ITEA Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands
T + 31 88 003 6136
E info@itea3.org
W www.itea3.org

ITEA 3 is a EUREKA strategic ICT cluster programme

D1.5.2 User Requirements Document (URD)

ModelWriter

Text & Model-Synchronized Document Engineering Platform

Project number: ITEA 2 13028
Work Package: WP1
Task: T1.5 – Consolidated User Requirements and Review

Edited by:

Ferhat Erata <ferhat@unitbilisim.com> (UNIT)
Moharram Challenger <Moharram.challenger@unitbilisim.com> (UNIT)
Serhat Çelik <serhat.celik@unitbilisim.com> (UNIT)
Hasan Emre Kırmızı <emre.kirmizi@unitbilisim.com> (UNIT)
Ümit Anıl Öztürk <anil.ozturk@unitbilisim.com> (UNIT)

Date: 24-Aug-2015
Version: 1.0.0

Document History

Version	Author(s)	Date	Remarks
0.1.0	Ferhat Erata Moharram Challenger	30-Apr-2015	Draft
0.5.0	Serhat Çelik Hasan Emre Kırmızı Ümit Anıl Öztürk	24-Aug-2015	Updating the list of deliverables from GitHub
1.0.0	Moharram Challenger	24-Aug-2015	Release

Table of Contents

1. Introduction

1.1. Role of the deliverable

This document provides the list of higher level user requirements for the ModelWriter project. The main objective of this deliverable is to formalize and prioritize User Requirements (incl. commercial requirements) on ModelWriter platform. This document is the first version of the user requirements collected from industrial partners. It may be up-dated depending on the further details and requirements we get from our industrial use case providers.

The **User Requirements Document** (URD) (D1.5.2) is automatically compiled from the issue list of the public "Requirements" repository of ModelWriter's GitHub organization[1]. The information about confirmed requirements and the prioritization of the requirements are programmatically gathered from the public 'waffle' scrum board of the "Requirements" repository[2].

1.2. The List of Technical Work Packages

UC Code	Requirements derived from
WP2	Semantic Parsing and Generation of Documents and Documents Components
WP3	Model to/from Knowledge Base (synchronization mechanism)
WP4	Knowledge Base Design and Implementation
WP6	Architecture, Integration and Evaluation

1.3. The List of Use Cases

UC Code	Requirements derived from
UC-FR-01	Synchronization between Models and Documentation
UC-FR-02	Enterprise Architecture
UC-FR-03	Synchronization of regulation documentation with a design rule repository

[1] https://github.com/ModelWriter/Requirements/issues
[2] https://waffle.io/modelwriter/requirements

UC Code	Requirements derived from
UC-FR-04	Production of a context specific design document
UC-TR-01	Production of a proposal in response to an IPA Invitation To Tender
UC-TR-02	Collaborative production of a proposal for an IPA project
UC-TR-03	Synchronization of ReqIF/Clafer models with requirement specifications
UC-TR-04	Requirement Engineering for System Modelling
UC-TR-05	Synchronous Business Process Design with Use Cases
UC-BE-01	Requirements IT
UC-BE-02	Automated Test Generation

1.4. Conventions

The requirements are prefixed by “REQ-UR-xxx”, and are written in a roman typeface, where “REQ” stands for “Requirement”, “UR” indicates “User Requirements” and “xxx” is the positive integer identifier of the requirement. You can add this id (xxx) which is unique entire ‘requirements’ repository, at the end of https://github.com/ModelWriter/Requirements/issues/ to access the latest version of the requirement.

1.5. Structure of the document

This document is organized as follows:

- Chapter 1 introduces the document.
- Chapter 2 describes each requirements.

1.6. Terms, abbreviations and definitions

Abbreviation	Definition
RDF	Resource Description Framework
WP	Work Package
UC	Use Case

Abbreviation	Definition
BPMN	Business Process Monde & Notation
ReqIF	Requirements Interchange Format
RMF	Requirements Modeling Framework
SBVR	Semantics Of Business Vocabulary And Rules

2. User Requirements

2.1. REQ-UR-53 [Mandatory]

ModelWriter as a Next Generation Requirements Engineering Tool: ModelWriter should be equipped with Requirements Engineering features.

Use Cases: UC-BE-01 Requirements IT, UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer
Description:
The text below is an excerpt from the FPP (Section 2.2.2.4):

A requirements engineering tool such as the dominant Rationale DOORS is basically comparable to a database management system. ModelWriter equipped with the envisioned Requirements Engineering features will be far superior to such “advanced” tool by providing the following innovative features (list far from being exhaustive!):

The most natural (i.e. word processor-like rather than database-like) interface for users, resulting in quantum leap of efficiency in requirements capture. Indeed, users naturally express their requirements in a written way (incl. graphics and nice layout), not in a database-oriented manner. Moreover, requirements documents generated from the database by today’s tools are easily recognizable by the questionable readability, which is a problem that is simply avoided by the ModelWriter’s approach.
Intelligent features for checking the readability (lack of ambiguity), completeness and consistency of the proposed requirements, e.g. by using ORM verbalization mechanism and model checkers (to be developed by KUL2, see UC-BE-01): computer-assisted Quality Reviews.
Intelligent features for further manipulating the requirements, e.g. for creating or verifying traceability matrices between 2 sets of requirements (=computer-assisted traceability matrices), with e.g. applications for verifying compliance to standards.

URI: https://github.com/ModelWriter/Requirements/issues/53
Created: 17.05.2015
Created By: ferhaterata
Assignee: ferhaterata

2.2. REQ-UR-52 [Mandatory]

Text-Based Knowledge Extraction with ModelWriter (Semantic Word Processor)

Use Cases:
Description:
The text below is an excerpt from the FPP (Section 2.1.1-3):

Nowadays, text that has been captured by a Word Processor is “frozen”, i.e. hardly exploitable as actual “piece of knowledge”. Computer assistance is limited to spelling & grammar checkers, or keyword-based search engines, which all work indeed at the level of “words”, and really not at the level of “knowledge” (which would reveal concepts and relationships between concepts).

Moreover, words are often ambiguous, whereas “knowledge” is expected to be free from this flaw. There is nowadays no tooling offered to technical authors to (semi-)automatically transform those “words” into exploitable & ambiguity-free “pieces of knowledge”.

This is why ModelWriter will not include just a plain “Word Processor”: it will rather include a “Semantic Word Processor”. This means that ModelWriter will try to understand the various textual parts of a document expressed in Natural Language (e.g. formal Requirements on a product expressed in English). It will (among other NLP techniques) exploit any presentation & formatting hints as well as any industry standard document structures and conventions (incl. standard glossary of terms) to semi-automatically and accurately transform pieces of text into a further processable knowledge model counter-part. Any ambiguities detected will be revealed to the human actor, for resolution (“do you mean this or that?”).

Note that this Knowledge Extraction feature (which creates an accurate internal knowledge model out of text) is distinct from the Knowledge Capture’s text-model synchronization mechanism (point 1.3). The later indeed maintains a link between a user-provided piece of text and a user-provided model.

The text below is an excerpt from the FPP (Section 3.3.6)

T6.3 Writer Part enhancements:

This task is about augmenting an existing word-processor (chosen during the prototyping phase - see T6.1) in order to create a true “Semantic word processor” fulfilling the ModelWriter requirements (as established in WP1 and the architecture design document T6.2).

This will be done by:

Wrapping the existing word processor as an Eclipse Plugin, compatible with the modelling standards of the Eclipse IDE (Eclipse Modeling Framework) and able to detect notifications sent by the Bi-directional Synchronization Mechanism.
Plugging the additional behaviour brought by each ModelWriter component (Knowledge Base, Semantic parsing...) inside this traditional word-processor. For example, when a document gets saved, the augmented word-processor should perform a semantic parsing of all requirements expressed in the document, and update the Knowledge Base with the resulting models.

The resulting Writer part (D6.3.1-x) will be integrated in the ModelWriter product (built thanks to T6.7). The University of Bremen will bring to this task his experience gained during previous work on word-processor augmentation.

URI: https://github.com/ModelWriter/Requirements/issues/52
Created: 17.05.2015
Created By: ferhaterata
Assignee: ferhaterata

2.3. REQ-UR-51 [Mandatory]

“MW” Knowledge Dissemination Standard

Use Cases:
Description:
The text below is an excerpt from the FPP (Section 2.1.1-5):

Nowadays, people exchange mostly documents, in which the knowledge is somehow implicitly buried & hidden, hence the need for “Knowledge Extraction” tools (a feature of ModelWriter). However, once extracted and further cleaned-up (i.e. after disambiguation and other consistency checks), it would be counter-productive to serialize the knowledge back into a hardly reusable piece of text or document.

There is therefore the objective to promote a standard “ModelWriter” exchange format (.mw) for saving both text and knowledge models linked together, and to demonstrate this on a document that promotes a standard.

URI: https://github.com/ModelWriter/Requirements/issues/51
Created: 17.05.2015
Created By: ferhaterata
Assignee: ferhaterata

2.4. REQ-UR-50 [Desirable]

ModelWriter shall support Rich-Blended Modeling Environments.

Use Cases: UC-FR-04 Production of a context specific design document, UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer
Description:
Several industrial use cases indicate that a technical document may contain (semi-)structured data partially in text and partially in a visual model.

URI: https://github.com/ModelWriter/Requirements/issues/50
Created: 16.05.2015
Created By: ferhaterata
Assignee: N/A

2.5. REQ-UR-49 [Mandatory]

The system shall provide a unified Graphical User Interface (for both Model and Writer parts).

Use Cases: UC-TR-03 Generation and management of feature models, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
URI: https://github.com/ModelWriter/Requirements/issues/49
Created: 30.04.2015
Created By: ferhaterata
Assignee: N/A

2.6. REQ-UR-48 [Mandatory]

The system should be able to perform semantic parsing.

Use Cases: UC-FR-01 Sync. between Models and Documentation, UC-FR-02 Enterprise Architecture, UC-FR-03 Synchronization of regulation documentation..., UC-FR-04 Production of a context specific design document, UC-TR-01 Production of a proposal in response to an IPA..., UC-TR-02 Collab. production of a proposal for an IPA project, UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
URI: https://github.com/ModelWriter/Requirements/issues/48
Created: 30.04.2015
Created By: ferhaterata
Assignee: N/A

2.7. REQ-UR-47 [Mandatory]

The user does not want to be bothered with information such as mapping links.

Use Cases: UC-FR-01 Sync. between Models and Documentation
Description:
URI: https://github.com/ModelWriter/Requirements/issues/47
Created: 30.04.2015
Created By: mrostren
Assignee: N/A

2.8. REQ-UR-42

ModelWriter should support at least one Document Markup Language and one Lightweight Markup Language

Use Cases: UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases
Description:
This requirement implies #43.

This requirement implies #44.
URI: https://github.com/ModelWriter/Requirements/issues/42
Created: 07.04.2015
Created By: ferhaterata
Assignee: ferhaterata

2.9. REQ-UR-41 [Mandatory]

The system shall help to synchronize the SIDP natural language document with the modeled rules without forced modification.

Use Cases: UC-FR-03 Synchronization of regulation documentation

Description:
SIDP documents constitute the Airbus corpora.
SIDP means System Installation Design Priniple.
A set of rules has been derived from the original texts and modeled into a database.

URI: https://github.com/ModelWriter/Requirements/issues/41
Created: 18.03.2015
Created By: annemonceaux
Assignee: N/A

2.10. REQ-UR-40 [Mandatory]

The system shall show on demand (coloured mark or other mean) the text elements that are linked to the "visual model" concepts, and conversely

Use Cases: UC-FR-03 Synchronization of regulation documentation..., UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
URI: https://github.com/ModelWriter/Requirements/issues/40
Created: 18.03.2015
Created By: annemonceaux
Assignee: N/A

2.11. REQ-UR-39 [Mandatory]

The system shall allow the user to configure the document generation content

Use Cases: UC-FR-04 Production of a context specific design document, UC-TR-01 Production of a proposal in response to an IPA..., UC-TR-02 Collab. production of a proposal for an IPA project, UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
In the context of Airbus case, when generating a new document, we need to specify some criteria such as the Targeted system, Program, etc to which the rules applies

URI: https://github.com/ModelWriter/Requirements/issues/39
Created: 31.01.2015
Created By: annemonceaux
Assignee: N/A

2.12. REQ-UR-38 [Mandatory]

The system shall provide a user friendly way to manage any additional concepts needed.

Use Cases: UC-FR-04 Production of a context specific design document, UC-TR-01 Production of a proposal in response to an IPA..., UC-TR-02 Collab. production of a proposal for an IPA project, UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
For example in the context of Airbus, the users are likely to add new content to the rule repository
URI: https://github.com/ModelWriter/Requirements/issues/38
Created: 31.01.2015
Created By: annemonceaux
Assignee: N/A

2.13. REQ-UR-37 [Out of Scope]

A specification for an improve and controlled formulation of the rules in semi-structured natural language should be proposed

Use Cases: UC-FR-03 Synchronization of regulation documentation
Description:
URI: https://github.com/ModelWriter/Requirements/issues/37
Created: 31.01.2015
Created By: annemonceaux
Assignee: N/A

2.14. REQ-UR-35 [Optional]

The system shall allow semantic retrieving or reasoning using the model elements.

Use Cases: UC-FR-03 Synchronization of regulation documentation..., UC-FR-04 Production of a context specific design document
Description:
URI: https://github.com/ModelWriter/Requirements/issues/35
Created: 31.01.2015
Created By: annemonceaux
Assignee: N/A

2.15. REQ-UR-33 [Mandatory]

The system shall allow the end user to edit text and "visual" model (such as tables, diagrams or 2D drawings) synchronously

Use Cases: UC-FR-03 Synchronization of regulation documentation..., UC-TR-03 Generation and management of feature models, UC-TR-04 Requirements Engineering with SysML Designer, UC-TR-05 Sync. Business Process Design with Use Cases

Description:
URI: https://github.com/ModelWriter/Requirements/issues/33
Created: 31.01.2015
Created By: annemonceaux

Assignee: N/A

2.16. REQ-UR-32

The system shall allow the end user to keep his/her usual working environment.

Use Cases: UC-FR-03 Synchronization of regulation documentation...
Description:
URI: https://github.com/ModelWriter/Requirements/issues/32
Created: 31.01.2015
Created By: annemonceaux
Assignee: N/A

2.17. REQ-UR-31 [Mandatory]

The system shall allow the user to activate/deactivate a synchronization direction

Use Cases: UC-FR-02 Enterprise Architecture
Description:
The SmartEA team and user Won't any modifcation of their models basing on the documentation modification. That's why ModelWriter must allow to deactivate the Text -> Model synchronization so in SmartEA context this direction will not be used.

URI: https://github.com/ModelWriter/Requirements/issues/31
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.18. REQ-UR-30 [Mandatory]

The system shall allow the users to work in a collaborative manner

Use Cases: UC-FR-02 Enterprise Architecture
Description:
To fit this requirement the ModelWriter solution shall be able to be used on remote server. This will allow SmartEA users to work together on the same document/model.

URI: https://github.com/ModelWriter/Requirements/issues/30
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.19. REQ-UR-29 [Mandatory]

The system shall be easy to integrate in an RCP application

Use Cases: UC-FR-02 Enterprise Architecture, UC-TR-04 Requirements Engineering with SysML Designer
Description:

The system shall be integrated to SmartEA to facilitate documentation typing and synchronization. The ModelWriter engine should headless to be called by external tools, like Obeo SmartEA. the system must have a standalone engine.
It will be called to facilitate documentation typing and synchronization between the text content and all models offered by the SmartEA diagrams.

URI: https://github.com/ModelWriter/Requirements/issues/29
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.20. REQ-UR-28 [Mandatory]

The system shall offer a notification system

Use Cases: UC-FR-02 Enterprise Architecture
Description:
ModelWriter must allow notifying users about all the documentation changes of other users.

URI: https://github.com/ModelWriter/Requirements/issues/28
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.21. REQ-UR-25 [Mandatory]

The system should allow to filter synchronization warnings, errors and information.

Use Cases: UC-FR-01 Sync. between Models and Documentation
Description:
ModelWriter should allow to filter synchronization warnings, errors and information.
It also should allow to keep the filter information "ON" or activated during the documentation life cycle.

PS: This requirement is based on the way how proceed the Check-Style tool.

URI: https://github.com/ModelWriter/Requirements/issues/25
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.22. REQ-UR-23 [Mandatory]

The System shall not enforce any dependency on non-open source artifacts such as tools, applications, and libraries.

Use Cases: UC-FR-01 Sync. between Models and Documentation
Description:

The Sirius tool is an open-source tool, using ModelWriter Must Not lead the team to add dependencies to open source applications or tools which are accepted EPL.

URI: https://github.com/ModelWriter/Requirements/issues/23
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.23. REQ-UR-20 [Mandatory]

The System shall propose an eclipse editor to handle documentation/models mappings

Use Cases: UC-FR-01 Sync. between Models and Documentation
Description:
The system must offer an eclipse editor to allow users mapping documents parts to models element editing their mappings and visualizing asynchronous mappings.

URI: https://github.com/ModelWriter/Requirements/issues/20
Created: 20.01.2015
Created By: mrostren
Assignee: N/A

2.24. REQ-UR-17

The system must support an open requirement authoring tool (such as RMF)

Use Cases: UC-TR-03 Generation and management of feature models
Description:
RMF (Eclipse Requirements Modeling Framework) is a framework for working with textual requirements, structured as ReqIF models, the industry standard for exchanging requirements. RMF uses natively ReqIF, the open standard for requirements exchange, allowing you to exchange requirements with many industry applications like Rational DOORS® or PTC integrity®.

https://www.eclipse.org/rmf/

/CC @jastram @ejuliot
URI: https://github.com/ModelWriter/Requirements/issues/17
Created: 27.11.2014
Created By: ferhaterata
Assignee: ferhaterata

2.25. REQ-UR-16

The system should support ReqIF standard as user visible model.

Use Cases: UC-TR-03 Generation and management of feature models
Description:
OMG Requirements Interchange Format (ReqIF) 1.1

Article of Dr. Michael Jastram on Requirement Engineering Magazine is informative.

/CC @jastram @ejuliot
URI: https://github.com/ModelWriter/Requirements/issues/16
Created: 27.11.2014
Created By: ferhaterata
Assignee: ferhaterata

2.26. REQ-UR-15 [Desirable]

BPMN 2.X standard shall be supported as a user visible model

Use Cases: UC-TR-05 Sync. Business Process Design with Use Cases
Description:
The tool of OBEO can be used
http://marketplace.obeonetwork.com/module/bpmn

Eclipse MDT - BPMN 2.x:
https://www.eclipse.org/modeling/mdt/?project=bpmn2
https://www.eclipse.org/bpmn2-modeler/

/CC @ejuliot
URI: https://github.com/ModelWriter/Requirements/issues/15
Created: 20.11.2014
Created By: ferhaterata
Assignee: ferhaterata

References

N/A

Appendixes

N/A