

D1.8.1 Technical Risk Assessment

ModelWriter

Text & Model-Synchronized Document Engineering Platform

Work Package: WP1

Task: T1.8 – Technical Risk Assessment and Management

Edited by:

Marwa Rostren marwa.rostren@obeo.fr (OBEO)

Yvan Lussaud yvan.lussaud@obeo.fr (OBEO)

Date: 01-June-2015

Version: 1.0.0

Apart from the deliverables which are defined as public information in the Project Cooperation Agreement (PCA), unless otherwise specified by the consortium, this document will be treated as strictly confidential.

Document History

Version	Author(s)	Date	Remarks
0.1.0	Marwa Rostren	27-Apr-2015	Draft
1.0.0	Yvan Lussaud	01-Jun-2015	Initial Release

Table of Contents

DOCUMENT HISTORY	2
1. INTRODUCTION	4
■ ROLE OF THE DELIVERABLE	4
■ STRUCTURE OF THE DOCUMENT.....	5
■ TERMS, ABBREVIATIONS AND DEFINITIONS.....	5
2. TECHNICAL RISKS BASING ON THE DEFINED REQUIREMENTS	6
■ USER REQUIREMENTS	6
2.1.1. <i>Risks and solutions</i>	<i>8</i>
■ SOFTWARE REQUIREMENTS	8
2.2.1. <i>Risks and solutions</i>	<i>10</i>
3. TECHNICAL RISKS BASING ON THE EVALUATION OF TECHNOLOGIES.....	12
■ RISKS AND SOLUTIONS.....	14
3.1.1. <i>R1.X-y Risk for Industrial use cases</i>	<i>14</i>
3.1.2. <i>R2.5-1 Risk associated to Natural Language Processing</i>	<i>14</i>
3.1.3. <i>R4.6-1 Semantic Comparison</i>	<i>15</i>

1. Introduction

|Role of the deliverable

This document is the first version of the Technical Risk Assessment (TRA). It defines all possible risks on the achievement of the project.

As responsible of the project results, the ModelWriter Consortium needs to define, as soon as possible after the definition of the user and software requirements (T1.5 and T.6), the limitations of the proposed approaches and technologies.

In this document, risk is measured as either expected value or expected behaviour (using the Actuarial Approach of TRA) based upon previous occurrences of unwanted actions. The proposed TRA approach assumes that *“sufficient data exists to make meaningful predictions about future events and that the causal mechanism that underlies the occurrence of previous undesirable events will remain stable over the prediction period”*.

Assessing the technical risk of the project must imperatively be based on identifying and evaluating the maturity of each technology for each work package and expressing the expected results of each. That's why; the risks defined in this document are based either on the user and the software requirements (see T1.5 and T1.6) or on the previous experience of partners with their proposed technologies.

For each of the identified risks, the consortium must provide a way to control it, and if possible how could we propose an alternative solution if risk cannot be assessed.

The ModelWriter Consortium has set up a process allowing monitoring risks during the implementation of the tool:

- **Technical risks control** is especially addressed by the current document as a deliverable of WP1 "Technical Risk Assessment and Management". For short: all risks are monitored throughout the project, with the special involvement of the **key technical leaders** of WP6 "ModelWriter Architecture, Integration and Evaluation", and the **technical leaders of technological component** research & development WP2 to WP4 leaders. **Technical risks control** is also implemented by the participation of Technical leaders to all technical work packages (apart from the WP6 integration), so as to ensure the technical consistency of the ModelWriter product.
- **PRINCE2 Project Management method**, on the other hand, proposes concrete "best practices" techniques for capturing and monitoring risks (e.g. use of a Risk Register, how to properly formulate risks and opportunities, and what actions can be imagined to address risks, etc.). This is included in WP5 "Project Management".
- **Scrum**, the leading agile framework for building complex products in an iterative and incremental manner, is also a part of the selected practices (-> "Minimally Viable Products" are considered first, in a stepped approach).
- Finally, it must be noted that ModelWriter FPP and PO have been iterated multiple times (ModelWriter-2012, ModelWriter-2013) and each time the proposal has systematically been reinforced so as to always further address this "technical risk" concern. This is why a number of Tasks within WPs have been added to already mitigate all foreseeable technical risks. This is why (among others) we have a Task T6.1 "Experimental Prototyping", T1.7 "Annual Product Review" and an overall **iterative & adaptive approach** (-> "intermediary goals") to the overall project logic.

The document may be up-dated throughout the project with special review at the same time as for the software requirements and the architectural design review, depending on the further details and requirements we get from the industrial use case providers.

Structure of the document

This document is organized as follows:

- Chapter 1 introduces the document.
- Chapter 2 describes the list of all risks provided basing on requirements of T1.5 and T1.6.
- Chapter 3 describes the list of all risks provided basing on the evaluation of each technology for each work package.

Terms, abbreviations and definitions

Abbreviation	Definition
TRA	Technical Risk Assessment
WP	Work Package
SW	Software
UC	Use Case

2. Technical Risks basing on the defined requirements

All the user and the software requirements are defined as issues on github:
<https://github.com/ModelWriter/Requirements>

User Requirements

The user requirements below are retrieved from the list of User Requirements Document labelled issues on April 27th.

It may be up-dated further details and additional requirements we get from the industrial use case providers.

Requirement	Risk	Comment
UReq01 - BPMN 2.X standard shall be supported as a user visible model	NONE	There is no risk related to this requirement.
UReq02 - The system should support ReqIF standard as user visible model.	NONE	There is no risk related to this requirement.
UReq03 - The system must support an open requirement authoring tool (such as RMF)	NONE	An RMF project is already eclipse integrated https://eclipse.org/rmf/ This will be studied to integrate in the ModelWriter product.
UReq04 - The System shall propose an eclipse editor to handle documentation/models mappings	NONE	The ModelWriter editor will be used to handle the mappings with models elements and to handle the synchronization of existing mapping basing on the current status of the documents and models contents.
UReq05 - The System shall not enforce any dependency on non-open source artefacts such as tools, applications and libraries	NONE	MW will allow the use of open document editors and open graphical modelers
UReq06 – The user does not want to be bothered with information such as mapping links	X	Those links shall also be a technical artefacts from the knowledge base at not shown to the user directly. Also links should be saved in a non-intrusive manner from a document or a model point of view. Thus ModelWriter must provide a reconciler mechanism to handle with locating the right information at the right place.
UReq07 - The System shall allow to filter synchronization warnings, errors and information	NONE	Eclipse PDE provides filters. We can base on the existing eclipse UI filters to provide ModelWriter specific filters.
UReq08 - The System shall allow users to work in collaborative manner	This might have impact on the architecture.	Different mode of collaborative work can be acceptable: <ul style="list-style-type: none"> - Connected mode - Disconnected mode with comparison and reconciliation In practice user will alternate through both modes.
UReq09 - The System shall offer a	NONE	There is no risk related to this requirement.

Requirement	Risk	Comment
notification system		
UReq10 [UC-FR-02] The System shall be easy to integrate to an Eclipse Rich Client Platform application (RCP).	NONE	There is no risk related to this requirement.
UReq11 - The System shall allow the user to activate/deactivate a synchronization direction	NONE	There is no risk related to this requirement.
UReq12 - The system shall allow the end user to keep his/her usual working environment	This might be difficult to prove if the MW answer is that you must develop the right connector...	For Airbus case, the end-user editor is MS-Word.
UReq13 - The system shall allow the end user to edit text and "visual" model (such as tables, diagrams or 2D drawings) synchronously	This might add items to the list of To be supported editors	To limit the workload we should think of contributing our commands and views to existing editors. But we will have to provide specific code in the background to handle references.
UReq14 - The system shall allow semantic retrieving or reasoning using the model elements	This might introduce the need for other mechanisms than synchronization	
UReq15 - A specification for an improve and controlled formulation of the rules in semi-structured natural language	NONE	There is no risk related to this requirement.
UReq16 - The system shall provide a user friendly way to manage any additional concepts needed	This might introduce inconsistencies in the Knowledge Base	Use KB techniques to detect inconsistencies when extending KB with new concepts
UReq17 - The system shall show on demand (coloured mark or other mean) the text elements that are linked to the "visual model" concepts, and conversely	NONE	There is no risk related to this requirement.
UReq18 - ModelWriter should support at least one Document Markup Language and one Lightweight Markup Language	NONE	There is no risk related to this requirement.
UReq19 – The system shall allow the user to configure the document	X	A change in a model might impact different documents (in different ways)

Requirement	Risk	Comment
generation content		
UReq20 – The system shall help to synchronize the SIDP natural language document with the modeled rules without forced modification.	The gap between natural language expressions and model elements cannot be bridged	Reduce the synchronisation to those elements for which a mapping between natural language expressions and model elements can be predefined and stored in a dictionary
UReq21 – The system should be able to perform semantic parsing.	Ambiguities cannot be resolved	Work on normalised text
UReq22 – The system shall provide a unified Graphical User Interface (for both Model and Writer parts).	NONE	There is no risk related to this requirement.
UReq23 – “MW” Knowledge Dissemination Standard	NONE	There is no risk related to this requirement.
UReq24 – ModelWriter shall support Rich-Blended Modeling Environments.	NONE	There is no risk related to this requirement.
UReq26 – ModelWriter as a Next Generation Requirements Engineering Tool: ModelWriter should be equipped with Requirements Engineering features.	NONE	There is no risk related to this requirement.

2.1.1. Risks and solutions

The list of all risks R-UReq-x-z basing on the evaluation of a user requirement can be found in the sub-sections below. Each sub-section is related to only one risk which is described briefly by the Consortium members. For each identified risk, the project Consortium is invited to discuss and find a solution to resolve the concerned risk.

Software Requirements

The software requirements below are also retrieved from the list of Software Requirements Document labelled issues on April 27th.

It may be up-dated further details and additional requirements we get from the industrial use case providers.

Requirement	Risk	Comment
SReq01 - Transformation Manager Plug-in must be a component in the M2M Transformation Framework	NONE	There is no risk related to this requirement.

Requirement	Risk	Comment
SReq02 - Configuration Manager Plug-in must be a component in the M2M Transformation Framework	NONE	There is no risk related to this requirement.
SReq03 - Traceability Manager (TRAM) Plug-in must be a component in the M2M Transformation Framework	NONE	There is no risk related to this requirement.
SReq04 - Synchronization Manager Plug-in must be a component in the M2M Transformation Framework	NONE	There is no risk related to this requirement.
SReq05 - The Transformation Framework must be built on top of the Eclipse Modelling Framework (EMF)	NONE	There is no risk related to this requirement.
SReq06 - Transformation Manager shall obtain one or several output models from one or several input models.	NONE	There is no risk related to this requirement.
SReq07 - M2M Transformation Framework should configure the transformations to be able to produce different outputs using the same inputs.	NONE	There is no risk related to this requirement.
SReq08 - M2M Transformation Framework should compose simple transformations to obtain more complex ones.	NONE	There is no risk related to this requirement.
SReq09 - M2M Transformation Framework must keep traces between transformed models and its source models.	NONE	There is no risk related to this requirement.
SReq10 - M2M Transformation Framework must synchronize the output models after its input models or configurations have been modified.	NONE	There is no risk related to this requirement.
SReq11 - A mechanism is needed to register the available transformations in ModelWriter	NONE	There is no risk related to this requirement.
SReq12 - The TRAM validates such parameters and also the input models before a transformation takes place.	NONE	There is no risk related to this requirement.
SReq13 - The TRAM will be able to	NONE	There is no risk related to this requirement.

Requirement	Risk	Comment
compose transformations.		
SReq14 - The Documentation Editor shall not display explicitly links to the model	X	Since mapping will be registered in the knowledge base. And Since the knowledge will never save last versions of models and documentation. ModelWriter must provide a reconciler mechanism to handle with locating the right information at the right place.
SReq15 - Text Connector should support Office Open XML (docx) standard.	NONE	The text connector can support the office open XML format using the apache poi API
SReq16 - Text Connector should support Textile (textile) and/or Markdown (mw) standards.	NONE	This requirement can be supported by using a correct markup model representing textile and other markdown standards.
SReq17 - The Editor shall be able to display choices to facilitate the selection of concepts	X	This requirement is related on the results of the WP2. The semantic parser and the semantic annotations results.
SReq18 - The concepts proposal shall be provided in acceptable time for human being	X	This requirement provides one significant risk related to the performance of the MW product.
SReq19 - The system shall provide a new keyboard shortcut to weave concepts and make mappings	NONE	This requirement is feasible and does not provide any risk.
SReq20 - The system shall store the mapping (text - concept) in the knowledge base	NONE	The knowledge must be robust enough to support storing all the needed information. There is no risk if the design of the knowledge is robust enough and supports all the needed features information.
SReq21 - The system shall provide an optional highlighting action to show mapped text parts in the editor	NONE	This requirement is related to the editor.
SReq22 - The system shall provide an additional glossary view	NONE	This requirement is related to the eclipse integrated MW product. It can be realized later (major release 2 or major release 3)
SReq23 - The notification system shall indicate the de-synchronization and the reconciliation failures	NONE	This requirement is related to the MW editor, eclipse views and on the notification and logger system we need to implement. This does not provide any risk.
SReq24 - The system shall be a standalone	NONE	The components responsibilities must be separated from all IHM issues.

2.2.1. Risks and solutions

The list of all risks R-SReq-x-z basing on the evaluation of a software requirement can be found in the sub-sections below. Each sub-section is related to only one risk which is described briefly by

the Consortium members. For each identified risk, the project Consortium is invited to discuss and find a solution to resolve the concerned risk.

3. Technical Risks basing on the evaluation of technologies

The major value added to the ModelWriter project is the identification of integration risks and dependencies, particularly across work packages.

We first define the set of all proposed technologies per work package. Then we propose to discuss the technical related risks basing on technologies evaluation.

WP identifier	SW Deliverables	Technologies	Risk	Comments
WP1	NONE	NONE	NONE	All the deliverables of this WP are documents.
WP2	T2.5.1 Semantic Parsing	- Natural language processing		Restrict semantic parsing to normalised text Extend the size of the training corpus
	T2.3	- Common Representation Language for Parsing and Generation	Mismatch between the representations produced	Define a mapping between the representations produced by the parser and used by the generator
	T2.1.2	- Data Collection	Data (text or models) is not available	Work with outside data so as to be able to make progress and test the ModelWriter architecture
	T2.5.2 Natural Language Generation	- Natural language processing	Incorrect lexicalisations	Restrict generation to input for which lexicalisation are given
	T2.5.2 Natural Language Generation	- Natural language processing	Ungrammatical or disfluent Output	Use templates (instead of grammars or stochastic methods)
WP3	T3.3 The transformation manager component	- M2M approach	NONE	There is no risk related to this requirement.
	T3.4 The configuration manager component	NONE	NONE	There is no risk related to this requirement.
	T3.5 The traceability manager component	NONE	NONE	There is no risk related to this requirement.
	T3.6 The synchronization manager component	NONE	NONE	There is no risk related to this requirement.

WP identifier	SW Deliverables	Technologies	Risk	Comments
WP4	T4.3 The knowledge base	- EMF based approach	NONE	There is no risk related to this requirement.
	T4.4 The proof-of-concept model checker	- EMF based approach - Syntactic comparison	NONE	This part does not introduce any risk. Since the used technologies are already used by technical partners and previous experience with the Intent tool ensures that the model checker is feasible.
	T4.6 The proof-of-concept semantic comparison	- EMF based approach - Semantic comparison	X	The technical risk related to this task is the same related to the natural language processing, it introduces the same technological risk as for the T2.5. More efforts are expected WP4 Leader is not able today to clearly define the needed approaches to make semantic comparisons possible in ModelWriter.
WP5	NONE	NONE	NONE	All the deliverables of this WP are documents.
WP6	T6.1 The prototype	- EMF - Apache poi API - M2M Approach	NONE	Even if the poi API is not an exhaustive API. Contributing to the API is possible to improve it. That's why this task does not present any technological risk.
	T6.3 The writer enhancements	NONE	NONE	There is no risk related to this requirement.
	T6.4 The User Interface	- Eclipse - SWT	NONE	These technologies are used by Obeo's teams and no risk is identified.
	T6.5 The automated acceptance Tests	- Junit - RCPTT	NONE	There is no risk related to this requirement.
	T6.7 The integration and major releases	- Eclipse - Maven	NONE	There is no identified risk for this task.
WP7	D7.4 The project website	- Joomla framework - PHP	NONE	the website has a secured access (https://modelwriter.eu/admin) In addition, it has internal local access on Obeo's servers to ensure its content recovering in case of hack problems.
	T7.6 The project social groups	- LinkedIn - Twitter	NONE	We use existing social groups which present no technological risk.
	T7.7 The project and standardization	- ISO standardisation process	X	ISO's standards related to terminology, knowledge, content, and linguistic resources management should be

WP identifier	SW Deliverables	Technologies	Risk	Comments
	activities			<p>considered, as well as other international standards such as ISO/IEC/IEEE 29148:2011 and ISO/IEC/IEEE 42010:2011.</p> <p>Using standards frequently imposes a framework that may not be flexible enough for a fast development of prototype or testing tools.</p>

Risks and solutions

The list of all risks Rx.y-z basing on the evaluation of WPx technologies can be found in the sub-sections below. Each sub-section is related to only one risk which is described briefly by the Consortium members. For each identified risk, the project Consortium is invited to discuss and find a solution to resolve the concerned risk.

3.1.1. R1.X-y Risk for Industrial use cases

By choice ModelWriter will use open technologies and cannot develop connectors to all possible users' usual working environment tools (e.g. MSWord editor??). This will limit integration capability within real industrial environments, and consequently limit the validation of the technical usability.

3.1.2. R2.5-1 Risk associated to Natural Language Processing

Natural Language Processing is often perceived as an important challenge and therefore causing an important **technical risk** on a **usable** ModelWriter. However, both the likelihood and impact of this risk are not totally as expected at first glance by the reviewer. Indeed:

- The consortium includes specialists on the subject of Natural Language Processing, such as LORIA (France) and KUL (Belgium) universities, making the NLP risk with lower likelihood.
- For the transformation of pieces of text (like those expressing requirements on aerospace products) to an Object-Role Modeling (ORM) model, we can already rely on internal technical notes demonstrating the ideas on actual examples.
- Very importantly, we will not parse arbitrary documents like Google may do, but **technical documents** who have multiple constraints, e.g. they must comply to company and domain structure rules and have been written with as less ambiguity as possible, even sometimes (e.g. for expressing requirements) using template-based phrases. Also, we do not envision nor consider as suitable (given the use cases) to fully automate NLP without any human in the loop.

With these elements in mind, we will therefore address a **niche of research** where results are obviously much more easily achievable: the risks are definitively not the ones usually associated when the word "NLP" is used in a very generic and arbitrary manner, for a completely open domain.

- Last and not least: being able to parse natural language is definitively **not a prerequisite for having a functional ModelWriter**. If we ever do not manage to parse natural language, despite the specialist on-board, and hence do not ever manage to deliver the

corresponding NLP deliverables, then we **will still have a useful ModelWriter** synchronization platform able to synchronize models and documentation.

A typical example is to assess that semantically parsing a part of text cannot be possible all the time. To control that risk, we will define constraints on the document (e.g. specify its structure). We also propose an alternative solution keeping the ability to manually link the same part of text with an element of the Knowledge Base by providing a tooling allowing to drag a model element on this piece of text (e.g. The same behaviour is available in Intent).

3.1.3. R4.6-1 Semantic Comparison

The technical risk related to the T4.6 is exactly the same addressed by the natural language processing; it introduces the same technological risk as for the T2.5 since it is also often perceived as an important challenge and therefore causing an important **technical risk** on a **usable** ModelWriter.

The Semantic comparison is already addressed in the literature like an emerging aspect of Natural Language Processing. The Short Text Semantic Similarity method and the Lightweight Semantic Similarity method address both the same issue.

Much more research efforts are expected at the WP4 leadership level to clearly define the needed approach to make semantic comparisons possible in ModelWriter.