Next Generation Embedded Systems Requirements Engineering

Silvia Mazzini
(silvia.mazzini@intecs.it)

The NextGenRE Project

- Next Generation Requirements Engineering (NextGenRE) is a study funded by ESA/ESTEC in the period October 2010 - April 2012
  - Intecs Prime Contractor (I)
  - Thales Alenia Space (F)
  - ObjectSecurity (UK)

Main Objective

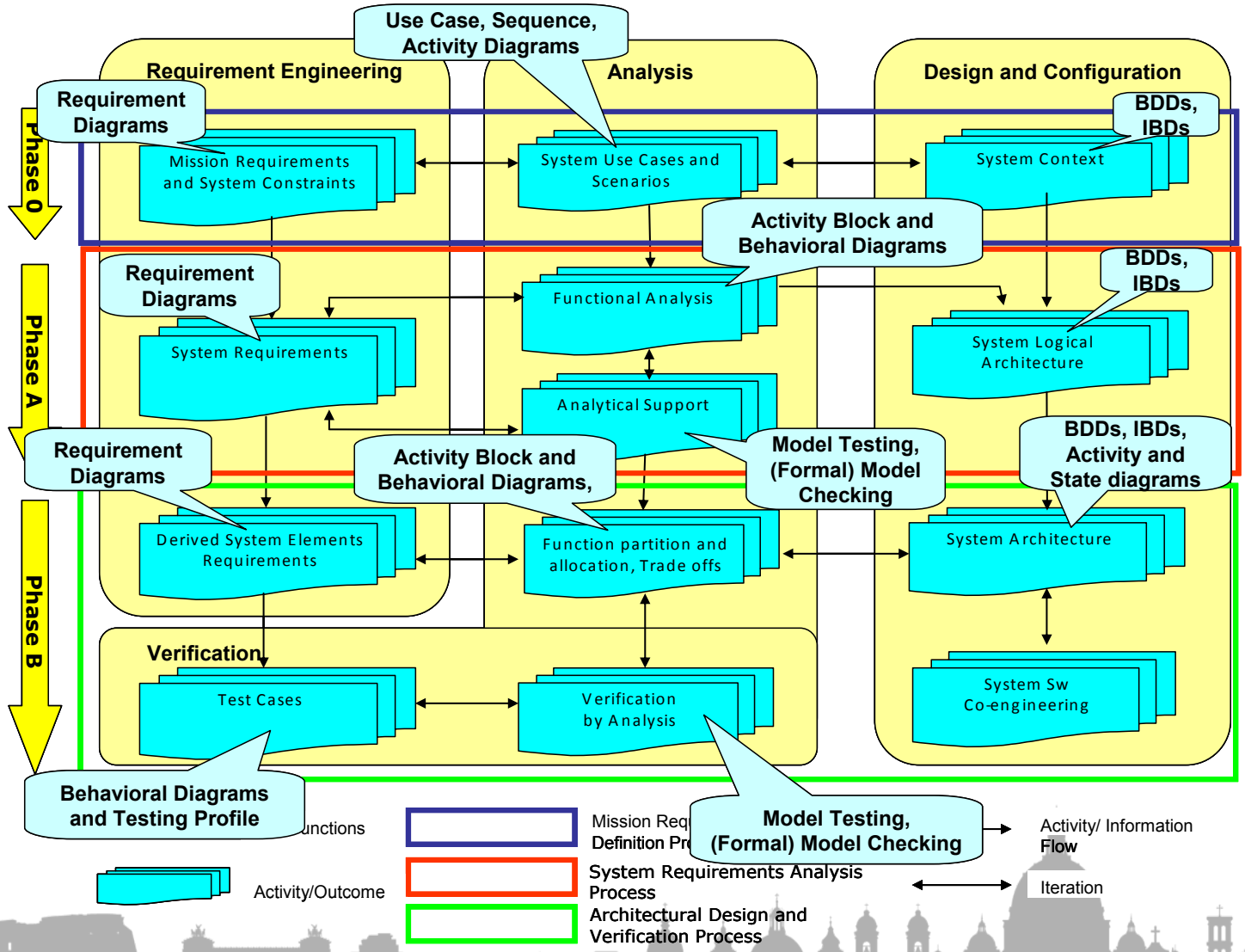
- Improve the requirements engineering process within the European Space industry in connection with Model-based System Engineering (MBSE)

The MBSE Context

- Previous work was funded by ESA/ESTEC to investigate integrated modeling technologies to support Space System and Software Development processes, based on
  - European Cooperation for Space Standardization (ECSS) Standards
  - ISO/IEC 15288
  - The INCOSE Systems Engineering Handbook
  - The SysML OMG Standard

Model-based Space System Engineering (MBSSE)

The MBSSE Methodology



The Core Problem

- Establishing and managing a “good” set of requirements is one of the critical success factors for any complex system project
- Previous work revealed some weaknesses in Model Based Systems Engineering as it is currently practiced
- The essential problem is the “requirement string”: pure text, no semantics
- First steps toward improving on the “requirement string” are:
  - Adding semantics
  - But keeping it acceptable for users
  - Keep it possible to interact with existing methods and tools (e.g. Model Based System Engineering, DOORS)

Model-Based Requirements Engineering

- In **document-centric** requirements engineering, it was acceptable that structure and semantics were only in the minds of author and reader
- But in **model-based requirements engineering**, we can add structure and semantics to the contents of the requirements themselves

Using Wiki Technology

- The central component of the NextGenRE tool is a **Wiki**
- Wikis are a natural technology for requirements engineering support
  - Full support for collaborative work
  - Built-in powerful text processing and formatting
  - Paradigm of pages and links extends naturally to requirements engineering: one page / one requirement
- Wikis have been investigated in recent years for requirements engineering
  - But information is still pure text, “understandable” only by humans

Semantic Wiki Technology

- **Semantic wikis** now make it possible to define machine-processable content and to query information in the generation of pages
- Definition of **page properties**
  - E.g. "**`Rome`**" page has property **`population`** with value **`2 700 000`**
- **Relationships** between pages
  - E.g. "**`Rome`**" page has semantic link **`is_in`** with value **`Italy`**.

Adopted WIKI Technology

- We have constructed our own powerful semantic wiki by extending XWIKI
  - Scalable and stable wiki
  - Macros are available to implement semantic support
  - Classes and objects values are available for pages
  - Tags allows to add keywords (free text) to pages and retrieve all pages to which the keyword is associated

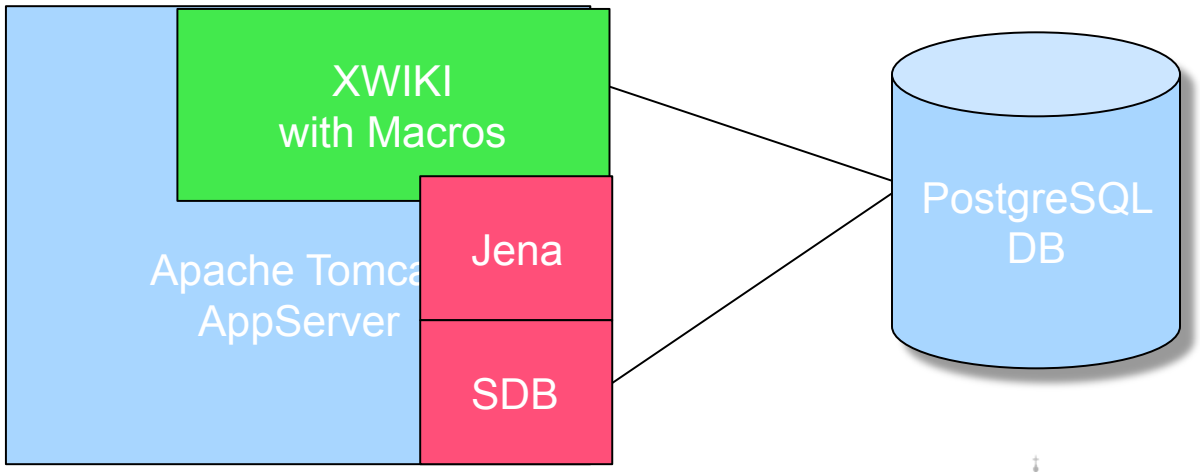
    An Eclipse user interface for XWIKI pages is supported through XEclipse
    - Supports all editing capabilities of XWIKI
  - XEclipse can be the main user interface but the XWIKI can also be accessed using a web browser

Adopted Semantic Web Technology

- To introduce semantics, we have adopted the technology of the **Semantic Web**
- Data is stored in Resource Description Framework (RDF) format so that reasoning can be performed
- The RDF database has been connected to the Wiki in order to create the **Semantic Wiki**

Our SemXWiki Architecture

- Apache Tomcat as Java AppServer
- Jena Ontology API as library for semantic web technology
- SDB as interface between Jena and backend database
- PostgreSQL as backend database



SemXWIKI Features

- Practical semantic support facilities
  - Define resources and their semantic properties in pages
    - E.g. define requirements and properties
  - Obtain semantic properties in pages
    - use SPARQL for semantic queries
- Integrated with

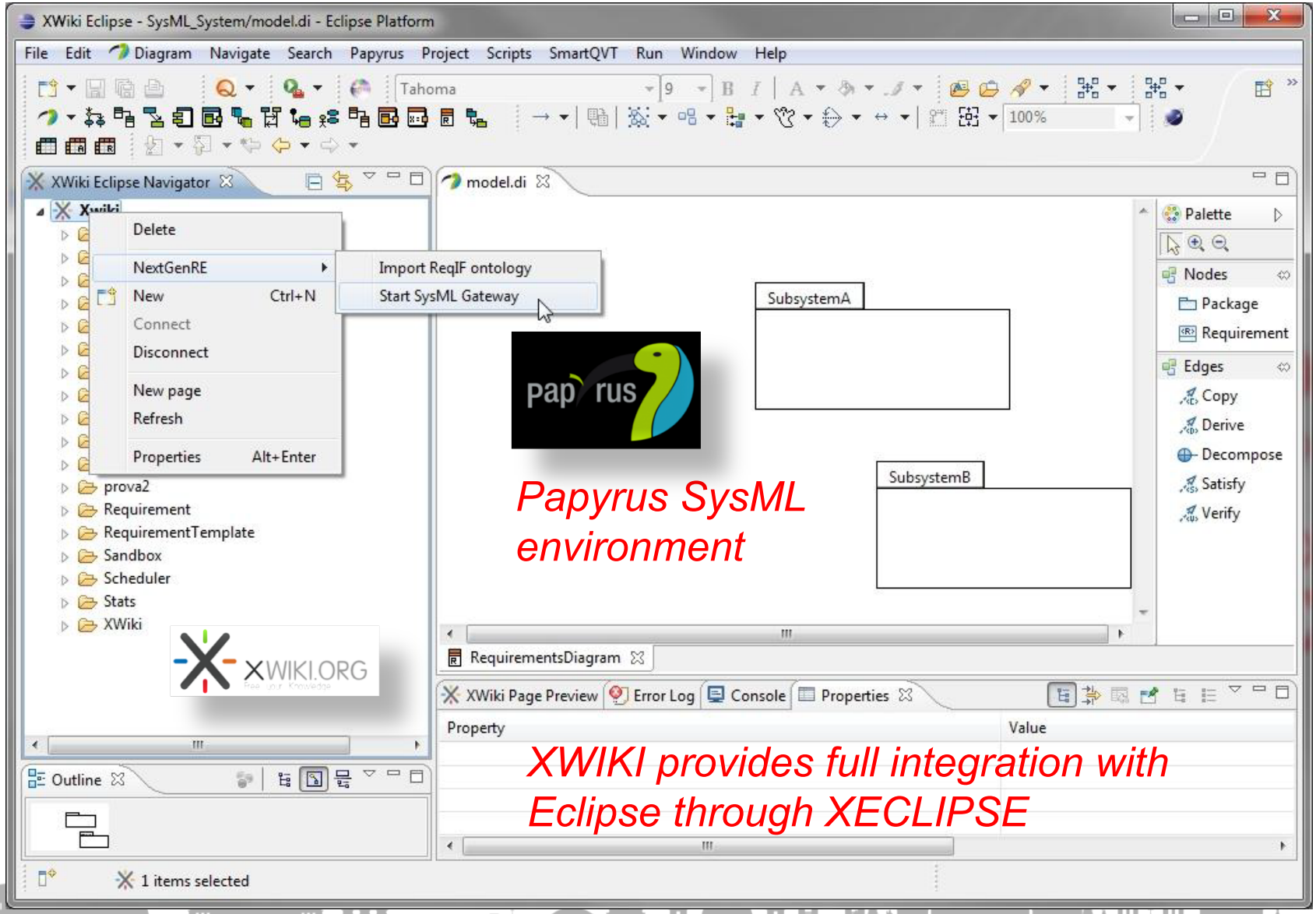
Reasoning Power

- The use of semantic web technology enables the construction of powerful **reasoners**
  - Whenever the semantics can be well-defined
  - Automatic processors can look for contradictions and inconsistencies
  - E.g:
    - a high level requirement is specified for a system architecture,
    - a reasoner could check whether redundant or even conflicting requirements have been derived at lower levels

Connection to Design

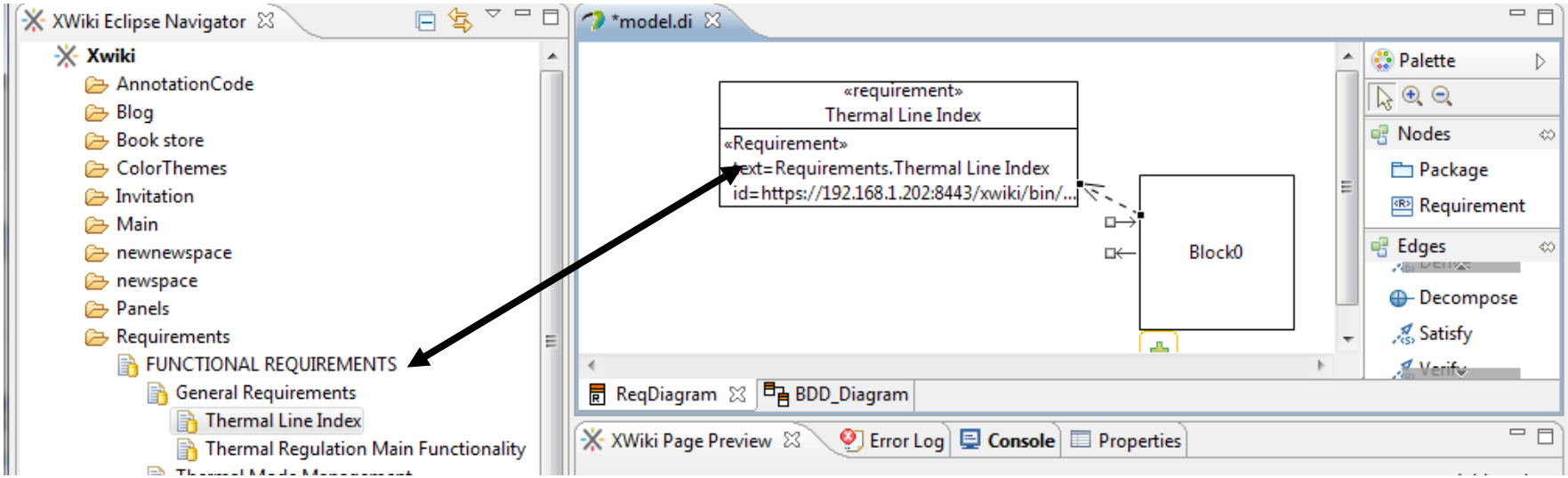
- In support of Model Based System Engineering, we have implemented a **dual environment** for explicit requirements and design engineering
  - connection to SysML artifacts
  - requirements and design evolve together over iterations and refinement

SysML Gateway



Connection to SysML

- Requirements stored in XWiki can be represented in SysML by using drag&drop facility…
- Then links to SysML modeling elements can be created (e.g. satisfy relationship)



Connection to RE tools

- We support connections to existing requirement tools (e.g. DOORS) through Import / Export facilities based on the **Requirements Interchange Format** (ReqIF), now managed by the Object Management Group and under implementation in the Eclipse Requirements Modeling Framework

The NextGenRE Toolset

- A very flexible and configurable prototyping environment including
  - **Semantic web technology** – storage of data in standard RDF format as a basis for reasoning
  - **Extended wiki technology** – stable open source wiki platform with powerful semantic extensions
  - **Template patterns** – customizable templates allowing efficient creation and reuse of requirements
  - **Model-based technology** – industry-standard SysML based modeling, model-to-text, model-to-model transformation capabilities

Demonstration of the NextGenRE Capabilities

- The new approach was validated in a selected representative space system development scenario by Thales Alenia Space system engineers
  - Requirements engineering experimentation on xWiki
    - Scenario from the Mars Sample Return Orbiter (MSRO)
  - System modelling experimentation on Eclipse MDT Papyrus
    - Scenario from Exomars

Requirement templates

Wiki Home » XWiki Space » Data types » Requirement capture

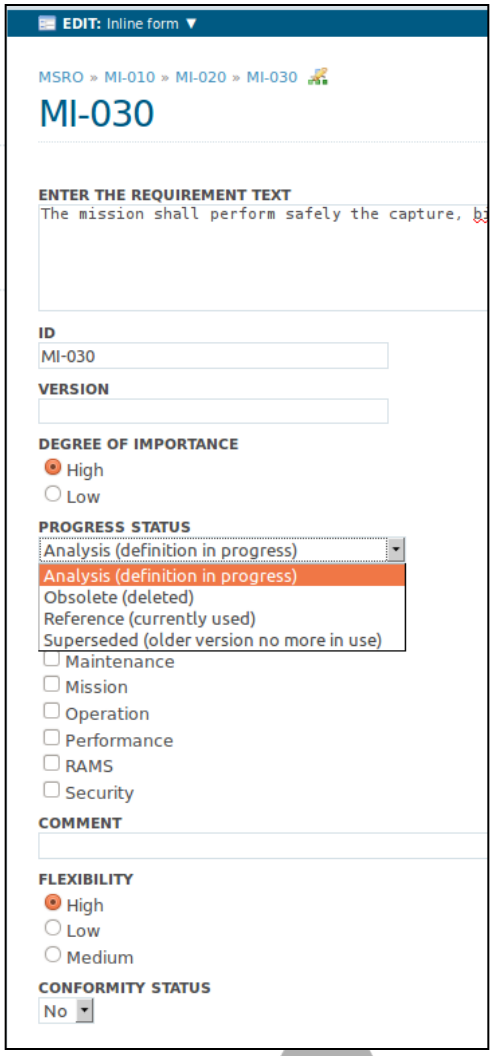
Requirement capture

Last modified by Administrator on 2011/12/19 12:19

Class: Requirement

Class properties:

- Enter the requirement text (Content: TextArea)
- ID (ID: String)
- Version (Version: Number)
- Degree of importance (Priority: Static List)
- Progress Status (ProgressStatus: Static List)
- Category (Category: Static List)
- Comment (Comment: String)
- Flexibility (Flexibility: Static List)
- Conformity Status (ConformityStatus: Boolean)
- parent (parent: String)
- *You can use the class editor to add or modify the class properties.*



Design options management

Specific requirement template has been defined including alternatives for date launch and injection

Class: URD_MSRO_Requirement

Class properties:

- Enter the requirement text (Content: TextArea)
- ID (ID: String)
- Version (Version: Number)
- Degree of importance (Priority: Static List)
- Progress Status (ProgressStatus: Static List)
- Category (Category: Static List)
- Comment (Comment: String)
- Flexibility (Flexibility: Static List)
- ~~Conformity Status (ConformityStatus: Boolean)~~
- Launch Date option (launchdate: Static List)
- Injection option (option: Static List)
- ~~You can use the class editor to add or modify the~~ *class properties.*

MSRO-URD » MI-010a1

MI-010a1

Last modified by Administrator on 2012/02/23 10:01

Enter the requirement text

The system mass should be lower than 4086 kg for a launch

ID
MI-010a1

Version

Degree of importance
High

Progress Status
Analysis (definition in progress)

Category
Mission Performance

Comment

Flexibility
Low

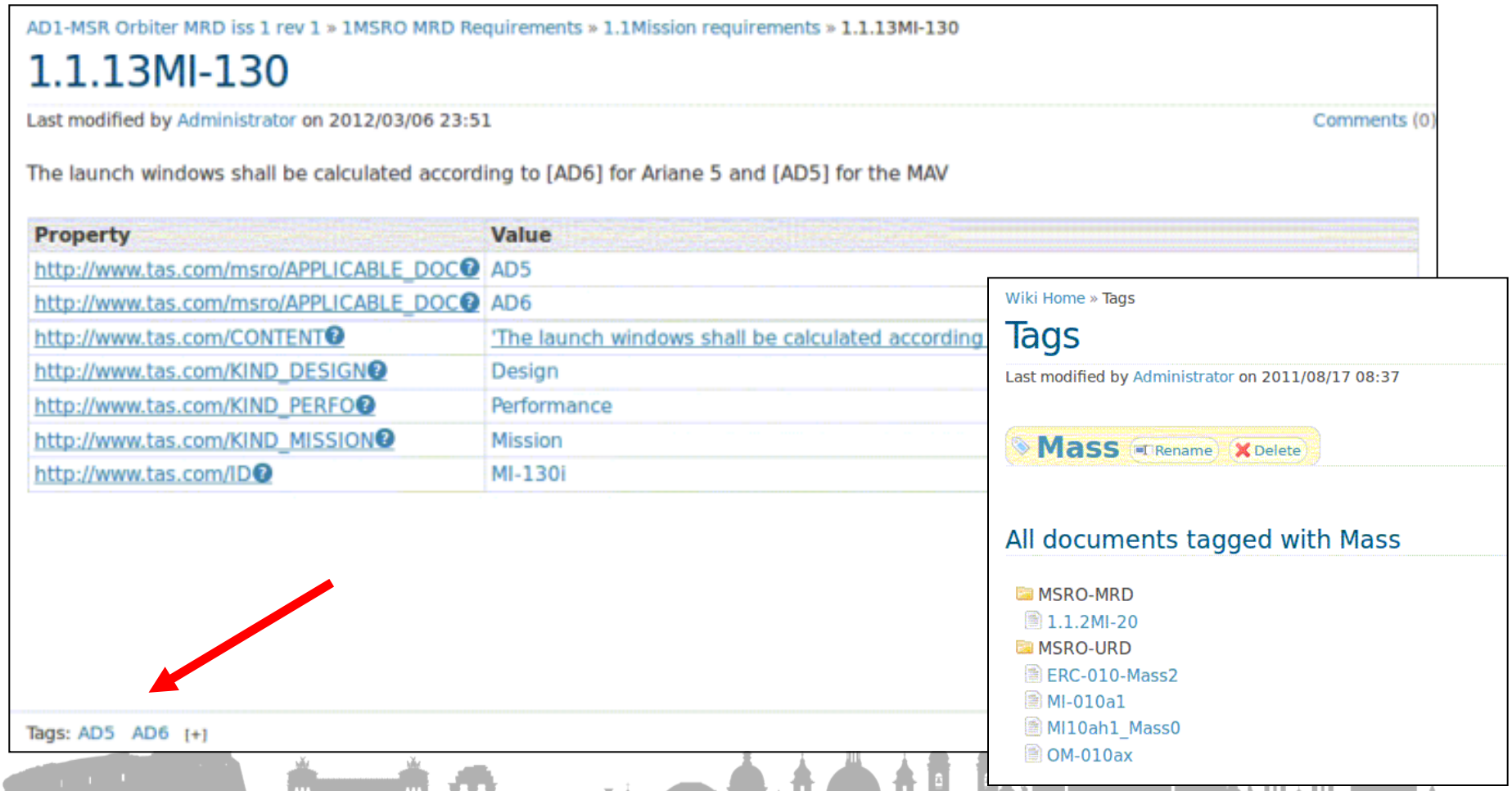
Conformity Status
Yes

Launch Date option
Launch Date in 2022

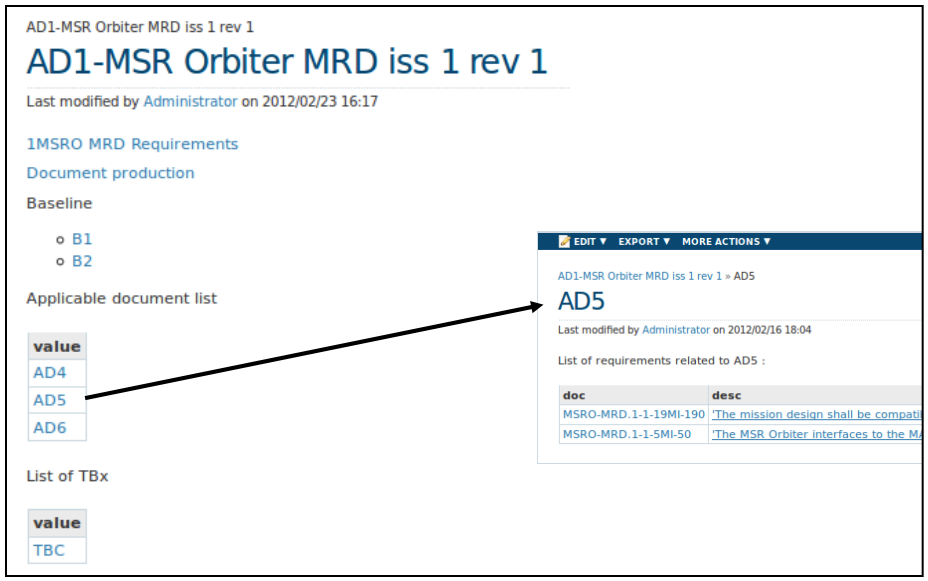
Injection option
Direct HEO

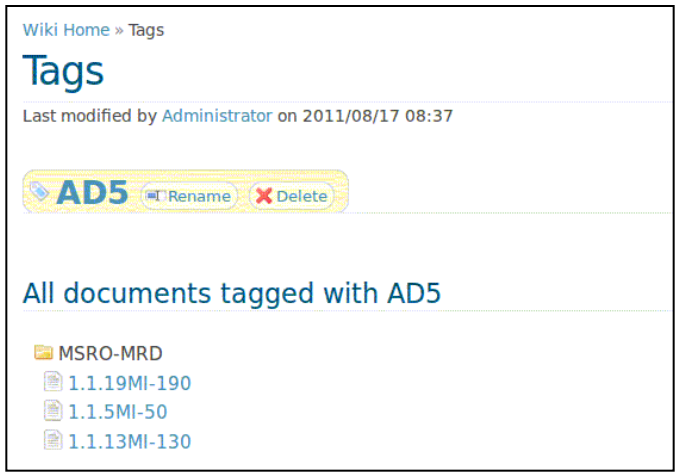
Tag definition

- Tags are free text, which can be referenced automatically by xWiki



Applicable documents management





Addition of a semantic property on each page

Use of semantic query to request all the pages containing a value of this property

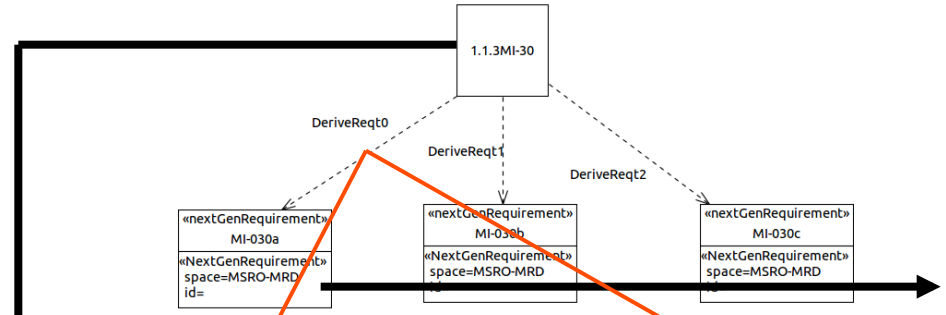
Definition of tag on each page

xWiki manages the page and request creation

Requirement traceability

Based on semantic properties

- Implicit part of the template



AD1-MSR Orbiter MRD iss 1 rev 1 » 1MSRO MRD Requirements » 1.1Mission requirements » 1.1.3MI-30

1.1.3MI-30

Last modified by Administrator on 2012/02/17 11:55

The mission shall perform safely the capture, bio-seal and return to Earth of the MSR Orbiting Sample

Property	Value
http://www.tas.com/DERIVREQ	MSRO-MRD.MI-030a
http://www.tas.com/DERIVREQ	MSRO-MRD.MI-030b
http://www.tas.com/DERIVREQ	MSRO-MRD.MI-030c

MSRO-MRD » MSRO URD example » MI-030a

MI-030a

Last modified by Administrator on 2012/02/17 11:52

Enter the requirement text

The mission shall perform safely the capture of a sample

ID
MI-030a

Version

Degree of importance
High

Progress Status
Analysis (definition in progress)

Category
Functional Mission

Comment

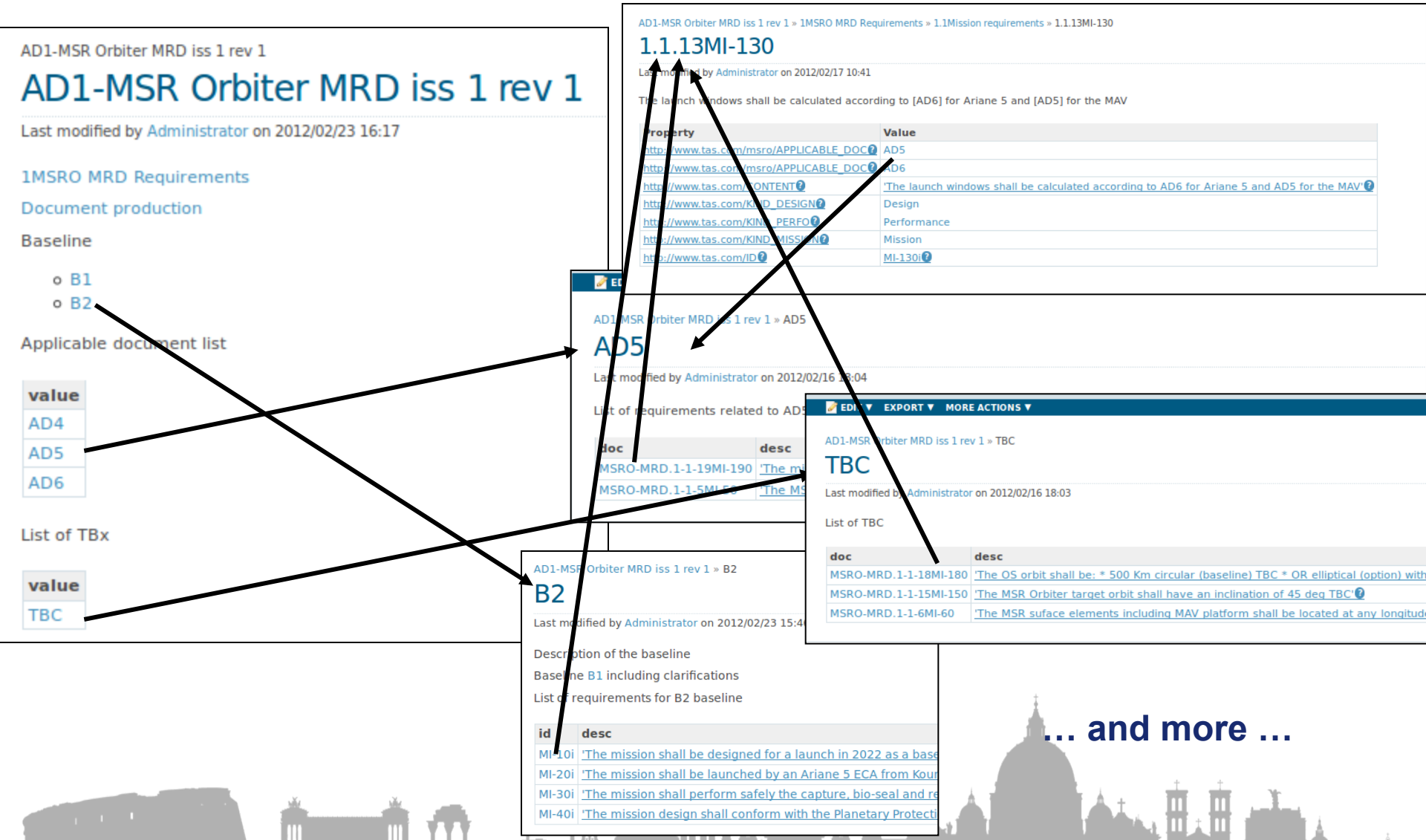
Flexibility
Low

Conformity Status
No

<set property operation aborded!>

Property	Value
http://www.tas.com/REQSPACE	MSRO_URD
http://www.tas.com/REQPARENT	MSRO-MRD.1-1-3MI-30
http://www.tas.com/REQPARENT	MSRO-MRD.1-1-4MI-40
http://www.tas.com/ID	MI-030a
http://www.tas.com/CONTENT	{{html clean="false" wiki="false"}
http://www.tas.com/PRIORITY	High

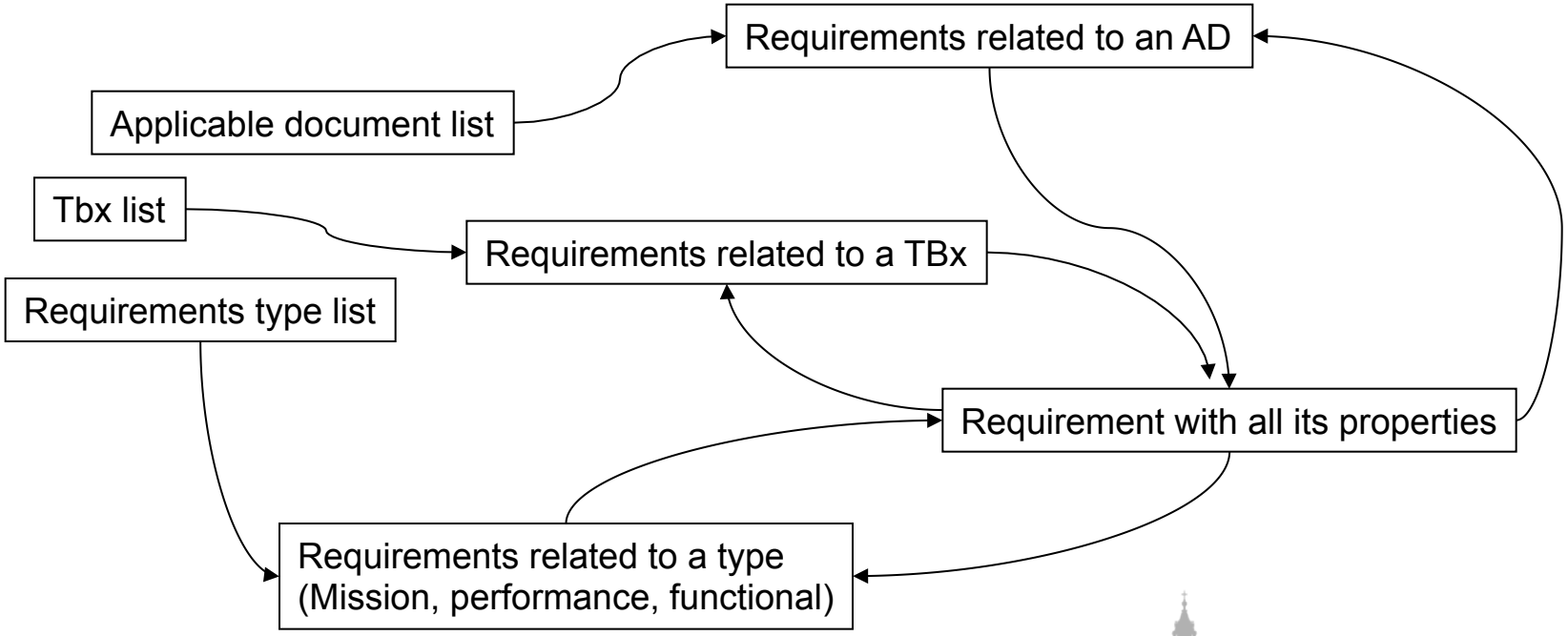
Requirement navigation



… and more …

Semantic properties

- A way to have a fully navigable model …
- … if semantic property values are filled



Conclusion

- The transition of RE to next generation requires new approaches along multiple dimensions.
  - The move from a document-centric to a model-centric RE provides support for the automation of key activities
  - The introduction of dual semantic wiki and SysML modelling environments
- The tool as it stands can be used as a starting point for the implementation of a larger overall vision for requirements engineering.
- It has the potential to lead to
  - lower costs through seamless collaboration,
  - a lower rate of human errors,
  - improved quality
  - less testing through quality-by-design.
- This represents a large step forward for Model Based Systems Engineering

THANK YOU !