Document Generation with M2Doc

Yvan Lussaud
Why documents?

• Written Documents are still necessary
  – Transmit information to people who will not use the modeling workbench directly
  – Fulfil certain obligations

• The Model should be the reference
  – Up to date documentation
Actors Involved

CASID/DGSIC
(French Department of Defense)
Our Solution

M2Doc
Our Solution

Model to documentation
Basic Principles

Model

Template

M2Doc

Document
Basic Principles

Model → M2Doc → Document

Template
Basic Principles
Roles Involved

**Metamodel Expert**
- Creates, tests and debugs reusable AQL services
- Registers services

**Templatist**
- Creates M2Doc templates
- Uses AQL services
- Tests and debugs templates

**End-User**
- Edit models
- Produces documents by running M2Doc templates
- Edit ‘userdoc’ in generated documents

These are just roles

Can be 1, 2, or 3 persons depending on complexity and organization
Writing Templates

- Plain MS Word™ document (.docx files)
- Use Alt+F9 to activate fields
- Use Ctrl+F9 to insert fields
- M2Doc syntax is very simple
\{ m:XXX \ldots \}
Simple query

\{ m:conf.name \} is located in Ludwigsburg

EclipseCon Europe is located in Ludwigsburg
Loop

{ m:for track | conf.tracks}

• Track: {m:track.name}

{ m:endfor }

• Track: IoT Day
• Track: Eclipse Technologies
• Track: OSGi
## Loop

\[
\{ \text{m:for track} \mid \text{conf.tracks}\} \\
\{\text{m:track.name}\} \quad \{\text{m:track.description}\} \\
\{\text{m:endfor}\} \\
\]

<table>
<thead>
<tr>
<th>IoT Day</th>
<th>Internet of Things</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eclipse Technologies</td>
<td>Eclipse-related technologies and eclipse community</td>
</tr>
<tr>
<td>OSGi</td>
<td>Talks about OSGi</td>
</tr>
</tbody>
</table>
Conditional

{ m:if conf.name.contains('Eclipse') }

... { m:else if ...

... { m:else }

... { m:endif }
User Doc

{m:userdoc <queryID>}

...

{m:enduserdoc}

• Incremental generation
Bookmark and link

{m:bookmark <queryID>}

…

{m:endbookmark}

{m:link <queryID> <queryText>}
Let

\[
\{m:\text{let cName} = \text{conf.name}\}
\]

\[
\ldots
\]

\[
\{m:\text{cName}\} \text{ is located in Ludwisburg}
\]

\[
\ldots
\]

\[
\{m:\text{endlet}\}
\]

- Define variables
Using Services

{m:if
  conf.tracks.talks
  ->select(t | t.name.contains('Sirius'))
  ->notEmpty()}

This document is the property of Obeo and cannot be transmitted without prior written agreement.
Using Services

{ m:if conf.hasTalkAbout('Sirius') }

public boolean hasTalkAbout(Conference, String)
Using Services

- Services are Java methods
- Facilitate creation and maintenance of templates
- Can be reused
- Can insert images/tables/formatted text/…
Validation

\{ \texttt{m:conf.} \} \text{ is located in Ludwisburg}

\{ \texttt{m:conf.} \} \textless---Expression "conf." is invalid: missing feature access or service call \text{ is located in Ludwisburg}
M2Doc + Sirius = ♥

• Diagrams and Tables can be inserted
• The existence of a diagram/table can be checked
  – Dedicated services are provided
    • isRepresentation(DescriptionName)
    • asImageByRepresentation(DescriptionName)
    • asTableByRepresentation(DescriptionName)
Architecture Overview

- M2Doc UI
- M2Doc IDE
- M2Doc core
- AQL
- Apache POI
- EMF

M2Doc Sirius Integration

Sirius
Architecture Overview

M2Doc UI
M2Doc IDE
M2Doc core
AQL  Apache POI  EMF

M2Doc your product Integration
Your product
Show me the code!

https://github.com/ObeoNetwork/M2Doc/
M2Doc – Roadmap

• V1.0.0 has been released
  – First maintained version
  – Fully documented

• Next
  – Gather feedback
  – Keep improving + features from feedback
  – Improve Apache POI integration
Keypoints

• Produces MS Word™ documents from EMF-compliant Models
• Can include Sirius Diagrams & Tables
• Easy to use: simple syntax, tooling helps
• Has Useful Error Reporting
• Supports Incremental Generation
Evaluate the Sessions

Sign in and vote at eclipsecon.org

- 1  0  + 1