Decision Management

University of Camerino

Lorenzo Rossi

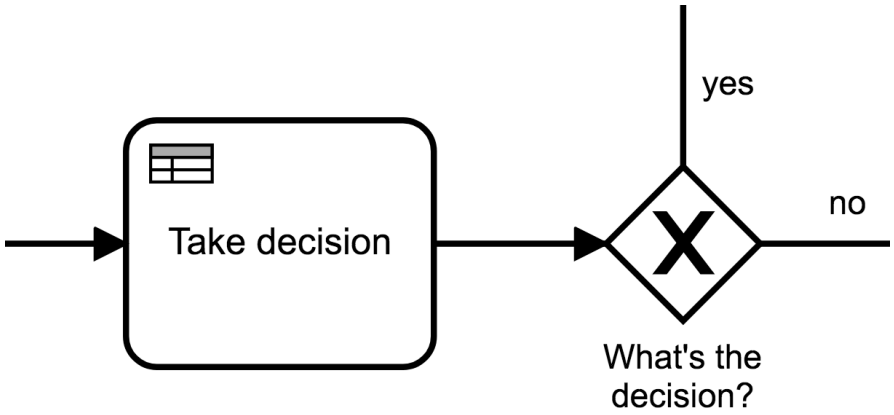
May 22, 2018

Outline

Decisions in BPs

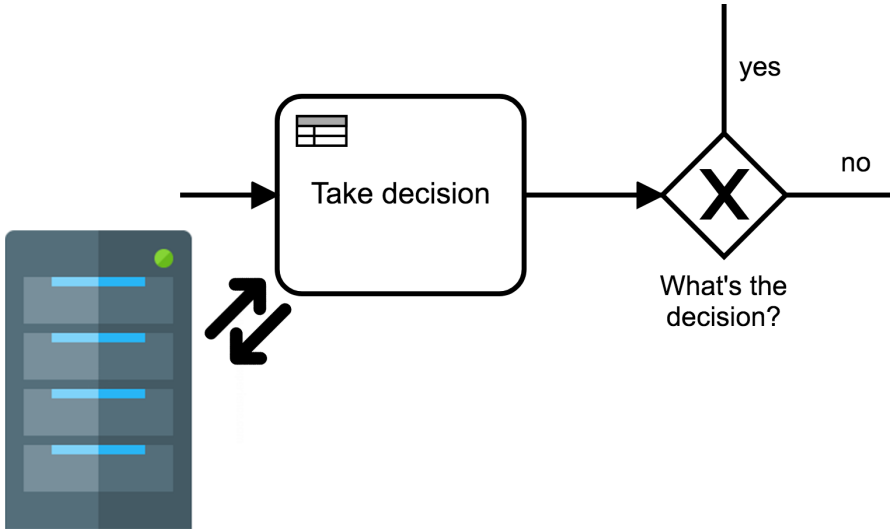
Many business processes contain decisions

- Decisions might impact sequence flow (XOR gateway)
- Decisions assign values (e.g., discount for specific customers)



Decisions in BPs

Decisions come from: business rules, human knowledge, policies ...



In BPs management systems, decisions are implemented in software involving issues such as:

- How is the decision taken?
- Which data is being used?
- Which policies are taken into account?
- Who is responsible for the decision?

Decision Management

Company's value chain is directly affected by how well it designs decision making. **Decision modeling is complementary to process modeling**

Decision management aim to model well-defined, operational decisions, thereby improving transparency, maintainability, and supporting change and automation of decisions

DMN

DMN is the OMG standard notation (Sept 2015) that is understandable by who will manage and monitor decisions.

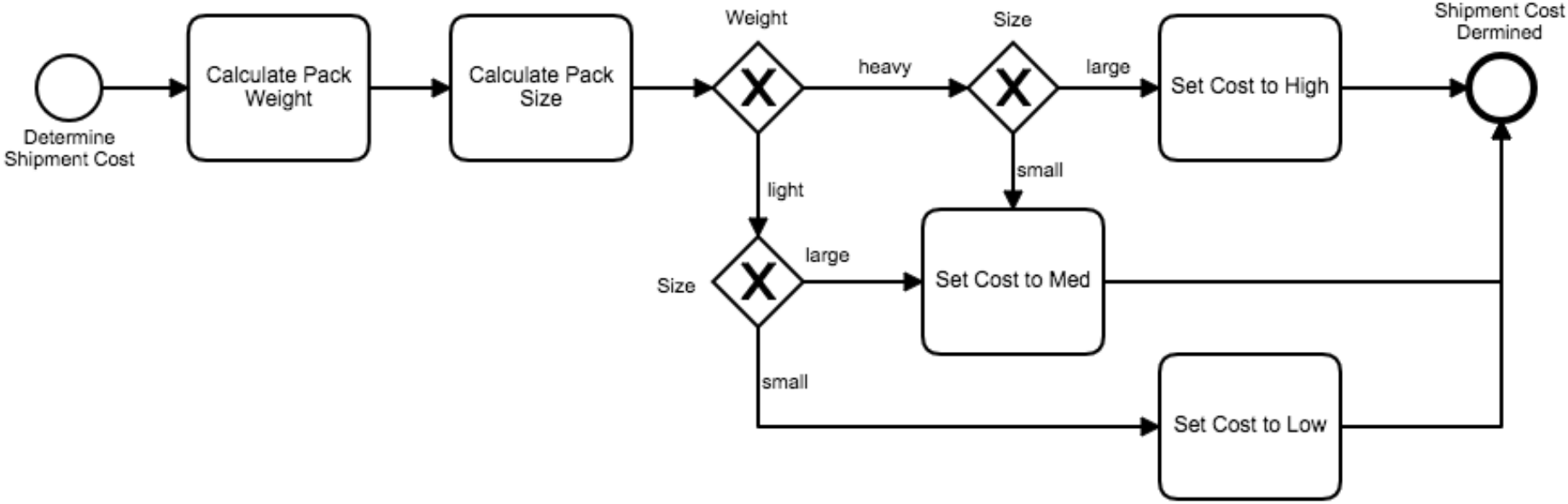
DMN is designed to be useable alongside the standard BPMN business process notation.
BPMN can describe the coordination of decision-making within business processes by defining specific tasks or activities within which the decision-making takes place

DMN

DMN will provide a third perspective forming a **bridge between business process models and decision logic models**:

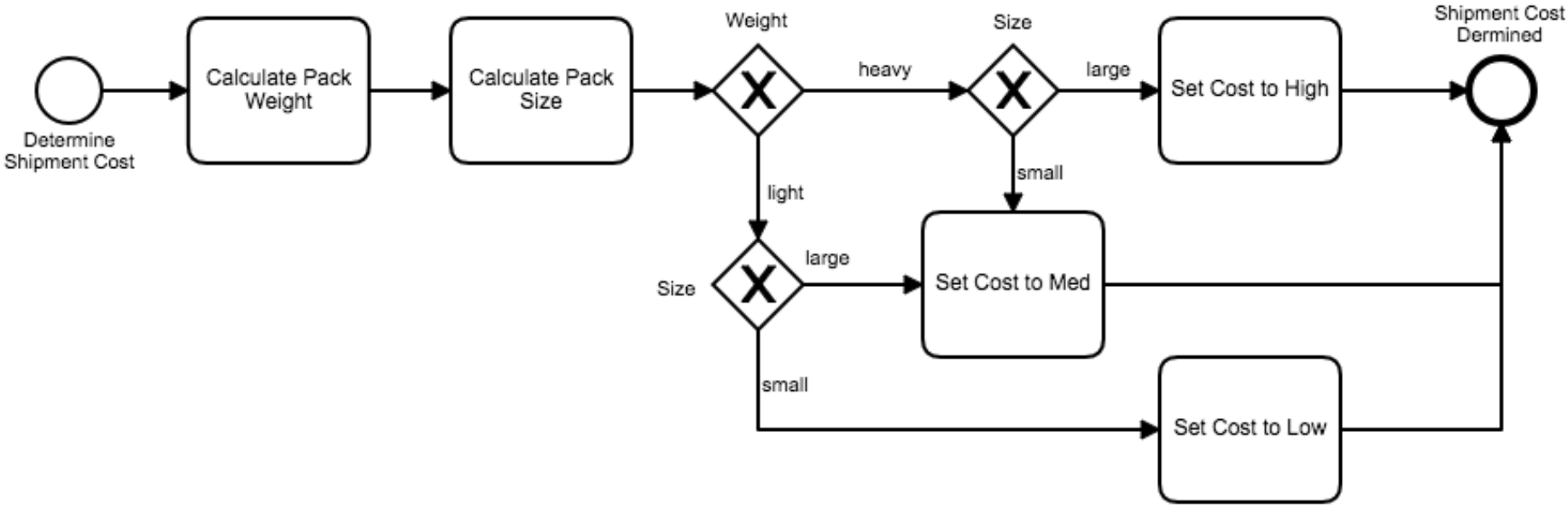
- **Business process model**s will define tasks within business processes where decision-making is required to occur
- **Decision Requirement**s Diagrams will define the decisions to be made in those tasks, their interrelationships, and their requirements for decision logic
- **Decision logic** will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

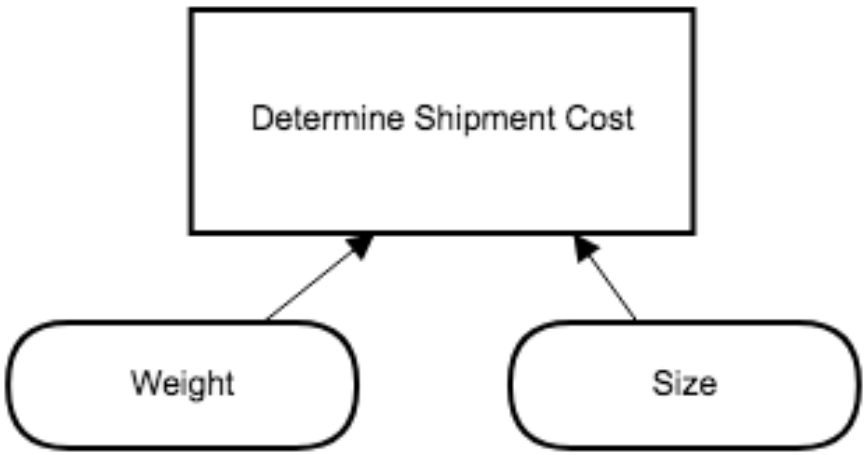
Shipment Cost Process



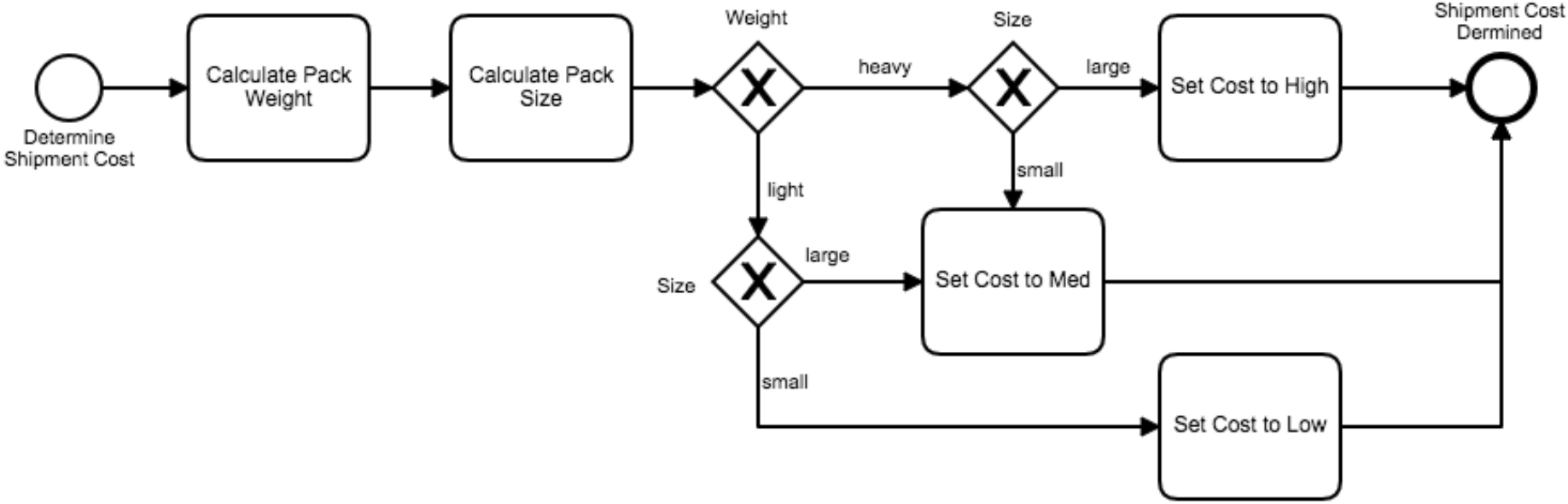
Weight	Size	Cost
light	small	low
light	large	med
heavy	small	med
heavy	large	high

Shipment Cost Process





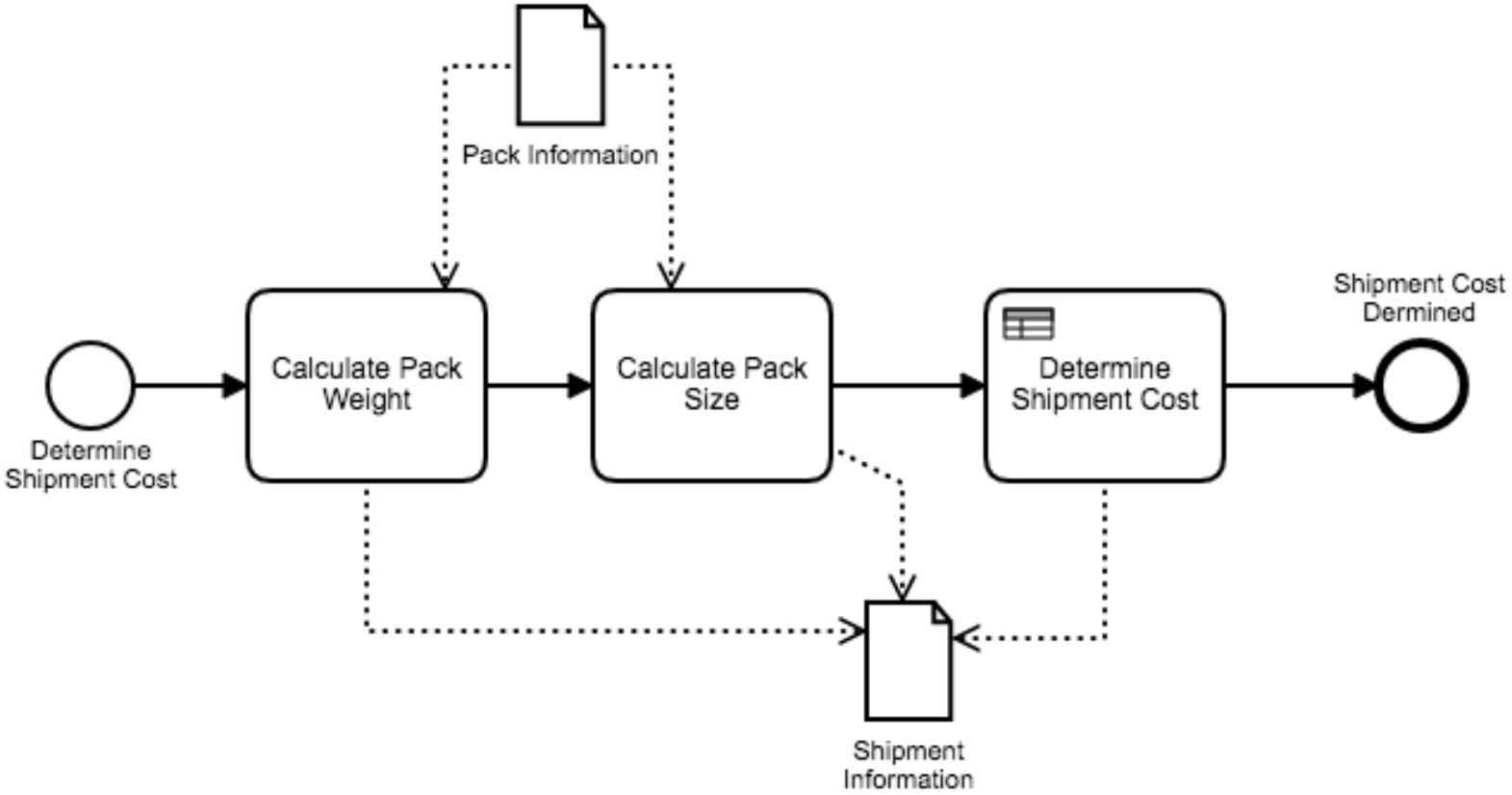
Shipment Cost Process

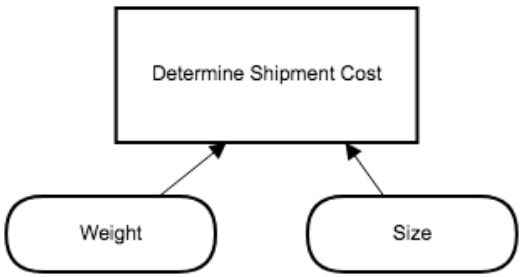


Determine Shipment Cost

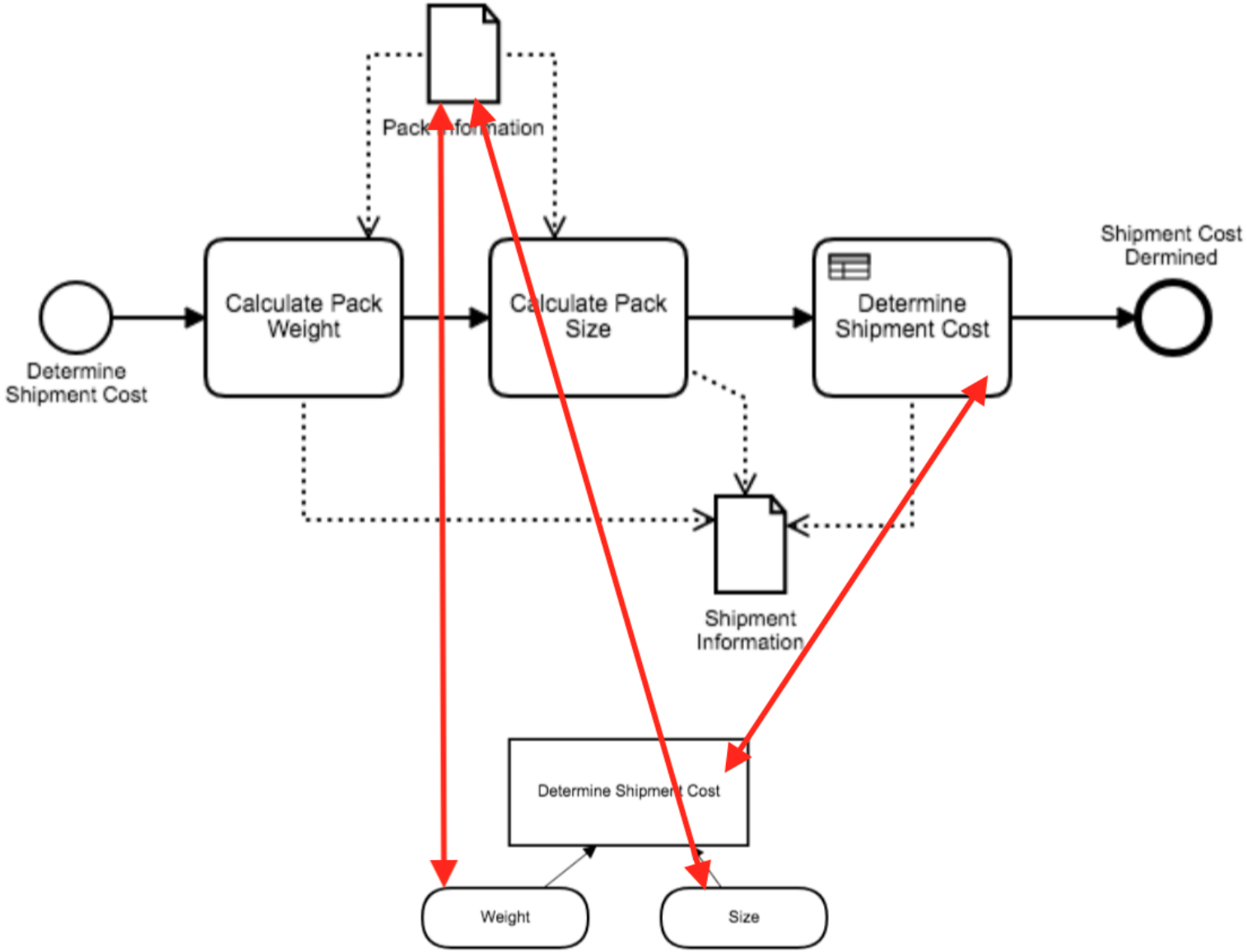
U	Input +		Output +
	Weight	Size	Cost
	string	string	string
1	"light"	"small"	"low"
2	"light"	"large"	"med"
3	"heavy"	"small"	"med"
4	"heavy"	"large"	"high"

BPMN and DMN Integration



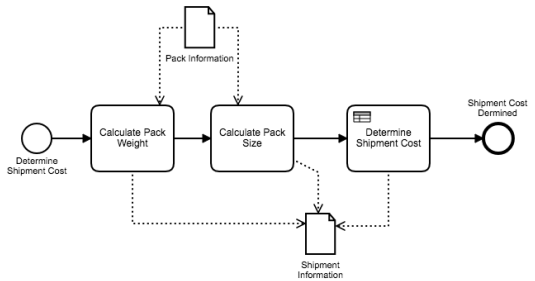


BPMN and DMN Integration



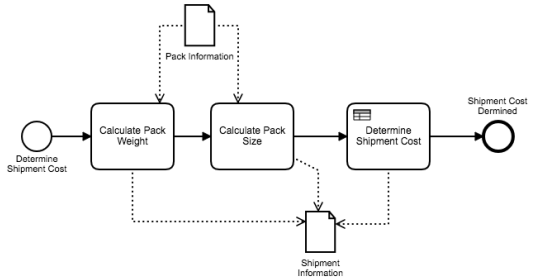
BPMN and DMN Integration

- **Process model**

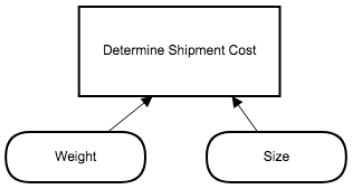


BPMN and DMN Integration

- **Process model**

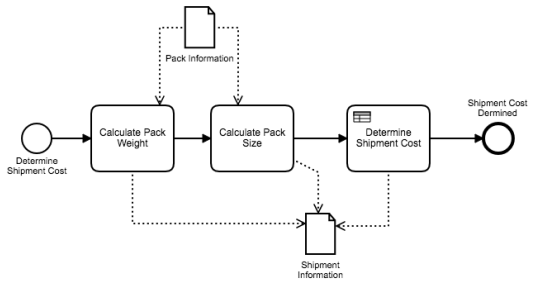


- **Decision requirements**

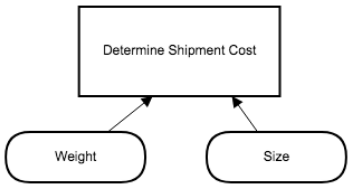


BPMN and DMN Integration

- **Process model**



- **Decision requirements**

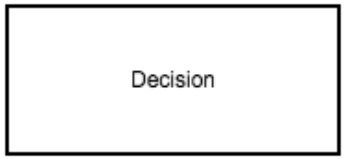


- **Decision logic**

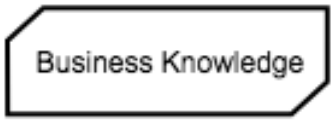
Determine Shipment Cost

U	Input +		Output +
	Weight	Size	Cost
	string	string	string
1	"light"	"small"	"low"
2	"light"	"large"	"med"
3	"heavy"	"small"	"med"
4	"heavy"	"large"	"high"

DMN Elements

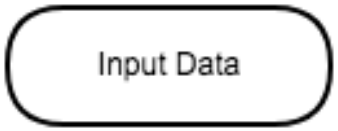


- Determines from a set of input values (typically one) output value
- Is assigned a decision logic, but can also invoke a business knowledge



- Contains knowledge on how a decision should be taken
- Can be re-used by different decisions
- Is invoked by decisions (e.g.; as an oracle)
  - Input data is transferred to the business knowledge
  - Business knowledge provides output to the decision

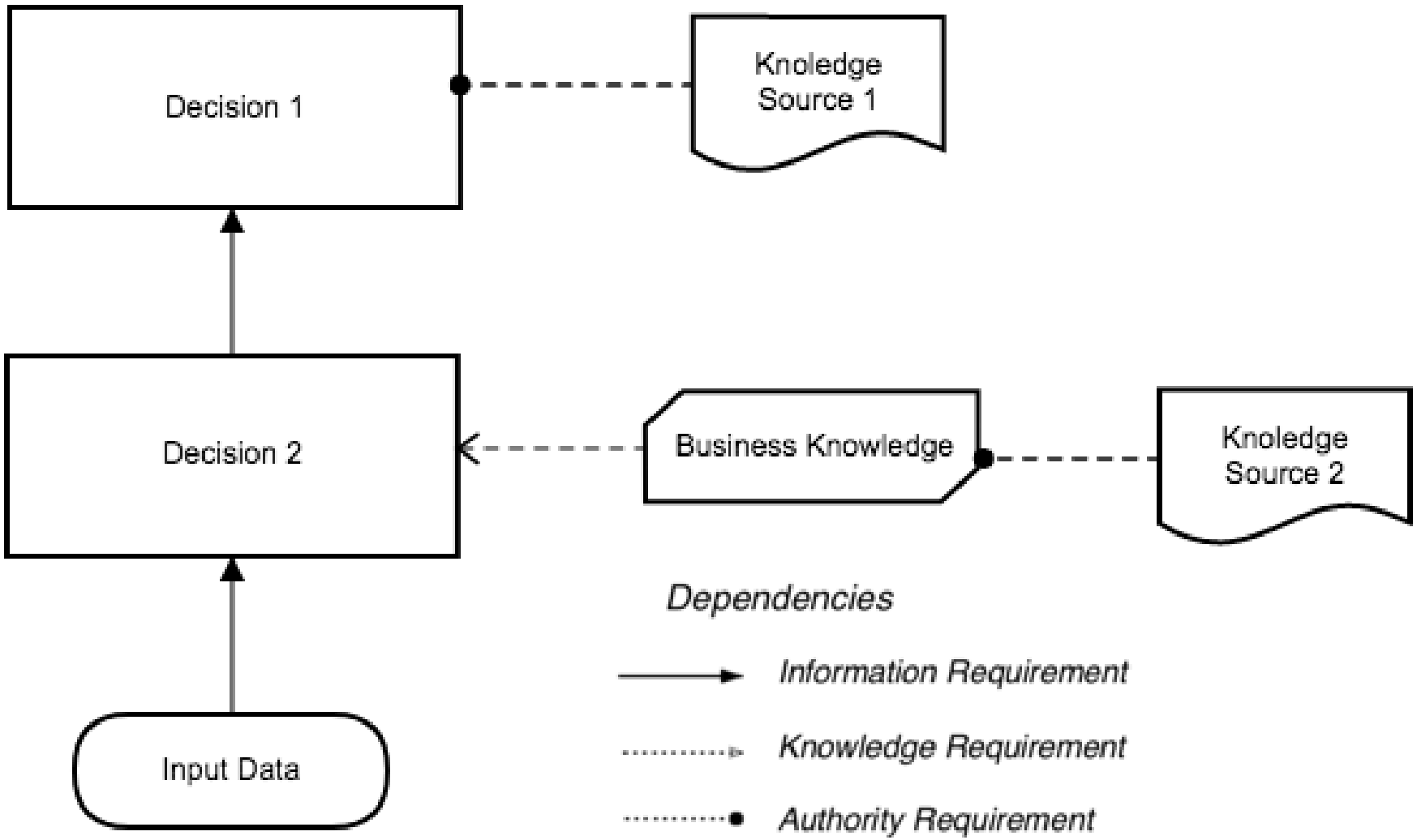
DMN Elements



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

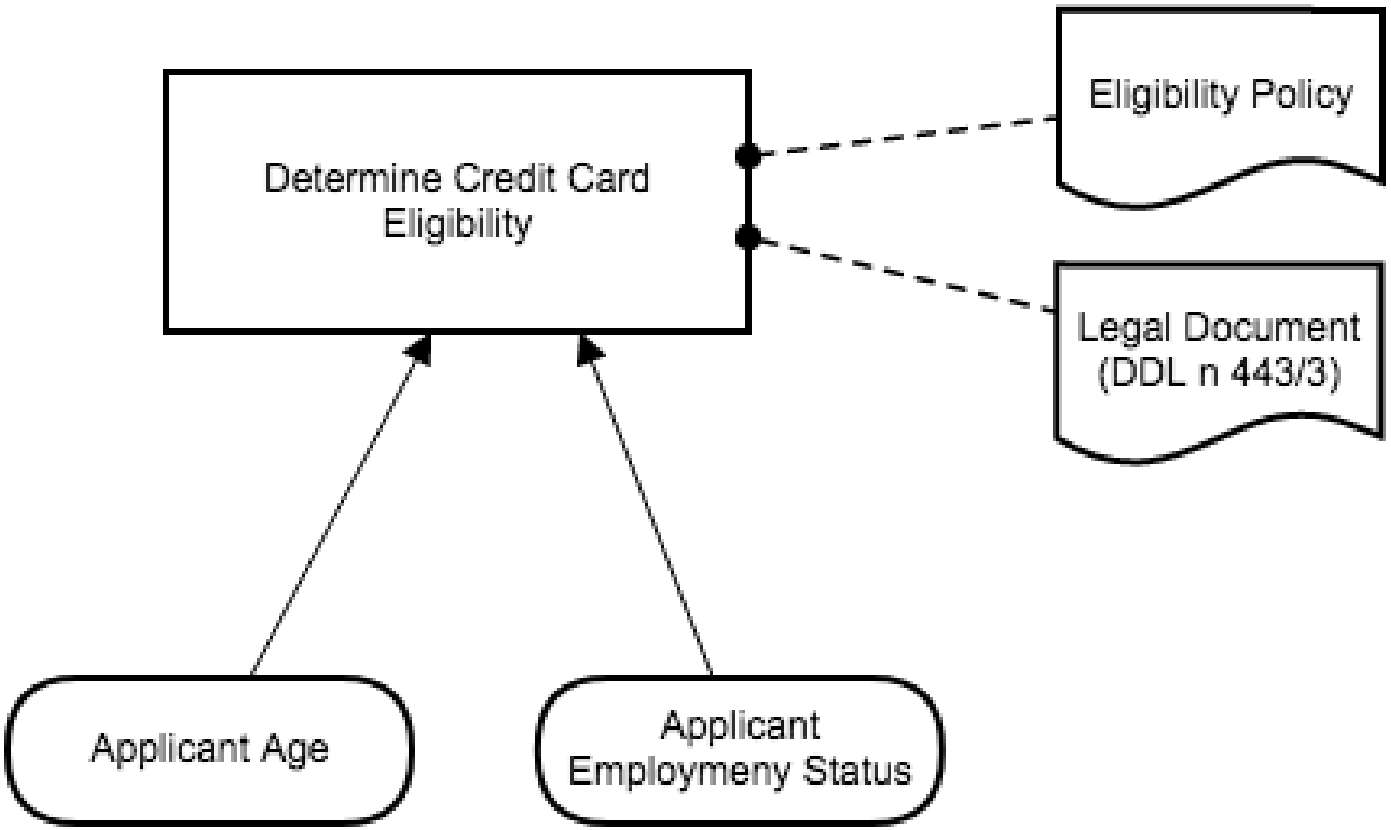
- Represents reasons for decisions, to document decisions
  - Legal regulations
  - Business policies
  - Human experts

Decision Requirements Graph



Credit Card Example

Credit card eligibility is determined using: (i) applicant age and employment status; and (ii) eligibility policy of the institution and a legal document



Decision Logic

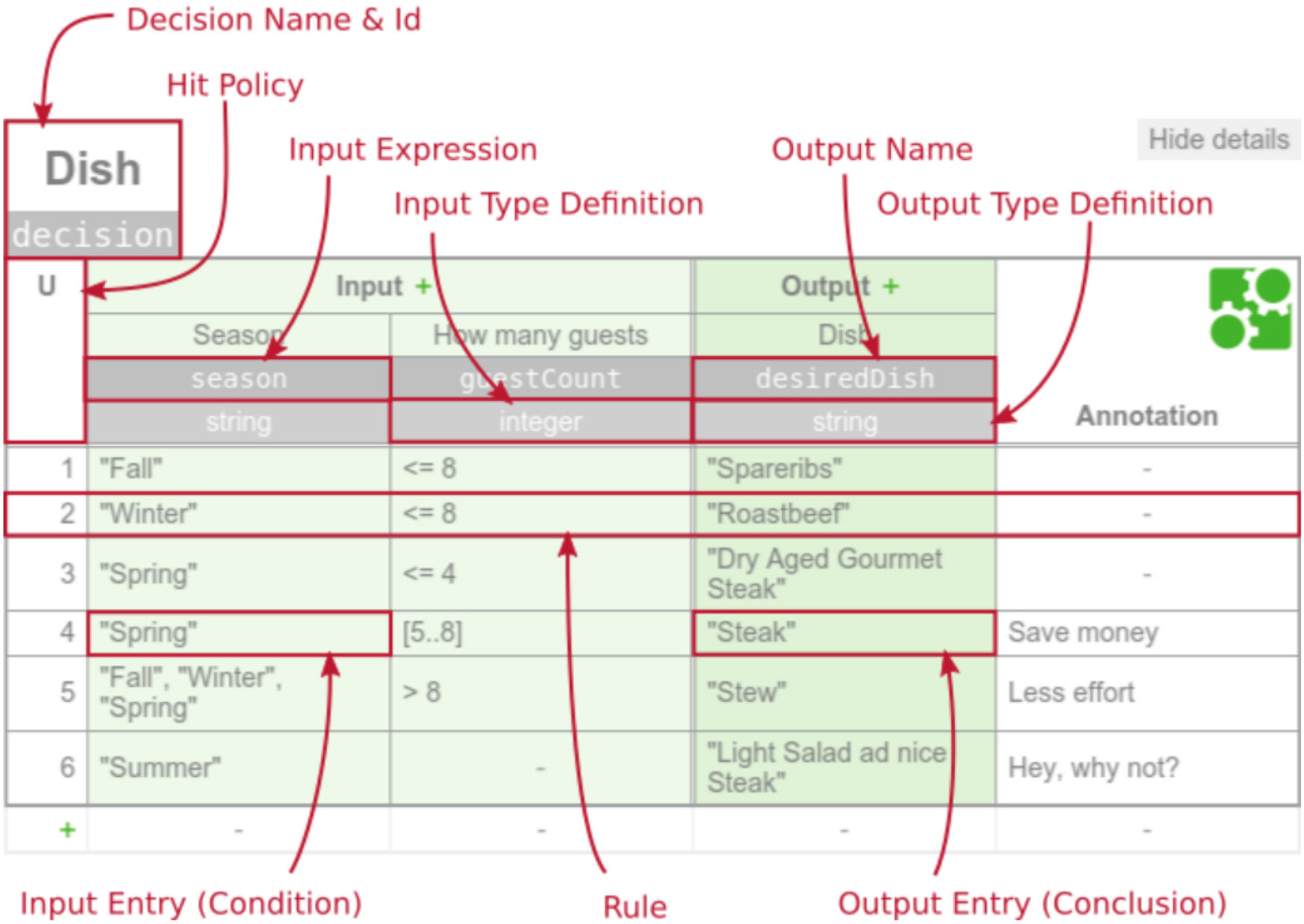
Describes how decision output is derived from its input through value expressions, which might be formal and can be defined in DMN in different ways

- 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

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 Tables



Hit Policy

Specifies how apply rules to get the output

- Unique (U) - For any given input combination, exactly one rule matches
- Any (A) - For a given input, several rules can match. All rules that match return the same output
- Priority (P) - For a given input, several rules can match. Output with highest priority is returned
- First (F) - For a given input, several rules can match. Output of the first satisfied rule is returned
- Output order (O) - Returns all hits in decreasing output priority order
- Rule Order (R) - Outputs of all matching rules are returned as a sorted list, in the order of the rules
- Collect (C) - Outputs of all matching rules are returned as unsorted list

In O,R,C aggregation can be applied (sum $+$, min $<$, max $>$, count $\#$)

Rules in FEEL

DMN defines a Friendly Enough Expression Language (FEEL). It can be used to evaluate expressions in a decision table.

Data Type:

- Numbers
- Strings
- Boolean
- Date and Time

FEEL

The expression can contain different elements:

- Comparison ($<$, $>$, $=$, $>=$,$<=$)
- Range ([1..10],]1..10], (1..10], ...)
- Disjunction ($"3, 5, 7"$, $" < 2, > 10"$, $"0, [20..30]"$, ...)
- Negation ($not("Steak")$, $not(> 10)$, $not(3, 5, 7)$, ...)
- Qualified Names ($x > 1$, $"y = "blue"$, ...)
- Date Functions date and time ("2015-11-30T12:00:00" test if the input is the date November 30th, 2015 at 12:00:00 o'clock)

Analysis of Decision Tables

Decision tables are important assets for organizations

Quality of decision-aware processes should be guaranteed:

- **Completeness** Are all possible input combinations actually covered by the decision table?
- **Consistency** Is the decision table in line with its hit policy?

Completeness of Decision Tables

For all combinations of input values, is there a rule that covers it?

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

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

Consistency of Decision Tables

Single hit properties (U,A, P (?)) can be analyzed

Unique property is violated if there is an input combination that matches two rows

	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

Consistency of Decision Tables

Single hit properties (U,A, P (?)) can be analyzed

Any property is violated if a given input combination matches two rows with different outputs

	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

Consistency of Decision Tables

Single hit properties (U,A, P (?)) can be analyzed

Can **P**riority be violated as well?

P	Inputs: Age		Inputs: Medical history		Outputs: 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! Even if several rows match, a single one will be chosen deterministically

Analysis of BPMN + DMN

Soundness is one of most used correctness criterion for workflow systems

Classical soundness guarantees:

- option to complete
- proper completion
- no dead activities

Analysis of BPMN + DMN

Decision-aware processes rely on dedicated soundness notion

Decision-aware soundness guarantees:

- classical soundness
- decision are deadlock free (any output matches a branch condition)
- no dead decision branch (any branch is fired in at least a run)