UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT

BUSINESS RULES & BUSINESS RULES MANAGEMENT
PART 1A: BUSINESS RULES AS QUERY INPUTS
PART 1B: DECLARE

AGENDA

- Business Rules Introduction
- Business Rules Management
  - Part I (today) – Capturing & Authoring
  - Part II (next week) – Organizing & Implementing

MOTIVATION

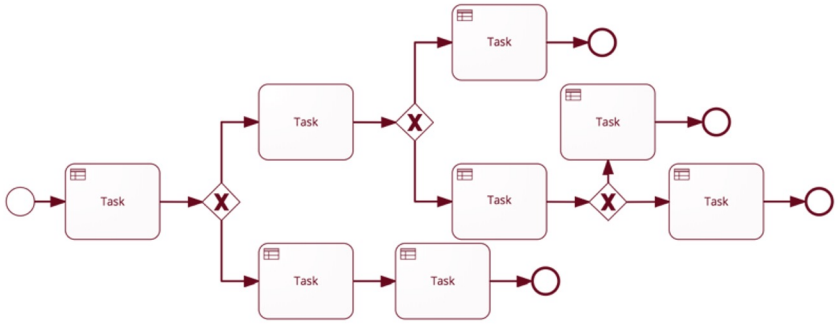
- CTL / GMQL / DMQL are great, but we also need to know what we should search for!

- Main areas of interest in the Information Systems discipline
  - Business Process Improvement
  - Business Process Compliance Management

- In both areas of interest, we want to know in which places our business processes contain problematic structures or behavior

WHAT ARE BUSINESS RULES?

- Model queries normally check so-called business rules
- Business rules depict principles that business processes must comply with
  - concerning their structure
  - concerning the behavior of single instances



BUSINESS RULES DEFINITION

- A business rule is a compact, atomic, well-formed, declarative statement about an aspect of a business, using simple unambiguous language

- General form: IF A_1, A_2, …, A_n THEN B

- With business rules, we can express rules about a domain of interest

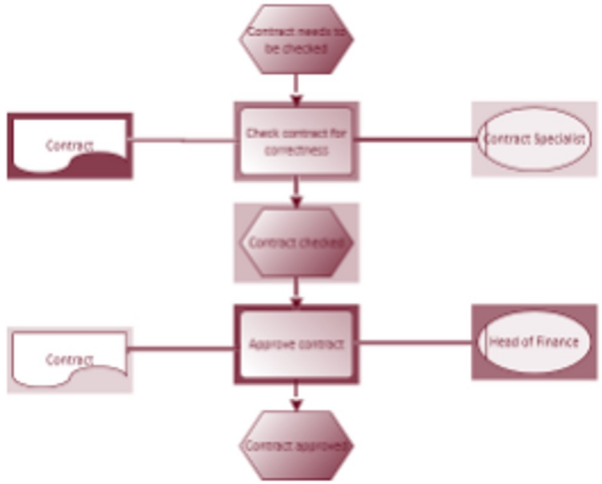
EXAMPLES FOR BUSINESS RULES

- Structure and common behavior-related rules:
  see examples from the last three lectures

- Behavior of single instances:
  - Bob is only credit-worthy, if his credit history score is better than BBB and he is older than 24 years and he has no mental condition problems

WHY DO WE NEED BUSINESS RULES?

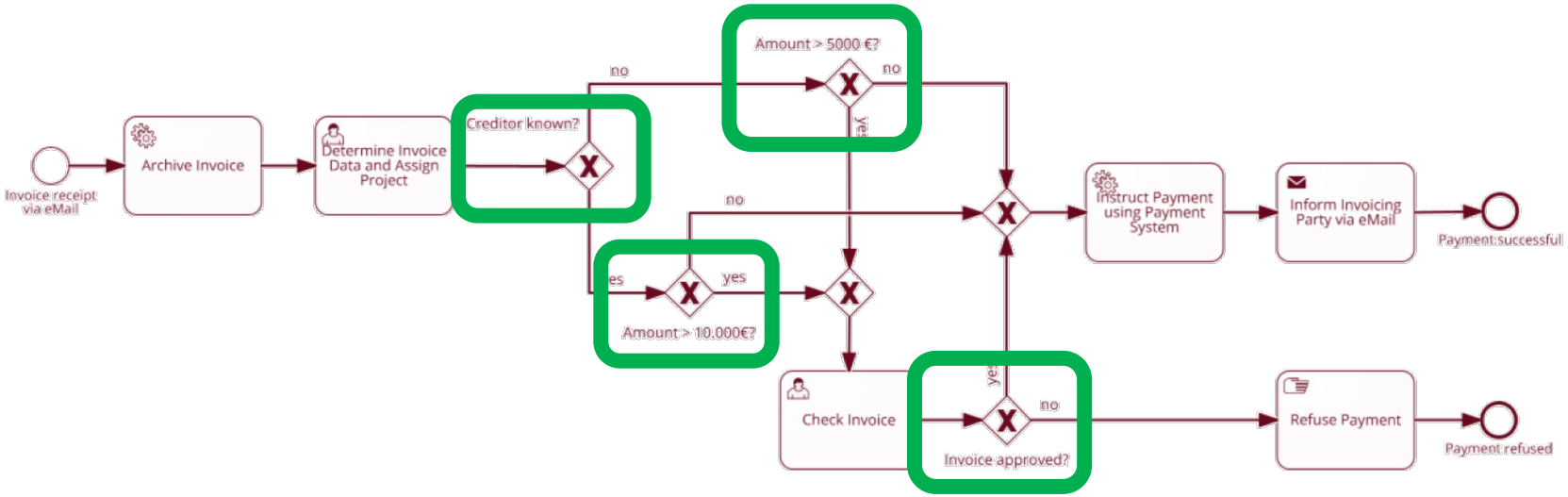
- Ensure the compliance of business process(es) (models), resp. company activities
  - Rules and regulations affect companies and how they are allowed to conduct activities
    - E.g. Sarbanes-Oxley Act
    - Basel III
    - Recent debate on Article 13 of German UrhG



"A loan application must always be approved by the supervisor (4-eye-principle)"

MOTIVATION FOR ABSTRACTION

- Process models may contain multiple rules



- Makes models unclear
- Slows down (automatic) analysis (!)

[© viadee]

BUSINESS RULE MANAGEMENT

- Exemplary Lifecycle Model (Nelson et al. (2009))
- Systematic Methodology to derive and implement a set of business rules

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

BUSINESS RULE MANAGEMENT

Today

Next week

BUSINESS RULE MANAGEMENT

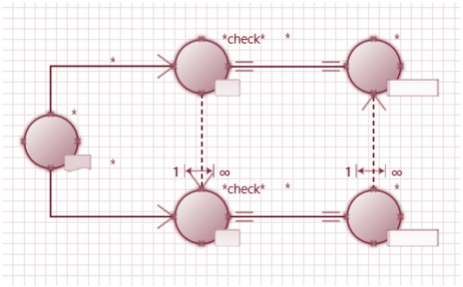
CAPTURE

PROCEDURE

Where does this come from?

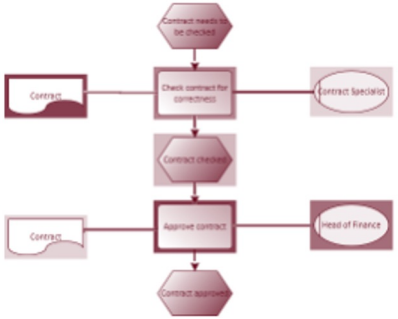
- Description of a shortcoming/violation

Transformation



- Pattern of a model query language

Query



- Query results

SOURCES OF KNOWLEDGE

SHORTCOMINGS AND VIOLATIONS

- Own experiences
- Consultants
- Checklists
- For compliance: compliance rule providers
- Rule mining
- Weakness pattern collections
- Compliance pattern collections

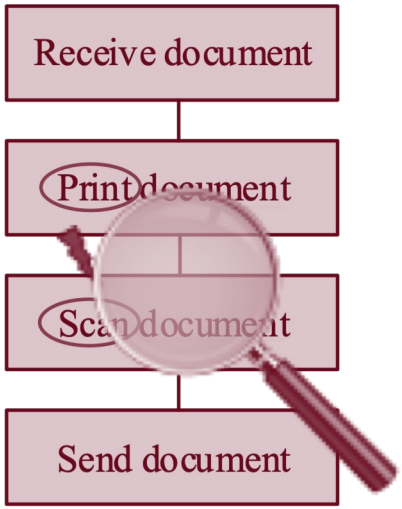
PROBLEMATIC STRUCTURES OR BEHAVIOR

- Business Process Improvement (BPI):
  Eliminate inefficiencies and quality shortcomings and replace corresponding process structures

- Business Process Compliance Management (BPCM):
  Eliminate illegal behavior, violation of corporate guidelines and/or violation of generally accepted rules (e.g., proposed by compliance task forces such as the Sarbanes-Oxley-Act, Basel III, MaRisk etc.)

OVERVIEW OF PROBLEMATIC STRUCTURES AND BEHAVIOR (BPI)

- Overview based on empirical findings
- Manual (!) analysis of over 2000 business process models
- Evaluation of the models by 15 business process management experts
- Double analysis (4-eye-principle)
- Detection of 280 distinct weaknesses
- Consolidation into 111 weakness patterns
- Subdivision in 7 weakness categories

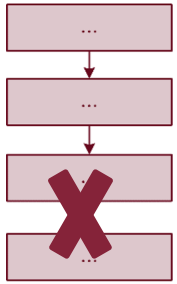


WEAKNESS CATEGORIES I

Modeling Error 23

Incorrect way of modeling

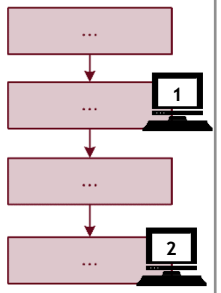
Examples:
Decision without impact
Sent document never received



Information Handling 14

Various ways of information processing

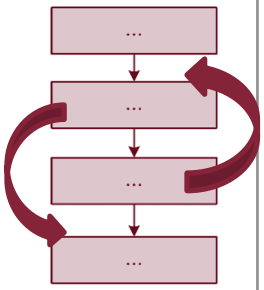
Examples:
Data digitalized twice
Creation without use



Process Flow 39

Inefficient order or missing aggregation

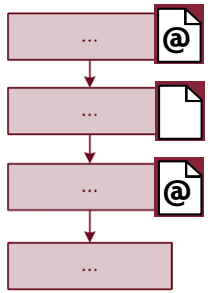
Examples:
Redundant activities
Edit document after copying



Technology Switch 6

Change of used technology and information medium

Examples:
Digitalization after printing
Double archiving

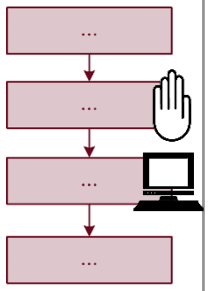


WEAKNESS CATEGORIES II

Automation 20

Unnecessary manual execution of activities

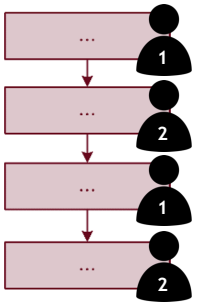
Examples:
Manual calculation
Data transferred manually



Organization 7

Inefficient organizational structure

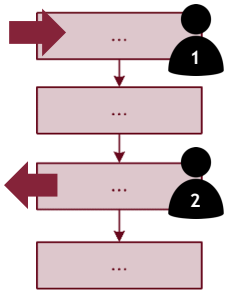
Examples:
Distributed responsibilities
Excessive inform. exchange



Environment 2

Inefficient communication with external partners

Examples:
Multiple negotiations
Varying communication ch.



WEAKNESS CATEGORIES

MODELING ERROR

- Inappropriate use of model elements
  (e.g., a process consists of only one element)
- Control flow errors
  (e.g., decision without impact (“check” → “checked”)
- Unclear interfaces
  (e.g., a business object to be processed has not (yet) entered the process)

WEAKNESS CATEGORIES

PROCESS FLOW

- Process modeled unnecessarily complicated or unclear
  (e.g., too many split/join branches)
- Redundancy (e.g., redundant sub-processes)
- Tasks in inappropriate order
  (e.g., decision before check)
- Inappropriate handling of business objects
  (e.g., receive but never use again)
- Avoidable waiting time (e.g., bad system performance)
- Information (transmission) deficit
  (e.g., decision followed by inquiry)
- Delayed transfer of document slows down process
  (e.g., information could be obtained/sent earlier)
- Missing task composition or decomposition
  (e.g., unnecessary batch processing and/or periodic activities)

WEAKNESS CATEGORIES

INFORMATION HANDLING

- Inefficient document forwarding
  (e.g., receive – archive & do nothing else – send)
- Multiple information existence
  (e.g., digitize information twice)
- Inappropriate communication channel
  (e.g., send information via different channels)

WEAKNESS CATEGORIES

TECHNOLOGY SWITCH

- Archiving
  (e.g., IT and paper based archiving)
- Printer usage
  (e.g., first print, then scan)
- Software usage
  (e.g., use different systems for the same information)
- Editing
  (e.g., enter into IT but edit the paper document)

WEAKNESS CATEGORIES

AUTOMATION

- Inappropriate exchange of information
  (e.g., exchange digital information via fax or phone)
- Manual execution of tasks
  (e.g., check manually for formal correctness)
- Inappropriate usage of software applications (e.g., redundant activities using multiple software applications)
- Information stored inappropriately (e.g., both scan and enter document into a software application”)

WEAKNESS CATEGORIES

ORGANIZATION

- Several faces to the customer (e.g., several organizational units make agreements with the same customer)
- Distributed responsibilities
  (e.g., ping-pong responsibilities)
- Unclear responsibility
  (e.g., multiple or missing responsibilities)
- Inappropriate capacity allocation (e.g., bottleneck)
- Inappropriate capability allocation
  (e.g., resource too specialized or generalized)
- Excessive information exchange (e.g., ping-pong inquiries)

OVERVIEW OF PROBLEMATIC STRUCTURES AND BEHAVIOR (BPCM)

- At present, no exhaustive list
- Still subject of research (formalized and categorized compliance pattern catalogues for different industries)
- Some examples from information-intensive industries:
  - 4-eye-principle
  - Required provision of consultation documents
  - Required provision of regulation documents
  - Required gathering of customer information
  - Required authorization levels of approving persons

RULE MINING

- Applying data mining approaches to extract/abstract business rules
  - Extracting decision logic from source code
  - Extracting business rules from text (natural language)

BUSINESS RULE MANAGEMENT

CAPTURE

AUTHORING

- Now that we have an idea of what our business rules are, we need a representation formalism
  - Rules need to be stored in a machine accessible way to facilitate automated analysis
- Exemplary rule formalisms you know: CTL, GMQL
  - Problem

AUTHORING

- High diversity of (technical) expertise in employees involved in business rule management!
  - E.g. domain experts, knowledge engineers, programmers, business analysts, CEOs, business process modelers, etc.
- Need for formalisms with a shared understanding!

AUTHORING

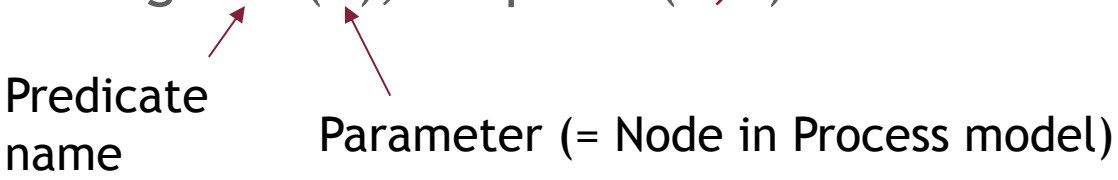
- Reminder – Business Rules…
  - concerning the structure and common behavior (of models)
    - Formalism I: Declare
  - concerning the behavior of single instances
    - Formalism II: Decision Model and Notation (DMN)

AUTHORING

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

DECLARE

- "New" Layer on top of CTL, to bridge gap between technicians and business analysts
- Set of "intuitive", predefined predicates
  - E.g. Init(X), Response(X,Y)



  - Predicates have a CTL-Semantics "under the hood"
    - Allows end-users to defined model queries, with the complexity of CTL "hidden" from them

DECLARE - EXAMPLES

- Init(A):
  - *"The process model should start with task A"*
- Response(A,D):
  - "Task A must be eventually followed by D"
- ChainResponse(A,B):
  - "Task A must be directly followed by B"

DECLARE - EXAMPLES

- Response(A,D):
  - “Task A must be eventually followed by D”

- How does this work “under the hood”
  - Predicates have predefined CTL Semantics!
  - Response(A,D) = $AG(A \rightarrow \neg\, EF(D))$

- *"Any task A should be eventually followed by a task D"*
  (legal text)

- Response(A,D)
  (Declare)

- $A \rightarrow EF(D)$
  (CTL-Syntax)

- $M,s \not\models A \quad \vee \quad M,s \models \forall \langle sRs' .. Rs_n \rangle, \exists i : M,s_i \models D$
  (CTL-Semantics)

DECLARE

- A set of DECLARE rules is a tuple $M = (A, T, C)$, where A is a set of tasks (e.g. labels from the process model), T is a set of constraint templates, and C is the set of actual constraints, which instantiate the template elements in T with tasks in A.

- Example:
  - Init (A) *(= constraint in C)*

Template in T

Task in A

DECLARE – CONSTRAINTS (SET T)

Template Name	Description	Semantics
Participation(a)	„a must occur at least once“	tbd
Response(a,b)	„If a occurs, then a must be *eventually* followed by b“	
ChainResponse(a,b)	„If a occurs, then a must be *directly* followed by b“	
Precedence(a,b)	„b occurs only if preceeded (somewhere) by a“	
ChainPrecedence(a,b)	„b occurs only if directly preceeded by a“	
NotResponse(a,b)	„if a occurs, it is **not** followed by b“	
NotChainResponse(a,b)	And so on…	
NotPrecedence(a,b)		
NotChainPrecedence(a,b)		

DECLARE - SYNTAX

- φ ::= a|(¬φ)|(φ1 ∧ φ2) | Xφ | Yφ | φ1Uφ2 | φ1Sφ2 | Fφ | Pφ | Gφ
  - X: next (directly following)
  - Y: previous (directly preceeding)
  - U: until
  - S: since
  - F: future (somewhere)
  - P: past (somewhere)
  - G: Globally

DECLARE – SYNTAX (RECAP)

- $M,s \models a$ iff a in $L(s)$
- (negation and $\wedge$, c.f. slides on CTL)
- $M,s_i \models X\varphi$ iff $\forall s'$, if $s_i R s'$, then $M,s' \models \varphi$
- $M,s_i \models Y\varphi$ iff $\forall s'$, if $s' R s_i$, then $M,s' \models \varphi$
- $M,s_i \models \varphi_1 U \varphi_2$ iff $\forall \langle s_i R s' \ldots R s_n \rangle$, $M,sj \models \varphi_2$ with $i \leq j \leq n$, and $M,s_k \models \varphi 1$ for all $k<j$
- $M,s_i \models \varphi_1 S \varphi_2$ iff $\forall \langle s_1 R s' \ldots R s_i \rangle$, $M,sj \models \varphi_2$ with $1 \leq j \leq i$, and $M,s_k \models \varphi 1$ for all k s.t. $j<k \leq i$

DECLARE – SYNTAX (RECAP) CONT‘D

- Helper: $M,s \models True$
- $M,s_i \models F\varphi$ iff $M,s_i \models True\ U\ \varphi$
- $M,s_i \models P\varphi$ iff $M,s_i \models True\ S\ \varphi$
- $M,s_i \models G\varphi$ iff $M,s_i \models \neg F \neg \varphi$

DECLARE – CONSTRAINTS (SET T)

Template Name	Description	Syntax
Participation(a)	„a must occur at least once)	EF a
Response(a,b)	„If a occurs, then a must be eventually followed by b“	G(a → Fb)
ChainResponse(a,b)	„If a occurs, then a must be directly followed by b“	G(a → Xb)
Precedence(a,b)	„b occurs only if preceeded (somewhere) by a“	G(b → Pa)
ChainPrecedence(a,b)	„b occurs only if directly preceeded by a“	G(b → Ya)
NotResponse(a,b)	„if a occurs, it is not followed by b“	G(a → ¬Fb)
NotChainResponse(a,b)	And so on…	G(a → ¬Xb)
NotPrecedence(a,b)		G(b → ¬Oa)
NotChainPrecedence(a,b)		G(b → ¬Ya)

REMINDER

"Task A must be eventually followed by D"

formalize

ChainResponse(A,D)

check

AUTHORING

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

DECISION MODEL AND NOTATION

Dish

decision

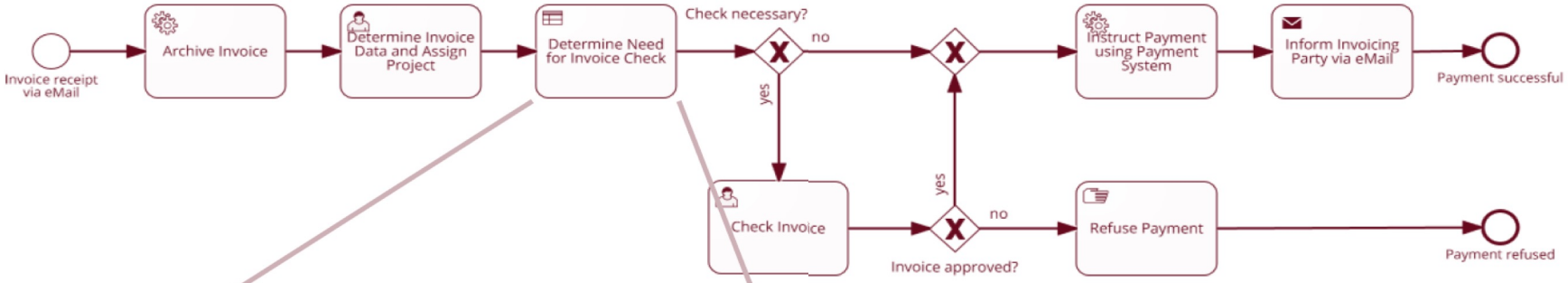
U	Input +		Output +
	Season	How many guests	Dish
	season	guestCount	desiredDish
	string	integer	string
1	"Fall"	<= 8	"Spareribs"
2	"Winter"	<= 8	"Roastbeef"
3	"Spring"	<= 4	"Dry Aged Gourmet Steak"
4	"Spring"	[5..8]	"Steak"
5	"Fall", "Winter", "Spring"	> 8	"Stew"
6	"Summer"	-	"Light Salad and a nice Steak"

- IF A_1 AND A_2 THEN B

[https://camunda.com/dmn]

IDEA

- Separate the process model from (parts of) the rules



Decision: Invoice check necessary?

Known creditor?	Amount >	Amount<=	Check necessary?
no		5.000	no
	5.000		yes
yes		10.000	no
	10.000		yes

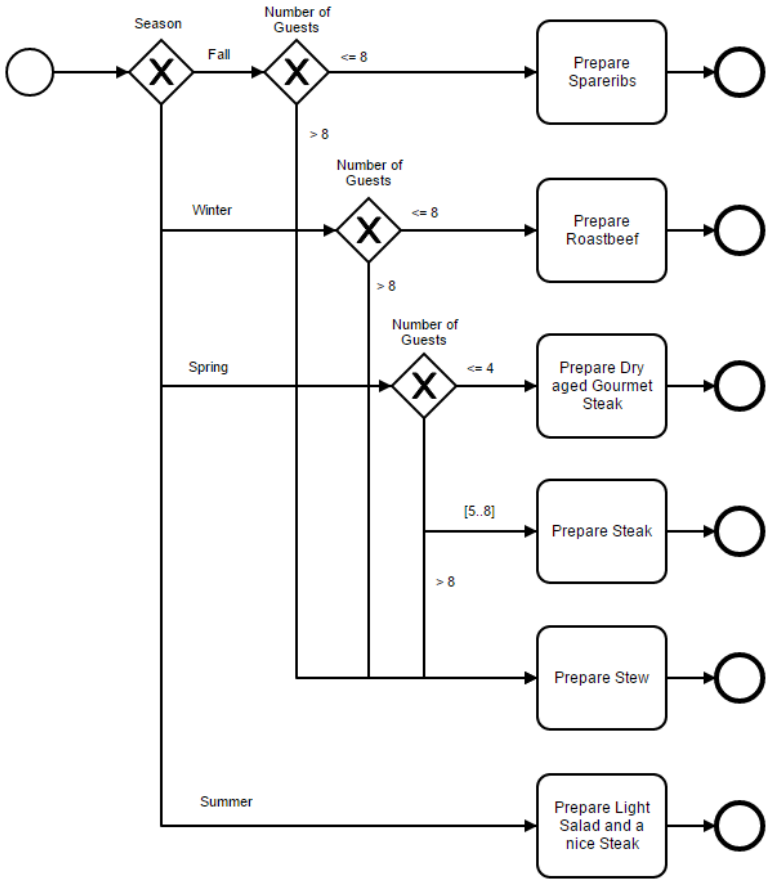
- Business rules are now encapsulated
- → Process base + rule base

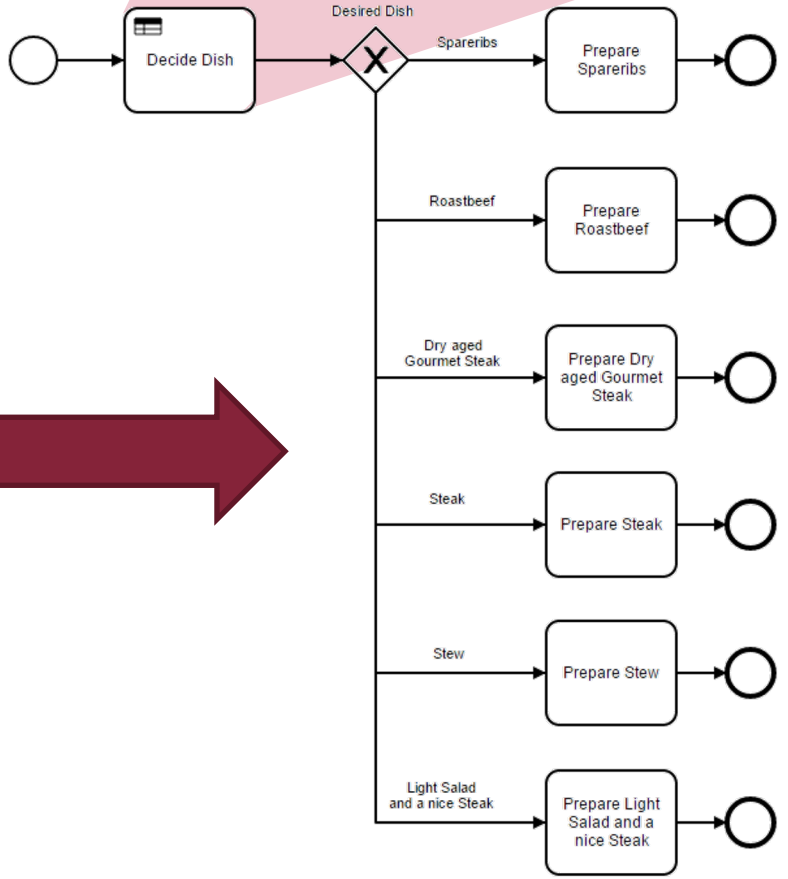
USING DMN TABLES AS DECISION OPERATORS

Dish

decision

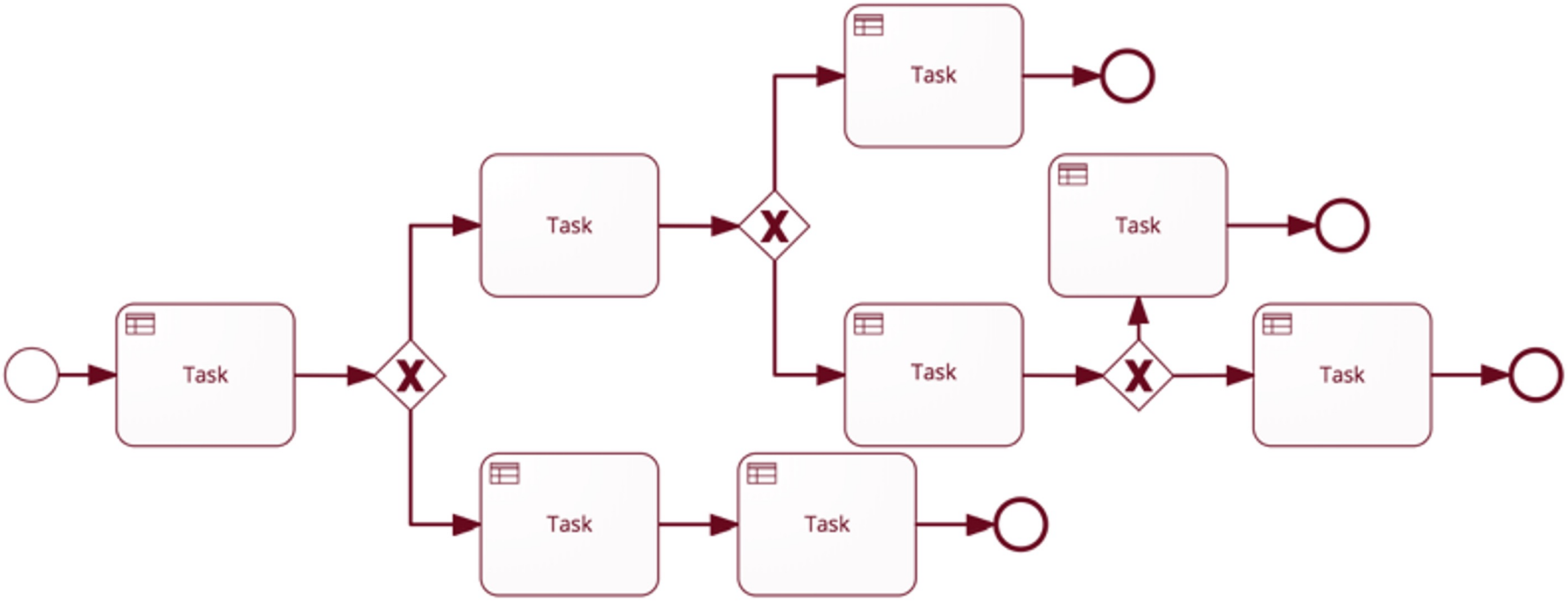
U	Input +		Output +
	Season	How many guests	Dish
	season	guestCount	desiredDish
	string	integer	string
1	"Fall"	<= 8	"Spareribs"
2	"Winter"	<= 8	"Roastbeef"
3	"Spring"	<= 4	"Dry Aged Gourmet Steak"
4	"Spring"	[5..8]	"Steak"
5	"Fall", "Winter", "Spring"	> 8	"Stew"
6	"Summer"	-	"Light Salad and a nice Steak"



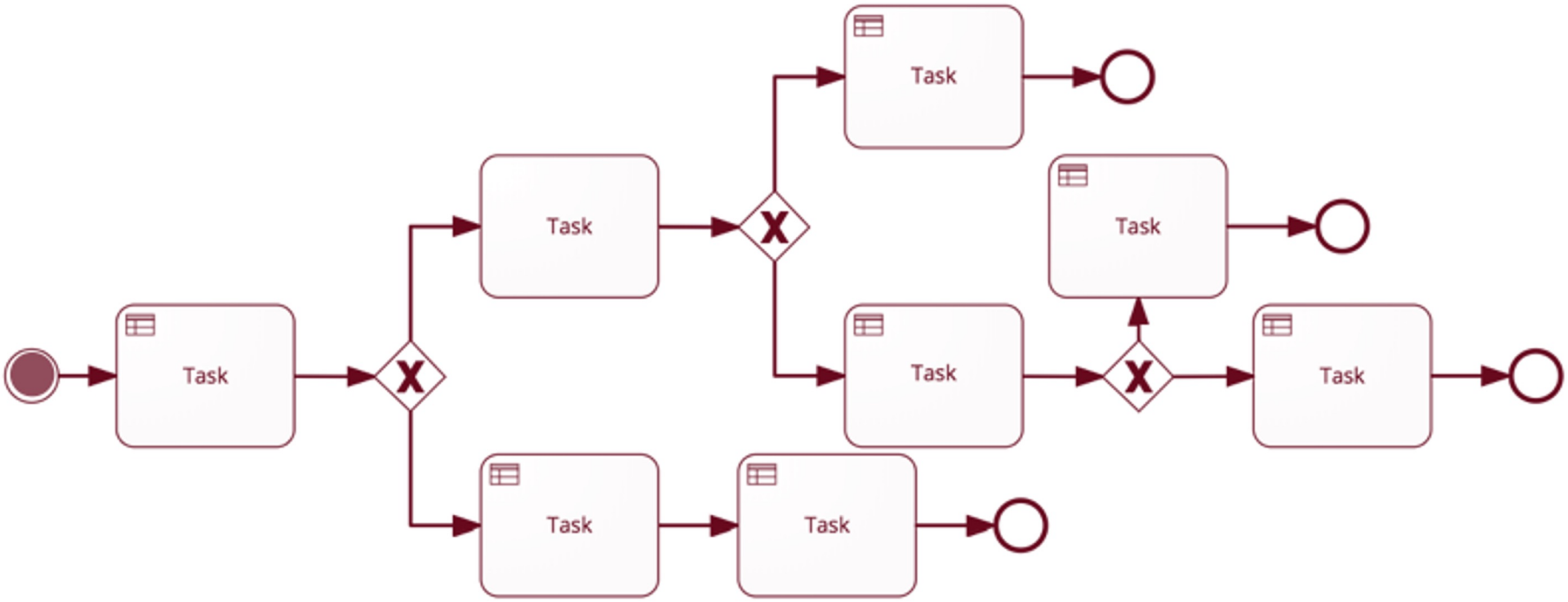


[https://camunda.com/dmn]

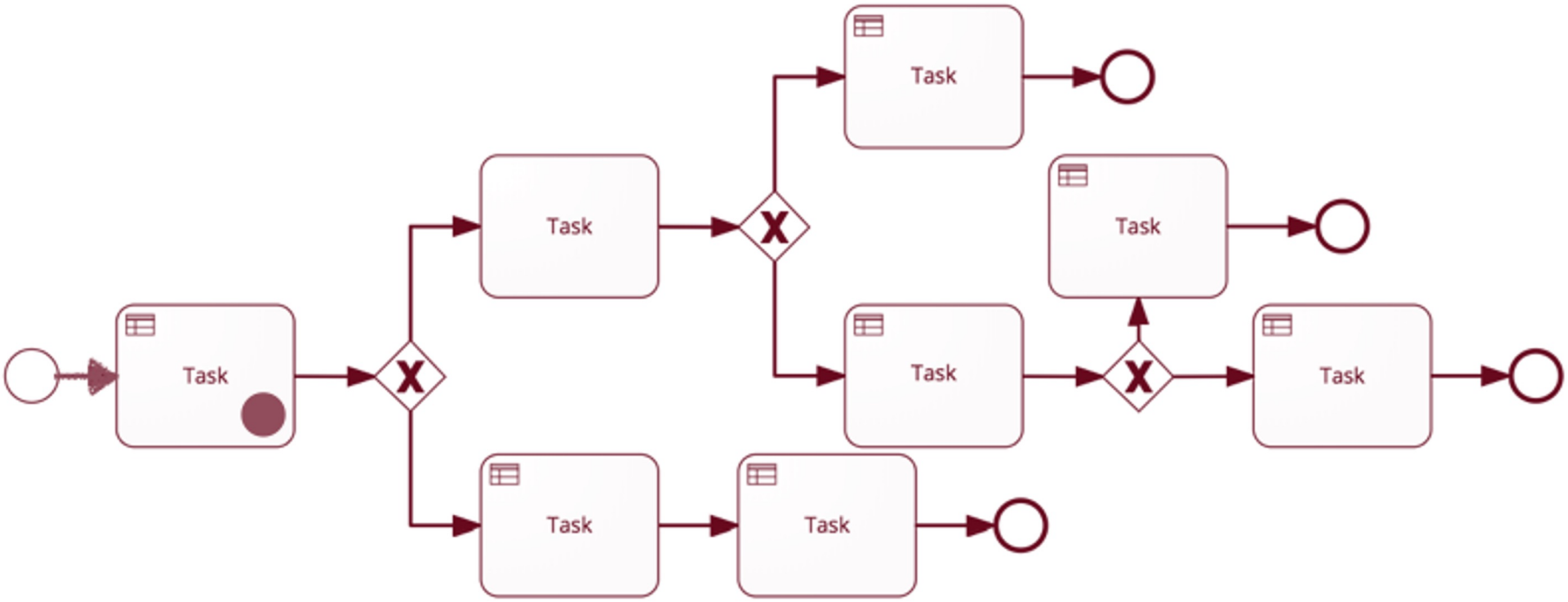
USING DMN TO ROUTE PROCESS INSTANCES



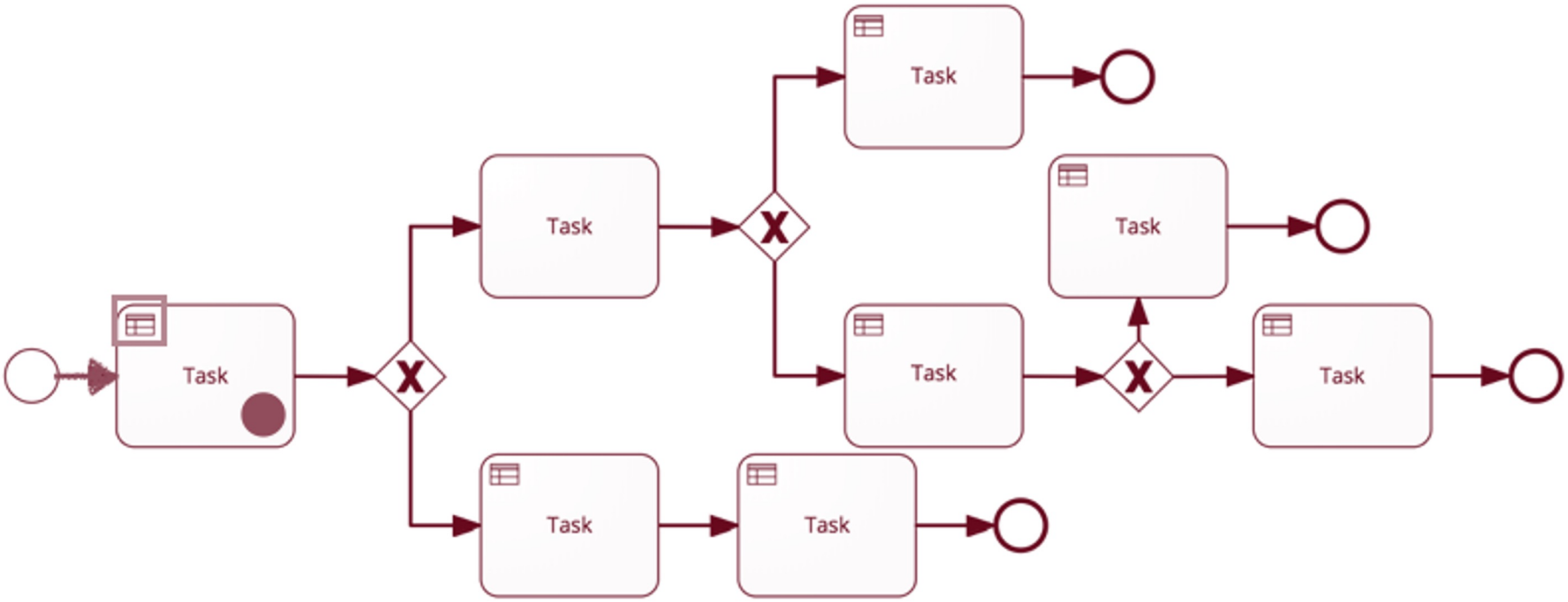
USING DMN TO ROUTE PROCESS INSTANCES



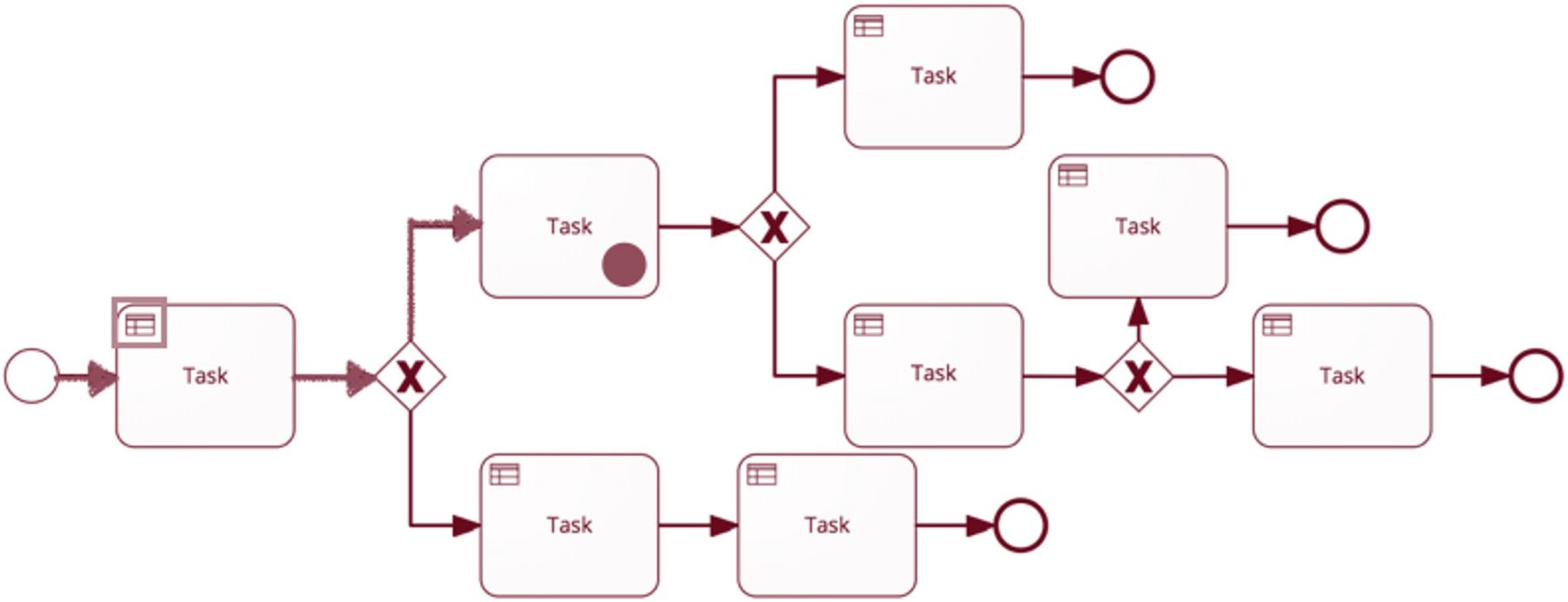
USING DMN TO ROUTE PROCESS INSTANCES



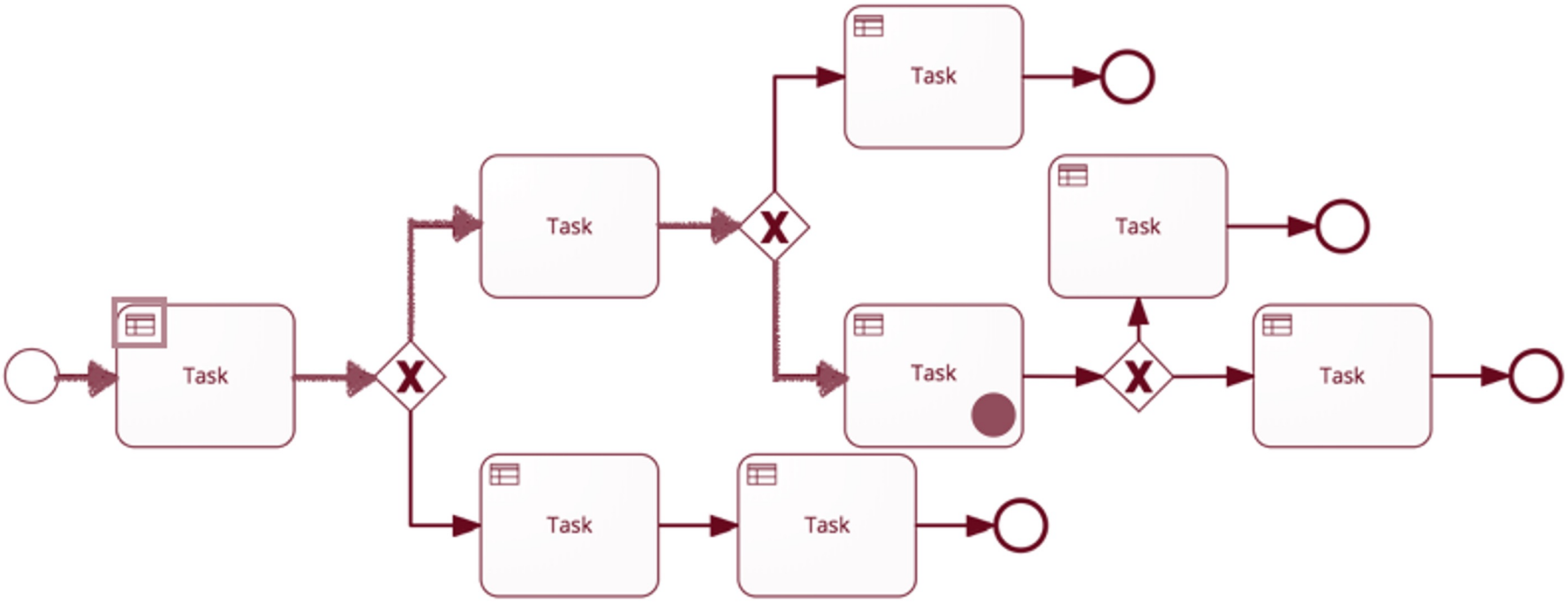
USING DMN TO ROUTE PROCESS INSTANCES



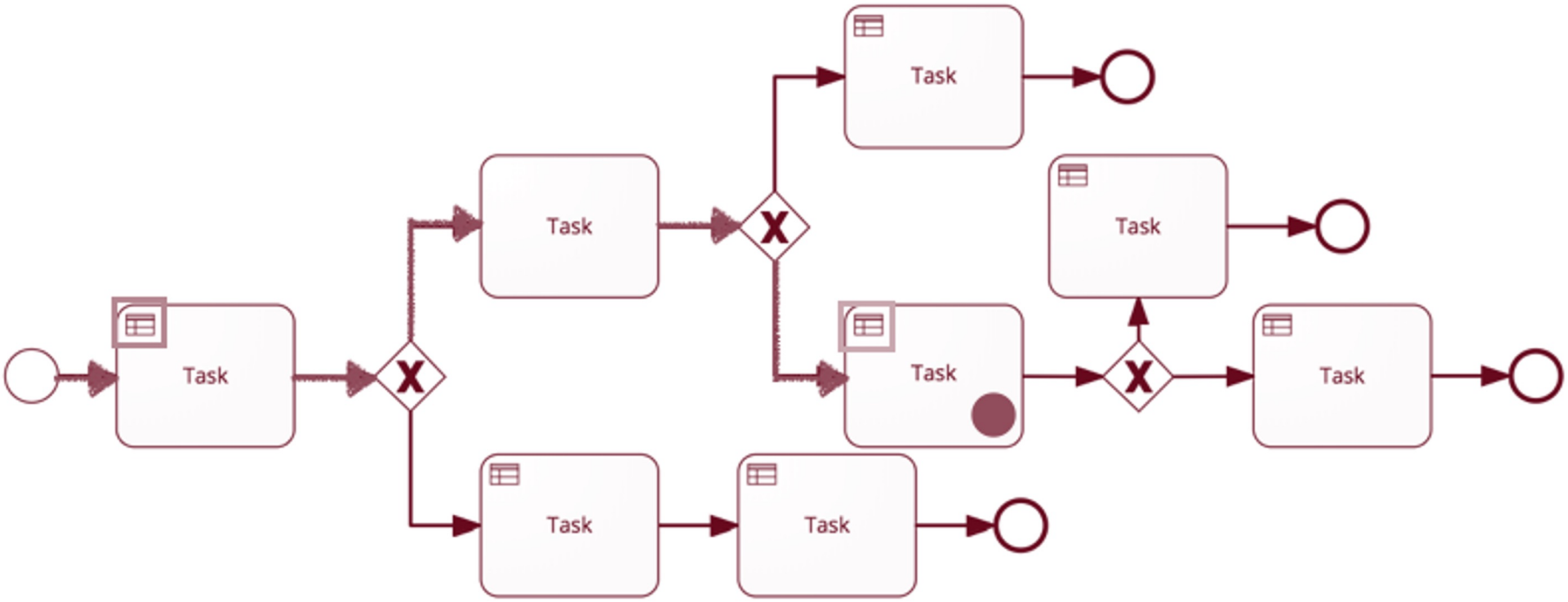
USING DMN TO ROUTE PROCESS INSTANCES



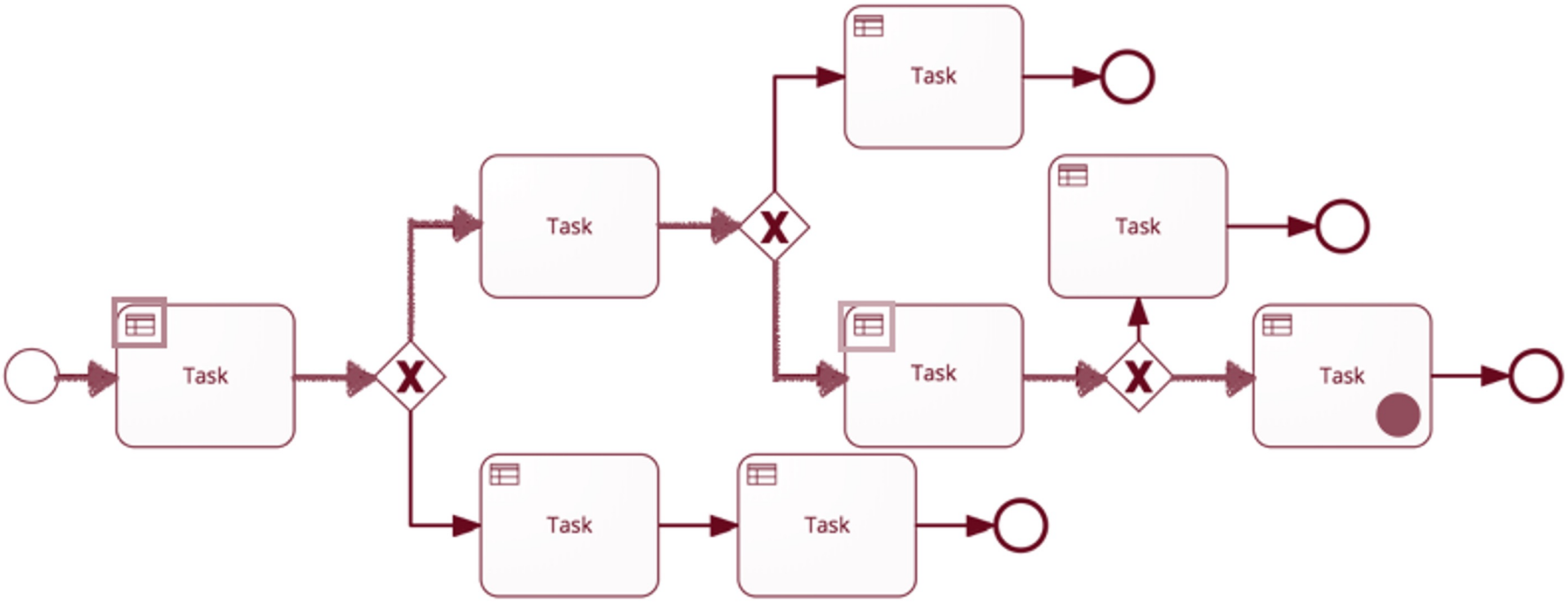
USING DMN TO ROUTE PROCESS INSTANCES



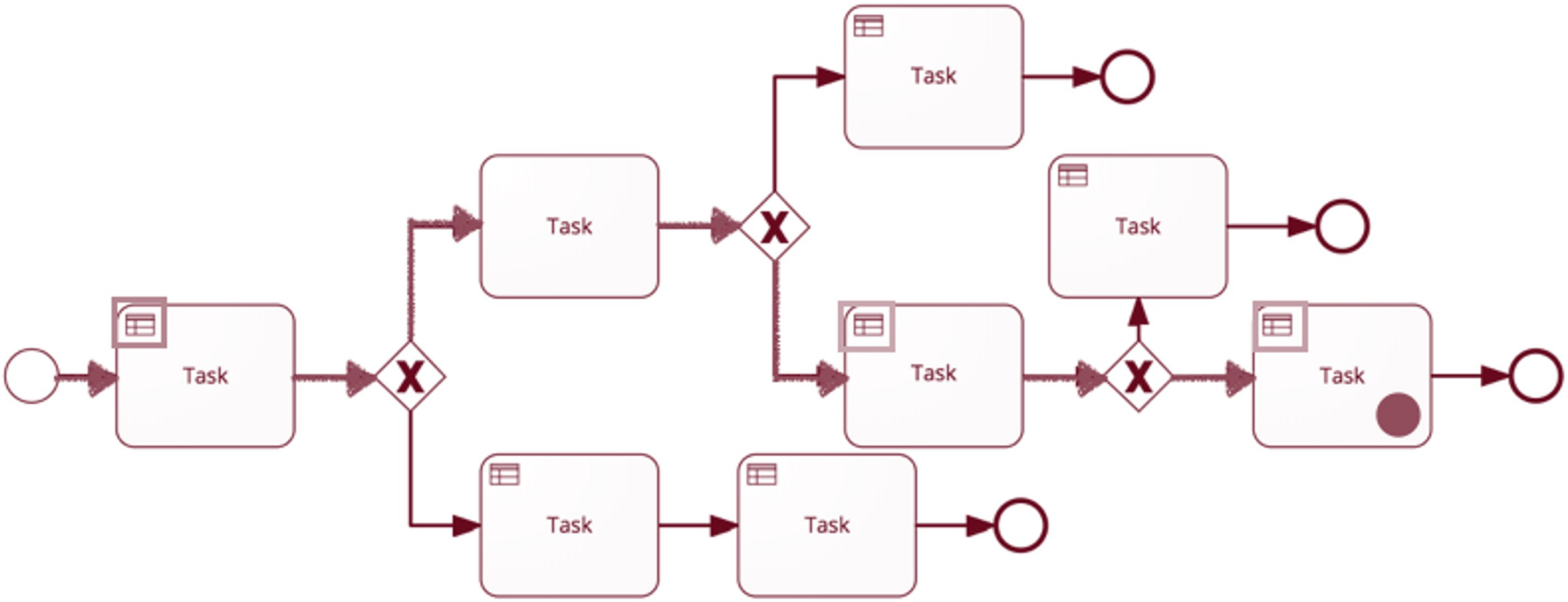
USING DMN TO ROUTE PROCESS INSTANCES



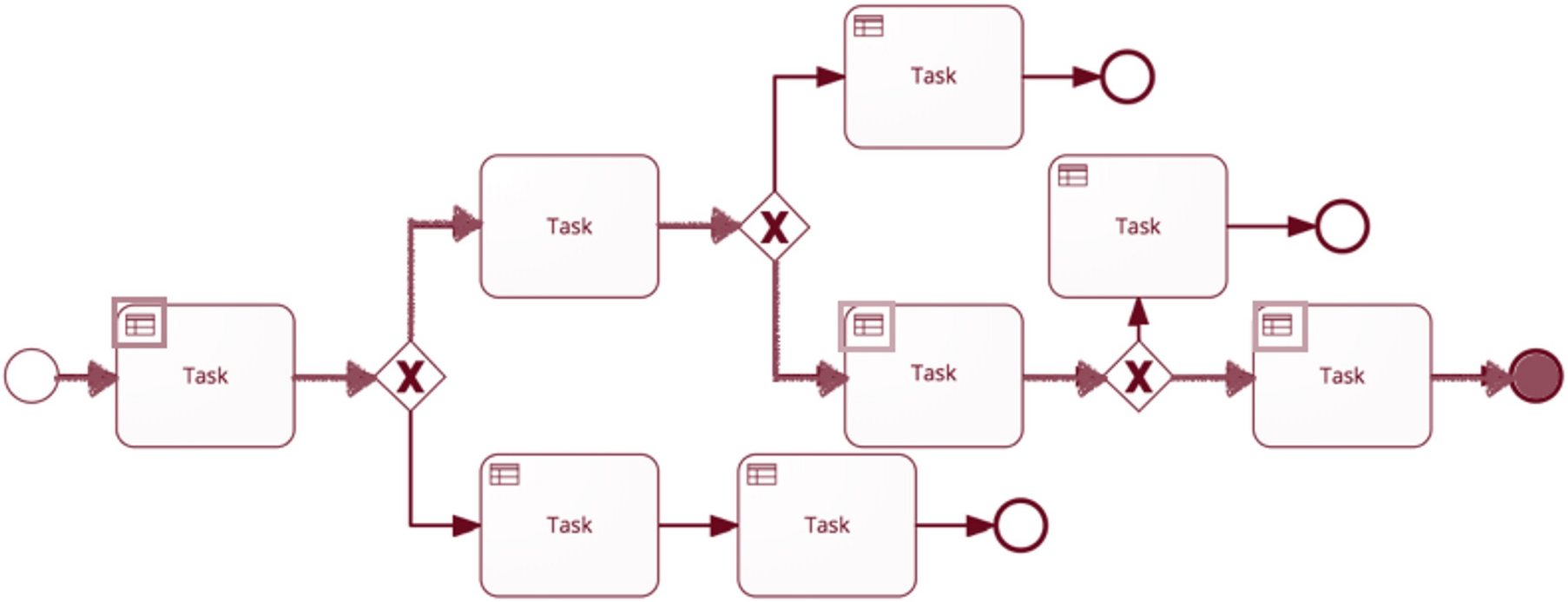
USING DMN TO ROUTE PROCESS INSTANCES



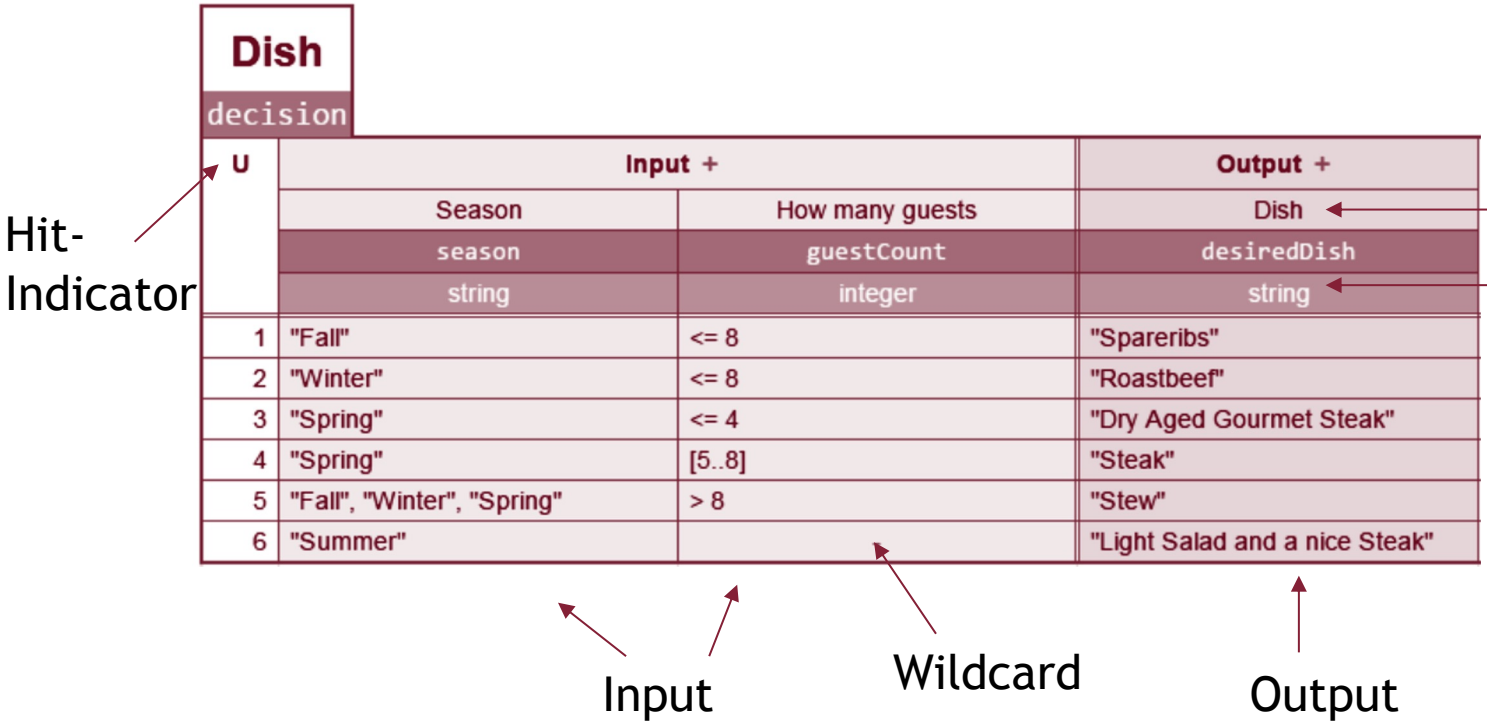
USING DMN TO ROUTE PROCESS INSTANCES



USING DMN TO ROUTE PROCESS INSTANCES



DMN - STANDARD



Each row = 1 business rule

[https://camunda.com/dmn]

DMN – HIT INDICATOR

- Unique: Only 1 rule may be activated (= „fire“)
- First: The first row which matches the input is activated (regardless of if there are other rules that may also fire)
- Collect: All rows that match the input are activated

DMN – DEFINITION

- A decision table ***D*** is a tuple (*T*, *I*, *O*, *Type*, *R*, *Order*, *H*) where:
  - *T* is the table name
  - *I* and *O* are disjoint, finite sets of input and output attributes (represented as strings)
  - *Type* : *I* ∪ *O* $\rightarrow \Gamma$ is a typing function that associates each input/output attribute to its corresponding data type (in Γ)
  - *R* is a finite set of rules $\{r_1, \ldots, r_n\}$. Each rule r_i is a pair $\langle If_i, Then_i \rangle$, where If_i is an input entry function that associates each input attribute m ∈ *I* to a condition over *Type*(m), and $Then_i$ is an output entry function that associates each output attribute n ∈ O an object in *Type*(n)

DMN – DEFINITION (CONT‘D)

- A decision table ***D*** is a tuple (*T, I, O, Type, R, Order, H*) where:
  - Order: R $\rightarrow$ {1, . . . , |R|} is a priority function injectively mapping rules in R to a corresponding rule number defining its priority (important for „first“ hit indicator)
  - H $\in$ {u, f, c} is the hit indicator, where u = unique, f = first and c = collect

DMN – SEMANTICS

- Reminder:

Dish

decision

U	Input +		Output +
	Season	How many guests	Dish
	season	guestCount	desiredDish
	string	integer	string
1	"Fall"	<= 8	"Spareribs"
2	"Winter"	<= 8	"Roastbeef"
3	"Spring"	<= 4	"Dry Aged Gourmet Steak"
4	"Spring"	[5..8]	"Steak"
5	"Fall", "Winter", "Spring"	> 8	"Stew"
6	"Summer"	-	"Light Salad and a nice Steak"

- IF A_1 AND A_2 THEN B

[https://camunda.com/dmn]

SUMMARY

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

LITERATURE

- Nelson, M. L., Rariden, R. L., & Sen, R. (2008, January). A lifecycle approach towards business rules management. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)* (pp. 113-113). IEEE.
- Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F. M., Montali, M., & Teinemaa, I. (2016, September). Semantics and analysis of DMN decision tables. In *International Conference on Business Process Management* (pp. 217-233). Springer, Cham.
- Pesic, M., Schonenberg, H., & Van der Aalst, W. M. (2007, October). Declare: Full support for loosely-structured processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)* (pp. 287-287). IEEE.
- Burattin, A., Maggi, F. M., & Sperduti, A. (2016). Conformance checking based on multi-perspective declarative process models. *expert systems with applications*, *65*, 194-211.

UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT

BUSINESS RULES MANAGEMENT PART 1

INSTITUTE FOR IS RESEARCH

www.uni-koblenz.de