Multi-domain modeling assistance tool

Technische Universiteit
Eindhoven
University of Technology

/ Department of
Mathematics and
Computer Science
/ PDEng Software
Technology

Multi-domain Modeling Assistance Tool: Design and Implementation

Munkhdalai Sainbileg

Where innovation starts

Multi-domain Modeling Assistance Tool: Design and Implementation

Munkhdalai Sainbileg

Eindhoven University of Technology
Stan Ackermans Institute / Software Technology

Partners	TNO IVS department	Eindhoven University of Technology
Steering Group	Arash Khabbaz Saberi (TNO project owner and supervisor) Mark van den Brand (TU/e supervisor) Serguei Roubtsov (TU/e supervisor)	
Date	October 2019	
Document Status	Confidential	
PDENG report no.	2019/092	

The design described in this report has been carried out in accordance with the TU/e Code of Scientific Conduct.

Contact Address	Eindhoven University of Technology Department of Mathematics and Computer Science MF 5.080A, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands +31402474334

Published by	Eindhoven University of Technology Stan Ackermans Institute
Printed by	Eindhoven University of Technology *UniversiteitsDrukkerij*
PDEng Report No.	2019/092

Abstract	These comments should be the abstract issued for the SAI report no. request.

Keywords	Conceptual modeling, ISO 26262, Integrated Vehicle Safety (IVS), Modeling assistance tool
Preferred reference	MULTI-DOMAIN MODELING ASSISTANCE TOOL: DESIGN AND IMPLEMENTATION, SAI Technical Report, October 2019. (2019/092)
Partnership	This project was supported by Eindhoven University of Technology and TNO IVS department.

1. Foreword

Conceptual modeling is a powerful tool for bringing clarity and consistency to complex ideas within a specific domain. In most cases, there are specifications (written in natural langue) of a particular domain-specific topic that seem to be agreeable for the community in that domain. However, in practice various viewpoints and differences in interpretations get in the way of effective communication and collaboration on that topic. There is literature that supports this idea that domain models (or meta-models) facilitate sharing information and ensures consistency of resulting designs. One of the challenges in this field is that creating these models requires substantial effort. It usually involves analysis of an original text, followed by discussions regarding the definitions and relationships between concepts in the target domain. During this process, it is essential to keep the trace between decisions to model elements as well as the original text. The objectives of this project were to design and implement a software solution that facilitates these activities and increase the effectiveness of modeling efforts. In this project, we took the first step in this direction by implementing a prototype solution for Multi-domain Modeling Assistance (MMA) tool. This prototype provided much-needed knowledge on requirements for such tool, and we hope that future projects take on the improvement points and continue developing MMA.

Arash Khabbaz Saberi / PROJECT MENTOR
23-10-2019

2. Preface

This report describes the project "Multi-domain Modeling Assistance Tool: Design and Implementation." This project addresses the challenges of a conceptual modeling approach of informal or semiformal specifications in natural languages such as standardization and norms.

This project was executed by Munkhdalai Sainbileg from the Stan Ackerman's Institute, PDEng Software Technology program of the Eindhoven University of Technology. The project serves as a final project for the Software Technology (ST) program for the Eindhoven University of Technology (TU/e), towards the Professional Doctorate in Engineering (PDEng) degree. The project was implemented at the Integrated Vehicle Safety (IVS) department of TNO in Helmond.

This report is primarily targeted for readers with a technical background in disciplines, such as Object-Oriented Principle (OOP), Model-Based Engineering (MBE), Unified Modeling Language (UML), and general software engineering.

Readers with a non-technical background or who are interested in knowing the goals and results of this project should read Chapters 1-3 and 10-11, which describe the essential points which the tool is aimed, and the project results that we achieved with this project.

Readers who are interested in the detailed technical solution of this project should read Chapters 3-9. These chapters cover technical details, namely: requirement specification, architecture, design, implementation, deployment, and validation of the tool.

Munkhdalai Sainbileg
October 2019

3. Acknowledgments

This project would not have been completed without the encouragement and support of several great individuals. My appreciation belongs to all of them for being part of this journey and making this project possible.

I am very thankful to TNO IVS department scientists who contributed to this project. Special thanks to Scientist Arash Khabbaz Saberi who has been my mentor and supervisor during this project at the TNO IVS department. His continuous support, motivation, enthusiasm, and cooperation are ingredients for the success of this project. I enjoyed working with him during this project. Most of the time, we were hugely enthusiastic about this project, and the last ten months passed so fast. It was a great experience working with him. Thanks so much for everything he did for this project. I would also like to thank Scientist Dennis van der Brand. He helped with the integration of this project in the TNO IVS internal server. It was always a pleasure to work with Dennis.

I am a very blessed student who was supervised by two excellent supervisors from the university in this project. It was a unique opportunity and honor to work with them. I want to express my thankfulness to my university, Full Professor Mark van der Brand, who guided me in achieving the academic viewpoint of the project. His questions always gave me big thoughts and powerful ideas that bring new value from a different perspective for this project. I want to extend my thanks to Dr. Serguei Roubstov, my supervisor. His professional guidance and concrete feedback were vital for this project execution.

My acknowledgment goes to everybody who involved in the PDEng program. Especially Yanja Dajsuren and Desiree van Oorschot who were supporting me during the last two years. My thanks extend to all coaches and teachers of the PDEng program. I also want to thank my colleagues from the PDEng program for sharing beautiful experiences, knowledge, culture, and many other beautiful memories.

Furthermore, I would like to express my warm gratitude to my friends for their valuable support during these two years.

Finally, my deep thankfulness belongs to my wife and my gorgeous two kids for their unconditional love and endless support. This journey would not have been possible without their support and love.

4. Executive Summary

Compliance with norms and standards is a vital part of quality assurance in the automotive domain. Compliance assurance is becoming challenging due to the rapid increase of system complexity. Thanks to the introduction of new features such as automated driving, and ever-changing standards (e.g., ISO 26262:2018 the second revision on functional safety, ISO/SAE CD 21434: 2019 on cybersecurity, and ISO/PAS 21448 on the safety of the intended functionality).

The IVS department of TNO is developing innovative a tool for compliance assurance. This tool ensures the quality of the department product meets the requirements of the ISO standard, and it detects design errors in the early stage of the system development process using conceptual models, which come from the ISO standard requirement. The essential part of the tool is the creation of the conceptual models.

Conceptual modeling is a labor-intensive task for the modeler (end-user) because of a large amount of manual work in the department. The modeling process depends on the end-user's carefulness. When the user loses his/her attention, it causes several drawbacks in the resulting models. The user can easily skip or make mistakes during the modeling process due to less-structured and low-controlled manual work. As a result, inconsistent models can be produced. Therefore, a systematic modeling process is needed to support the end-user to increase the consistency of the resulting models.

Further, the modeling process has several steps to produce the resulting models. The primary source of the input, the ISO standards, includes a large number of requirement specifications. The modeler needs to be aware of associations between requirement specifications and the resulting model elements. In each step, every requirement specification should be recorded for the further traceability. Currently, the modeler manually maintains the association between the requirement specifications and the resulting model elements to trace the modeling process. The manual task is time-consuming. The manual traceability is not sufficiently scalable. Hence, some level of automation for traceability is desired.

The goal of this project is to reduce the end-user's manual work by supporting a software solution. To achieve the project objective, we designed and developed a software tool that provides a systematic process for conceptual modeling to improve the consistency of the resulting models and the traceability of the modeling process.

Besides, the tool separates conceptual modeling from identifying domain concepts. The separation allows the tool to accept various ISO standards, and it is named Multi-domain Modeling Assistance (MMA) tool. Thus, the MMA tool can be applied to different domains.

Added-value of this project is the following pair usages of the MMA tool. First, the tool contributes to centralize models by coupling with Simplexity, and the resulting models of the MMA tool can be used for TNO projects and products. Second, the tool can be useful for the different ISO standards, the norms, and the European Union (EU) project requirement.

5. Table of Contents

1. Foreword i

2. Preface iii

3. Acknowledgments v

4. Executive Summary vii

5. Table of Contents viii

6. List of Figures 11

7. List of Tables 12

1. Introduction 13

1.1 Context *13*

1.2 Motivation *13*

1.3 Background *13*

1.3.1. The standards and their structure 14
1.3.2. Conceptual modeling 14
1.3.3. Holistic Domain model and the proposed process 15
1.3.4. Natural Language Tasks study and the result it 17

1.4 Outline *17*

2. Problem Analysis 18

2.1 Problem statement *18*

2.2 Project goal *20*

3. Stakeholder Analysis 21

3.1 TNO IVS department *21*

3.2 Eindhoven University of Technology (TU/e) *22*

4. Tool Requirements 24

4.1 Requirement Gathering *24*

4.1.1. Interviewing the stakeholders 24
4.1.2. Prototyping the proposed software tool 27

5. Tool Architecture 34

5.1 Overview of MMA tool *34*

5.2 Process model of the MMA tool *35*

5.3 Architecture pattern of the MMA tool *35*

5.3.1. Microservice 35
5.3.2. Batch sequential 36
5.3.3. Three-tier 36
5.3.4. The decision of the architecture pattern 36

5.4 Deployment view of MMA tool *39*

5.5 Sequence Diagrams and Screens ... *40*
5.5.1. Sequence diagram and Screen of the Preprocess step ... 40
5.5.2. Sequence diagram and Screen of the Parsing step ... 41
5.5.3. Sequence diagram and Screen of the Analysis step ... 43
5.5.4. Sequence diagram and Screen of the Initial modeling step ... 44

6. Tool Design ... 46

6.1 Design decision and constraint ... 46

6.2 Technology decisions and rationales ... 47
6.2.1. Presentation tier technologies ... 47
6.2.2. Application tier technology ... 47
6.2.3. Data tier technology ... 48

6.3 The presentation tier design ... 49
6.3.1. Angular concept ... 49
6.3.2. Component decomposition ... 49
6.3.3. Service ... 52

6.4 Component diagram of MMA tool ... 53

6.5 The application and data tier design ... 54

7. Tool Implementation ... 55

7.1 Implementation of the presentation tier design ... 55

7.2 Implementation of the application tier design ... 56

7.3 Implementation of the data tier ... 56

8. Tool Deployment ... 57

8.1 Introduction ... 57

8.2 Deployment of the presentation tier ... 57

8.3 Deployment of the application tier ... 58

8.4 Deployment of the data tier ... 58

9. Tool Validation ... 60

9.1 Validation ... 60
9.1.1. Requirement satisfaction of the MMA tool ... 60
9.1.2. Testing of the MMA tool ... 61
9.1.3. The presentation tier testing ... 62
9.1.4. The application tier testing ... 62

10. Conclusions ... 64

10.1 Results ... 64

10.2 Future work ... 64

11. Project Management ... 66

11.1 Introduction ... 66

11.2 Project planning and milestones ... 66

11.3 Project implementation ... 67

11.4 Project integration ... 67

12. Acronyms 68

13. References 69

14. Appendices 71

15. Appendix A 72

16. Appendix B 73

17. Appendix C 74

18. Appendix D 78

18.1 Natural Language Processing Tasks 78

18.2 Fundamental NLP tasks 78

18.3 NLP task experiment 79

18.4 Decisions on the NLP tasks 80

6. List of Figures

Figure 1. The modeling method of the Holistic domain model [10]. 15
Figure 2. The overview of the proposed process for the conceptual modeling 15
Figure 3. Examples of consistent and inconsistent models. 19
Figure 4. An example of the traceability of the modeling process 19
Figure 5. The manual modeling process and the automated solution.........................20
Figure 6. Welcome view in the mock-up 27
Figure 7. Preprocess view in the mock-up 28
Figure 8. Overview of the preprocess view in the mock-up.......................................29
Figure 9. Parsing view in mock-up..30
Figure 10. Analysis view in mock-up..31
Figure 11. Initial Modeling view in the mock-up...32
Figure 12. The overview of the MMA tool 34
Figure 13. Process model of the MMA tool 35
Figure 14. Microservice architecture pattern...36
Figure 15. Batch sequential architecture pattern 36
Figure 16. Three-tier architecture pattern of MMA...37
Figure 17. The deployment view of the MMA tool...39
Figure 18. Preprocess screen...40
Figure 19. Sequence diagram of the Preprocess step 41
Figure 20. Parsing screen 41
Figure 21. Sequence diagram of the Parsing step..42
Figure 22. The analysis screen 43
Figure 23. Sequence diagram of the Analysis step..44
Figure 24. Initial Modeling screen 44
Figure 25. Sequence diagram of the Initial Modeling step..45
Figure 26. The derived design decision and constraint 46
Figure 27. Presentation tier with the technologies...47
Figure 28. Application tier with the technology..47
Figure 29. Data Tier with the technology..48
Figure 30. The welcome component screen 50
Figure 31. PreprocessOverview Component screen..50
Figure 32. The term checker in the Preprocess step 51
Figure 34. The term checker in the Parsing step 51
Figure 35. Term checker in the Analysis step 52
Figure 36. The component diagram of the MMA tool 53
Figure 37. Data models are generated by Django graph extension 54
Figure 38. The implementation of the presentation tier design 55
Figure 39. The implementation of the application tier design...................................56
Figure 40. The implementation of the data tier design..56
Figure 41. Deployment process on the command line of Visual Studio Code tool....57
Figure 42. Uploading the result of deployment using FileZilla.................................58
Figure 43. Installation for required packages in a virtual environment......................58
Figure 44. The deployment of the data tier 59
Figure 45. Angular CLI test 62
Figure 46. Jasmine framework window 62
Figure 47. Unit test of the application tier...63
Figure 48. Project overview planning 66
Figure 49. Milestones of Phase 1 of this project 66
Figure 50. Milestones of Phase 2 of this project 67
Figure 51. Milestones of Phase 3 of this project 67
Figure 52. POST extraction...80

7. List of Tables

Table 1. The structure of the ISO 26262 14
Table 2. An example of the proposed process overview 16
Table 3. An example of the manual modeling process. 18
Table 4. IVS department stakeholder overview 21
Table 5. TU/e stakeholder overview 22
Table 6. High-level functional requirements 24
Table 7. Detailed functional requirements 25
Table 8. Non-functional requirements with specification and rationale 26
Table 9. Features of the preprocess view 28
Table 10. Features of the overview of the preprocess view 29
Table 11. Features of the parsing view 30
Table 12. Features of the analysis view 31
Table 13. Features of the initial modeling view 32
Table 14. The comparison matrix of the candidate architecture patterns. 37
Table 15. The design decision and constraint with its rationale 46
Table 16. Requirement specification with justification 60
Table 17. KISS concepts 72
Table 18. Definition table for terms of the requirement specifications 73
Table 19. Detailed requirement justification 74
Table 20. Fundamental NLP tasks 78

1. Introduction

This chapter gives an introduction to the project by briefly revealing its context, the motivation for doing this project, and background information. The chapter is concluded with an outline that provides an overview of the structure of this report.

1.1 Context

The PDEng degree program in Software Technology is provided by the Department of Mathematics and Computer Science of the Eindhoven University of Technology in the context of the 4TU School for Technological Design, Stan Ackerman's Institute. It is a two-year, third-cycle (doctorate-level), engineering degree program designed to prepare a trainee for an industrial career as a technological designer, and later on as a software or system architect. The focus of the program is on strengthening technical and non-technical competencies of the trainees related to efficient, elegant, and efficient design, as well as the development of software for resource-constrained software-intensive systems, such as real-time embedded or distributed systems, in an industrial setting. Mainly, the focus is on designing and large-scale design and development of this type of software.

TNO is a research organization in The Netherland and is recognized by Original Equipment Manufacturers (OEM) and their tiers as a valuable knowledge partner with unique expertise, tools, and facilities. The Integrated Vehicle Safety (IVS) department of TNO researches innovative methodologies for safety assessment and assurance of the automated vehicles.

1.2 Motivation

Compliance with norms and standards is an essential part of quality assurance in the automotive domain. The challenge of compliance assurance is becoming significant, resulting from an accelerated increase in system complexity. The ISO standards and norms (e.g., ISO 26262:2018 the second revision on functional safety, ISO/SAE CD 21434: 2019 on cybersecurity, and ISO/PAS 21448 on the safety of the intended functionality) are being improved continually because of primary needs of autonomous driving and smart mobility.

The IVS department of TNO is promoting innovative a tool for compliance assurance. This tool ensures the quality of the department product meets the requirements of the ISO standard, and it detects design errors in the early stage of the system development process using conceptual models, which come from the ISO standard requirement. The essential part of the tool is the creation of the conceptual models. Currently, conceptual modeling is a labor-intensive task for the end-user because of a large amount of manual work in the department.

This project ultimately aims to decrease the end-user's manual work by supporting an automated tool. The primary input is the ISO standards, and the outcomes are models.

1.3 Background

In this section, several fundamental concepts are introduced that are used in this project. First, the ISO standards and their structure are briefly explained. The next two subsections introduce the conceptual modeling and the conceptual modeling method sequentially. Several Natural Language Processing (NLP) tasks also are studied and, the result of NLP tasks are described at the end of this section.

1.3.1. *The standards and their structure*

The ISO standards are the primary input for conceptual modeling. International Organization for Standardization (ISO) has created and published 22834 [1] International standards that provide requirements, specifications, guidelines, or characteristics that can be used consistently ensure that materials, products, processes, and services are fit their purpose.

In this project, the primary domain is the automotive domain. In the automotive domain, IEC 61508 [2] and ISO 26262 [3] are used. Both can be used as input for the conceptual modeling of the proposed solution. However, we focused on the ISO 26262 standard because the standard is to address the sector-specific needs of electrical and/or electronic systems within road vehicles, and the ISO 26262 is an adaptation of IEC 61508.

Table 1 shows the structure of ISO 26262. Clause 1 declares to the scope of the ISO standard, and Clause 2 refers to normative references. Clause 3 introduces terms and definitions. Clauses 4-10 contain the requirements for the safety life-cycle, and Clauses 11-14 provide supporting information such as examples and tools. Clauses 3-10 include all of the essential requirements and guidelines for the modeling. In addition, Clauses 11-14 can be used for modeling.

Table 1. The structure of the ISO 26262

ID	Clauses	Definition
1	Scope	This clause describes the scope of the ISO standard. It states that organizations need to meet statutory and regulatory requirements, which are being improved. All the requirements are generic and applicable to all organizations, regardless of type, size, and product/service provided.
2	Normative references	This clause presents referenced documents are indispensable for this document.
3	Terms and definitions	This clause includes terms, definitions, and abbreviated terms.
4-10	General requirement and recommendations	These clauses provide requirements and recommendations for the safety-life cycle.
11-14	Supporting information	These clauses include additional information, such as examples and tools.

Furthermore, ISO 14971 [4] standard is used as an input for the proposed solution. ISO 14971 is dedicated to the application of risk management to medical devices. This standard establishes the requirements for risk management to determine the safety of a medical device by the manufacturer during the product life cycle. Such activity is required by higher-level regulation and other quality management system standards. The ISO 14971 establishes a framework for risk analysis, evaluation, control, and management, and also specifies a procedure for review and monitoring during production and post-production.

1.3.2. *Conceptual modeling*

Conceptual modeling is a technique that makes a domain-specific idea comprehensible by using models. Stachowiak [5] described the three fundamental features that make a model. First, the model always has a source, and if there is no source of the model, there is no model. The source can be representations of natural or artificial things or can be another model. Second, the model covers less information than the source of the model. Third, the model has some specific purpose and is dedicated to someone, a human, or an artificial model user.

Conceptual models including the underlying concepts and relations are typically presented in notations such as UML [6] or EMF [7]. The notation brings beneficial in communication and visualization to modeling a more extensive system.

The formalization of the ISO standards is a big topic. For example, the formalization of the ISO 26262 is not new, and there are several publications on different aspects of it. The structural domain model was explored by M.Adedjouma [8], the snowball approach (extraction of the conceptual models from the ISO 26262) [9] was introduced by Y. Luo, and the Holistic domain model [10] was provided by A. Khabbaz Saberi. In this project, the conceptual modeling method is based on the Holistic domain model and inspired by the KISS method [11]. The next section explains the Holistic domain model and the proposed process for conceptual modeling.

1.3.3. Holistic Domain model and the proposed process

The main purpose of the Holistic domain model [10] is to provide a consistent and systematic method for the ISO standard modeling from the natural language to the domain model, which can be seen in the graphical notation. The method consists of five-step, namely *define scope*, *analyze text*, *make the initial model*, the *engineer model*, and *make specifications*. The modeling method of the Holistic domain model is given in Figure 1.

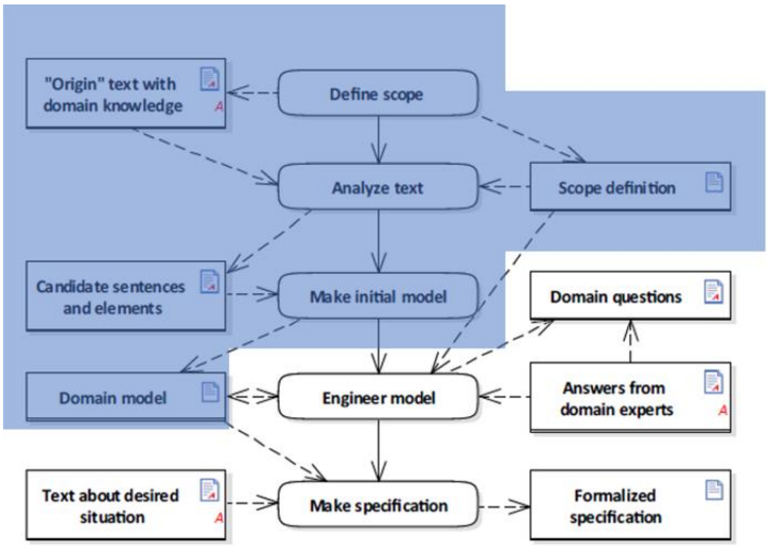


Figure 1. The modeling method of the Holistic domain model [10]. The highlighted blue region shows the scope of this project.

For the scope of this project, we mainly focused on the first three steps of the method. After the third step, the end-user gets the proposed domain model with the chosen graphical notation.

We are introducing the following process for the conceptual modeling of the ISO standard and norm. The *define scope* step is renamed Preprocess; the *analyze text* step is divided into two different steps namely, Parsing and Analysis; and *Make the initial model* is shaped Initial modeling. The overview of the proposed process is given in Figure 2, and an example is shown in Table 2.

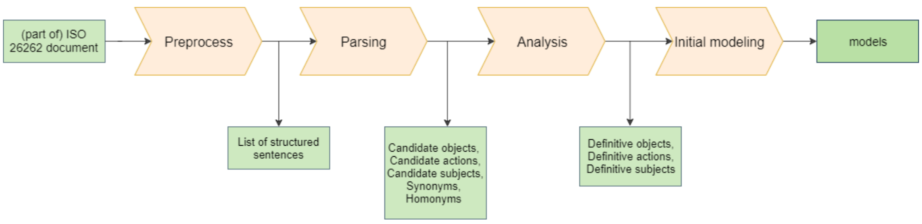


Figure 2. The overview of the proposed process for the conceptual modeling

The Holistic Domain Model [10] uses some of the KISS method [11] concepts. We also followed the KISS method concepts. The definition of the KISS method concepts can be found in Table 17 of Appendix A. In the example, we are using several KISS concepts, namely *original text, structured sentence, candidate object, candidate subject, candidate action, definitive object, definitive subject, definitive action, model object, model subject,* and *model action*.

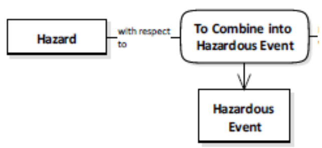
Preprocess: The *original text* comes from the ISO standard and the norm. The end-user provides an appropriate interpretation of the *original text* and builds a *structured sentence*. The interpretation process highly depends on the content of the *original text* and perspective of the end-user. A single *original text* can slip into a list of *structured sentences* or a single *structured sentence*. This step is an essential step to remove ambiguity and inconsistency of the requirements and recommendations of the ISO standard or the norm.

Parsing: In this step, the end-user classifies the *candidate subject*, *object*, and *subjects* from the *structured sentence*. The *structured sentence* is not restricted by the *candidate subject,* the *candidate action*, and the *candidate objects*. It can include *synonyms* as well as *homonyms*. In the next step, the *candidate subject*, the *candidate action*, and the *candidate objects* become a *definitive subject, action*, and *object,* respectively.

Analysis: In this step, the synonym and homonym are needed to merge with the *definitive subject*, *definitive action*, and *definitive object*. The end-user provides a definition for the *definitive subject*, *action*, and *subject*. For instance, Part 1 (Vocabulary) of the ISO 26262 standard introduces all of the terms that are used for this standard. The definition of these terms needs to be filled by the end-user. It is even possible to add or change the definition regarding the end-user's perspective. The *definitive object*, s*ubject*, and *action* are considered to the *object*, *subject*, and *action model* in the Initial modeling step.

Initial Modeling: In this step, the end-user gives UML [6] graphical notation *to the object, subject,* and *action model*. The *object* and *subject model* can be a UML class, UML abstract class, or UML class attribute. The *action model* can be a UML relationship, and it requires input and output models.

Table 2. An example of the proposed process overview

Step	Input	Output
Preprocess	**Original text** [hazardous event: combination of a hazard]	**Structured Sentences** [Hazardous event is a combination. Hazardous event has hazard.]
Parsing	**Structured Sentence** [Hazardous event is a combination. Hazardous event has hazard.]	**Candidate Objects** [Hazardous event, combination] Candidate Actions [is, has] Candidate Subject [hazard]
Analysis	**Candidate Objects** [Hazardous event, combination] Candidate Actions [is, has] Candidate Subject [hazard]	**Definitive Objects** [Hazardous event, combination] Definitive Actions [is, has] Definitive Subject [hazard]
Initial modeling	**Definitive Objects** [Hazardous event, combination] Definitive Actions [is, has] Definitive Subject [hazard]	**Models** []

However, this NLP task study is valuable for different projects in the TNO IVS department. Notably, the NERF task is a useful outcome for TNO stakeholders, and it motivated this section.

1.3.4. Natural Language Tasks study and the result it

At the beginning of this project, several Natural Language Processing (NLP) tasks such as Part-Of-Speech Tagging (POST), Stemming, and Named Entity Recognition (NER) are studied, and an experiment is conducted about the POST task. The purpose of this study and experiment is to support the proposed solution by reducing the modeling time.

The scope of this project's NLP tasks could not help to speed up the proposed process. There are two main reasons; the first reason is the lack of data the NER, and the second reason is the content of the ISO standard, and the norms required a significant degree of interpretation from the end-user during the modeling process.

However, this NLP task study is valuable for different projects in the TNO IVS department. Notably, the concept of the NERF task is a useful outcome for TNO stakeholders. The detailed information of study, experiment result, decisions of NLP tasks are found in Appendix D.

1.4 Outline

The problem statement is given in Chapter 2.

The detailed stakeholder analysis is given in Chapter 3.

The functional and non-functional requirements are described in Chapter 4.

The architecture decisions and their associated rationales are given in Chapter 5.

The full design is revealed in Chapters 6.

The implementation details are given in Chapter 7.

The deployment phase of the project is described in Chapter 8.

The process of validation of the tool is discussed in Chapter 9.

The project results and future work are addressed in Chapter 10.

Project management progress is described in Chapter 11.

2. Problem Analysis

This chapter describes problem analysis by presenting essential aspects needed for the definition of a problem statement. The chapter includes the problem statement and project goals.

2.1 Problem statement

The TNO IVS department always has a passion for discovering innovating solutions and is working on many in-house projects. Currently, the department is enhancing a tool for compliance assurance.

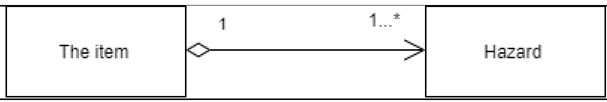
TNO IVS department tooling

The TNO IVS is developing model-based tooling, which is called Simplexity that ensures the quality of the department product meets the requirements of the ISO standard. This tool detects design errors in the early stage of the system development process using the product and process model [9], which comes from the ISO standard requirement. The crucial part of the tool is populating the product and process model using conceptual modeling.

The conceptual modeling is manual and labor-intensive work in the department. The end-user executes the modeling process (see Figure 5) in two applications, namely MS Word and Enterprise Architect (EA). First, the end-user starts the modeling process on the MS Word application. Then, he or she makes several tables for the KISS Method [11] concepts. After that, the end-user completes the tables in using the ISO standard requirements. Finally, the end-user creates models on the EA application using UML diagrams based on the information in the tables filled in the previous step.

In more detail, Table 3 shows an example of the manual modeling process. In this example, the end-user creates tables for the KISS method [11] concepts, namely the *original text* and the *structured sentence*. Then, the end-user fills the tables in using the ISO standard requirements. Lastly, the end-user illustrates models on the EA application using UML diagrams based on the table's information.

Table 3. An example of the manual modeling process.

Original text	*The hazards of the item shall be determined systematically*
	⇩
Structured sentence	*The hazards of the item are determined.*
	⇩
Modeling	

The conceptual modeling of the ISO standards is a complicated task due to a large amount of manual work. The manual modeling process highly depends on the end-user's carefulness. When the user loses his/her attention, it causes several drawbacks in the resulting models. The user can easily skip or make mistakes during the modeling process because of this less-structured and low-controlled manual work. As in result, inconsistent models can be produced. Therefore, a systematic modeling process is needed to support the end-user to increase the consistency of the resulting models. Figure 3 shows examples of consistent and inconsistent models which are derived from the ISO 26262 requirement.

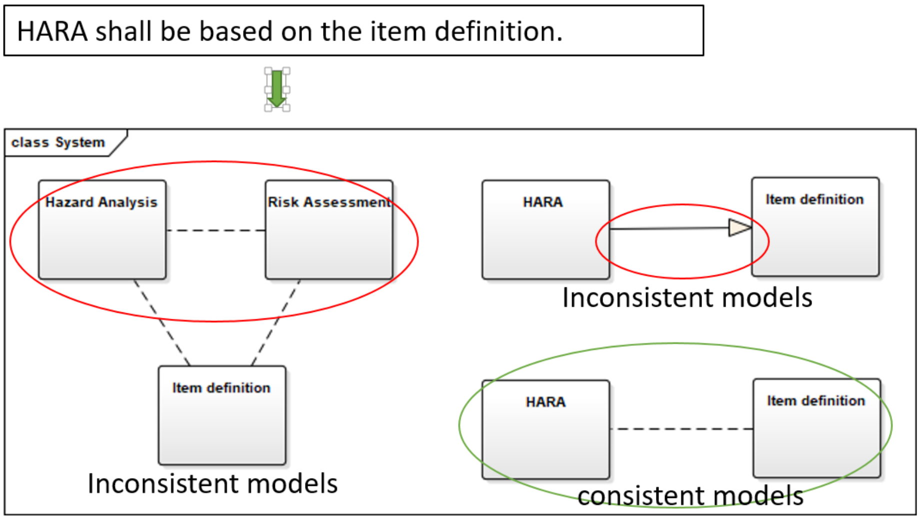


Figure 3. Examples of consistent and inconsistent models.

The manual modeling process has several steps to produce the resulting models. The main source of the input, the ISO standards, includes a large number of requirement specifications. The end-user needs to be aware of that which requirement specifications are associated with the resulting models. In each step, every requirement specifications should be recorded for the further traceability. Currently, the user manually maintains the association between the requirement specifications and the resulting models to trace the manual modeling process. This is a time-consuming task for the end-user. The manual traceability is not always sufficient for further usage. Hence, the traceability can be automated. Figure 4 shows the traceability of the modeling process with derived the ISO 26262 requirement.

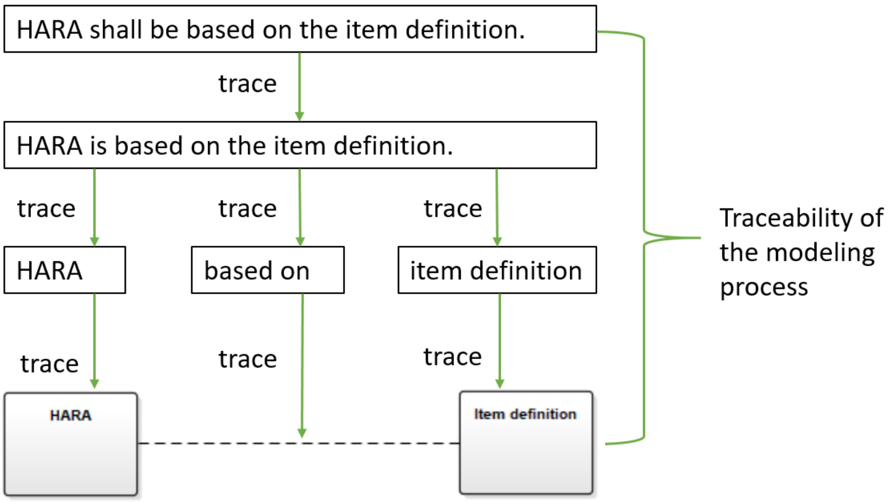


Figure 4. An example of the traceability of the modeling process

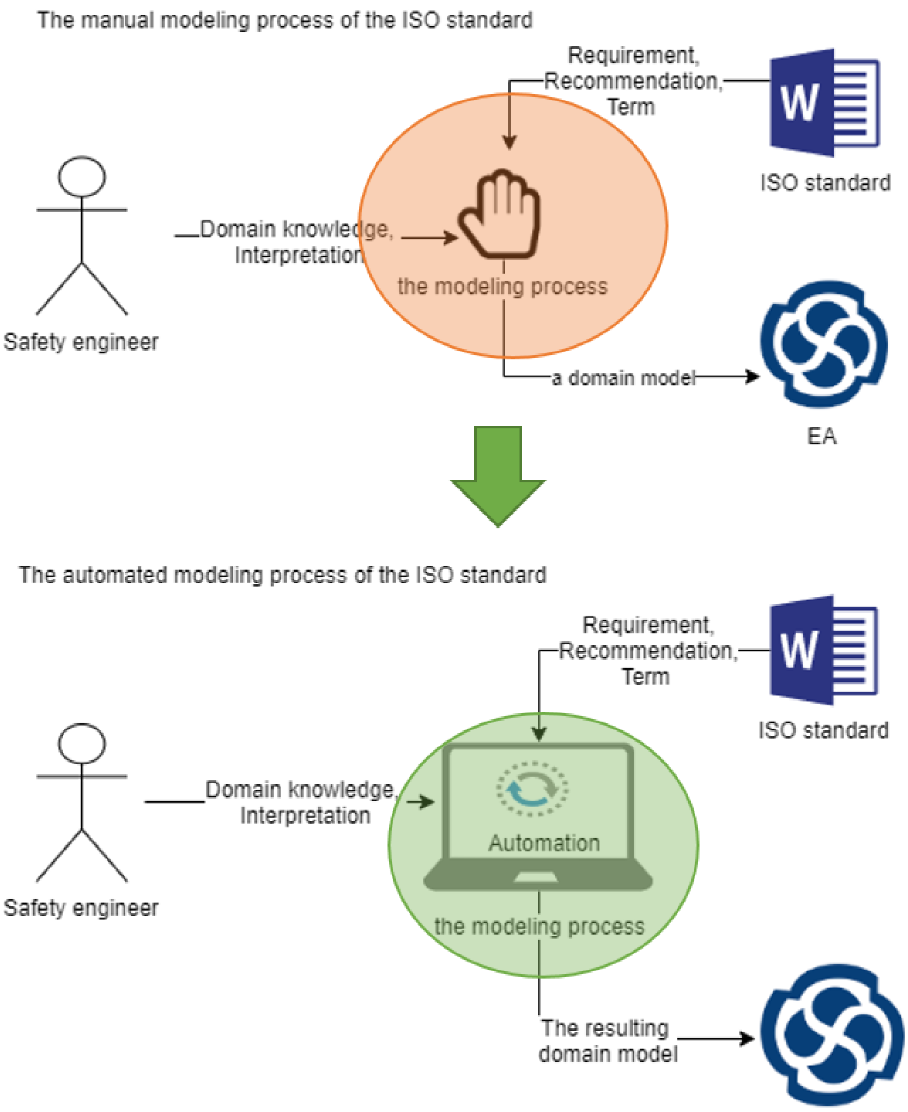


Figure 5. The manual modeling process and the automated solution

2.2 Project goal

Based on the problems stated in the previous section, the project ultimately aims to reduce the end-user's manual work by supporting an automated tool. The primary objective is to design and develop the automated tool that provides a systematic process for conceptual modeling to increase the consistency of the resulting models and the traceability of the modeling process.

Further, the tool can be applied to different domains by separating the conceptual modeling from identifying domain concepts. The separation allows the tool to accept various ISO standards, and it is named Multi-domain Modeling Assistance (MMA) tool.

3. Stakeholder Analysis

This chapter presents the stakeholders of the project, their interests, and their goals. The TNO IVS department and the Eindhoven University of Technology are the involved parties, and each of them has specific interests towards the project. The following sections introduce the stakeholders and present each party in detail.

3.1 TNO IVS department

TNO IVS is considered as the owner and initiator of this project. The TNO IVS provides the source of knowledge, requirements, and expectations of this work. Table 4 describes the stakeholder interest of the TNO IVS department.

Table 4. IVS department stakeholder overview

Stakeholder	**Company Supervisor**
Role	By project Owner: • Manage the project progress and ensure defined project goals are achieved on time within the budget • Evaluate the project progress and provide feedback By project mentor: • Breakdown project goals to achievable tasks • Mentor the daily progress of the project and assist the trainee in the project related conceptual tasks
Concern	By project owner: • achieve the project goals
Representative	Arash Khabbaz Saberi
Responsibilities	• Mentor, evaluate, and provide regular feedback on project progress and the project's final deliverables • Provide the domain knowledge, references, other related project materials, and the ISO standard • Support boarding process at the company, including apply for employee ID card and request a new TNO account. • Evaluate and assess the project progress and project deliverables in terms of defined criteria. • Review the final report
Acceptance Criteria	• A timely report of the project • Project deliverables
Involvement	During the project timeline, continuous communication by weekly or ad-hoc based meetings with the trainee and monthly project steering meeting, including TU/e supervisors.
Stakeholder	**Safety engineer and the potential developer**
Role	Safety engineer: • Provide a solid knowledge of ISO 26262 • Share the insights regarding tool usage experience Potential developer:

	• Provide technical support to deploy the tool to TNO built environment
Concern	By safety engineer: • Speed up the modeling process By potential developer: • Receive the technical report
Representatives	Safety engineer: Arash Khabbaz Saberi Potential developer: Dennis van den Brand
Responsibilities	• Provide feedback on tool features • Provide supports to get access to TNO internal server • Provide necessary information about Simplexity tool
Acceptance Criteria	N/A
Involvement	• During the project timeline, continuous communication via emails or meetings based on project needs.

3.2 Eindhoven University of Technology (TU/e)

As an education program, the PDEng in software technology program is conducted by TU/e. This project is considered as the final project for the PDEng trainee. The TU/e dictates certain standards that have to be met. Those standards are related to the design of process, project management, and project implementation. Besides, TU/e is one party of the project. Its main focus is on project execution, and some stakeholders are identified from the TU/e. Table 5 shows the TU/e stakeholder overview.

Table 5. TU/e stakeholder overview

Stakeholder	**University supervisors**
Role	The university supervisor guards the educational interests of the university and the trainee.
Representatives	Mark van den Brand and Serguei Roubtsov
Concern	To achieve the project goals in the project timeframe
Responsibilities	• Monitor, evaluate, and provide regular feedback on project progress and project final deliverables • Provide the related knowledge during the project's progress • Help to identify high risks of the project • Evaluate the project progress and project deliverables in terms of qualities of the design • Review the final report
Acceptance Criteria	• A timely report of the project deliverables • Project design, implementation, and documentation meet PDEng level
Involvement	During the project timeline, continuous communication by weekly or ad-hoc based meetings with the trainee and monthly project steering meeting, including the Company Supervisor.
Stakeholder	**PDEng trainee**
Role	Software designer

Concern	Delivery the project deliverables with required quality on time
Representative	Munkhdalai Sainbileg
Responsibility	• Handle the whole project progress at all levels • Design, implement, and deploy a modeling tool for ISO standards • Apply soft and technical skills for the project • Deliver the project deliverables on time
Acceptance Criteria	• A timely report of the project deliverables • Project design and implementation have to meet PDEng level
Involvement	Full involvement

4. Tool Requirements

This chapter introduces the requirement analysis phase of this project by presenting the requirement gathering process.

4.1 Requirement Gathering

The requirement gathering process consists of interviewing stakeholders and prototyping the software tool.

4.1.1. Interviewing the stakeholders

During the stakeholder analysis phase, two potential users (safety engineer and potential developer) are identified from the TNO IVS department. The end-user of the tool is a safety engineer who is considered the main resource of knowledge. We conducted many interviews with the end-user and ended up with five high-level requirements (see Table 6), based on project goals. The definition of the term in italics can be found in Table 18 of Appendix B.

Table 6. High-level functional requirements

Id	High-level requirements
1.1	The tool shall provide a systematic process for conceptual modeling.
1.2	The tool shall be compatible with the MS Word application by exchanging information between a MS Word document containing the ISO standards and norms and the tool.
1.3	The tool shall export the resulting models to the Simplexity application.
1.4	The tool shall provide a particular Graphic User Interface (GUI) for the conceptual modeling based on the proposed process.
2.1	The tool shall keep all the records of the conceptual modeling step by step.

The high-level requirements are divided into detailed functional requirements by defining the specification. The high-level requirement decomposition is based on the proposed process that is introduced in Section 1.3.3. In total, 24 detailed functional requirement specifications are identified in this project, which is given in Table 7.

We used the MoSCoW method [14] for requirement prioritization and the method divides the requirements into four categories:

Must Have: This category indicates an essential requirement that has to be fulfilled.
Should have: This category presents a requirement that needs to be carefully considered and the valid reason it could be ignored.
Could have: This category shows a requirement that is not mandatory, but it would be preferable satisfied if it fits the project timeframe.
Won't have: This category describes a requirement that is not planned in the project timeframe.

To give appropriate priority for each requirement, we defined criteria that are considerable and measurable. The criteria were **Value**, **Risk**, and **Difficulty**. The value criterion indicates the relative importance of the requirement for the stakeholder. The risk shows the potential cause that might damage this project. The difficulty indicates the complexity from the point of view of the implementation. The full requirement justifications are given in Table 19 of Appendix C.

Table 7. Detailed functional requirements

Related High-level REQ ID	REQ ID	Requirement specification	The modeling step	Priority
1.2	1.2.0	The tool shall be able to show its Graphic User Interface (GUI) in the MS Word application via a shared add-in.	N/A	Must
	1.2.1	The tool shall be able to highlight the *processed sentence* in the MS Word document automatically.	Preprocess	Should
	1.2.2	The tool shall be able to select the next sentence from the MS Word document and take it to the *original text.*	Preprocess	Must
	1.2.3	The tool shall be able to copy the selected sentence from the MS Word document to the *original text* automatically.	Preprocess	Should
1.4	1.4.1	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step	Must
	1.4.2	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence.*	Preprocess, Parsing, and Analysis	Should
	1.4.3	The tool shall be able to accept multiple *original texts*.	Preprocess	Must
	1.4.4	The tool shall be able to accept multiple *structured sentences* for a single *original text.*	Preprocess	Must
2.1	2.1.1	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step	Could
	2.1.2	The tool shall be able to generate *Log ID* based on the *metadata.*		Must
	2.1.3	The tool shall be able to show the *Log ID.*	All the modeling step	Should
1.1	1.1.1	The proposed process shall have four steps: Preprocess, Parsing, Analysis, and Initial modeling.		Must
	1.1.2	The Preprocess step might have a sub-step.	Preprocess	Should
	1.1.3	The tool shall be able to offer the start function of the modeling process.		Must
	1.1.4	The Safety engineer should be able to give *metadata: artifact name* and *section Id* for the *Log ID.*		Should
	1.1.8	The tool shall be able to show a list of the *structured sentences* with their *status.*	Preprocess	Must
	1.1.10	The tool shall be able to extract *candidate objects*, *candidate subjects*, and *candidate actions* from the	Parsing	Must

		structured sentence on the tool based on the safety engineer input.		
	1.1.11	The tool shall be able to show the *candidate object*, *candidate subject*, and *candidate action* in the *structured sentence*.	Parsing	Must
	1.1.12	The tool shall be able to convert the *candidate object*, *candidate subject*, and *candidate action* to *definitive object*, *definitive subject*, and *definitive action* respectively.	Analysis	Must
	1.1.14	The safety engineer shall able to add the definition for a *definitive object*, *definitive subject*, and *definitive action*.	Analysis	Must
1.3	1.3.1	The tool shall be able to assign a type (*Class, Abstract Class,* and *Attribute*) for the *object* and *subject model* based on the end-user selection.	Initial modeling	Should
	1.3.2	The tool shall be able to assign a type (*Associate, Generalize,* and *Compose*) for the *action model* based on the end-user selection	Initial modeling	Should
	1.3.4	The tool shall be able to show *the start point* and *the endpoint to the action model.*	Initial modeling	Should
	1.3.5	The tool shall be able to export the *initial models* to *Simplexity*.	Initial modeling	Must

Furthermore, we identified some non-functional requirements in this project. Table 8 describes the non-functional requirements with specifications and rationale. The primary justification is *Importance,* which is based on the specification or rationale of each non-functional requirement. We focused on *vital* non-functional requirements. *Important* non-functional requirements are considered as criteria for the architecture pattern of the tool, and *not-important* non-functional requirements are ignored.

Table 8. Non-functional requirements with specification and rationale

REQ ID	Non-functional requirement	Specification (S) or rationale (R)	Importance
3	Extensibility	S: The tool shall be extensible regarding adding a new feature or changing an existing feature.	Important
4	Performance	R: Only a few end-users work on the MMA tool, and the performance is not the first focus in non-functional requirements.	Not important
5	Scalability	R: Limited number of users works on the tool, and the tool scalability is not important.	Not important
6	Security	R: The tool works only for the TNO internal network, and the network is secured.	Not important
7	Usability	S: The tool shall be easy to use it by providing GUI.	Vital
8	Deployment	S: The tool shall be deployed easily on different virtual or physical machines.	Vital

9	Maintainability	S: The tool shall be decomposed to a suitable number of components.	Important

4.1.2. Prototyping the proposed software tool

The second activity of the requirement gathering process is to illustrate mock-ups. The mock-ups allow visualizing aspects of the tool that can be seen and shared before it is built. The mock-ups lead to fewer changes later and hence reduce overall costs considerably. It also prevents a misunderstanding of the detailed functional requirement specifications in this project. We illustrate seven different mock-ups that are seen in Figures 5-10. We agreed on the Graphic User Interface (GUI) of the tool with the end-user based on the mock-ups. In the next subsections, the mock-ups are explained based on the proposed process that is described in Section 1.3.3.

Welcome view in the mock-up

This view (see Figure 6) is the first mock-up of the MMA tool, and the main purpose of this view is to presents the initial point of the proposed process. The safety engineer can start the proposed process by the *Start Modeling* feature. This view also gives general information about the MMA tool.

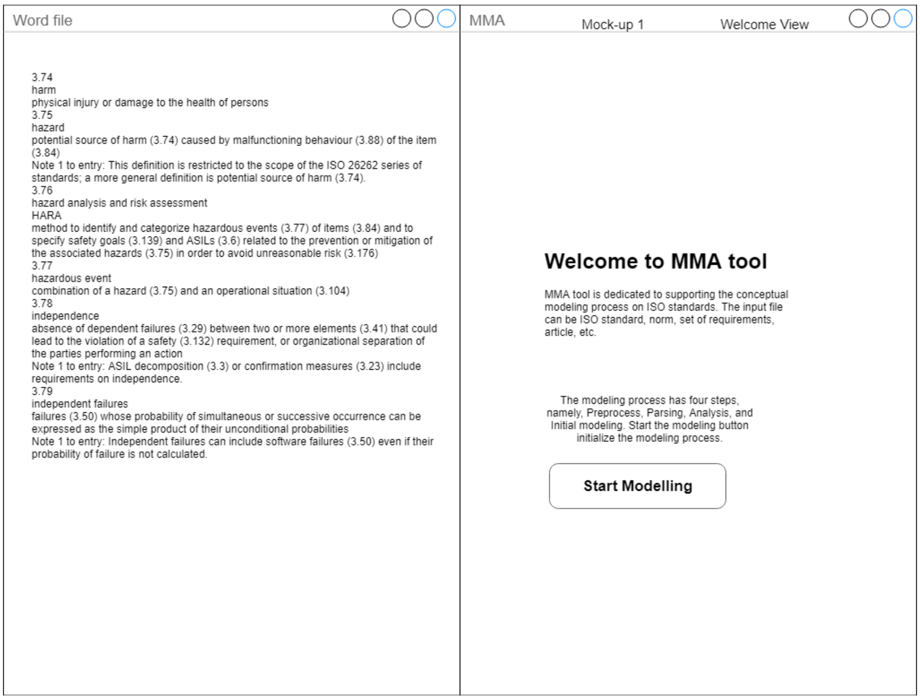


Figure 6. Welcome view in the mock-up

Preprocess view in the mock-up

This view (see Figure 7) is to illustrate the Preprocess step of the proposed process regarding the tool features and derived requirement specifications. The main functionality of this step is an interpretation process from the *original text* to the *structured sentences*. This modeling step is related to the following requirement specifications, and the requirement specifications led to certain features on the MMA tool. Table 9 shows the features of this view with derived requirement specifications.

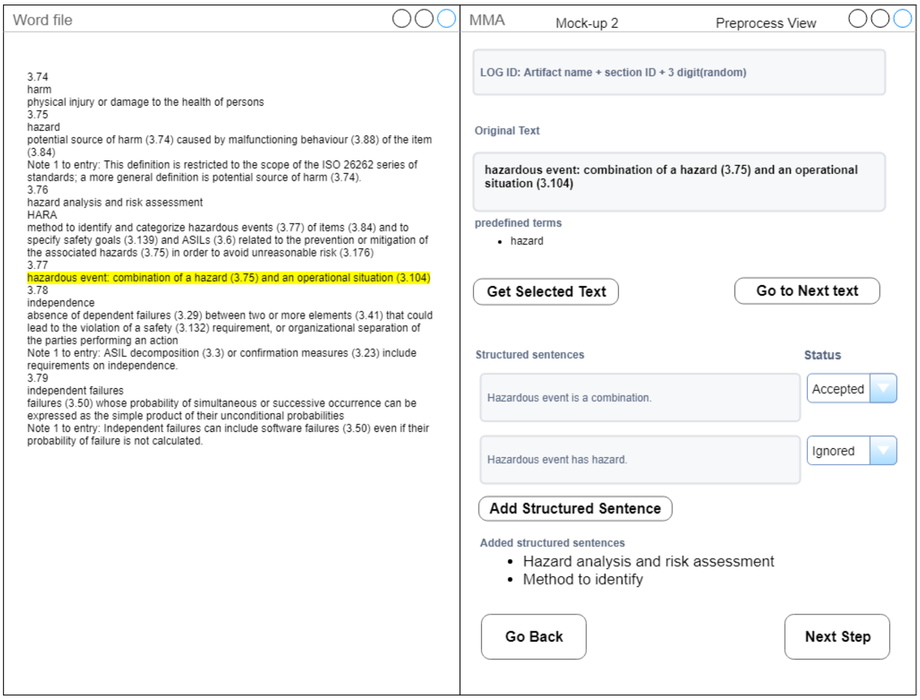


Figure 7. Preprocess view in the mock-up

Table 9. Features of the preprocess view

REQ ID	Features of the view	Derived requirement specification	The modeling step
1.2.1	*Go To Next text*	The tool shall be able to highlight the *processed sentence* in the MS Word document automatically.	Preprocess
1.2.2	*Go To Next text*	The tool shall be able to select the next sentence from the MS Word document and take it to the *original text.*	Preprocess
1.2.3	*Get selected sentence*	The tool shall be able to copy the selected sentence from the MS Word document to the *original text* automatically.	Preprocess
1.4.1	*Preprocess view*	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step
1.4.2	*Go To Next text* and *Get selected sentence*	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence*	Preprocess
1.4.3	*Go To Next text*	The tool shall be able to accept multiple *original texts*.	Preprocess
1.4.4	*Add Structured Sentence*	The tool shall be able to accept multiple *structured sentences* for a single *original text*	Preprocess
2.1.1	*Next step*	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step

2.1.3	*LOG ID*	The tool shall be able to show the *Log ID*	All the modeling step

Overview of the preprocess view in the mock-up

This view (see Figure 8) is an extra mock-up of the Preprocess step of the proposed process. It shows a list of the *original text*, which includes related *structured sentences* that are processed in the Preprocess step. Table 10 captures features of the view with the derived requirement specifications.

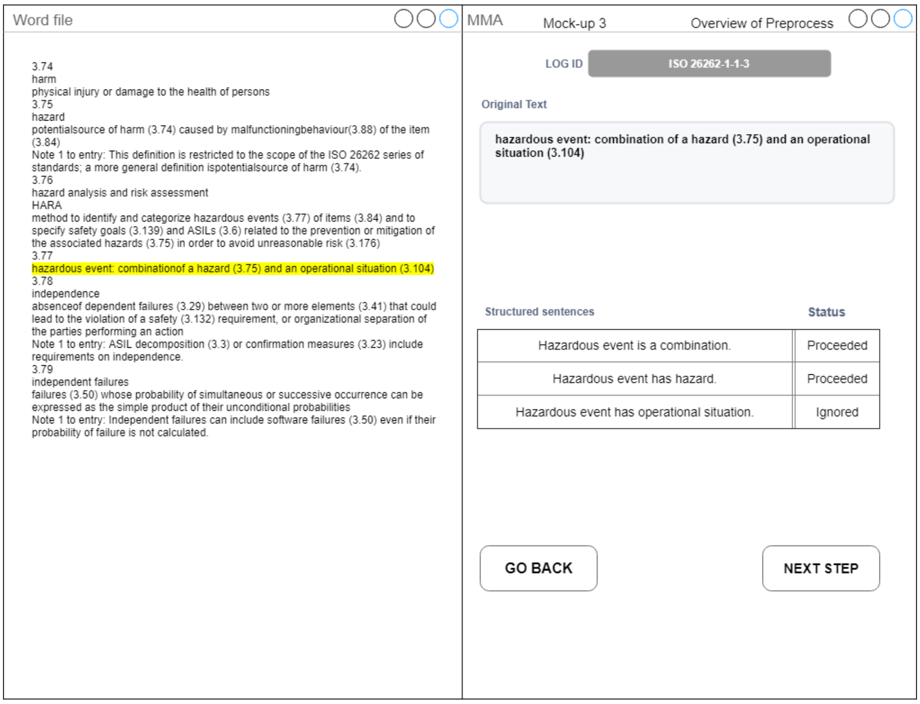


Figure 8. Overview of the preprocess view in the mock-up

Table 10. Features of the overview of the preprocess view

REQ ID	Feature of the view	Derived requirement specification	The modeling step
1.1.2	*Clicking on of the original text*	The Preprocess step might have a sub-step.	Preprocess
1.4.1	*Overview of Preprocess view*	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step
2.1.1	*Next step*	The tool shall be able to save the KISS method concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step
2.1.3	*LOG ID*	The tool shall be able to show the *Log ID.*	All the modeling step

Parsing view in the mock-up

The parsing view (see Figure 9) is to illustrate the Parsing step of the proposed process in terms of the tool features and requirement specifications. The main purpose of this step is to extract the *structure sentence* into the KISS modeling concepts*: candidate object*, *action*, and *subject*. The list of *structured sentences* came from the Preprocess step, and the safety engineer identifies a word or a combination of words as the *candidate object*, *action*, and *subject*. Then the end-user selects a type (object, action, and subject) and ID of the *structured sentence* for the *candidate object*, *action*, and *subject*. Table 11 presents the features of this view with the derived requirement specifications.

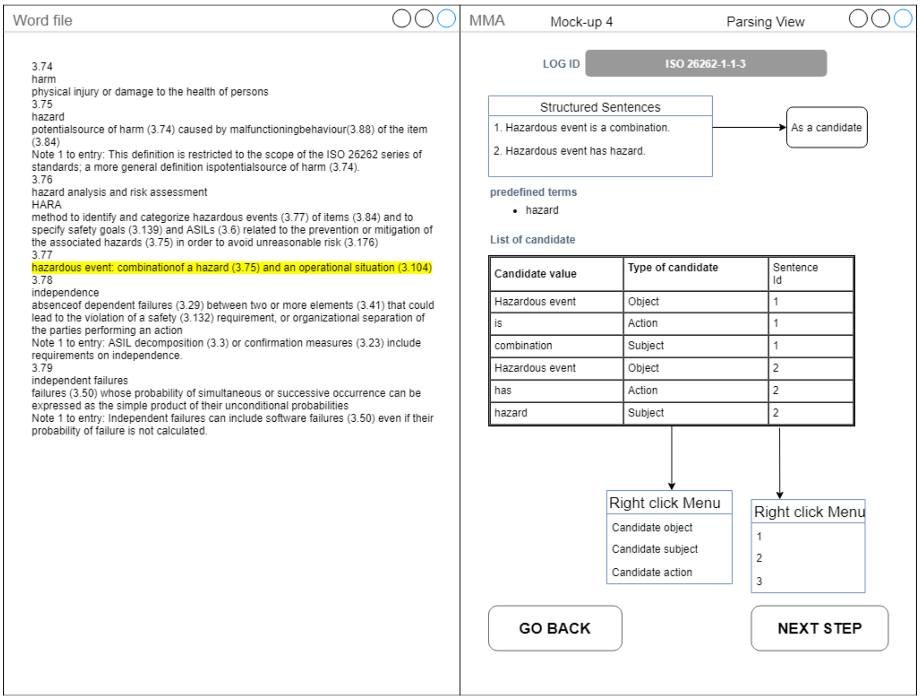


Figure 9. Parsing view in mock-up

Table 11. Features of the parsing view

REQ ID	Feature of the view	Derived requirement specification	The modeling step
1.1.10	*As a candidate*	The tool shall be able to extract *candidate objects*, *candidate subjects*, and *candidate actions* from the *structured sentence* on the tool based on the safety engineer selection.	Parsing
1.1.11	*List of candidate*	The tool shall be able to show *candidate object*, *candidate subject*, and *candidate action* in the *structured sentence*	Parsing
1.4.1	*Parsing view*	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step
1.4.2	*Predefined term*	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence*	Preprocess, Parsing, and Analysis
2.1.1	*Next step*	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step

2.1.3	*LOG ID*	The tool shall be able to show the *Log ID*	All the modeling step

Analysis view in the mock-up

The main purpose of this view (see Figure 10) is to illustrate the Analysis step of the proposed process regarding the tool features and requirement specifications. The core functionality of this step is to give definitions for the *definitive object*, *action*, and *subject*. Table 12 presents the features of this view with the derived requirement specification.

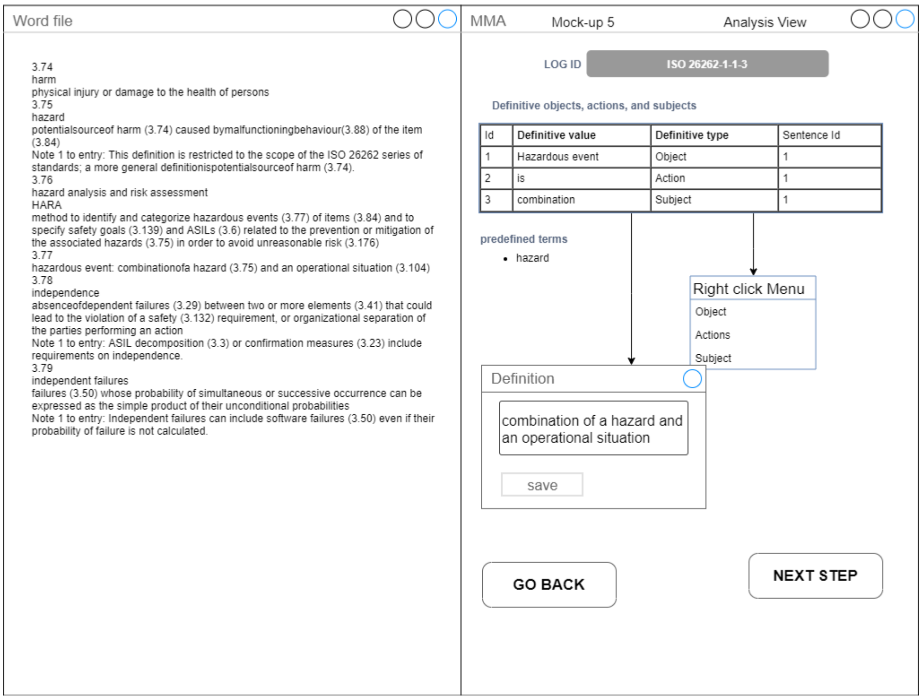


Figure 10. Analysis view in mock-up

Table 12. Features of the analysis view

REQ ID	Feature of the view	Derived requirement specification	The modeling step
1.1.12	*Definitive objects, actions, and subjects*	The tool shall be able to convert the *candidate object, candidate subject*, and *candidate action* to *definitive object, definitive subject*, and *definitive action* respectively	Analysis
1.1.14	*Save*	The safety engineer shall able to add the definition for a *definitive object, definitive subject*, and *definitive action*	Analysis
1.4.1	*Analysis view*	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step
1.4.2	*Predefined term*	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence*	Preprocess, Parsing, and Analysis
2.1.1	*Next step*	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step

2.1.3	*LOG ID*	The tool shall be able to show the *Log ID*	All the modeling step

Initial Modeling view in the mock-up

The purpose of this view (see Figure 11) is to illustrate the Initial modeling step of the proposed process in terms of requirement specifications and features of the MMA tool. The main feature of this step to give UML notation for the *model object*, *subject*, and *action*. Table 13 shows the features of this view with the derived requirement specifications.

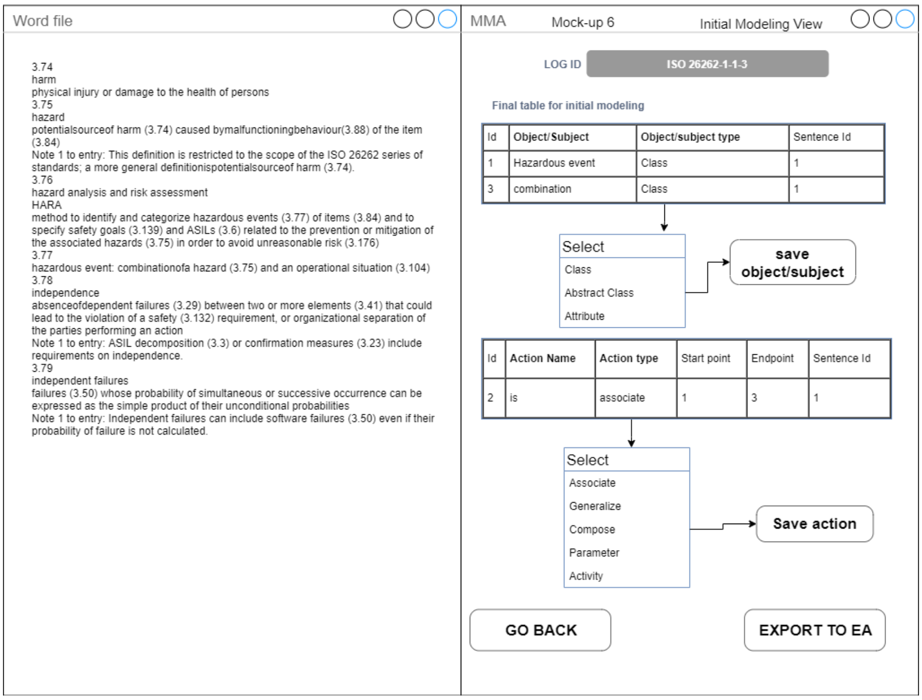


Figure 11. Initial Modeling view in the mock-up

Table 13. Features of the initial modeling view

REQ ID	Feature of the view	Derived requirement specification	The modeling step
1.3.1	*Save object/subject*	The tool shall be able to assign a type (*Class, Abstract Class*, and *Attribute*) for the *object* and *subject model* based on the end-user selection	Initial modeling
1.3.2	*Save action*	The tool shall be able to assign a type (*Associate, Generalize*, and *Compose*) for the *action model* based on the end-user selection	Initial modeling
1.3.4	*Start point and Endpoint*	The tool shall be able to show *the start point* and *the endpoint* to the action model	Initial modeling
1.3.5	*Export to EA*	The tool shall be able to export the *initial models* to *Simplexity*	Initial modeling
1.4.1	*Initial modeling view*	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step

2.1.3	*LOG ID*	The tool shall be able to show the *Log ID*	All the modeling step

5. Tool Architecture

This chapter presents the architecture of the MMA tool. First, the overview of the MMA tool is given in Section 5.1. The process model is described in Section 5.2. The architecture pattern of the MMA tool is unveiled in Section 5.3, and then the deployment view of the MMA tool is explained in Section 5.4. Sequence diagrams and screens are revealed in Section 5.5. This chapter ends the high-level decision of the MMA tool in Section 5.6.

5.1 Overview of MMA tool

The primary input of the MMA tool is the ISO standard, and the output is the domain model. An overview of the MMA tool is shown in Figure 12.

As shown in Figure 12, first, requirements, recommendations, and terms are collected from the ISO standard as the tool input. These requirements and recommendations provide guidelines about a particular subject while terms define the meaning of certain concepts. For example, requirements and recommendations of Part 3 (Concept phase) of the ISO 26262, describe the item definition, the initiation of the safety lifecycle, the hazard analysis, risk assessment, and the functional safety concepts. All of these concepts are defined in a separated part (Part 1: Vocabulary) of the ISO standard.

Second, the end-user gives an appropriate interpretation and domain knowledge to remove inconsistency and ambiguity of the requirements and recommendations.

Finally, the end can create the resulting domain model according to the process model on the MMA tool. The process model is explained thoroughly in the next section.

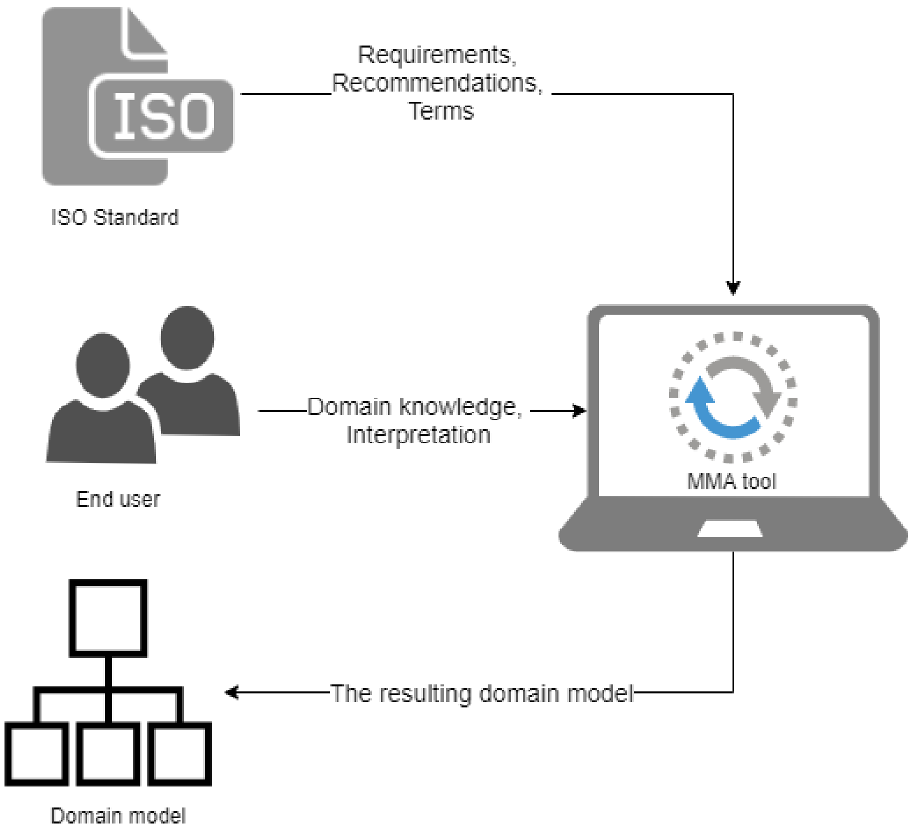


Figure 12. The overview of the MMA tool

5.2 Process model of the MMA tool

The process model, as shown in Figure 12, covers the modeling process from the start to the end based on the proposed process that is explained in Section 1.3.3.

The modeling process is initiated on the MS Word application and finished on the MMA tool. The MMA tool also exports the resulting models to the Enterprise Architect (EA) application via the Simplexity.

In general, the ISO standards are published by PDF file, and the MS Word application could open any pdf file that allows the MMA tool to process it.

The *term checker* (see Figure 13) is an extra feature of the MMA tool. It helps the end-user to identify domain concepts in the early steps of the modeling process. This feature identifies a particular term in the three modeling steps, namely Preprocess, Parsing, and Analysis of the modeling process.

- In the Preprocess step, this feature checks the term from the *original text*. If this feature finds the term, it shows the definition of the term.
- In the Parsing step, this feature identifies the term from the *structured sentence*. If it detects the term, it shows the definition of the term.
- In the Analysis step, this feature provides the definition of the term, which is found from the *definitive object, definitive subject*, and *definitive action.*

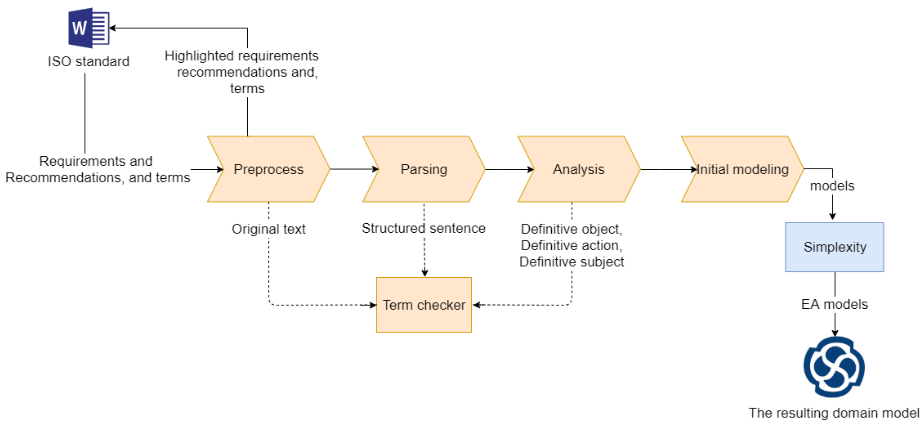


Figure 13. Process model of the MMA tool

5.3 Architecture pattern of the MMA tool

The architecture of the MMA tool is one of considerable important decision because changing the architecture pattern is an expensive task for TNO stakeholders in terms of cost (time and resource). The architecture pattern helps to define the fundamental characteristics and behavior of an application. For instance, some architecture patterns naturally lend themselves toward highly scalable applications, whereas other architecture patterns naturally lend themselves toward highly agile applications. Considering the characteristics and strengths of each architecture pattern are necessary to choose the one that meets this project goal. We considered three architecture patterns, namely microservice, batch sequential, and three-tier architecture pattern.

5.3.1. Microservice

The fundamental concept of this pattern is a distributed architecture. It means all the components within the architecture are entirely decoupled from one other and accessed by some remote access protocol. This concept enables easy deployment by a sufficient and streamlined delivery pipeline, increased scalability of the application. Figure 14 presents the microservice architecture pattern.

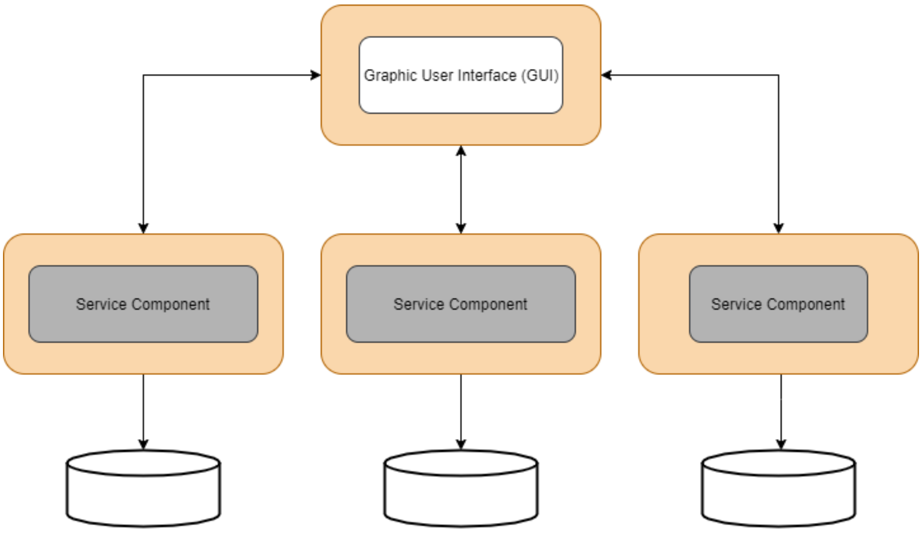


Figure 14. Microservice architecture pattern

5.3.2. *Batch sequential*

In this pattern, the complex system can be divided into several smaller subsystems that can be defined as a series of independent computations. In this pattern, the whole task is slipped into small processing steps, which are implemented as separate, independent components. Each step runs to completion and then calls the next sequential step until the whole task is completed. During each step, a batch of data is processed and sent as a whole to the next step. Figure 15 shows the batch sequential architecture pattern regarding its models.

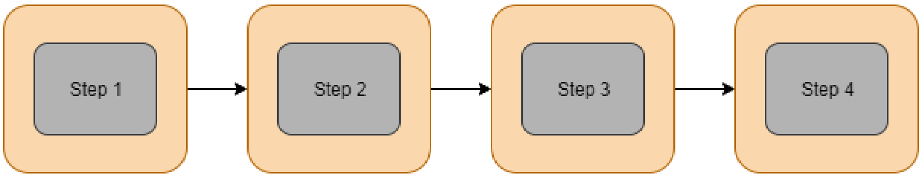


Figure 15. Batch sequential architecture pattern

5.3.3. *Three-tier*

Tier separation (layering) is the most common and well-known technique, and it is used to break apart a complicated or complex software system. The architecture pattern is organized into horizontal tiers, each tier performing a specific role within the application. The key concept of tier (layer) isolation means that changes made in one tier of the architecture generally have a low impact or affect components in other tiers: the change is isolated to the components within that tier and possibly another associated tier.

5.3.4. *The decision of the architecture pattern*

We made a comparison matrix for the candidate architecture pattern, which is given in Table 14. The non-functional requirements, design decision (DD1), and design constraint (DC1) are criteria for the judgments(Good, Normal, Weak). The judgments are evaluated by TNO IVS department developers and the PDEng trainee.

In this project, we found a design decision (**DD1**) and design constraint(**DC1**) that had significant impacts on technology decisions. Both are explained in Section 6.1.
DD1: The MMA tool needs to integrate through an add-in with the MS Word application.
DC1: The MMA tool needs to integrate with the Simplexity application.

Table 14. The comparison matrix of the candidate architecture patterns.

Criterion	**Microservice**	**Batch sequential**	**Three-tier**
DD1	Weak	Normal	**Good**
DC1	Good	Good	**Good**
Extensibility	Good	Weak	**Normal**
Testability	Good	Weak	**Good**
Modularity	Good	Good	**Good**
Reusability	Normal	Good	**Good**
Modifiability	Good	Normal	**Normal**
Analyzability	Normal	Good	**Good**

The **microservice** architecture pattern received an overall good score on the comparison matrix. The weak point was DC1 because the MMA tool shall have individual Graphic User Interface (GUI) that GUI exchanges only a particular way to communicate with MS Word application. The implementation of this pattern could lead to more components, and that was the major drawback of this pattern.

The **batch sequential** architecture pattern was considered a weak pattern on the comparison matrix. There were two cons: the first weak point was extensibility. According to this pattern, the components of the MMA tool highly depend on the process model. It becomes much hard to add a sub-step to the implementation of this pattern. The second point was the testability of the tool. It is complicated to simulate each of the modeling steps during the test.

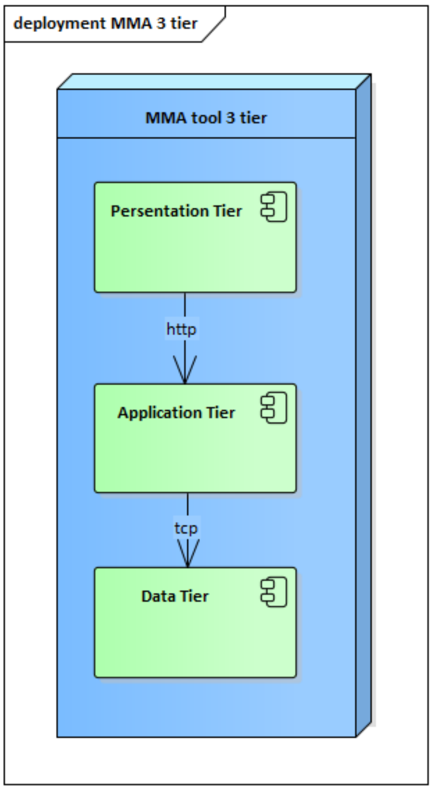


Figure 16. Three-tier architecture pattern of MMA

The **three-tier architecture** pattern was selected for the MMA tool. The high-level architecture pattern of the MMA tool relies upon three-tier concepts. Tier isolations enable more flexible design opportunities. For example, DD1 and DC1 can be separated from the process model at the design level. At the same time, we are able to apply different design patterns in each tier of the MMA tool, which is explained in the next chapter.

In the high-level, we defined three-tiers (see Figure 16), namely Presentation, Application, and Data tier. Each tier can be deployed in a separated virtual or physical machine.

Presentation Tier

This tier is the top level of the MMA tool. The primary function of this tier is to handle the interaction between the end-user and the tool. The primary responsibilities of this tier are to display information to the end-user, interpret the end-user inputs into tasks, and initiate the tasks to communicate with the application tier. The end-user can easily access and execute the modeling process on this tier by a particular GUI.

Application Tier

This tier is the middle level of the MMA tool. The main responsibility of this tier is to provide a communication channel between the presentation tier and the data tier. In this tier, certain fixed points are designed to receive information. The information is transferred by the following order:

- Receive input data from the end-user in fixed endpoints via the presentation tier.
- Parse the input data to fixed data models
- Transfer the data models to the data tier

Data tier

The main responsivity of the data tier is to be easy to access and maintain our data in the desired format. We decided to save the progress of the modeling process on the database management system because it offers a more reliable and straightforward way to handle data.

5.4 *Deployment view of MMA tool*

The deployment view of the MMA tool describes the interaction between the MMA tool and external and internal applications. Figure 17 illustrates the deployment view of MMA tool. Currently, the three-tier of the MMA tool is deployed in the same machine (Internal Server). However, in the future, three-tiers will be deployed on different machines. The build blocks are captured in three different colors. Green blocks denote MMA tiers, blue blocks represent external applications, and an orange block captures the internal application (Simplexity). The external applications describe applications that are developed by the third party. TNO IVS department has developed an application that is the internal application.

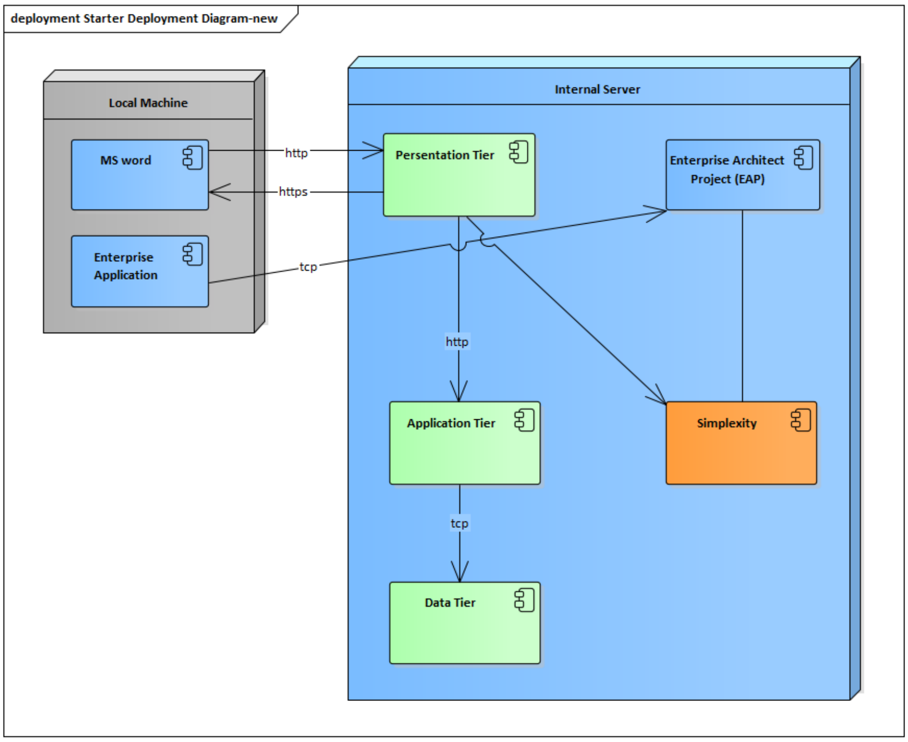


Figure 17. The deployment view of the MMA tool

5.5 Sequence Diagrams and Screens

Screens provide particular GUI for the end-user depends on the process model. While sequence diagrams show the communication flow behand the screens between the end-user (safety engineer) and the MMA tool regarding the three-tiers, the MS Word application, and the Simplexity application. The entire sequence diagram is divided into four sequence diagrams based on the process model.

5.5.1. Sequence diagram and Screen of the Preprocess step

The Preprocess screen (see Figure 18) is the first step of the modeling process. The main task of this screen is to process the part of the source text (in *Original Text*) and take the end-user interpretation (in *Structured Sentence*) for the *original text*. It depends on the complexity of the *original text*, and the end-user can add *multi-structured sentences* to the *original text*. Also, the end-user can process multiple *original texts* on this component screen.

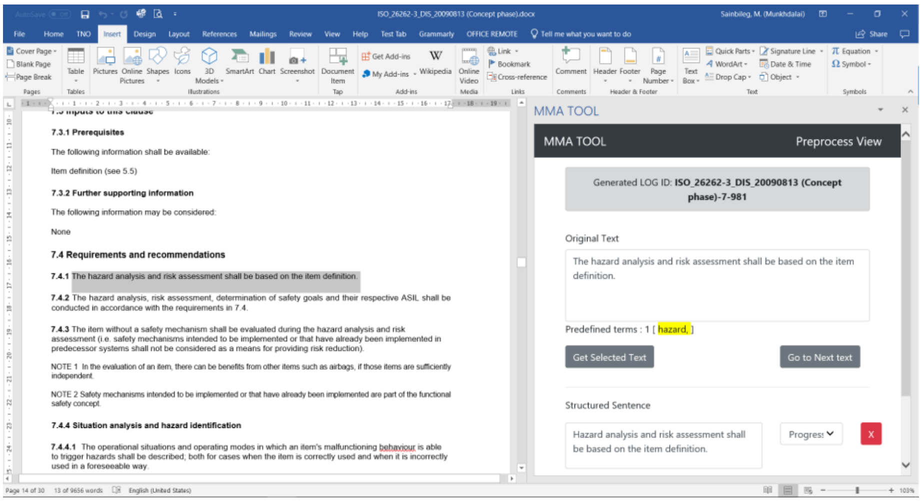


Figure 18. Preprocess screen

Background of this screen, the communication flows (see Figure 19) has the following order in the Preprocess step.

1. The safety engineer selects part of the text from the ISO standard in the MS Word application.
2. The safety engineer gives a command to get the selected text from the ISO standard in the Presentation tier of the MMA.
3. The Presentation tier receives the selected text from the MS Word application to the *original text* as an input.
4. Then, the safety engineer adds a *structured sentence* to the *original text*. Also, it is possible to add a *multi-structured sentence* to the *original text*.
5. After that, the safety engineer remains to add more original text using *Go To Next Text* features. The *Go To Next Text* repeats from step 1 to 4 automatically.
6. The safety engineer goes to the Preprocess Overview page using the *Go Next* feature in the Presentation tier. Background of that operation, the Presentation tier transfers processed *original text*, and *structured sentences* to the Application tier and the Application tier requests the Data tier. After the Data is saved in the database, the Data tier returns a response to the Application tier. Thus, the result is displayed in the Presentation tier based on that response.
7. In the end, the safety engineer moves the Parsing step using the *Go Next* feature.

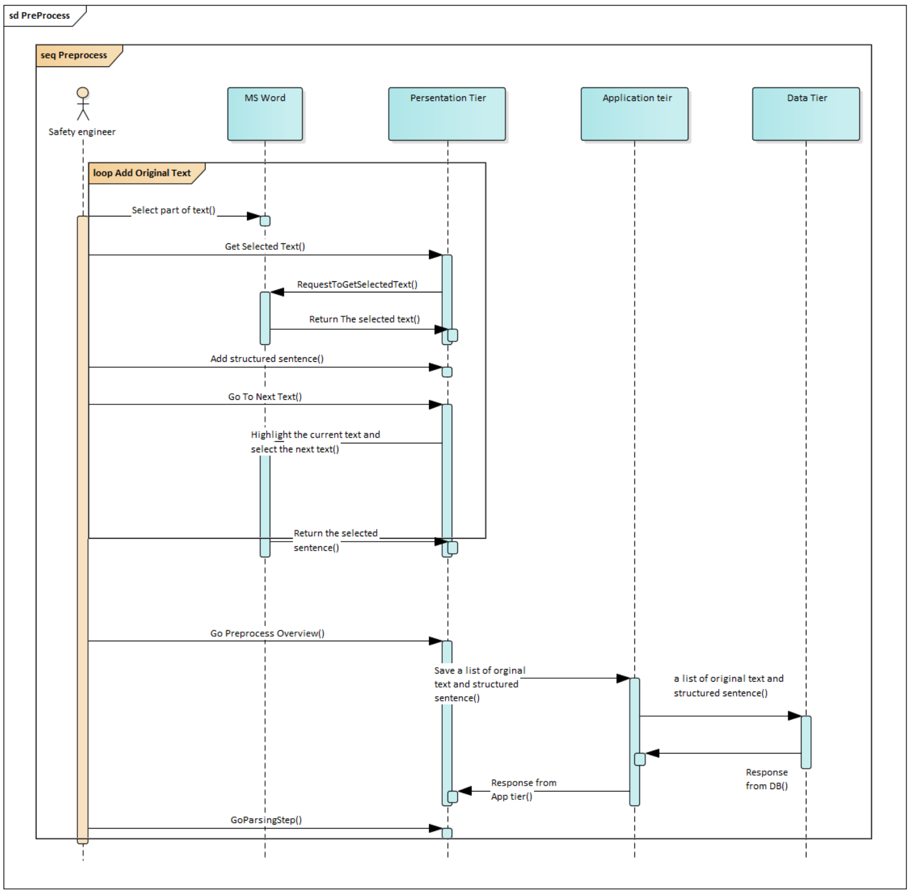


Figure 19. Sequence diagram of the Preprocess step

5.5.2. Sequence diagram and Screen of the Parsing step

The Parsing component screen is seen in Figure 20. It is the second step of the modeling process. The main responsibility of this screen is to break down the *structure sentence* into the KISS modeling concepts*: candidate object, action*, and *subject*. The list of *structured sentences* came from the Preprocess step, and the end-user selects a word or a combination of words as the *candidate object*, *action*, and *subject*. Then the end-user chooses a type (object, action, and subject) and ID of the *structured sentence* for the *candidate object*, *action*, and *subject*. After this step, *the candidate object*, *action*, and *subject* are considered as a *definitive object*, *action*, and *subject*, respectively.

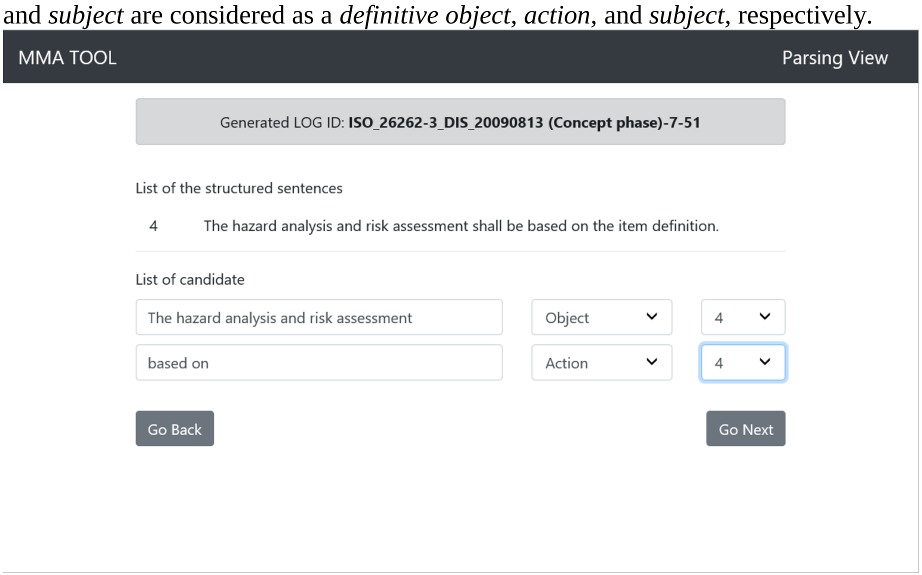


Figure 20. Parsing screen

After the Preprocess step, the communication flow continues with the Parsing step background of the Parsing screen. The parsing step (see Figure 21) consists of the following two activities.

1. The safety engineer picks a word or combination of words as a *candidate object, action,* and *subject* in the Presentation tier. While the safety engineer is adding more candidates on this tier, it handles all picked candidate *objects, actions,* and *subjects.*
2. When the safety engineer moves to the Analysis step using the *Go Next* features, the picked *candidate objects, actions, and subjects* are saved to the database through the application tier and become the *definitive object, action,* and *subject, respectively*.

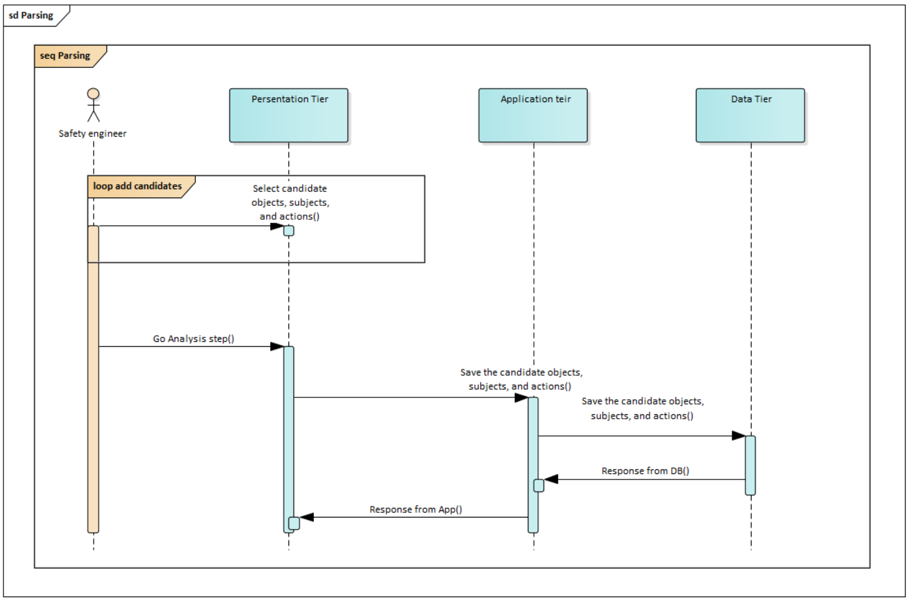


Figure 21. Sequence diagram of the Parsing step

5.5.3. *Sequence diagram and Screen of the Analysis step*

The analysis screen (see Figure 22) is the third step of the modeling process. The main task is to receive a definition for the *definitive object*, *action*, and *subject* from the end-user. In most of the cases, the definition comes from the ISO 26262 part 1, but it is possible to provide a distinct definition (*in Definition*) regarding the modeling purpose. It is essential to define the *definitive object* and *subject*. In the initial modeling step, it becomes the model *object* and *subject* with UML notation. Moreover, when the end-user clicks one of the *definitive object*, *action*, and *subject*, the tool shows a relevant *structured sentence*. After this step, *definitive object*, *action*, or *subject* is considered as a *model object*, *action*, and *subject*, respectively.

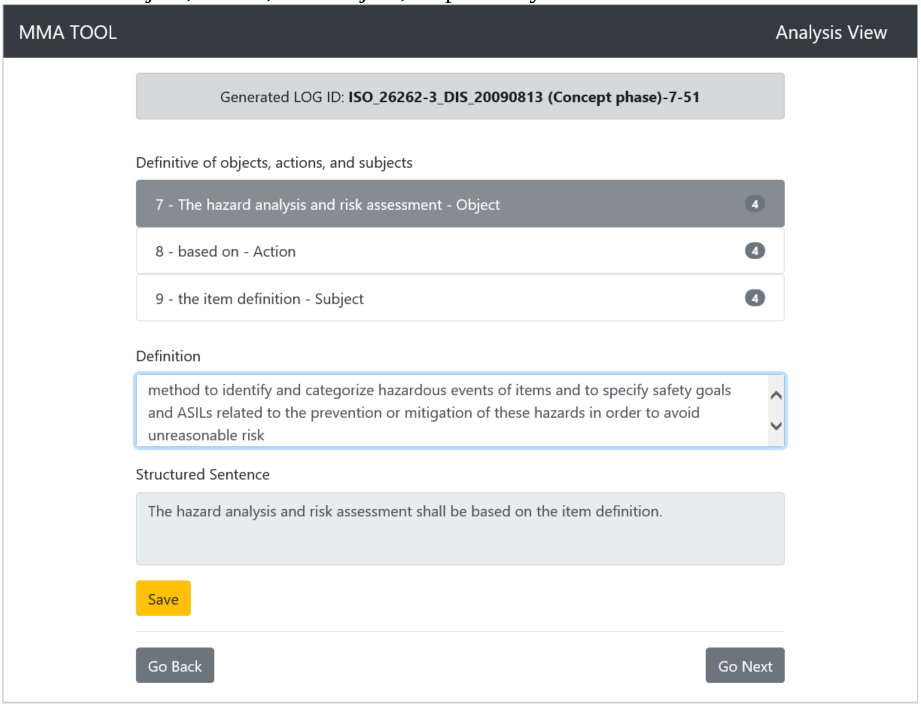


Figure 22. The analysis screen

Background of the analysis screen, the communication remains the sequence diagram (see Figure 23) of this step. The sequence diagram shows the two following activities:

1. The safety engineer appends a definition to *definitive object*, *action*, or *subject* in the Presentation tier using *Save* feature. During the *Save* feature, the presentation tier sends the *definitive object*, *action*, or *subject* to the Application tier. The application tier accepts the request on fixed end-points and passes the *definitive object*, *action*, or *subject* to the database.
2. The safety engineer moves to the Initial modeling step using the *Go Next* feature.

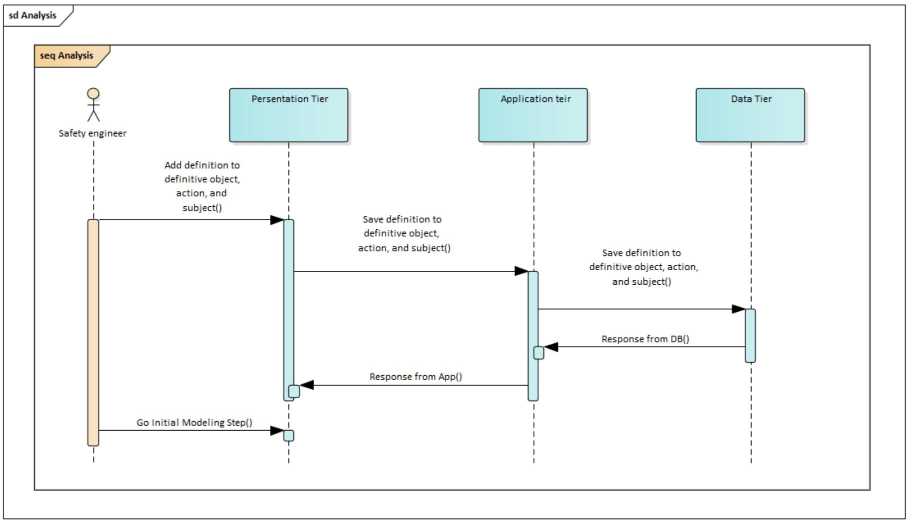


Figure 23. Sequence diagram of the Analysis step

5.5.4. *Sequence diagram and Screen of the Initial modeling step*

The Initial modeling screen (see Figure 24) is the last step of the modeling process in the MMA tool. The main responsibility of this screen is to provide notations for initial models. As it is stated, the resulting model needs to be exported to the *UML Domain model* diagram type in Enterprise Architect (EA) via the Simplexity application.

EA UML notation types:

EA Class, Abstract, or Class type is acceptable for the *model object* and *subject*.

EA Associate, Generalize, Parameter is options for the *model action*. In the *model action*, we need to define the *EA Input model ID* and *EA Output model ID* because the model action denotes an integration between the *EA Input model* and the *EA Output model*. When the end-user finishes all notation mapping in this screen, the end-user can export EA models to EA application using the *Export to EA* feature.

This feature triggers an exporting process from the MMA application to Simplexity.

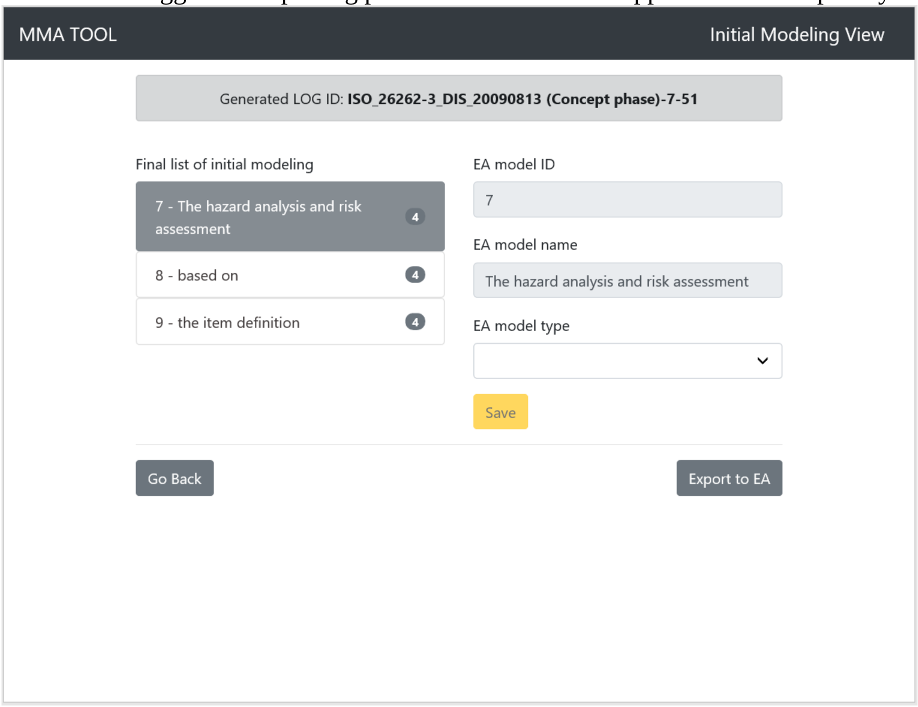


Figure 24. Initial Modeling screen

Background of the analysis screen, the communication flow is ended by the sequence diagram of the Initial Modeling step. This sequence diagram of this step is given in Figure 25 and consists of the following activities.

1. The safety engineer gives a UML notation type for a *model object, action*, and *subject* using the *save* feature.
2. When the safety engineer clicks the *save* feature, the following operations are executed sequentially.
    a. The Presentation tier sends the *model objects, action*, or *subject* with the UML notation type to the Application tier.
    b. The Application tier receives the *model objects, action, or subject*, and it transmits the received information to the Data tier.
    c. The Data tier returns a response for that operation to the application tier.
    d. The Application tier returns the response to the Presentation tier.
3. Then, the safety engineer can export the models to the Simplexity using the *Export EA* feature. The background process of this feature executes the following sub-step:
    a. The Presentation tier sends the model objects, action, and subjects with the notation type to the Simplexity application.
    b. The Simplexity gives a response to the request to the Presentation tier.
    c. The Presentation tier saves the Simplexity response.

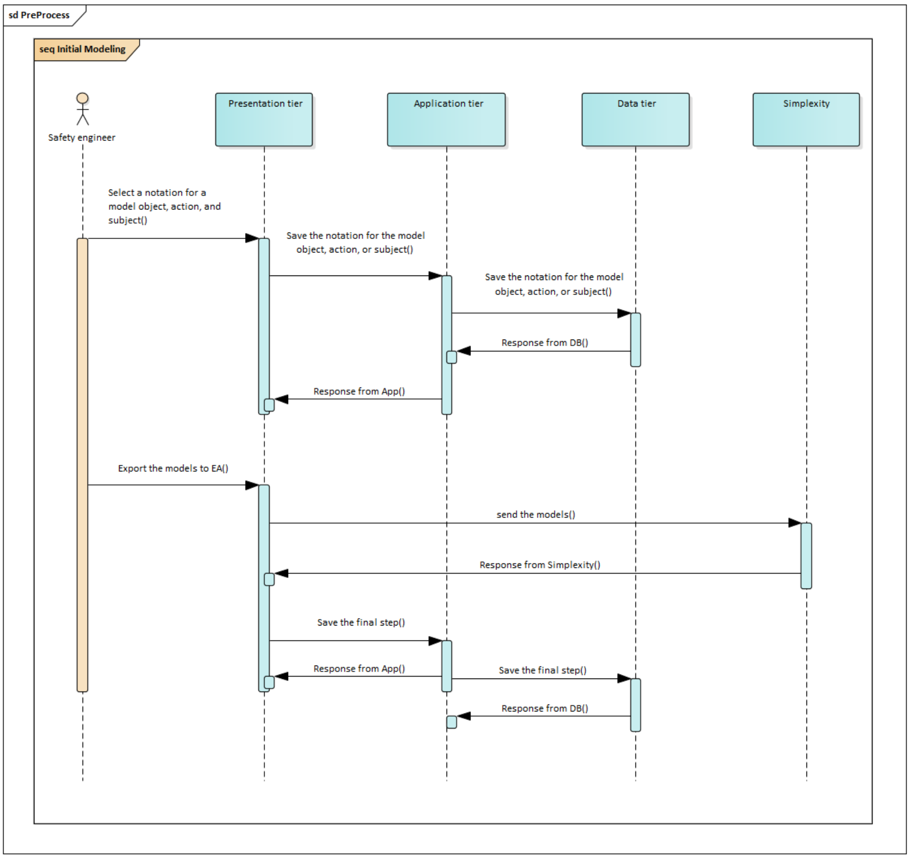


Figure 25. Sequence diagram of the Initial Modeling step

6. Tool Design

The chapter describes the full design of the MMA tool. It starts with the design decision and constraint in Section 6.1, and the chapter extends with the technology decisions and their rationales. The full design of the Presentation tier is revealed in Section 6.3. The component diagram of the MMA tool is described in Section 6.4. This chapter concludes with the detailed design of the application and data tier in Section 6.5.

6.1 Design decision and constraint

The project objective derived software product (MMA tool), and the product was built based on requirements. Some of the requirements led to important design decisions, while some of them directed to design constraints. Figure 26 captures the decision and constraint on how it is derived from the requirements.

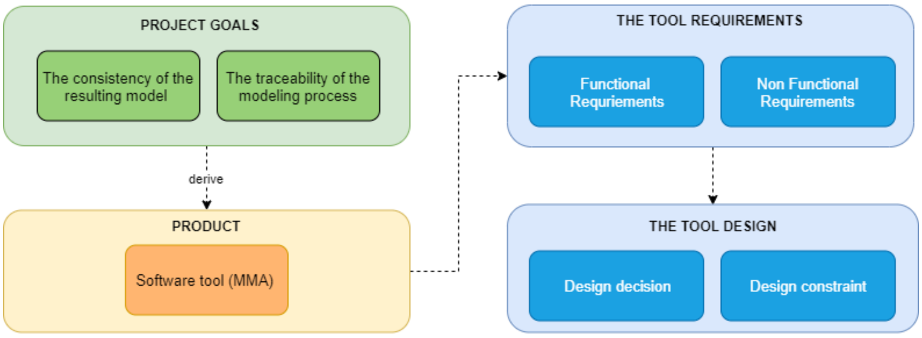


Figure 26. The derived design decision and constraint

This design decision and constraint have a high impact on technology decisions and the implementation of this project. Table 15 explains the design decision and constrain with their rationale.

Table 15. The design decision and constraint with its rationale

Design decision (DD) or constraint (DC) definition	Derived requirement	Rationale
DD1: The MMA tool needs to integrate through an add-in with the MS Word application.	The tool shall be compatible with the MS Word application by exchanging information between a MS Word document containing the ISO standards and norms and the tool.	There are two alternatives on the design level. The MMA tool can integrate through an add-in with the MS Word application, or the MMA tool needs to open the MS Word file inside itself. We decided to take the first option, and the second option was ignored because the end-user prefers to use the full functionality of the MS Word application.
DC1: The MMA tool needs to integrate with the Simplexity application.	The tool shall export the resulting models to the Simplexity application.	Simplexity tool has not sufficient amount of models due to manual work. The company stakeholders had a general idea to tackle this issue based on Simplexity project experiences. This reasoning initiated this requirement and resulted in this design constraint.

6.2 Technology decisions and rationales

The technology decisions are vital for this project's feasibility. Limitations of technologies can easily derive to the design changes or the project implementation delay. Another important consideration was cost efficiency from a business viewpoint. The next sub-sections explain the technology decision for the tier by tier.

6.2.1. Presentation tier technologies

The presentation tier has to carry out most of the complex functionality, such as processing multiple sentences, exchanging information to the MS Word document. Design decision 1(**DD1**) mainly influenced the technology decisions of the presentation tier. Regarding Design Decision 1(**DD1**), there were two options at the technology level. The first option was using the JavaScript API [15], and another opportunity was using the .NET framework [16] for the entire presentation layer.

TNO IVS stakeholders preferred the first option due to cost efficiency. We identified two types of cost had the second option. The first expense was deployment cost and an editing tool cost. NET application is only deployed on Internet Information Service (IIS), and it could lead to extra charge for the deployment of the MMA tool. This cost also could be extended as the editing tool cost; it was essential for purchasing MS Visual Studio. The second charge was the training cost of the potential developer after the project finished.

JavaScript API of MS Word [15] enables any web application that interacts with the object models in Office host applications. It is based on JavaScript [17]. The leading technology of the presentation tier is Angular [18], and Angular is a JavaScript Framework that allows creating reactive applications in HTML and Typescript [19]. Supporting technology is Bootstrap [20], and it is a component library for responsive and extensive HTML Graphic User Interface. Figure 27 presents the Presentation tier regarding the chosen technologies.

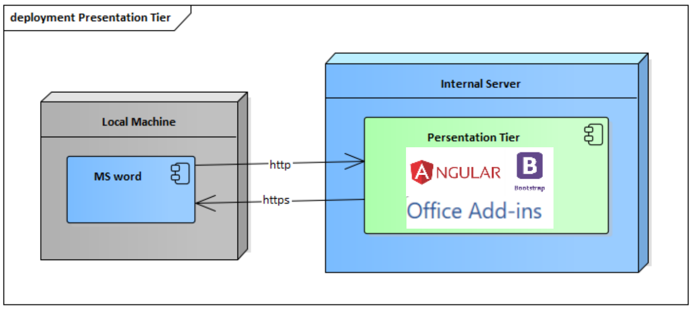


Figure 27. Presentation tier with the technologies

6.2.2. Application tier technology

Design Constraint 1 (**DC1**) affected the technology decision of the application tier. The technology (see Figure 28) of the Application is the Django Rest framework [21]. The potential developer stakeholder from the TNO IVS wanted to have this technology for the application tier because of the backend part of the Simplexity application using this technology. Furthermore, the decision relied on supporting materials of the Django rest framework, namely the official documentation and tutorials.

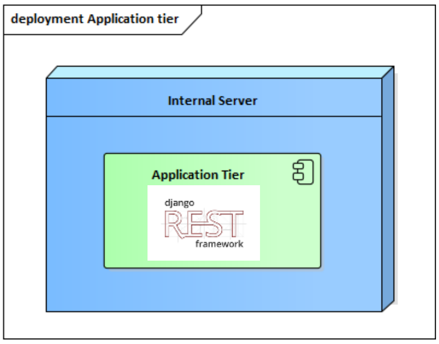


Figure 28. Application tier with the technology

6.2.3. Data tier technology

In the date tier, there were two candidates, namely Graph Database and Relational Database. We decided to have the Relational Database because our data models are predefined in the application tier. The data model and their relations are represented easily in the database table and relationships of the table, respectively. In technology, MySQL [22] is the chosen Data Base Management System (DBMS), and the main reason was the MySQL database commonly used for the TNO IVS department. Figure 29 illustrates the data tier with the chosen technology.

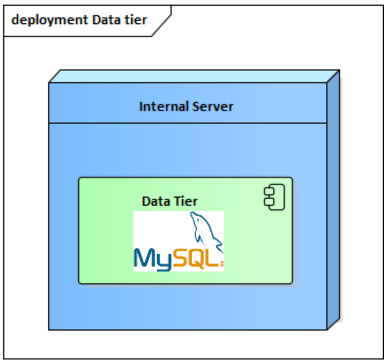


Figure 29. Data Tier with the technology

6.3 The presentation tier design

6.3.1. Angular concept

The Word add-in is designed in a web-based manner, and this tier provides a GUI for the end-user. In essence, the responsibilities of this tier required advanced technology, and Angular was chosen. Angular is a Single-Page Application (SPA) and component-based framework.

The concept of SPA was introduced in 2003 by Stuart Morris [23]. The SPA is a web application that interacts with the end-user through dynamically overwriting the current page rather than loading the entire new page from a server. The behavior of the SPA is more similar to a desktop application because it serves as an excellent user experience between successive pages and web applications.

As a component-based framework, Angular encapsulates the processing logic into modular components, and each component consists of class and view. The class contains application data and logic, and the view defines the HTML template to be displayed.

Angular distinguishes between component and service in terms of modularity and reusability. However, Angular doesn't differentiate the notation of the service and components, and it provides different libraries for them. The component increases the modularity by separating view and application data logic. The service manages specific tasks such as fetching data from the user, logging, and exchanging data across the components, and it can be reusability for the different parts of an application.

6.3.2. Component decomposition

Every Angular application has to have one component, which is the root component. The root component connects a hierarchy with the page Document Object Model (DOM). In the presentation tier of MMA, we designed seven components, namely Welcome, Preprocess, PreprocessOverview, Parsing, Analysis, and Initial modeling, under the root component of Angular by decomposing parts of this tier. We were breaking down into several and smaller components that allowed improving two non-functional requirements (Modularity and Modifiability). Each of the components provides a specific GUI and has a particular purpose based on the process model of the tool. In the next sub-section, we describe only Welcome and PreprocessOverview components in a more detail level regarding GUI of it. Screens of Preprocess, Parsing, Analysis, and Initial modeling components are explained in Section 5.5 in terms.

Welcome Component

The GUI of the welcome component is seen in Figure 30. The welcome component is the starting page. When the end-user opens the MMA tool, It shows this component. The left pane shows the MS Word document, and the right pane shows the MMA tool in the add-in. The screen of this component provides only a few features, namely the tool name (*MMA TOOL*), Current view (*Welcome View*), the tool guideline (*MMA tool guideline*), and initial actions(*Start modeling* and *Section Id*). The MMA requires *Section Id* from the end-user and takes the opened file name to *Artifact Name* automatically. The *Artifact Name* and *Section Id* are used for traceability of the modeling process.

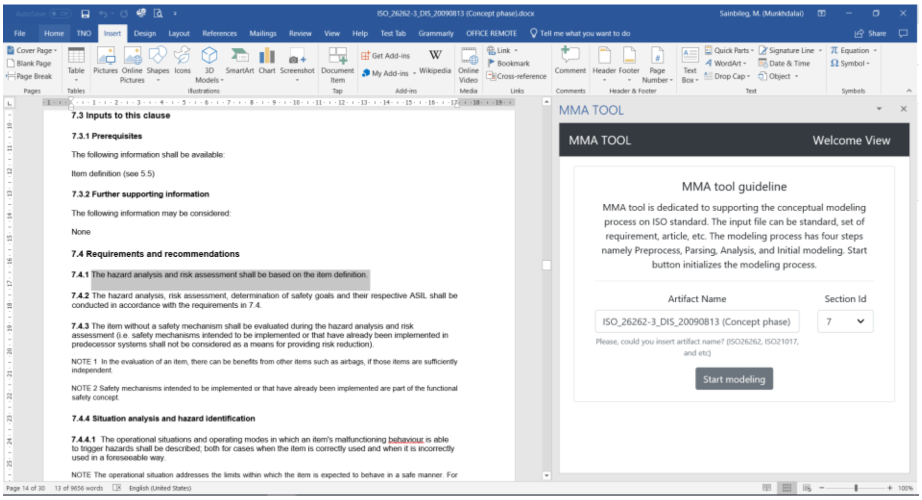


Figure 30. The welcome component screen

PreprocessOverview Component

The Preprocess screen (see Figure 31Figure 31) is the third page of the tool and a sub-step of the Preprocess step. The main purpose of this screen is to give an overview of the Preprocess step. It displays a list of the *original texts* which is processed in the Preprocess step. If the end-user clicks one of them, it shows the number of *structured sentences* with a different background, which is related to the *original text*. The green background represents that a *structured sentence* goes to the next step while the red background denotes a *structured sentence* that does not go the next step.

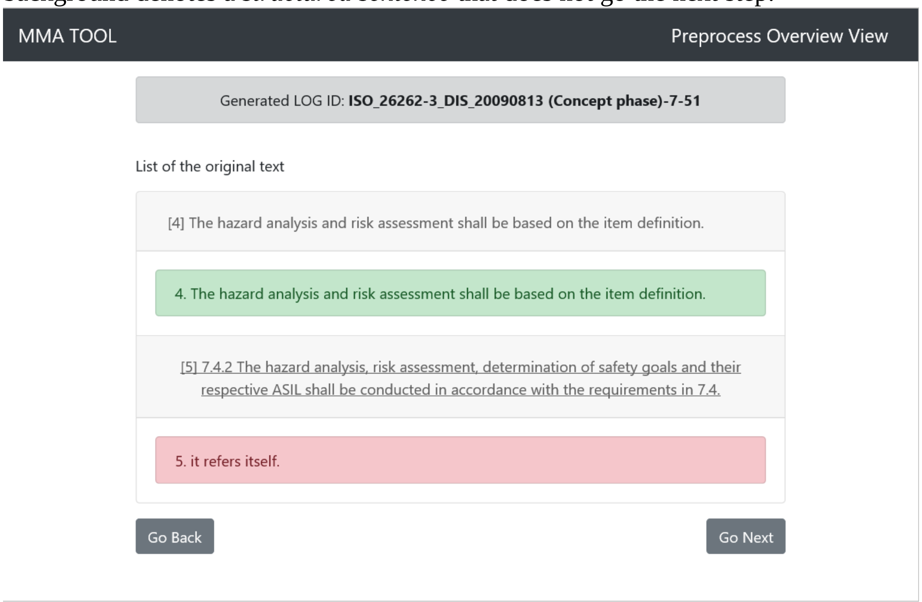


Figure 31. PreprocessOverview Component screen

The term checker feature

The standards usually include particular terms. For example, ISO 26262 part 1:Vocabulary introduces all terms which can be used for the rest of the parts of the ISO standard. The term checker can identify these terms during the modeling process, namely Preprocess, Parsing, and Analysis steps.

In the Preprocess step, the term checker checks the *predefined term* from the *original text* and shows the definition of the term. In Figure 32, the checker found two *predefined terms* (*hazard* and *hazard analysis and risk assessment*) from a particular sentence.

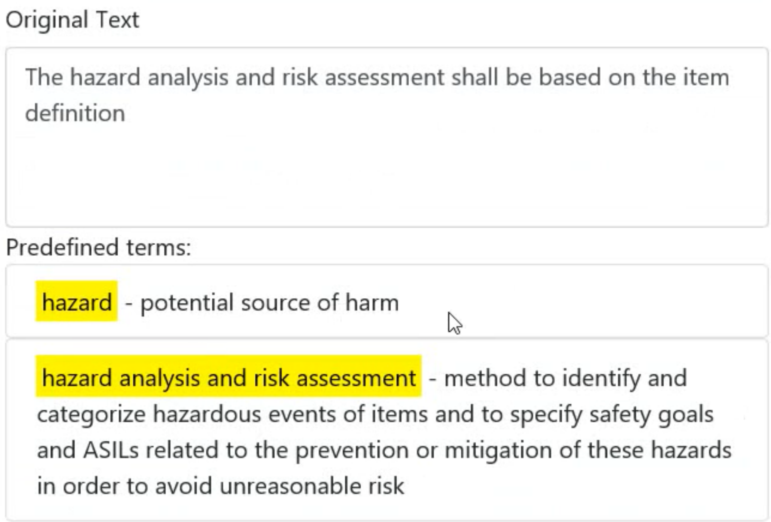


Figure 33. The term checker in the Preprocess step

In the Parsing step, this feature also detects the *predefined term* from the *structured sentence* and presents the definition of the term. In Figure 34, the checker found two predefined terms (*hazard* and *hazard analysis and risk assessment*) from a certain *structured sentence.*

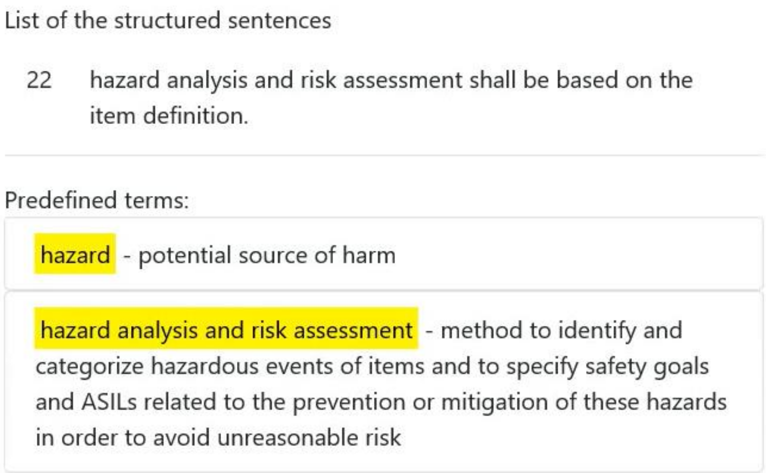


Figure 34. The term checker in the Parsing step

In the Analysis step, If the term checker finds the *predefined term*, it automatically provides the definition of the term. In Figure 35, the feature found one predefined term (*hazard analysis and risk assessment*), and it provided the definition of the term in *Definition input.*

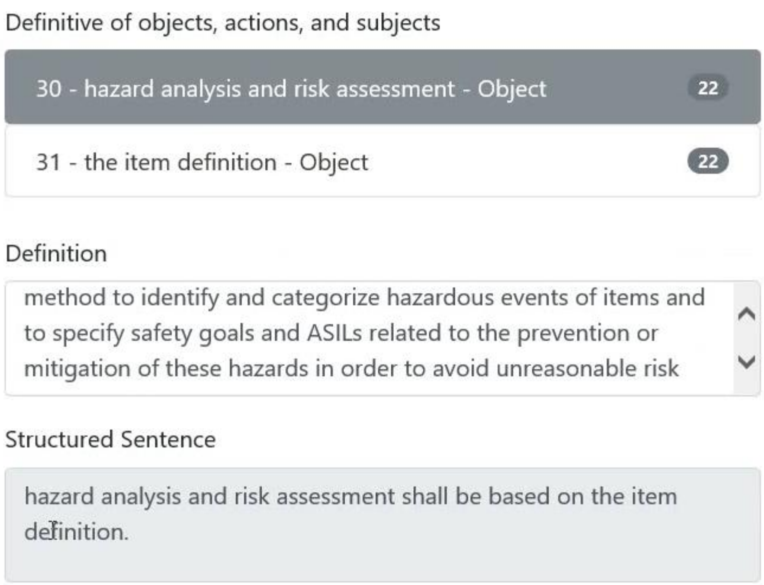


Figure 35. Term checker in the Analysis step

6.3.3. Service

Service is typically a class with a narrow, well-defined purpose in Angular. The service provides specific tasks such as fetching data from the server, logging to different resources, and exchanging information across components. We defined four different types of services (**Word API service**, **App Tier service, Common Services, and Simplexity Services**) for building a communication channel between the components and external applications (Simplexity and The application tier of the MMA tool) and exchanging information cross-components.

Word API service: This service is designed to handle tasks that are related to the MS Word application.

App Tier service: This service is designed to fetch, save, filter, update data models to the Application tier. Also, it manages communication between the Application tier and the presentation tier.

Common Services: These services are designed for Dependency Injection (DI) of all defined components. It supplies an object with the instances of its dependent objects from outside instead of creating them inside it.

Simplexity Services: This service is designed to fetch and save the resulting model to the Simplexity application.

6.4 *Component diagram of MMA tool*

In this section, the MMA tool features are explained at the component level. The component diagram (see Figure 36) describes the MMA tool design on a detailed level. As mentioned, Angular does not distinguish the notation for component and service, and we followed the Angular principle. All components and services are the same in the component diagram naming are different.

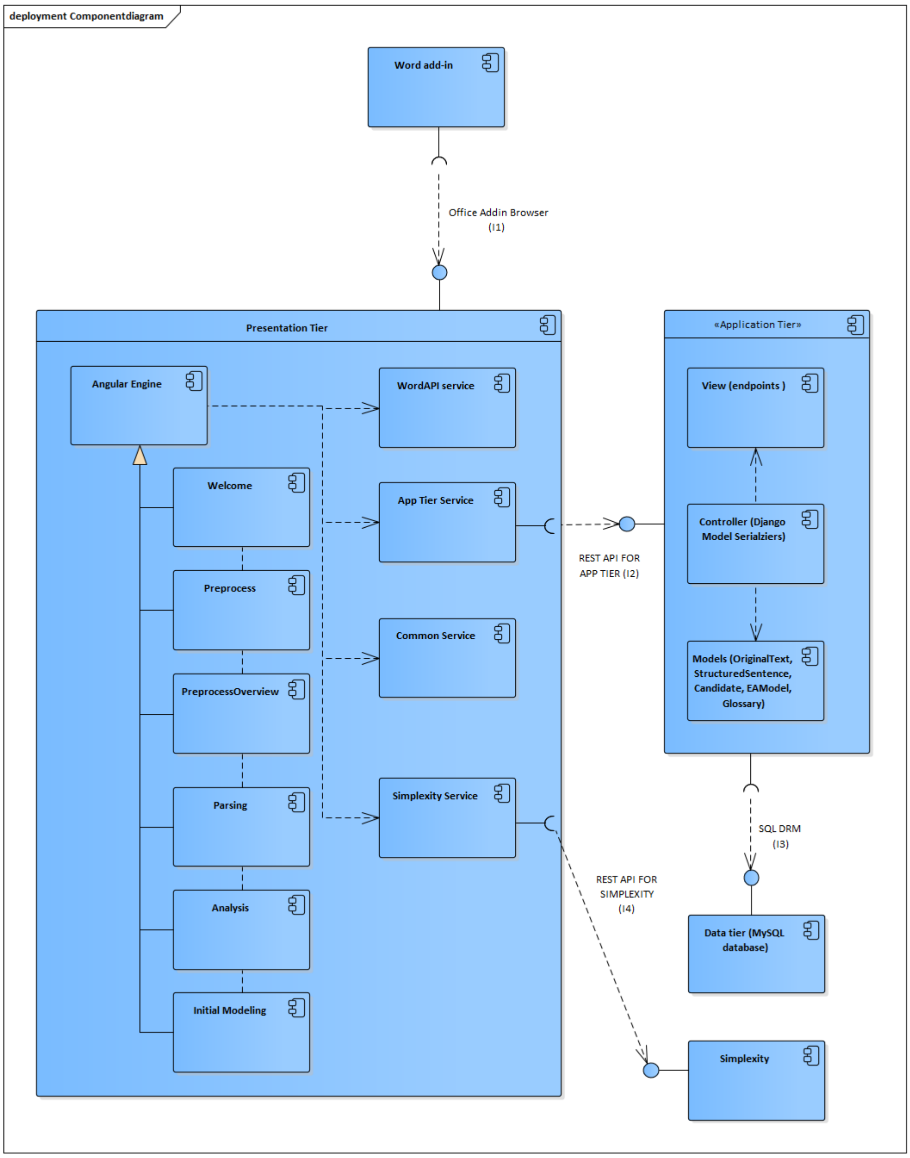


Figure 36. The component diagram of the MMA tool

The tool contains four interfaces: Office Add-in Browser (I1), Rest API for the App tier (I2), SQL DRM (I3), and the Rest API for Simplexity (I4).

- **Office Add-in (I1)**: The main task of this interface is data exchange between the presentation tier of the MMA tool and MS Word application by the shared add-in. Due to data confidentially and security, only the shared-add-in is considered as the safest way.
- **REST API for App tier (I2)**: This interface serves as a communication channel from the Presentation tier to the Application tier by REST API. Five endpoints are defined in the Application tier, and it needs to have Create, Read, Update, Delete (CRUD) operations on the data models. A detailed explanation of the Application Tier component is given in Section 6.4 .

- **SQL-DRM (I3) interface between the Application tier and the Data tier:** This interface is used to store the data models to a database. This database is accessible via an SQL Database Relational Model (DRM) in Django Framework. The data models and associated relations are defined in the Application tier, and it maps to tables on the database scheme.
- **REST API for Simplexity(I4)**: Satisfying Requirement ID 1.3, it is essential to establish a communication channel between the MMA tool and Simplexity. The resulting models pass to Simplexity through this interface.

6.5 The application and data tier design

The application tier is built upon the Django Rest framework using the Model View Controller (MVC) design pattern. All View is defined in the Django Rest framework ViewSet while the Model is declared the Django Rest framework model. The Django Serializer manages both and acts in the Controller role in the Django Rest Framework. Regarding the endpoints, five data models were designed and are seen in Figure 37. So the data models mapped directly to the database table on the data tier.

The data models are designed to save the KISS method [11] concepts, namely *original text, structured sentence, candidate object, subject, action,* and *model object, subject,* and *action* with their relations. Saved the model, and their relationships indicate the traceability of the modeling process based on the Log ID. The Log ID is a unique ID for each modeling process, and it consists of Artifact Name, Section Id, and the three random digit.

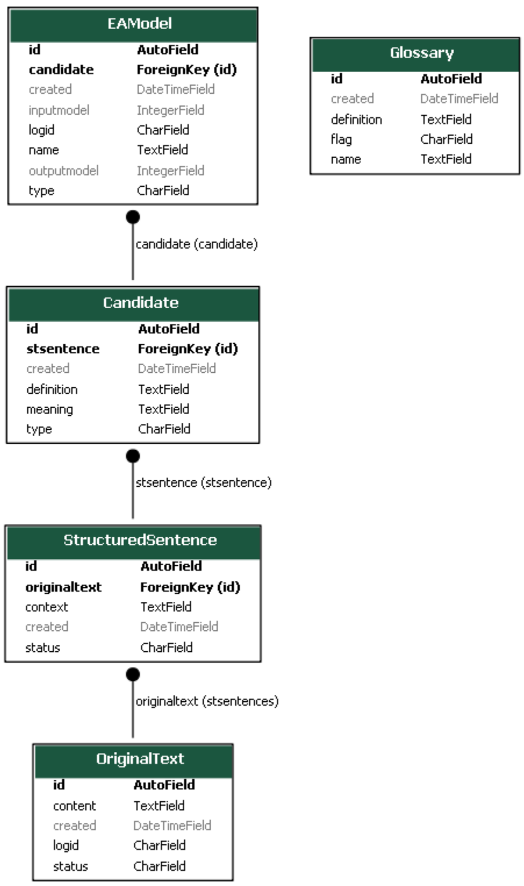


Figure 37. Data models are generated by Django graph extension

7. Tool Implementation

This chapter elaborates on the implementation of the project by presenting the MMA tool design implementation based on the design of it.

7.1 Implementation of the presentation tier design

The implementation of the Presentation tier design is shown in Figure 38 as the *MMA-PER-TIER* Angular project in the Visual Studio Code tool. In the implementation, all of the component and service names are in a lower case. It can be easier to read the source code and prevents from Angular CLI (specific compiler of Angular) complaining. All of the services are packed in the *service* folder. Also, All of the data model types are defined in the *model* package. The Initial Modeling component is renamed *modeling* in the implementation. *Footer* and *header* components are only for styling purposes.

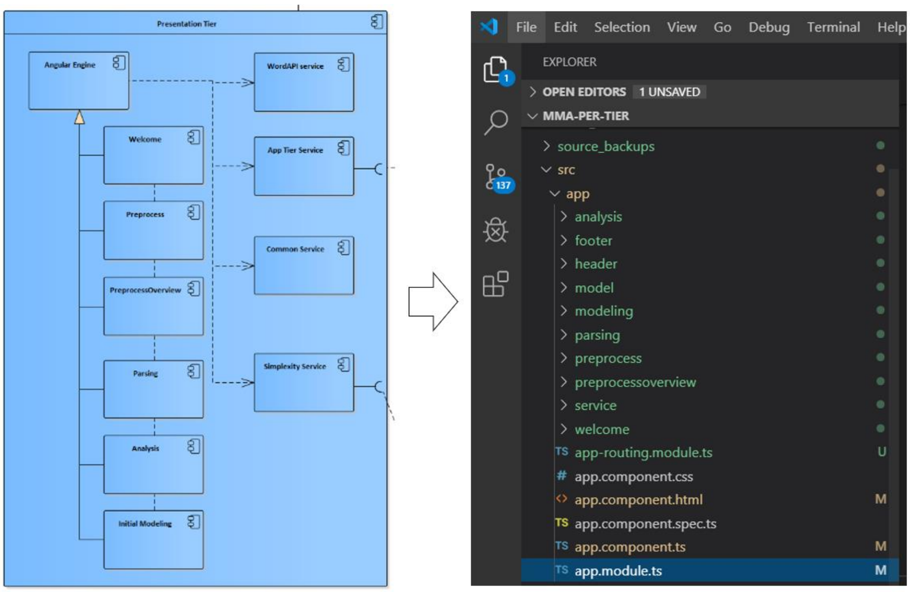


Figure 38. The implementation of the presentation tier design

7.2 Implementation of the application tier design

The implementation of the application tier design is presented in Figure 39 as a Python project (*mma-app-tier*) in the PyCharm editing tool. All models are defined in the *models.py*, and all endpoints are declared in *views.py*. In the Django Rest framework, *Serializers.py* acts as a controller role and handles parsing and formatting operations.

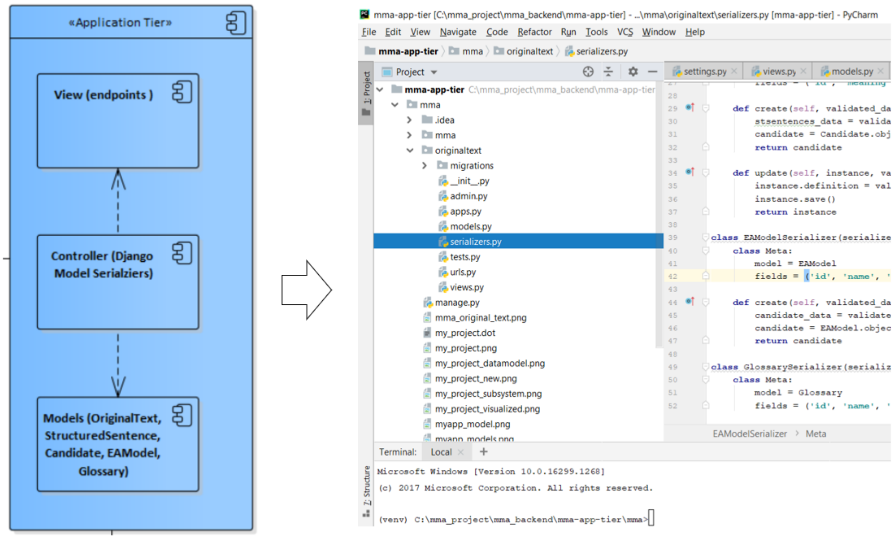


Figure 39. The implementation of the application tier design

7.3 Implementation of the data tier

The implementation of the data tier design is seen in Figure 40 as a database schema (*mma_db*) database in the MySQL Workbench tool. All models are predefined in the application tier and map to tables in this tier.

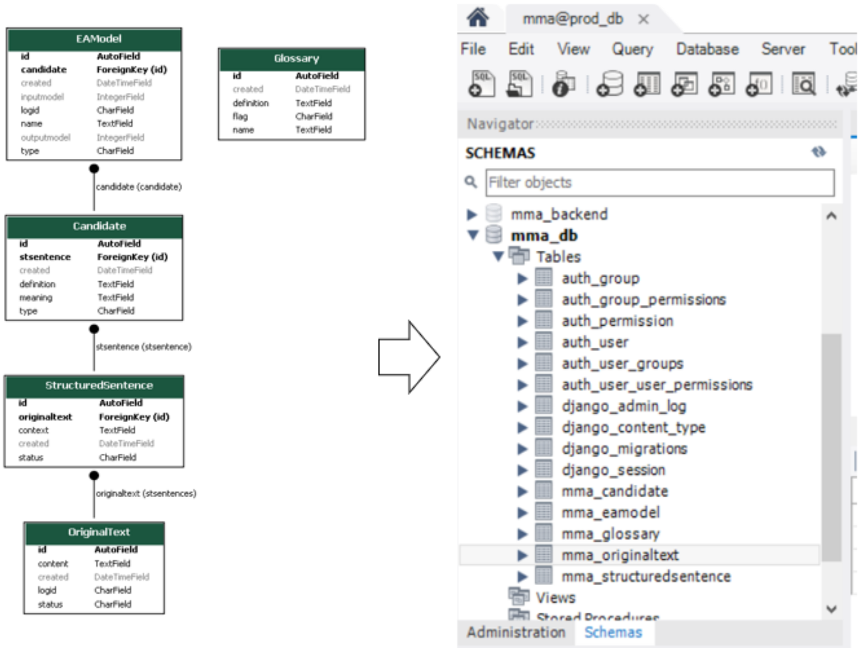


Figure 40. The implementation of the data tier design

8. Tool Deployment

This chapter describes the deployment of the MMA tool.

8.1 Introduction

As mentioned, we are using the three-tier architecture for the MMA tool. It enables us to deploy each tier on a different machine. However, the TNO IVS department stakeholders want to deploy the three-tiers in the single machine.

8.2 Deployment of the presentation tier

The Angular application can be easily tested, configured, and deployed by a webpack [24]. The webpack is a static module bundler for modern JavaScript applications, including the Angular application. When the webpack progresses the Angular application, it internally builds a dependency graph that maps module every module of the angular application needs and generates one or more bundles.

It is essential to give a specific command (*ng build*) to the webpack to initiate deployment progress. The deployment process is shown in Figure 41.

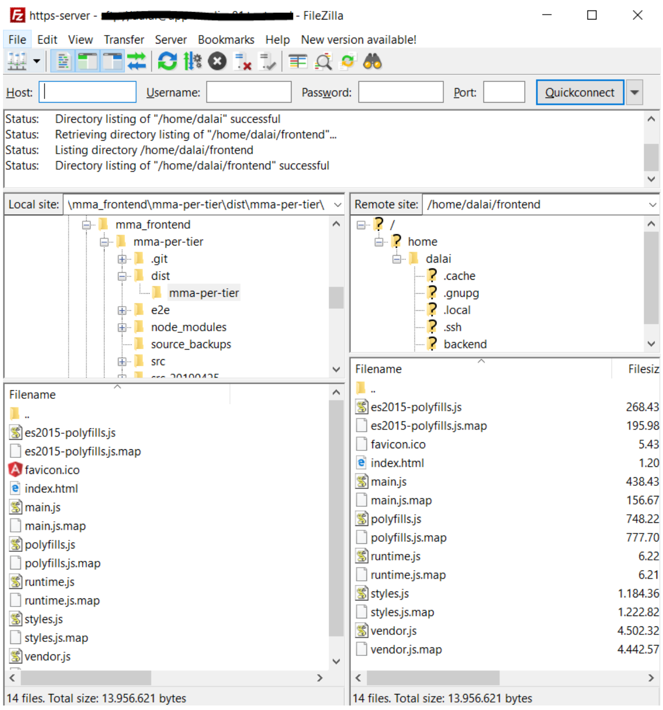
- *Ng build* is the main command that starts the deployment progress
- *base-href* specifies the root of the resources (style sheet and media files.)
- *aot* is an extra command for the angular project optimization in the production environment. The Angular Ahead-of-time (aot) compiler converts the Angular HTML and Typescript code into efficient JavaScript code the build phase before the web browser downloads and runs that code. Compiling the Angular application during the build process gives a faster rendering in the browser.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\mma_project\mma_frontend\mma-per-tier> ng build --base-href "/mma/" --aot

Date: 2019-08-30T09:35:16.782Z
Hash: 52970d773f906c37183a
Time: 32865ms
chunk {es2015-polyfills} es2015-polyfills.js, es2015-polyfills.js.map (es2015-polyfills) 262 kB [initial] [rendered]
chunk {main} main.js, main.js.map (main) 428 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 731 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 1.13 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.29 MB [initial] [rendered]
PS C:\mma_project\mma_frontend\mma-per-tier>
```

Figure 41. Deployment process on the command line of Visual Studio Code tool



The successful deployment process results are stored in the desired directory. Then it should be uploaded to the TNO IVS internal server. Figure 42 presents how to upload the result of the deployment to the TNO IVS internal server using FileZilla application.

Figure 42. Uploading the result of deployment using FileZilla

8.3 *Deployment of the application tier*

In Python, the best practice is creating a separated virtual environment for the production server. When the production server is ready, it is easy to install required libraries using *pip* commands in *-r requirements_prod.txt* file. Figure 43 presents the installation on the command line of the PyCharm tool. *Requirement_dev.txt* file dedicated to the Windows machine and *requirement_prod.txt* file is prepared for the Linux machine due to a library difference between the two different operating systems.

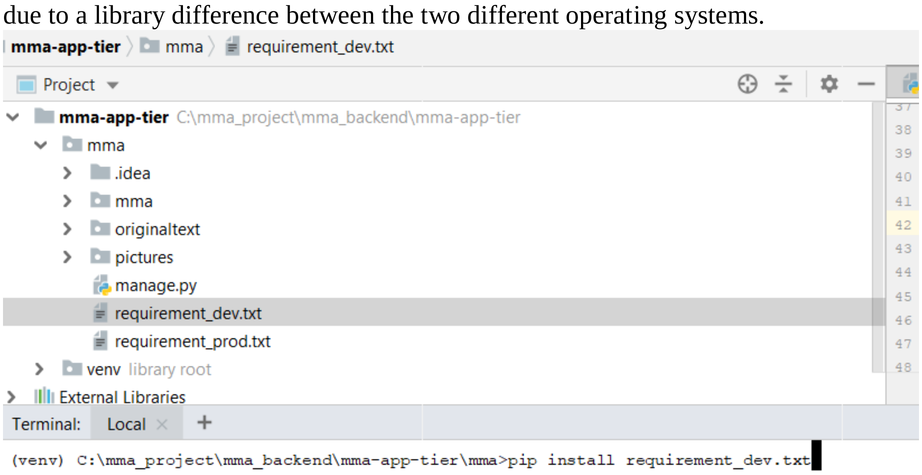


Figure 43. Installation for required packages in a virtual environment

After the installation process, the developer copies the *mma-app-tier* to the TNO IVS internal server. Then it is ready.

8.4 *Deployment of the data tier*

In the data tier, one task of the deployment process is creating an empty database scheme with a database user. Indeed, there is one need to give Create Read Update Delete (CRUD) permission to the database user. This database user needs to be configured in the *settings.py* file inside of the application tier. After these two activities, the database tables are created by the *migrate* command in the application tier (see Figure 44). The application tier of the MMA tool creates all required tables and their relations based on the defined models.

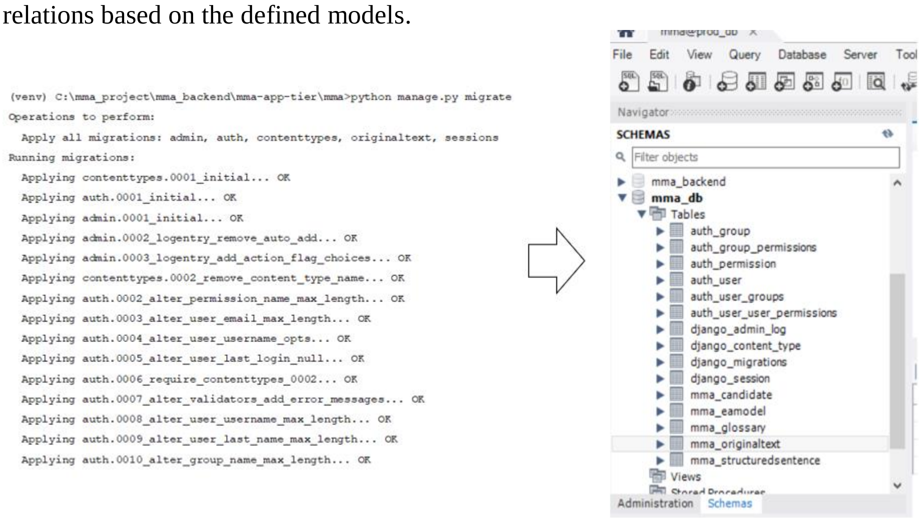


Figure 44. The deployment of the data tier

9. Tool Validation

This chapter explained the validation process of this project.

9.1 Validation

In this project validation process is undertaken by iteratively. In the requirement analysis phase of this project, all of the mock-ups were validated by the end-user. After that, we agreed on developing features of the tool based on the mock-ups views. In the implementation phase of this project, we collected the end-user feedback during the demo, and we improved the particular functionalities of the tool.

9.1.1. Requirement satisfaction of the MMA tool

The tool designed and developed has to comply with the functional requirement specification from Chapter 4 (Tool Requirement). Once the tool is developed, it is possible to evaluate how well the tool satisfies them. The answer for each requirement specification is **satisfied, partially satisfied,** or **not satisfied**. The answers reflect the user's perspective, and all requirement specifications are given in Table 16. Twenty-two requirement specifications are **satisfied**, and two requirement specifications are **partially satisfied.**

Table 16. Requirement specification with justification

REQ ID	Functional requirement specification	The modeling step	Priority	Satisfaction
1.2.0	The tool shall be able to show its Graphic User Interface (GUI) in the MS Word application via a shared add-in.	N/A	Must	**Satisfied**
1.2.1	The tool shall be able to highlight the *processed sentence* in the MS Word document automatically.	Preprocess	Should	**Satisfied**
1.2.2	The tool shall be able to select the next sentence from the MS Word document and take it to the *original text.*	Preprocess	Must	**Satisfied**
1.2.3	The tool shall be able to copy the selected sentence from the MS Word document to the *original text* automatically.	Preprocess	Should	**Satisfied**
1.4.1	The tool shall be able to show the step the Safety engineer is working on.	All of the modeling step	Must	**Satisfied**
1.4.2	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence.*	Preprocess, Parsing, and Analysis	Should	**Satisfied**
1.4.3	The tool shall be able to accept multiple *original texts*.	Preprocess	Must	**Satisfied**
1.4.4	The tool shall be able to accept multiple *structured sentences* for a single *original text.*	Preprocess	Must	**Satisfied**
2.1.1	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	All the modeling step	Could	**Satisfied**
2.1.2	The tool shall be able to generate *Log ID* based on the *metadata.*		Must	**Satisfied**

2.1.3	The tool shall be able to show the *Log ID.*	All the modeling step	Should	**Satisfied**
1.1.1	The proposed process shall have four steps: Preprocess, Parsing, Analysis, and Initial modeling.		Must	**Satisfied**
1.1.2	The Preprocess step might have a sub-step.	Prepro-cess	Should	**Satisfied**
1.1.3	The tool shall be able to offer the start function of the modeling process.		Must	**Satisfied**
1.1.4	The Safety engineer should be able to give *metadata: artifact name* and *section Id* for the *Log ID.*		Should	**Satisfied**
1.1.8	The tool shall be able to show a list of the *structured sentences* with their *status.*	Prepro-cess	Must	**Satisfied**
1.1.10	The tool shall be able to extract *candidate objects*, *candidate sub-jects*, and *candidate actions* from the *structured sentence* on the tool based on the safety engineer input.	Parsing	Must	**Satisfied**
1.1.11	The tool shall be able to show the *candidate object*, *candidate sub-ject*, and *candidate action* in the *structured sentence.*	Parsing	Must	**Satisfied**
1.1.12	The tool shall be able to convert the *candidate object*, *candidate subject*, and *candidate action* to *definitive object*, *definitive sub-ject*, and *definitive action* respec-tively.	Analysis	Must	**Satisfied**
1.1.14	The safety engineer shall able to add the definition for a *definitive object*, *definitive subject*, and *de-finitive action.*	Analysis	Must	**Satisfied**
1.3.1	The tool shall be able to assign a type (*Class, Abstract Class,* and *Attribute*) for the *object* and *sub-ject model* based on the end-user selection.	Initial modeling	Should	**Satisfied**
1.3.2	The tool shall be able to assign a type (*Associate, Generalize,* and *Compose*) for the *action model* based on the end-user selection	Initial modeling	Should	Partially **satisfied**
1.3.4	The tool shall be able to show *the start point* and *the endpoint to the action model.*	Initial modeling	Should	Partially **satisfied**
1.3.5	The tool shall be able to export the *initial models* to *Simplexity.*	Initial modeling	Must	**Satisfied**

9.1.2. Testing of the MMA tool

Due to project execution time, we did a limited number of unit tests on the presentation tier and application tier. The data tier is simple, and there was no additional need to define it.

9.1.3. *The presentation tier testing*

Angular uses the Jasmine test framework for unit testing. When a developer creates an Angular project using CLI, it installs the Jasmine framework to the project. The developer can quickly test the angular project giving ng test CLI command to the Angular project. Figure 45 shows *ng test* command the Visual Studio Tool.

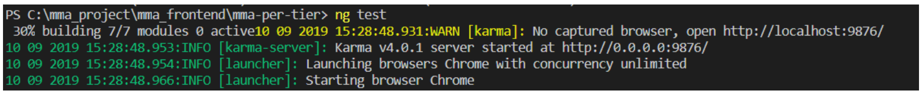


Figure 45. Angular CLI test

As a result, the Jasmine framework checks all defined test cases. It opens a default browser and shows the test result in the template in Figure 46. In this picture, three marked the test cases were checked and passed.

Figure 46. Jasmine framework window

9.1.4. *The application tier testing*

As it is stated, Python provides a variety of testing libraries and frameworks. We used the Django framework test library (*rest_framework.test*) for the unit testing.
This library provides Mock client API, which can send any GRUD rest to the application tier. Also, we defined some test cases for the endpoint of the application tier inside of *tests.py*. Figure 47 shows 5 test cases passed the unit testing.

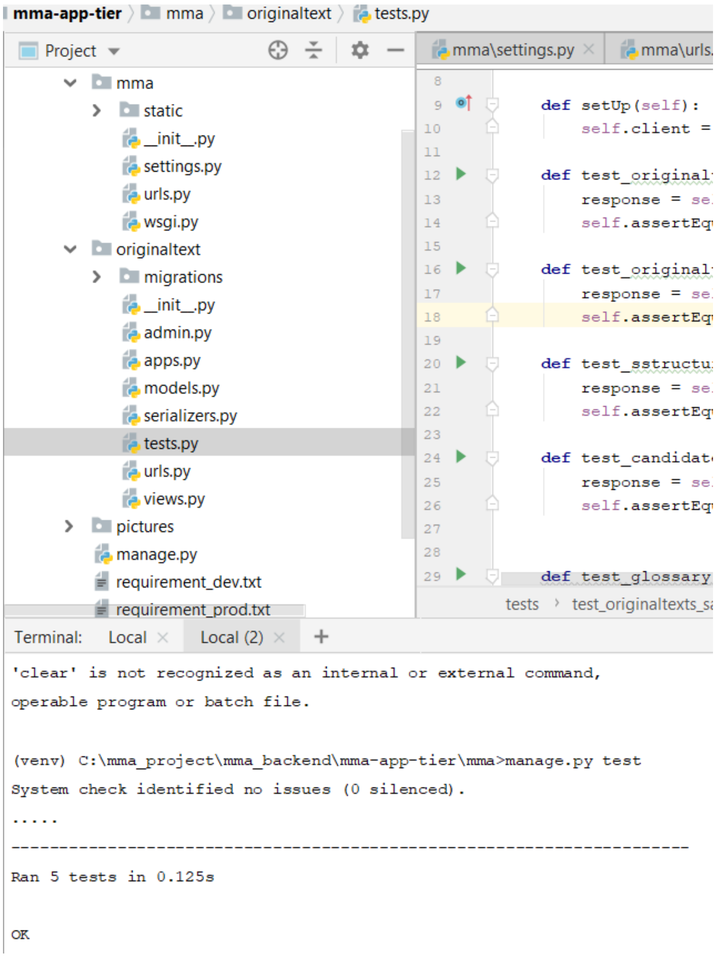


Figure 47. Unit test of the application tier

10. Conclusions

This chapter addresses the conclusions of this project by presenting the results and future work.

10.1 Results

Ultimately, this project aims to reduce a large amount of manual work by supporting the automated tool for modeling. To achieve this goal, we designed and developed the MMA tool. Now, this work is partially automated by the MMA tool.

In more precisely, in the scope of this project, we addressed two aspects of the manual work.

The first aspect is the consistency of the resulting models, and it highly depended on the end user's carefulness. There was a room for a mistake during the modeling process because of less-structured and low-controlled manual work. The mistake led to resulting inconsistent models. Now, the MMA tool offers a systematic process for conceptual modeling. As a result, the end-user can make more effective and correct models from the ISO standards by step by step on the MMA tool.

The second aspect is the traceability of manual modeling. It had several steps to produce the resulting models from the ISO standard requirements. The end-user needed to aware of that which the ISO standard requirement specifications are associated with the resulting models. This task was entirely manual and time-consuming for the end-user. Currently, The MMA tool keeps track of requirements of the ISO standard step by step to illustrates the resulting models. It offers the ability to track each the resulting model that comes from the particular requirement specifications of the ISO standard. At the same time, the tool can highlight with yellow color the processed sentence or paragraph from opened MS Word document. In the mentioned two ways, the traceability of the modeling process is automated and improved.

In addition, the tool separates conceptual modeling from identifying domain concepts. The separation allows the tool to accept various ISO standards, and it is named Multi-domain Modeling Assistance (MMA) tool. Thus, the MMA tool can be applied to different domains. We made simple domain models from ISO 26262 and ISO 14971 using the MMA tool.

Added-value of this project is the following pair usages of the MMA tool. First, the tool contributes to centralize models by coupling with Simplexity, and the resulting models of the MMA tool can be used for TNO projects and products. Second, the tool can be useful for the different ISO standards, the norms, and the European Union (EU) project requirement.

In conclusion, during the lifetime of the project, we have successfully designed, implemented the MMA tool in the TNO IVS department. The tool is also deployed on the TNO internal server, and it is ready for daily use.

10.2 Future work

During the development phase of the project, we have identified future possibilities and improvements as well as features that were not implemented due to various constraints such as time, technology, complexity, and Simplexity changes.

The most important feature is saving the modeling process status in any step and at any time of the modeling process. This feature enables the end-user to create a larger complex domain model from the ISO standard. The time of the modeling process highly depends on the size and complexity of the proposed domain model. In some cases, the end-user wants to take a long or short time pause during the modeling process.

Another improvement is presenting more Enterprise Architect (EA) UML notations. Currently, the MMA tool can give an EA UML class diagram to the conceptual model, and it is essential to provide the EA UML relationship and attribute notations. Due to the development of the Simplexity application, the EA UML relationship and attribute notations are not implemented in this project.

In general, the MMA is a software product, and the end user feedback is an aspect of the future of the MMA tool. It is essential to collect feedback from the end-user experience. Based on the feedback, the MMA tool can be improved by new features.

11. Project Management

This chapter describes the project management process that was conducted during the project timeline.

11.1 Introduction

This project was practical. It was essential to have a delivery that fulfills the goals of the stakeholders (a tool and a technical report). We decided to follow an iterative-incremental developed combined with an agile methodology. The goal of the project was general and high-level at the beginning of the project. So, the iterative approach was followed throughout the implementation phase of this project.

11.2 Project planning and milestones

This project is implemented for a ten-month between the start of January 2019 and the end of October 2019. The duration of then months consists of 5 weeks on university events and vacation, and 38 working weeks were dedicated to project activities, namely planning and execution. Figure 48 presents the project overview planning.
In this project, we distinguished the project three phases that map 15 activities of this project planning.

Task Name	Duration	Start	Finish	
1 Project analysis and planning	13 days	Wed 1/2/19	Fri 1/18/19	Phase 1
2 Project scope definition	5 days	Mon 1/21/19	Fri 1/25/19	Phase 1
3 Literature study	20 days	Mon 1/28/19	Fri 2/22/19	Phase 1
4 Requirement specifications	10 days	Mon 2/25/19	Fri 3/8/19	Phase 1
5 Architecture and design of the proposed solution	10 days	Mon 3/11/19	Fri 3/22/19	Phase 1
6 Implementation stage 01	19 days	Mon 3/25/19	Thu 4/18/19	Phase 2
7 Short holiday	6 days	Fri 4/19/19	Fri 4/26/19	Phase 2
8 Report and Feedback of Implementation stage 01	10 days	Mon 4/29/19	Fri 5/10/19	Phase 2
9 Implementation stage 02	25 days	Mon 5/13/19	Fri 6/14/19	Phase 2
10 Report and Feedback of Implementation stage 02	10 days	Mon 6/17/19	Fri 6/28/19	Phase 2
11 Implementation stage 03	25 days	Mon 7/1/19	Fri 8/2/19	Phase 2
12 Report and Feedback of Implementation stage 03	15 days	Mon 8/5/19	Fri 8/23/19	Phase 2
13 Report and Integration tasks	20 days	Mon 8/26/19	Mon 9/23/19	Phase 3
14 Final deliverables and Final defence	12 days	Mon 9/23/19	Tue 10/8/19	Phase 3
15 Vacation	13 days	Fri 10/11/19	Tue 10/29/19	Phase 3

Figure 48. Project overview planning

The first phase of the project covers the high-level decisions about the project. It starts on *Project analysis and planning* and finishes on *the Architecture and design of the proposed solution*.

The first two activities (*Project analysis and planning* and *Project scope definition*) are proposed to a concrete scope for this project. The *Literature study* is aimed to find out the primary methodology for the modeling process, and a milestone was *an approach decision* (see Figure 49).

In the next two activities, we mainly focused on requirement specifications and architecture decisions. The milestone was *Architecture and technology decisions* (see Figure 49).

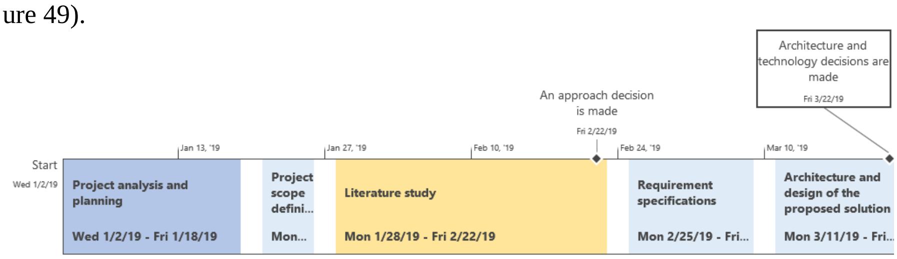


Figure 49. Milestones of Phase 1 of this project

11.3 *Project implementation*

The second phase of the project covers all the implementation of this project. It starts on *Implementation stage 01* and ends on *Report and Feedback of Implementation stage 03*. We defined three iterations for the implementation of this project. In every stage of implementation, we improved the tool based on functional requirements specifications. Milestones of this project are given in Figure 50.

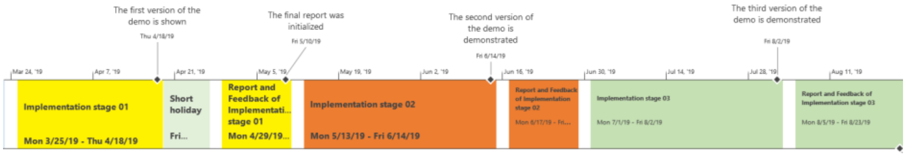


Figure 50. Milestones of Phase 2 of this project

11.4 *Project integration*

As mentioned, this implementation was needed to deploy in the TNO IVS department build environment. Also, this project had a decision constraint. These two reasons initiated the project integration phase. The third phase of this project and its milestones are given in Figure 51.

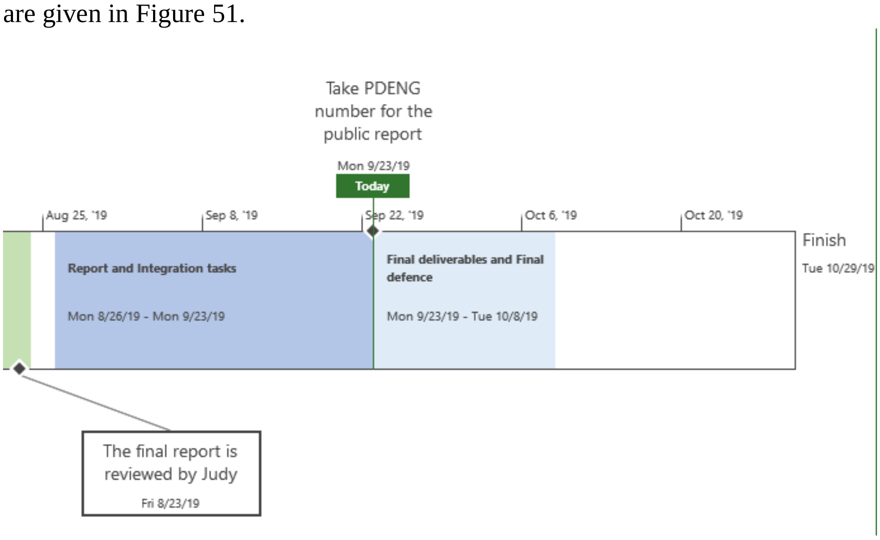


Figure 51. Milestones of Phase 3 of this project

After this three-phase, the final documentation and the tool are delivered.

12. Acronyms

DC1	Design Constraint 1
DD1	Design Decision 1
EA	Enterprise Architect
HLS	The High-Level Structure
IVS	Integrated
IVS	Integrated Vehicle Safety
MBE	Model-Based Engineering
MMA	Multi-domain Modeling Assistance
MS Word	Microsoft Word
NER	Named Entity Recognition
NLP	Natural Language Processing
PDEng	Professional Doctorate in Engineering
PMP	Project Management Plan
POST	Part of Speech Tagging
SPA	Single-Page Application
Term	Definition
TU/e	Eindhoven University of Technology
UML	Unified Modeling Language
UML	Unified Modeling Language
GUI	Graphic User Interface

13. References

[1] ISO, "ISO standards," International Organization for Standardization, [Online]. Available: https://www.iso.org/standards.html.

[2] International Electrotechnical Commission, "International Electrotechnical Commission," International Electrotechnical Commission, [Online]. Available: https://www.iec.ch/functionalsafety/.

[3] ISO 26262: Road Vehicles - Functional safety, Geneva: ISO, 2018.

[4] I. O. f. Standardization, "Medical devices - Application of risk management to medical devices," ISO, Switzerland, 2000.

[5] Stachowiak, in *Allgemeine Modelltheorie*, Springer-Verlag, 1973, p. Part 4.

[6] O. M. G. (OMG), "Unified Modeling Language (UML) specification V2.5.1, 2017," International Organization for Standardization,, 2016. [Online]. Available: https://www.omg.org/spec/.

[7] Eclipse Foundation, "eclipse," Eclipse Foundation, [Online]. Available: https://www.eclipse.org/modeling/emf/.

[8] M. Adedjouma, "Requirements engineering process according to automovite standards in a model-driven framework," Université Paris Sud, Paris, 2012.

[9] Y. Luo, "From conceptual models to safety assurance: applying model-based techniques to support safety assurance," Eindhoven, 2016.

[10] A. K. Saberi, "A Holistic Domain Model of ISO 26262 for Model-Based Safety Engineering," Helmond, 2017.

[11] G.Kristen, "Grammatical analysis," in *Object Orientation The KISS Method*, Massachusetts, Addison-Wesley, 1994, pp. 83-114.

[12] Wikipedia, "Natural language processing," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[13] C. D. M. S. J. B. J. F. S. J. B. a. D. M. Manning, "The Stanford CoreNLP Natural Language Processing Toolkit in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics," *System Demonstrations*, pp. 55-60, 2014.

[14] Wikipedia, "MoSCoW method," [Online]. Available: https://en.wikipedia.org/wiki/MoSCoW_method.

[15] Microsoft, "JavaScript API for Office," Microsoft, 30 07 2019. [Online]. Available: https://docs.microsoft.com/en-us/office/dev/add-ins/reference/javascript-api-for-office.

[16] Microsoft, ".NET framework," Microsoft, 30 07 2019. [Online]. Available: https://dotnet.microsoft.com/download/dotnet-framework.

[17] Ecma International, "https://www.javascript.com/," 30 07 2019. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[18] Google and MIT, "Angular," Google, 30 07 2019. [Online]. Available: https://angular.io/.

[19] Microsoft, "TypeScript Language," Microsoft, 30 07 2019. [Online]. Available: https://www.typescriptlang.org/.

[20] Bootstrap team and MIT, "Bootstrap," Bootstrap team and MIT, 30 07 2019. [Online]. Available: https://getbootstrap.com/.

[21] Encode OSS Ltd, "Django REST framework," Encode OSS Ltd, 30 07 2019. [Online]. Available: https://www.django-rest-framework.org/.

[22] Oracle, "MySQL," Oracle, 30 07 2019. [Online]. Available: https://www.mysql.com/.

[23] S. Morris, "Inner-browsing extending the browser navigation paradigm," Netscape, 2003. [Online]. Available: https://developer.mozilla.org/en-US/docs/Archive/Inner-browsing_extending_the_browser_navigation_paradigm.

[24] T. Koppers, "webpack," 30 08 2019. [Online]. Available: https://webpack.js.org/.

[25] B. Thalheim, "The Theory of Conceptual Modelling and Foundations of Conceptual Modelling," in *Handbook of Conceptual Modeling*, Berlin, Springer, 2011, pp. 543-577.

[26] P. Sternudd, "Unambiguous requirements in Functional Safety and ISO 26262: dream or reality?," Uppsala University, 2011, pp. 83-87.

[27] A. Cockburn, Writing Effective Use Cases, Addison-Wesley Longman, 2000.

[28] ISO, "ISO/IEC 25010: SQuaRE System and software quality models," ISO, Geneva, 2011.

[29] OPEN Process Framework Repository Organization (OPFRO), "OPEN Process Framework Repository Organization (OPFRO)," OPFRO, 05 03 2006. [Online]. Available: http://www.opfro.org/Components/WorkProducts/ArchitectureSet/Architectures/Architectures.html.

[30] The Open Group, "Open Group," IBM, 28 08 2019. [Online]. Available: http://www.opengroup.org/public/arch/p4/patterns/patterns.htm.

14. Appendices

15. Appendix A

This appendix describes all KISS method concepts [11] that are used in this project in Table 17.

Table 17. KISS concepts

Concept	Definition
Action model	It comes from the definitive action, is analyzed by the end-user.
Candidate action	A verb of the structured sentence can be considered as a candidate action
Candidate object	An object of the structured sentence can be considered as a candidate object
Candidate subject	A subject of the structured sentence can be considered as a candidate subject
Definitive action	It comes from the candidate action, is reviewed by the end-user.
Definitive object	It comes from the candidate object, is reviewed by the end-user.
Definitive subject	It comes from the candidate subject, is reviewed by the end-user.
Object model	It comes from the definitive object, is analyzed by the end-user.
Original text	It can be a small paragraph, a single sentence, or multiple sentences of a specific report, article, and standard.
Structured sentence	The structured sentence needs to be in a simple tense. It can include an adjective, adverb, preposition, etc. However, a structured sentence cannot be complex.
Subject model	It comes from the definitive subject, is analyzed by the end-user.

16. Appendix B

This appendix presents a definition table for terms of the requirement specifications.

Table 18. Definition table for terms of the requirement specifications

Artifact Name	The document name which is opened by the MS Word application
Log ID	It is a unique ID for each modeling process, and it consists of Artifact Name, Section Id, and the other random number.
Metadata	
Original text	See Table 17, Appendix A.
Predefined term	It is the specific terms that are defined inside of the ISO standard. For example, ISO 26262 part 1 (Vocabulary) introduces all the words which are used for the rest of the parts of the ISO standard.
Processed sentence	It is a small paragraph or sentence from the MS Word document that is inputted to the original text on the MMA tool.
Section Id	Specific requirement Id of the ISO standard
Simplexity	An application has been developed in the TNO IVS department.
Structured sentence	See Table 17, Appendix A.

17. Appendix C

This appendix shows detailed requirement justification regarding justification criteria, namely value, risk, and difficulty.

Table 19. Detailed requirement justification

REQ ID	Requirement	Value	Risk	Difficulty	Priority
1.2.0	The tool shall be able to show its Graphic User Interface (GUI) in the MS Word application via a shared add-in.	Important	High	Complex	Must
1.2.1	The tool shall be able to highlight the *processed sentence* in the MS Word document automatically.	Important	High	Complex	Should
1.2.2	The tool shall be able to select the next sentence from the MS Word document and take it to the *original text.*	Important	High	Complex	Must
1.2.3	The tool shall be able to copy the selected sentence from the MS Word document to the *original text* automatically.	Important	Medium	Complex	Should
1.4.1	The tool shall be able to show the step the Safety engineer is working on.	Vital	High	Complex	Must
1.4.2	The tool shall be able to show a *predefined term* from the *original text* and *structured sentence.*	Important	Medium	Moderate	Should
1.4.3	The tool shall be able to accept multiple *original texts*.	Vital	Medium	Moderate	Must
1.4.4	The tool shall be able to accept multiple *structured sentences* for a single *original text.*	Vital	High	Moderate	Must
2.1.1	The tool shall be able to save the KISS concepts, namely original text, structured sentence, candidates object, subject, and action depends on the modeling step.	Optional	Low	Complex	Could
2.1.2	The tool shall be able to generate *Log ID* based on the *metadata.*	Vital	Medium	Simple	Must

2.1.3	The tool shall be able to show the *Log ID.*	Vital	Medium	Simple	Should
1.1.1	The proposed process shall have four steps: Preprocess, Parsing, Analysis, and Initial modeling.	Vital	High	Complex	Must
1.1.2	The Preprocess step might have a sub-step.	Important	Medium	Complex	Should
1.1.3	The tool shall be able to offer the start function of the modeling process.	Vital	High	Moderate	Must
1.1.4	The Safety engineer should be able to give *metadata: artifact name* and *section Id* for the *Log ID.*	Important	Medium	Moderate	Should
1.1.8	The tool shall be able to show a list of the *structured sentences* with their *status.*	Vital	High	Simple	Must
1.1.10	The tool shall be able to extract *candidate objects*, *candidate subjects*, and *candidate actions* from the *structured sentence* on the tool based on the safety engineer input.	Vital	High	Complex	Must
1.1.11	The tool shall be able to show the *candidate object*, *candidate subject*, and *candidate action* in the *structured sentence.*	Vital	High	Moderate	Must
1.1.12	The tool shall be able to convert the *candidate object*, *candidate subject*, and *candidate action* to *definitive object*, *definitive subject*, and *definitive action* respectively.	Vital	High	Moderate	Must
1.1.14	The safety engineer shall able to add the definition for a *definitive object*, *definitive subject*, and *definitive action.*	Vital	High	Moderate	Must
1.3.1	The tool shall be able to assign a type (*Class*, *Abstract Class*, and *Attribute*) for the *object* and *subject model* based on the end-user selection.	Important	High	Moderate	Should
1.3.2	The tool shall be able to assign a type (*Associate*, *Generalize*, and	Important	High	Moderate	Should

	Compose) for the *action model* based on the end-user selection				
1.3.4	The tool shall be able to show *the start point* and *the endpoint to the action model*.	Important	High	Moderate	Should
1.3.5	The tool shall be able to export the *initial models* to *Simplexity*.	Important	High	Moderate	Must

18. Appendix D

This appendix describes Natural Language Processing tasks that are studied and the result of the study.

18.1 Natural Language Processing Tasks

At the beginning of this project, several Natural Language Processing (NLP) tasks are studied. The purpose of this study to support the proposed solution by reducing the modeling time. The main question is "How NLP task can help to speed up the proposed process for conceptual modeling?" The study and experiment are addressed this question and are focused on the usage of NLP tasks for conceptual modeling.

NLP [12] is a broad subject. There are various NLP tasks such as Stemming, Lemmatization, Word Embedding's, Part-of-Speech Tagging (POS), Named Entity Disambiguation, Named Entity Recognition (NER), Sentiment Analysis, Semantic Text Similarity, Language Identification, and Text Summarizations for different usages. The common usage of the NLP is high-level applications such as language translation, question answering systems, chatbots, requirement automation, and document content summarizers.

The question is mostly related to requirement automation. Stemming, Part-of-Speech Tagging, Named Entity Recognition, and Sentiment Analysis tasks are studied.

18.2 Fundamental NLP tasks

Table 20 **Error! Reference source not found.**describes the fundamental tasks of the NLP with the decision is made.

Table 20. Fundamental NLP tasks

NLP task name	Description	Example	Decision
Stemming	Stemming is the process of reducing the words (generally modified or derived) to their word stem or root form. The objective of stemming is to reduce related words to the same stem even if the stem is not a dictionary word.	The words 'Beautiful' and 'Beautifully' have the same root form, 'beauty.'	This technique mainly used for raw data. But the ISO standard or norms are precisely described. There was no need to apply this technique.
Part-Of-Speech Tagging	Part-Of-Speech Tagging (POST) is the process of making up of words in a sentence as nouns, verbs, adjectives, adverbs, etc.	"The boy killed the snake with a stick." The result of the POST: 'the' – DET, 'boy' – PROPN, 'killed' -VERB, 'snake' – NOUN, 'with' –ADP 'a' – DET 'stick' - NOUN	We experimented on it, and the result is seen in **Error! Reference source not found.**.

		.- PUNCT	
Named Entity Recognition	In NLP, Named Entity Recognition (NER) is the task of determining the identification of entities mentioned in the text.	"President Putin of Russia, on his first state visit to the United States on Tuesday night." The result of the NMR: 'Putin' – Person, 'Russia' – Location, 'First' - Ordinal 'United States'- Location, 'Tuesday' - Date 'Night' - Time	This task was essential to identify the predefined terms. However, We were not able to do any experiment due to lack of data because we had only the ISO 26262 standards that include 12 files.
Sentiment Analysis	Sentiment Analysis is a broad range of subject analysis that uses NLP techniques to perform tasks such as identifying the sentiment of a customer review, positive or negative feeling in a sentence, or judging mood via voice analysis or written text analysis.	"I did not like the chocolate ice-cream " – is a negative experience of ice-cream. "I did not hate the chocolate ice-cream" – may be considered as a neutral experience	No need to study this technique because the standards and norms do not include any emotional information.

18.3 NLP task experiment

POST and NER tasks were considered useful for the proposed process. We experimented with the usage of the POST task. Figure 52 shows the results and the input of the experiment. The input was one sentence from the ISO standard 26262, and the result is given in Language Tree Structure. In this experiment, we used the Stanford NLP core application [13].

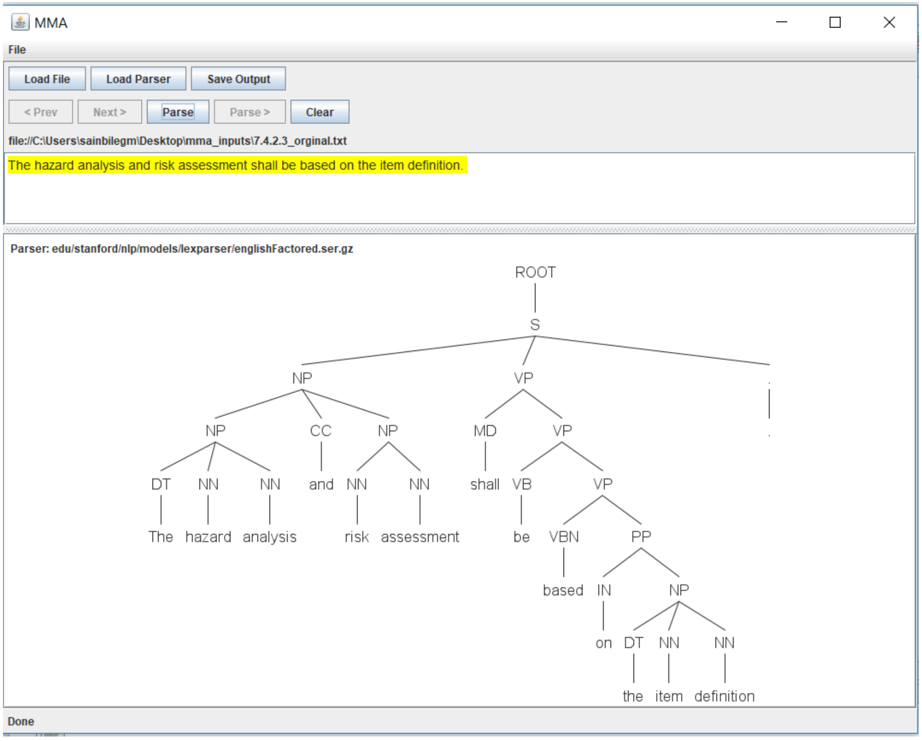


Figure 52. POST extraction

The input sentence was *The hazard analysis and risk assessment shall be based on the item definition*. Stanford NLP core application extracted this sentence to 'the,' 'hazard,' 'analysis,' 'risk,' 'assessment,' 'shall,' 'be,' 'based,' 'on,' 'the,' 'item,' and 'definition.' The safety engineer extracted this sentence to 'The hazard analysis and risk assessment,' 'based on,' and 'the item definition' parts. Therefore, the POST task was not useful for the ISO standard conceptual modeling due to the degree of interpretation from the safety engineer.

We were not able to do any experiment on the NER task. The NER task relies on supervised machine learning, and there were not have a sufficient amount of data for supervised machine learning in this project. For example, ISO 26262 consists of 12 parts, and it has approximately 750 pages. This data can be used as a classification or label of supervised machine learning. However, this data can't be used as a training data set.

18.4 Decisions on the NLP tasks

We completed the study and experiment with NLP tasks regarding project planning. After the short time of NLP study and experimentation, it was hard to give a concrete answer to our original question, which is asked at the beginning of this section. However, the scope of this project's NLP tasks could not help to speed up the proposed process. There were two main reasons; the first reason was the lack of data for the NER task. The second reason was that the extraction process from a sentence to words highly depends on the content of the ISO standards or the norms. The content of the ISO standard and the norms required a significant degree of interpretation from the safety engineer. In that situation, the POST task was not able to accelerate the proposed process.

Where innovation starts