

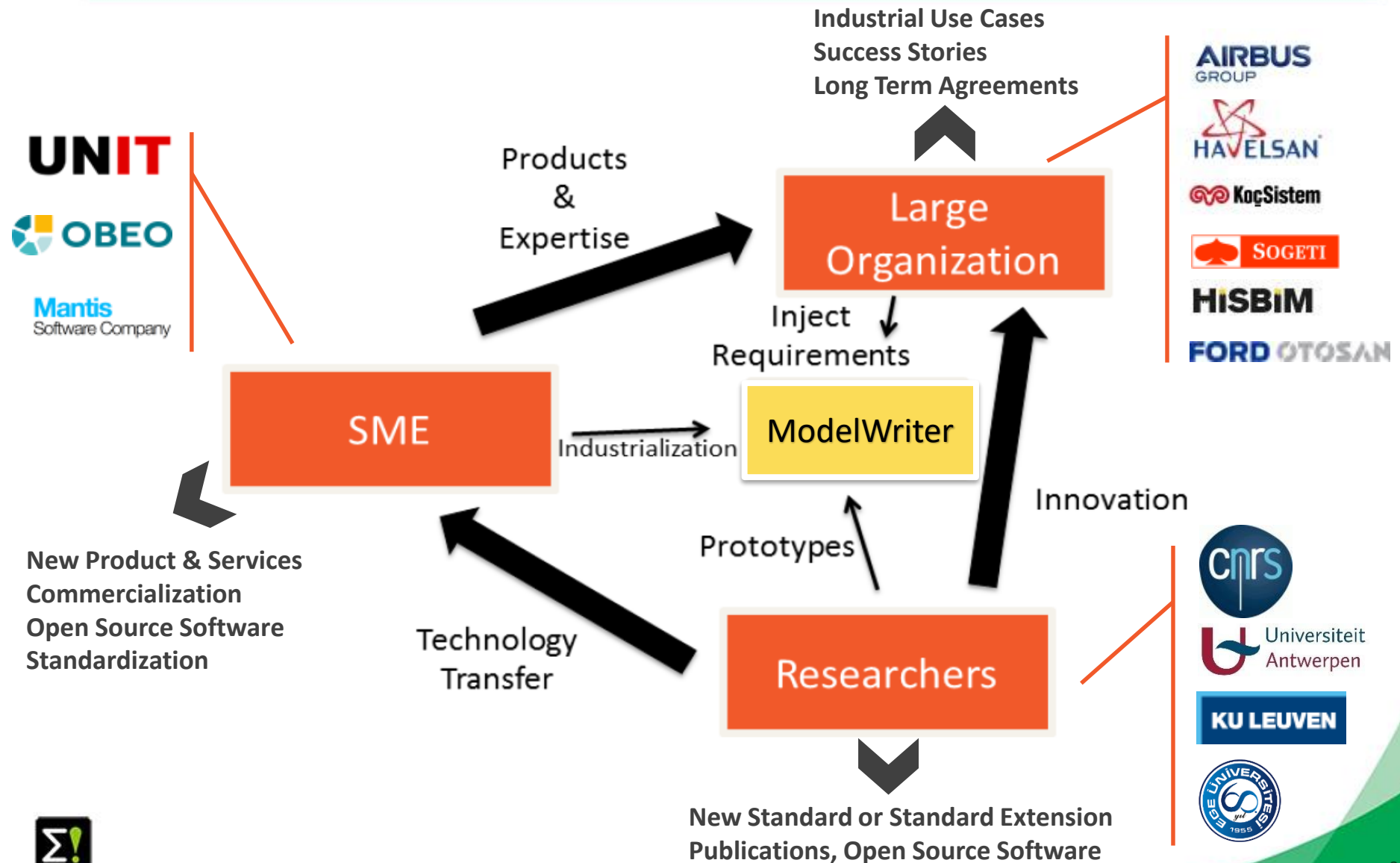
2 Overview of the Project

Ferhat Erata

UNIT, ModelWriter Project Leader

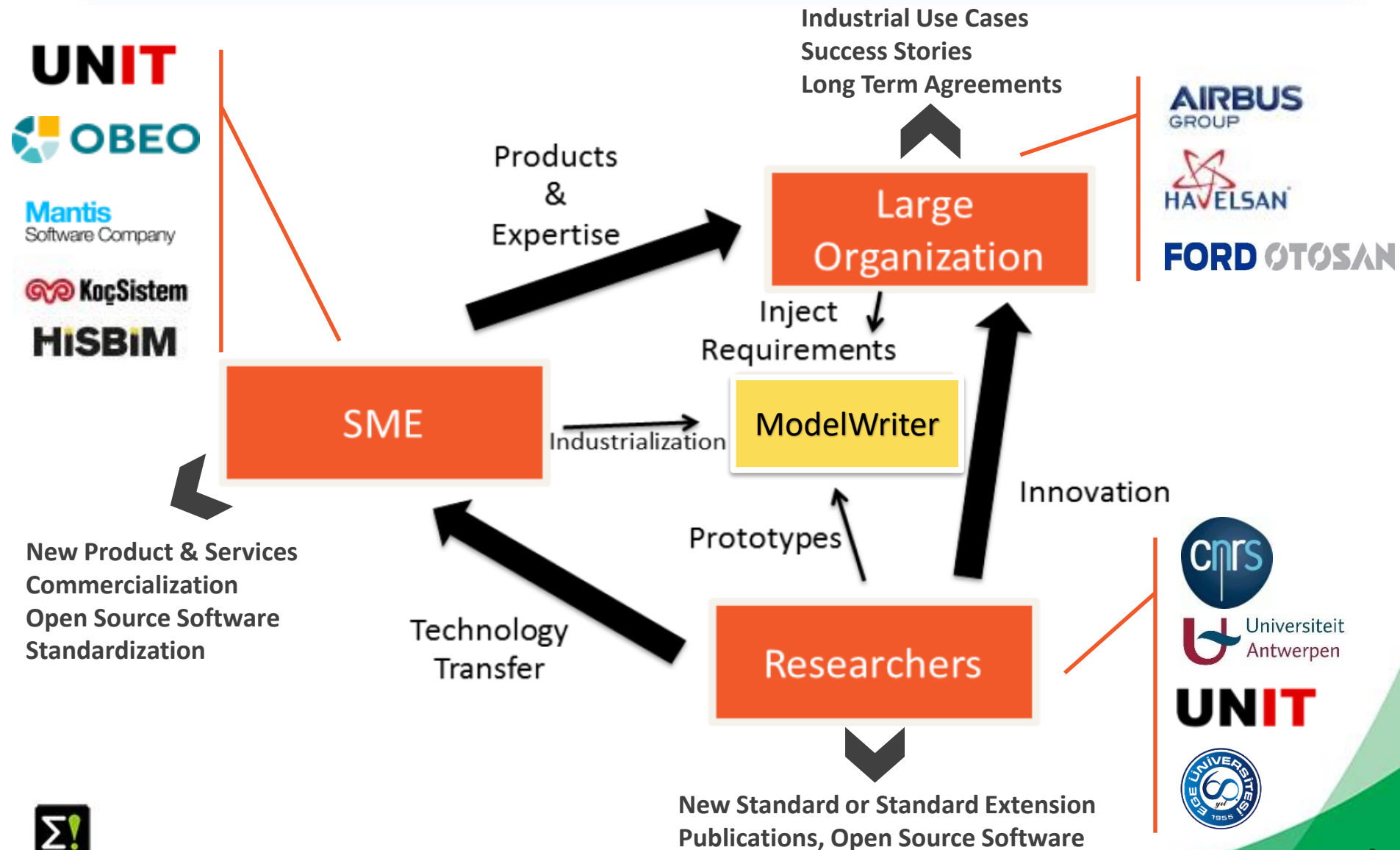
Industrialization Triangle in ModelWriter

Open Source Software (year #1)



Industrialization Triangle in ModelWriter

Open Source Software (year #2)



ModelWriter

Project Overview



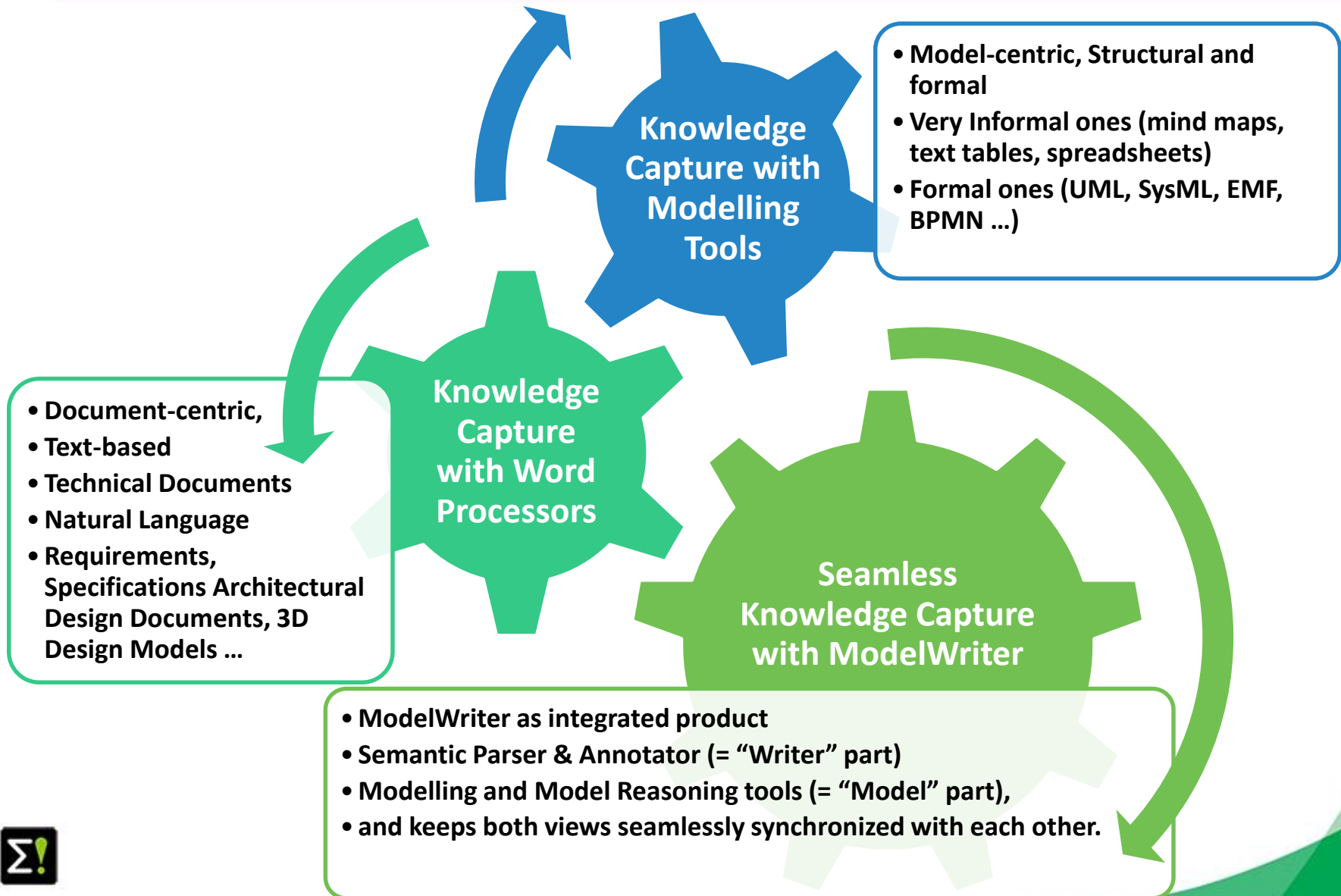
Resource Allocation: 68,71 person year

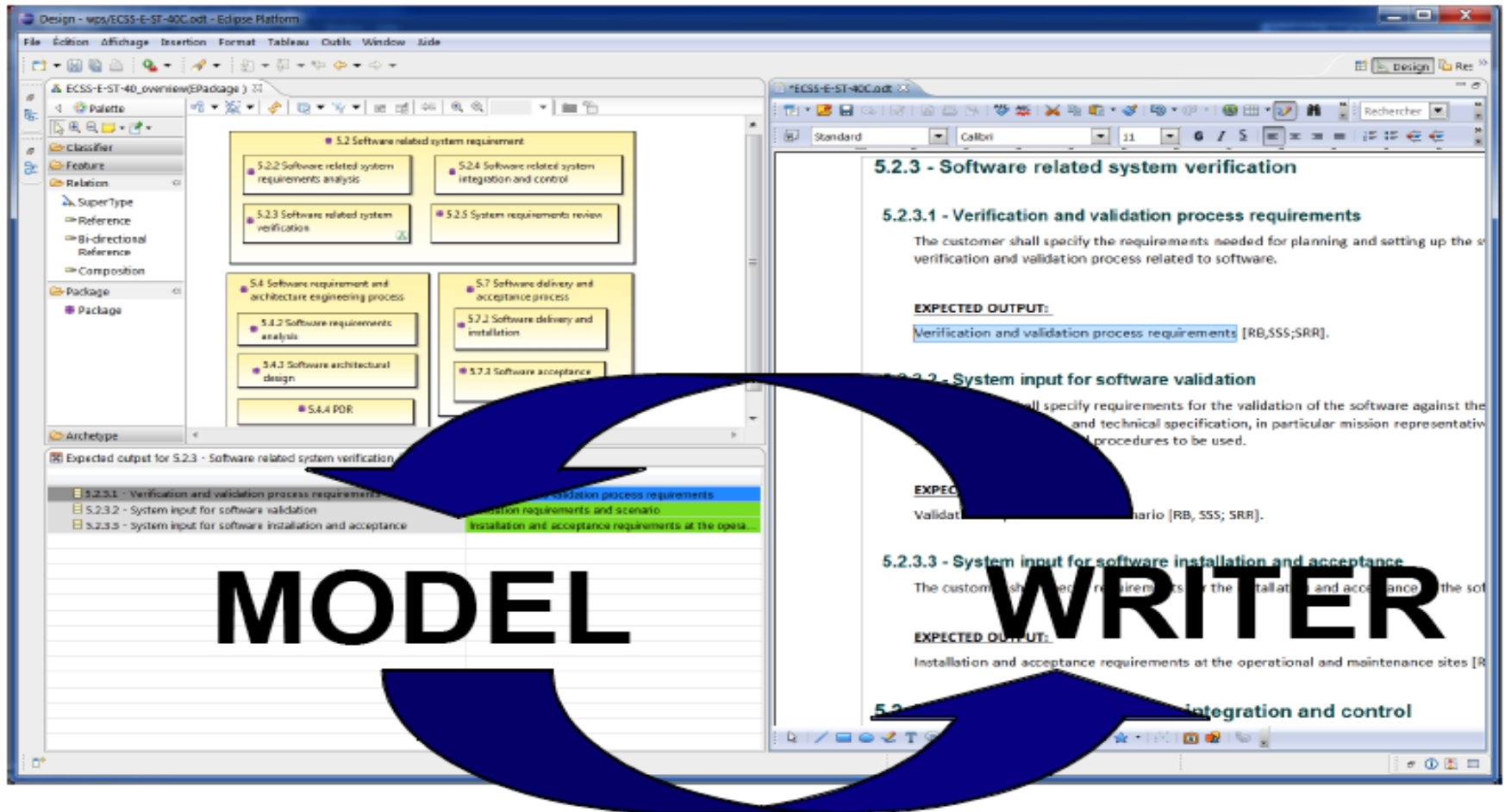
Project Duration: 36 months

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






**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

Further Knowledge Valorizations



“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

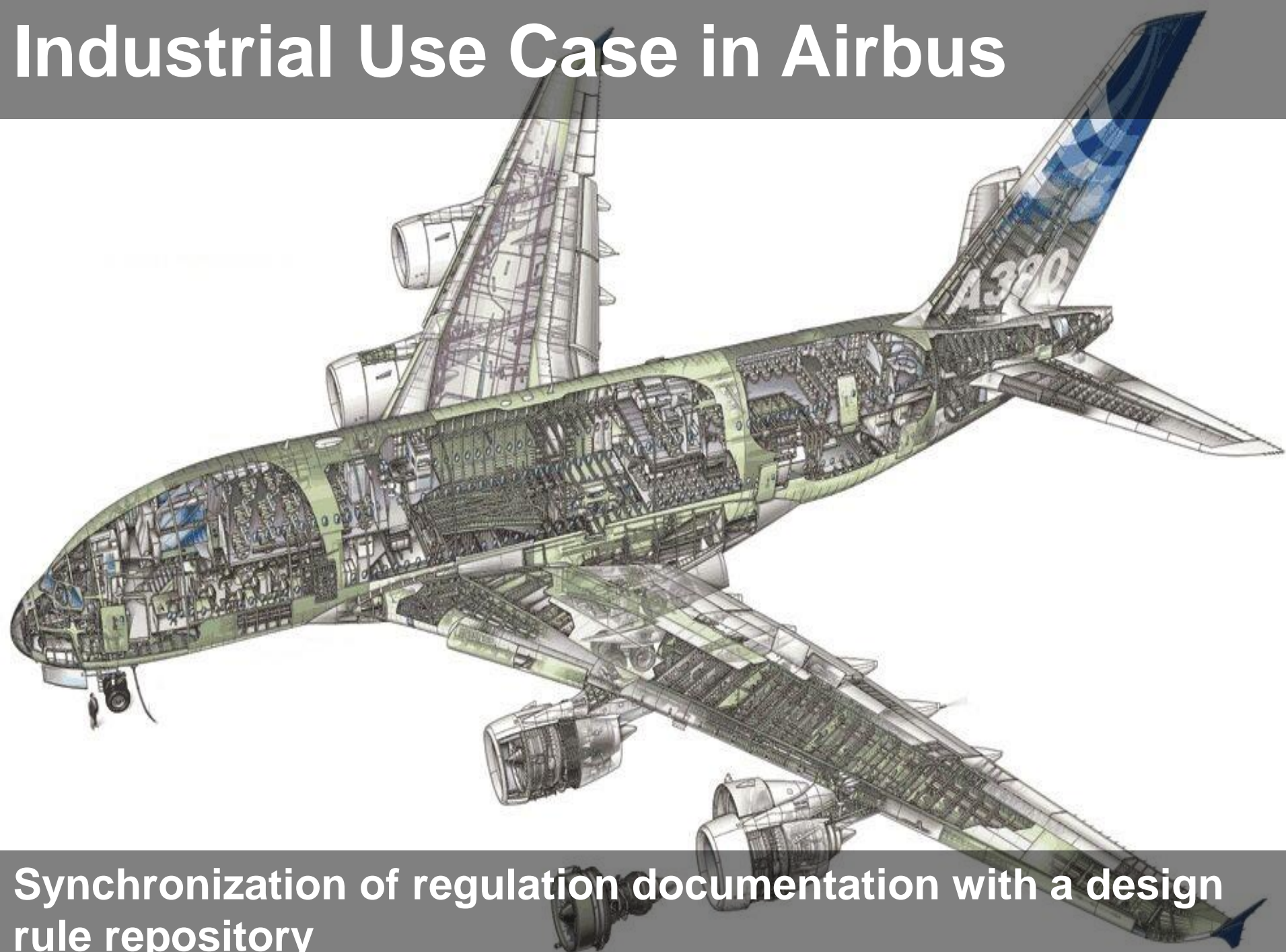
Open Source Software under Eclipse Foundation for Future
Dissemination and Exploitation to further extend the Business Value Chain

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



What is the problem?

Industrial Use Case in Airbus

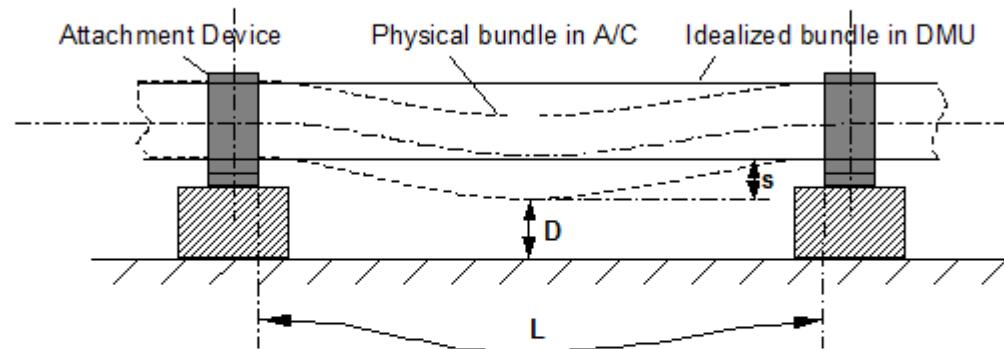


Synchronization of regulation documentation with a design rule repository

SIDP: System Installation Design Principles

SIDP92A001V-A-784

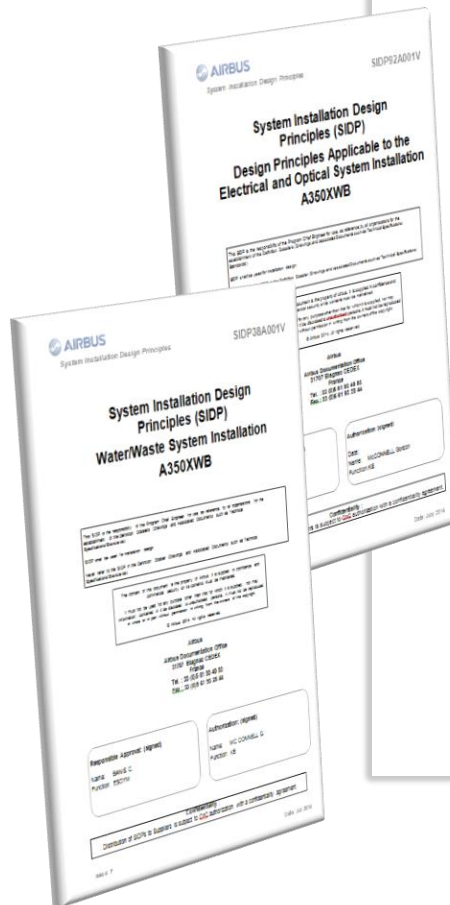
For installation of optical and electrical harnesses additional clearance for sagging (s) shall be provided as detailed below:



s ... Sagging of bundle (real behavior of physical bundle in A/C due to gravity, ageing, etc.)
 D ... Required Distance
 L ... Actual length of a bundle segment between two Attachment Points (as designed in DMU)

Figure 6: Sagging of bundles between attachment points

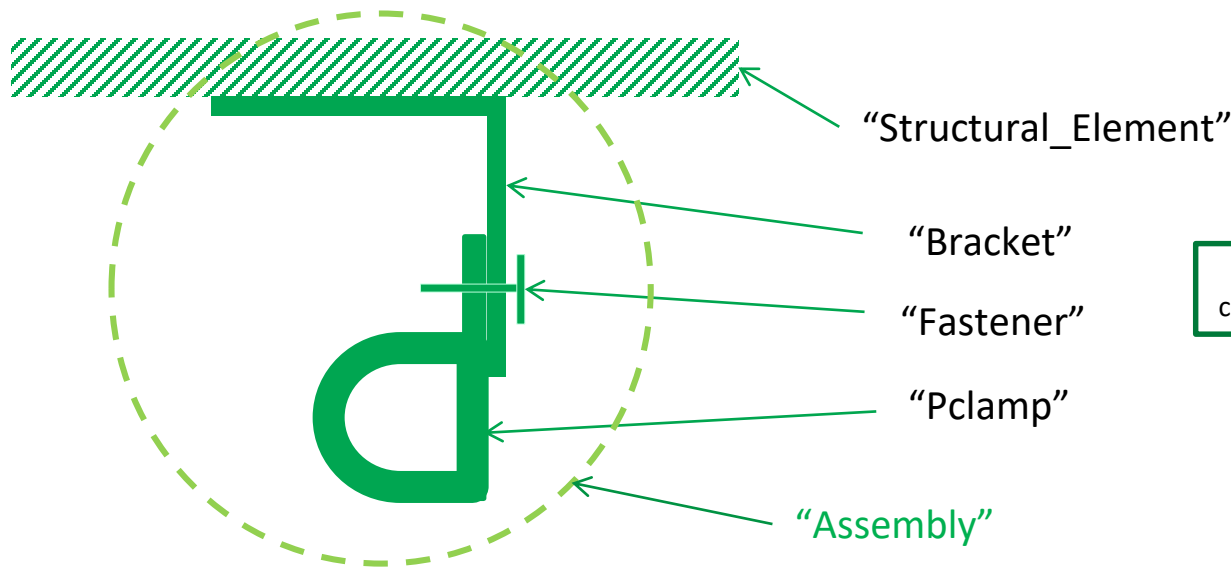
Note: Unless the bundle has a straight routing, L is bigger than the pitch between the Attachment Points.



Component Ontology and Rules

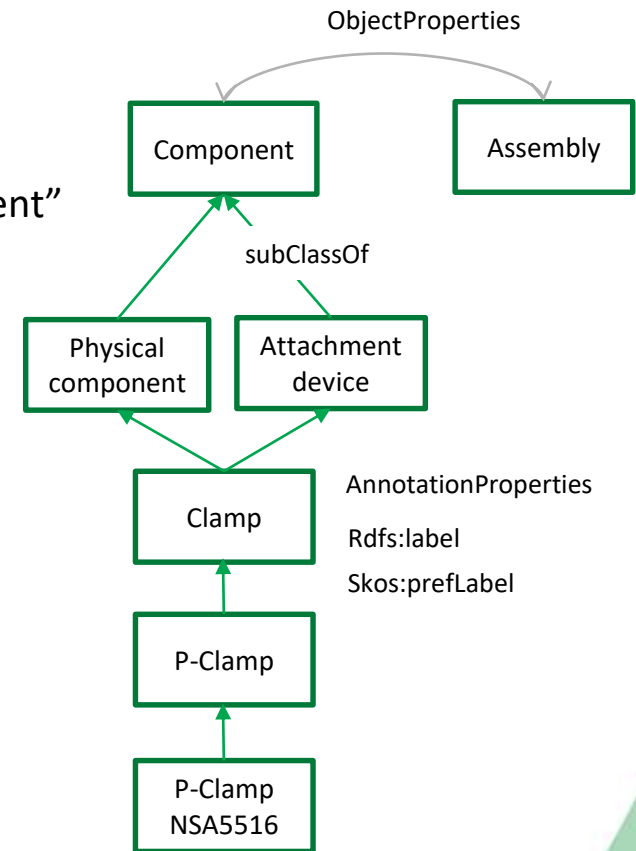
Objectives:

- Manage rules/design principles and improve traceability
- Automate identification of design conflicts against rules

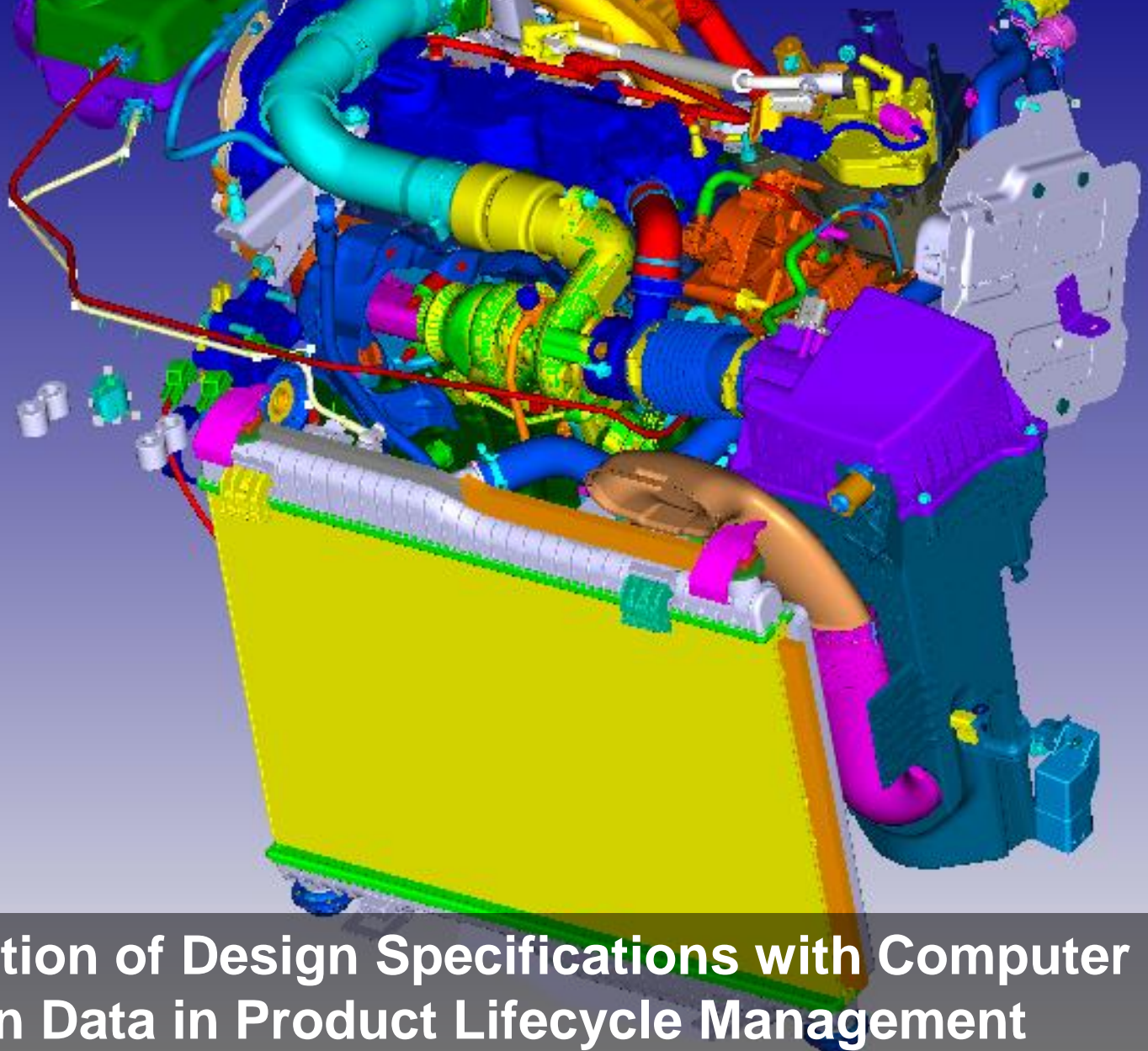


"P-clamp NSA5516 can be fixed on X with Y"

"Physical component" "Standard reference"



Industrial Use Case in Ford Otosan



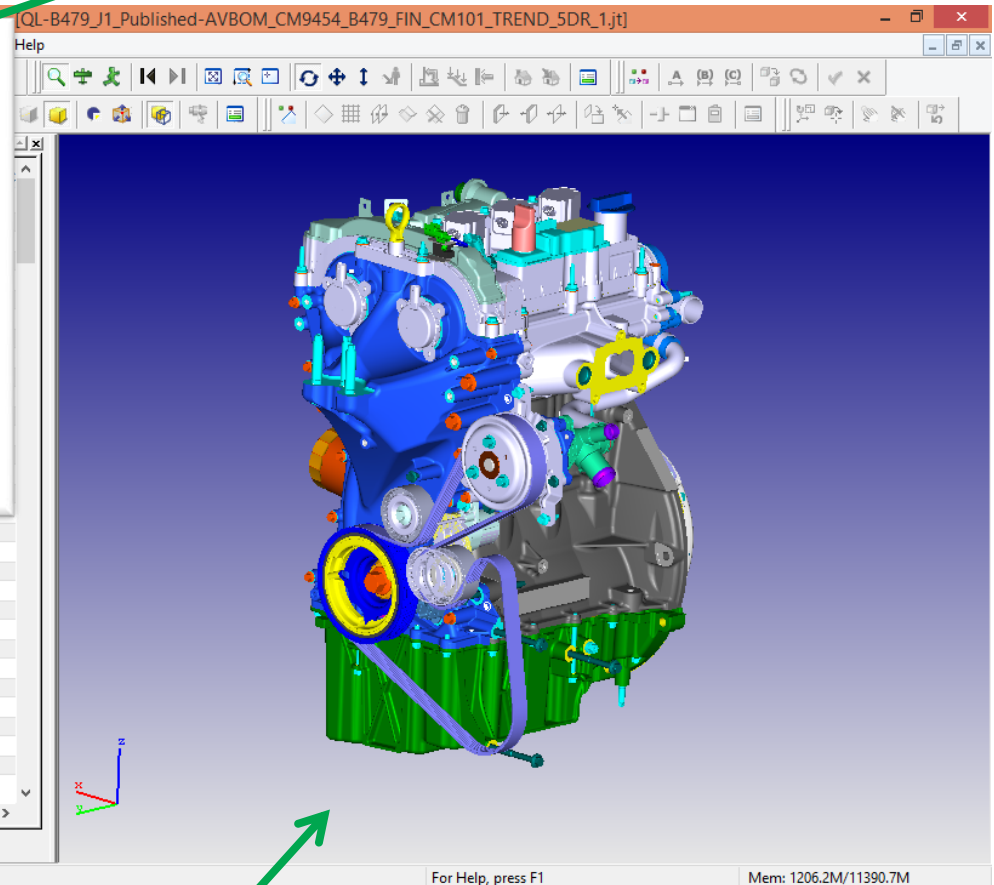
Synchronization of Design Specifications with Computer Aided Design Data in Product Lifecycle Management

BOM and Design Specifications

“Design Rules”

KPAC Set ID	Title	Author	Type
PT_FAST-S12	Cup Plug Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Dryseal Pipe Thread Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Dowel and Bushing Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Fastener and Joint Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Threaded Fastener Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Wrench and Socket Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Ball Plug Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Port Plug Design Rule Saved Set	Murphy, Mark (mmurphy9)	Check List
PT_FAST-S12	Thread Forming Screws in Plastic Saved Set	Murphy, Mark (mmurphy9)	Check List

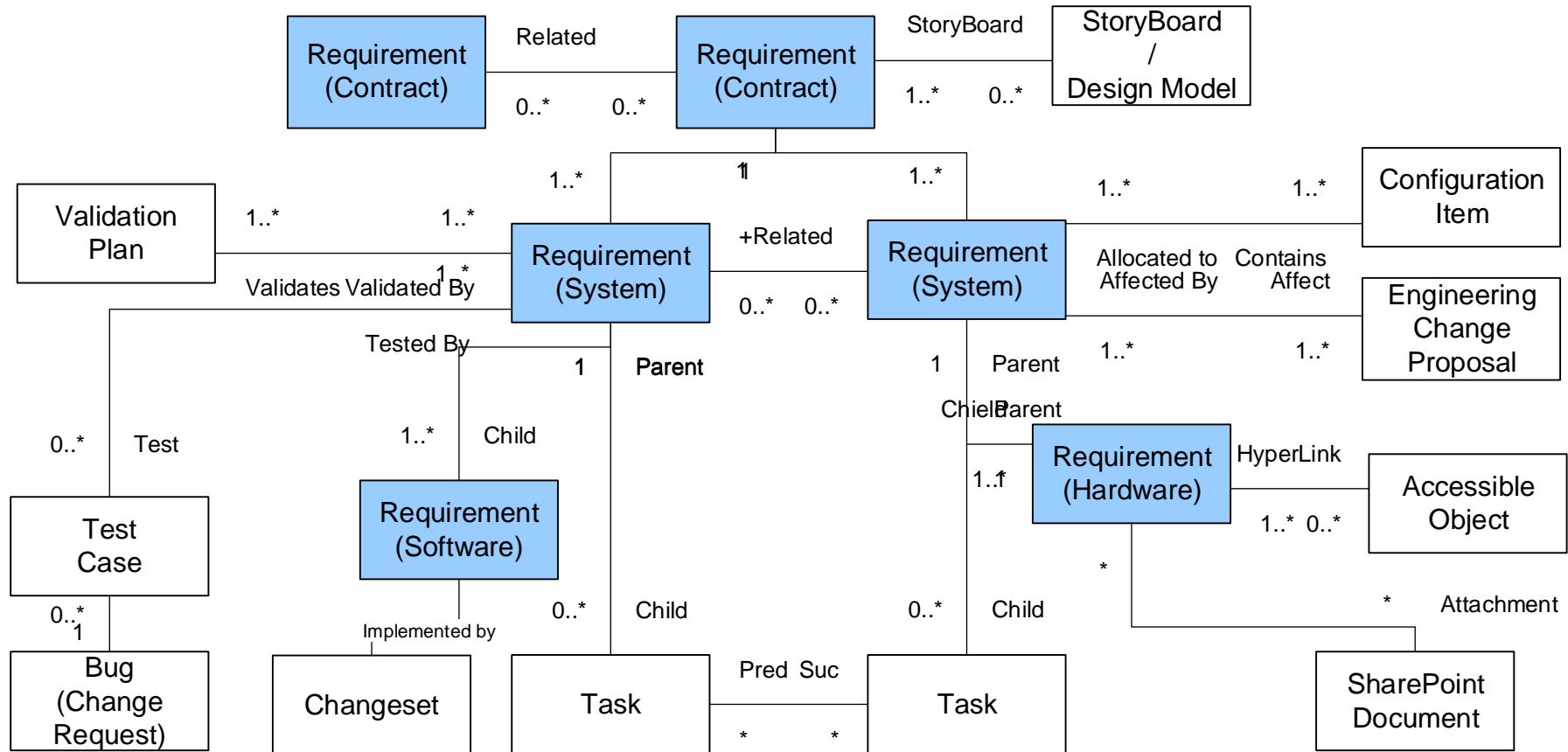
- ☐ PH-B479-031100/1-Engine Pwr Conversion Subsystem (View)
- ☐ PH-B479-031200/1-Air Charging Subsystem (View)
- ☐ PH-B479-031300/1-Evaporative Emissions Subsystem (View)
- ☐ PH-B479-031400/1-Electr Engine Control Subsystem (View)
- ☐ PH-B479-031500/1-Sound Control Subsystem (View)
- ☐ PH-B479-031600/1-Electr Throttle Control Subsys (View)
- ☐ PH-B479-031700/1-Electr Trct Motor Control Subsys (View)
- ☒ PH-B479-032100/1-Engine Modules (View)
- ☒ PH-B479-032101/2-Module - Power unit complete (View)
- ☒ PH-B479-032103/5-Module - Engine as Shipped (View)
- ☒ PAF-B479-WH7UL/3-Engine as Shipped - Hot Tested Assembly (V...
- ☒ DI-B479-18G-6007-A-WH7UL-001/2-ENG ASY1.1L FOX.TIVC...
- ☒ H18C-1007-A/38-FOX 1.1L PFI 85PS B479 EUR EAS (GASOLI...
- ☒ E3BG-020201-E40/2-OIL PUMP AND PICKUP (View)



“Bill of Material Data”

“Computer-aided Design Data”

Industrial Use Case in Havel-san

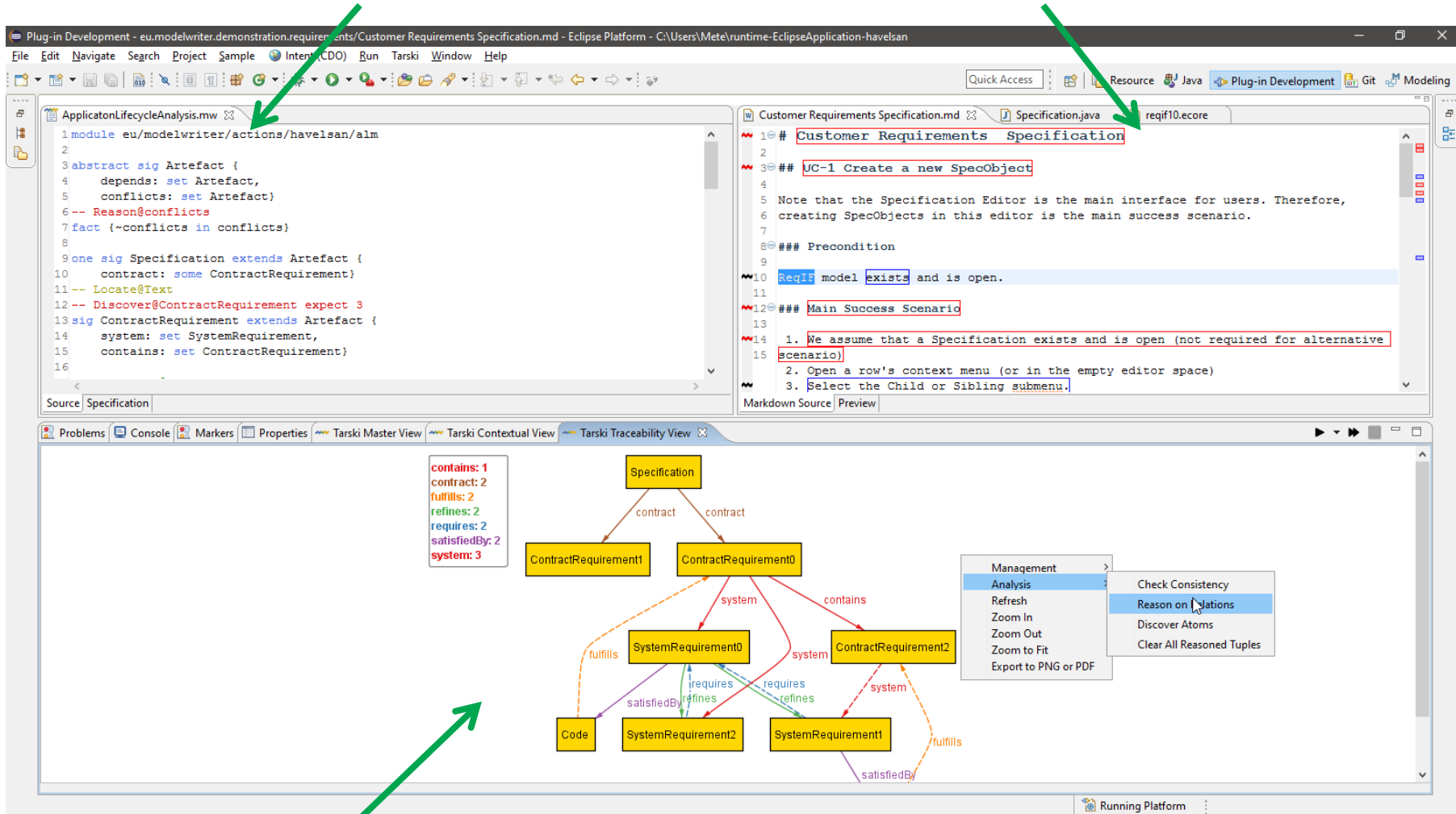


Integration with Application Lifecycle Management to ensure reliability and consistency in the system under development.

Automated Analysis of Dynamically Configured Traceability Semantics

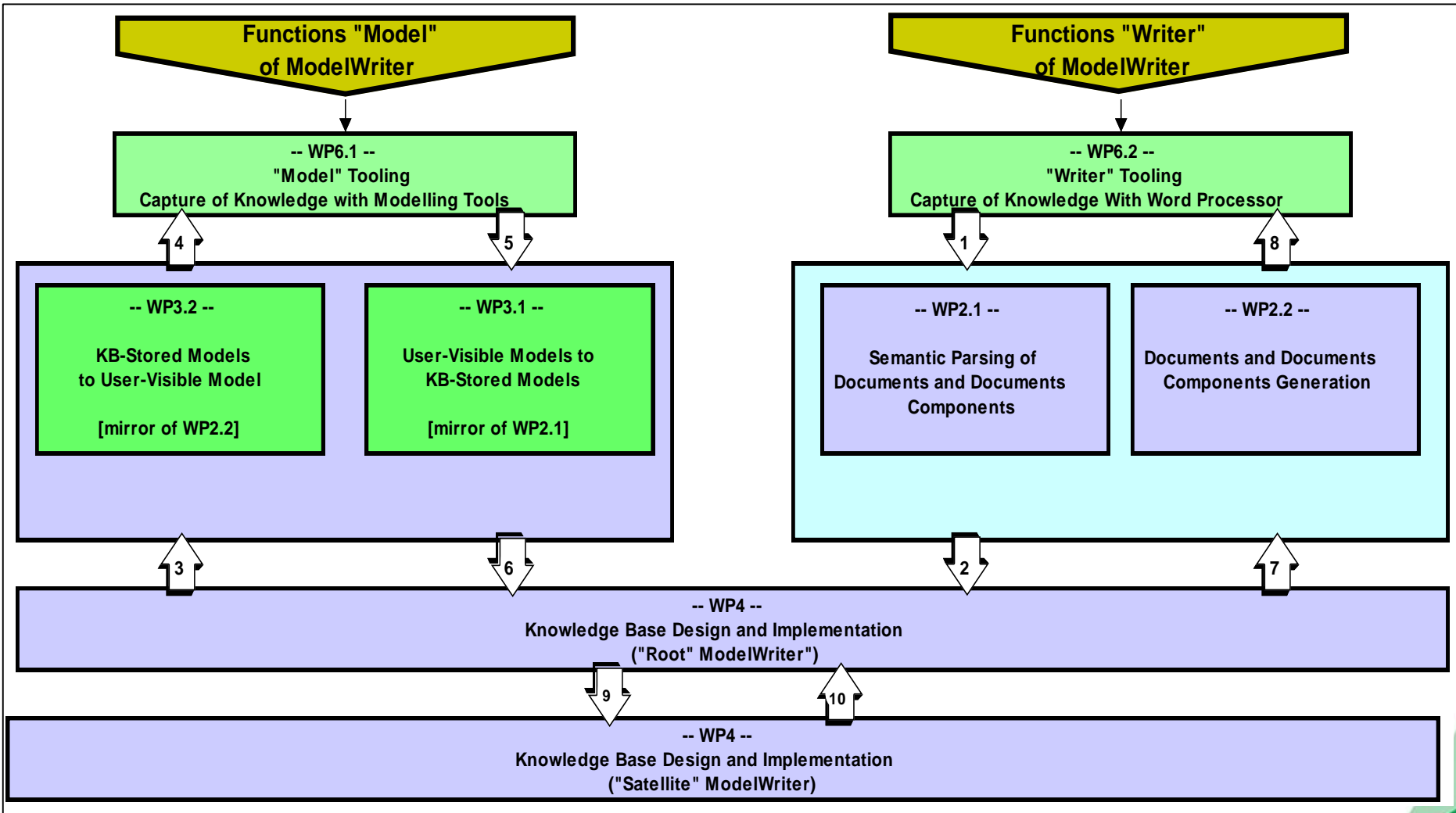
“Traceability Rules to define traceability semantics”

Artefacts or part of artefacts



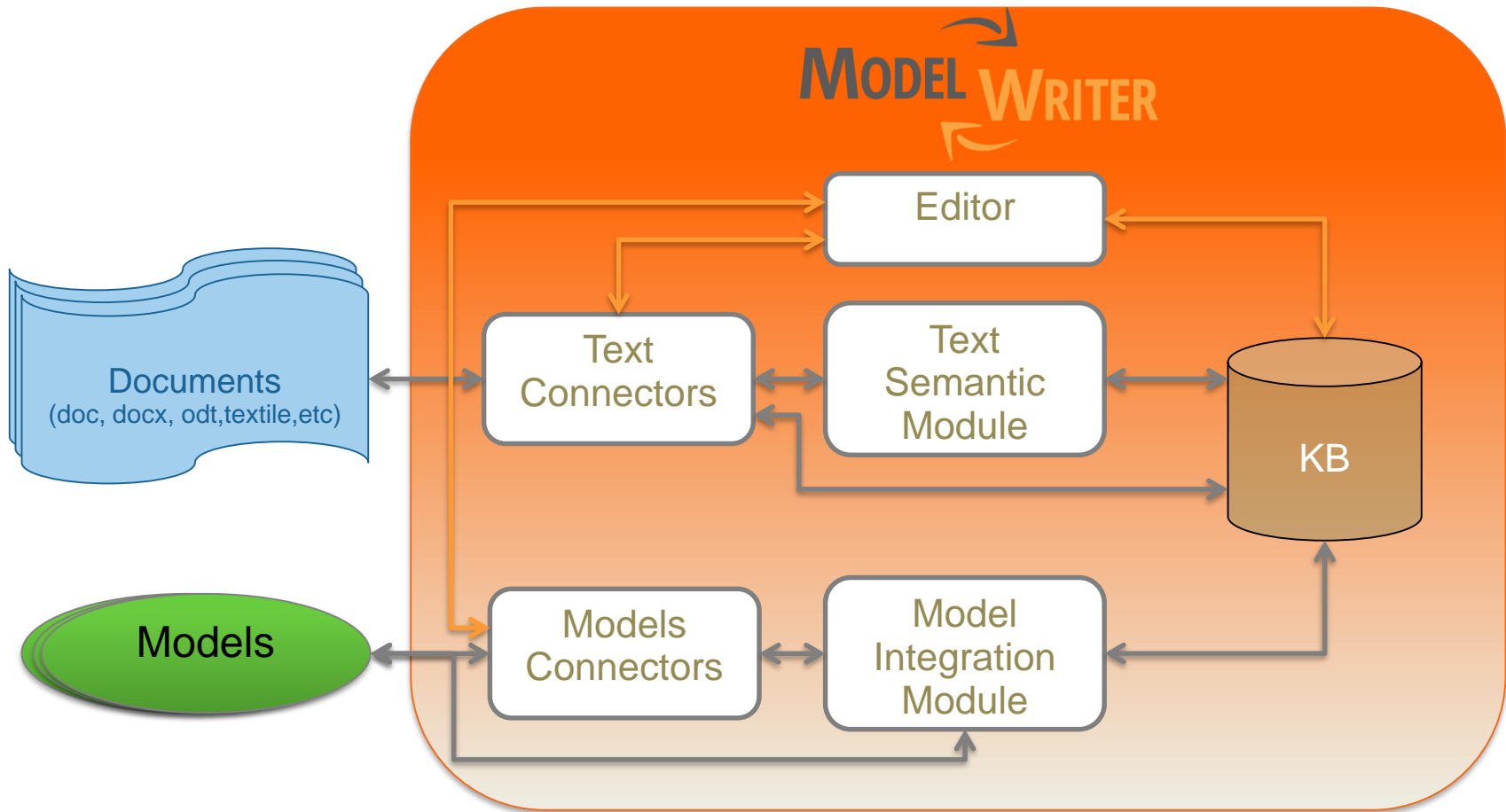
“Various Traceability Analysis might be performed”

Technological components & interactions



ModelWriter

Conceptual Architecture



Work Packages & Technical Innovations

WP1 Industrial Use Cases and Requirements (AIRBUS)

WP2 (LORIA)

- Semantic Parser
- Document Generation
- bi-directional transformation between text and formal knowledge representation

WP3 (UNIT)

- Bi-directional synchronization mechanism between texts and models
- Configuration & Traceability Components
- Consistency checker plug-in for consistency

WP4 (MANTIS)

- A federated Knowledge Base and its API
- Synchronization mechanism between texts/models & knowledge base

WP6 (OBEO)

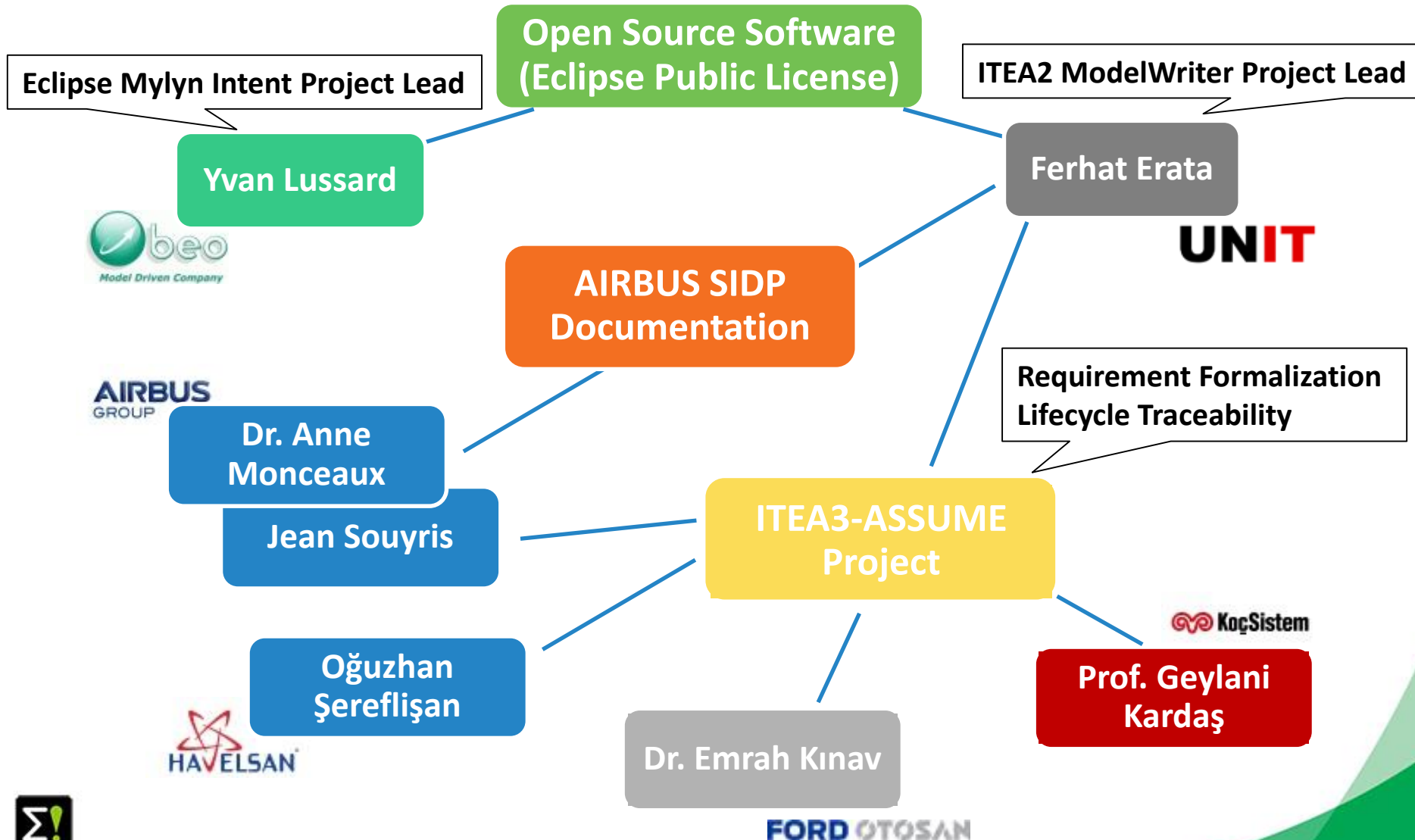
- A complete “ModelWriter” tool integrating of all these in a consistent way
- User Interfaces

WP5 Project Management (UNIT)

WP7 Standardization, Dissemination and Exploitation (OBEO)

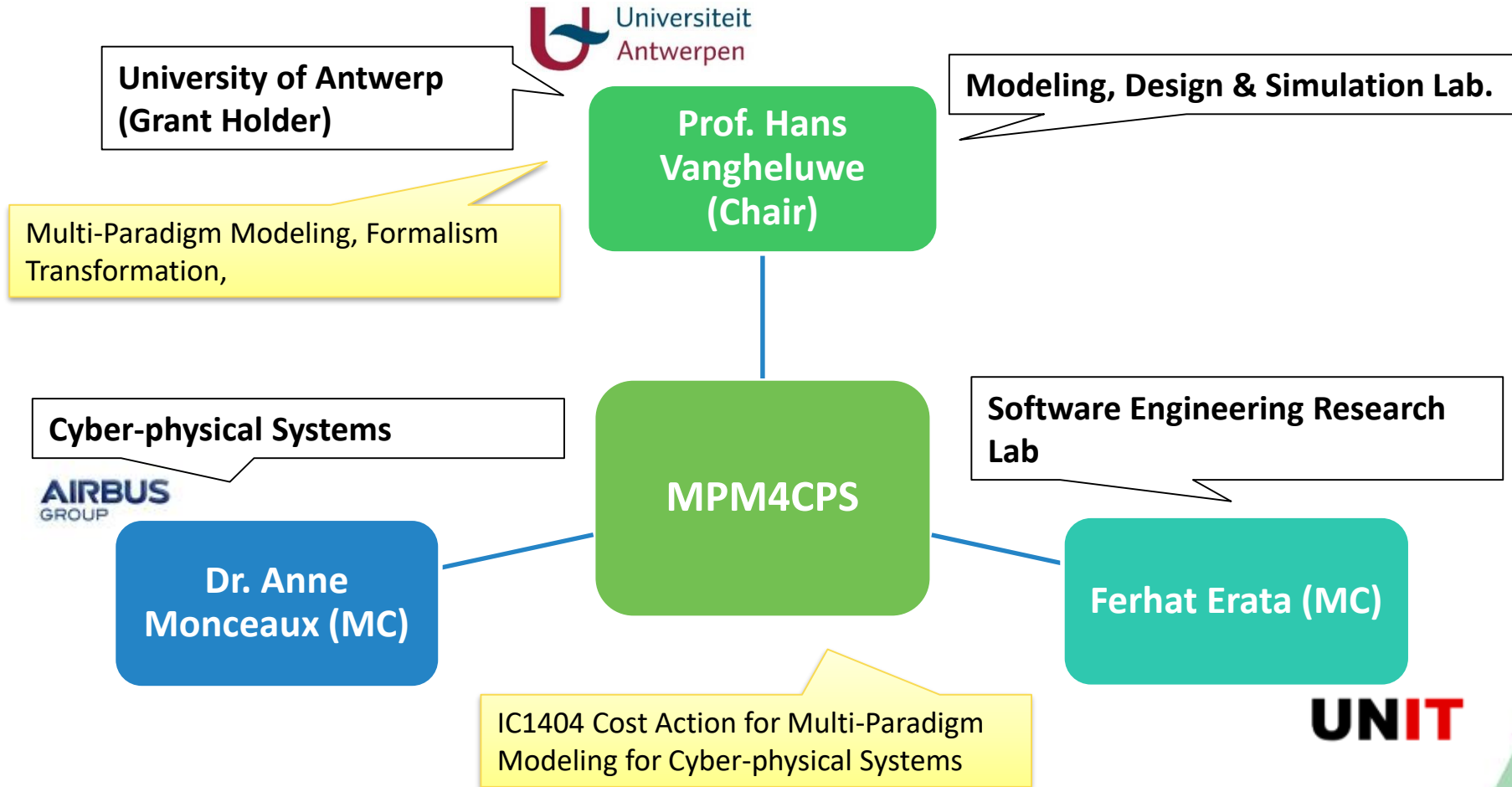
Level of Collaboration within ModelWriter

International Collaboration



Level of Collaboration within ModelWriter

International Collaboration



ModelWriter Activities in the First Year

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

Project Kick-off in Istanbul, Turkey (Nov 08, 2014) [M1]

Initial Architectural Design, Industrial Use Cases, Technical WP discussions

Collaboration Infrastructure

The 1st International ModelWriter Workshop in Izmir, Turkey (Jan 15-17, 2015) [M4]

Exploitation: Havelsan's participation

The 1st International Eureka Project Exhibition in Berlin, Germany (Mar 10-11, 2015) [M6]

Consolidated User Requirements & Review

The 2nd International ModelWriter Workshop in Brussels, Belgium (Apr 30, 2015) [M7]

Software Requirements Review & Architecture

The 3rd International ModelWriter Workshop in Toulouse, France (Jun 22-23, 2015) [M10]

Rehearsal & Review

The 4th International ModelWriter Workshop in Brussels, Belgium (Sep 23-24, 2015) [M12]

Integration of software components

The 5th International ModelWriter Workshop in Ludwigsburg, Germany (Nov 2-5, 2015) [M16]

ModelWriter Activities in the Second Year

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

The 6th International ModelWriter Workshop in Paris, France (Feb 15-16, 2016)



ICT Cost Action - MPM4CPS WG meeting at Vienna, Austria, on the 15-16 April, 2016



The 7th International ModelWriter Technical Workshop in Toulouse, France (6 June 2016)



The 7th Int'l ModelWriter Brainstorming Session in Toulouse, France at Airbus (9 June 2016)



The 7th Int'l ModelWriter Coordination Meeting in Toulouse, France at Airbus (10 June 2016)



ModelWriter Poster Presentation SAT/SMT/AR Summer School 2016



Participation in International Joint Conference on Automated Reasoning (IJCAR) 2016

ModelWriter Workshops in the First Year

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



Verification Technology, Systems & Applications (VTSA) Summer School (Aug 29- Sept 02, 2016)



ICT Cost Action - MPM4CPS WG meeting at Gdansk, Poland (Sep 13-16, 2016)



A paper is submitted to ACM Applied Computing Symposium and under review.



Participation in Workshop on Software Correctness and Reliability (Oct 7-8 2016)



The 8th International ModelWriter Technical Workshop in Toulouse, France (13 June 2016)

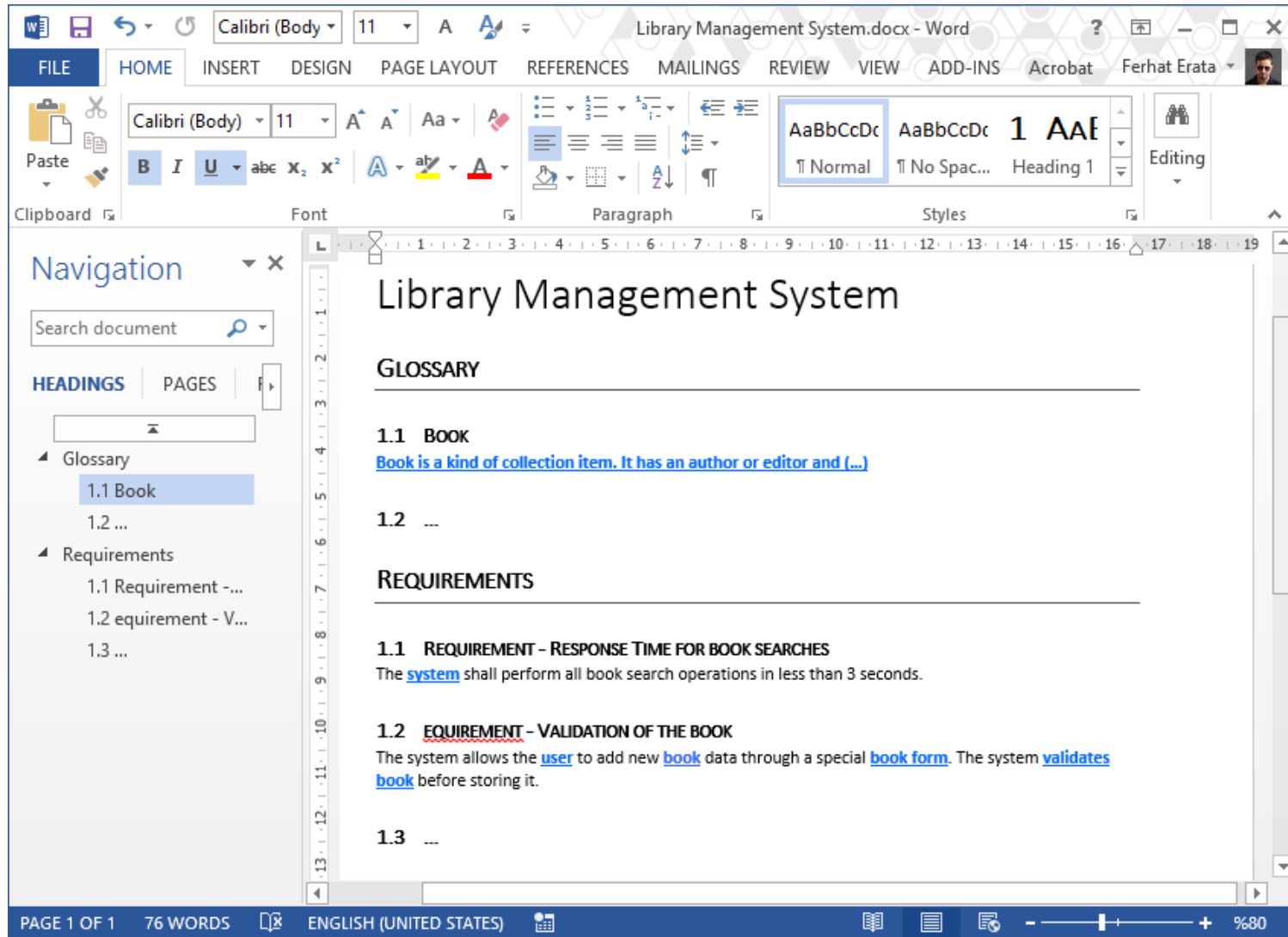


ICT Cost Action - MPM4CPS WG meeting at Malaga, Spain (Nov 24-25, 2016)

What is a text?

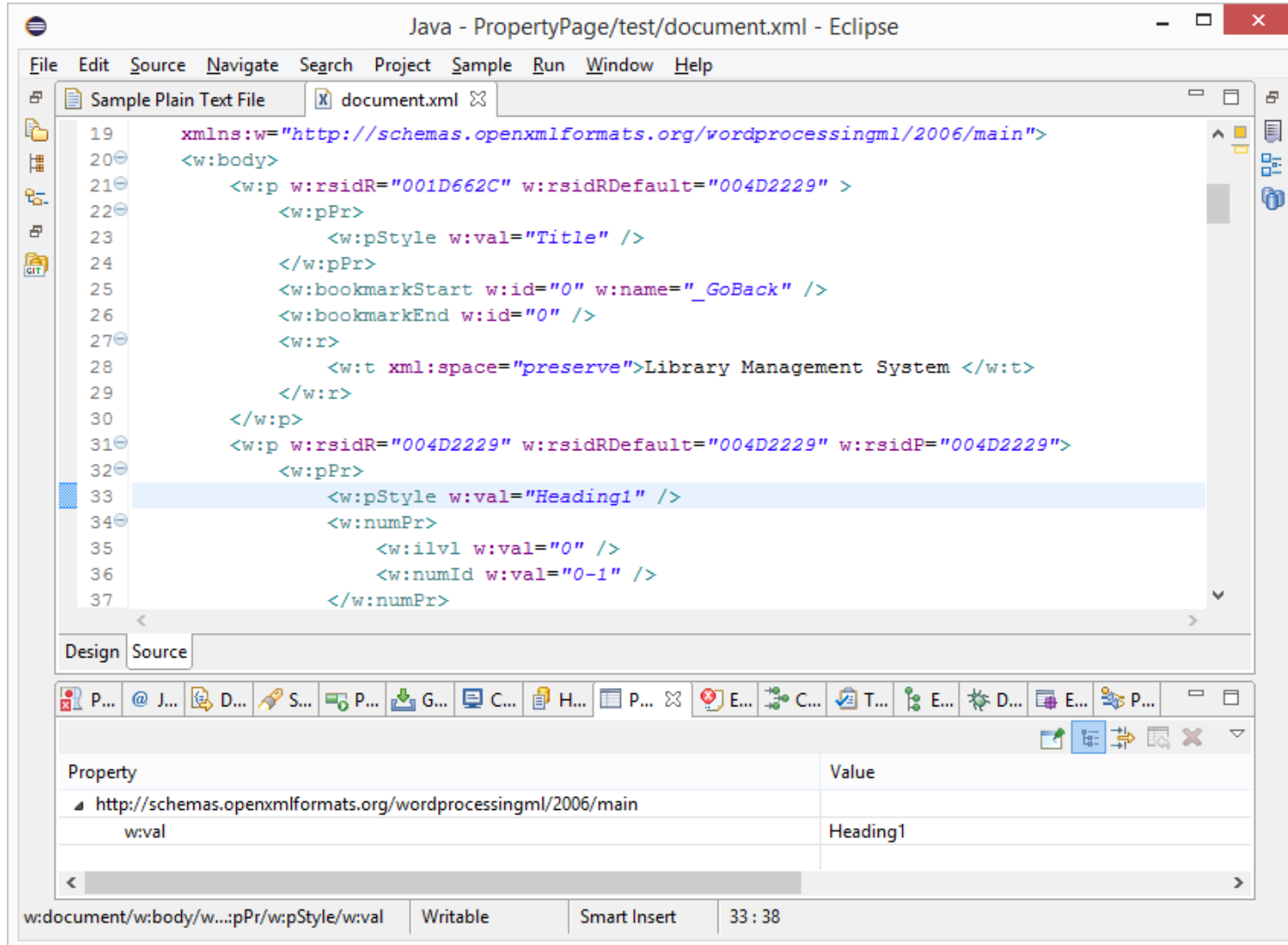
What is a text? (document file formats)

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



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>

```

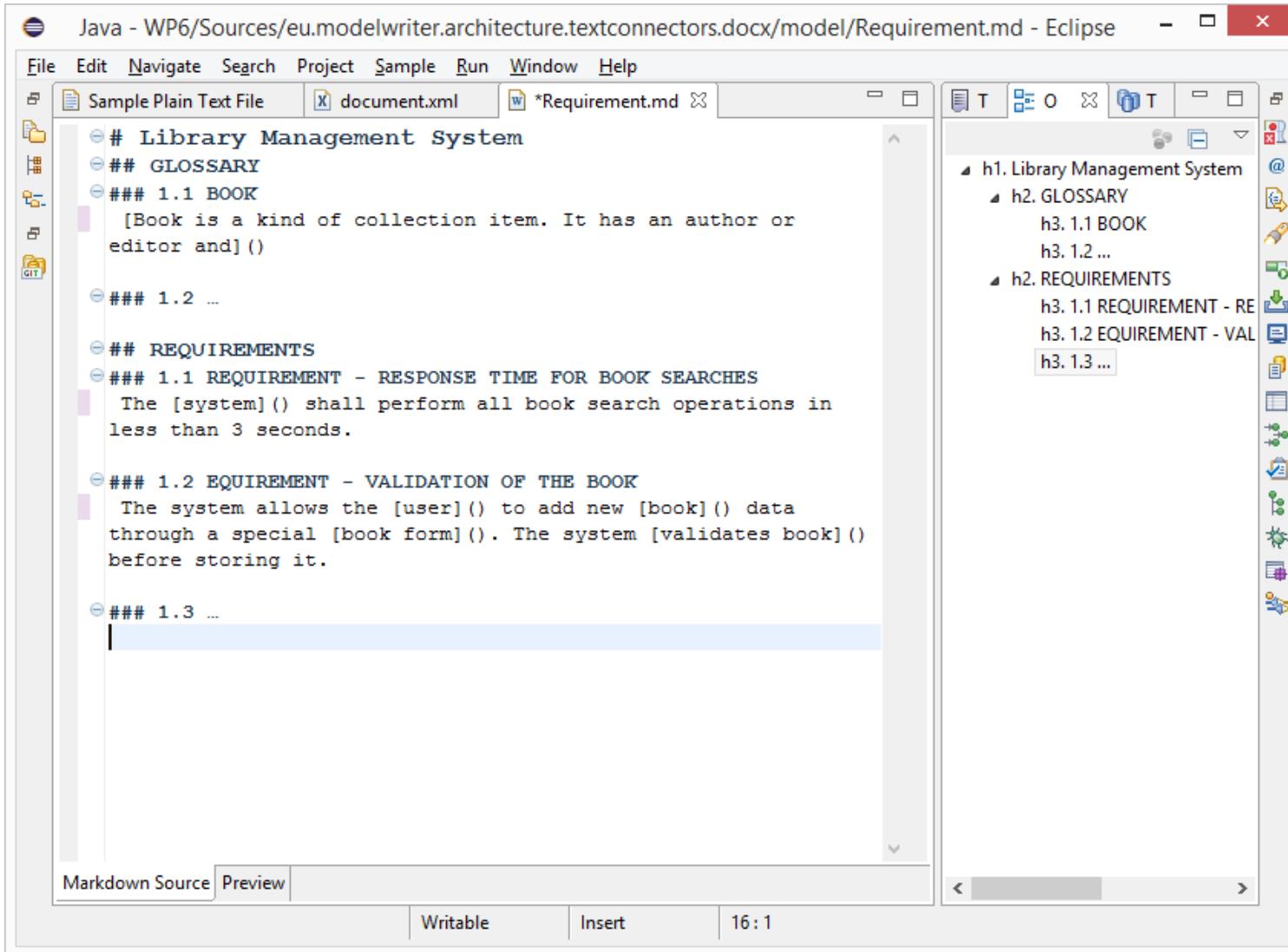
Design Source

Property	Value
http://schemas.openxmlformats.org/wordprocessingml/2006/main	
w:val	Heading1

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

What is a text? (.md source file)

text/markdown (ICANN Standard)



The screenshot shows the Eclipse IDE interface. The main editor window displays a markdown file named `*Requirement.md`. The content of the file is as follows:

```
# 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 EQUIREMENT - 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 ...
```

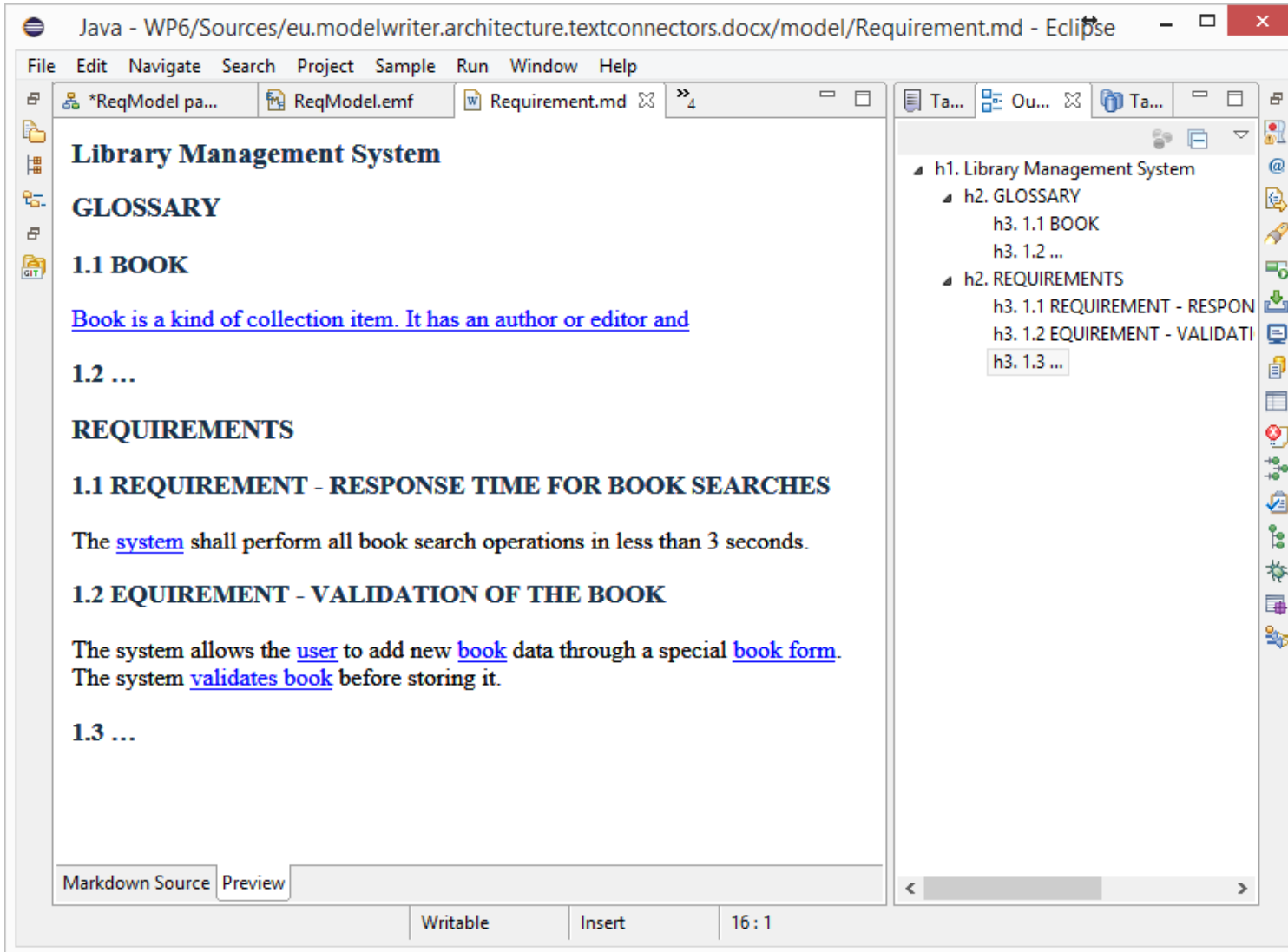
The right-hand side of the IDE shows a tree view of the document structure, listing the following hierarchy:

- 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 ...

The status bar at the bottom indicates the current mode is "Markdown Source", with other options like "Preview", "Writable", "Insert", and a cursor position of "16:1".

What is a text? (HTML Preview)

text/markdown (ICANN Standard)



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 EQUIREMENT - 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 ...

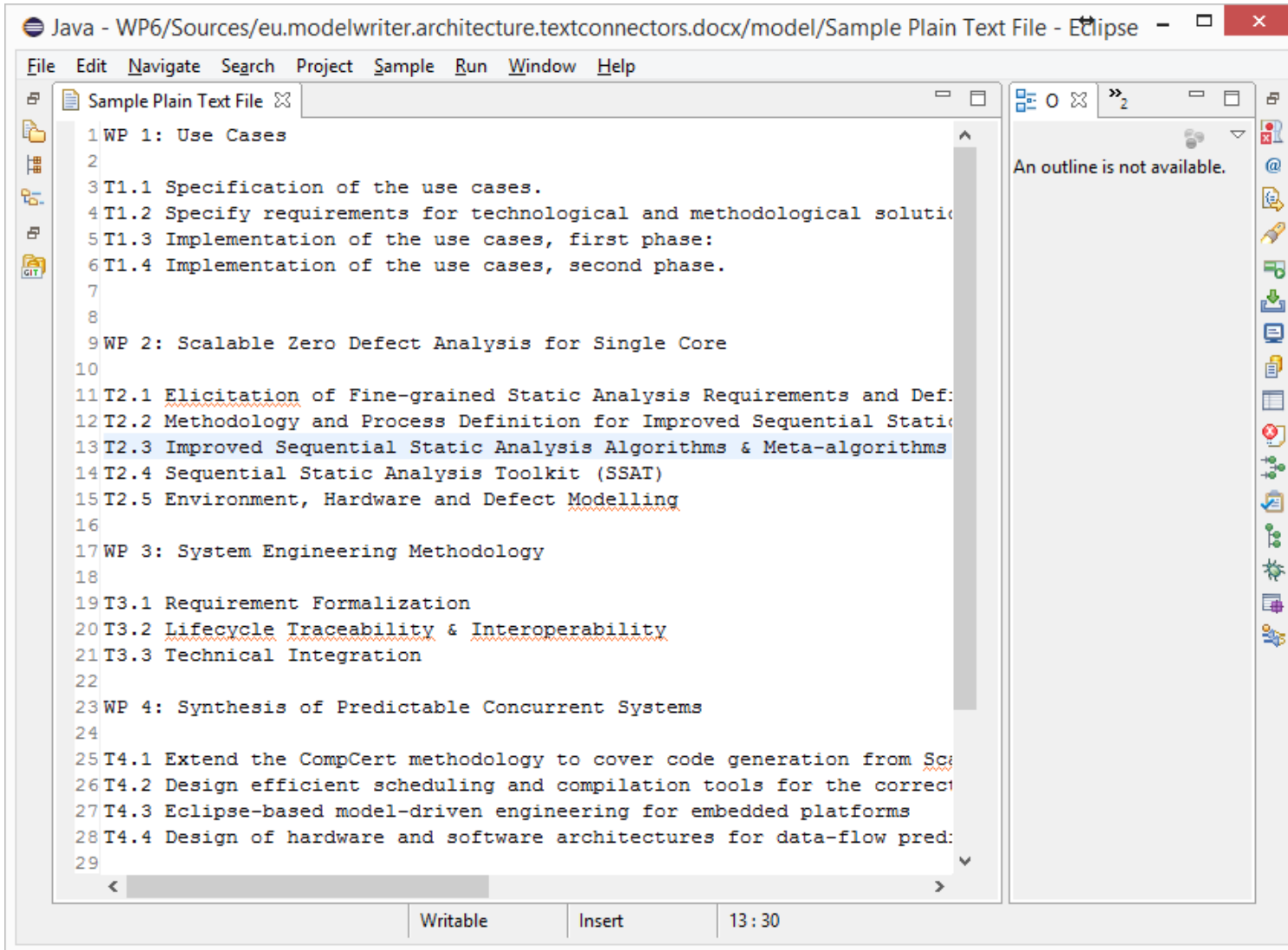
- h1. Library Management System
 - h2. GLOSSARY
 - h3. 1.1 BOOK
 - h3. 1.2 ...
 - h2. REQUIREMENTS
 - h3. 1.1 REQUIREMENT - RESPON
 - h3. 1.2 EQUIREMENT - VALIDATI
 - h3. 1.3 ...

Markdown Source Preview

Writable Insert 16 : 1

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

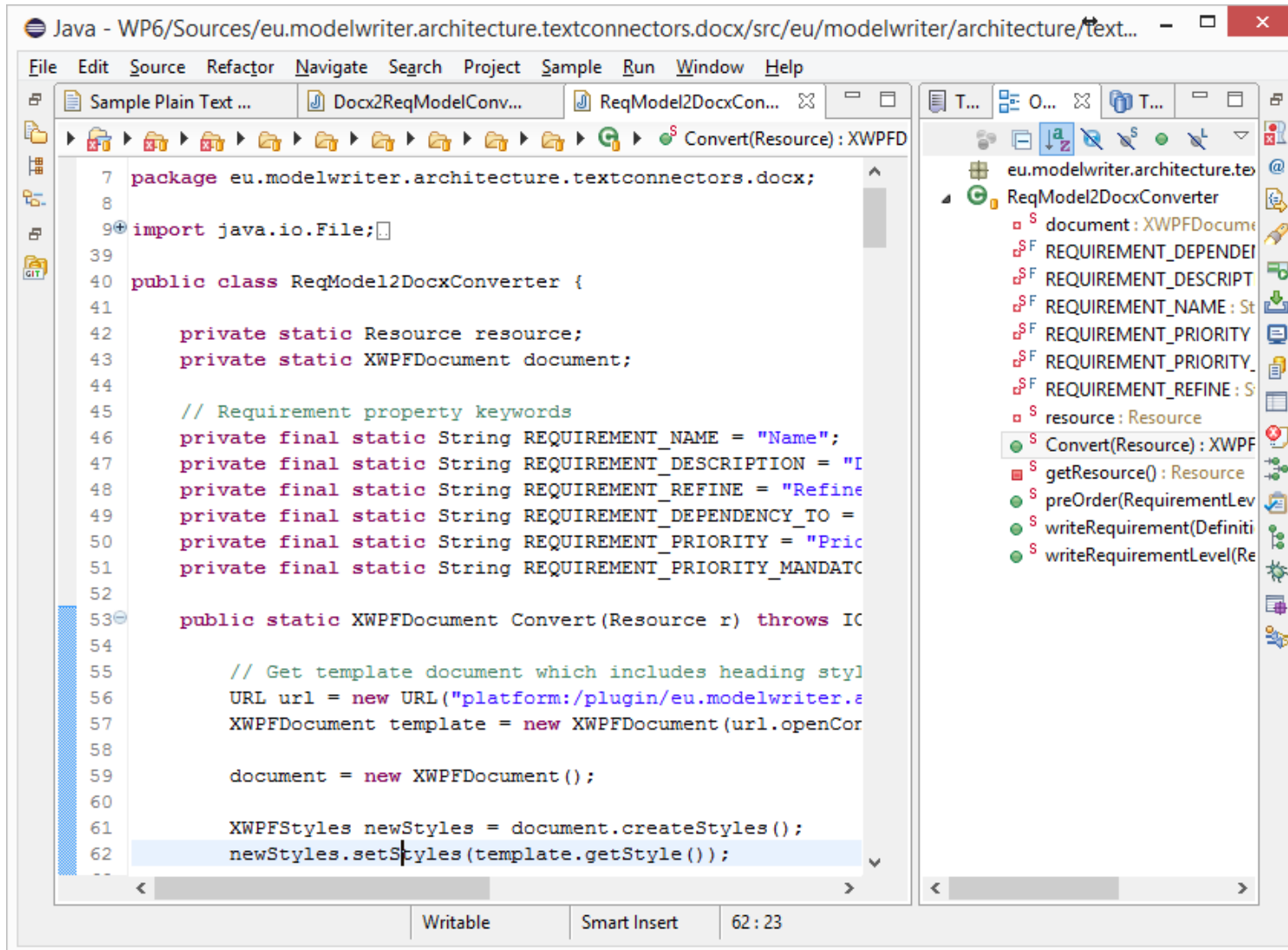
```
1 WP 1: Use Cases
2
3 T1.1 Specification of the use cases.
4 T1.2 Specify requirements for technological and methodological solutions
5 T1.3 Implementation of the use cases, first phase:
6 T1.4 Implementation of the use cases, second phase.
7
8
9 WP 2: Scalable Zero Defect Analysis for Single Core
10
11 T2.1 Elicitation of Fine-grained Static Analysis Requirements and Definitions
12 T2.2 Methodology and Process Definition for Improved Sequential Static Analysis
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 Specifications
26 T4.2 Design efficient scheduling and compilation tools for the correct compilation
27 T4.3 Eclipse-based model-driven engineering for embedded platforms
28 T4.4 Design of hardware and software architectures for data-flow prediction
29
```

An outline is not available.

Writable Insert 13:30

What is a text? (code files)

Java, C++ ... Programing Languages



The screenshot shows an IDE window titled "Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/src/eu/modelwriter/architecture/text...". The main editor displays the following Java code:

```
7 package eu.modelwriter.architecture.textconnectors.docx;
8
9 import java.io.File;
10
39
40 public class ReqModel2DocxConverter {
41
42     private static Resource resource;
43     private static XWPFDocument document;
44
45     // Requirement property keywords
46     private final static String REQUIREMENT_NAME = "Name";
47     private final static String REQUIREMENT_DESCRIPTION = "Description";
48     private final static String REQUIREMENT_REFINE = "Refine";
49     private final static String REQUIREMENT_DEPENDENCY_TO = "Dependency To";
50     private final static String REQUIREMENT_PRIORITY = "Priority";
51     private final static String REQUIREMENT_PRIORITY_MANDATORY = "Mandatory";
52
53     public static XWPFDocument Convert(Resource r) throws IOException {
54
55         // Get template document which includes heading styles
56         URL url = new URL("platform:/plugin/eu.modelwriter.architecture.textconnectors.docx/ReqModel2DocxConverter.template.docx");
57         XWPFDocument template = new XWPFDocument(url.openConnection().getInputStream());
58
59         document = new XWPFDocument();
60
61         XWPFFont newStyles = document.createStyles();
62         newStyles.setStyles(template.getStyle());
63     }
64 }
```

The right sidebar shows a project explorer with the following structure:

- eu.modelwriter.architecture.textconnectors.docx
 - ReqModel2DocxConverter
 - document: XWPFDocument
 - REQUIREMENT_DEPENDENCY_TO
 - REQUIREMENT_DESCRIPTION
 - REQUIREMENT_NAME: String
 - REQUIREMENT_PRIORITY
 - REQUIREMENT_PRIORITY_MANDATORY
 - 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 ProR

My.reqif Requirements Document

ID	Name	Description
1		
1.1	Librarian	Librarian
1.2	Response Time for Book Searches	The system shall perform all book search operations in less than 3 second
1.3	Add new Book	A user form to add new book
1.4	Validation of the Book	Validation of the Book

Outline

- Spec Hierarchy
 - Librarian
 - The system shall
 - A user form to
 - Validation of th

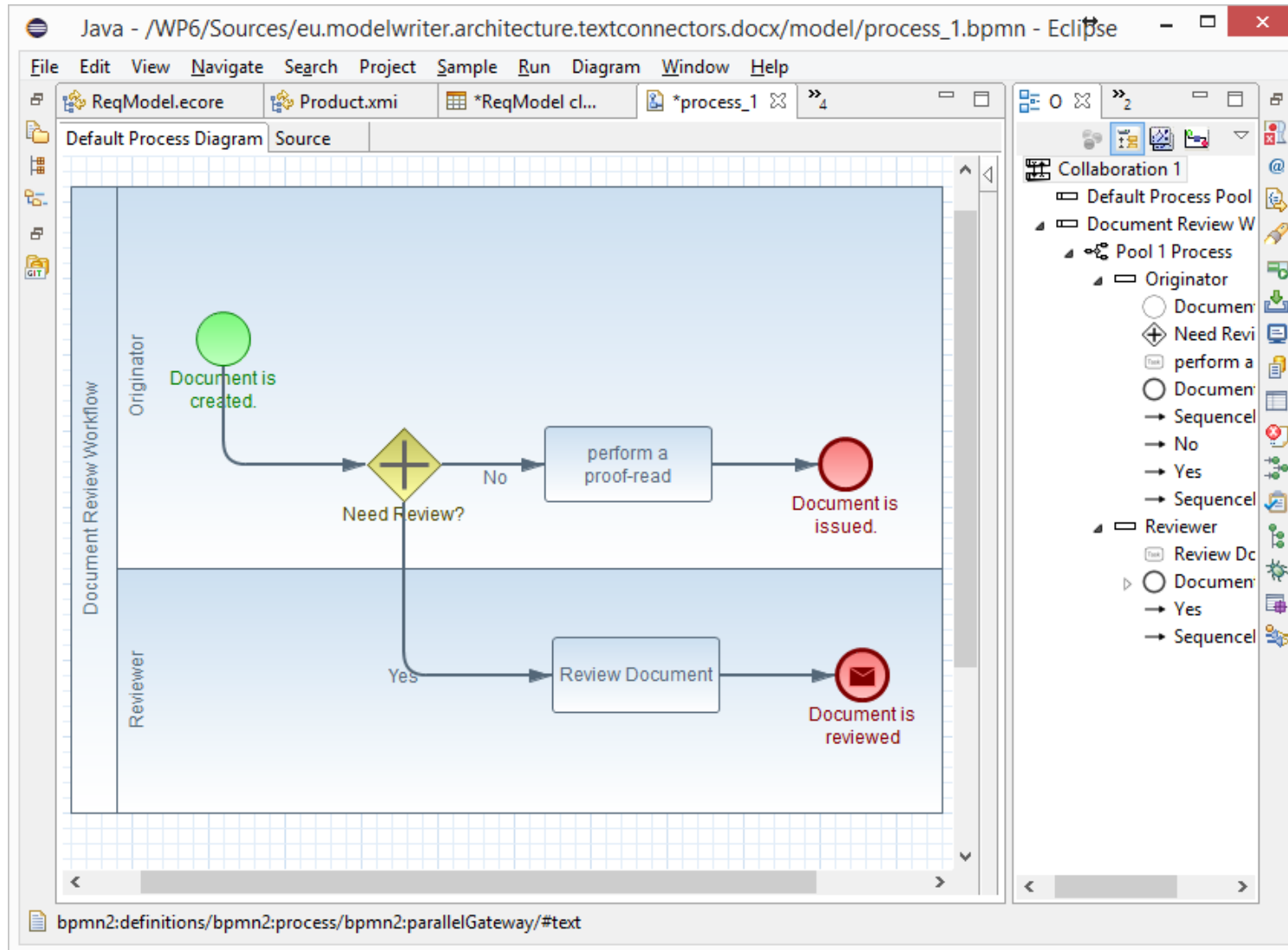
Properties

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

Standard Attributes All Attributes

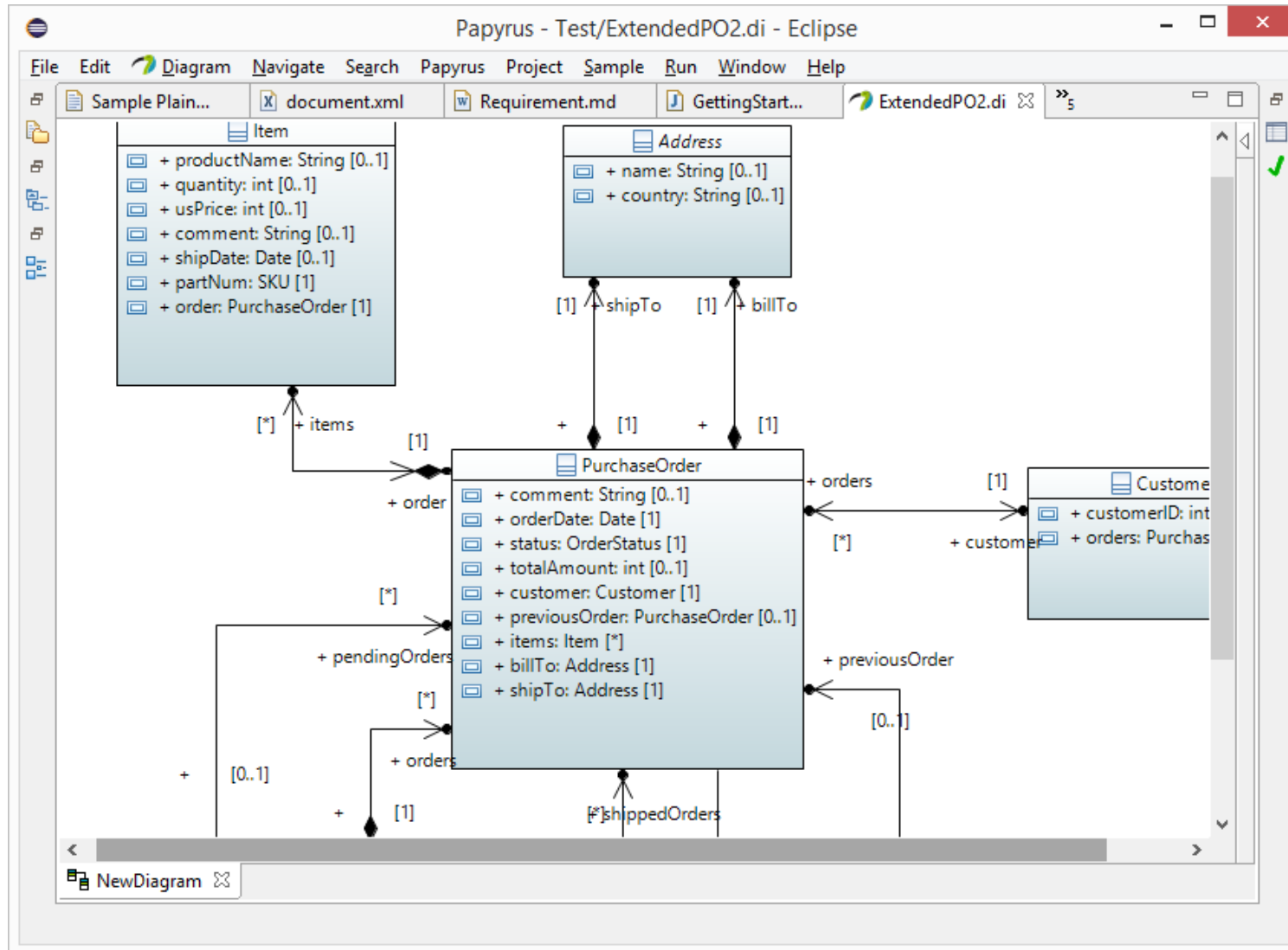
Everything is a model! (BPMN Standard)

Business Process Model & Notation



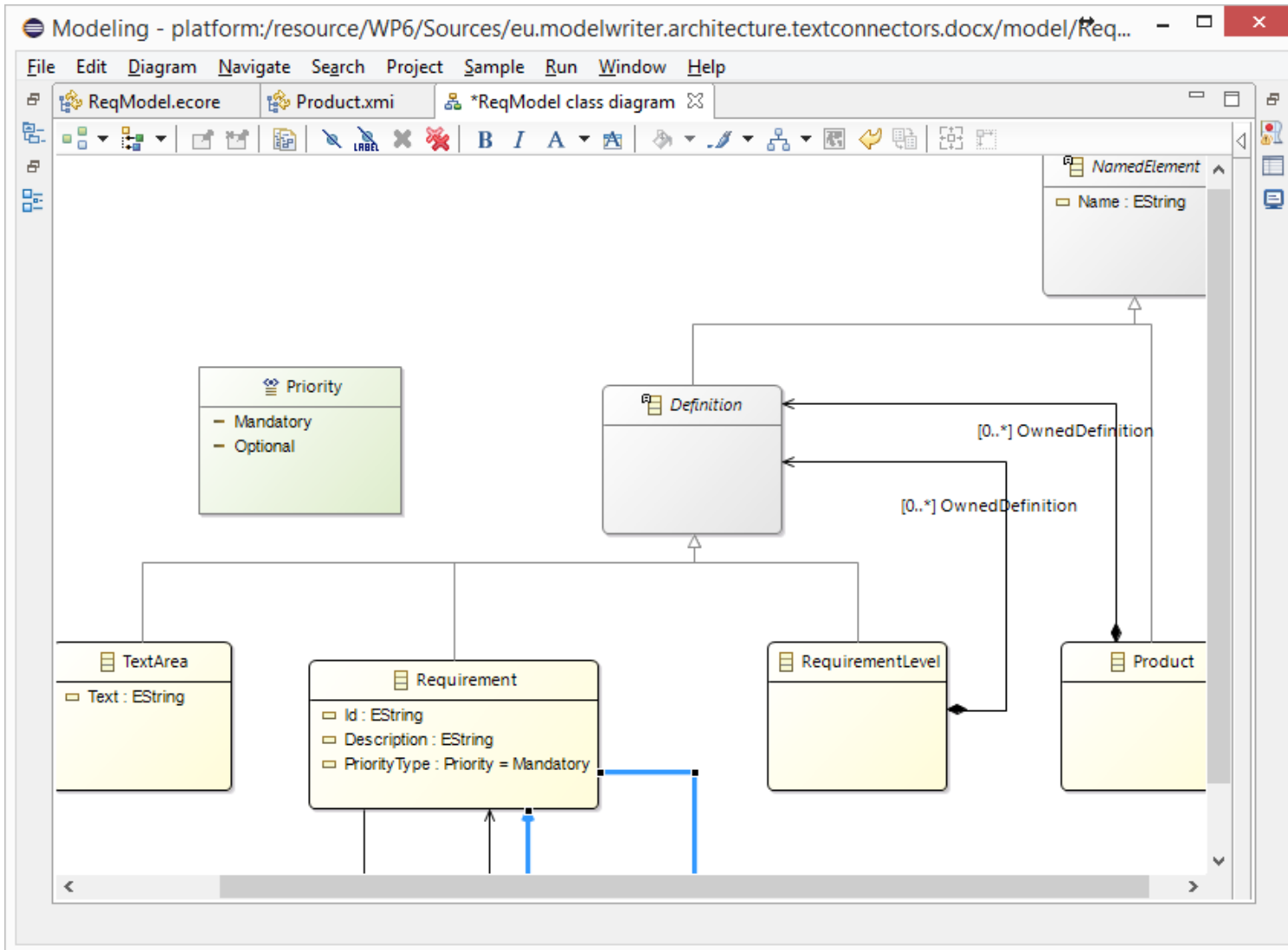
Everything is a model! (UML Standard)

UML Modeling Languages



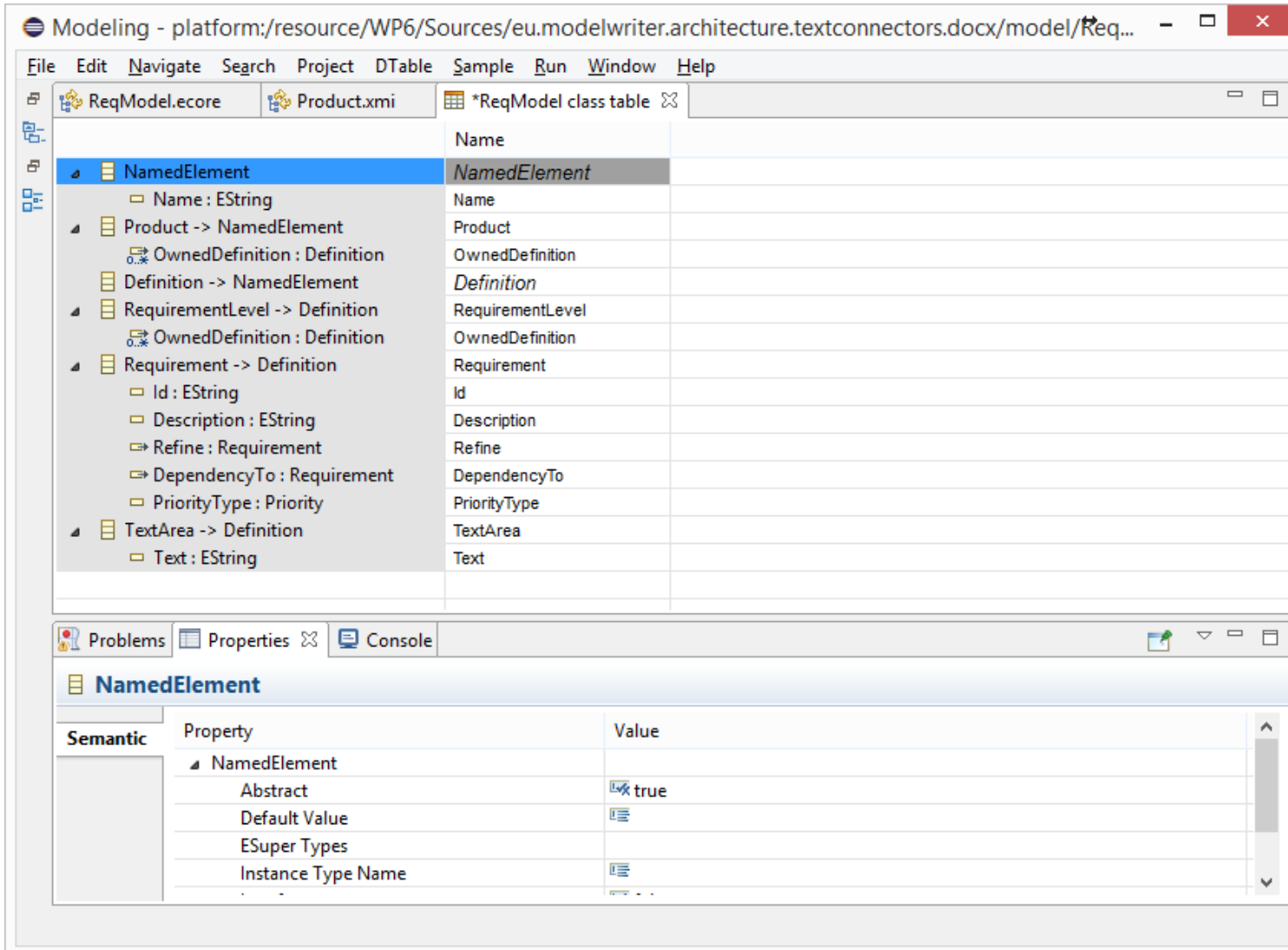
Everything is a model!

Eclipse Modeling Framework (EMF)



Everything is a model!

Tree-based or Tabular Representations



The screenshot shows the Modeling IDE interface. The main window displays a tree-based representation of a model. The tree structure is as follows:

- 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

The right pane shows the *ReqModel class table with the following data:

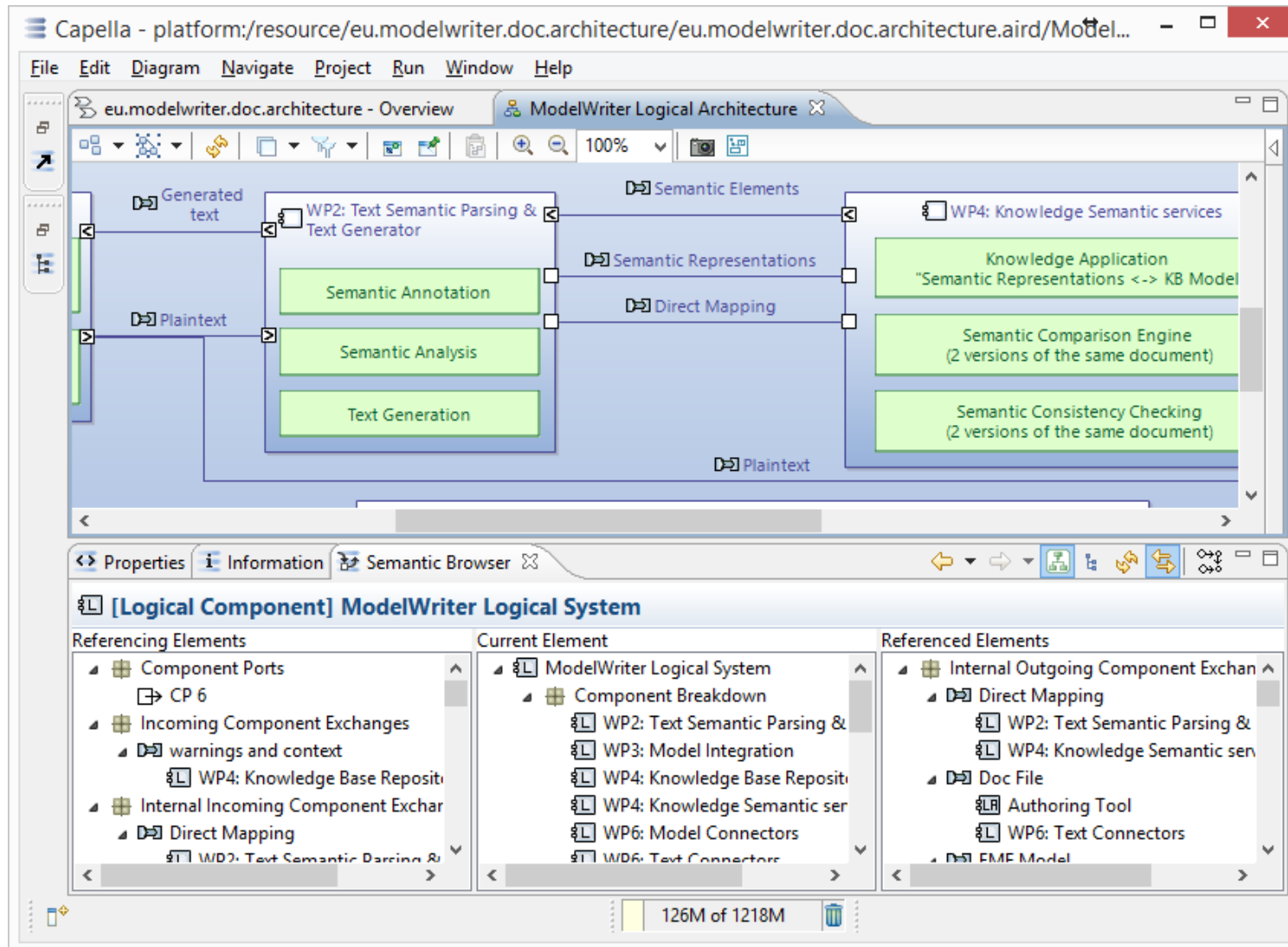
Name
NamedElement
Name
Product
OwnedDefinition
Definition
RequirementLevel
OwnedDefinition
Requirement
Id
Description
Refine
DependencyTo
PriorityType
TextArea
Text

The bottom pane shows the Properties view for the NamedElement. The Semantic tab is selected, showing the following properties:

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

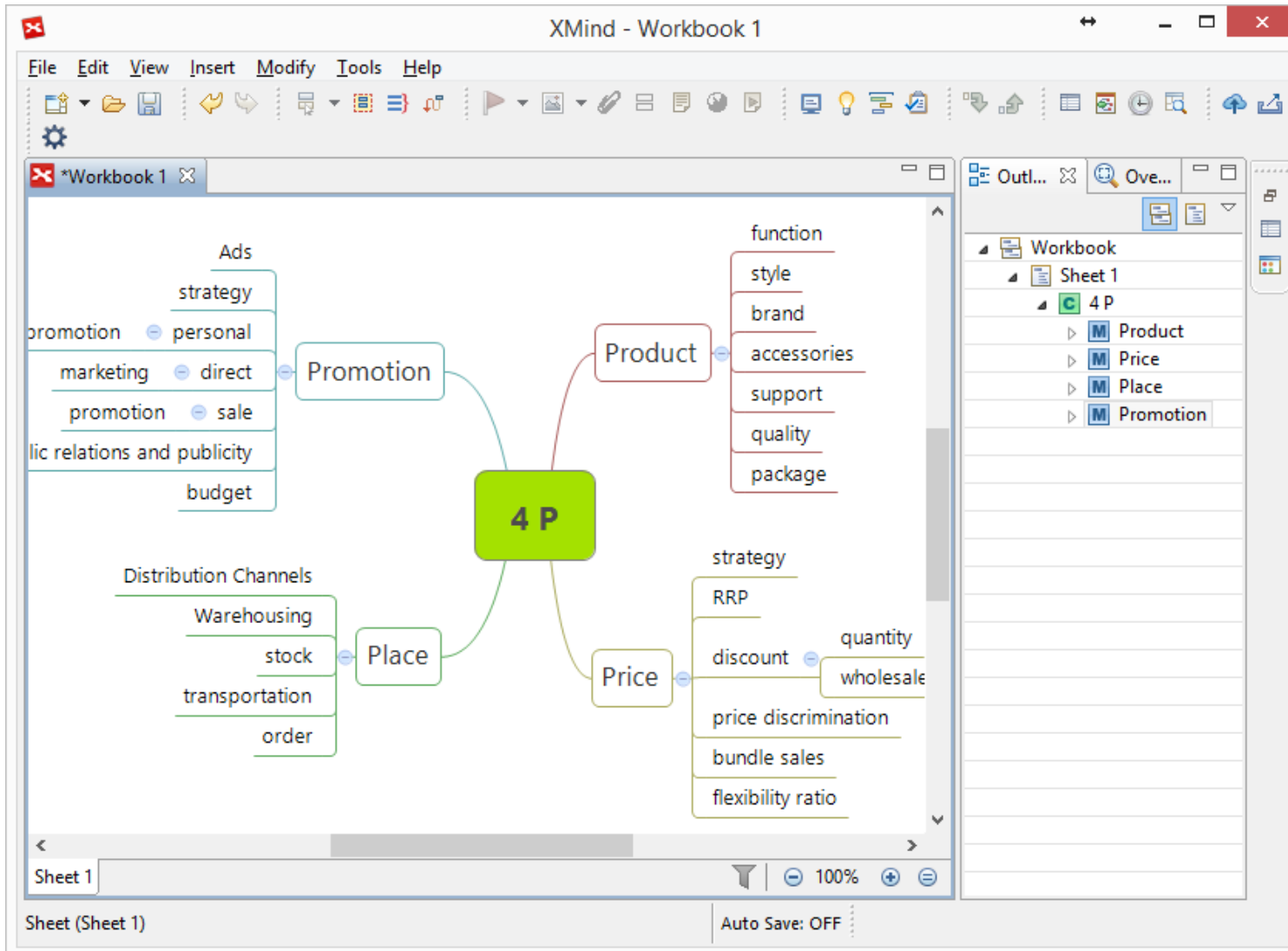
Everything is a model!

Software/System Architecture Design



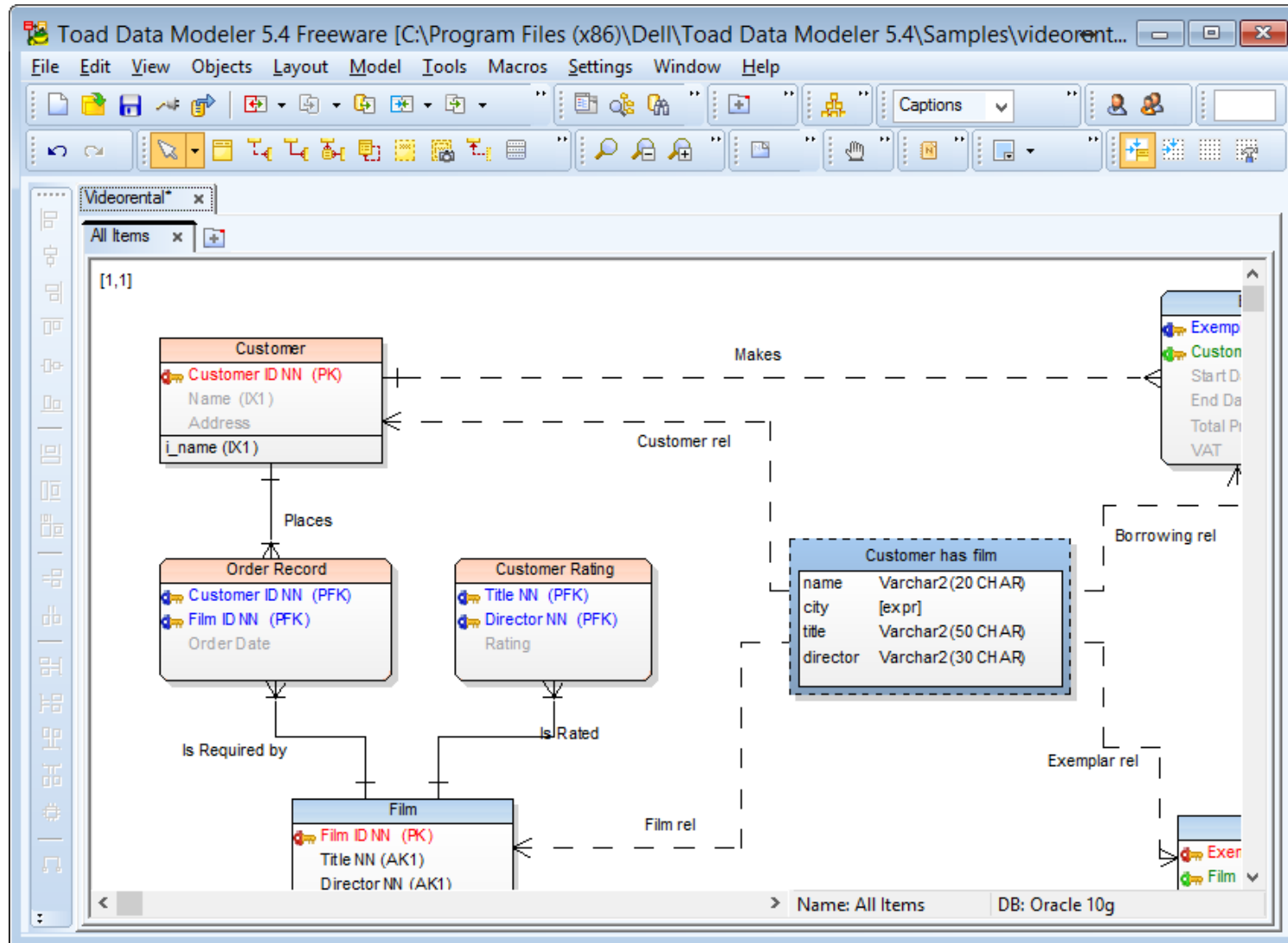
Everything is a model!

Topic Maps, Mind Maps, Vocabularies ...



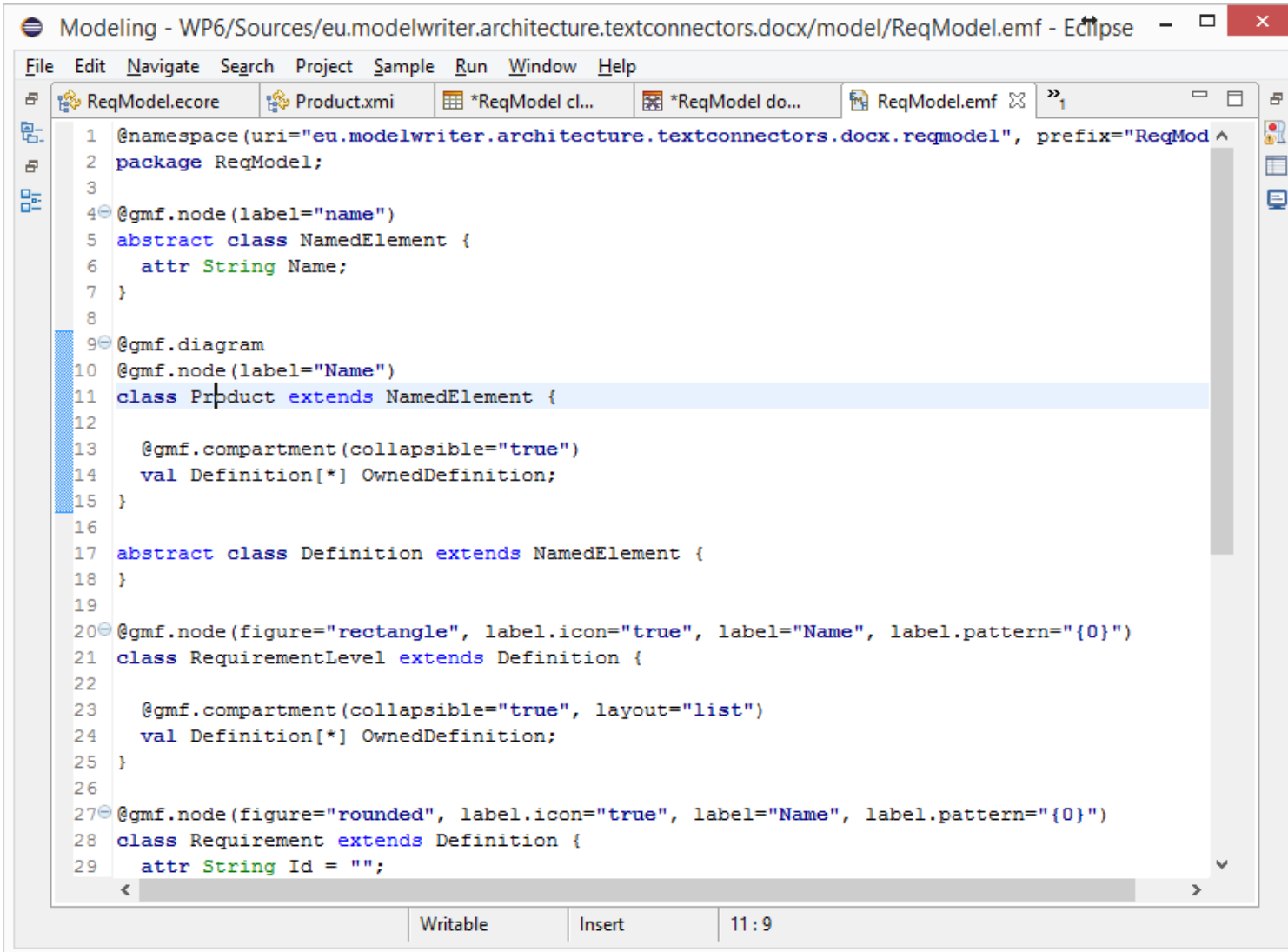
Everything is a model!

Databases (ER, IDEF1.x)



Everything is a model! (Textual Lang.)

Domain Specific Languages



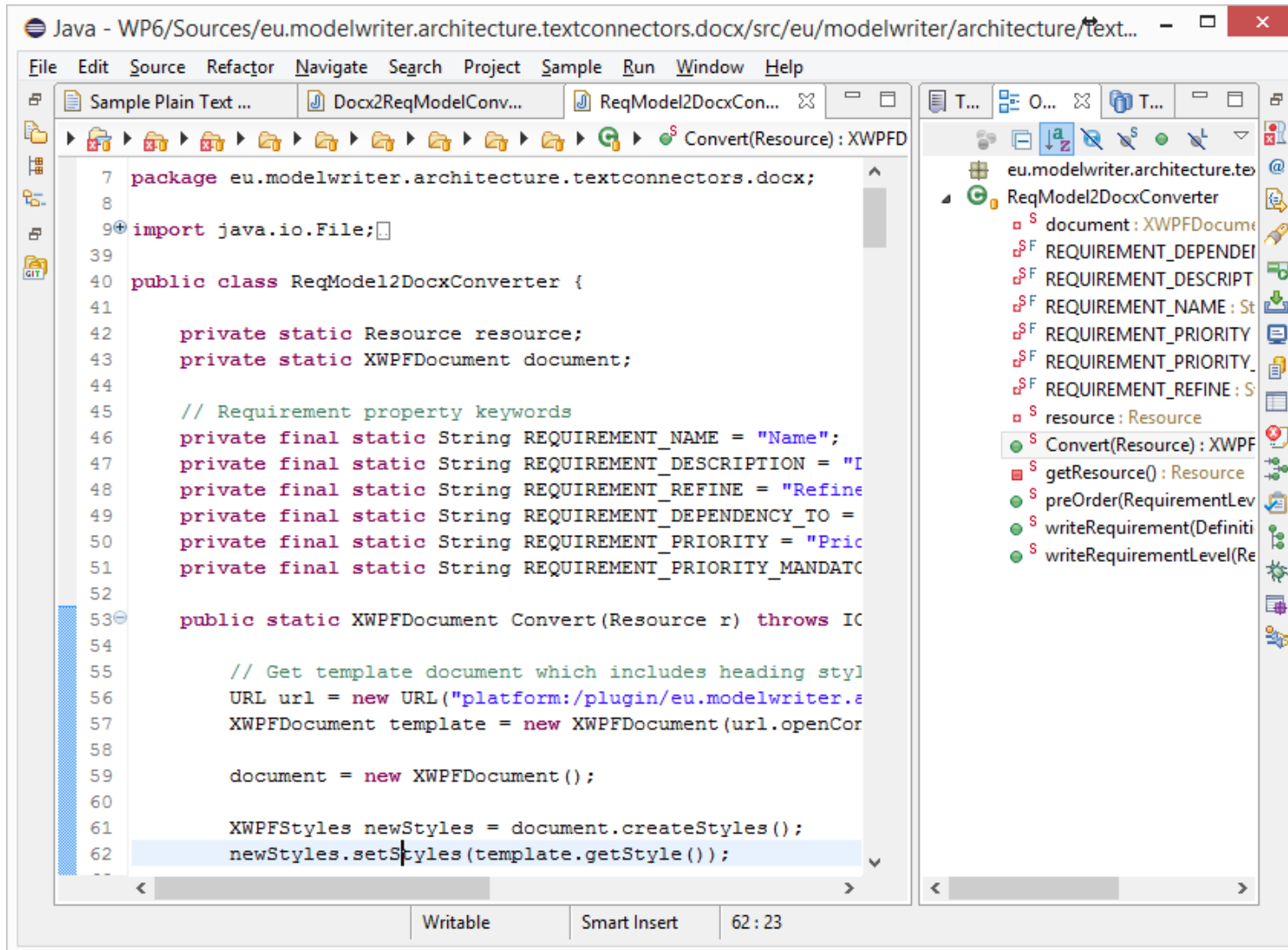
The screenshot shows the Eclipse IDE with a window titled "Modeling - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/model/ReqModel.emf - Edmpse". The menu bar includes File, Edit, Navigate, Search, Project, Sample, Run, Window, and Help. The toolbar shows icons for opening, saving, and running. The editor displays the following GMF diagram model code:

```
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 = "";
```

The status bar at the bottom shows "Writable", "Insert", and "11:9".

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

Even Programming Languages (ASTs)



The screenshot shows an IDE window titled "Java - WP6/Sources/eu.modelwriter.architecture.textconnectors.docx/src/eu/modelwriter/architecture/text...". The main editor displays the following Java code:

```
7 package eu.modelwriter.architecture.textconnectors.docx;
8
9 import java.io.File;
10
39
40 public class ReqModel2DocxConverter {
41
42     private static Resource resource;
43     private static XWPFDocument document;
44
45     // Requirement property keywords
46     private final static String REQUIREMENT_NAME = "Name";
47     private final static String REQUIREMENT_DESCRIPTION = "Description";
48     private final static String REQUIREMENT_REFINE = "Refine";
49     private final static String REQUIREMENT_DEPENDENCY_TO = "Dependency To";
50     private final static String REQUIREMENT_PRIORITY = "Priority";
51     private final static String REQUIREMENT_PRIORITY_MANDATORY = "Mandatory";
52
53     public static XWPFDocument Convert(Resource r) throws IOException {
54
55         // Get template document which includes heading styles
56         URL url = new URL("platform:/plugin/eu.modelwriter.architecture.textconnectors.docx/reqmodel2docxconverter.xwpf");
57         XWPFDocument template = new XWPFDocument(url.openConnection().getInputStream());
58
59         document = new XWPFDocument();
60
61         XWPFStyles newStyles = document.createStyles();
62         newStyles.setStyles(template.getStyle());
63     }
64 }
```

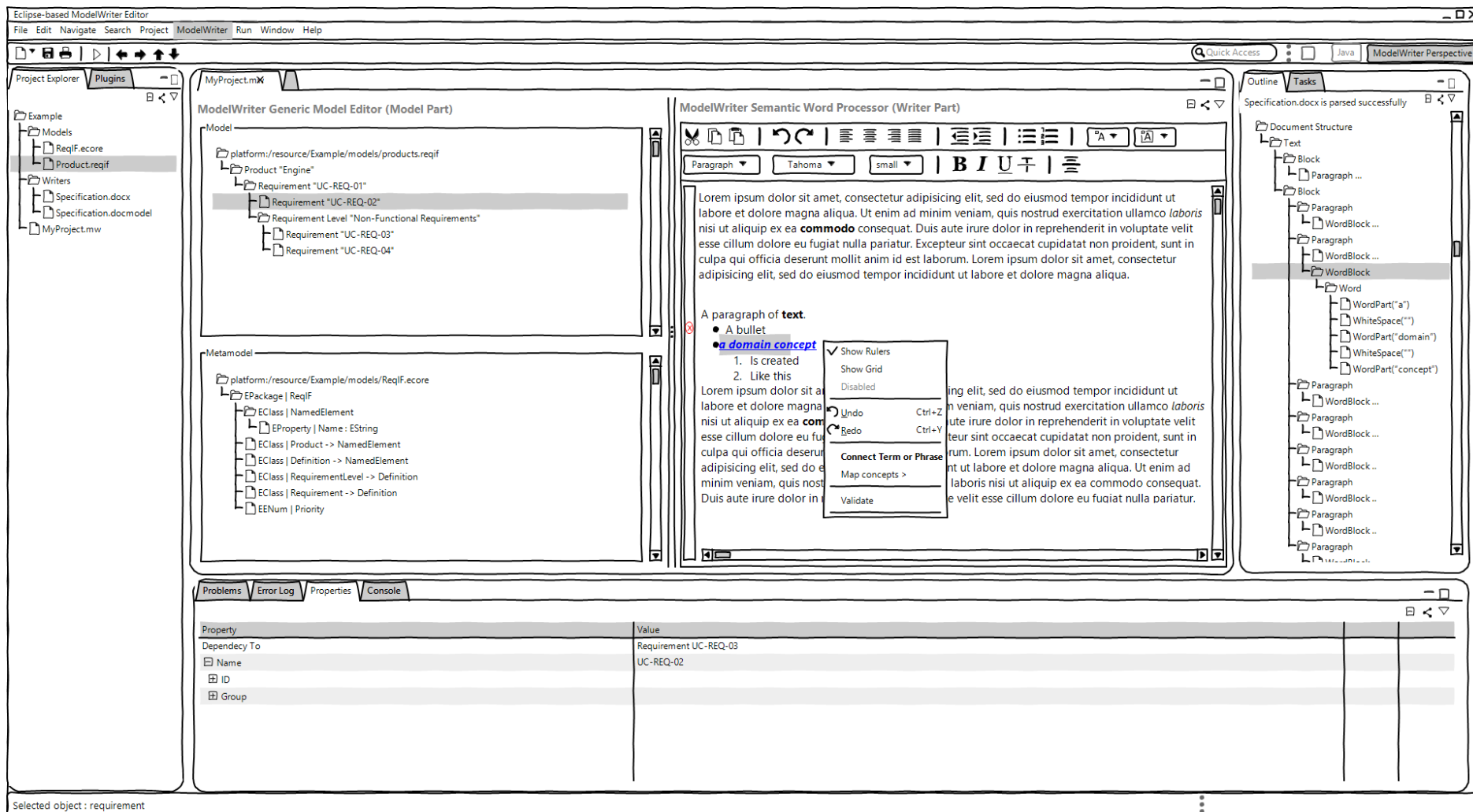
The right-hand pane shows the project structure for "eu.modelwriter.architecture.textconnectors.docx". It includes a package "eu.modelwriter.architecture.textconnectors.docx" and a class "ReqModel2DocxConverter". The class has several attributes and methods listed:

- 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): void
- writeRequirement(Definition): void
- writeRequirementLevel(RequirementLevel): void

The status bar at the bottom indicates "Writable", "Smart Insert", and "62 : 23".

**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

The screenshot displays the Eclipse IDE interface for a project named "Plug-in Development - eu.modelwriter.demonstration.requirements/". The main editor shows a file named "Customer Requirements Specification.md" with the following content:

```
1 # Customer Requirements Specification
2
3 ## UC-1 Create a new SpecObject
4
5 Note that the Specification Editor is the main interface for users. Therefore, creating
6 SpecObjects in this editor is the main success scenario.
7
8 ### Precondition
```

Below the editor, a diagram view shows a hierarchical structure of elements: "Specification" (yellow), "ContractRequirement1" (yellow), "ContractRequirement0" (grey), "SystemRequirement2" (yellow), "Code" (yellow), "SystemRequirement0" (yellow), "Model" (yellow), "SystemRequirement1" (yellow), and "SystemRequirement0" (yellow). The diagram is annotated with relationships like "contract", "system", "fulfills", "satisfiesBy", "requires", and "refines".

Three red circles highlight specific features:

- 1** Points to the "Mapping Action" dialog box, which shows "Relations" for the selected marker. The relations are: "depends: Artifact -> set of Artifact", "conflicts: Artifact -> set of Artifact", "satisfiesBy: SystemRequirement -> set of Implementation", "requires: SystemRequirement -> set of SystemRequirement", and "refines: SystemRequirement -> set of SystemRequirement".
- 2** Points to the "Customer Requirements Specification" text editor, which shows the "UC-1 Create a new SpecObject" section.
- 3** Points to the "Management" menu, which includes options like "Analysis", "Refresh", "Zoom In", "Zoom Out", "Zoom to Fit", "Export to PNG or PDF", "Check Consistency", and "Reason on relations".

Text & Model-Synchronized Document Engineering Platform

**Is it possible to extract
knowledge from texts
fragments based on a given
ontology (model) ?**

Solution – Knowledge Extraction

The screenshot displays the ModelWriter Project application window, which is designed for knowledge extraction and model synchronization. The interface is divided into several key sections:

- File Menu:** Located at the top left, it includes options for **File**, **Link**, **Change**, and **Statistic**. The **Link** menu is currently open, showing sub-options: **Generate Links**, **Search Link**, **Add Link**, and **Remove Link**.
- The Model:** This section on the right displays the RDF model in a **Plain** view. It shows a series of namespace declarations for various standards, including `xmlns:acs="http://airbus-group/aircraft-system#"`, `xmlns:evt="http://airbus-group.installsys/event#"`, `xmlns:rdp="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#"`, and `xmlns:mat="http://airbus-group/material#"`.
- The links between text and model:** This section at the bottom provides a detailed view of the synchronization links. It includes tabs for **T2M**, **M2T**, and **Link**. The **Link** tab is active, showing a list of RDF descriptions and their corresponding model identifiers. For example, it lists `<rdf:Description rdf:about="http://ModelWriter/TxtDocument/id270">` with associated properties like `<j.0:hasOffset>270</j.0:hasOffset>` and `<j.0:isSameAs>http://www.linkedmodel.org/schema/vaem#id</j.0:isSameAs>`. Other entries include `<rdf:Description rdf:about="http://ModelWriter/TxtDocument/attach818">` and `<rdf:Description rdf:about="http://ModelWriter/TxtDocument/attached709">`.

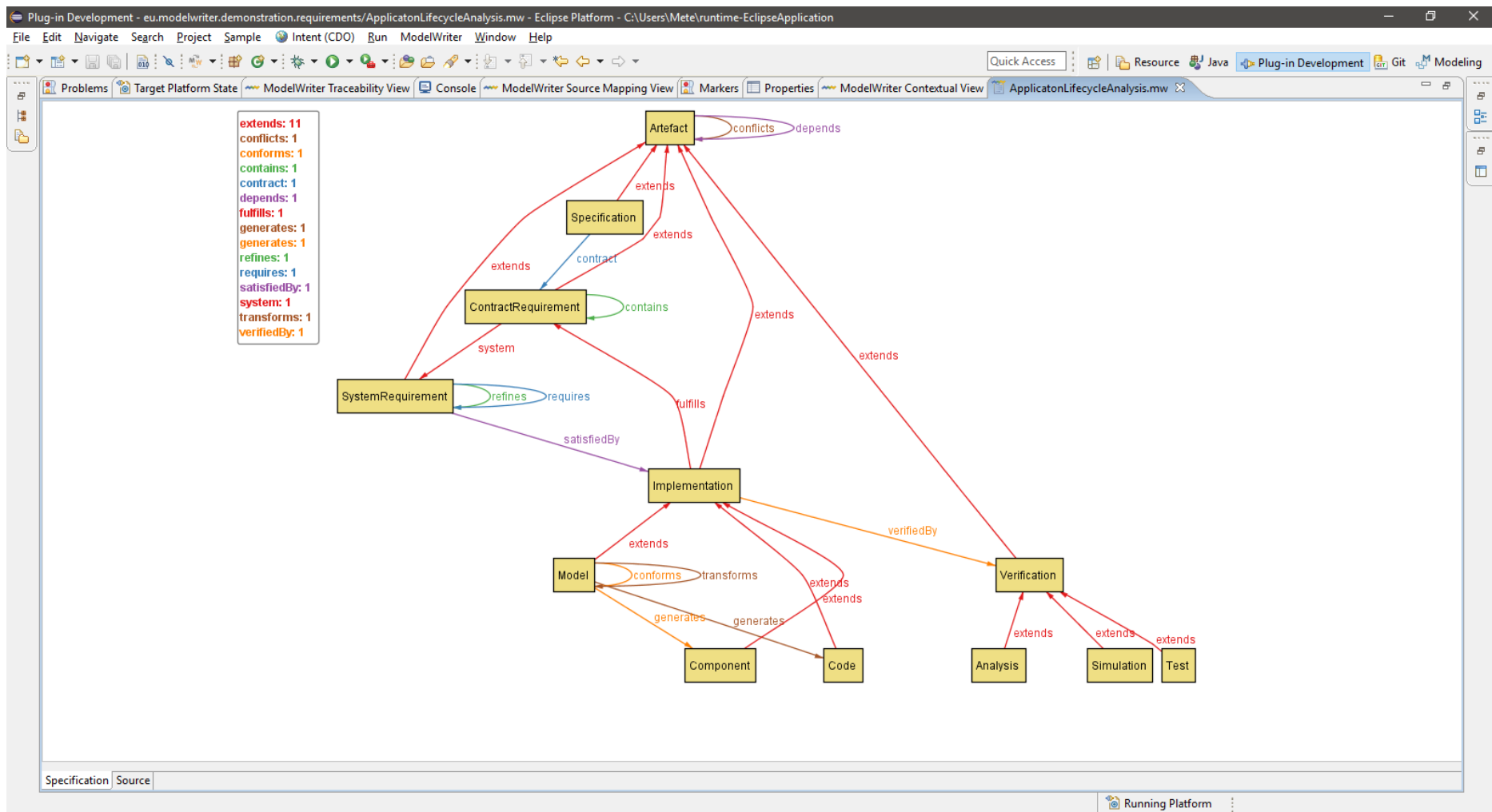
Text & Model-Synchronized Document Engineering Platform



Synchronization is maintained!

What about configuration/formalization of the platform?

Configuration: Havelsan example



A Formal Specification Model to configure the ModelWriter

Is it possible to visualize the trace links?

Traceability: Havelsan example

Plug-in Development - eu.modelwriter.demonstration.requirements/Custom Requirements Specification.md - Eclipse Platform - C:\Users\Mete\workspace\runtime-EclipseApplication-havelsan

File Edit Navigate Search Project Sample Intent (CDO) Run Tarski Window Help

Quick Access Resource Java Plug-in Development Git Modeling

ApplicationLifecycleAnalysis.mw

```
1 module eu.modelwriter/actions/havelsan/alm
2
3 abstract sig Artefact {
4   depends: set Artefact,
5   conflicts: set Artefact
6 -- Reason@conflicts
7 fact {~conflicts in conflicts}
8
9 one sig Specification extends Artefact {
10  contract: some ContractRequirement
11 -- Locate@Text
12 -- Discover@ContractRequirement expect 3
13 sig ContractRequirement extends Artefact {
14  system: set SystemRequirement,
15  contains: set ContractRequirement
16 }
```

Customer Requirements Specification

10 ## Customer Requirements Specification

11

12 ## UC-1 Create a new SpecObject

13 Note that the Specification Editor is the main interface for users. Therefore, creating SpecObjects in this editor is the main success scenario.

14

15 ### Precondition

16 Req10 model exists and is open.

17

18 ### Main Success Scenario

19

20 1. We assume that a Specification exists and is open (not required for alternative scenario)

21

22 2. Open a row's context menu (or in the empty editor space)

23

24 3. Select the Child or Sibling submenu.

Source Specification

Problems Console Markers Properties Tarski Master View Tarski Contextual View Tarski Traceability View

contains: 1
contract: 2
fulfills: 2
refines: 2
requires: 2
satisfiedBy: 2
system: 3

Specification

ContractRequirement1

ContractRequirement0

SystemRequirement0

ContractRequirement2

Code

SystemRequirement2

SystemRequirement1

Management

Analysis

Refresh

Zoom In

Zoom Out

Zoom to Fit

Export to PNG or PDF

Check Consistency

Reason on Relations

Discover Atoms

Clear All Reasoned Tuples

Running Platform

A Formal Specification Model to configure the ModelWriter

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