

AN EXECUTABLE DMN SOLUTION FOR BUSINESS USERS

Bruce Silver, methodandstyle.com
bruce@brsilver.com

Edson Tirelli, RedHat
ed.tirelli@gmail.com

First Complete DMN Implementation

Redhat Decision Test

Page 1

Cars

Make-Model

ModelYear

Style

Price

isConvertible
 True False

hasDriverAirbags
 True False

hasPassengerAirbags
 True False

hasSideAirbags
 True False

hasRollbar
 True False

hasAlarm
 True False

AutoRisk
(car, HighTheftList)

	inputs	outputs		
U	onHighTheftList	car.Price	car.isConvertible	PotentialTheftCategory
	Boolean true, false	Number (..)	Boolean true, false	Text [High, Moderate, Low]
1	true	-	-	"High"
2	false	>45000	-	"High"
3	false	<=45000	true	"High"
			false	"Moderate"

<http://www.omg.org/spec/DMN/20151101/>
<http://www.omg.org/spec/FEEL/20140401/>
<http://www.omg.org/spec/DMN/20151101/>

```

<itemDefinition name="tCar">
  <itemComponent name="Make-Model">
    <typeRef>feel:string</typeRef>
  </itemComponent>
  <itemComponent name="ModelYear">
    <typeRef>feel:number</typeRef>
  </itemComponent>
  <itemComponent name="Style">
    <typeRef>feel:string</typeRef>
    <allowedValues>
      <text>&quot;compact&quot;& &
    </allowedValues>
  </itemComponent>
  <itemComponent name="Price">
    <typeRef>feel:number</typeRef>
  </itemComponent>
  <itemComponent name="isConvertible">
    <typeRef>feel:boolean</typeRef>
  </itemComponent>
  <itemComponent name="hasDriverAirbags">
    <typeRef>feel:boolean</typeRef>
  </itemComponent>
  <itemComponent name="hasPassengerAirbags">
    <typeRef>feel:boolean</typeRef>
  </itemComponent>
  <itemComponent name="hasSideAirbags">
    <typeRef>feel:boolean</typeRef>
  </itemComponent>
  <itemComponent name="hasRollbar">
    <typeRef>feel:boolean</typeRef>
  </itemComponent>
    
```

1

AutoPremiumList

sum(AutoPremiumList)

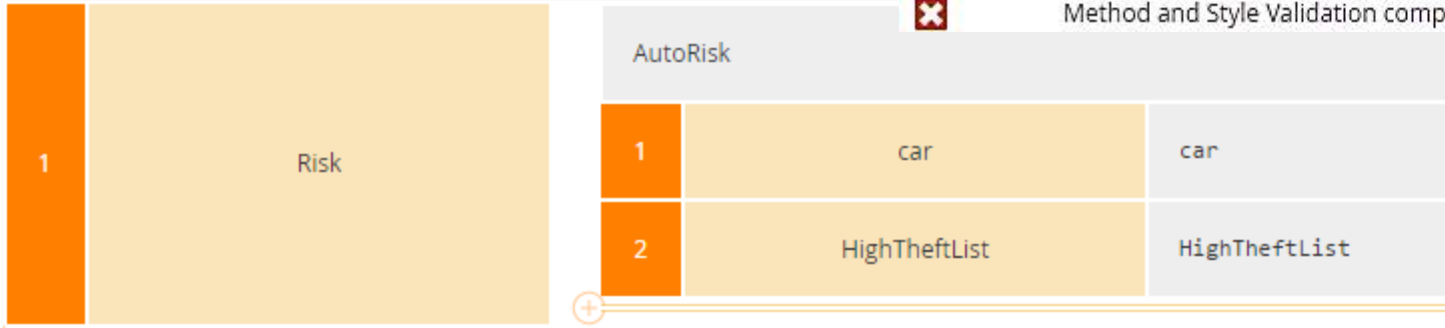
for car in Cars return AutoPremium(car, HighTheftList, CurrentYear)

Why Did It Take So Long?

Validate  M&S M&S Data Type
Diagram Analysis About Data
Method & Style

AutoEligibilityScore

(car, HighTheftList)



Overlap: Rules 3, 4 match with input values [660, >=0.38]. Hit policy U allow

Gap: No rule matches for input values [>660, (0.35..0.38)].

Gap: No rule matches for input values [>660, 0.35].

Warning: Table is not fully contracted. Combine rules 1, 2.

Method and Style Validation completed with 4 errors on 1 decision table.

```
if Risk.AutoEligibility="Not eligible" then 100 else if Risk.AutoEligibility="Provisional" then 50 else 0
```

Standards-Enabled Integration

- Trisotech
- RedHat
- Method and Style

DMN METHOD *& Style*



PARTNERS

Technologies

Services & support

Success S

Let's do something great. Together.

Executable Logic...Created Graphically

- Diagrams and tables, not code

Redhat Decision Test

Page 1

Cars

Make-Model

ModelYear

Style

Price

isConvertible
 True False

hasDriverAirbags
 True False

hasPassengerAirbags
 True False

hasSideAirbags
 True False

hasRollbar
 True False

hasAlarm
 True False

AutoRisk

(car, HighTheftList)

1 onHighTheftList if car.MakeModel in (HighTheftList) then true else false

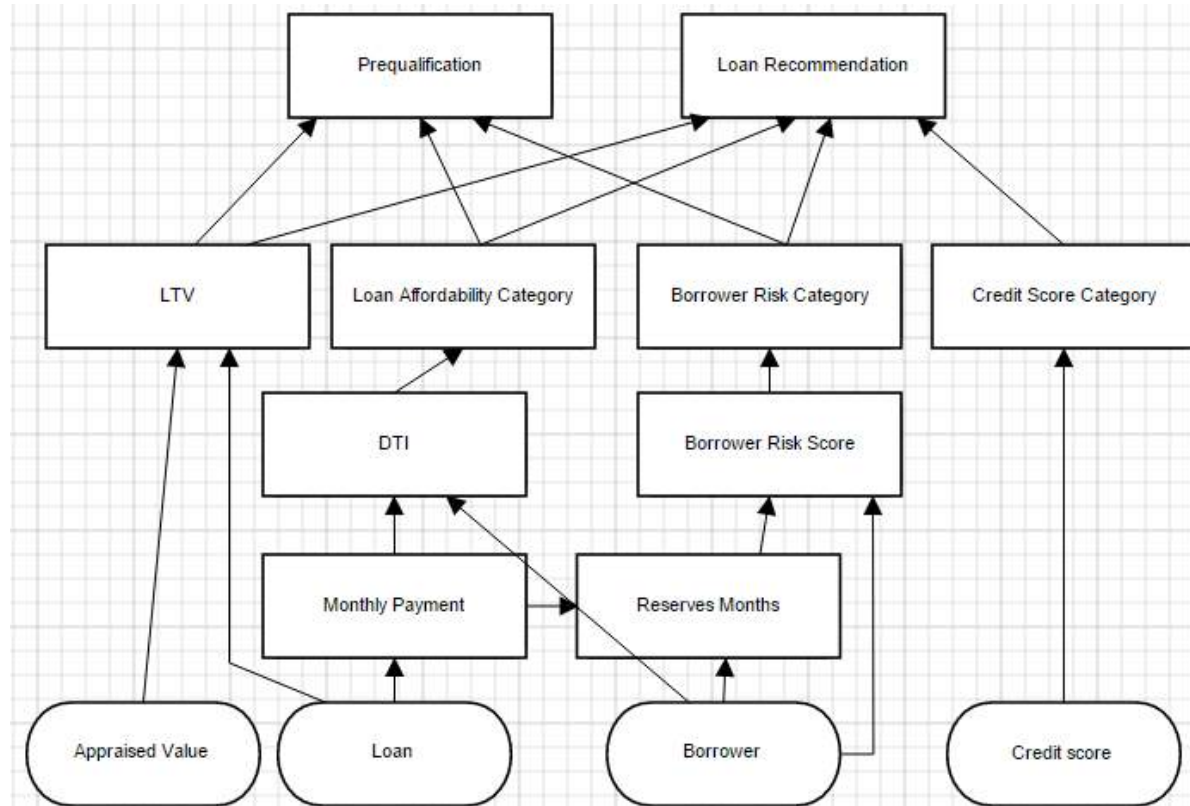
2 PotentialTheftCategory

	inputs			outputs
U	onHighTheftList	car.Price	car.isConvertible	PotentialTheftCategory
	Boolean true, false	Number (,)	Boolean true, false	Text [High, Moderate, Low]
1	true	-	-	"High"
2	false	>45000	-	"High"
3	false	<=45000	true	"High"
4	false	[20000..45000]	false	"Moderate"
5	false	<20000	false	"Low"

inputs

DRD Reveals Names in Scope

- Key to parsing names with spaces



Decision Table Analysis



Validate

Diagram

M&S

M&S

DT
Analysis

About

Method & Style



Data Type

Data

U	Mortgage type	Number of units	isSecondary	Max LTV
	<i>Text</i> [primary residence, second home]	<i>Number</i> [1, 2, 3, 4]	<i>Boolean</i> true, false	<i>Number</i> (..)
1	"primary residence"	1	true	0.95
2	"primary residence"	1	false	0.95
3	"primary residence"	2,3,4	true	0.75
4	"primary residence"	2,3,4	false	0.80
5	"second home"	-	true	0.80
6	"second home"	-	false	0.85

U	Credit score	DTI	Loan Approval
	<i>Number</i>	<i>Number</i>	<i>tLoanApproval</i> ["Approved", "Declined"]
1	>660	<=0.35	"Approved"
2	>=660	>=0.38	"Declined"
3	[600..660]	<=0.35	"Approved"
4	[600..660]	>0.35	"Declined"
5	<600	-	"Declined"

Log Viewer

Severity Message



Overlap: Rules 2, 4 match with input values [660, >=0.38]. Hit policy U allows only 1 matching rule.



Gap: No rule matches for input values [>660, (0.35..0.38)].



Warning: Table is not fully contracted. Combine rules 1, 3.

Severity Message



Second normal form violation. Input 3 is irrelevant for rules [1, 2]. Combine those rules for input 3.

Business-Friendly Interface

- Create types automatically from table column headings

The diagram illustrates the process of automatically creating a type from a table column heading. It shows a table with columns 'FICO Score' and 'Credit Risk Category'. An arrow points to the same table where the 'Credit Risk Category' column now has a type definition 'tCreditRiskCategory' with a list of values 'A', 'B', 'C', 'D', 'E', 'F'. A dialog box titled 'Credit Risk Category' is open, showing the 'Editing Header Type' configuration for 'Text' with 'Enumeration' constraints and a list of values 'A' through 'F'. A 'Create Type' button is highlighted in red.

	FICO Score	Credit Risk Category
U	tFICOScore [300..850]	Text
1		

	FICO Score	Credit Risk Category
U	tFICOScore [300..850]	tCreditRiskCategory ["A", "B", "C", "D", "E", "F"]
1		

Credit Risk Category
Editing Header Type

Type: Text

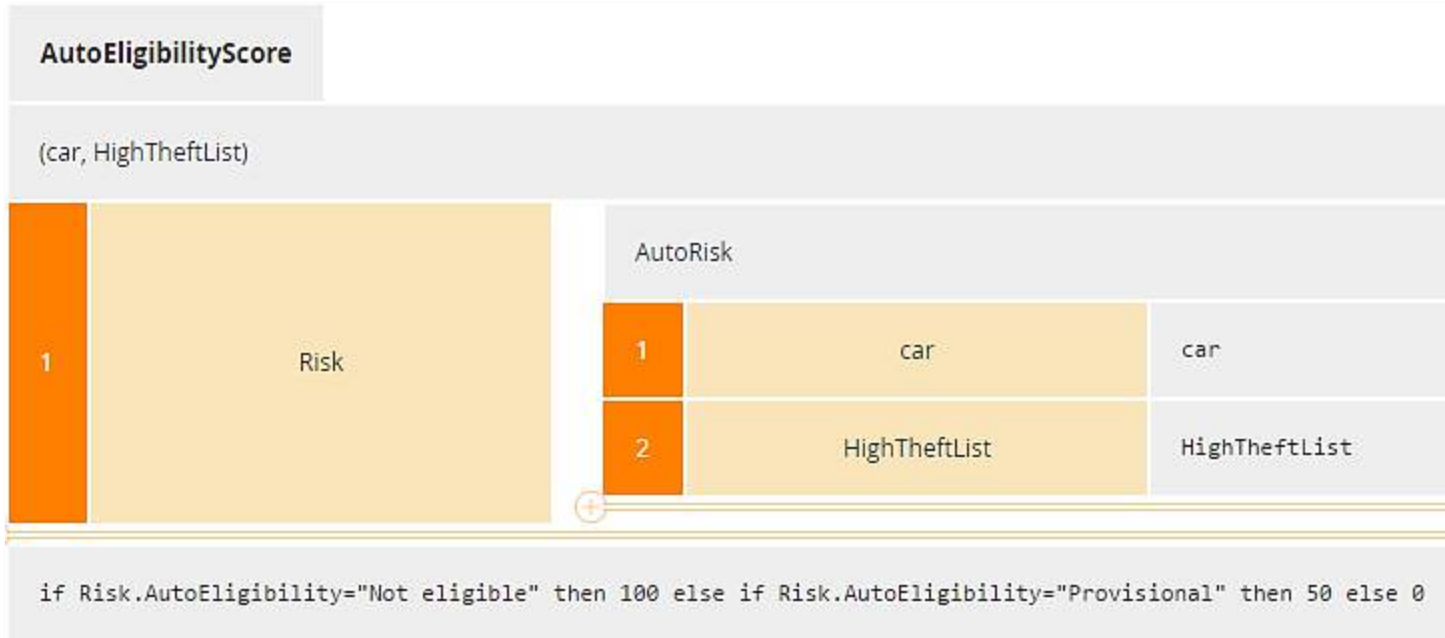
Constraints: Enumeration

Values:

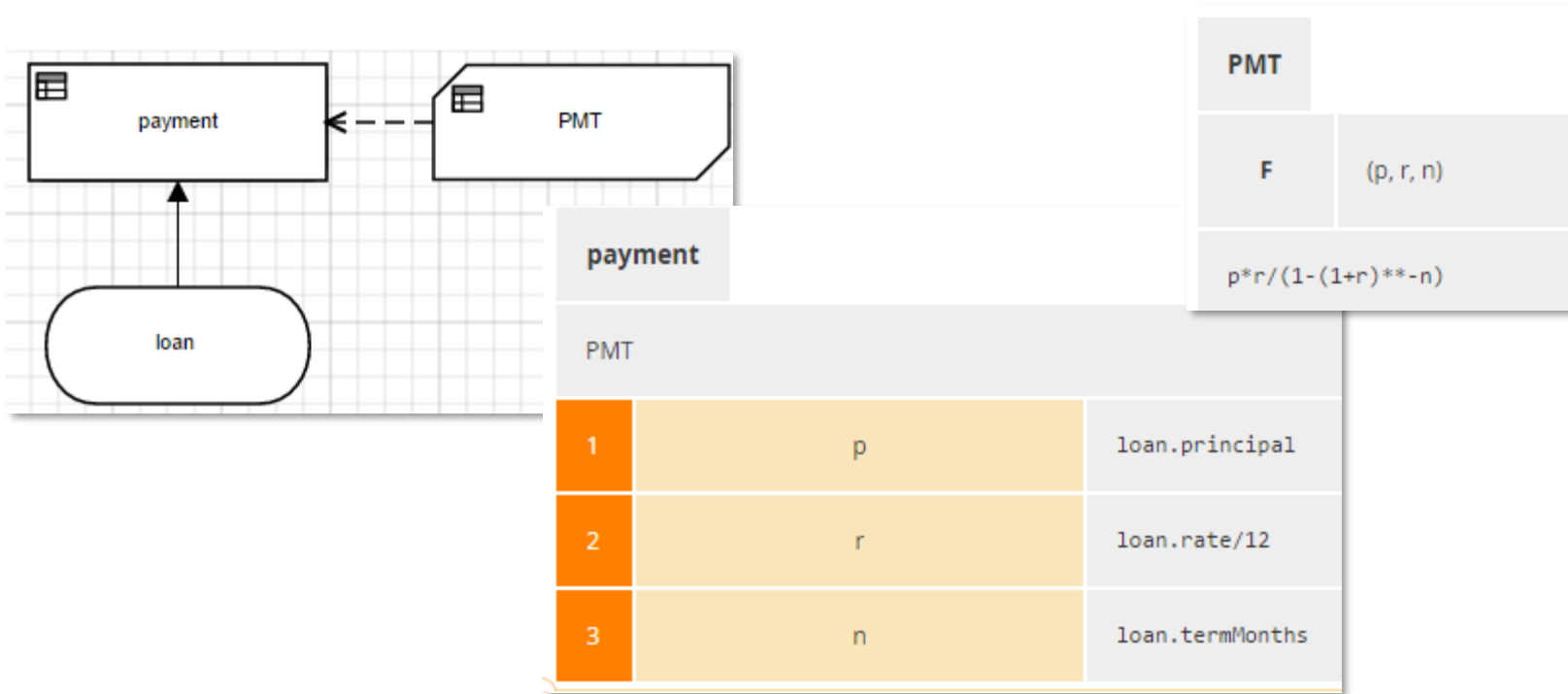
- "A"
- "B"
- "C"
- "D"
- "E"

Create Type Close

Full Boxed Expression Support



Logic Reuse via BKMs

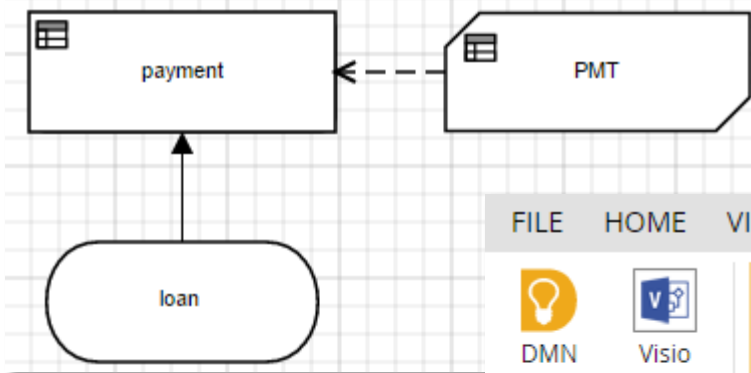


All FEEL Functions and Operators

Table 61: Semantics of list functions

Name(parameters)	Parameter Domain	Description	Example
<code>list contains(list, element)</code>	list, any element of the semantic domain including null	does the <i>list</i> contain the <i>element</i> ?	<code>list contains([1, 2, 3], 2) = true</code>
<code>count(list)</code>	list	return size of <i>list</i>	<code>count([1, 2, 3]) = 3</code>
<code>min(list)</code> <code>min(c₁, ..., c_N), N > 1</code> <code>max(list)</code> <code>max(c₁, ..., c_N), N > 1</code>	(list of) comparable items	return minimum(maximum) item	<code>min([1, 2, 3]) = 1</code> <code>min(1, 2, 3) = 1</code> <code>max([1, 2, 3]) = 3</code> <code>max(1, 2, 3) = 3</code>
<code>sum(list)</code> <code>sum(n₁, ..., n_N), N > 1</code>	(list of) numbers	return sum of numbers	<code>sum([1, 2, 3]) = 6</code> <code>sum(1, 2, 3) = 6</code>
		return arithmetic mean	<code>mean([1, 2, 3]) = 2</code>

Graphical Model to XML



```
1 <?xml version="1.0" encoding="UTF-8"?><semantic:definitions xmlns:semantic="http://www.omg.org/spec/
  xmlns:triso="http://www.trisotech.com/2015/triso/modeling" xmlns="
  http://www.trisotech.com/definitions/_53e21b5c-63d7-4326-ae82-ac75834f21b6" xmlns:feel="http://www.on
  " xmlns:trisoFeed="http://trisotech.com/feed" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ex
  exporterVersion="5.1.6" id="_53e21b5c-63d7-4326-ae82-ac75834f21b6" name="invocation" namespace="
  http://www.trisotech.com/definitions/_53e21b5c-63d7-4326-ae82-ac75834f21b6" triso:logoChoice="Default"
2 <semantic:extensionElements/>
3 <semantic:itemDefinition label="tLoan" name="tLoan">
4 <semantic:itemComponent id="_ac130904-2574-4dd3-a78a-63338ccb6ed7" name="principal">
5 | <semantic:typeRef/feel:number/</semantic:typeRef>
6 </semantic:itemComponent>
7 <semantic:itemComponent id="_29ef2bac-b4dd-42e0-aff8-bfd7d48e10ff" name="rate">
8 | <semantic:tvœRef/feel:number/</semantic:typeRef>
9 </semantic:itemComponent>
10 <semantic:itemComponent id="_29ac3fed-bdfc-4e23-8448-027f1b9c951a" name="termMonths">
11 | <semantic:typeRef/feel:number/</semantic:typeRef>
12 </semantic:itemComponent>
13 <semantic:literalExpression id="literal_2274920-4eb8-4e2d-b435-2e700d081761" name="payment" triso:displayName
14 | <semantic:literalExpression id="literal_d3ac934-fd22-4d97-99f8-d99adffc4f2" name="payment" typeRef="feel:number
15 | <semantic:literalExpression id="literal_8c96a5e7-47e0-464b-9efe-025e48482c03"/>
16 </semantic:literalExpression>
17 <semantic:literalExpression id="literal_4d451f7cf-792e-4a1f-814e-7d9d9120d791"/>
18 </semantic:literalExpression>
19 <semantic:literalExpression id="literal_13f51350-378b-4e40-922e-4c216d3da8aa">
20 </semantic:literalExpression>
21 </semantic:literalExpression>
22 </semantic:literalExpression>
23 </semantic:literalExpression>
24 <semantic:text>PMT</semantic:text>
25 </semantic:literalExpression>
26 <semantic:binding>
27 <semantic:parameter id="_4bcff448-70a3-4185-97bc-c43d3b399e34" name="p"/>
28 <semantic:literalExpression id="_1764113e-a5be-4b13-b36f-1295f4b62d5e">
29 <semantic:text>loan.principal</semantic:text>
30 </semantic:literalExpression>
```

FILE HOME VIEW DMN IMPORT-EXPORT EX

DMN Visio DMN Image Word HTML

Import Documentation

Shapes DMN 1.1 DMN 1.2 (Alpha)

DMN 1.1

Model Validation

The screenshot displays the DMN Modeler interface. The top menu bar includes 'FILE', 'HOME', 'VIEW', 'DMN', and 'IMPORT'. The 'DMN' menu is open, showing options like 'Validate' (highlighted with a red box), 'M&S', 'DT', 'Analysis', 'Method & Style', 'Data Type', and 'Data'. The main workspace shows a decision model diagram with three nodes: 'Loan Approval' at the top, 'Risk Category' on the left, and 'is Affordable' on the right. Arrows point from 'Risk Category' and 'is Affordable' to 'Loan Approval'. Each node has a red 'X' icon in its top right corner, indicating a validation error. The 'Log Viewer' at the bottom shows a table of errors.

Severity	Message	Element	Element Type
✖	Decision is missing a value expression	is Affordable	Decision
✖	Decision is missing a value expression	Risk Category	Decision
✖	Decision is missing a value expression	Loan Approval	Decision
✖	No expression defined for node 'Loan Approval'	Loan Approval	Decision
✖	No expression defined for node 'is Affordable'	is Affordable	Decision
✖	No expression defined for node 'Risk Category'	Risk Category	Decision

Execution... not “Requirements”

Redhat Decision Test

Page 1

Cars

Make-Model

ModelYear

Style

Price

isConvertible

True False

hasDriverAirbags

True False

hasPassengerAirbags

True False

hasSideAirbags

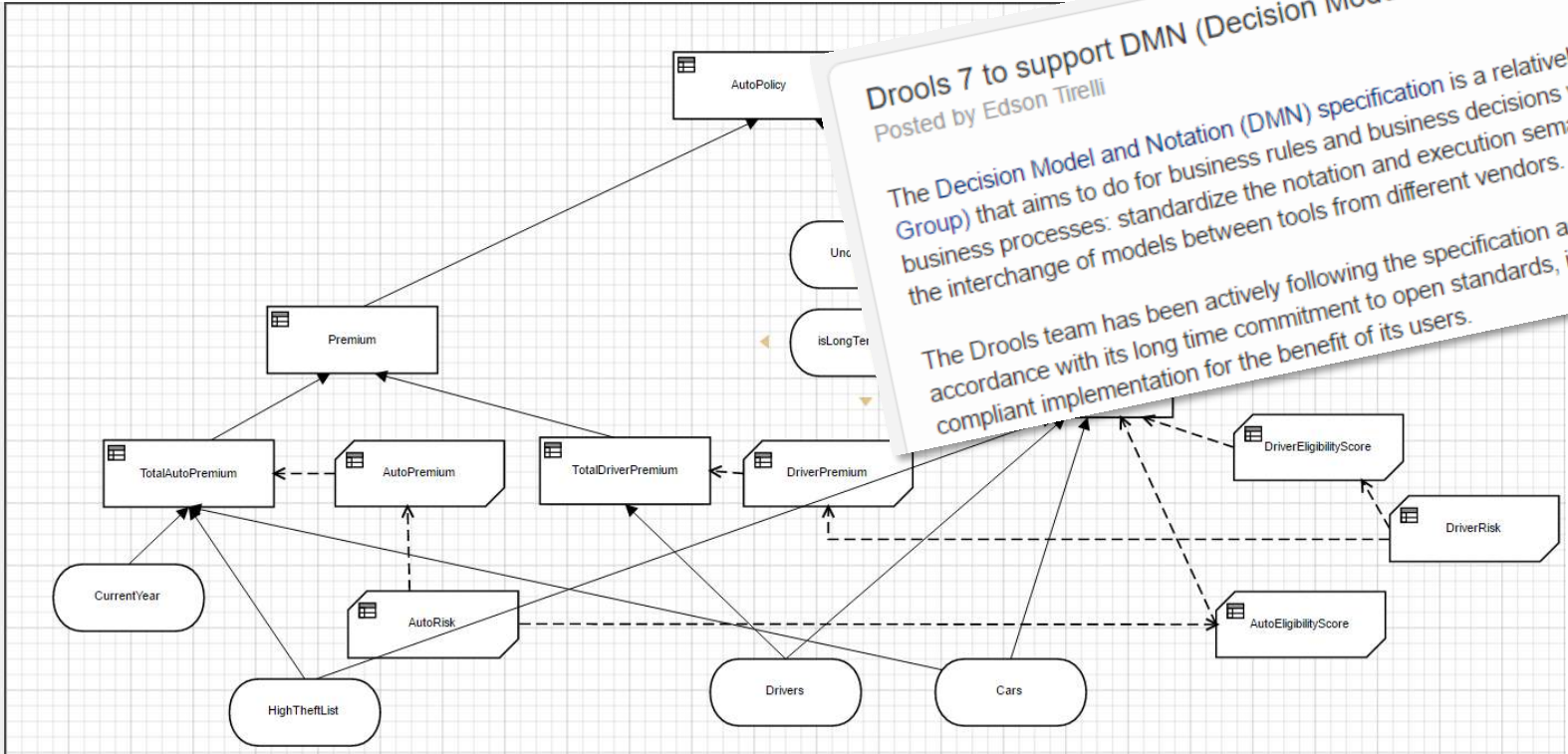
True False

hasRollbar

True False

hasAlarm

True False



Drools 7 to support DMN (Decision Model and Notation)
Posted by Edson Tirelli

The Decision Model and Notation (DMN) specification is a relatively new standard (BPM Group) that aims to do for business rules and business execution semantics to enable the interchange of models between tools from different vendors.

The Drools team has been actively following the specification and the direction of the standard in accordance with its long time commitment to open standards, it is now time for a compliant implementation for the benefit of its users.

Flexible Execution Architecture



- Test logic in the tool
- ... embed it in an app
- ... or deploy it as a cloud service

Redhat Decision Test

Outputs

Reserves Months

78.54796517032221

Credit Risk Category

C

DTI

0.23277289454894703

Income Risk Category

Medium

Collateral Risk Category

Medium

LTV

0.7722007722007722

NonSalaryPct

6.666666666666667

Income Risk Score

70

Loan Payment

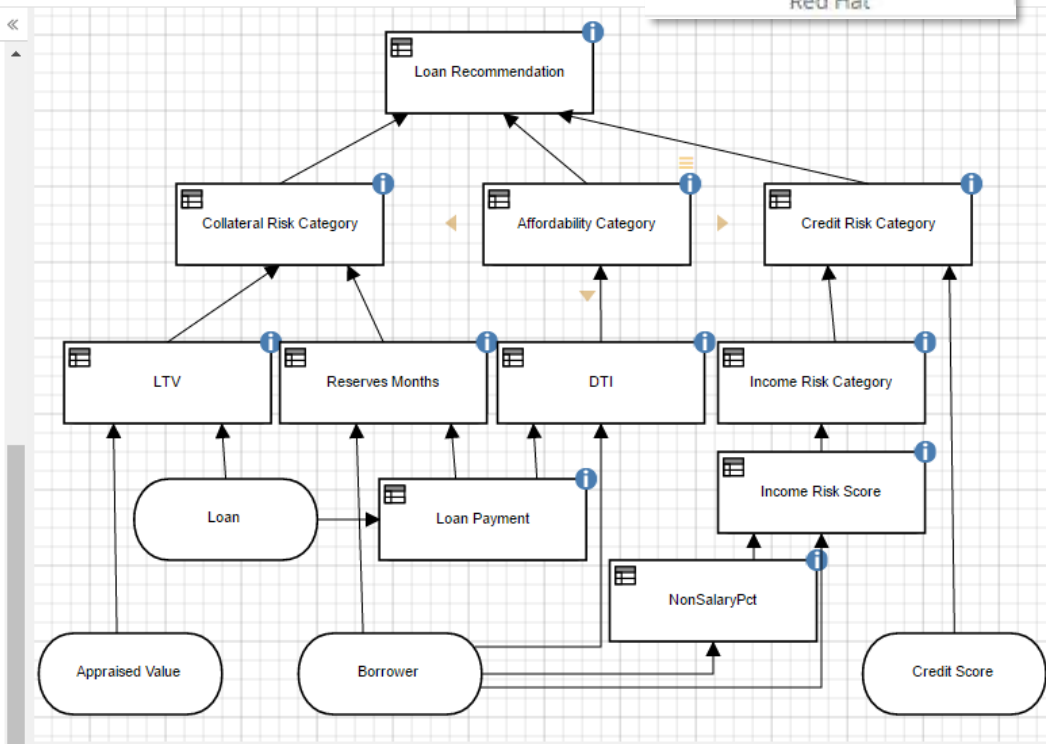
954.830590930919

Affordability Category

Affordable

Loan Recommendation

Decline



RedHat DMN Runtime

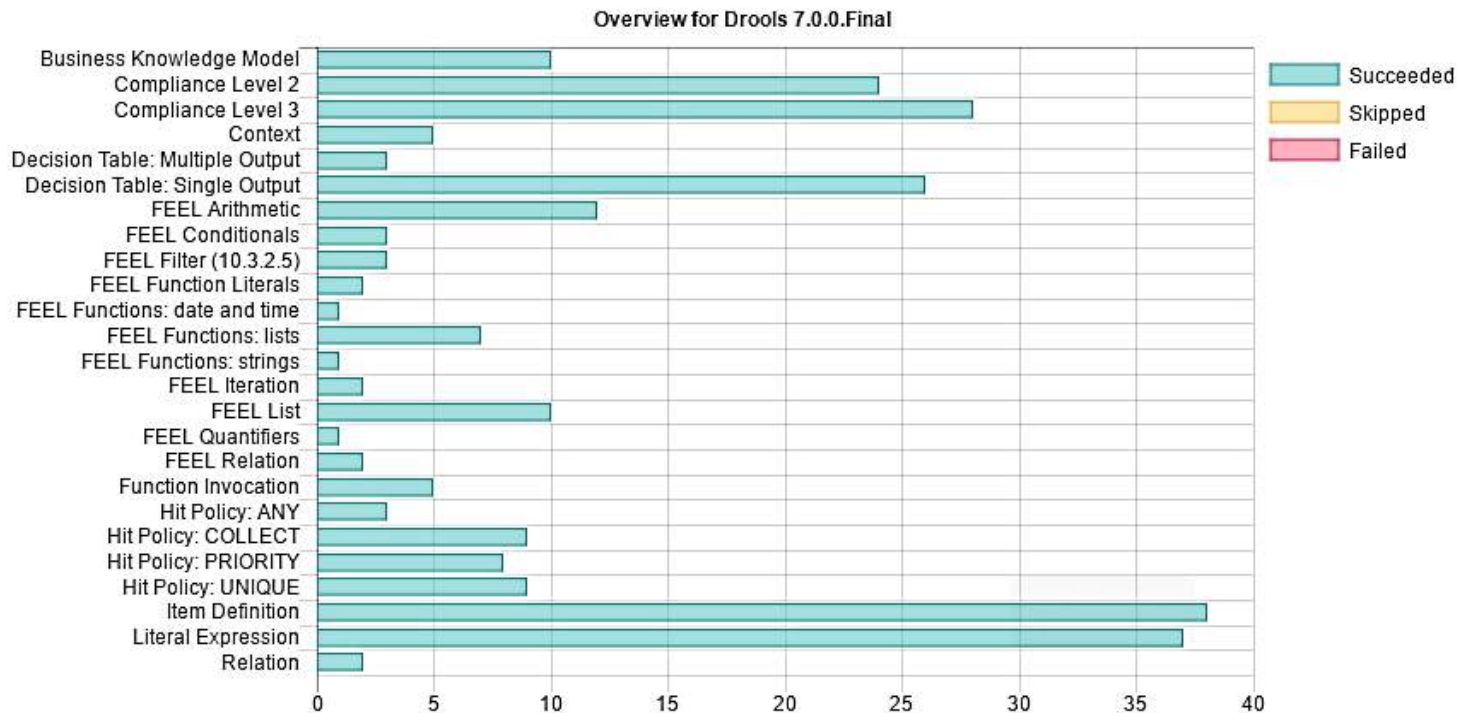
- Open source extensions to Drools Rule Engine

The screenshot shows the GitHub repository page for `droolsjbpm / kie-dmn`. At the top, there are navigation links for `Code`, `Pull requests` (0), `Projects` (0), `Wiki`, `Pulse`, and `Graphs`. On the right, there are buttons for `Watch` (43), `Star` (5), and `Fork` (17). Below the navigation, the repository is described as "KIE DMN implementation". A summary bar shows `243 commits`, `2 branches`, `4 releases`, `6 contributors`, and `Apache-2.0` license. Below this, there are buttons for `Branch: master`, `New pull request`, `Create new file`, `Upload files`, `Find file`, and `Clone or download`. The commit history shows a recent commit by `mariofusco` with `etirelli` [DROOLS-1454] `osgify kie-dmn (#49)` committed 15 minutes ago. Below the commit, there is a list of files:

File Name	Commit Message	Time
<code>kie-dmn-api</code>	[DROOLS-1454] <code>osgify kie-dmn (#49)</code>	15 minutes ago
<code>kie-dmn-backend</code>	[DROOLS-1454] <code>osgify kie-dmn (#49)</code>	15 minutes ago
<code>kie-dmn-core</code>	[DROOLS-1454] <code>osgify kie-dmn (#49)</code>	15 minutes ago
<code>kie-dmn-feel</code>	[DROOLS-1454] <code>osgify kie-dmn (#49)</code>	15 minutes ago
<code>kie-dmn-model</code>	[DROOLS-1454] <code>osgify kie-dmn (#49)</code>	15 minutes ago

Only Solution Passing TCK

Tests Overview



Trisotech DMN QuickStart



METHOD *& Style*

Trisotech

in collaboration with

Bruce Silver Associates

Allegiance Advisory

Red Hat

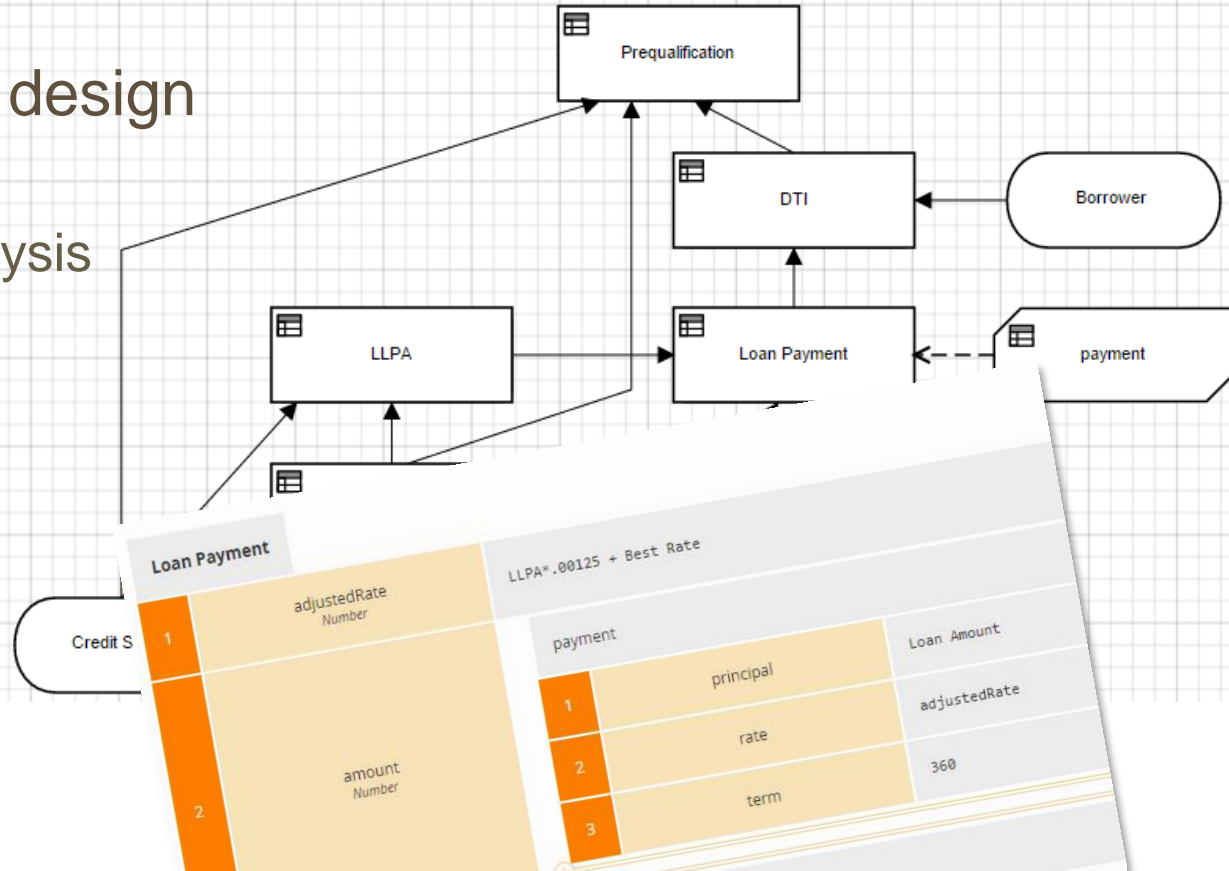
is offering the definitive Decision



redhat.

About the Demo...

- Business-oriented design
 - Ease of use
 - Decision Table Analysis
 - Completeness
 - Correct hit policy
 - Normalization
 - Boxed expressions



About the Demo...

- Fully executable
 - Model validation
 - FEEL parsing and execution
 - Execution in the tool
 - Cloud deployment

