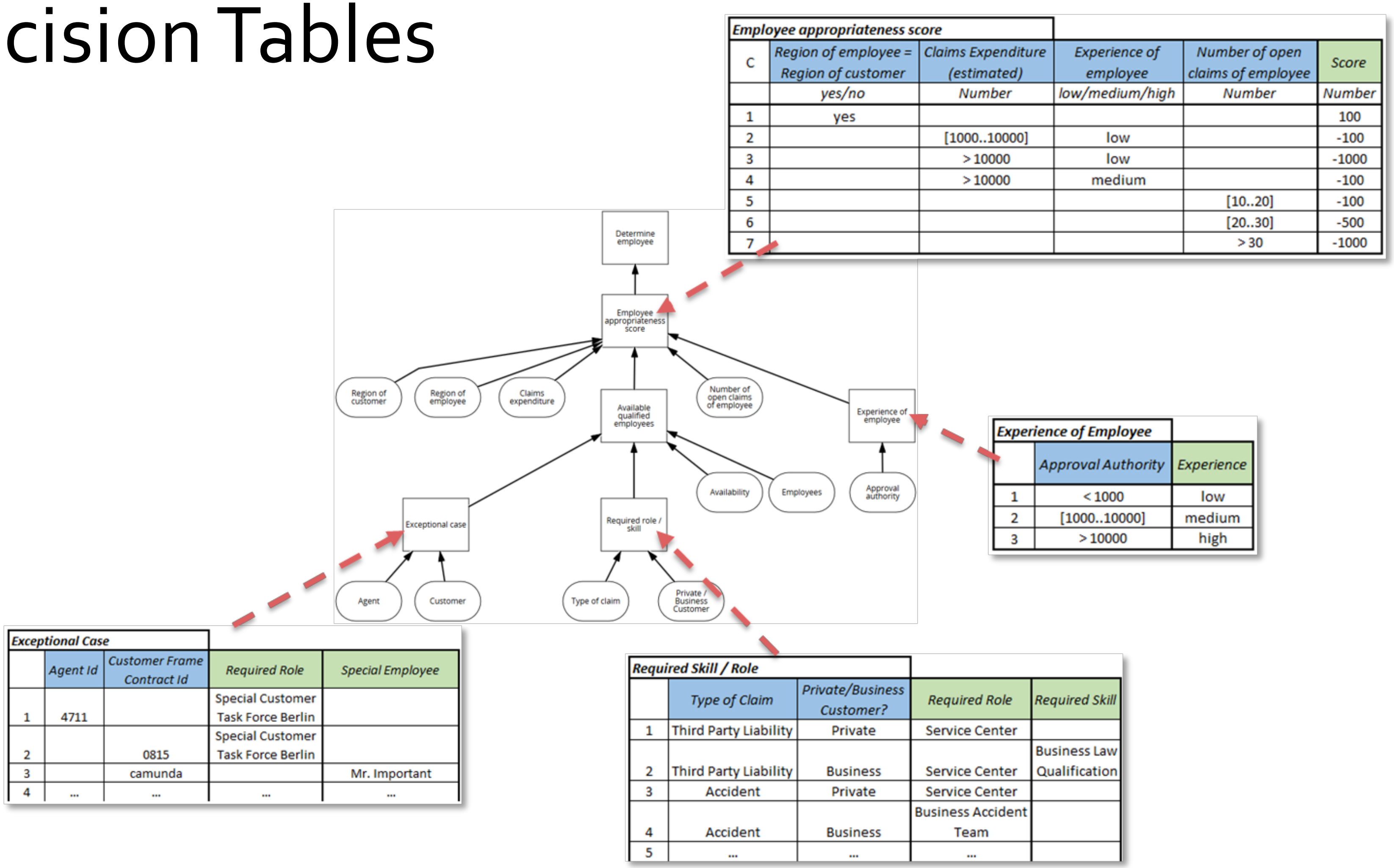
Small Example



Customer	OrderSize	Discount
Business	<10	0.10

If Customer = “Business” **and** OrderSize < 10 **then** Discount = 0.10

Decision Tables



Employee appropriateness score

C	*Region of employee = Region of customer*	*Claims Expenditure (estimated)*	*Experience of employee*	*Number of open claims of employee*	*Score*
	yes/no	*Number*	*low/medium/high*	*Number*	*Number*
1	yes				100
2		[1000..10000]	low		-100
3		> 10000	low		-1000
4		> 10000	medium		-100
5				[10..20]	-100
6				[20..30]	-500
7				> 30	-1000

Experience of Employee

	Approval Authority	*Experience*
1	< 1000	low
2	[1000..10000]	medium
3	> 10000	high

Exceptional Case

	Agent Id	*Customer Frame Contract Id*	*Required Role*	*Special Employee*
1	4711		Special Customer Task Force Berlin	
2		0815	Special Customer Task Force Berlin	
3		camunda		Mr. Important
4	...	...	...	...

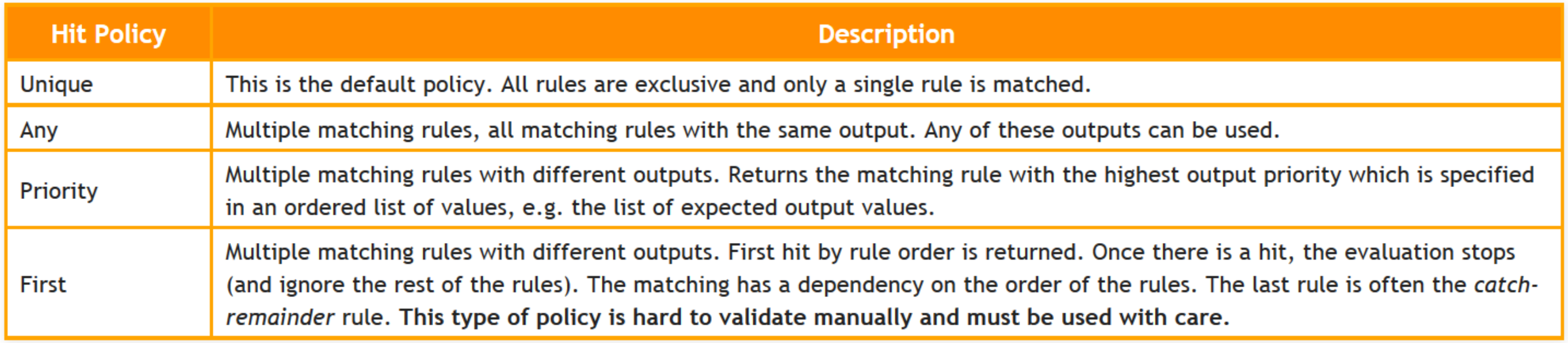
Required Skill / Role

	Type of Claim	*Private/Business Customer?*	*Required Role*	*Required Skill*
1	Third Party Liability	Private	Service Center	
2	Third Party Liability	Business	Service Center	Business Law Qualification
3	Accident	Private	Service Center	
4	Accident	Business	Business Accident Team	
5	...	...	...	

Hit Policies (1)

- The hit policy specifies what the result of the decision table is, if there are multiple matches for a given set of inputs
- The hit policy indication is mandatory and is summarized using a single character in a particular decision table cell

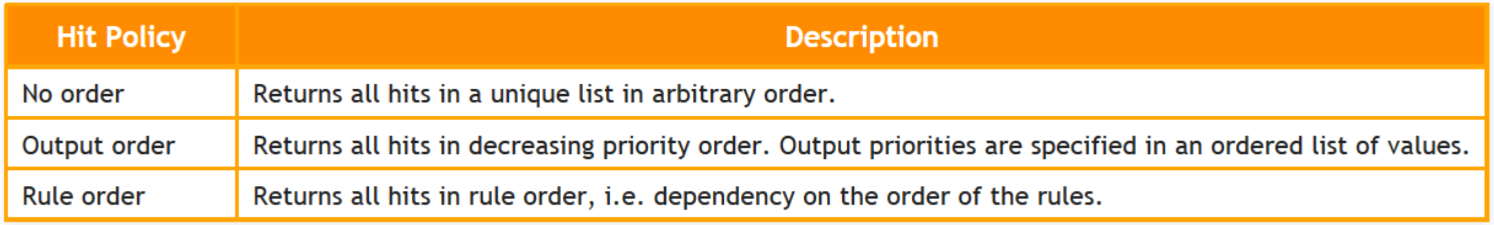
Single Hit Policies:



Hit Policy	Description
Unique	This is the default policy. All rules are exclusive and only a single rule is matched.
Any	Multiple matching rules, all matching rules with the same output. Any of these outputs can be used.
Priority	Multiple matching rules with different outputs. Returns the matching rule with the highest output priority which is specified in an ordered list of values, e.g. the list of expected output values.
First	Multiple matching rules with different outputs. First hit by rule order is returned. Once there is a hit, the evaluation stops (and ignore the rest of the rules). The matching has a dependency on the order of the rules. The last rule is often the *catch-remainder* rule. **This type of policy is hard to validate manually and must be used with care.**

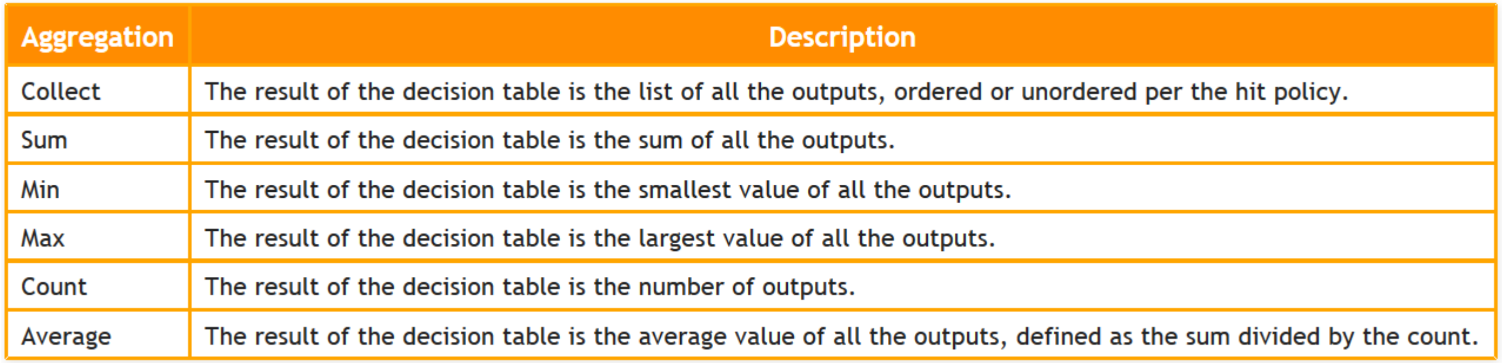
Hit Policies (2)

Multiple Hits Policies for Single Output



Hit Policy	Description
No order	Returns all hits in a unique list in arbitrary order.
Output order	Returns all hits in decreasing priority order. Output priorities are specified in an ordered list of values.
Rule order	Returns all hits in rule order, i.e. dependency on the order of the rules.

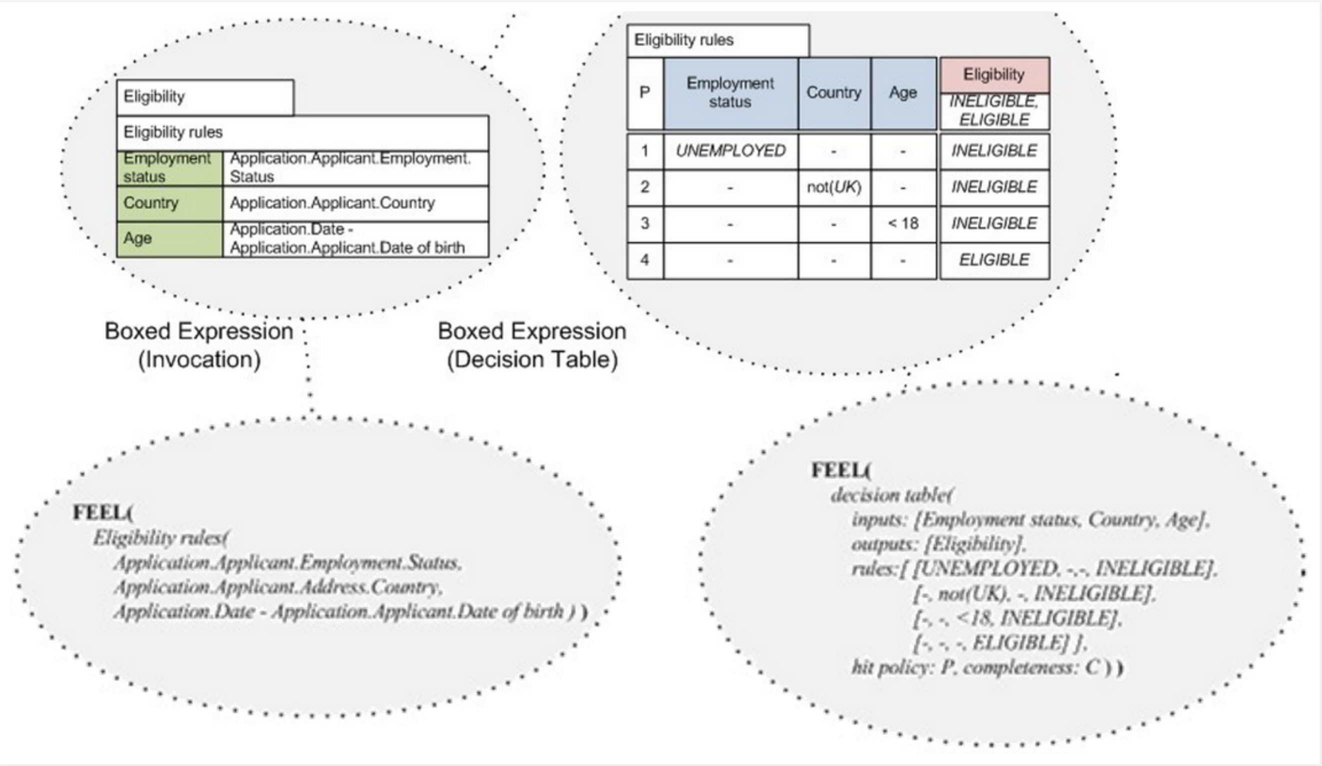
Aggregation for Multiple Hits Policy



Aggregation	Description
Collect	The result of the decision table is the list of all the outputs, ordered or unordered per the hit policy.
Sum	The result of the decision table is the sum of all the outputs.
Min	The result of the decision table is the smallest value of all the outputs.
Max	The result of the decision table is the largest value of all the outputs.
Count	The result of the decision table is the number of outputs.
Average	The result of the decision table is the average value of all the outputs, defined as the sum divided by the count.

FEEL = Friendly Enough Expression Language

FEEL is a script language for decision tables



Eligibility

Eligibility rules	
Employment status	Application.Applicant.Employment.Status
Country	Application.Applicant.Country
Age	Application.Date - Application.Applicant.Date of birth

Boxed Expression (Invocation)

Eligibility rules

P	Employment status	Country	Age	Eligibility INELIGIBLE, ELIGIBLE
1	UNEMPLOYED	-	-	INELIGIBLE
2	-	not(UK)	-	INELIGIBLE
3	-	-	< 18	INELIGIBLE
4	-	-	-	ELIGIBLE

Boxed Expression (Decision Table)

```
FEEL(
  Eligibility rules(
    Application.Applicant.Employment.Status,
    Application.Applicant.Address.Country,
    Application.Date - Application.Applicant.Date of birth ) )
```

```
FEEL(
  decision table(
    inputs: [Employment status, Country, Age],
    outputs: [Eligibility],
    rules:[ [UNEMPLOYED, -,-, INELIGIBLE],
            [-, not(UK), -, INELIGIBLE],
            [-, -, <18, INELIGIBLE],
            [-, -, -, ELIGIBLE] ],
    hit policy: P, completeness: C ) )
```

Orientation of Rules in a DMN Decision Table

Rules as Rows:

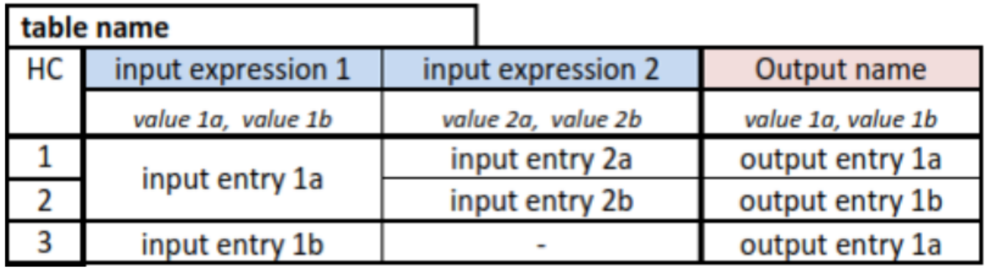


table name

HC	input expression 1	input expression 2	Output name
	value 1a, value 1b	*value 2a, value 2b*	*value 1a, value 1b*
1	input entry 1a	input entry 2a	output entry 1a
2		input entry 2b	output entry 1b
3	input entry 1b	-	output entry 1a

Rules as Columns:

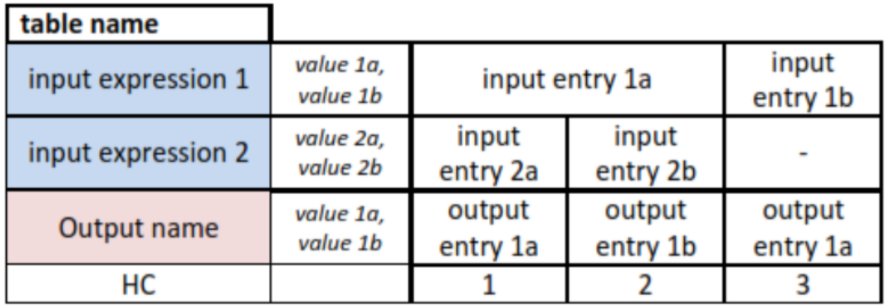


table name

input expression 1	*value 1a, value 1b*	input entry 1a		input entry 1b
input expression 2	*value 2a, value 2b*	input entry 2a	input entry 2b	-
Output name	*value 1a, value 1b*	output entry 1a	output entry 1b	output entry 1a
HC		1	2	3

Rules as Crosstabs:

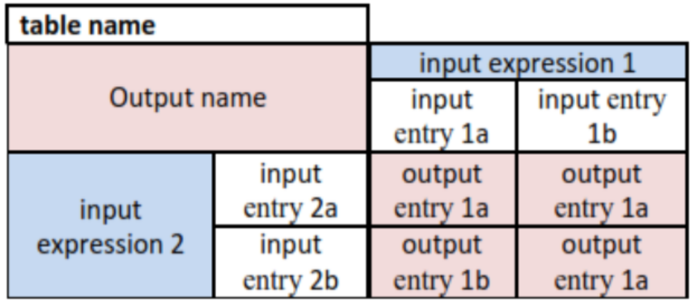


table name

Output name		input expression 1	
		input entry 1a	input entry 1b
input expression 2	input entry 2a	output entry 1a	output entry 1a
	input entry 2b	output entry 1b	output entry 1a

Exercise

When a claim notification has arrived, the claims manager checks if the claim is covered by the insurance policy. During this task the claims manager can ask for an expert's opinion. If the claim is covered, the accounting department pays the claim. If not, the claims manager rejects the claim. The insurance covers damages over CHF 500 and theft. If the claim is about a damage, the payment depends on the amount of the damage. Only if the amount is higher than CHF 500, the claim is covered. Claims about theft are covered independent of the amount.

M	claim (YES - NO)	Type claim	Amount	
1	YES			pays the claim
2	NO			rejects the claim
3		theft		independent of the amount
4		damage	>500	covered
5		damage	< 500	Not covered