Externalize Business Logic

With the help of DMN decisions

How we implement decisions currently

- tMap Filters
  - Variables needed
  - In- and Output Expressions
  - Can become complex
- Lookup-Tables
  - Without special logic only simple lookups possible
  - Actual logic must be implemented within the job
  - More complex logic expects dedicated SQL
- tJavaRow
  - Java implementation of if then else logic
- Problems:
  - We need to rebuild the job to update the logic
  - More complexity causes more complex jobs

Solution: Decision Modelling Notation

- Key element is the decision table
- A decision table consist of one or more rules (a row in the table)
- A decision table has input and output columns
- Different modi to evaluate the rules (hit policies)
- Multiple decisions can be used together
- Decisions can be constructed from various sources:
  - Excel
  - Database tables
  - Flat files
  - XML files
- Advantage: we only need to change the job if the decision needs more information or provides more information

DMN hit-policies

- Single Result (0, 1)
  - Unique. : checks if only one rule can be applied
  - First : takes the first matching rule
  - Priority : decide in which order the result should be chosen
  - Any : any rule matching (should return the same output)

- Multiple Results (0…n)
  - Collect: return all matching rule results
  - Collect (Sum, Min, Max, Count): aggregate outputs (returns 1 result)
  - Rule Order: like collect but ordered by rule order
  - Output order: like collect but ordered by predefined output value order

DMN hit-policies – use case examples

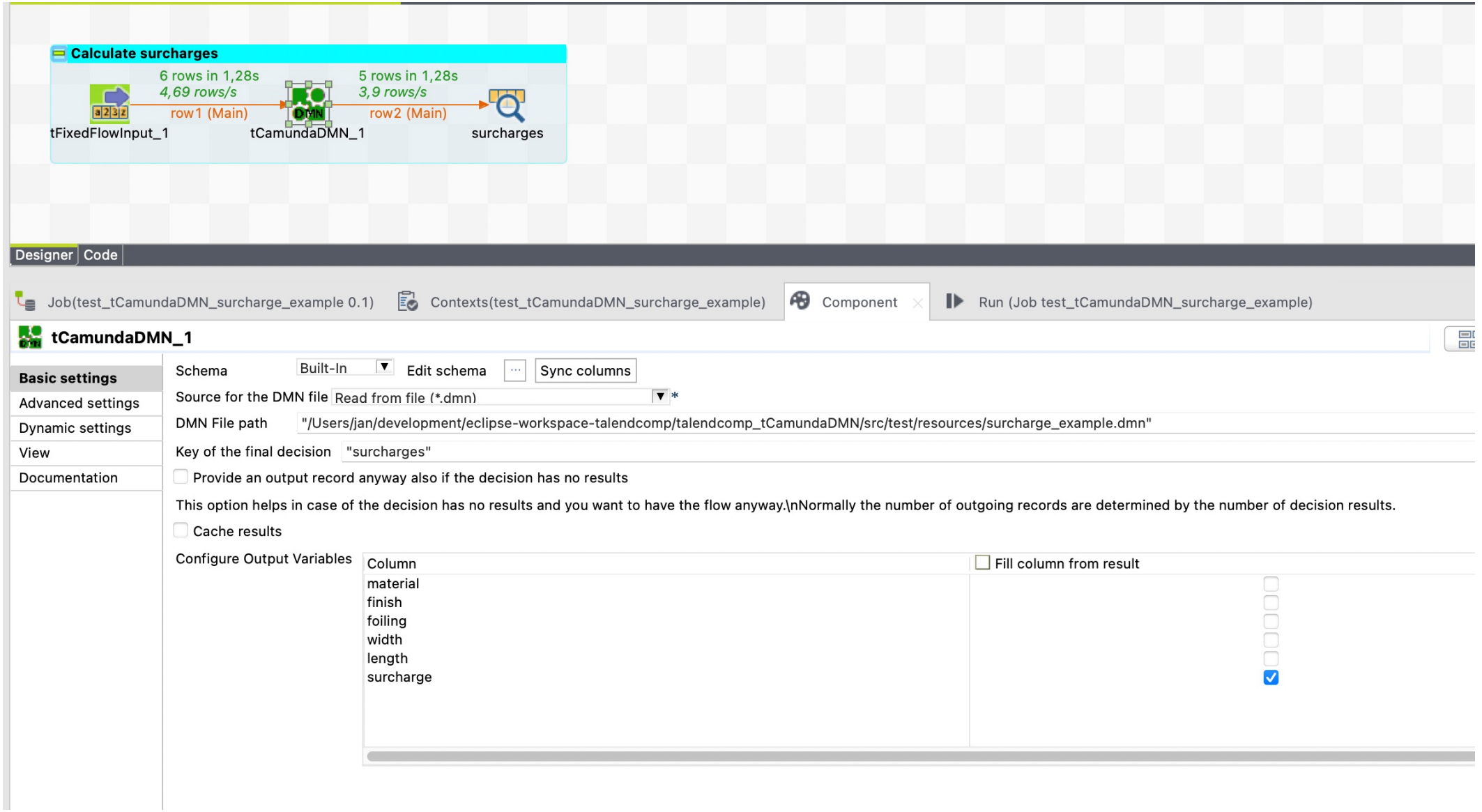
- Single Result (0, 1)
  - How to treat something based on different requirements
  - Who is current responsible as sales representative
  - Based on various condition choose one result (unique, first …etc)

- Multiple Results (0…n)
  - Surcharge calculation
  - Collect necessary items for an order for a product
  - Tools to be used for a special task
  - Based in various condition find all matching – optionally aggregate them
  - Ranking

Talend component tCamundaDMN

- Maps decision input variables to schema columns from the incoming flow
  - Detects if expected input variables are not present in the incoming schema
- Maps decision output variables to schema columns from the outgoing flow
  - Checks if the expected output does not match the decision output
  - Automatic type conversion
- Provides multiple rows per incoming row
  - Has also the ability to provide a row if no matching rule was found
  - All input columns will be transferred to the output (except the columns checked as output – means to be filled from the decision)
- Can read decisions from file or from resources (classpath)
- Caching of results to improve performance

How it looks like in the Job



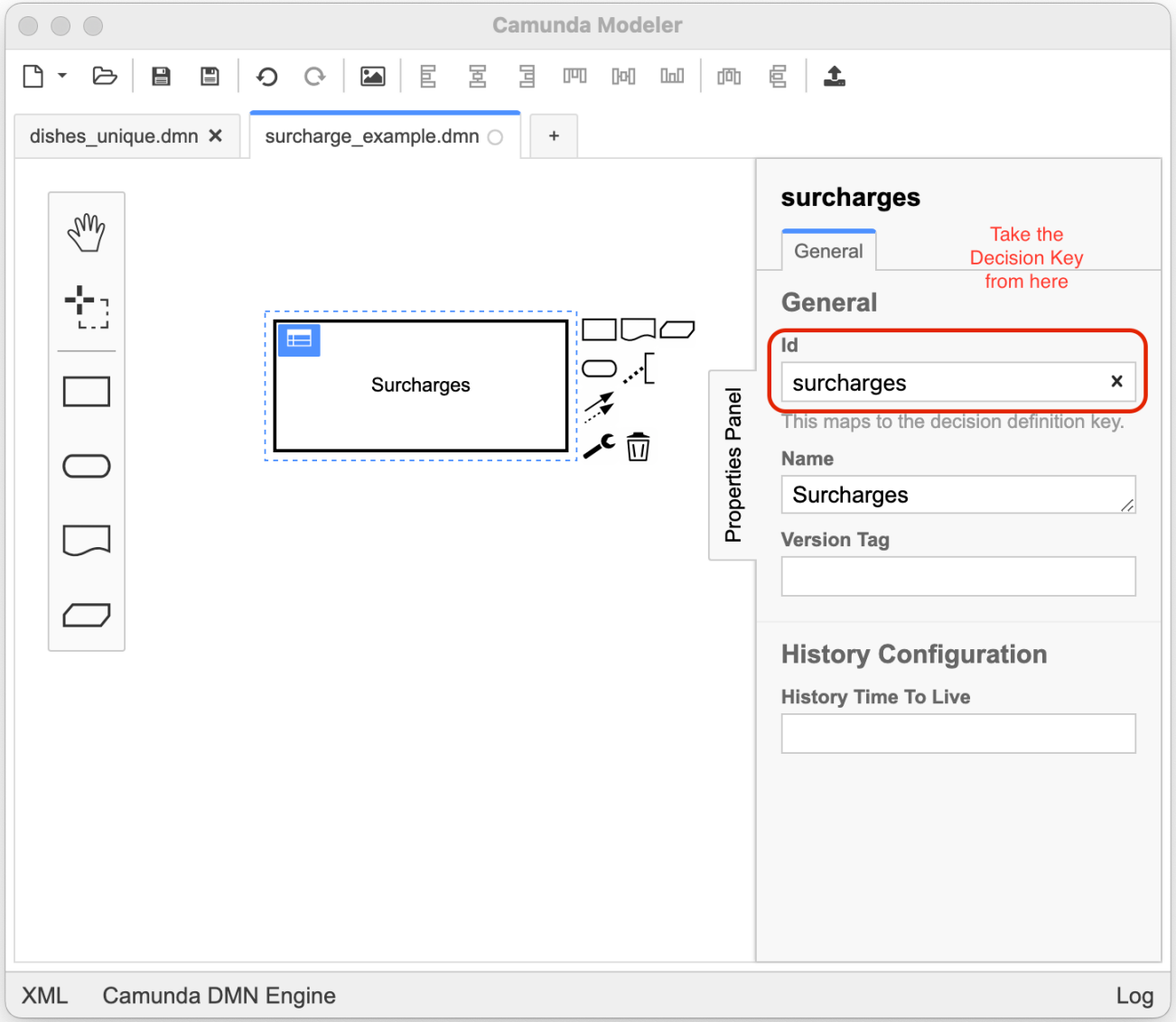
How to find key and vars in the Modeller

Change to the DRD view
(mostly upper left there is a “Edit DRD” button)

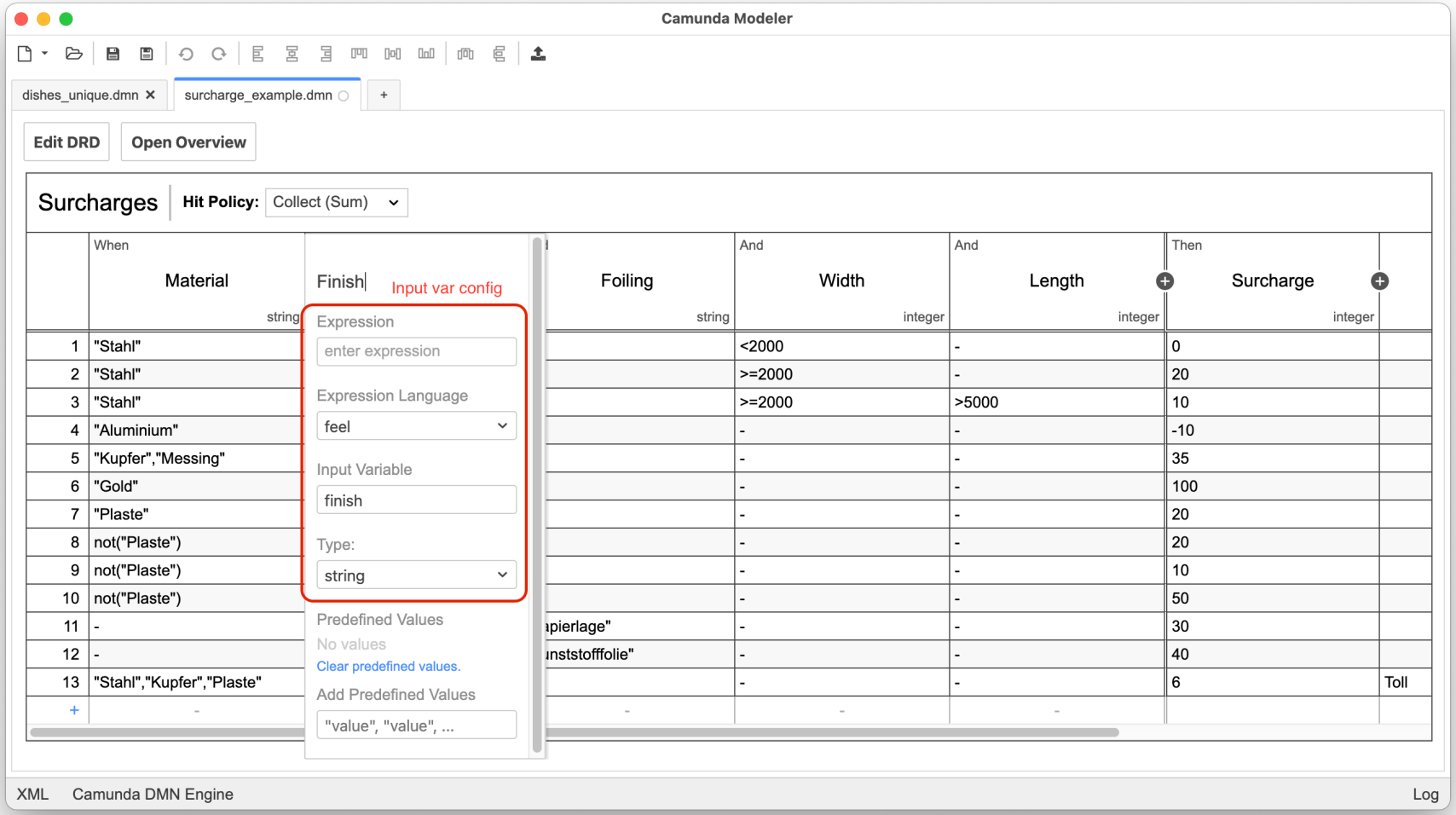
Select the (final) decision
(final if there are more than one decision)

Pick on the
Right side the Properties Panel.

The Id is the key we need in the
component as decision key



How to find key and vars in the Modeller



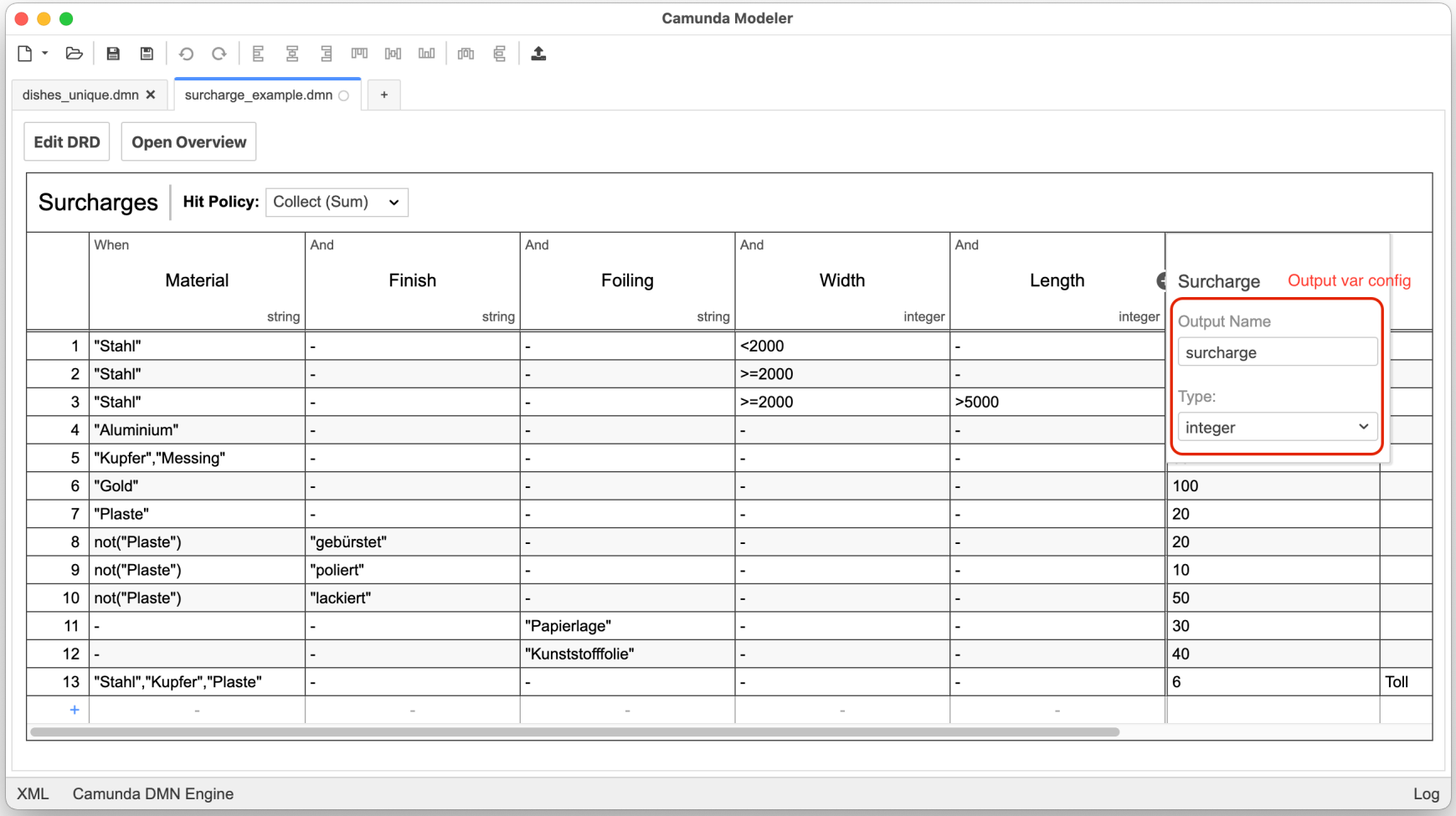
Click on the blue filled rectangle to edit a decision

Double click in a column in the header

Either the Input Variable or the Expression must be filled and must match to a Talend incoming schema column

Task care the type matches the schema column type

How to find key and vars in the Modeller



Click on the blue filled rectangle to edit a decision

Double click in a column in the header

The Output Variable must be filled and must match to a Talend outgoing schema column (marked as “Fill column from result”)

Task care the type matches the schema column type

Resources

https://camunda.com/dmn/

https://camunda.com/best-practices/choosing-the-dmn-hit-policy/

https://consulting.camunda.com/dmn-simulator/

https://github.com/xommaterials/talendcomp_tCamundaDMN

https://exchange.talend.com/#marketplaceproductoverview:marketplace=marketplace%252F1&p=marketplace%252F1%252Fproducts%252F1691&pi=marketplace%252F1%252Fproducts%252F1691%252Fitems%252F2986