Decision Model and Notation 101

Overview and Demo

Edson Tirelli, Red Hat
Sr Principal Software Engineer, Drools Project Lead

Denis Gagné, Trisotech
CTO & CEO

May 2\textsuperscript{nd}, 2017
DECISION MODEL AND NOTATION 101

Agenda

- Brief History of DMN
- Components of a DMN solution
- Conformance levels
- DMN – The big picture
- Demo: an end-to-end solution for DMN
WHAT IS DMN?

“DMN, which stands for Decision Model and Notation, is a relatively new standard managed by OMG, the organization behind BPMN. It is trying to do for Business Decision Management what BPMN did for Business Process Management a decade ago: empower the business to take charge of the logic that drives its operations, through a vendor-independent diagramming language.” (Bruce Silver, http://methodandstyle.com/what-is-dmn/)

- The spec defines among other things:
  - a graphical language for business decision modelling
  - a standard notation for decision tables
  - an expression language called FEEL (which stands for Friendly Enough Expression Language)
  - a metamodel and an interchange format (XML)
  - 3 conformance levels for tool implementations
- More details: http://www.omg.org/spec/DMN/1.1/
Versions released:

- 1.0 – September 2015
  - Introduced DMN
  - Had several problems (bugs) on both the XML schema definition and the FEEL language that prevented its full implementation.
- 1.1 – June 2016
  - RTF group worked to fix most of the problems from the version 1.0
  - It is the first version of the spec that is fully implementable (with a few minor issues)
DMN GRAPHICAL LANGUAGE – EXAMPLE
DMN GRAPHICAL LANGUAGE – EXAMPLE

[Diagram of DMN graphical language example]

Input Node
DMN GRAPHICAL LANGUAGE – EXAMPLE

Input Node
Decision Node

Eligibility

<table>
<thead>
<tr>
<th>Eligibility Rules</th>
<th>Application</th>
<th>Applicant.Employment.Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employment Status</td>
<td>Application</td>
<td>Applicant.Employment.Status</td>
</tr>
<tr>
<td>Country</td>
<td>Application</td>
<td>Applicant.Country</td>
</tr>
<tr>
<td>Age</td>
<td>Application</td>
<td>Applicant.Age</td>
</tr>
</tbody>
</table>
DMN GRAPHICAL LANGUAGE – EXAMPLE

- Decision Node
- Input Node
- Business Knowledge Model

[Diagram showing nodes and arrows to represent the DMN graphical language example.]
DMN GRAPHICAL LANGUAGE – EXAMPLE

Decision Node
Input Node
Business Knowledge Model
## DECISION TABLES

### Pre-bureau risk category table

<table>
<thead>
<tr>
<th>Rule number</th>
<th>Conditions</th>
<th>Application Risk Score</th>
<th>Result column</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;100</td>
<td>&quot;HIGH&quot;</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>[100..120]</td>
<td>&quot;MEDIUM&quot;</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>[120..130]</td>
<td>&quot;LOW&quot;</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&gt;130</td>
<td>&quot;VERY LOW&quot;</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>&lt;80</td>
<td>&quot;DECLINE&quot;</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>[80..90]</td>
<td>&quot;HIGH&quot;</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>[90..110]</td>
<td>&quot;MEDIUM&quot;</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>&gt;110</td>
<td>&quot;LOW&quot;</td>
<td></td>
</tr>
</tbody>
</table>
WHAT IS FEEL?

FEEL, which stands for *Friendly Enough Expression Language*, is a new expression language defined by the DMN specification. It aims to bridge the gap between decision modelling and execution, by assigning semantics to the decision model constructs.

- Aims to be usable by any user “capable of using Excel macros”
- Is used both in DRD (Decision Requirement Diagrams) as well as Decision Tables
- Two levels of compliance:
  - Level 2: requires only the features of S-FEEL (a simplified FEEL defined in chapter 9 of the spec)
  - Level 3: require full implementation of FEEL (as defined in chapter 10 of the spec)
### FEEL Example on a Boxed Expression

<table>
<thead>
<tr>
<th>BKM name</th>
<th>Expressions</th>
</tr>
</thead>
</table>
| **Installment calculation** | Monthly Fee: if Product Type = "STANDARD LOAN" then 20.00  
else if Product Type = "SPECIAL LOAN" then 25.00  
else null |
|                   | Monthly Repayment: (Amount * Rate/12) / (1 - (1 + Rate/12)**-Term) |
|                   | Monthly Repayment + Monthly Fee |

#### Parameters
- ProductType
- Rate
- Term
- Amount

#### Context entries ("Local variables")
- Monthly Fee
- Monthly Repayment

#### BKM result
CONFORMANCE LEVELS

The spec defines 3 incremental conformance levels for implementations:

• Conformance Level 1:
  • Requires support for authoring of Decision Requirements Diagram, Decision Logic and Decision Tables
  • This conformance level is basically “documentation only”, no execution required

• Conformance Level 2:
  • Everything from conformance level 1, plus support for the S-FEEL (simplified FEEL) expression language
  • Requires execution, and requires the logic modelled in CL1 to be expressed in S-FEEL

• Conformance Level 3:
  • Everything from conformance level 2, plus support for the full FEEL language
  • This includes additional modeling elements like “boxed expressions”
DMN - THE BIG PICTURE
DEMO: An end-to-end DMN solution
AN INTEGRATED END-TO-END SOLUTION

- Modeling
- Authoring

- Methodology
- Training

- Runtime
- Validation

Decision Model and Notation (DMN)
MORE INFORMATION

- DMN Quick Start program: www.trisotech.com/DMNQuickStart
- Drools project: www.drools.org
- Red Hat BRMS: https://access.redhat.com/products/red-hat-jboss-brms/
- DMN specification: http://www.omg.org/spec/DMN/1.1/
- DMN book and training: http://methodandstyle.com/
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.