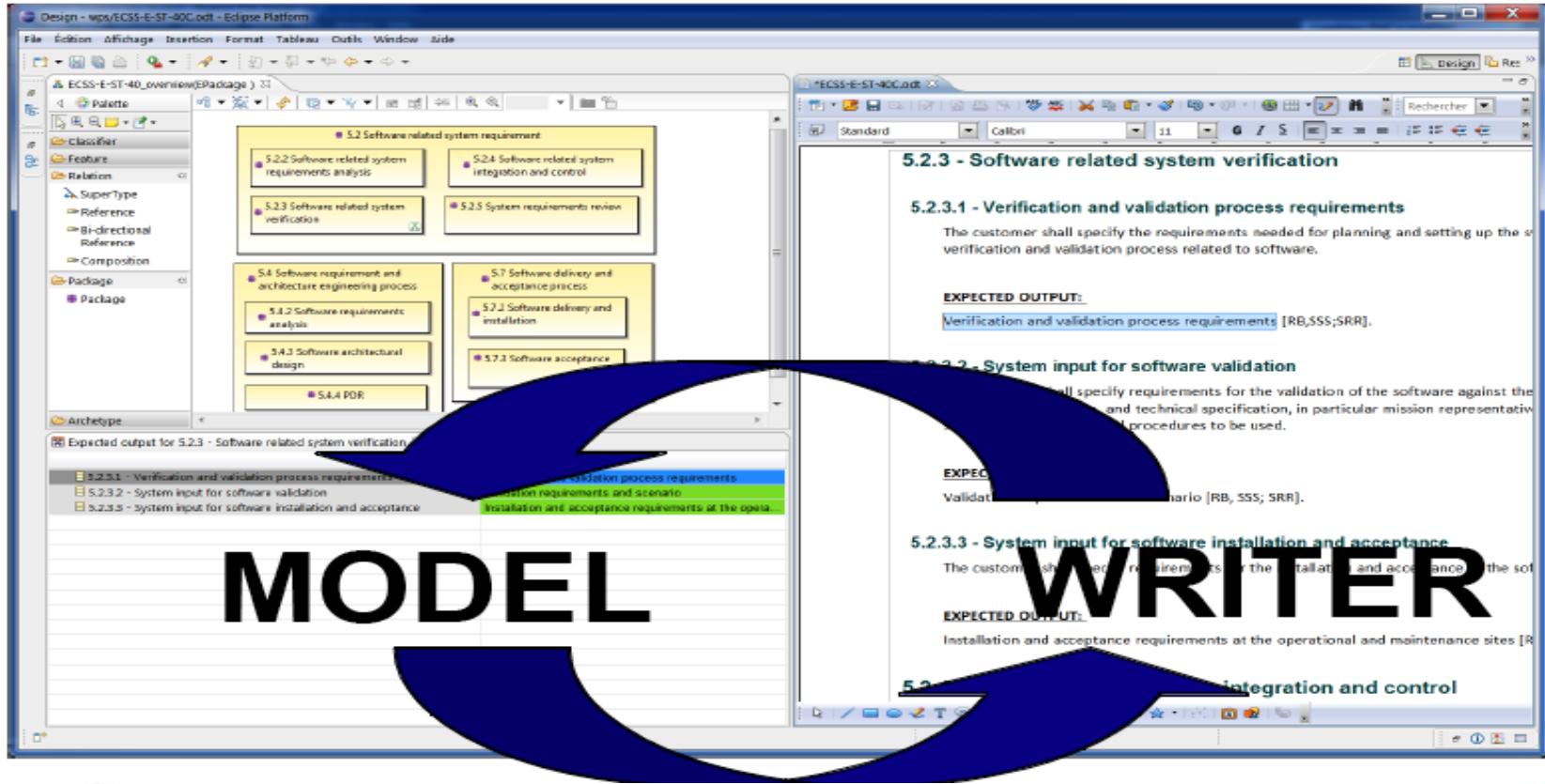


ModelWriter

Text & Model-Synchronized Document Engineering Platform



Project Leader: Ferhat Erata (ferhat@computer.org)

Project Email: project@modelwriter.eu

Agenda [9:00 - 12:30]

- Introduction (10 min) [09:00 - 09:10]
- Overview of the project (15 min) [09:00 - 09:15]
- Use-cases and exploitation prospects (45 min) [09:15 - 10:00]
- Progress Status per WP (60 min) [10:00 - 11:00]
- Demonstrations (30 min): [11:00 - 11:30]
- Summary (achievements, expectations and exploitations)
- Reviewers Private Section (30 min) [11:45 - 12:15]
- Feedback Session (15min) [12:15 - 12:30]
- Lunch (30 min) [12:30 - 13:00]

Participants

UNIT (TUR):

- Ferhat Erata
- Dr. Moharram Challenger

LORIA (FRA):

- Prof. Claire Gardent,
- Prof. Samuel Cruz-Lara
- Dr. Mariem Mahfoudh

AIRBUS (FRA):

- Dr. Anne Monceaux

HAVELSAN (TUR):

- Dr. Eray Tuzun

OBEO (FRA):

- Yvan Lussaud

KOÇSISTEM (TUR):

- Prof. Geylani Kardas
- Hale Gezgen

MANTIS (TUR):

- Prof. Erhan Mengüsoğlu

HISBIM (TUR):

- Ersan Gürdoğan
- Taskin Kızıl

Overview of the Project



ModelWriter

Text & Model-Synchronized Document
Engineering Platform

The project envisions an integrated authoring environment called "ModelWriter" for Technical Authors (such as Software or Systems Engineers etc.) which will combine a Semantic Word Processor (= the "Writer" part) and a Knowledge Capture Tool (= the "Model" part).

Project information

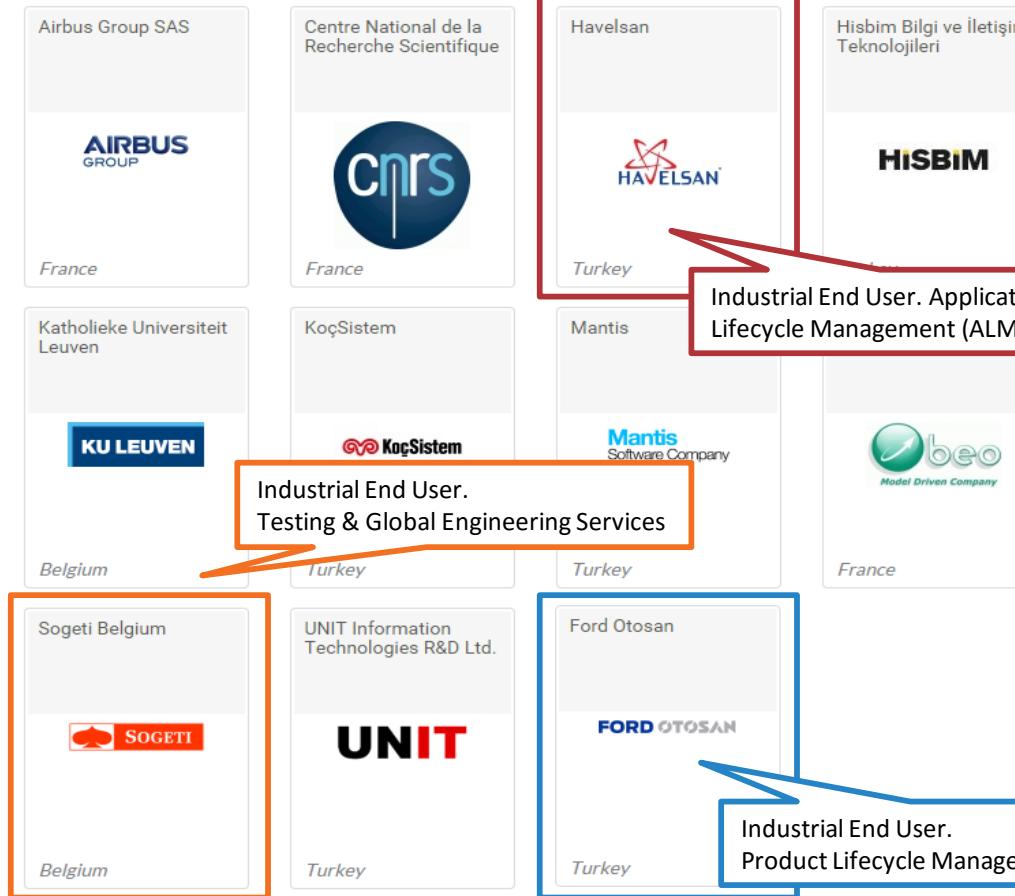
Project name	13028 ModelWriter
Status	Running
Period	Oct 2014 - Sep 2017
Call	ITEA 2 Call 8
Challenge	Knowledge-based society
Website	www.modelwriter.eu
Partners	10
Countries	Belgium France Turkey

Project leader



Name
Ferhat Erata
Organisation
UNIT Information Technologies R&D Ltd.
Country
Turkey
Project involvement
13028 ModelWriter

Project partners



Project documents

Project publications

- [ITEA Annual report 2013 published online](#)
- [ModelWriter Posters Co-summit 2015](#)
- [ModelWriter Project Leaflet](#)

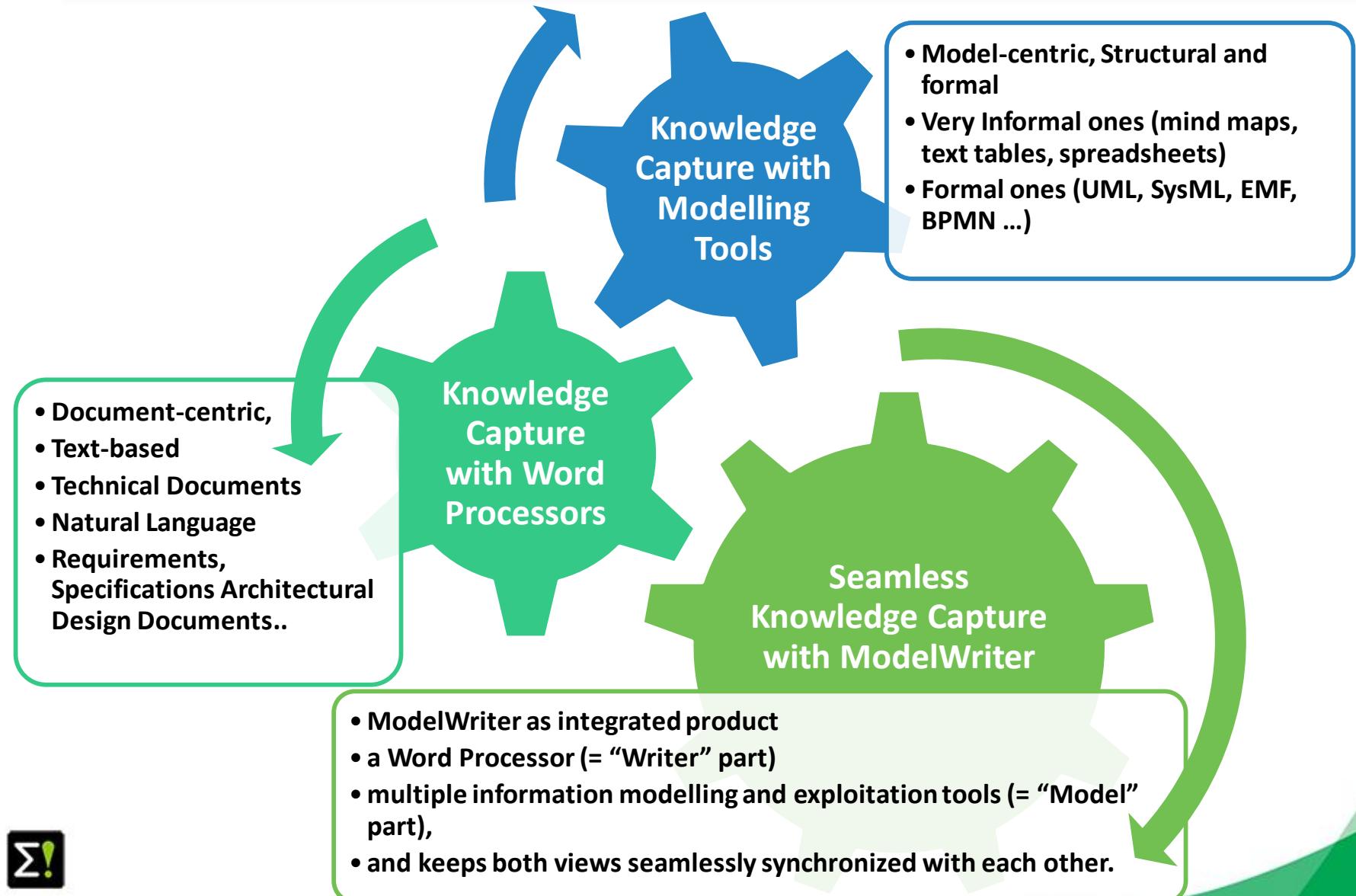
Resource Allocation: 68,71 person year

Project Duration: 36 months

Planned Budget: 5,543,000 Euro

Start and Finish Date: 01 Oct 2014 – 30 Sep 2017

Open-Source Software Platform to be submitted to Eclipse Foundation



ModelWriter

bi-directional Knowledge Capture tool



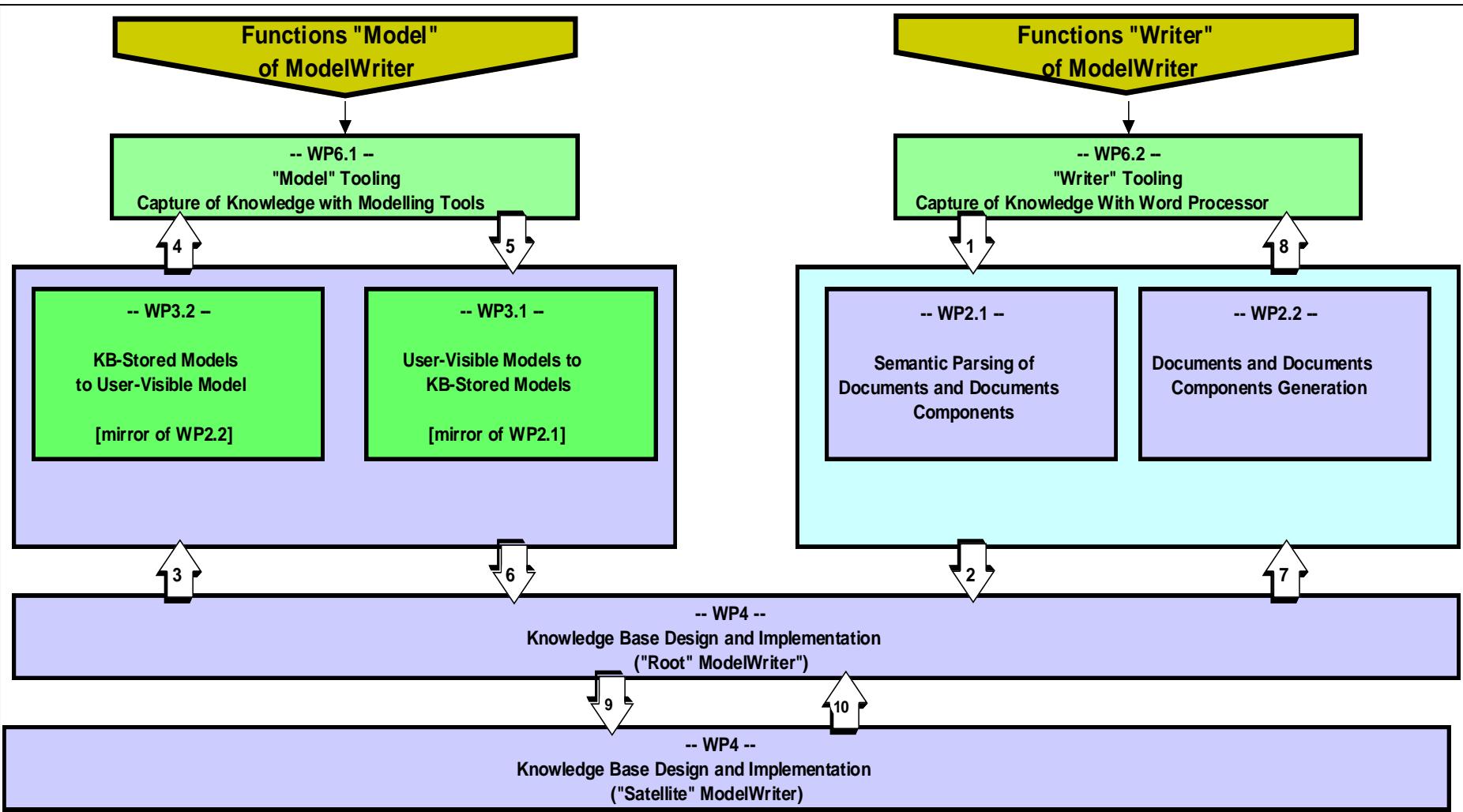
The screenshot displays the ModelWriter application interface. On the left, a model editor window titled "Design - mps/ECSS-E-ST-40C.otp - Eclipse Platform" shows a hierarchical structure of requirements. The main tree view includes nodes like "5.2 Software related system requirement", "5.2.2 Software related system requirements analysis", "5.2.4 Software related system integration and control", "5.2.3 Software related system verification", "5.2.5 System requirements review", "5.4 Software requirement and architecture engineering process", "5.4.2 Software requirements analysis", "5.4.3 Software architectural design", and "5.4.4 PDR". Below this, a detailed view of "5.2.3 Software related system verification" is shown, containing sections for "Verification and validation process requirements", "System input for software validation", and "System input for software installation and acceptance". A large blue arrow points from the "5.2.3" node in the model editor towards the corresponding section in the document viewer. On the right, a document viewer window titled "ECSS-E-ST-40C.otp" is open, displaying the same section with headings like "5.2.3 - Software related system verification", "5.2.3.1 - Verification and validation process requirements", and "5.2.3.2 - System input for software validation". The word "MODEL" is overlaid in large black letters across the bottom left of the model editor, and "WRITER" is overlaid in large black letters across the bottom right of the document viewer.

Semantic Word Processor (Text-Based Knowledge Extractor)

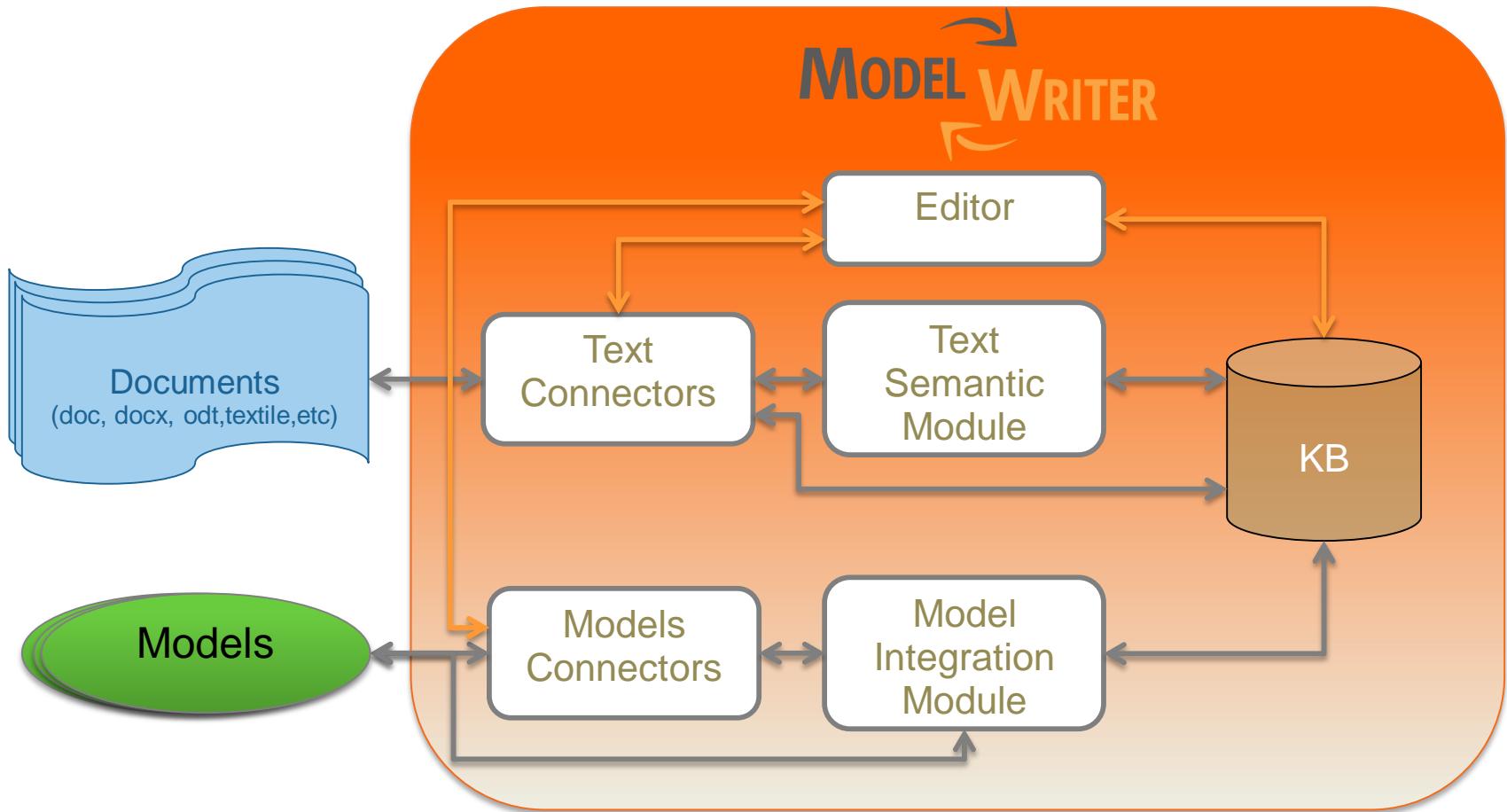
Understands the various textual parts of a document expressed in Natural Language

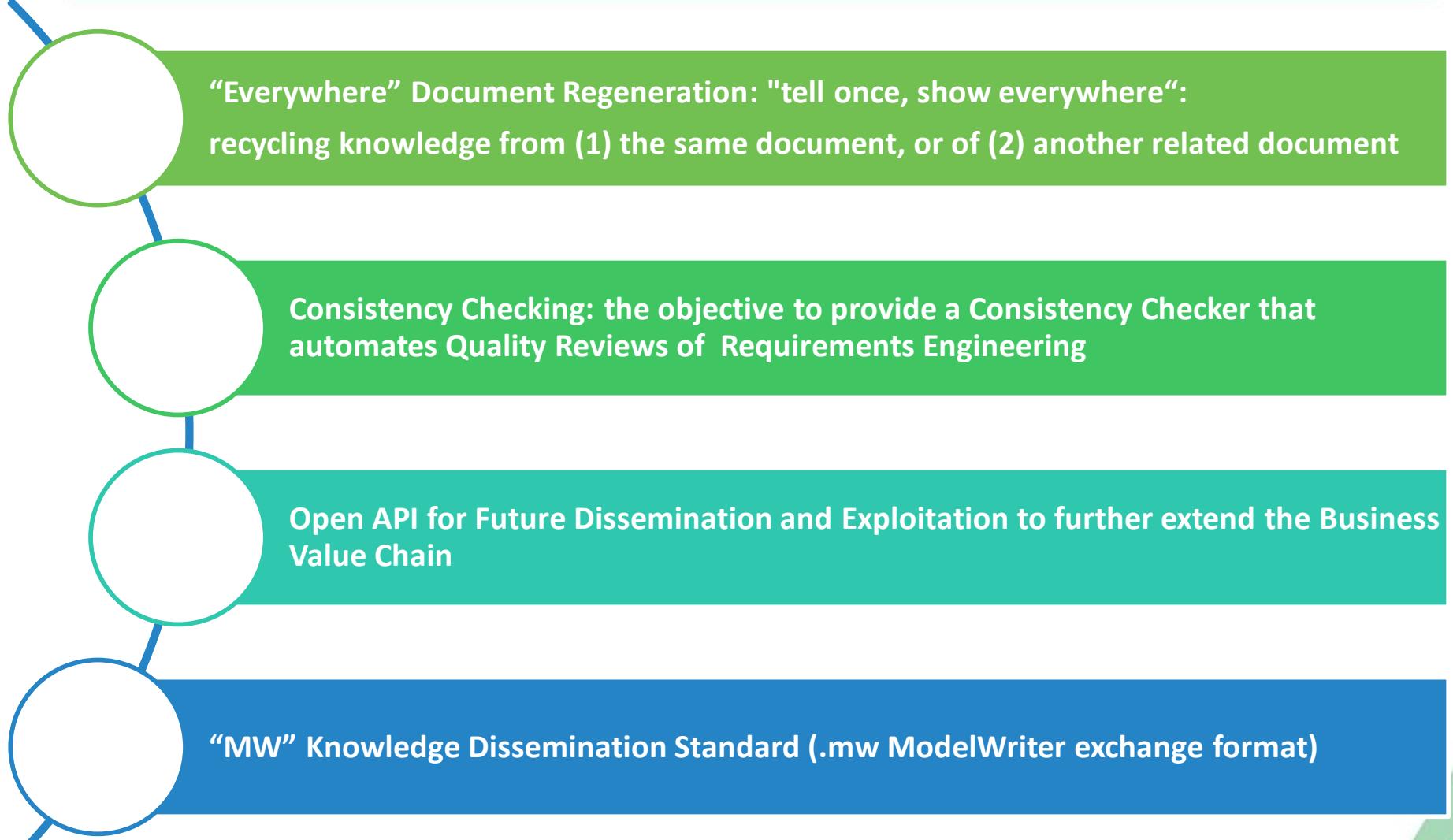
Reveals concepts and relationships between them (“Model”-part)

Consistency & Completeness Checking



ModelWriter Architecture





“Everywhere” Document Regeneration: “tell once, show everywhere”: recycling knowledge from (1) the same document, or of (2) another related document

Consistency Checking: the objective to provide a Consistency Checker that automates Quality Reviews of Requirements Engineering

Open API for Future Dissemination and Exploitation to further extend the Business Value Chain

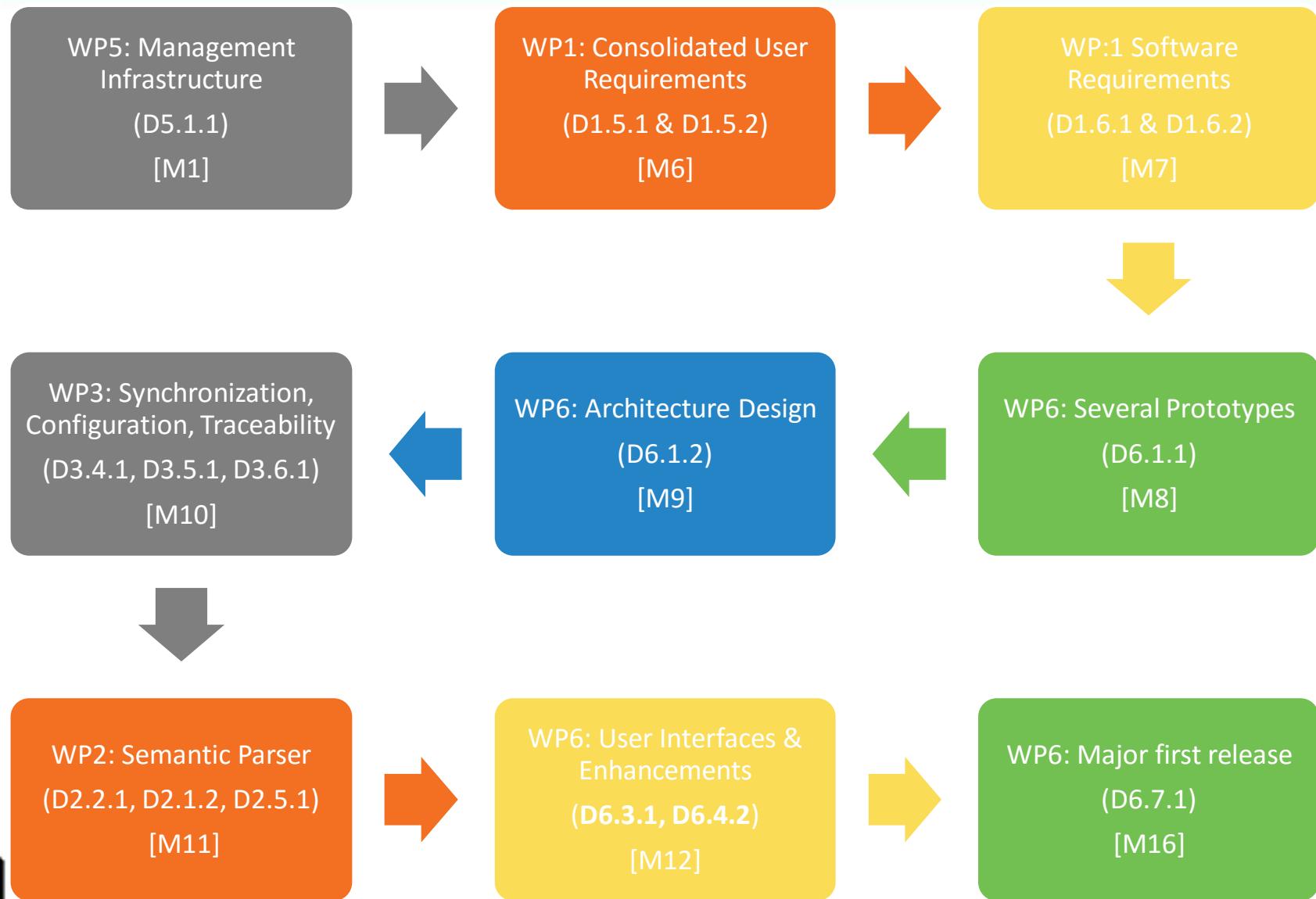
“MW” Knowledge Dissemination Standard (.mw ModelWriter exchange format)

WP1 Industrial Use Cases and Requirements (AIRBUS)

WP2 (LORIA)	WP3 (UNIT)	WP4 (MANTIS)	WP6 (OBEO)
<ul style="list-style-type: none">• Semantic Parser• Document Generation• bi-directional transformation between text and formal knowledge representation	<ul style="list-style-type: none">• Bi-directional synchronization mechanism between texts and models• Configuration & Traceability Components• Consistency checker plug-in for consistency	<ul style="list-style-type: none">• A federated Knowledge Base and its API• Synchronization mechanism between texts/models & knowledge base	<ul style="list-style-type: none">• A complete “ModelWriter” tool integrating of all these in a consistent way• User Interfaces

WP5 Project Management (UNIT)

WP7 Standardization, Dissemination and Exploitation (OBEO)



ModelWriter Workshops in the First Year



<https://github.com/modelwriter/workshops>

Initial Architectural Design, Industrial Use Cases, Technical WP discussions

in Istanbul, Turkey (Nov 08, 2014)

Collaboration Infrastructure

The 1st International ModelWriter Workshop in Izmir, Turkey (Jan 15-17, 2015)

Exploitation: Havelsan's participation

The 1st International Eureka Project Exhibition in Berlin, Germany (Mar 10-11, 2015)

Consolidated User Requirements & Review

The 2nd International ModelWriter Workshop in Brussels, Belgium (Apr 30, 2015)

Software Requirements Review & Architecture

The 3rd International ModelWriter Workshop in Toulouse, France (Jun 22-23, 2015)

Rehearsal & Review

The 4th International ModelWriter Workshop in Brussels, Belgium (Sep 23-24, 2015)

Rehearsal & Review

The 5th International ModelWriter Workshop in Ludwigsburg, Germany (Nov 2-5, 2015)

ModelWriter Business Strategy

Industrialization Triangle

Open Source Campaign

Open Call for Industrial User Stories

- *Shape the future ModelWriter*
- *Early adaptation of the technology*
- *Long Term Support*

ModelWriter Open Source Campaign

<https://github.com/modelwriter>



Screenshot of the GitHub repository page for the ITEA2-ModelWriter Project.

ITEA2-ModelWriter Project
Text & Model-Synchronized Document Engineering Platform
Europa | <http://www.modelwriter.eu> | project@modelwriter.eu

Repositories (highlighted) | **People** 25 | **Teams** 27 | **Settings**

Filters | | **+ New repository**

WP5
Work Package 5 - Project Management (UNIT)
Updated 2 hours ago

WP3
Work Package 3 - Model to/from Knowledge Base (UNIT)
Updated 3 days ago

Workshops
Repository dedicated to workshops
Updated 3 days ago

Deliverables
Project Monitoring, Tracking, Communication Management and Infrastructure
Updated 5 days ago

People 25 >

A grid of 25 user profile pictures. Below the grid is a button labeled "Invite someone".



ModelWriter Requirements & User Stories

<https://waffle.io/modelwriter/requirements>



modelwriter/requirements

Add Issue

Filter Board

User Stories 13

Confirmed 3

In Progress 0

Done 2

14 The TRAM will be able to compose transformations.
Software Requirements Document (SRD)
WP3 - Model to/from Knowledge Base

13 The TRAM validates such parameters and also the input models before a transformation takes place.
Software Requirements Document (SRD)
WP3 - Model to/from Knowledge Base

12 A mechanism is needed to register the available transformations in ModelWriter
Software Requirements Document (SRD)
WP3 - Model to/from Knowledge Base

11 M2M Transformation Framework must synchronize the output models after its input models or configurations have been modified.
Software Requirements Document (SRD)
WP3 - Model to/from Knowledge Base

10 M2M Transformation Framework must keep traces between transformed models and its source models.

17 The system must support an open requirement authoring tools (such as RMF)
UC-TR-03 Generation and management of feature models
User Requirements Document (URD)

16 The system should support ReqIF standard.
UC-TR-03 Generation and management of feature models
User Requirements Document (URD)

15 BPMN 2.X shall be supported as the Business Process Modeling Language in the tool.
UC-TR-05 Synchronous Business Process Design with Use Cases
User Requirements Document (URD)

In Progress

Let others know you're working on an issue by dragging it to In Progress.

19 Fe

18 readme folder added

This screenshot shows a Waffle board interface for tracking requirements and user stories. The board is divided into four columns: User Stories (13 items), Confirmed (3 items), In Progress (0 items), and Done (2 items). Each item is represented by a card containing a summary, source documents, and a comment section. The 'In Progress' column contains a placeholder message encouraging users to drag issues there. The 'Done' column has two completed items, one of which is a 'readme folder added'.

UC-TR-03

IBM Rational Doors Plug-in and Eclipse Requirement Frameworks Sub-component which supports OMG's ReqIF Standard.

Sync between Requirement Documentation and Feature Models

KoçSistem, UNIT,
HAVELSAN, AIRBUS

UC-TR-04

Demonstrate that the ModelWriter concept can be usefully applied on ALM

Integration with Application Lifecycle Management (ALM) Tools

UNIT, HAVELSAN
KoçSistem

UC-TR-05

Integration with Product Lifecycle Management (PLM), Technical Engineering Documents

UNIT, KoçSistem,
Ford Otosan

Integration with Siemens Workspace

UC-FR-02

Help determining the impacts of a change in the Enterprise Architecture

Enterprise Architecture

CEISAR - Center of Excellence in Enterprise Architecture

**OBEO, TOTAL,
AIR France, AXA**

UC-FR-03

This use case will explore the use of ModelWriter concept to structure and manage a repository of system installation rules.

Synchronization of regulation documentation with a design rule repository

**OBEO, AIRBUS,
LORIA**

UC-FR-04

Focus on building synchronized links between regulation documents in natural language and rules (as models).

Production of a context specific design document

**OBEO, AIRBUS,
LORIA**

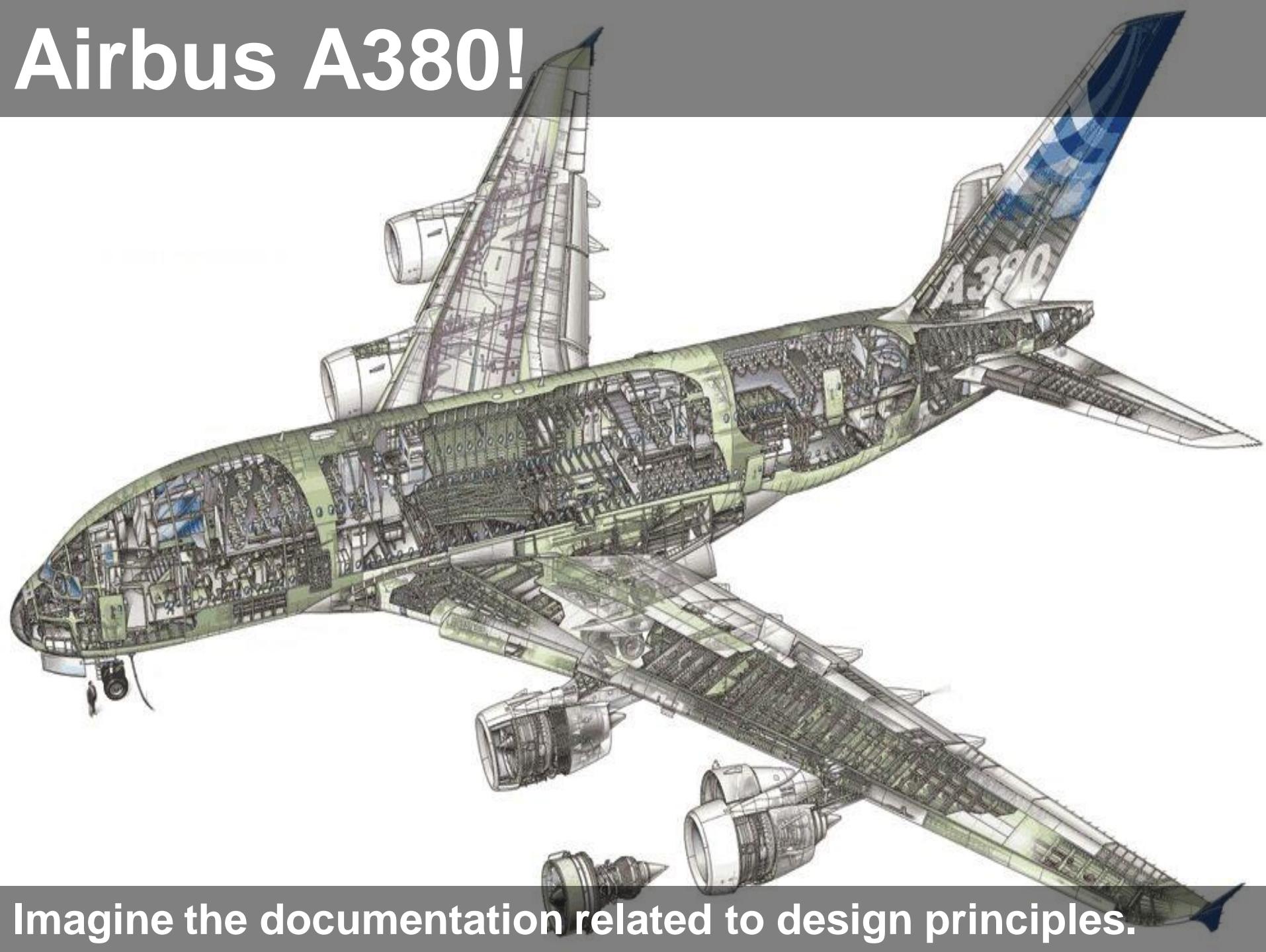
The Project Idea



What is the problem?



Airbus A380!

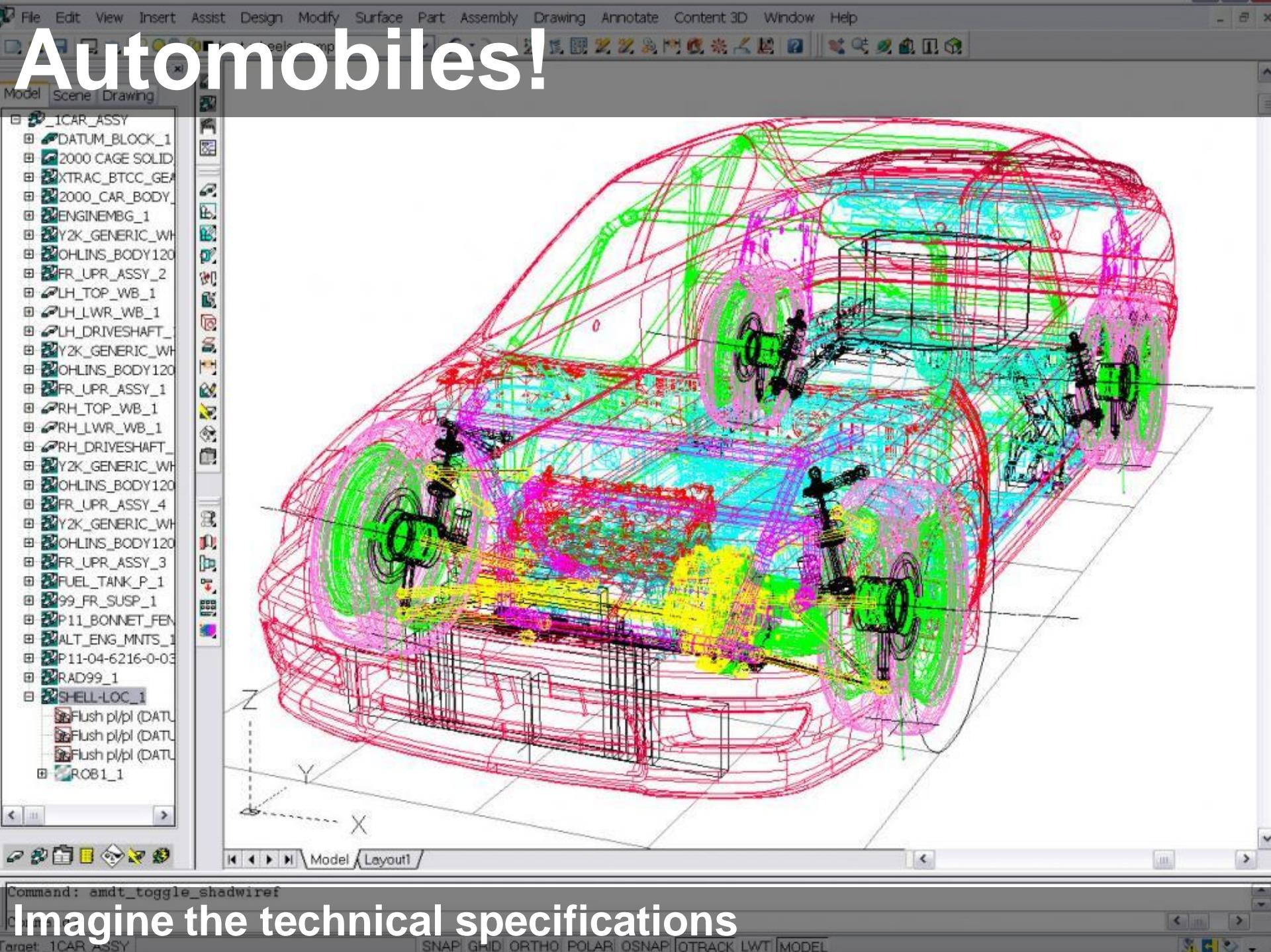


Imagine the documentation related to design principles.



Plants!

Imagine the documentation of a construction site.



Automobiles!

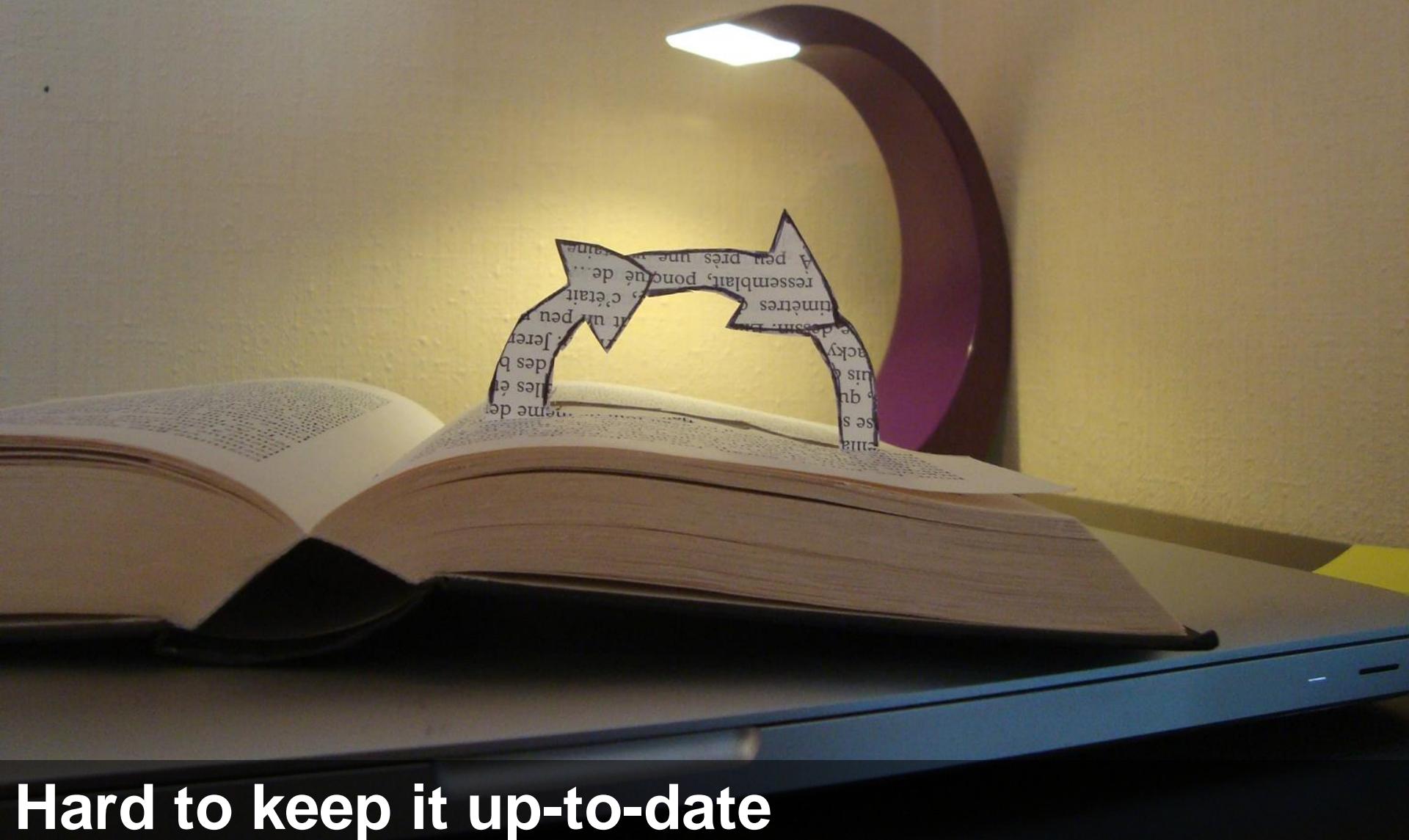
Imagine the technical specifications

Documentation!



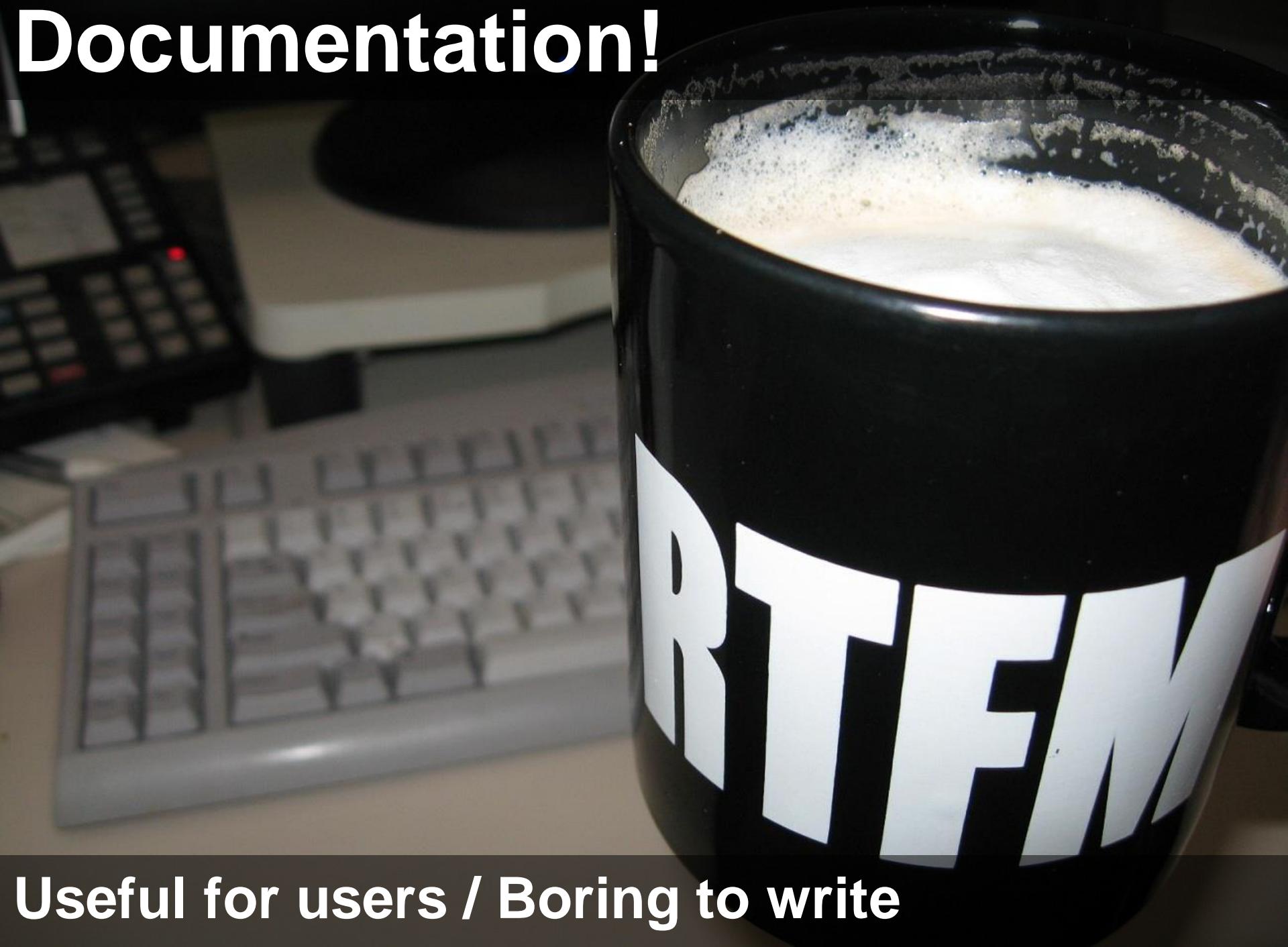
Write once ... and never look at after

Documentation!



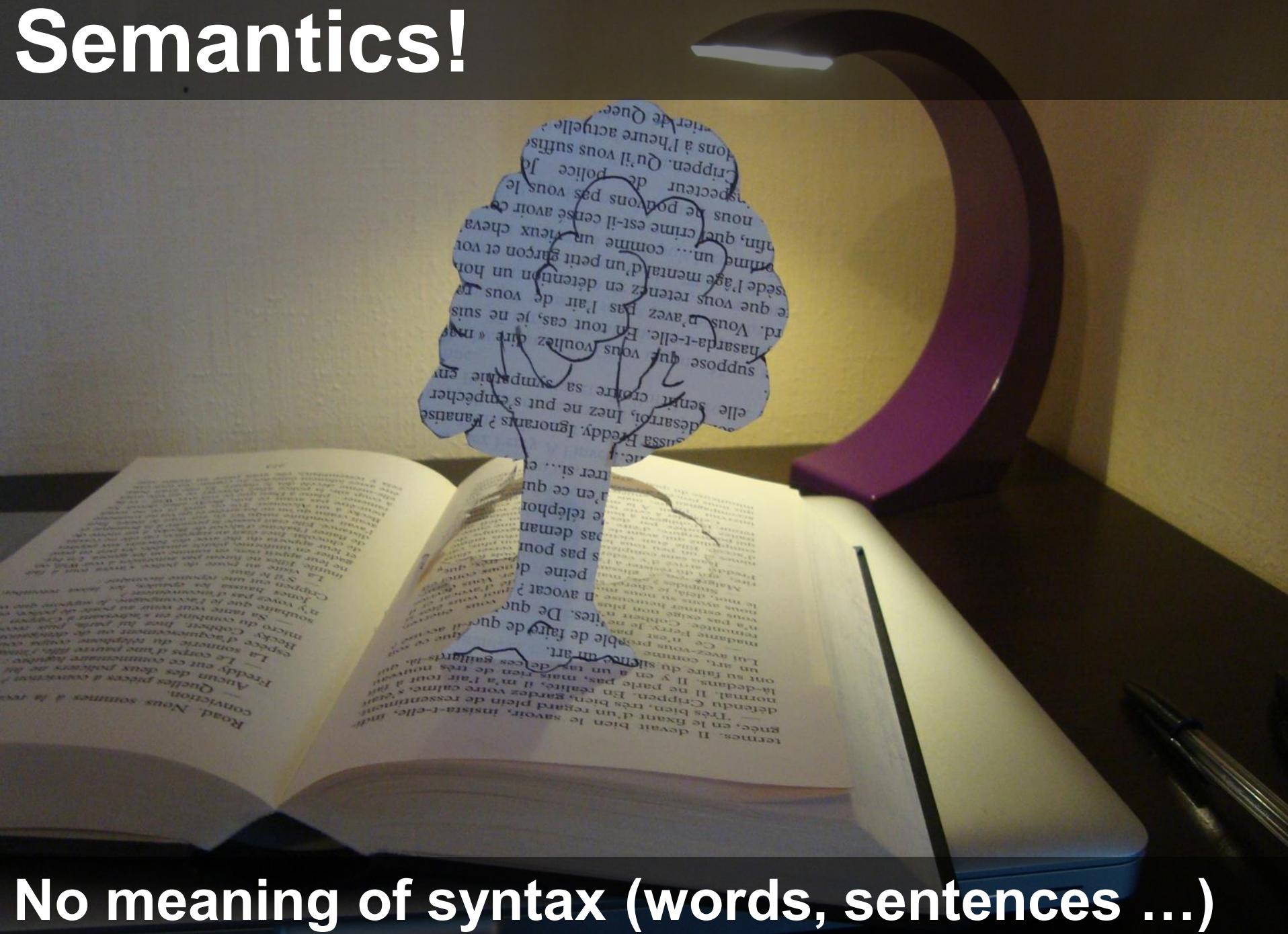
Hard to keep it up-to-date

Documentation!



Useful for users / Boring to write

Semantics!



No meaning of syntax (words, sentences ...)

What is a text?

What is a text? (document file formats)

Office Open XML (.docx) (ISO/IEC 29500)



The screenshot shows a Microsoft Word document titled "Library Management System.docx". The document structure is as follows:

- GLOSSARY**
 - 1.1 BOOK
Book is a kind of collection item. It has an author or editor and (...)
 - 1.2 ...
- REQUIREMENTS**
 - 1.1 REQUIREMENT - RESPONSE TIME FOR BOOK SEARCHES
The system shall perform all book search operations in less than 3 seconds.
 - 1.2 REQUIREMENT - VALIDATION OF THE BOOK
The system allows the user to add new book data through a special book form. The system validates book before storing it.
 - 1.3 ...

The Word ribbon is visible at the top, showing tabs like FILE, HOME, and INSERT. The left sidebar includes a navigation pane with headings and pages, and a search bar. The status bar at the bottom indicates "PAGE 1 OF 1", "76 WORDS", "ENGLISH (UNITED STATES)", and a zoom level of "%80".

What is a text? (document file formats)

Office Open XML (.docx) (ISO/IEC 29500)



Java - PropertyPage/test/document.xml - Eclipse

File Edit Source Navigate Search Project Sample Run Window Help

Sample Plain Text File document.xml

```
19    xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
20    <w:body>
21        <w:p w:rsidR="001D662C" w:rsidRDefault="004D2229" >
22            <w:pPr>
23                <w:pStyle w:val="Title" />
24            </w:pPr>
25            <w:bookmarkStart w:id="0" w:name="_GoBack" />
26            <w:bookmarkEnd w:id="0" />
27            <w:r>
28                <w:t xml:space="preserve">Library Management System </w:t>
29            </w:r>
30        </w:p>
31        <w:p w:rsidR="004D2229" w:rsidRDefault="004D2229" w:rsidP="004D2229">
32            <w:pPr>
33                <w:pStyle w:val="Heading1" />
34            <w:numPr>
35                <w:ilvl w:val="0" />
36                <w:numId w:val="0-1" />
37            </w:numPr>
```

Design Source

P... @ J... D... S... P... G... C... H... P... E... C... T... E... D... E... P...

Property	Value
w:val	Heading1

w:document/w:body/w:...:pPr/w:pStyle/w:val Writable Smart Insert 33 : 38

The screenshot shows the Eclipse IDE interface with the Java - PropertyPage/test/document.xml window open. The code editor displays the XML structure of a Microsoft Word document (.docx). The XML includes elements like w:body, w:p, w:pPr, w:pStyle, w:bookmarkStart, w:bookmarkEnd, w:r, w:t, w:numPr, and w:ilvl. The 'Heading1' style is selected in the XML code. Below the code editor is a toolbar with various icons for document operations. At the bottom, there's a property grid showing the 'w:val' property of a 'w:pStyle' element set to 'Heading1'. The status bar at the bottom indicates the path 'w:document/w:body/w:...:pPr/w:pStyle/w:val', the status 'Writable', and the time '33 : 38'.

What is a text? (.md source file)

text/markdown (ICANN Standard)



The screenshot shows the Eclipse IDE interface with a Markdown file open. The title bar reads "Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/model/Requirement.md - Eclipse". The menu bar includes File, Edit, Navigate, Search, Project, Sample, Run, Window, and Help. The toolbar has icons for Sample Plain Text File, document.xml, and *Requirement.md. The left pane shows a tree view of the document structure:

- # Library Management System
 - ## GLOSSARY
 - ### 1.1 BOOK
 - [Book is a kind of collection item. It has an author or editor and] ()
 - ### 1.2 ...
 - ## REQUIREMENTS
 - ### 1.1 REQUIREMENT - RESPONSE TIME FOR BOOK SEARCHES
 - The [system] () shall perform all book search operations in less than 3 seconds.
 - ### 1.2 REQUIREMENT - VALIDATION OF THE BOOK
 - The system allows the [user] () to add new [book] () data through a special [book form] (). The system [validates book] () before storing it.
 - ### 1.3 ...

The right pane shows a detailed tree view of the document structure:

- h1. Library Management System
 - h2. GLOSSARY
 - h3. 1.1 BOOK
 - h3. 1.2 ...
- h2. REQUIREMENTS
 - h3. 1.1 REQUIREMENT - RE
 - h3. 1.2 EQUIREMENT - VAL
 - h3. 1.3 ...

At the bottom, there are tabs for "Markdown Source" and "Preview", and status bars for "Writable", "Insert", and "16:1".

What is a text? (HTML Preview) text/markdown (ICANN Standard)



The screenshot shows the Eclipse IDE interface with a Markdown editor open. The title bar indicates the file is "Requirement.md". The left pane displays the content of the Markdown file:

```
Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/model/Requirement.md - Eclipse
File Edit Navigate Search Project Sample Run Window Help
*ReqModel pa... ReqModel.emf Requirement.md >4
```

Library Management System

GLOSSARY

1.1 BOOK

Book is a kind of collection item. It has an author or editor and

1.2 ...

REQUIREMENTS

1.1 REQUIREMENT - RESPONSE TIME FOR BOOK SEARCHES

The system shall perform all book search operations in less than 3 seconds.

1.2 REQUIREMENT - VALIDATION OF THE BOOK

The system allows the user to add new book data through a special book form. The system validates book before storing it.

1.3 ...

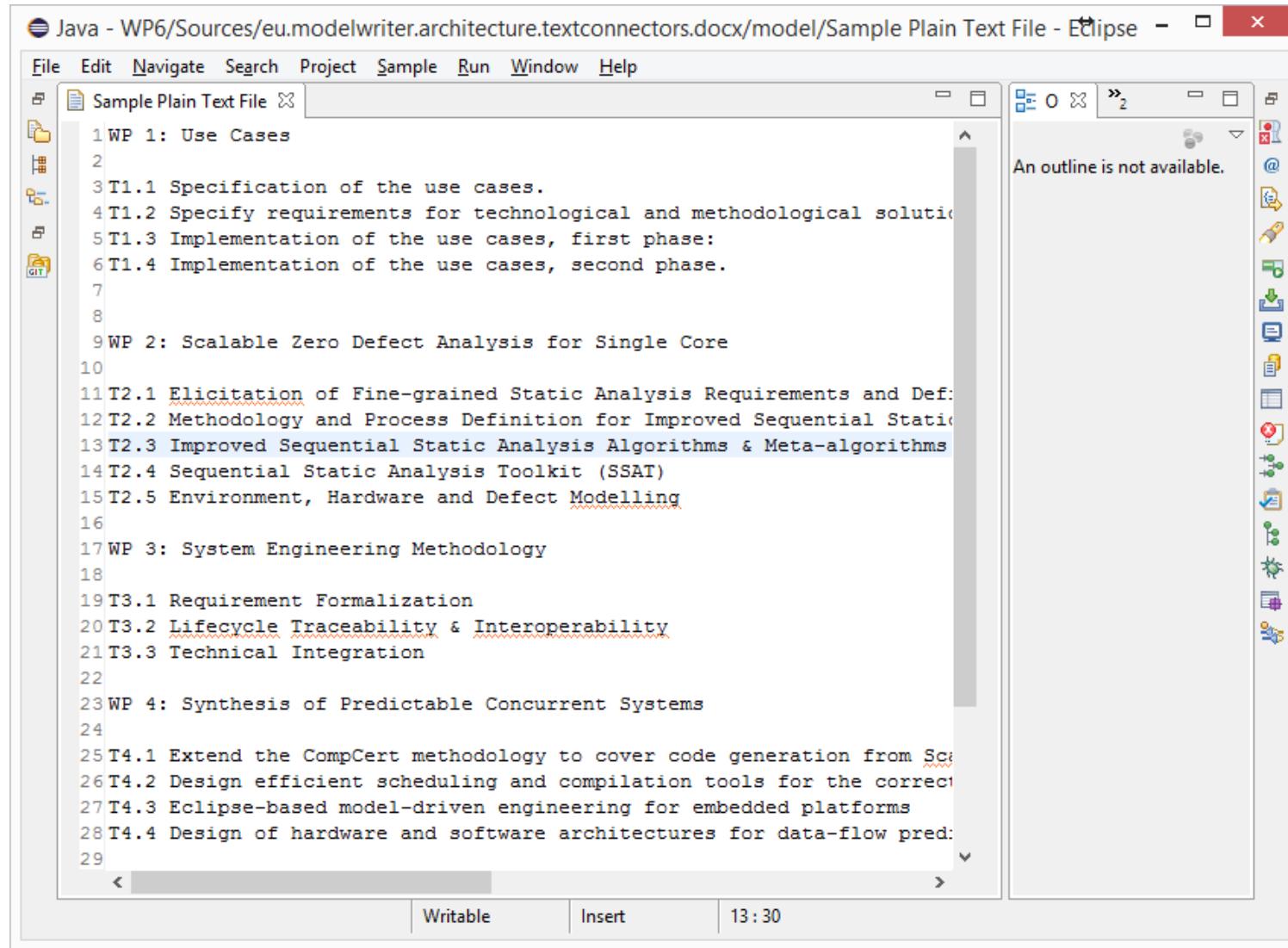
Markdown Source Preview

Writable Insert 16 : 1

The right pane shows a tree view of the document structure:

- h1. Library Management System
 - h2. GLOSSARY
 - h3. 1.1 BOOK
 - h3. 1.2 ...
 - h2. REQUIREMENTS
 - h3. 1.1 REQUIREMENT - RESPON
 - h3. 1.2 EQUIREMENT - VALIDATI
 - h3. 1.3 ...

What is a text? (unformatted text) text/plain (ICANN Standard)



Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/model/Sample Plain Text File - Eclipse

File Edit Navigate Search Project Sample Run Window Help

Sample Plain Text File

```
1 WP 1: Use Cases
2
3 T1.1 Specification of the use cases.
4 T1.2 Specify requirements for technological and methodological solution
5 T1.3 Implementation of the use cases, first phase:
6 T1.4 Implementation of the use cases, second phase.
7
8
9 WP 2: Scalable Zero Defect Analysis for Single Core
10
11 T2.1 Elicitation of Fine-grained Static Analysis Requirements and Defe
12 T2.2 Methodology and Process Definition for Improved Sequential Static
13 T2.3 Improved Sequential Static Analysis Algorithms & Meta-algorithms
14 T2.4 Sequential Static Analysis Toolkit (SSAT)
15 T2.5 Environment, Hardware and Defect Modelling
16
17 WP 3: System Engineering Methodology
18
19 T3.1 Requirement Formalization
20 T3.2 Lifecycle Traceability & Interoperability
21 T3.3 Technical Integration
22
23 WP 4: Synthesis of Predictable Concurrent Systems
24
25 T4.1 Extend the CompCert methodology to cover code generation from Sc
26 T4.2 Design efficient scheduling and compilation tools for the correct
27 T4.3 Eclipse-based model-driven engineering for embedded platforms
28 T4.4 Design of hardware and software architectures for data-flow pred:
29
```

An outline is not available.

Writable Insert 13:30

What is a text? (code files)

Java, C++ ... Programming Languages



The screenshot shows a Java IDE interface with two main panes. The left pane displays the source code for a Java class named `ReqModel2DocxConverter`. The right pane shows a UML class diagram for the same class, with various methods and their signatures listed.

```
Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/src/eu/modelwriter/architecture/text... - □ X
```

File Edit Source Refactor Navigate Search Project Sample Run Window Help

Sample Plain Text ... Docx2ReqModelConv... ReqModel2DocxCon... Convert(Resource) : XWPFD

```
7 package eu.modelwriter.architecture.textconnectors.docx;
8
9 import java.io.File;
10
11 public class ReqModel2DocxConverter {
12
13     private static Resource resource;
14     private static XWPFDocument document;
15
16     // Requirement property keywords
17     private final static String REQUIREMENT_NAME = "Name";
18     private final static String REQUIREMENT_DESCRIPTION = "I";
19     private final static String REQUIREMENT_REFINE = "Refine";
20     private final static String REQUIREMENT_DEPENDENCY_TO =
21     private final static String REQUIREMENT_PRIORITY = "Priorit";
22     private final static String REQUIREMENT_PRIORITY_MANDATORY =
23
24
25     public static XWPFDocument Convert(Resource r) throws IOException {
26
27         // Get template document which includes heading styles
28         URL url = new URL("platform:/plugin/eu.modelwriter.a";
29         XWPFDocument template = new XWPFDocument(url.openConnection());
30
31         document = new XWPFDocument();
32
33         XWPFFormats newStyles = document.createStyles();
34         newStyles.setStyles(template.getStyle());
35 }
```

Writable Smart Insert 62 : 23

T... O... T... T... L... R... @... E... G... ReqModel2DocxConverter

- eu.modelwriter.architecture.textconnectors.docx
- ReqModel2DocxConverter
- document : XWPFDocument
- REQUIREMENT_DEPENDENCY_TO : String
- REQUIREMENT_DESCRIPTION : String
- REQUIREMENT_NAME : String
- REQUIREMENT_PRIORITY : String
- REQUIREMENT_PRIORITY_MANDATORY : String
- REQUIREMENT_REFINE : String
- resource : Resource
- Convert(Resource) : XWPFDocument
- getResource() : Resource
- preOrder(RequirementLevel)
- writeRequirement(Definition)
- writeRequirementLevel(RequirementLevel)

What is a model?

Everything is a model! (ReqIF Standard)

Requirements Interchange Format



ProR - platform:/resource/LibraryManagementSystem/My.reqif - formalmind Studio

File Edit Search Requirements fmStudio Window Help

Quick Access

My.reqif Requirements Document

Outline

ID Name Description

ID	Name	Description
1	Librarian	Librarian
1.1	R123	Response Time for Book Searches
1.2	R123	The system shall perform all book search operations in less than 3 seconds.
1.3	UC071	Add new Book
1.4	R124	Validation of the Book

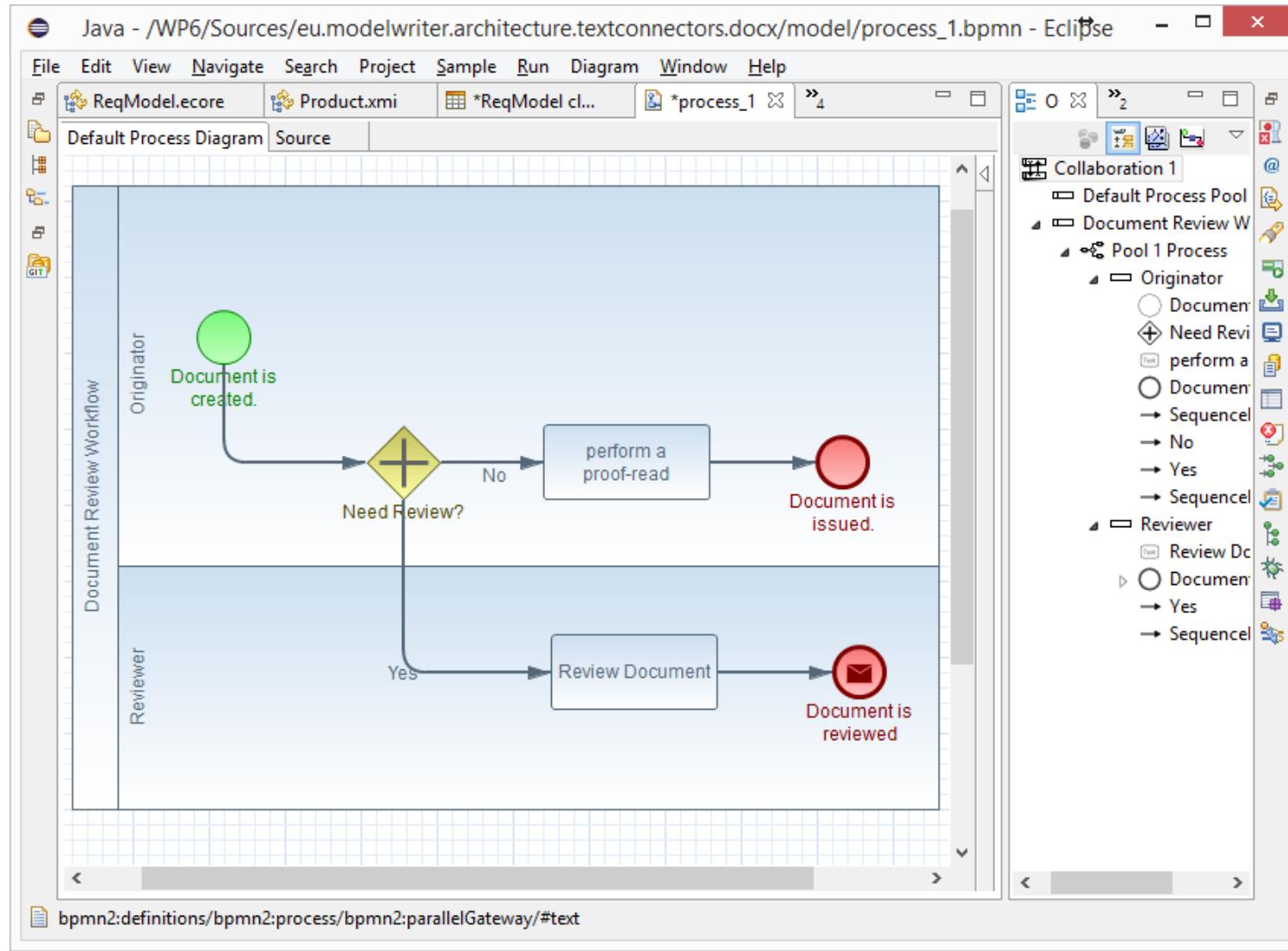
Properties

Property	Value
Requirement Type	
Description	The system shall perform all book search operations in less than 3 seconds.
ID	R123
Name	Response Time for Book Searches
Responsible	Ferhat
Version	1
Spec Object	
Type	Requirement Type (Spec Object)

Standard Attributes All Attributes

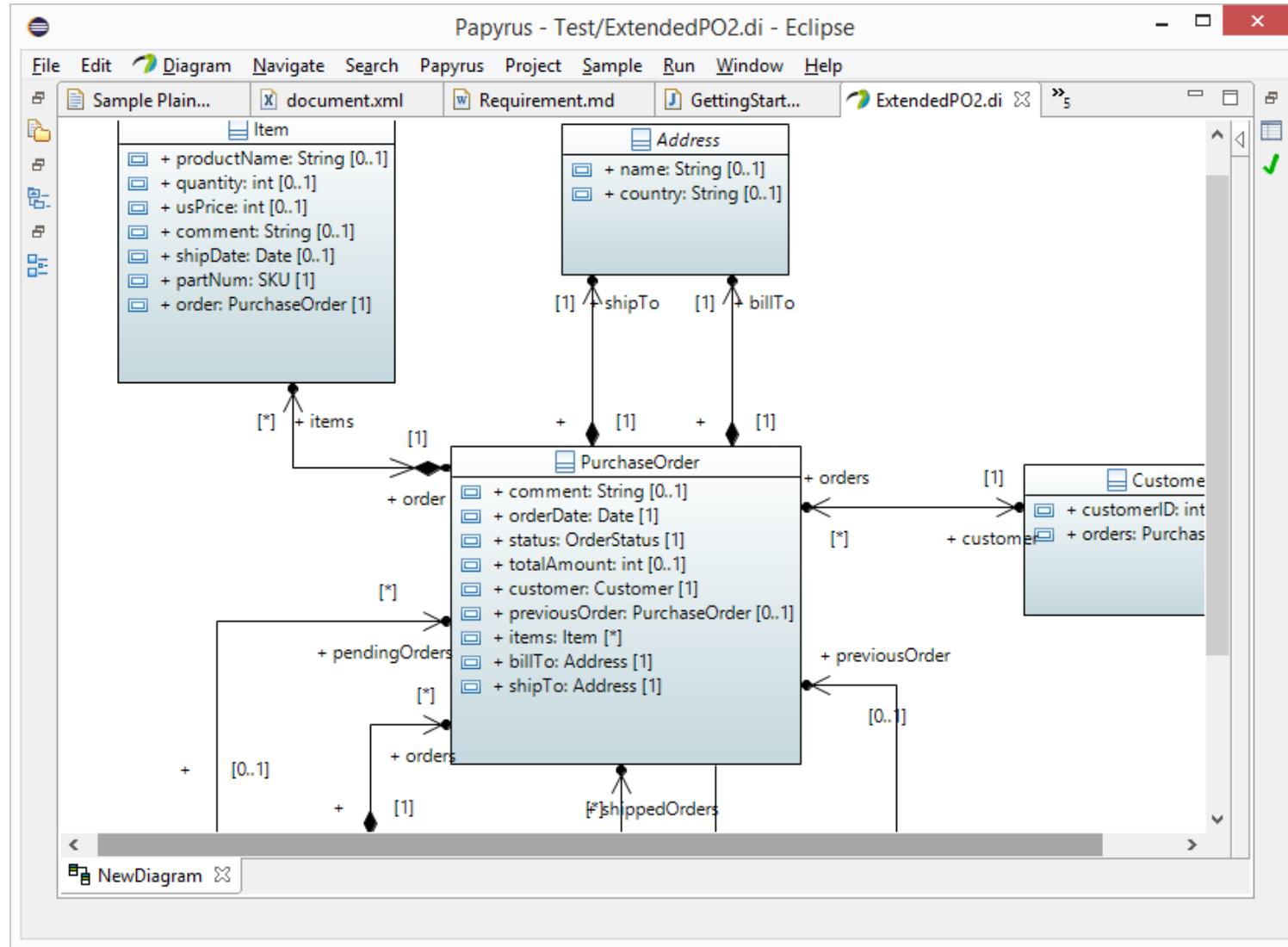
Everything is a model! (BPMN Standard)

Business Process Model & Notation



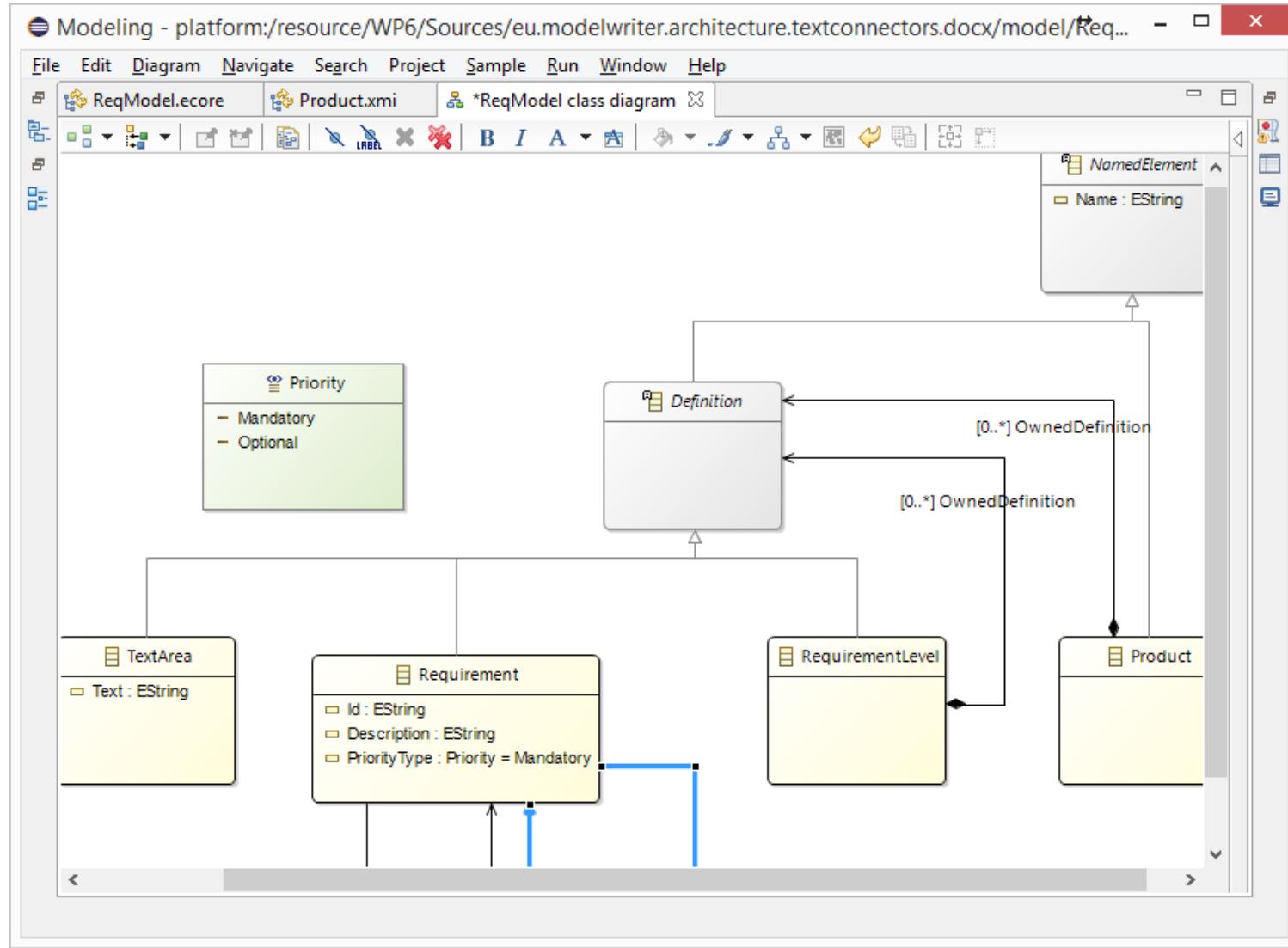
Everything is a model! (UML Standard)

UML Modeling Languages



Everything is a model!

Eclipse Modeling Framework (EMF)



Everything is a model!

Tree-based or Tabular Representations



The screenshot shows a modeling environment with the following components:

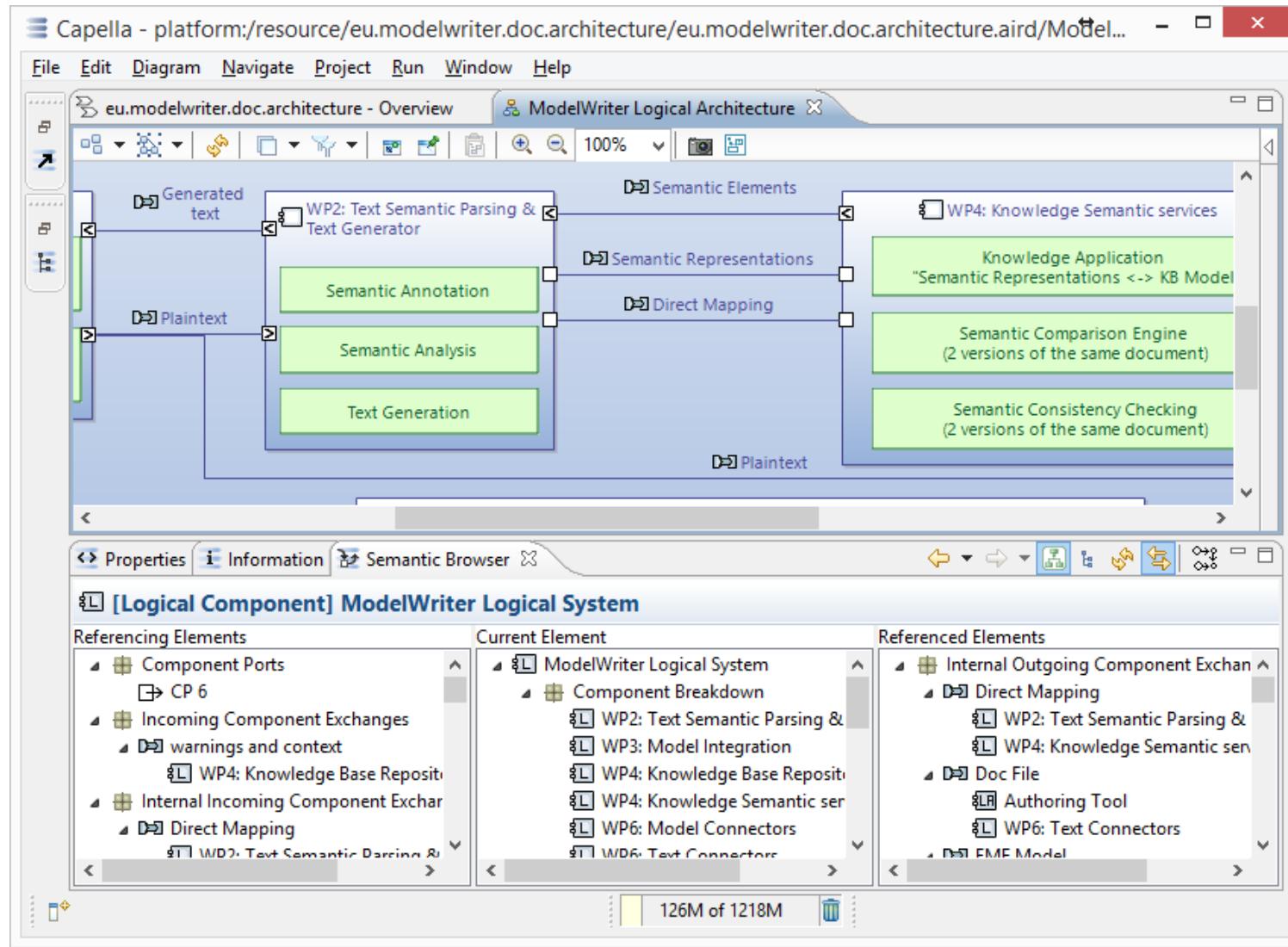
- Top Bar:** File, Edit, Navigate, Search, Project, DTable, Sample, Run, Window, Help.
- Left Sidebar:** Shows a tree view of model elements under "ReqModel.ecore":
 - NamedElement
 - Name : EString
 - Product -> NamedElement
 - OwnedDefinition : Definition
 - Definition -> NamedElement
 - RequirementLevel -> Definition
 - OwnedDefinition : Definition
 - Requirement -> Definition
 - Id : EString
 - Description : EString
 - Refine : Requirement
 - DependencyTo : Requirement
 - PriorityType : Priority
 - TextArea -> Definition
 - Text : EString
- Central View:** A table titled "*ReqModel class table" showing properties for "NamedElement".

Name	Value
Name	NamedElement
Product	
OwnedDefinition	
Definition	
RequirementLevel	
OwnedDefinition	
Requirement	
Id	
Description	
Refine	
DependencyTo	
PriorityType	
TextArea	
Text	
- Bottom View:** A table titled "NamedElement" showing semantic properties.

Semantic	Property	Value
NamedElement	Abstract	true
	Default Value	
	ESuper Types	
	Instance Type Name	

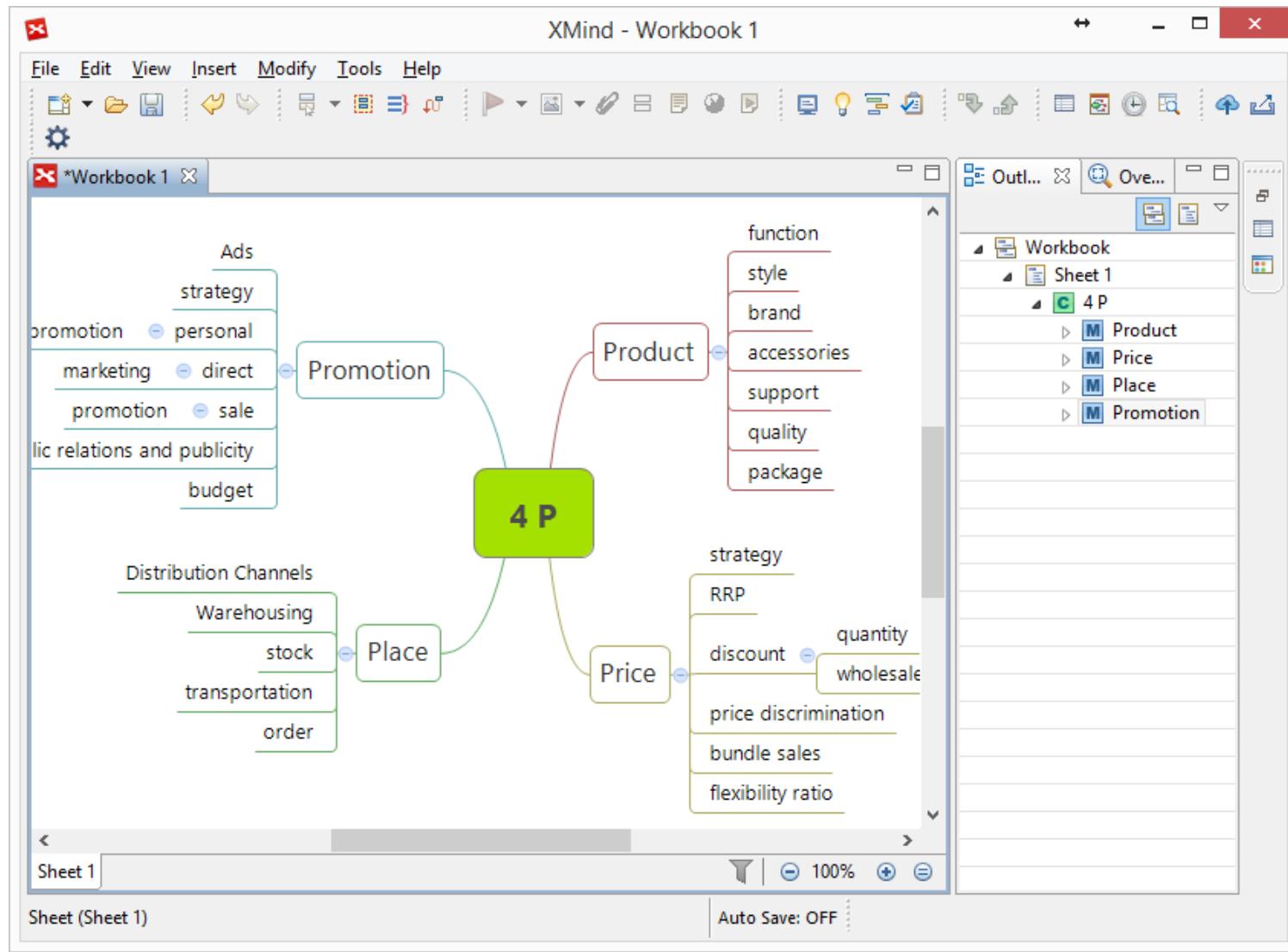
Everything is a model!

Software/System Architecture Design



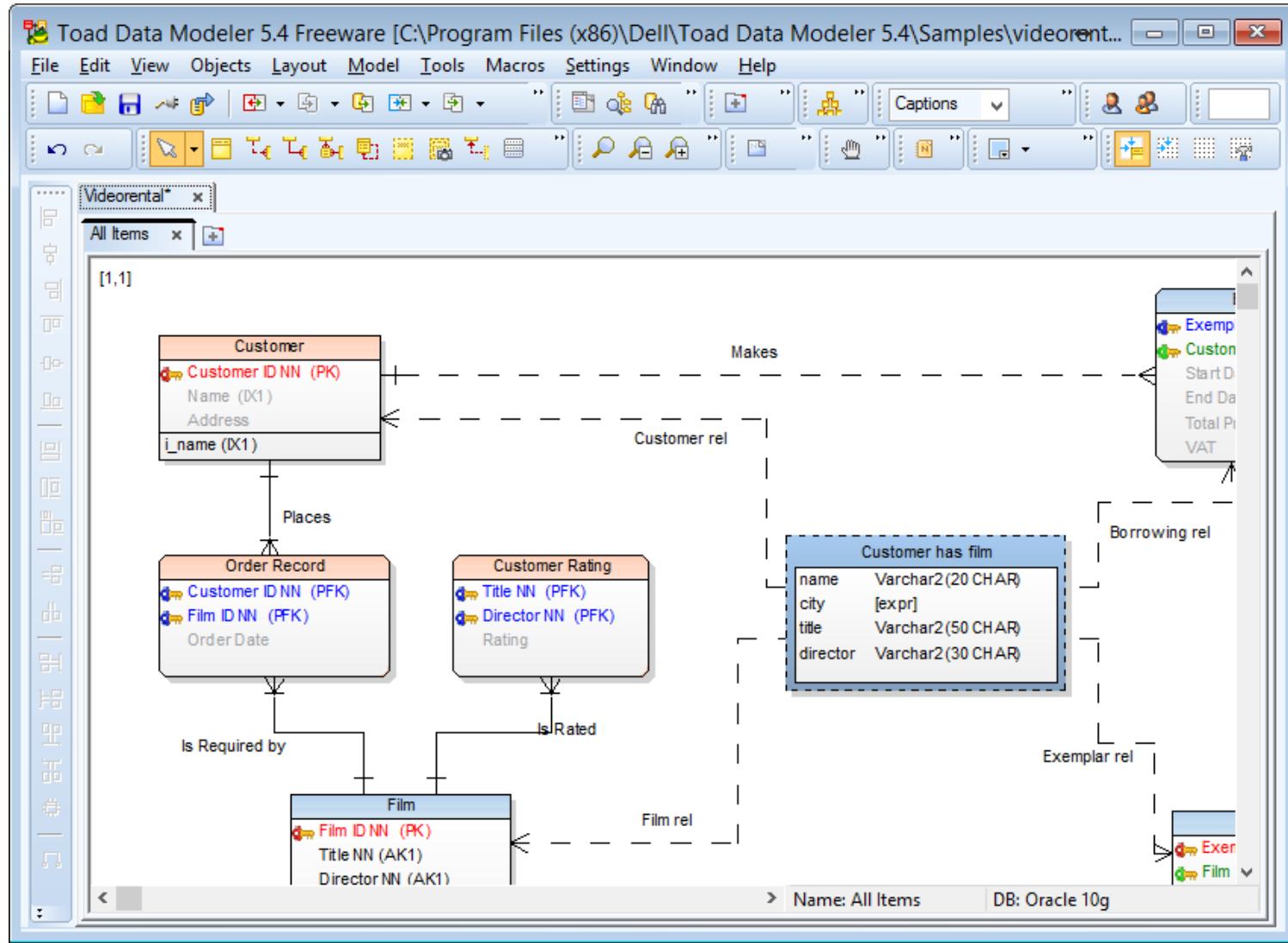
Everything is a model!

Topic Maps, Mind Maps, Vocabularies ...



Everything is a model!

Databases (ER, IDEF1.x)



Everything is a model! (Textual Lang.)

Domain Specific Languages



Modeling - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/model/ReqModel.emf - Eclipse

File Edit Navigate Search Project Sample Run Window Help

ReqModel.ecore Product.xmi *ReqModel cl... *ReqModel do... ReqModel.emf »

```
1 @namespace(uri="eu.modelwriter.architecture.textconnectors.docx.reqmodel", prefix="ReqMod")
2 package ReqModel;
3
4 @gmf.node(label="name")
5 abstract class NamedElement {
6     attr String Name;
7 }
8
9 @gmf.diagram
10 @gmf.node(label="Name")
11 class Product extends NamedElement {
12
13     @gmf.compartment(collapsible="true")
14     val Definition[*] OwnedDefinition;
15 }
16
17 abstract class Definition extends NamedElement {
18 }
19
20 @gmf.node(figure="rectangle", label.icon="true", label="Name", label.pattern="{0}")
21 class RequirementLevel extends Definition {
22
23     @gmf.compartment(collapsible="true", layout="list")
24     val Definition[*] OwnedDefinition;
25 }
26
27 @gmf.node(figure="rounded", label.icon="true", label="Name", label.pattern="{0}")
28 class Requirement extends Definition {
29     attr String Id = "";
30 }
```

Writable Insert 11:9

Everything is a model! (Java, C++, etc.)

Even Programming Languages (ASTs)



The screenshot shows a Java IDE interface with two main panes. The left pane displays the source code for `ReqModel2DocxConverter`. The right pane shows the Abstract Syntax Tree (AST) for the same code, with nodes corresponding to class definitions, static fields, and methods like `Convert`.

```
Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/src/eu/modelwriter/architecture/text...
File Edit Source Refactor Navigate Search Project Sample Run Window Help
Sample Plain Text ... Docx2ReqModelConv... ReqModel2DocxCon... Convert(Resource) : XWPFD
7 package eu.modelwriter.architecture.textconnectors.docx;
8
9 import java.io.File;
10
11 public class ReqModel2DocxConverter {
12
13     private static Resource resource;
14     private static XWPFDocument document;
15
16     // Requirement property keywords
17     private final static String REQUIREMENT_NAME = "Name";
18     private final static String REQUIREMENT_DESCRIPTION = "I";
19     private final static String REQUIREMENT_REFINE = "Refine";
20     private final static String REQUIREMENT_DEPENDENCY_TO =
21     private final static String REQUIREMENT_PRIORITY = "Priorit";
22     private final static String REQUIREMENT_PRIORITY_MANDATORY =
23
24
25     public static XWPFDocument Convert(Resource r) throws IOException {
26
27         // Get template document which includes heading styles
28         URL url = new URL("platform:/plugin/eu.modelwriter.a";
29         XWPFDocument template = new XWPFDocument(url.openConnection());
30
31         document = new XWPFDocument();
32
33         XWPFStyles newStyles = document.createStyles();
34         newStyles.setStyles(template.getStyle());
35
36     }
37
38 }
```

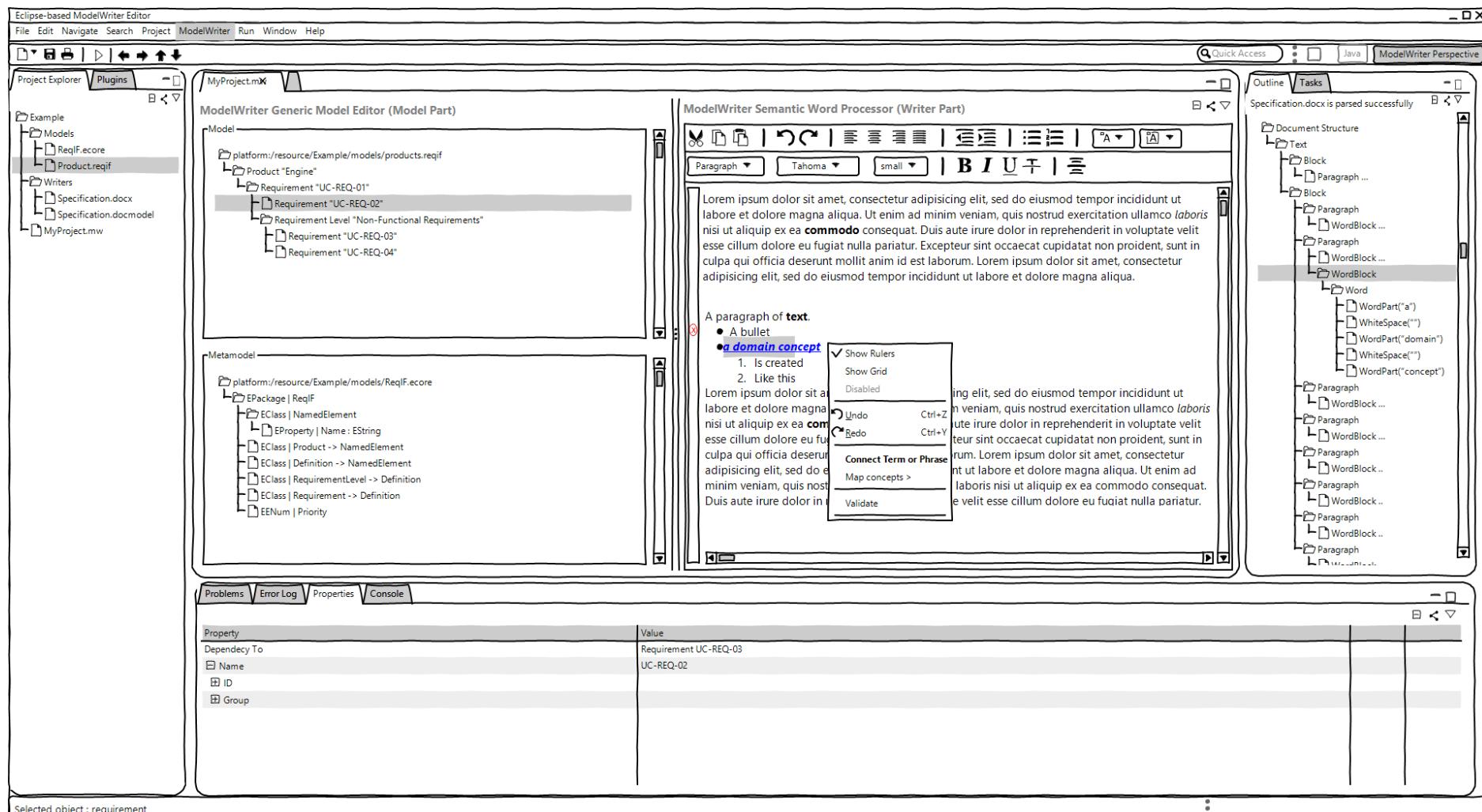
The right pane shows the AST structure:

- `ReqModel2DocxConverter`
 - `document : XWPFDocument`
 - `REQUIREMENT_DEPENDENCY_TO : XWPFText`
 - `REQUIREMENT_DESCRIPTION : XWPFText`
 - `REQUIREMENT_NAME : String`
 - `REQUIREMENT_PRIORITY : XWPFText`
 - `REQUIREMENT_PRIORITY_MANDATORY : XWPFText`
 - `REQUIREMENT_REFINE : XWPFText`
 - `resource : Resource`
- `Convert(Resource) : XWPFDocument`
- `getResource() : Resource`
- `preOrder(RequirementLevel) : void`
- `writeRequirement(Definition) : void`
- `writeRequirementLevel(RequirementLevel) : void`

Is it possible to connect and keep arbitrary software/system engineering artifacts synchronized ?

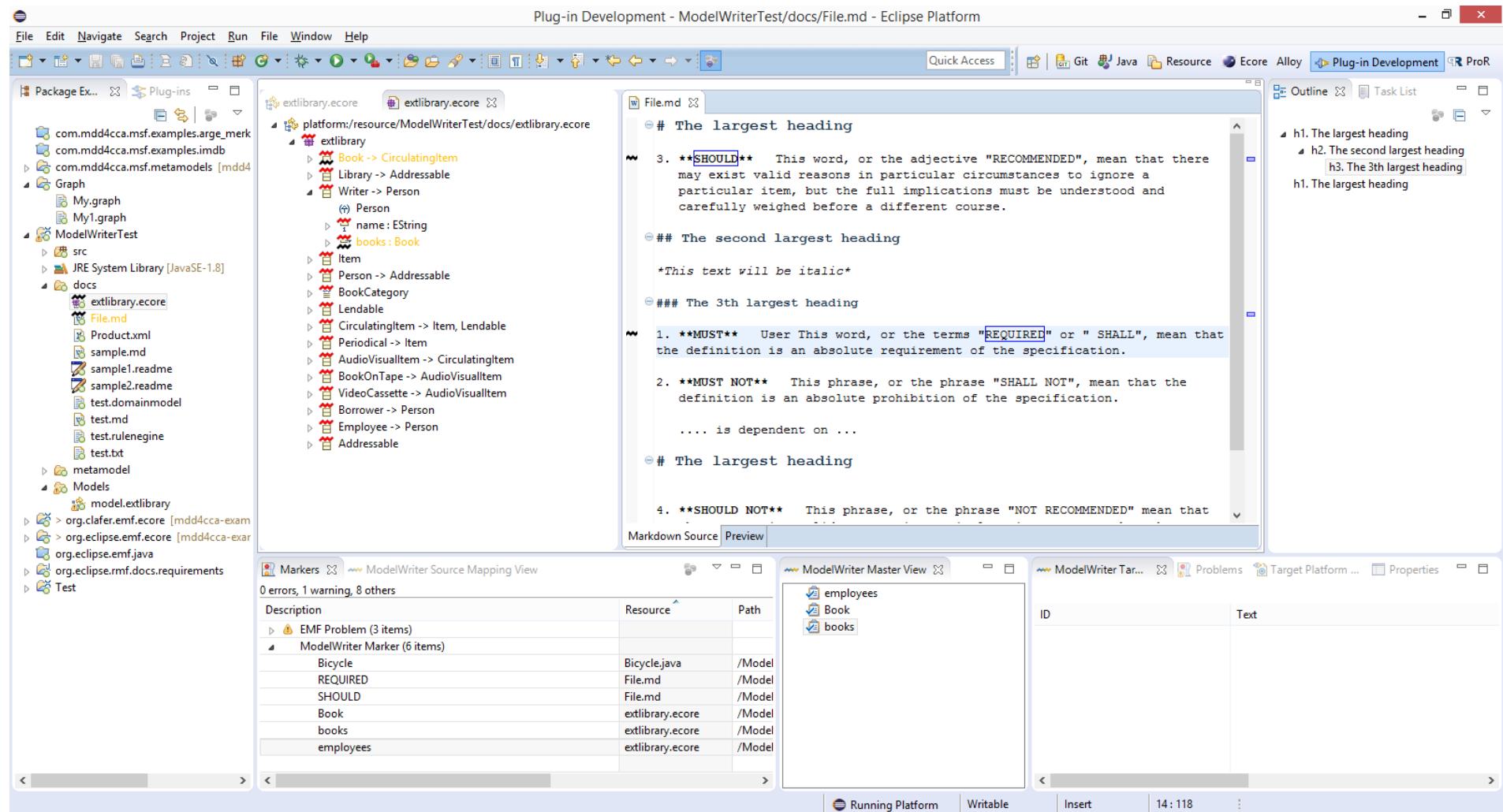


ModelWriter – The Solution



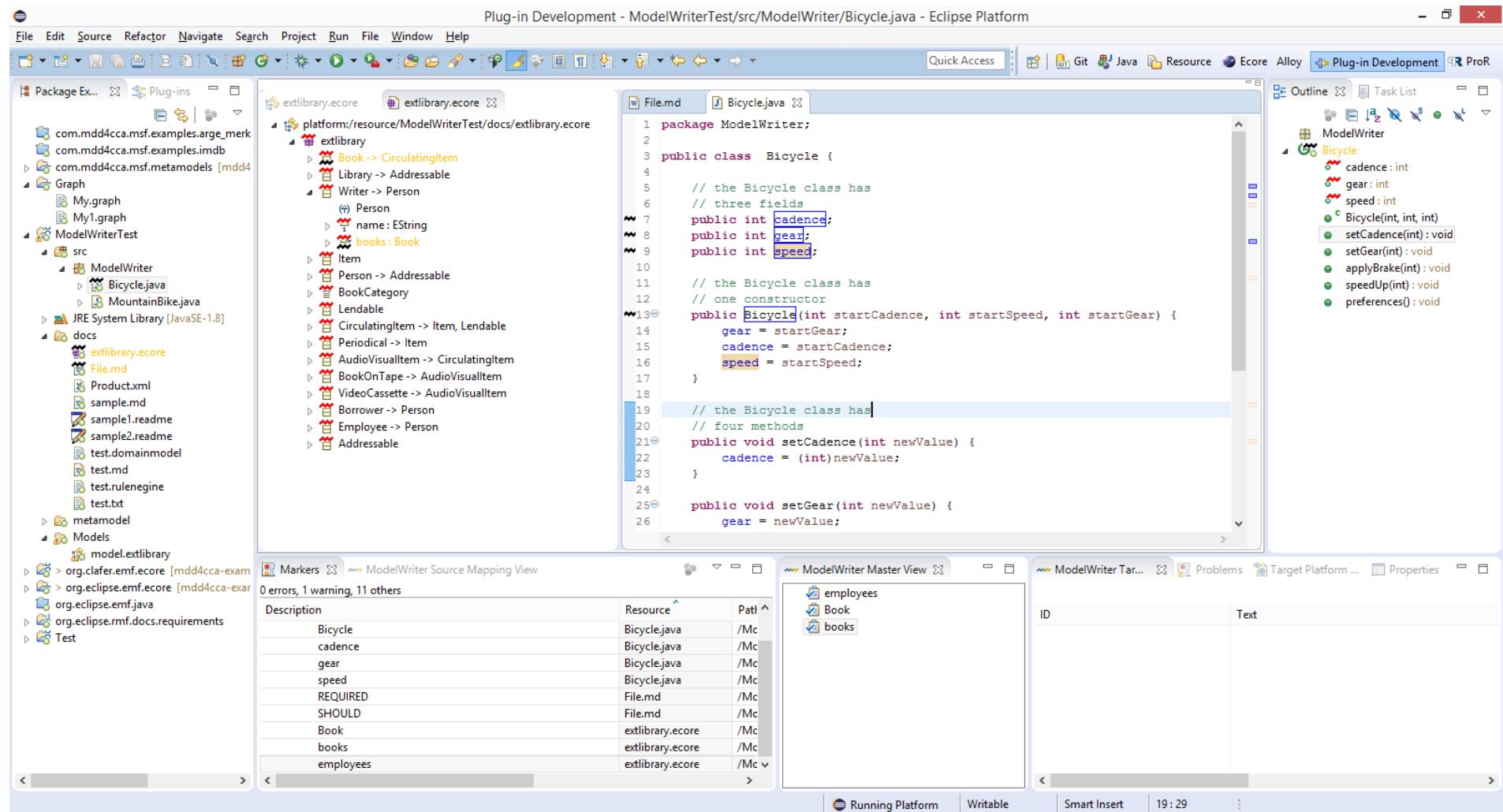
Text & Model-Synchronized Document Engineering Platform

Solution – Knowledge Capture



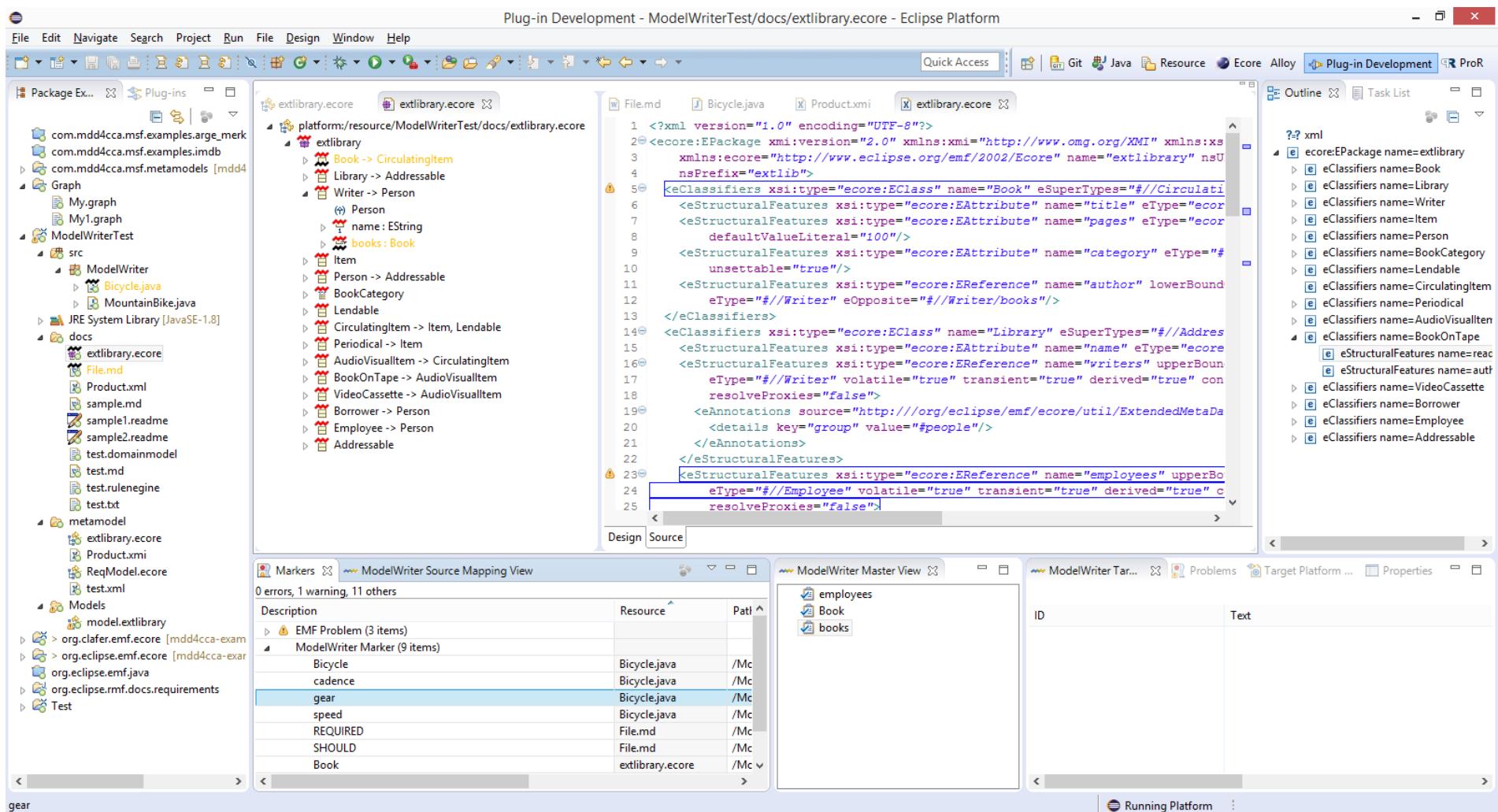
Text & Model-Synchronized Document Engineering Platform

Solution – Knowledge Capture



Text & Model-Synchronized Document Engineering Platform

Solution – Knowledge Capture



Text & Model-Synchronized Document Engineering Platform

Solution – Knowledge Extraction

ModelWriter Project

File Link Change Statistic

The
Generate Links
Search Link
2 P... Add Link
ABS... Remove Link

unction zones
shall be used

Flexible Hoses Shall be defined with a maximum length of 500 mm regardless of
ABS2195 -LRB- preferred for weight saving -RRB- or NSA5516J P-Clamp Shall be U
Rigid Pipes Shall be segregated to fixed Structure by not less than 10 mm as sh
Flexible Hoses Shall be segregated to rigid Component/Item/Object by not less t
Rigid Pipes Shall be segregated to movable Component/Item/Object by not less
Flexible Hoses Shall be segregated to movable Component/Item/Object by not le
Pipes Shall be fixed
Unions Shall be fixed on Pipes at alternating positions as shown in the attach
Unions Shall be positioned close to one fixation point .

The Model

Plain Tree

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:acs="http://airbus-group/aircraft-system#"
  xmlns:evt="http://airbus-group.installsys/event#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:spin="http://spinrdf.org/spin#"
  xmlns:qudt="http://qudt.org/schema/qudt#"
  xmlns:dct="http://purl.org/dc/terms/"
  xmlns:arg="http://spinrdf.org/arg#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:vaem="http://www.linkedmodel.org/schema/vaem#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:voag="http://voag.linkedmodel.org/voag/"
  xmlns:comp="http://airbus-group.installsys/component#"
  xmlns:qudt-dimension="http://qudt.org/vocab/dimension#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:iems="http://airbus-group/installationMeasure#"
  xmlns:dtype="http://www.linkedmodel.org/schema/dtype#"
  xmlns:mat="http://airbus-group/material#"
```

The links between text and model

T2M M2T Link

```
<rdf:Description rdf:about="http://ModelWriter/TxtDocument/id270">
  <j:0:hasOffset>270</j:0:hasOffset>
  <j:0:isSameAs>http://www.linkedmodel.org/schema/vaem#id</j:0:isSameAs>
  <j:0:hasValue>id</j:0:hasValue>
</rdf:Description>
<rdf:Description rdf:about="http://ModelWriter/TxtDocument/attach818">
  <j:0:hasOffset>818</j:0:hasOffset>
  <j:0:isSameAs>http://airbus-group/opp-function#Attach</j:0:isSameAs>
  <j:0:hasValue>attach</j:0:hasValue>
</rdf:Description>
<rdf:Description rdf:about="http://ModelWriter/TxtDocument/attached709">
  <j:0:hasOffset>709</j:0:hasOffset>
  <j:0:isMorphologySimilarTo>http://airbus-group/opp-function#Attach</j:0:isMorphologySim
  <j:0:hasValue>attached</j:0:hasValue>
</rdf:Description>
</rdf:RDF>
```

Text & Model-Synchronized Document Engineering Platform



Synchronization is maintained!

Quantification of the expected benefits

- Improvement in quality and productivity of technical documentation.
- Quality increase of the product with consistent requirements and designs.
 - For instance, according to AIRBUS's claims in their use cases, the global saving would be 5 M€ to 7 M€ (A350 Recurring Cost)
- 50% reduction of costs for keeping the documentation up-to-date with the developed software

**Thank you for your attention
We value your opinion and
questions.**

