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

D1.4.1 Industrial Use Cases for Turkish Consortium

ModelWriter

Text & Model-Synchronized Document Engineering Platform

Work Package: WP1
Task: T1.4 - Industrial Use Cases for Turkish Consortium

Edited by:

Ferhat Erata <ferhat@unitbilisim.com> (UNIT)
Moharram Challenger <moharram@unitbilisim.com> (UNIT)

Date: 15-SEP-2015
Version: 0.1.1

Document History

Version	Author(s)	Date	Remarks
0.1.0	Ferhat Erata Moharram Challenger	20-Apr-2015	Draft
0.1.1	Ferhat Erata	21-Apr-2015	Footer has been changed
0.1.2	Ferhat Erata	15-Sep-2015	HAVELSAN Use Case is updated.

Table of Contents

1. Introduction

1.1. Role of the deliverable

This document is the first version of the description of the use cases proposed by the Turkish consortium. It may be up-dated depending on the further details and requirements we get from our industrial use case providers.

1.2. Structure of the document

This document is organized as follows:

- Chapter 1 introduces the document.
- Chapter 2 describes for each use case: the scope and motivation, the approach and the available resources (corpora).
- Annex 1 lists for each use case the annex documents and associated data deliverables, so called “corpora”:
  - D1.4.2 Corpus for Turkish Use Cases (Public)
  - D1.4.3 Corpus for Turkish Use Cases (Private)

1.3. Terms, abbreviations and definitions

Abbreviation	Definition
RDF	Resource Description Framework
WP	Work Package
UC	Use Case
BPMN	Business Process Monde & Notation

2. Use Case Description

The use cases are provided by UNIT, Havelsan, Hisbim and KoçSistem companies.

- **Error! Reference source not found.**
- **Error! Reference source not found.**
- **Error! Reference source not found.**
- UC-TR-04 Requirement Engineering for System Modelling
- **Error! Reference source not found.**

Use Case Contribution table is shown below:

Company	Contributions
UNIT	ReqIF/Clafer's (Feature Modeling) documentation (UC-TR-03) Requirement Engineering for System Modelling (UC-TR-04) Synchronous Business Process Design with Use Cases (UC-TR-05)
KOCSISTEM	ReqIF/Clafer's (Feature Modeling) documentation (UC-TR-03) Synchronous Business Process Design with Use Cases (UC-TR-05)
HISBIM	**Error! Reference source not found.** (UC-TR-01) **Error! Reference source not found.** (UC-TR-01)
HAVELSAN	Requirement Engineering for System Modelling (UC-TR-04)

2.1. UC-TR-01 Production of a proposal in response to an IPA Invitation To Tender

BNB's main field of activity is developing proposals and implementation of the proposals of IPA, WB and local development agency projects in a multitude of fields. Preparing a first response to an ITT is requires both a methodical and thorough analysis of an ITT document and extensive knowledge on the previously generated information and methodologies of the company preparing the response.

Development and use of a well-defined model to streamline the production of the response will increase efficiency of preparation of the document as well as improving its quality. This use case will also perform as a smaller scale introduction to the UC-TR-02

UC-TR-01	Production of a proposal in response to an IPA Invitation To Tender (ITT)
Versioning Info	V1.0.0 dated 28-Sep-2012
Description	Defining and implementing a base model that will be used as basis for analysing an ITT producing a response draft quickly and consistently
Actors	Business Developers, Field experts (Varying per requirements of tender), proposal production teams
Assumptions	IPA ITT documents are assumed to follow PRAG guidelines used as a basis for the produced model. Minor consistency problems are assumed to be correctable. ITT document is obtained or can be converted without significant errors to a document format supported by ModelWriter.
Steps	ITT document is obtained from official source. ITT document is verified to be or corrected to supported document formats A predefined model consistent with input and the desired output is selected ITT is parsed nad a proposal draft is auto generated by the ModelWriter for review and improvement
Variations (optional)	Multiple models may be required to create output for different type of proposals and sectors even though ITT structure remains totally consistent with the model
Non-functional (optional)	(none)
Issues	Structural problems and of the ITT that may be corrected during Q&A phase must be accounted for.

https://github.com/ModelWriter/Project-Management/issues/8

2.2. UC-TR-02 Collaborative production of a proposal for an IPA project

UC-TR-02	Collaborative production of a proposal for an IPA project
Versioning Info	V1.0.0 dated 28-Sep-2012
Description	A larger scale use case of proposal draft production using Terms of Reference (ToR) as well as Instructions to Tenderers (ITT) as inputs to the same proposal model
Actors	Business Developers, Field experts (Varying per requirements of tender), proposal production teams, Proposal manager
Assumptions	IPA ITT and ToR documents are assumed to follow PRAG guidelines used as a basis for the produced model. Minor consistency problems are assumed to be correctable. ITT document and ToR are obtained or can be converted without significant errors to a document format supported by ModelWriter.
Steps	ITT document and ToR are obtained from official source. Documents are verified to be or corrected to supported document formats A predefined model consistent with inputs and the desired output is selected Documents are parsed and a proposal draft is auto generated by the ModelWriter for review and improvement Collaborators are defined to the system Knowledge bases to be utilized by collaborators are defined to the system Rest of the steps as described in 3.1
Variations (optional)	Multiple models and/or sub-models may be used to produce a more complex proposal
Non-functional (optional)	(none)
Issues	(none)

https://github.com/ModelWriter/Project-Management/issues/8

2.3. UC-TR-03 Synchronization of ReqIF/Clafer models from requirement specifications

This Use Case aims at applying ModelWriter on Clafer's feature models to generate documentation for domain models, architectural templates and product variants.

UC-TR-04	Synchronization of ReqIF/Clafer models from requirement specifications
Versioning Info	V1.0.0 dated 14-May-2013
Description	Show that the ModelWriter's artefacts can be fruitful for synchronization of ReqIF/Clafer's feature models and required documentation. In this way, the architects/developers ought not to reproduce the documentation due to changes or inconsistencies (dependencies and constraints between features in specification) in feature models.
Actors	Architect, developer
Assumptions	(none)
Steps	• From requirements specification, ModelWriter creates a ReqIF/Clafer representation • The ReqIF/Clafer model is generated and verified using Alloy analyser. Alloy analyser can find counter-examples which show inconsistencies or conflicts in feature model.
Variations (optional)	(none)
Non-functional (optional)	Interpretation of first-order logic from natural language can be important (we consider industrial validations and checking).
Issues	It is expected that using ModelWriter we will have models from ReqIF/Clafer documents which include desired rules of developer in natural language. So it is also expected that ModelWriter can interpret natural language text, e.g. in SBVR (Semantic of Business Rules and Vocabulary) format, to generate its model.

https://github.com/ModelWriter/Project-Management/issues/8

2.4. UC-TR-04 Requirement Engineering for System Modelling

Developing applications and tracing the lifecycle of the applications is a very complex task and it has many inherent challenges. Application lifecycle includes the entire time from the idea of developing the application to the end of application's life. In general, the main lifecycle activities of Application Lifecycle Management (ALM) are reported as follows: project & portfolio management, requirements management, architecture and design, software development, software testing, software configuration management, change (issue & defect) management, and build & release management.

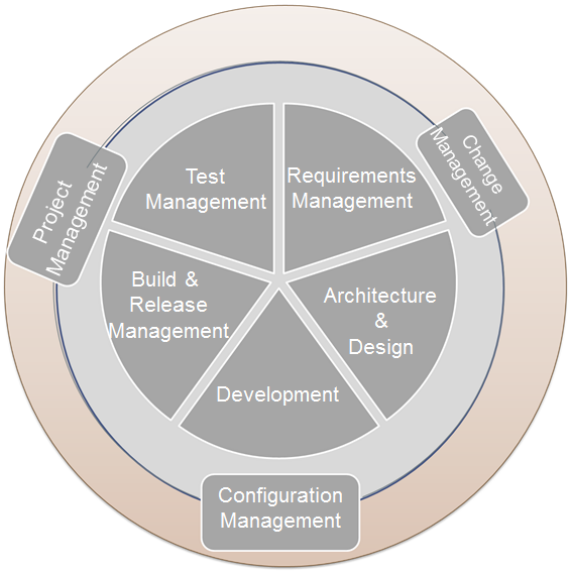

Figure 1. ALM basic activities

A lot of the big software development companies have made a lot of investments to build an integrated ALM platform which would cover the entire lifecycle of an application.

This Use Case aims at integrating ModelWriter with requirements management portion of the Application Lifecycle Management (ALM) tools. In a typical ALM platform, "requirement" is represented as a structured requirement object rather than text.

We defined a structured requirement object as an entity, that have a set of lifecycle states, defined transitions between these states, history of changes, related attributes (unique id, priority, severity, validation method etc.), and traceability information to various project artefacts such as other requirements, design model, source code, bugs, test cases etc.

Here, the textual representation is regarded as de-normalized output of the live object, however it is still needed because of contractual obligations in CMMI type projects.

"Context Diagram for Traceability"

https://github.com/ModelWriter/Project-Management/issues/8

UC-TR-03	Integration with ALM tools
Versioning Info	V1.0.0 dated 28-Apr-2015
Description	Show that the ModelWriter can extract required elements from structured requirement objects in ALM tool to generate automatic design model. Show that the ModelWriter's can generate requirements specification document from structured requirement objects in ALM tool.
Actors	Requirements Engineer/Manager, System/Software Architect
Assumptions	• All the traceability information would continue to be followed from the structured requirement object via ALM tool. (ModelWriter does not need to have any traceability information) • Textual representation of requirements are stored in MS Word documents. • Design models are stored in Sparx Systems Enterprise Architect. • Structured requirement objects are stored in Microsoft Team Foundation Server.

Steps	• Scenario1:
	  ○ From a set of structured requirement objects in the ALM tool, ModelWriter would create a natural-language text requirement document.
	• Scenario2:
	  ○ From a set of structured requirement objects in the ALM tool, ModelWriter would create/synch a design model.
	  ○ From a design model, ModelWriter would create/synch to a set of structured requirement objects.

Variations (optional)	Scenario1: From a natural-language text requirement document, ModelWriter would create/synch to a set of requirement objects.
Non-functional (optional)	The system should have a Word plugin for natural-language text transformation. The natural-language support should be in both English and Turkish
Issues	A common format between ALM platforms and ModelWriter may need to be implemented.

https://github.com/ModelWriter/Project-Management/issues/8

2.5. UC-TR-05 Synchronous Business Process Design with Use Cases

Business processes can be described with diagrams, e.g. BPMN diagrams, or as text. Use cases are a text-based notation. They are semiformal: a business process is expressed as a sequence of steps and each step is presented in a natural language. We aim to synchronize diagram-based and text-based notations and provide required tool.

UC-TR-05	Synchronous Business Process Design with Use Cases
Versioning Info	V1.0.0 dated 17-May-2013
Description	Show that the ModelWriter's semantic parser can extract required elements (e.g. activities, actors etc.) for business process model from use cases to generate automatic design documentation. Also ModelWriter's mechanism for transforming model-to-text can be applied in this case study to generate formally specified use case documents from business process models.
Actors	System Analyst, SW designer and developer
Assumptions	Ontological based transformation from model-to-text and vice versa.
Steps	This case study will show that use case documents can be converted to related business process models and vice versa.
Variations (optional)	(none)
Non-functional (optional)	(none)
Issues	Dynamically created actors and actor metamorphosis

https://github.com/ModelWriter/Project-Management/issues/8

References

N/A