Modeling Decisions with DMN and The Decision Model

Knut Hinkelmann
Decision Logic and Decision Task

(Ross 2011, p. 152f)
Modeling Decision Logic

There are two well-known modeling notations for Decision Logic

- The Decision Model
  - Based on a book from Barbara von Halle and Larry Goldberg

- Decision Model and Notation DMN
  - A new standard from OMG
Decision Model and Notation (DMN)
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
Main Concepts of DMN

Business Process

Decision Requirements

Decision Logic

(Coenen 2013)
Main concepts – Decision Requirements Level

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

Decision Requirements

(Coenen 2013)
## Constructs of a Decision Requirements Model

<table>
<thead>
<tr>
<th>Construct</th>
<th>DMN Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ELEMENTS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision</td>
<td>Decision</td>
<td>The act of determining an output from a number of inputs, using decision logic which may reference one or more business knowledge models.</td>
</tr>
<tr>
<td>Business Knowledge Model</td>
<td>Business knowledge</td>
<td>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.</td>
</tr>
<tr>
<td>Knowledge Source</td>
<td>Knowledge source</td>
<td>The authority for a business knowledge model or decision.</td>
</tr>
<tr>
<td>Input Data</td>
<td>Input data</td>
<td>Information used as an input by one or more decisions. It also denotes the parameters of a Business Knowledge Model.</td>
</tr>
<tr>
<td><strong>REQUIREMENTS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Requirement</td>
<td>Information - input data or decision output - required for a decision.</td>
<td></td>
</tr>
<tr>
<td>Knowledge Requirement</td>
<td>The invocation of a business knowledge model.</td>
<td></td>
</tr>
<tr>
<td>Authority Requirement</td>
<td>Showing the knowledge source of an element or the dependency of a knowledge source on input data.</td>
<td></td>
</tr>
</tbody>
</table>
A decision determines an output from a number of inputs by applying some decision logic.

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.

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 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.
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, or even analytic knowledge on how it might be made more accurate.

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

Input data **depends on** Knowledge Source

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

Knowledge Source **depends on** Decision

Knowledge Source **depends on** Business Knowledge

Knowledge Source 1 **depends on** Knowledge Source 2

(Coenen 2013)
Main concepts – Decision Logic

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

(Chenen 2013)
Modeling Decision Logic

"This will allow the import of many existing decision logic modeling standards (e.g. for business rules and analytic models) into DMN"

(Coenen 2013)
"I'm here because my boss said we should use more decisions tables for our project. What types of decision tables do you sell?"
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.
### Structure of a Decision Table in DMN

<table>
<thead>
<tr>
<th>Decision name</th>
<th>Set of Inputs</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hit policy</strong></td>
<td><strong>Completeness indicator</strong></td>
<td><strong>Name of Decision Table</strong></td>
</tr>
<tr>
<td><strong>HC</strong></td>
<td><strong>Input expression 1</strong></td>
<td><strong>Input expression 2</strong></td>
</tr>
<tr>
<td></td>
<td><strong>value 1a, value 1b</strong></td>
<td><strong>value 2a, value 2b</strong></td>
</tr>
<tr>
<td>1</td>
<td>value 1a</td>
<td>value 2a</td>
</tr>
<tr>
<td>2</td>
<td>value 1b</td>
<td>value 2b</td>
</tr>
</tbody>
</table>

- **Decision name**
- **Set of Inputs**
- **Output(s)**
- **Completeness indicator**
- **Name of Decision Table**
- **Hit policy**

- Lists of expected values (optional)
- Double line between inputs section and outputs section, and between inputs/outputs headers and the rule entry cells.
Decision Tables

Decision Modeling
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:

<table>
<thead>
<tr>
<th>Hit Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique</td>
<td>This is the default policy. All rules are exclusive and only a single rule is matched.</td>
</tr>
<tr>
<td>Any</td>
<td>Multiple matching rules, all matching rules with the same output. Any of these outputs can be used.</td>
</tr>
<tr>
<td>Priority</td>
<td>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.</td>
</tr>
<tr>
<td>First</td>
<td>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.</td>
</tr>
</tbody>
</table>
Hit Policies (2)

Multiple Hits Policies for Single Output

<table>
<thead>
<tr>
<th>Hit Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No order</td>
<td>Returns all hits in a unique list in arbitrary order.</td>
</tr>
<tr>
<td>Output order</td>
<td>Returns all hits in decreasing priority order. Output priorities are specified in an ordered list of values.</td>
</tr>
<tr>
<td>Rule order</td>
<td>Returns all hits in rule order, i.e. dependency on the order of the rules.</td>
</tr>
</tbody>
</table>

Aggregation for Multiple Hits Policy

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect</td>
<td>The result of the decision table is the list of all the outputs, ordered or unordered per the hit policy.</td>
</tr>
<tr>
<td>Sum</td>
<td>The result of the decision table is the sum of all the outputs.</td>
</tr>
<tr>
<td>Min</td>
<td>The result of the decision table is the smallest value of all the outputs.</td>
</tr>
<tr>
<td>Max</td>
<td>The result of the decision table is the largest value of all the outputs.</td>
</tr>
<tr>
<td>Count</td>
<td>The result of the decision table is the number of outputs.</td>
</tr>
<tr>
<td>Average</td>
<td>The result of the decision table is the average value of all the outputs, defined as the sum divided by the count.</td>
</tr>
</tbody>
</table>
FEEL = Friendly Enough Expression Language

FEEL is a script language for decision tables
Orientation of Rules in a DMN Decision Table

Rules as Rows:

Rules as Columns:

Rules as Crosstabs:
The Decision Model

The Decision Model

- The Decision Model was developed by Barbara von Halle and Larry Goldberg (2010)

- Objective:
  - a rigorous, repeatable, and technology-independent model of business logic that is simple to create, interpret, modify, and automate

- The Decision Model is a template for perceiving, organizing, and managing the business logic behind a business decision.

- It is a declarative representation of decision logic
  - specifies the conditions on which a decision is made
Decision Model Elements

A Decision Model has two different kinds of diagrams:

**Decision Model Diagram**

**Rule Family Table**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Student Loans</td>
<td>Person Business Loans</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
<tr>
<td>Is</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Decision Model Diagrams

- The root of a Decision Model diagram (its start) is an octagonal shape that represents the entire business decision.
  - It is this shape that relates to tasks within business process models.

- The other nodes in the Decision Model diagram represent Rule Families.

(von Halle & Goldberg 2010, p. 26f)
The Decision Model Diagram represents Rule Family Tables.

(von Halle & Goldberg 2010, p. 29)
Rule Family: Basic Element of the Decision Model

- Rule Family is a two-dimensional table relating conditions to one—and only one—corresponding conclusion.

![Diagram](image)

(von Halle & Goldberg 2010, p. 18f)
Rule Family Tables are Decision Tables

- A Rule Family table is a kind of decision table
  - In a Rule Family Table each row represents a rule
  - In a decision table each column represents a rule

- A Decision Model is a structured collection of decision tables

- There are some specialties:
  - A Rule Family must only have one conclusion column
  - Inferential relationships between Rule Family are made explicit in a Decision Model diagram

(von Halle & Goldberg 2010, p. 25)
Rule Family: Basic Element of the Decision Model

- A Rule Family node has three parts:
  - The name is the conclusion of the Rule Family
  - Inferred conditions: There are Rule Families with these names
  - Basic conditions: There are no Rule Families with these names

(von Halle & Goldberg 2010, p. 18f)
Determine Policy Renewal Method

Policy Renewal Method
Policy Tier Within Bounds (P2, P3)
Policy Renewal Override (P1)

This Condition column is part of a logical expression interpreted as “If/when the Policy Renewal Override is Yes”

The conclusion column is part of a logical expression interpreted as “then the Policy Renewal Method is Automatic”

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Manual Policy Override</th>
<th>Policy Tier Within Bounds</th>
<th>Policy Renewal Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>is</td>
<td>Yes</td>
<td>Manual Renewal Process</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>is</td>
<td>Manual Renewal Process</td>
</tr>
<tr>
<td>3</td>
<td>is</td>
<td>No</td>
<td>Automatic Renewal Process</td>
</tr>
</tbody>
</table>

A discrete business logic instance is a single row in the Rule Family table

This condition column is also part of a logical expression interpreted as “If/when the Policy Tier Within Bounds is Yes”

(von Halle & Goldberg 2010, p. 29)
Translating a Rule Family into Natural Language

- It is possible to convert each row in a Rule Family into a sentence that sounds natural to a business audience.

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Person Employment History</th>
<th>Person Mortgage Situation</th>
<th>Person Miscellaneous Loans Assessment</th>
<th>Person Likelihood of Defaulting on a Loan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Is Poor</td>
<td>Is Poor</td>
<td>Is High</td>
<td>Is High</td>
</tr>
</tbody>
</table>

- Possible Conversions

  ♦ If/when Person Employment History is Poor and Person Mortgage Situation is Poor and Person Miscellaneous Loans Assessment is High, then the Person Likelihood of Defaulting on a Loan is High.

  ♦ A Person with Poor Employment History and Poor Mortgage Situation and High Miscellaneous Loans Assessment has a High Likelihood of Defaulting on a Loan.

  ♦ It is obligatory that the Person Likelihood of Defaulting on a Loan is High if the Person Employment History is Poor and the Person Mortgage Situation is Poor and the Person Miscellaneous Loans Assessment is High.

(von Halle & Goldberg 2010, p. 20)
A Rule Family represents all Rules for one Conclusion

The Decision Model has only one Rule Family for each type of conclusion column, i.e. all rules for a conclusion are in one table.

(von Halle & Goldberg 2010, p. 29)
Rule Pattern

- A set of Rule Family rows with a common set of populated condition cells is called a Rule Pattern.

- The following Rule Family represents two rule patterns

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Conditions</strong></td>
<td><strong>Conclusion</strong></td>
</tr>
<tr>
<td></td>
<td>Person Student Loans</td>
<td>Person Customer Status</td>
</tr>
<tr>
<td></td>
<td>Person Business Loans</td>
<td>Person Miscellaneous Loans Assessment</td>
</tr>
<tr>
<td>1</td>
<td>Is</td>
<td>Is not Current customer</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Is Medium Risk</td>
</tr>
<tr>
<td>2</td>
<td>Is</td>
<td>Is not Current customer</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Is High Risk</td>
</tr>
<tr>
<td>1</td>
<td>Is</td>
<td>Is Current customer</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Is Low Risk</td>
</tr>
<tr>
<td>2</td>
<td>Is</td>
<td>Is Current customer</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Is Medium Risk</td>
</tr>
</tbody>
</table>

1. The 1st and 3rd rule have conditions for the fact types "Person Student Loans" and "Person Customer Status"
2. The 2nd and 4th rule have conditions for the fact types "Person Business Loans" and "Person Customer Status"

(von Halle & Goldberg 2010, p. 24)
Two dependent Rule Families

- Conditions of one rule family can depend on another rule family
- Example: Person Employment History in the first rule family depends on
  - Person Years at Current employer &
  - Person Number of Jobs in Past Five Years

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Person Employment History</strong></td>
<td>Person Mortgage Situation</td>
<td>Person Outside Credit Rating</td>
</tr>
<tr>
<td>Is</td>
<td>Poor</td>
<td>Is</td>
</tr>
<tr>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Pattern</th>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person Years at Current Employer</td>
<td>Person Number of Jobs in Past Five Years</td>
<td><strong>Person Employment History</strong></td>
</tr>
</tbody>
</table>

(von Halle & Goldberg 2010, p. 23)
The Decision Model diagram shows the inferential relationships between Rule Families.

Solid lines between Rule Family nodes represent inferential relationships:
- The name of the node at the end with the dot occurs as condition in the other node.

Leave nodes in a Decision Model diagram to not have inferred conditions.

(von Halle & Goldberg 2010, p. 26f)
The Rule family directly connected to the business decision shape is called the “Decision Rule Family”, its conclusion is the conclusion sought by the entire Decision Model.

A Decision Model diagram begins with an octagonal shape that represents the entire business decision. The other shapes in the Decision Model diagram represent Rule Families. This diagram has 6 Rule Families.

### The Decision Rule family

- **Policy Renewal Method**
  - Policy Tier Within Bounds (P2, P3)
  - Policy Renewal Override (P1, P3)

- **Insured Major Ownership Change**
  - Insured Minority Stockholder (P2)
  - Insured Majority Stockholder (P3)
  - Insured Board Change (P1)
  - Insured CEO Change (P1, P3)

- **Insured Major Location Change**
  - Insured Location Zip-5 (P1)
  - Insured Location Occupied Square Footage (P2)
  - Insured Location Construction (P3)

### Conditions

- **Policy Tier Within Bounds**
  - Policy Discount (P2)
  - Policy Tier (P1) (P2)

- **Policy Discount**
  - Policy Grade (P1)
  - Package Grade (P1)
  - Package Discount (P1)
  - Location State Category (P1)

### Conclusion

- **Determine Policy Renewal Method**

---

(von Halle & Goldberg 2010, p. 28)
The solid line terminated by the dot connects Rule Families that have an inferential relationship: The conclusion of one Rule Family is used as a condition in another.

The solid line: The conditional columns are shown below the solid line and above the dotted line.

Inferred Conditions

Conditions based on facts

The dotted line: The labels below the dotted line denote condition column headings that do not serve as a conclusion column heading in another Rule Family.

(von Halle & Goldberg 2010, p. 28)
All labels below the Rule Family name denote condition column headings.

The (Pnumber) denotes Rule Pattern numbers within the Rule Family. Where the (Px) appears with no header but a symbol [...] that indicates a pattern with no conditions.

The solid line terminated by the dot connects Rule Families that have an inferential relationship. In this case the condition column “Policy Renewal Override” in the Decision Rule Family has an inferential relationship with the conclusion column of the “Policy Renewal Override” Rule Family.

The labels below the solid line but above the dotted line denote condition columns that serve as a conclusion column heading in another Rule Family.

The labels below the dotted line denote condition columns that do not serve as conclusion columns in another Rule Family. These condition columns will be populated by known fact values (e.g. persistent data).

(von Halle & Goldberg 2010, p. 28)
The Decision Model vs. DMN: Diagrams

- On the graphical level, the Decision Model Diagram is a subset of DMN's Decision Requirements Diagram
  - Decision is in both diagrams
  - Rule Family corresponds to Business Knowledge Model
- DMN is more expressive; compared to the Decision Model Diagram contains Input data and Knowledge Sources
The Decision Model vs. DMN: Decision Tables

- Decision Model and DMN use decision tables to represent the decision logic.
- The main structural differences are down to the split cell versus single cell convention for the operator and operand.
- The semantics of decision tables in DMN is more expressive: It can return multiple values and can specify, how multiple values are aggregated.

# Decision Tables in DMN and TDM

### DMN Decision Table

<table>
<thead>
<tr>
<th>Student Course Eligibility Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

### TDM Rule Family View

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row ID</td>
<td>Rule Pattern</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

### Key Features

- **Single-hit decision table** (returns a single value)
- **Hit-Policy**: Any (signified by the 'A' in the top-left corner)
- Possible domain values are provided optionally underneath the name of the expression
- No specific order of reading

### Conclusion

- **Fact Type**: Regular (returns a single value)
- A RFV has no concept of a hit-policy per se; any row can always be hit
- The Rule Pattern denotes the pattern of cells populated. There are three different patterns in this RFV
- No specific order of reading
- The operator is spelled out in words - this is not mandatory, but is designed to be more business-friendly

---

Prof. Dr. Knut Hinkelmann

Decision Modeling
Literatur


