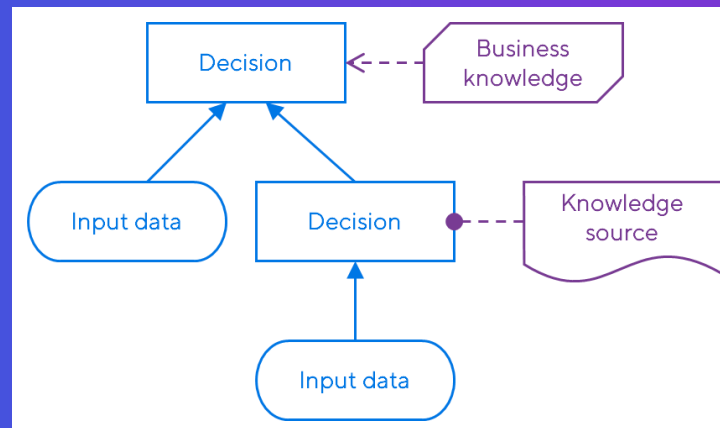


# Decision Management

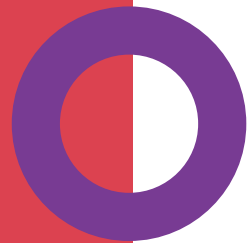
A deepdive into decisions, rules and DMN

Best practices in expressions, decision tables and business rules



blueriq

# Goals



## Understand

Decision management in general and what it means in Blueriq

Decision Model and Notation (DMN)

The value and purpose of a Decision Requirements Graph (DRG)

How to use DRG's in design time and in runtime

Design considerations regarding business values, and how to model accordingly

Design considerations regarding decision tables, and how to model accordingly

## Be able to

Spot and eliminate weaknesses in logic elements like decision tables, business rules, DRG's and expressions

Translate requirements into decisions, and from there into logic elements like decision tables, business rules, default expressions etc.

Draw a DRG, given a set of requirements

Evaluate and implement best practices into a comprehensible design

Design and model a transparent and maintainable decision model



blueriq

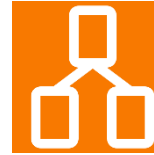
# Decisions versus Rules?



## Classic business rules

- Often large set of rules
- Can be difficult to understand... and maintain
- Change: impact unclear
- Can be used for other things than DM, like
- Control user interfaces
- Enforce data quality

**Complex**



## Decision management

- Limited set of decisions ... each possibly automated by a limited set of rules
- Recognizable by business
- Change: impact predictable
- Supports runtime analytics

**Transparent**

# Decision Model and Notation

Standard DMN makes use of a **Decision Requirements Graph (DRG)** with four shapes

- Decision
- ▭ Business knowledge
- Input data
- ▭ Knowledge source

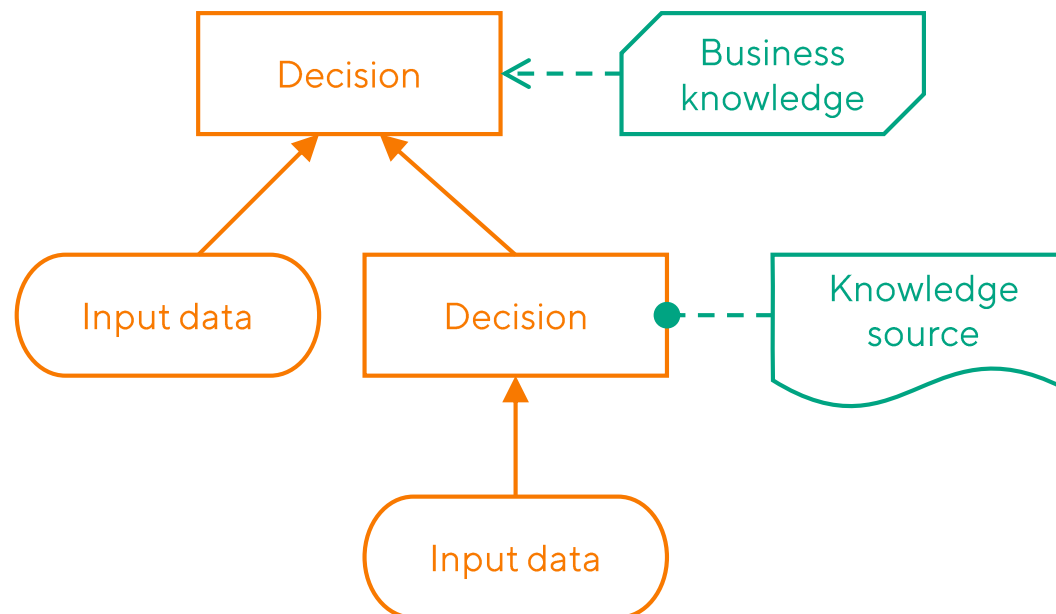
Three relation types

- Information
- > Knowledge
- Authority

Two levels

**Decision requirements level**

**Decision logic level**

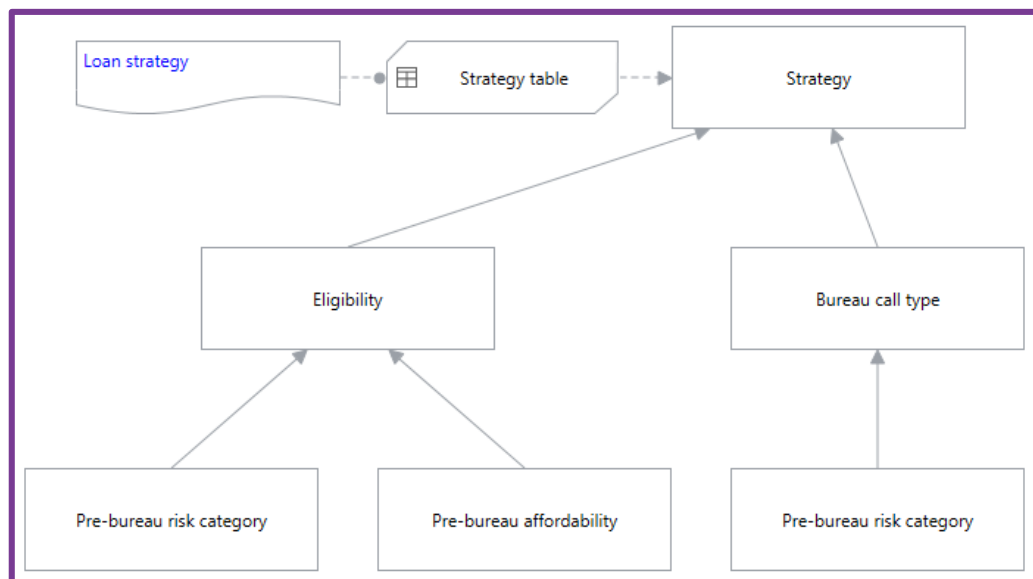


# DMN versus Blueriq

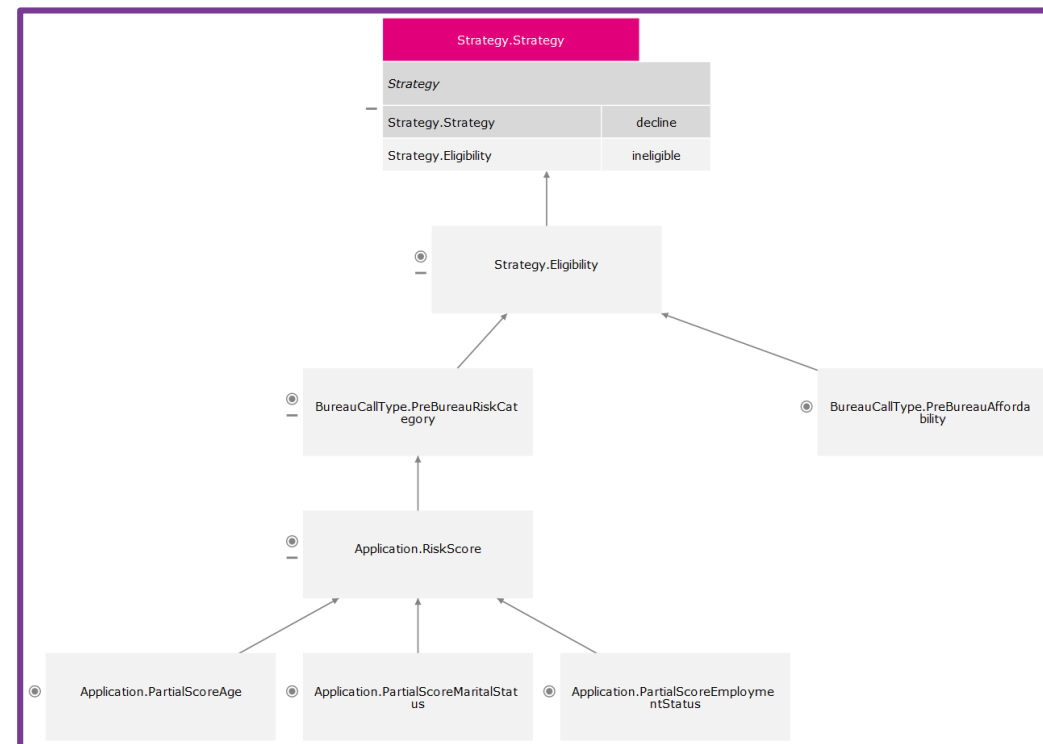
Decision Requirements Graphs in Blueriq differ slightly from standard DMN

- Icons are used for  decision table,  business rule and  expression
- Blueriq design time DRG and runtime DRG:

Follows DMN standard closely  
Shows all paths



Actual  
values  
user input  
or derived  
Only the  
path  
actually  
followed



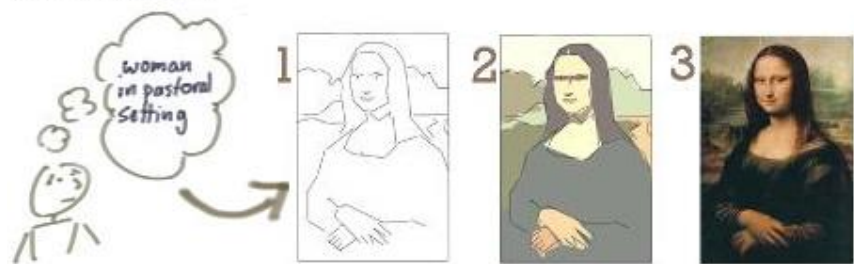
# Incremental versus Iterative (common mistake!)

- The best practices address the general principle of top down, iterative and incremental way of working
- DRG's support this way of modeling

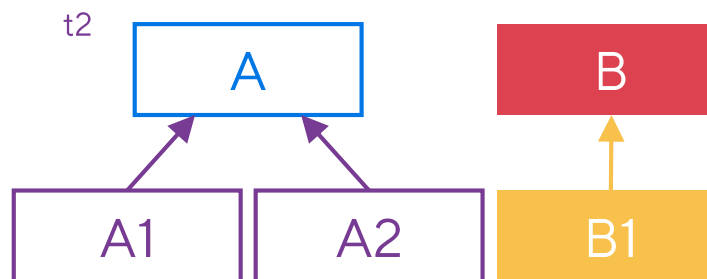
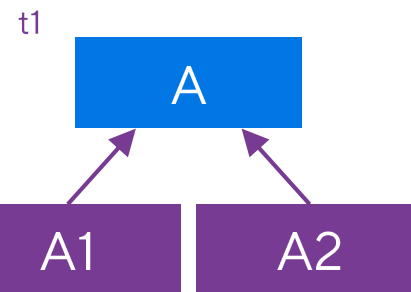
**Incremental**



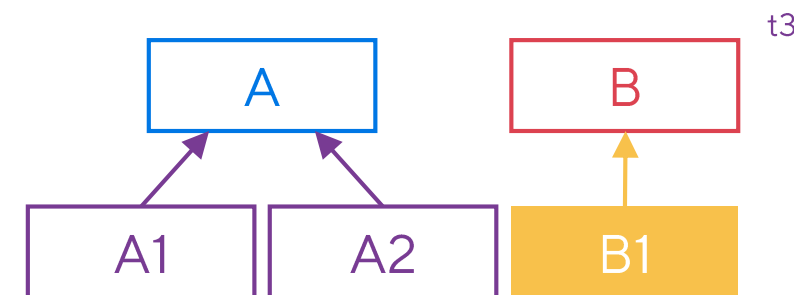
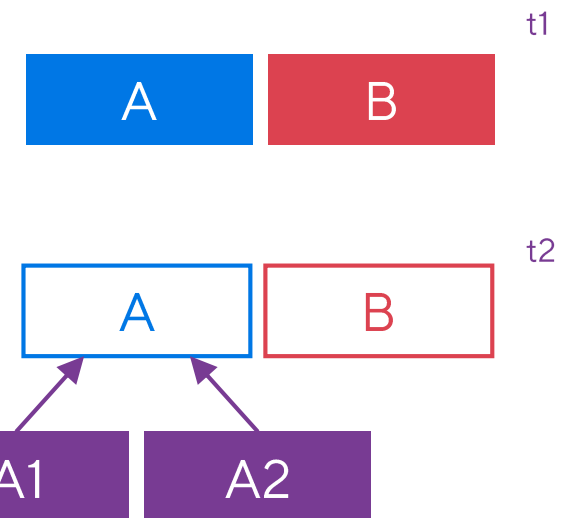
**Iterative**



**Incremental**



**Iterative**



# Setting a business value

Deriving a **binary attribute** (e.g. Male/Female or True/False), can be achieved in various ways:

- With a constant **default constant** and a **business rule**

- With **two business rules** (mutually exclusive)





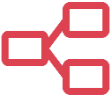

- With a **decision table**

Applicant.IncomeCategory	"Low"	"High"		?
Applicant.Discount	TRUE	FALSE	FALSE	FALSE

- With a **default expression** (for Boolean values)



# Order and hierarchy in logic

	<p>The inference engine first checks whether an attribute is <b>user-set</b>.</p> <ul style="list-style-type: none"> <li>- If so, all rules and defaults are ignored.</li> </ul>
	<p>If not, the inference engine tries to find a <b>rule</b> amongst all decision tables, business rules, external rules and data rules that can set the attribute. Therefore the inference engine evaluates the condition of these rules.</p> <ul style="list-style-type: none"> <li>- If more than rule is found that can actually set the attribute, an error is displayed.</li> <li>- If exactly one rule is found that can set the attribute, that particular rule is fired.</li> <li>- If no rule is found that can set the attribute, the default value (default constant, default expression, default decision table or default external rule) is used to set the attribute.</li> </ul>
	<p>If no such default value (constant, expression, external rule or decision table) is found, the value remains <b>unknown</b>.</p>
	<p>Furthermore, if any afore mentioned rule is part of a <b>rule group</b>, the participation of such rules is determined by the precondition of the rule group. In other words, if an attribute is not user-set and there is a decision table that could set it, but it is part of a rule group that has a precondition that is not met, the decision table does not participate.</p>
	<p>In case of <b>inheritance</b>, the inference engine will start searching for a way to derive the value of an attribute (a decision table, business rule, external rule or data rule) at the most specific level (the derived entity). If none exist there or their conditions are not satisfied, then the inference engine will look at the generic level (the base entity).</p>
	<p>In case of <b>specialization</b>, the most specialized rule is used. Inheritance has a higher priority however, i.e. if you have a rule in the lower module for the derived entity, it has priority over a specialized ruled for the generic entity. If you have defined two rules that both derive the same attribute, one in the lower module and one in the higher module, then both are used, and you get an error in case they derive a different value.</p>

# Quiztime!

- You are confronted with a model fragment
- You may find that there is something wrong with it
- If you find an error or see improvements, congratulations!
- And please keep it to yourself for now. Others might still be contemplating....



- Example (from a previous slide)

Applicant.IncomeBelowMaximum	TRUE	FALSE
Applicant.RentWithinLimits	TRUE	FALSE
Applicant.SavingsBelowMaximum	TRUE	FALSE
Applicant.EligibleForBenefit	TRUE	FALSE

# Exercise

Housing benefit is a benefit that is granted to any Dutch citizen that pays too much rent, in comparison with their income

The following rules apply for such a housing benefit:

- General rules such as nationality, age, type of residence, etc.
- Rules concerning your income
- Rules concerning your rent
- Rules concerning your savings



See Lab exercise in separate document

