TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

PETRI TIKKA

REQUIREMENTS TRACEABILITY IN SIMULATION DRIVEN MECHANICAL ENGINEERING

Master of Science thesis

Examiner: prof. Asko Ellman
Examiner and topic approved by the Faculty Council of the Faculty of Mechanical Engineering and Industrial Systems
on 9th December 2015

ABSTRACT

PETRI TIKKA: Requirements traceability in simulation driven mechanical engineering
Tampere University of technology
Master of Science Thesis, 82 pages, 11 Appendix pages
May 2016
Master's Degree Programme in Automation Engineering
Major: Product Development
Examiner: Professor Asko Ellman

Keywords: Requirements, Traceability, Systems Engineering, Simulation, Mechanical engineering, Impact analysis, Surrogate object method

Requirements traceability is an essential part of the product development process as all development work is based on requirements. When the amount of requirements increases so does the difficulty to notice changes in them and in other related engineering artefacts. To support the design process computer simulation and analysis were introduced to the discipline of requirements engineering. Simulation provides means to verify and validate engineering artefacts, which in this thesis was studied in the context of requirements traceability from stakeholder requirements to design and verification in a mechanical engineering case. Providing sufficient traceability also required version control and impact analysis to trace the impact of changes in the artefacts.

Relevant artefacts and their relations were defined by Systems Engineering Artefacts Model (SEAModel), which supports mechanical system design and simulation. SEAModel was transformed into a traceability information model (TIM) to form traceability links between artefacts. TIM was used to create a case dependent traceability demonstration model to depict the traceability chain in the case related environment. To implement traceability and impact analysis according to these models, a database oriented software platform is required. This thesis introduced an integration and traceability platform (ITP) composed of IBM Rational DOORS for an environment to trace artefacts and Subversion version control software (SVN) for version management. Traced artefacts were produced with IBM Rational DOORS, Papyrys SysML, SolidWorks and MATLAB. With this heterogeneous set of state-of-the-art software applications a logical architecture model was created to represent the mechanical structure of the machine depicted in the case. According to the logical model a detailed CAD model was updated to fit the new stakeholder requirements. Requirements were stored and managed in a requirements management tool. Later the CAD model was verified and validated with a simulation model. The integration of engineering artefacts was accomplished by adopting the so called surrogate object method, in which model files were represented as surrogate objects within the ITP.

The results of this thesis implicated that requirements engineering can be extended to cover simulation artefacts with SEAModel. Impact analysis and traceability were able to be combined with a tailored solution of the surrogate object method. Although optimal granularity and visibility of all data in all tools could not be achieved, a file level granularity of model elements was met with satisfactory results.

TIIVISTELMÄ

PETRI TIKKA: Vaatimusten jäljittäminen simulaatio-orientoituneessa mekaanisten järjestelmien suunnittelussa
Tampereen teknillinen yliopisto
Diplomityö, 82 sivua, 11 liitesivua
Toukokuu 2016
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Tuotekehitys
Tarkastaja: Professori Asko Ellman

Avainsanat: Vaatimukset, Jäljittäminen, Systeemitekniikka, Simulointi, Mekaaninen suunnittelu, Vaikutusanalyysi, Surrogaattiobjektimenetelmä

Vaatimusten jäljittäminen on oleellinen osa tuotekehitysprosessia, sillä siihen liittyvä suunnittelutyö pohjautuu vaatimuksiin. Vaatimusten määrän kasvaessa, muutosten hallinta vaikeutuu sekä vaatimusten että muiden asiaankuuluvien artefaktien tapauksessa. Suunnitteluprosessin tukemiseksi tietokonesimulaatio ja -analyysi esiteltiin vaatimusmäärittelyn yhteydessä. Simulaatio mahdollistaa artefaktien todentamisen (verification) ja kelpuuttamisen (validation). Tässä työssä simulaatiota hyödynnettiin vaatimusten jäljittämiseen, sidosryhmävaatimuksista suunnitteluun ja todentamiseen mekaaniseen suunnitteluun liittyvässä casessa. Tyydyttävä jäljitettävyyden tuottaminen vaatii myös versionhallintaa, sekä muutosten vaikutusten jäljittämistä vaikutusanalyysillä.

Työssä käytetyt artefaktit ja niiden väliset suhteet määriteltiin Systems Engineering Artefacts Mallilla (SEAModel), joka tukee mekaanista järjestelmä suunnittelua ja simulointia. SEAMalli muutettiin tiedonjäljitettävyysmalliksi (TIM) kuvaamaan artefaktien välisiä jäljitettävyyslinkkejä. TIMstä luotiin case kohtainen jäljitettävyydendemonstrointimalli kuvaamaan jäljitettävyysketjua caseen liittyvässä ympäristössä. Jäljitettävyyden ja vaikutusanalyysin toteuttaminen edellä mainittujen mallien avulla vaatii tietokantapohjaisen sovellusalustan. Työssä hyödynnetty integrointi- ja jäljitettävyysalusta (ITP) koostui artefaktien jäljitettävyysympäristönä sovelletusta IBM Rational DOORSista, sekä versionhallintatyökalusta SVN. Jäljitettävät artefaktit tuotettiin IBM Rational DOORS:lla, Papyrus SysML:llä, SolidWorks:llä ja MATLAB:lla. Tällä heterogeenisellä ohjelmistojoukolla luotiin looginen arkkitehtuurimalli kuvaamaan casessa esitellyn koneen mekaanista rakennetta. Loogisen mallin perusteella päivitettiin yksityiskohtainen CAD-malli vastaamaan uusia sidosryhmävaatimuksia. Myöhemmin CAD-malli todennettiin ja kelpuutettiin simulaatiomallin avulla. Artefaktien integroiminen saavutettiin käyttämällä ns. surrogaattiobjektimenetelmää, jolla mallitiedostot esitettiin surrogaattiobjekteina ITP:ssä.

Työn tulokset osoittivat, että vaatimusmäärittely voidaan laajentaa sisällyttämään simulaatioartefakteja SEAMallin avulla. Artefaktien jäljittäminen ja vaikutusanalyysi onnistuttiin yhdistämään räätälöidyllä surrogaattiobjektimenetelmällä. Vaikka optimaalista granulaarisuutta ja tiedon näkyvyyttä kaikkien työkalujen yhteydessä ei saavutettu, malli-elementtien granulaarisuus tiedostotasolla onnistuttiin luomaan tyydyttävästi.

PREFACE

This thesis was done for the Department of Mechanical Engineering and Industrial Systems at Tampere University of Technology (TUT). The thesis addresses requirements traceability in a simulation driven mechanical engineering case study. The case study relates to a computational methods project by SIMPRO project group. The work was commissioned by VTT.

I would like to thank VTT for providing the thesis subject and the SIMPRO project group for their support and induction into the field of Systems Engineering. Juan Sagarduy and Risto Kause of Mathworks and Mika Hyvönen of Tampere University of Technology I would like to thank for their invaluable guidance and advises concerning the simulation work of the thesis. Thesis supervisor's Senior scientist Jarmo Alanen of VTT and Professor Asko Ellman of Tampere University of Technology I thank for their professional input on the subject and thorough guidance throughout the project.

Tampere, 24.3.2016

Petri Tikka

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

BDD	Block Definition Diagram
CAD	Computer Aided Design
CM	Configuration Management
DoF	Degree of Freedom
DTP	Digital Product
DXL	Rational DOORS eXtension Language
ECM	Engineering Change Management
HTP	High-Performance-Computing
ICT	Information and Communication Technology
ITP	Integration and Traceability Platform
MEWP	Mobile Elevating Working Platform
PECS	Programmable Electric Systems
PLM	Product Lifecycle Management
SEAModel	System Engineering Artefacts Model
SIMPRO	Computational Methods in Mechanical Engineering Product Development
SysML	Systems Modelling Language
SVN	Subversion version control tool
TIM	Traceability Information Model
TIKOSU	Database oriented development of machine control systems
TTS	Theory of Technical Systems
TUT	Tampere University of Technology
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
VTT	Technical research centre of Finland
V&V	Verification and Validation
XML	Extensible Markup Language

E	energy
E_f	effects
I	information
M	material
M_{ch}	vertical moment of chassis
M_e	vertical moment of equipment
M_{es}	vertical moment of extending structure
M_p	vertical moment of persons
m_p	mass of a person
M_s	horizontal moment of manual force
M_{sf}	vertical moment of (scissors) sub frame
M_{we}	horizontal moment of equipment
M_{wp}	horizontal moment of persons
M_{ws}	vertical moment of work surface
$M_{ws\_sf\_es}$	combined horizontal moment of work surface, sub frame and extending structure
n	number of persons
O_d	operands
O_p	operations

1. INTRODUCTION

Product development can be described as an iterative process of joining solutions from different disciplines with specific requirements. Part of the process is changes of the requirements. Effects of the changes in the specification of the product properties may be broad, and the impact of these changes is not always easy to notice. Disarray is especially prominent in mechanical industry where traceability of requirements is not often realised.

Tracing a change from an engineering artefact to its corresponding requirement is a challenging task as different artefacts are typically scattered to separate data repositories. Furthermore, requirements engineering is often done separately from the other design work and merely seen as an initiator without having much of an impact later on. The lack of consistency between requirements and various engineering artefacts, as they evolve during the project, may lead to inaccurately produced impact analyses and give false image of the conditions of the project. Without knowing the original stakeholder requirements, the consequent system requirements and the rationale of a specific design task, the verification and validation (V&V) of engineering artefacts flounder. If traceability management is seen as a tedious and time consuming task, creation and maintenance of the artefact traces may be neglected; this results in failures.

This thesis explores a data model, created outside of the thesis, for capturing dependencies between engineering artefacts and tracing impacts of occurring changes. The data model, titled Systems Engineering Artefacts Model (SEAModel), integrates engineering artefacts from requirements management, design process, modelling, simulation and document generation. The focus is on achieving traceability of engineering artefacts in a simulation driven mechanical engineering systems development case by implementing this systematic data model for the artefacts. The engineering artefacts are created in a generic design process, which requires a study of feasible tools. These tools are at the centre point of this thesis as they manifest traceability and impact analysis among engineering artefacts such as requirements specifications, system functions specifications, system architecture descriptions and verification and validation artefacts, simulation related artefacts being a part of the verification and validation artefacts.

The main goals are to study whether a heterogeneous set of engineering tools support SEAModel, and on what granularity level the traceability chain can be created to support impact analysis. The compatibility of engineering tools in relation to each other and to SEAModel is crucial to make traceability management a more convenient practice and possibly more integrated activity of the development work. Additionally, challenges of traceability are noted and suggestions for optimal workflows as well as tools are studied.

1.1 Description of the background work

The thesis relates to a wider research project coordinated by VTT (Technical research centre of Finland) pertaining to computational methods in mechanical engineering product development, short for SIMPRO. SIMPRO research project studies how computational methods and tools could provide keys to success in engineering product process, and how computational approach in product development could provide real advantage in the markets. Computational methods, applied during the whole lifecycle of a product, could result in cheaper and better planned products. By applying computer simulations and analysis from the beginning of the development process, more information could be given to support the design, and the outcomes of the design could be validated. SIMPRO project was carried out during the years 2012 – 2015 and financed by organizations: Tekes (the Finnish Funding Agency for Innovation), Aalto University, Tampere University of Technology, Lappeenranta University of Technology, University of Jyväskylä, Wärtsilä Finland Oy, Patria Land Systems Oy, Kone Oyj, MeVEA Oy, FS Dynamics Finland Oy Ab, EDR & Medeso Oy, Dassault Systemes Oy, Techila Technologies Oy and VTT.

SIMPRO project is divided into four collaborating subtasks of which this thesis contributes to the task three; the requirement- and customer-based product development in simulation driven development process. The significance of the simulation is that it offers means to verify and validate engineering artefacts, such as requirements. Other three subtasks were HPC (High-Performance-Computing) in mechanical engineering, Optimisation, design studies and analysis, and Product lifecycle, and modelling and results data management. Subtasks are presented in Figure 1.1.

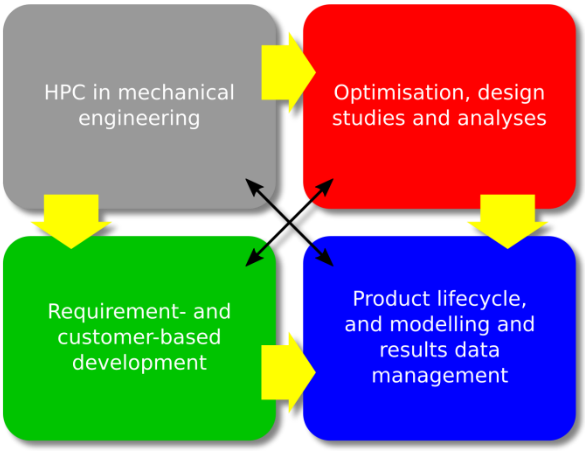


Figure 1.1. *The four subtasks of the SIMPRO project (VTT, 2015).*

Task three consists of five sub-tasks: fixing the data model for simulation based system development, capturing and transferring requirements, tracing requirements, system models and system simulation, networking and dissemination and final reporting and task management. Task three involved the use of TIKOSU model created by Tekes DTP (Digital Product) programme. TIKOSU comes from database oriented development of machine control systems, and it was used as the base model for the current SEAModel. SEAModel is a continuous data model that supports mechanical system design and simulation. It integrates together requirement management, design process, simulations, virtual prototyping and document generation. This model will set guidelines for a design process and it will connect simulations and virtual prototyping to requirements obtained from stakeholders.

In addition to SEAModel a traceability information model (TIM) is required to enable artefact relations support and impact analysis. This thesis utilises TIM generated outside of the thesis by VTT. TIM is a data model based on the principles of SEAModel. TIM studies and demonstrates the traceability of system models to system requirements, simulation models to system models, and verification results to simulation results. Engineering artefacts and the traceability chain between them are depicted in a traceability demonstration model, which is a case dependent process model. Traceability demonstration model is derived from TIM. To implement SEAModel and TIM, a database oriented software platform is required. Platform utilised in this thesis is called an integration and traceability platform (ITP). The general architecture of an ITP is depicted in Figure 1.2.

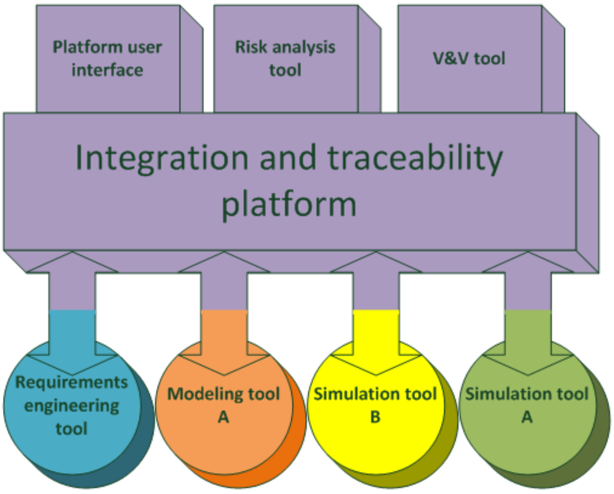


Figure 1.2. *General architecture of TIM implementation platform (ITP) (Alanen, 2015).*

Different engineering tools in Figure 1.2 are used to create artefacts that the ITP arranges to one common location and traces to each other. In the demonstration case, introduced later in Chapter 3, the ITP to be used is constructed of Subversion version control tool (SVN) and IBM Rational DOORS. SVN enables the control of version handling and the ability to distribute up-to-date files between relevant stakeholders without format restrictions. The integration of software is considered to be accomplished by adopting the so called surrogate object method, in which model files are represented as surrogate objects within DOORS. In addition to be used as an artefacts traceability environment, DOORS is also used as a requirements management tool. Other considered engineering tools to create traceable artefacts include Papyrus SysML tool for modelling the physical architecture, SolidWorks for creating the mechanical CAD model and MATLAB’s Simulink/SimMechanics for the simulation model.

1.2 Milestones to be achieved

The background work was accomplished by VTT and it included the collection and listing of stakeholder and system requirements (see Figure 4.7). Furthermore SEAModel, TIM and feature requests were defined and realised beforehand without the concern of this thesis. Feature requests are presented in Chapter 4 (see Figures 4.2, 4.6, 4.8, 4.17, 4.21, 4.22, 4.23). They explain and define the premeditated workflow that also set the milestones to be achieved in this thesis. Figure 1.3 depicts this workflow on a rough level indicating models and processes done outside of this thesis with green colour and milestones still to be achieved with red colour.

The case study included sketching a logical model representing the mechanical architecture of a new feature (see Section 4.4.1). The feature is required to be added to the case specific product by a new feature request. The product is presented in detail in Section 4.1. The functionality of the mechanical architecture is explained with use case diagrams (see Section 4.4.2). The physical structure represented in the CAD model is updated according to the inputs of the logical model (see Section 4.5).

The physical structure is ported to DOORS. Artefacts and traceability links are copied to DOORS from the traceability demonstration model after it is updated accordingly. Additionally DOORS is updated to fully represent the current contents of the traceability demonstration model with proper formal and link module representations of the corresponding artefacts. Also the attributes of the artefacts are updated into DOORS. The updating of DOORS is not documented in this thesis as it was considered irrelevant for the scope. Instead requirements management is discussed in Section 4.3.1. With proper artefacts and traceability links in place, the artefacts traceability environment is formulated into DOORS.

A simulation model is created from the updated CAD model to validate the stability of the mechanical structure (see Section 4.6). Loss of stability was identified as a potential hazard during the risk assessment, which was done outside of this thesis. Simulation results are checked with a separate statics analysis (see Section 4.6.2)

To connect the products of different engineering tools with one another and with requirements in DOORS a surrogate object method is studied (see Sections 4.7.1 – 4.7.3). The study focuses on the integration of MATLAB and DOORS with the scope being in tracing simulation results to the requirements. In addition to tracing, the surrogate object method should also support impact analysis. After a concept for surrogate object method is validated the implementation is applied between products of other chosen engineering tools and DOORS (dashed lines in Figure 1.3 represent the relation of the original model and the derived surrogate object). Surrogate objects and modules complete the artefacts traceability environment in DOORS.

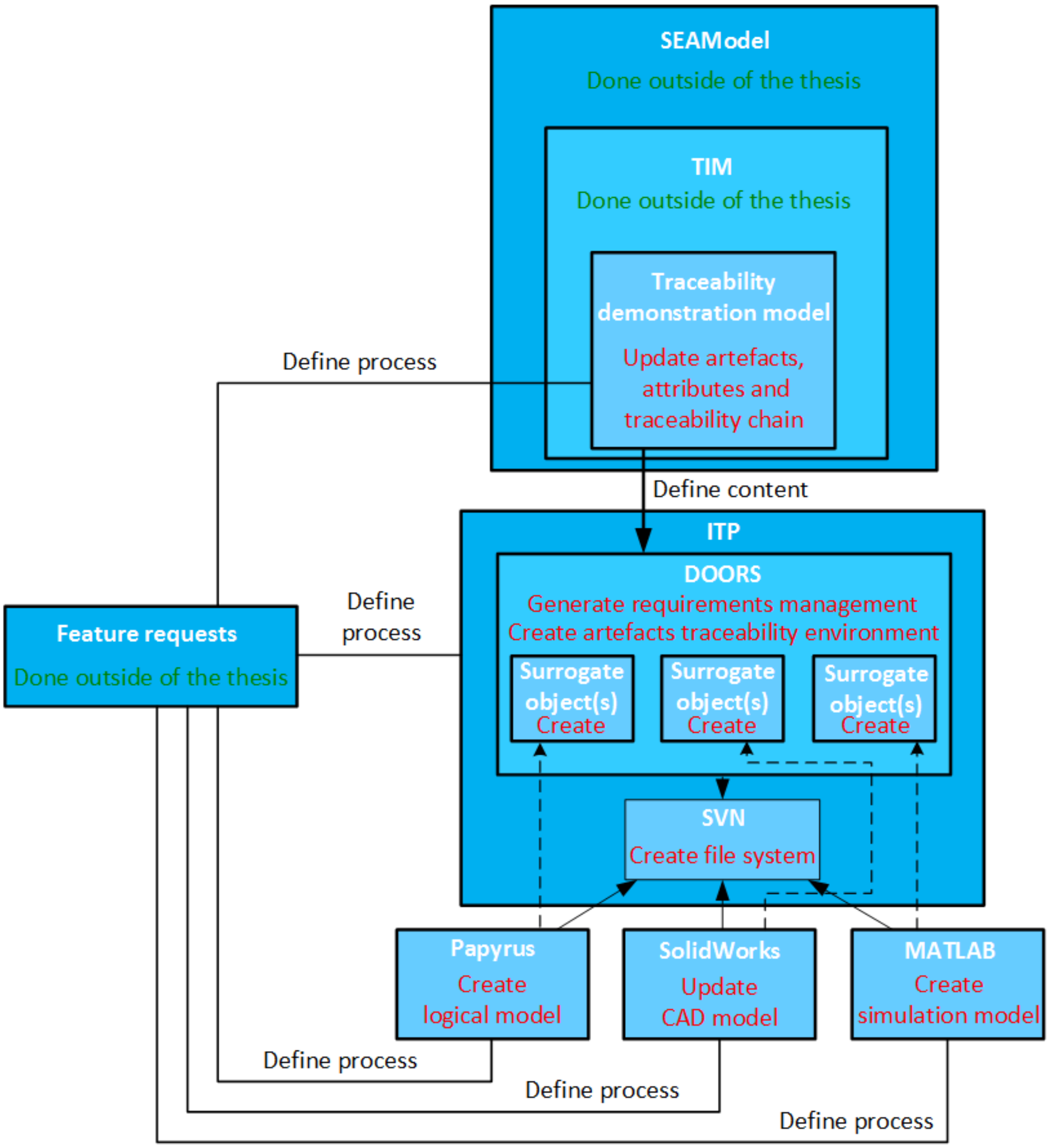


Figure 1.3. *A rough description of the thesis workflow.*

All the files are saved into SVN, which together with DOORS form the ITP used in the thesis. Therefore, to have the files logically organised and easily accessed by relevant stakeholders a file system is generated into SVN.

An iteration process is developed for testing the requirements traceability (see Section 4.8). The process includes changing a specific model or requirement and observing how the traceability and impact analysis functions in DOORS. The results of the iteration are reflected against the research questions.

1.3 Identification of the Problem and Objectives

Task of tracing requirements in the context of complex products is challenging. This is especially true in the mechanical industry where traceability of engineering artefacts is not often realized. Consequences of product change for its structure are not only difficult to predict, but also their propagation to the overall system is as well. The before mentioned challenges bring forward the challenge of creating TIM that is described more thoroughly in Section 3.2.

The aim of this thesis can be expressed with two main objectives that are derived from the task 3 of the SIMPRO project. The nature of the first objective is to verify the applicability of SEAModel in simulation driven machine design by testing the model in a case study. Through testing and demonstration it is intended to evaluate SEAModel. The outcome of this objective will influence on the acceptance of the considered engineering tools as well as their proficiency to support SEAModel. The chosen tools should support traceability and impact analysis. The second objective is to evaluate the feasibility and practicality of implementing SEAModel while using typical commercial mechanical engineering tools.

1.4 Research questions

The objectives are evaluated according to criteria which are derived from the SIMPRO project and the experience of researchers in VTT. These criteria can be seen as research questions that answer to the defined objectives. The research questions provide a guide path for the writing process and clarify the research objectives.

The first objective includes research questions:

1. Can impact analysis be made non-manually from requirements to design and to V&V through system subsystem hierarchy?
2. Does SEAModel provide an assurance case required by the machinery directing annex SFS-EN ISO 12100 (SFS-EN ISO 12100, 2010)?
3. Does SEAModel support traceability of requirements over organisational borders?

The first research question relates to studying how impact analysis can be achieved through the traceability demonstration model hierarchy. Traceability demonstration model consists of three levels. The second question relates to an assurance case, which is an evidence-based argument to support claims that satisfy specific requirements. Assurance case discloses a system provides claims such as safety and reliability. The third question relates to how traceability of artefacts can be achieved when artefacts are located in different organisations.

The second objective is addressed using the research questions:

4. Can a full traceability chain be achieved with model element level granularity?
5. Can the traceability be demonstrated with a heterogeneous set of engineering tools?
6. What are the core features of the mechanical engineering tools enabling requirements tracing according to SEAModel?
7. Is SEAModel based workflow unacceptably complex to be used by machine designers?
8. Does SEAModel ontology help to cut development costs while increasing the quality of the safety engineering?

The fourth research question studies the achievable granularity level of the traceability chain that supports impact analysis. Granularity here means divergence of information that is being traced. For example a simulation model consists of many model elements implicating a large granularity. The fifth question studies how traceability can be achieved in a heterogeneous development environment with a diverge set of engineering tools made by different manufacturers. The sixth question relates to the qualities that are collectively required from the engineering tools in order to support SEAModel based requirements engineering. The seventh question studies how difficult it is to apply SEAModel into mechanical industry. Finally the eighth question relates to studying how systematic approach could affect the costs of the system. Cutting the development costs, while increasing the quality of safety engineering with SEAModel, is studied only on a superficial level.

Some of the research questions are answered indirectly during the case demonstration in Chapter 4 as the demonstration case progresses further. Because all the research questions relate to the evaluation of the complete traceability chain and implementation of SEAModel, answers are mostly formulated at the end of the case study from scattered pieces of information. Answers are conducted from practical work while different engineering tools are utilised to support the fulfilment of SEAModel. The results of the demonstration are allocated to these questions, and they are brought forward in Chapter 5; Analysis of the results.

1.5 Scope and Depth

The coverage of this thesis is identified through the needs of the SIMPRO project team. Therefore, the scope of the research was chosen to compliment the tools used for the tracing of engineering artefacts in a simulation driven systems development case study. Iterative systems development is managed with properly arranged traceability and impact analysis, which are considered highly in the scope. Tools that provide management of structures and relational data, requirements management, modelling, simulation and traceability were considered beforehand by the SIMPRO research group leaving the applicability and implementation of these tools to concern this thesis.

Collaboration of the chosen tools is studied according to the artefacts model, SEAModel, which is involved with computational methods and systematic data management. Therefore this thesis pertains to Systems Thinking and Systems Engineering. These two theory approaches provide the base for the traceability demonstration case, which concentrates on technical processes. From technical processes the case study is chosen to concern the mechanical system level, leaving electrical systems and programmable electronic systems (PECS) out of the scope. This limitation came from the SIMPRO project group, and was determined to be the best solution for the study of simulation driven mechanical engineering in the context of a case study.

The traceability demonstration model pertains to a process model. The most significant design processes to provide a sufficient process model were considered to be the property-driven development, the theory of dispositions, and the system life cycle processes. Implementation of the demonstration case was studied among these processes from the perspective of design workflow. Sufficient design workflow is especially important in the requirements traceability. Meanwhile the design completeness was not regarded as highly.

Requirements tracing conducted in the demonstration case is limited to a single customer feature request. Study of a full set of requirements is seen unnecessary as this thesis concentrates only on the main corresponding requirements including a sample of safety requirements.

1.6 Structure of the thesis

This thesis is carried out as a constructive research with aim to produce a traceability chain that can support impact analysis by studying the chosen engineering tools. Structure and usability of chosen engineering tools are studied in practice and through related literature i.e., software manuals. Furthermore, the people of MathWorks and the department of mechanical engineering and industrial systems of Tampere University of Technology (TUT) are consulted for guidance regarding the simulation model. Solutions to establish traceability and impact analysis with chosen software are familiarized with the help of SIMPRO project researchers and by looking into already implemented concepts regarding the use of similar tools. Implementations involve the use of plug-ins between different software interfaces that aid software to collaborate. If there is an idea or an achievement brought up without a source, they are produced either by the author himself or as a product of conversations with SIMPRO project researchers. Also, this thesis is done in close collaboration with a VTT publication on Requirements traceability in simulation driven development (Alanen, et al., 2015).

Figure 1.4 depicts the structure of the thesis. Chapter 1 is dedicated to the introduction where the subject and contents of this thesis are presented in a curtly manner. The background of the study is revealed as are the research questions and the derived scope and depth. Chapters inside the dashed line are divided into theoretical and practical part of the study. Practical part, which includes the study of tracing requirements which again is implemented according to the architecture of the developed traceability demonstration model, is outlined by the dash-and-dot line.

Chapter 2 revolves around the theoretical concepts of technical systems and how they involve the subject of traceability. A special look is taken into Systems Thinking and technical processes it has led into. Theoretical background of technical systems is the preface for the Chapter 3 where the foundation is built for the traceability information system by exhibiting the developed data models (SEAModel and TIM) and process model (traceability demonstration model). These models show how the requirements traceability is meant to be executed and what it takes to build a traceability chain.

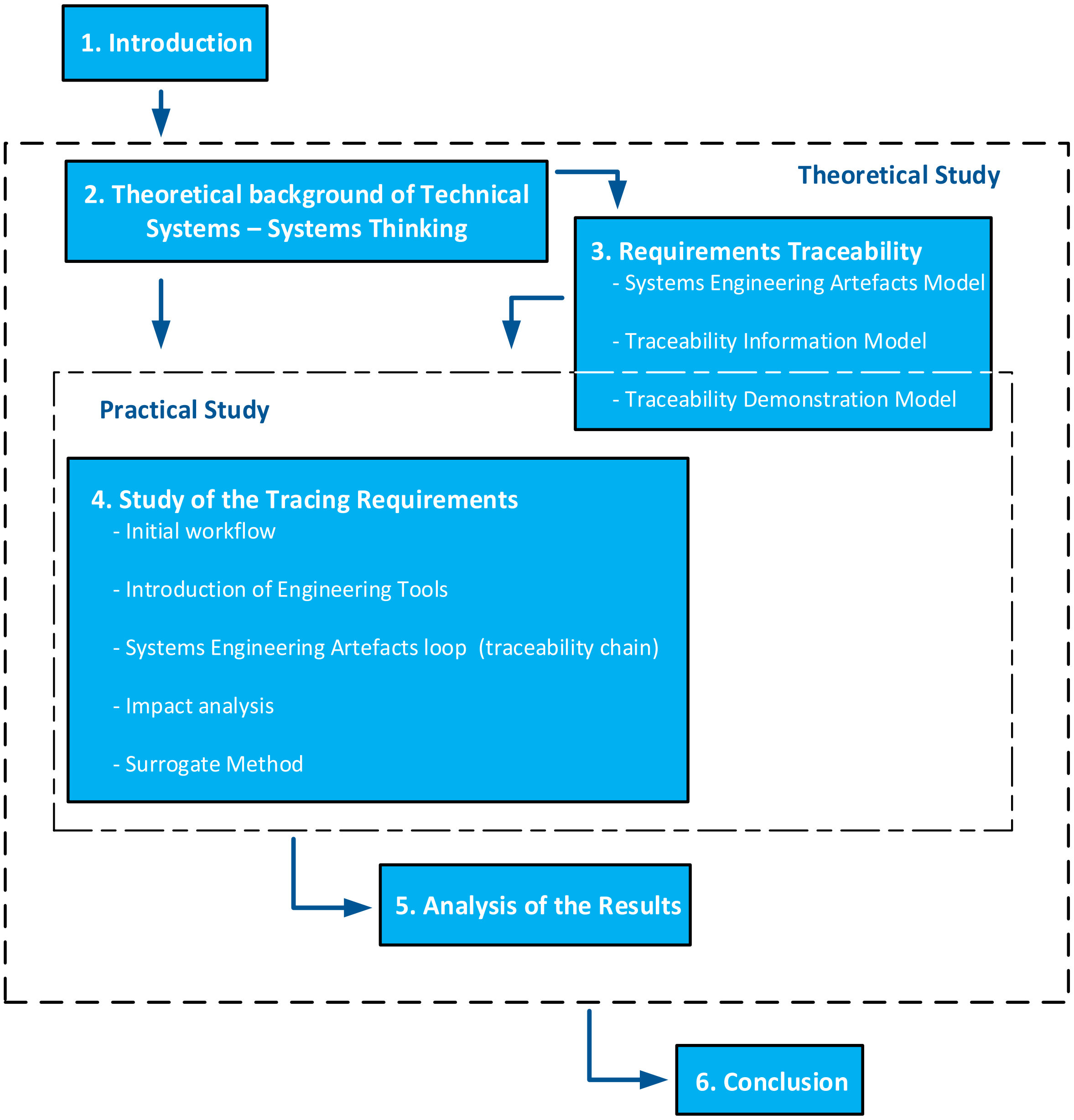


Figure 1.4. *Structure of the thesis.*

Chapter 4 discusses the practical implementation of the before mentioned models by concentrating on the initial workflow of the traceability demonstration case. After the simulation case is presented the actual building of traceability from requirements to design models and finally to verification and validation is formulated. SysML diagrams, architectural models, CAD models, and simulation model are presented in different interphases during the demonstration process as they are realised according to the systems engineering artefacts loop.

After the engineering tools and their work products (engineering artefacts, e.g. model files) are defined and implemented, the study of traceability is performed. A valid part of producing traceability is the possibility to perform an impact analysis. The implementation work of the impact analysis with a set of heterogeneous modelling tools also required to study surrogate object methodology.

The results of each interphase are gathered in Chapter 5 where a summary is produced from the practical part of the study. Results are reflected against the research questions as the outcomes of traceability and impact analysis are evaluated. Chosen methods are discussed and recommendations for improvement of the models and workflows are presented. The concluding Chapter 6 takes an overlook of the whole execution process of the requirements traceability in simulation driven mechanical engineering, where the process succeeded and what kind of significance the results have on requirements aware simulation engineering.

2. THEORETICAL BACKGROUND OF TECHNICAL SYSTEMS

This chapter studies different perceptions about implementation of Systems Engineering on which to base further study of simulation driven requirements traceability with respect to mechanical engineering. Studied perceptions pertain to Systems Thinking, which regards the study of systems influencing on one another in a complete entity. Systems Thinking considers the whole system and its context, and can be defined as a set of practices to analyse the technical and social components and interrelationships of a system in an engineering environment (Lamb & Rhodes, 2009). In other words, Systems Thinking defines interconnections between system elements and provides a general mind-set for modelling systems.

As a pervasive approach to systems and problem solving, Systems Thinking considers systems as “Hard” with a representation of computer systems analysis and Systems Engineering or as “Soft” where systems are related to human activities and represented by soft systems methodology. Hard Systems Thinking is more relevant to this thesis as it is based on goal seeking and assumes systems can be engineered to achieve precise and quantified objectives (Checkland & Haynes, 1994). Goal seeking ideology implies that any human activity is also regarded as a goal-seeking system and as such Hard Systems Thinking and Systems Engineering does not take into account the complex motivations of real human activity systems. The lack of human motivations, however, may be misleading in regards to this thesis. Even though the tracing requirements case study is mainly engineering based with computerized simulations, the systematic creation and storing of engineering artefacts depends on the motivation of the engineering personnel.

The concept of Systems Thinking can be regarded as the basis theory of this thesis as ST is where design science essentially began. Therefore Systems Thinking can also be regarded as a transition theory to the Theory of Technical Systems (TTS) by Vladimir Hubka and Wolfgang Eder which incorporates elements of a systems theory (Hubka & Eder, 1988). TTS is the predecessor of many design processes as it intrinsically is a design process itself. After introducing the origin and mind flow of design processes, a suitable process is exhibited to represent the design workflow of the traceability demonstration model.

2.1 Total Theory of Technical Systems – Hubka & Eder

Theory of Technical Systems is aimed to present a comprehensive theory capable of classifying and categorizing the intrinsic nature of technical systems. Theory of Technical Systems centres on a transformation system, pictured as a general model in Figure 2.1, in which a technical system is a process of reaching desired outcome by introducing transformation as a series of intermediate states. These states are operations (Op) in which certain properties of operands (Od) of the system are subjected to change.

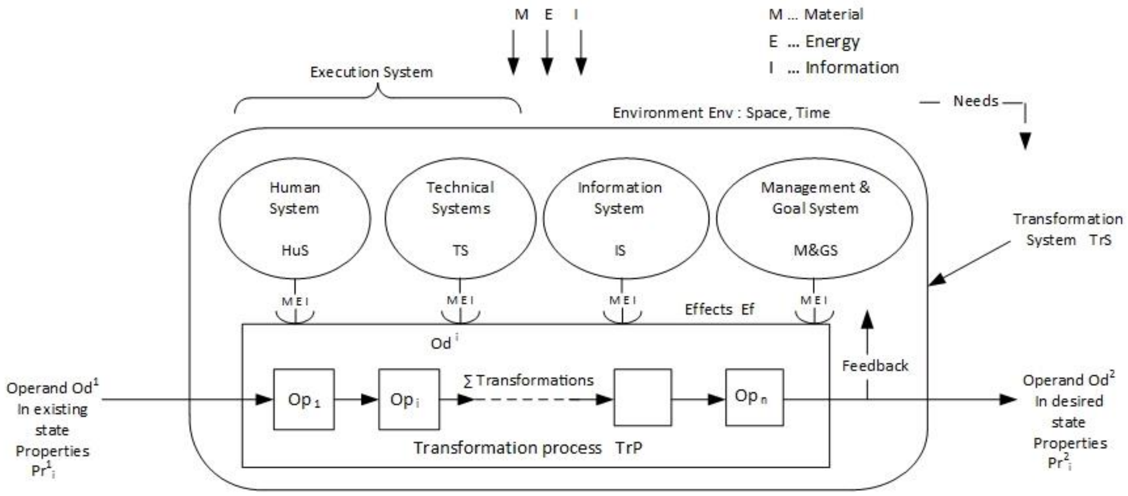


Figure 2.1. *Model of the Transformation System (Altered from Hubka & Eder, 1988).*

Operands are necessary for the accomplishment of the desired transformation whereas technical systems are the main means to achieve the transformation. According to Hubka & Eder, the theory assumes that any technical system exists to fulfil a need and hence the theory requires needs or demands as an initiator for the transformation system. For the sake of this thesis, needs and demands are seen here as requirements. Transformation process takes the existing original state of an operand as an input and transforms it into an output, a desired state defined to match the requirement. Other elements influencing in the cause of transformation are effects (Ef) which include human system, technical system, information system and management & goal system. Executing systems consists of human system and technical system. Also the immediate environment can have a significant influence on the transformation process.

Flows into these systems and from them into the transformation process are material, energy and information. These are resources invested by different systems. The output of the transformation system is measured by processes of feedback which compare the output to a desired goal set by a need. If deficiency or errors are detected the input is suggested to dynamic altering in attempt to correct the unwanted occurrence. This kind of verification and validation concerns operand, the process, and each of the operators as feedback takes place in internal and external feedback loops of each system. (Hubka & Eder, 1988)

2.2 Applying Theory of Technical Systems to the case study

The relevance of the theory of technical systems to this thesis derives from the way transformation system works. Each transformation system performs the intended transformations on the according operand, thus fulfilling the stated and implied needs (Hubka & Eder, 1988). Also transformation by operations can be seen as a design process where operations form different phases of design. The transformation process is in accordance with a development process of manufacturing a machine according to stakeholder requirements. Transformation system can represent both system process and systems engineering process depending on the studied system. Figure 2.2 illustrates the later by applying the transformation system to the case study of this thesis; mobile elevating working platform (MEWP). Case is demonstrated in more detail later in Chapter 3. Figure 2.2 depicts an example of the case where the studied system comprises of engineers related to design the machine.

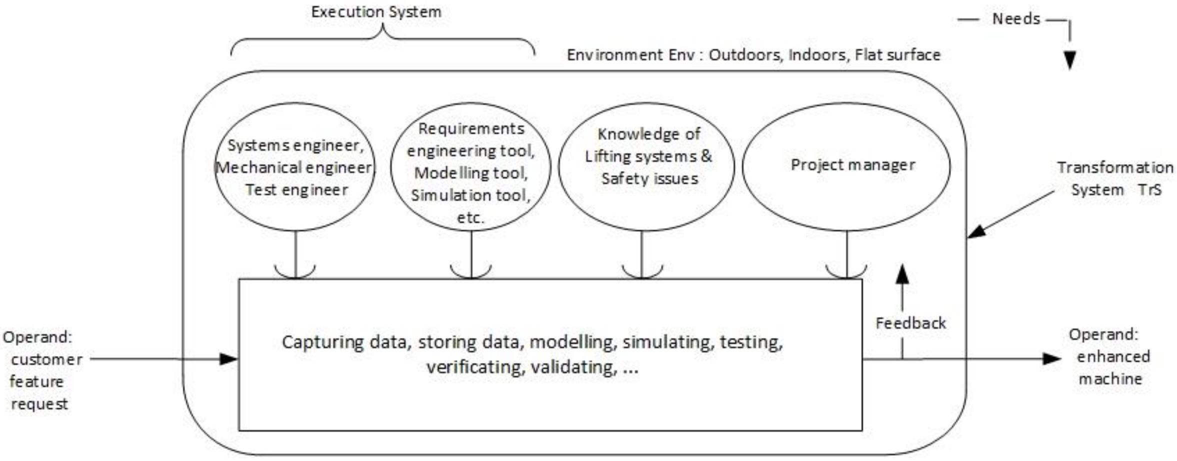


Figure 2.2 . *Example: "Upgrading a machine according to a customer feature".*

Input operand is the customer request for a specific feature which is applied to the mobile elevating working platform during the transformation process to produce output operand of an enhanced machine. The executive system consist of systems engineer, mechanical engineer and test engineer who apply systems tools and engineering tools such as requirements engineering tool, modelling tool and simulation tool to design the new feature. Design process is led by knowledge of lifting systems and related safety issues that need to be answered to. The overall process is watched over by a project manager. In the end, the design is verified and validated to match the desired standards of the customer.

2.3 Design processes

Design processes are the natural successor to the TTS. Both the theory of Design process and TTS share similar methodologies, (Juuti, 2008) and both theories follow the same pattern of developing products from abstraction level to detailed solutions. Design processes also realise the transformation process similar to the TTS by including the same concepts of artefacts, structure and operands of the technical system to the theory. In a design process the transformation process similarly always starts with a state concerning requirements and ends at final, desired state. Generally speaking, a design process defines the design workflow of a new product. (Hubka & Eder, 1988)

Currently there exists numerous different design processes, but they all share the mind-set for modelling systems and defining the possibility for interconnections between system elements as they are complementary to the theory of Systems Thinking. The traceability demonstration model presented in this thesis could have been created from theories such as Property-Driven Development by Christian Weber (Weber, 2012) or the Theory of Dispositions by Jen Olesen (Olesen, 1992), but a more systematic approach for the tracing of engineering artefacts was endowed by the Systems life cycle processes.

2.3.1 Systems life cycle processes – ISO/IEC/IEEE 15288

ISO/IEC/IEEE 15288 establishes a systematic framework for describing the life cycle models of man-made systems. Framework can be applied to one-of-a-kind systems, mass-produced systems and adaptable systems. The nature of the system can be a standalone and it can be embedded or integrated into a more complex and complete system as is the case in this thesis. These systems can be configured with system elements such as hardware, software, data and processes. However, in practice systems are seen as products or services in defined environments that benefit the stakeholders. (ISO/IEC/IEEE 15288, 2015)

The definition of a system, its architecture and the included system elements depend on the observer. A specific system-of-interest can be viewed as a system element by another observer in his system-of-interest. For instance, for the developer of a subsystem a subsystem is in fact a system-of-interest. Hence, the nature of the before mentioned artefact types depend on the point of view. Generally, a system element is an engineering artefact that satisfies a requirement. Additionally, a specific system-of-interest can be regarded as being part of the environment or operation for another system-of-interest. The relationship between the system and its set of system elements can be explained with Figure 2.3 that depicts the hierarchical structure of the systems.

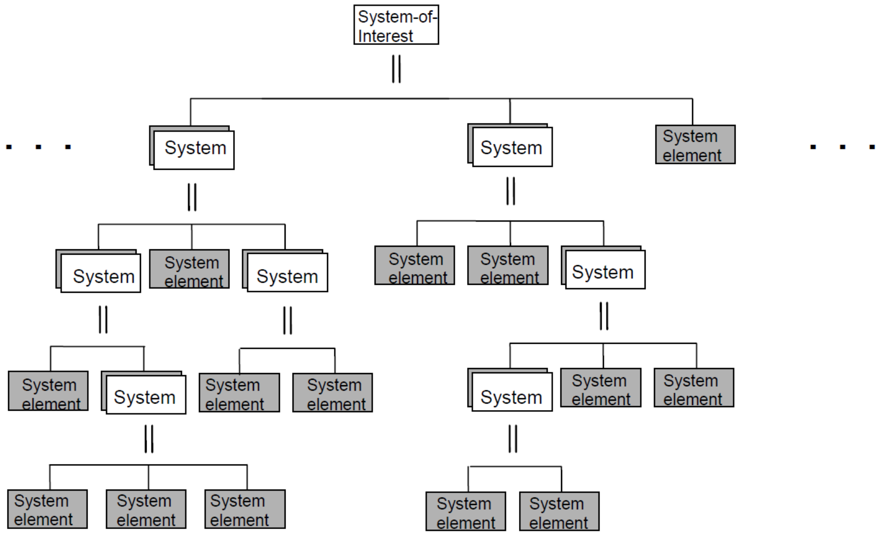


Figure 2.3. *System-of-interest structure model (ISO/IEC/IEEE 15288, 2015).*

Figure 2.3 displays how a system element can be considered as a system that comprises of individual system elements. System element can thereby be a subsystem of its own or an atomic element, i.e. a logical component or a physical component. ISO/IEC/IEEE 15288 does not editorialise on the nature of a system element, but in the models presented in this thesis a system element is considered as a logical component. Component is an artefact type like subsystem and external system that does not have a dedicated modelling element representation. Hence, it cannot be fixed according to every situation whether a system is a system-of-interest, subsystem or an external system. Yet, subsystem and component can be illustrated as a system or a system element, thus disclaiming the need for a dedicated representation in the data repository implementation.

As the complexity of systems increases so does the challenge to manage these systems. ISO/IEC/IEEE 15288 offers a common process framework to improve communication and cooperation among creation, utilization and management of modern systems covering the whole life cycle of systems and all levels of architectural detail. The full life cycle of systems includes conception, development, production, utilization, support and retirement. The standard provides a comprehensive set of life cycle processes that can construct systems life cycle models according to a specific purpose. The purpose can be a product or service and depending on its nature appropriate subsets can be chosen from the standard to fulfil that purpose. Provided processes can also be applied for supporting the life cycle processes of an already existing target.

Processes performed during the life cycle of a system are divided under four groups. Process groups consist of Agreement Processes, Organizational Project-Enabling Processes, Project Processes and Technical Processes. Relevant processes regarding this thesis are found under the group Technical Processes which best addresses the issues of requirements traceability in a simulation driven environment. Also the Configuration Management Process under the group Project Processes is included as a management side process in this thesis. However, not all processes from the Technical Process are necessary for development of the workflow for the demonstration case. The four process groups are depicted in Figure 2.4 with the relevant processes highlighted with red.

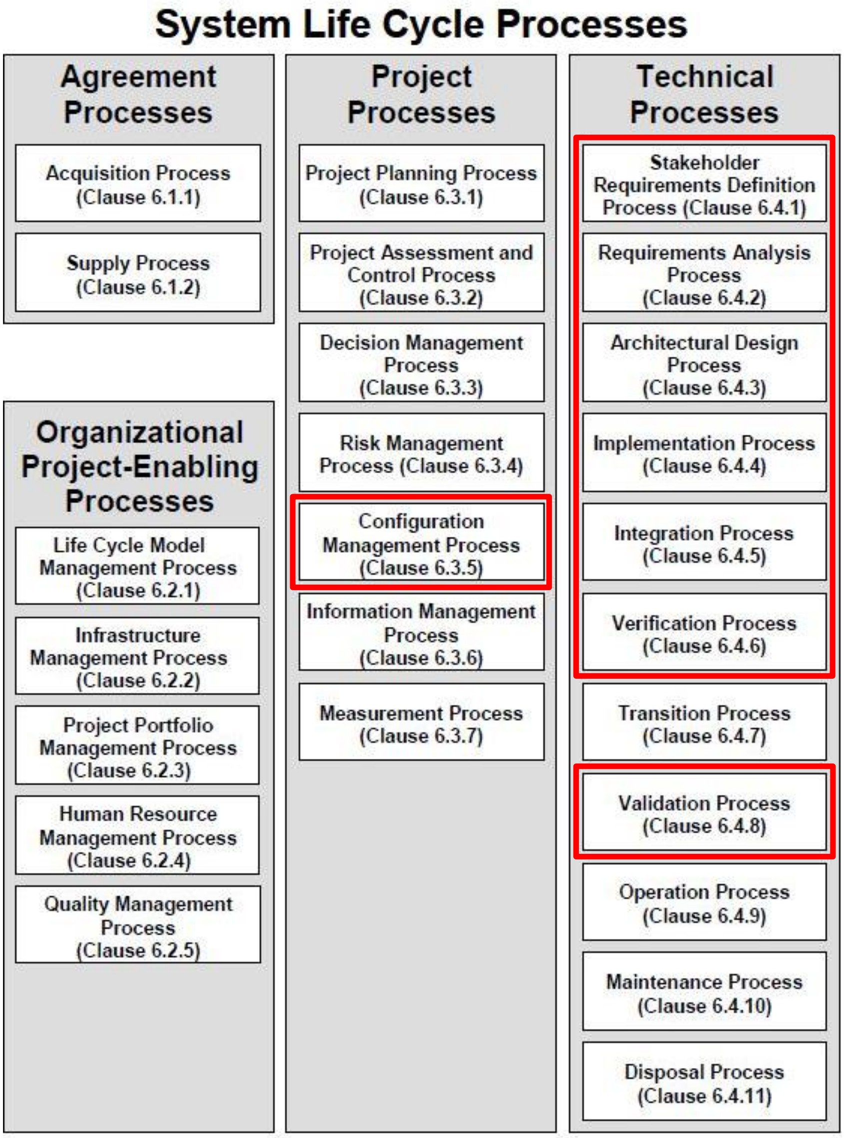


Figure 2.4. *System life cycle processes (ISO/IEC/IEEE 15288, 2015).*

Technical processes issue the technical actions concerning the life cycle of systems. Processes included in this group transform stakeholder needs into products and services in order to achieve customer satisfaction. Furthermore, these processes define the requirements of a system, and the activities that optimize the benefits and reduce the risks that may arise from technical decisions. These activities enable the outcome of the system to conform the expectations of stakeholders and safety regulators. Technical processes included in the thesis are Stakeholder Requirements Definition, Requirements Analysis, Architectural Design, implementation, Verification and Validation.

2.3.2 Stakeholder Requirements and Requirements Analysis

Stakeholder Requirements Definition Process identifies relevant stakeholders and their needs throughout the life cycle of a system. Needs are turned into stakeholder requirements that work as a reference against which resulting operations are validated. Consequently, stakeholder requirements explain what the potential new system has to accomplish in order to satisfy the needs of stakeholders. Typically stakeholders consist of users, customers, suppliers, developers, businesses and, for instance, safety standards. The challenge is to capture the need unambiguously so that requirements can be easily communicated and agreed on without resorting to conventions. Needs of the stakeholders are often varied and large in numbers, which may cause needs to conflict. Furthermore, these needs may not be clearly stated in the beginning and they may be constrained by factors outside the control of the specific need. They may also be influenced by goals that themselves may change during the course of time. To avoid the possible faltering of a development project, the requirements base needs to be stable and the requirements engineering in good order.

Requirements engineering is not directly defined in the ISO/IEC/IEEE 15288, but the activities related to it such as requirements management are indexed in the INCOSE (International Council of Systems Engineering) manual which also includes processes referred in ISO/IEC/IEEE 15288 (INCOSE, 2006). Requirements engineering refers to the processes of defining, documenting and maintaining requirements by defining the problem scope according to the requirements within the problem domain. The process links all the relevant development information into the scope. However, to meet the correct target values set for the new system or product the stakeholder requirements need to be specified accordingly.

Requirements Analysis Process takes the stakeholder requirements and creates a representation of a future system with measurable system requirements. System requirements specify characteristics to satisfy the stakeholder requirements from the perspective of product or service provider. In practice, the requirement-driven view of the stakeholder requirements is transformed into a technical view of a required system or product defined by the system requirements. A solution that complements system requirements is encapsulated by the Architecture Design Process.

2.3.3 Implementation of Design and Verification & Validation

Architectural Design Solution is defined by an implementation strategy according to requirements set for the system elements from which the system is configured. This process specifies design requirements that devise an assembly and verification strategy. Design requirements are satisfied through verification by the **Implementation Process** that produces a system element accordingly. Resulted system element also satisfies relevant stakeholder requirements through validation. System element is created according to the selected implementation technology. These system elements are combined into system configurations by the **Integration Process**.

Assembled systems answer to the architectural design and create a product specified by the system requirements. During the creation of the product, the specified design requirements are conformed in the **Verification Process**. Verification process provides objective evidence that the system and therefore the product fulfil the system requirements and that the relevant architectural design is provided. In the end, **Validation Process** assesses and confirms that the stakeholder requirements are correctly defined and they achieve their intended use in the intended operational environment. Possible variances are dealt with corrective actions after identification.

2.3.4 Configuration Management

The whole design process can be considered to be looked over by the **Configuration Management Process** that establishes and maintains the integrity of the identified process. Configuration management (CM) concerns that statuses of items under version control are consistently made available to relevant places throughout the life cycle. Respectively, the change is controlled and carried out according to defined configuration management strategy. Managing engineering changes is a relevant part of the interdisciplinary field of Systems Engineering as changes are unavoidable and might emerge during any phase of the product life cycle. Changes can propagate due to many reasons for instance as changes in customer requirements, immature decisions or because of development in manufacturing processes. Therefore, for companies to keep competitive advantage or to correspond to the technological innovations of the competition, it is important to handle engineering changes accordingly.

Engineering Change Management (ECM) is the process of determining and evaluating changes to a specific system or product. On a more common level, engineering change management covers the process to introduce modifications on a product similar way as in configuration management. Even though these two overlap considerably, they are not integrated by definition. ECM concentrates on supporting the processing and traceability of changes to interconnected factors whereas configuration management maintains consistency of the characteristics of a product with its requirements, design and operational information throughout the life cycle. According to SFS-EN-ISO 10007 standard ECM could be seen to work within the CM as a part of configuration control (SFS-EN ISO 10007, 1996). Configuration control involves deciding the degree of formality in processing the change according to the affected configuration baseline, customer requirements and the impact of the change.

The before mentioned technical and configuration processes form the basic workflow for the development of the traceability demonstration model presented later in Section 3.3. ISO/IEC/IEEE 15288 is also utilised among its daughter standards in the development of the data models; SEAModel and TIM.

3. REQUIREMENTS TRACEABILITY

Chapter 3 introduces the model structures that provide the requirements traceability. Traceability and impact analysis of engineering artefacts is facilitated with an implementable data model designed by the SIMPRO research group; Systems Engineering Artefacts Model (SEAModel). SEAModel specifies the main engineering artefact types, their relations, and provides the means for creating traceability information model (TIM) that enables the actual management of tracing. TIM is implemented with a traceability demonstration model defined by the ISO/IEC/IEEE 15288. Mentioned models can be represented in many ways. For a graphical representation, as it was in this study, models should be represented with a description language that is easy to understand and customize. In this study UML was chosen as it is the standard for the modelling of object oriented systems.

Sections 3.1 and 3.2 describe the basic characteristics of SEAModel and TIM by exhibiting answers to questions why and how these data models are constructed. The deeper insight of these models is out of the scope of this thesis and can be looked up in more precision from the publication of VTT (Alanen, et al., 2015). Later in Section 3.3 a traceability demonstration model is described where the demonstration case introduced in Chapter 4 is based on.

3.1 Systems Engineering Artefacts Model – SEAModel

SEAModel answers to the need to provide traceability between different artefacts. The importance of a traceability chain is highlighted from requirements to design and implementation, from implementation to test execution and from test execution to verification and validation reporting. As a data model, SEAModel strives to fill a hole for a model that can be easily implemented onto a relational database. Basically, any relational database can be used as a basis for a systems engineering artefacts data repository according to SEAModel.

The main idea of SEAModel is that the physical structure of the system is specified by its system elements, and by their relations. Hence, SEAModel is included to every system, whether a system-of-interest or a subsystem. The system-of-interest can have external subsystems that may not have utilized SEAModel as long as the necessary data is received from the subsystem and can be stored accordingly. However, in this case it may be difficult to arrange seamless traceability of requirements from the main system to the furthest system element, and traceability of verification artefacts from the system element to the main system.

SEAModel consists of artefact types that are divided under different upper level artefacts according to the knowledge they represent. Knowledge based division is justified, since many artefacts such as requirements fit to several sections of the project data repository. The current upper level artefacts and artefact types regarded in SEAModel are depicted in Figure 3.1 as packages and nested packages. These packages exchange elements between each other. Defined packages are System, System context, Specialty Engineering, and Requirements and Verification and Validation (V&V).

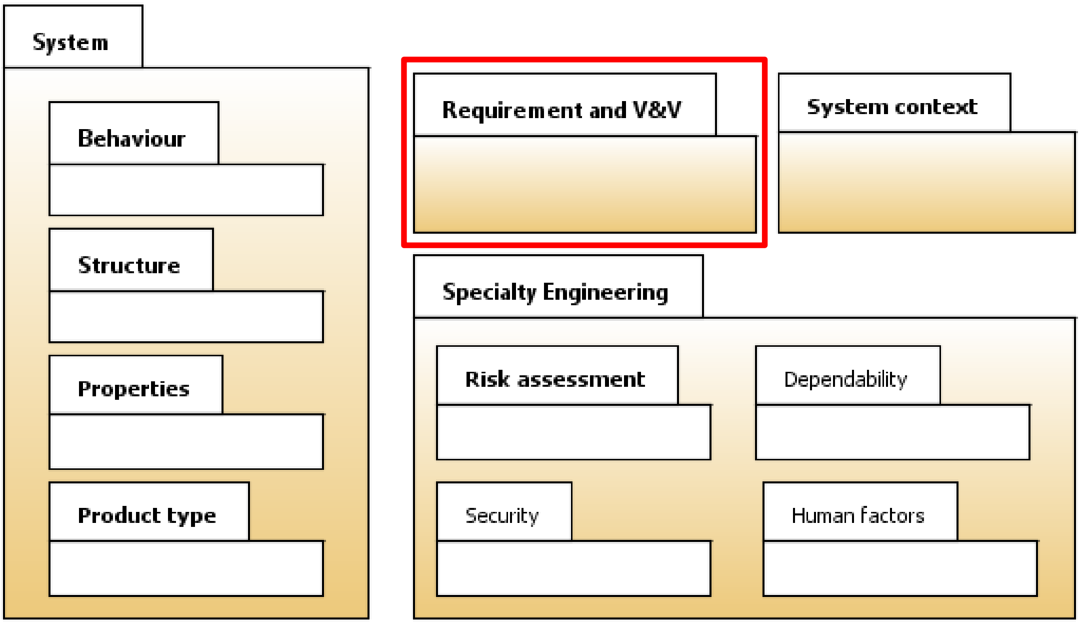


Figure 3.1. *SEAModel upper level artefacts and artefact types presented as packages (Altered from Alanen, et al., 2015).*

Requirements and V&V, which is highlighted with red in Figure 3.1, is the main focus of this thesis, and the starting point for SEAModel implementation. This package is opened in Figure 3.2 where the systems engineering core loop for traceability through requirements to verification or validation reporting is presented. Only the core diagram of SEAModel is demonstrated in this thesis.

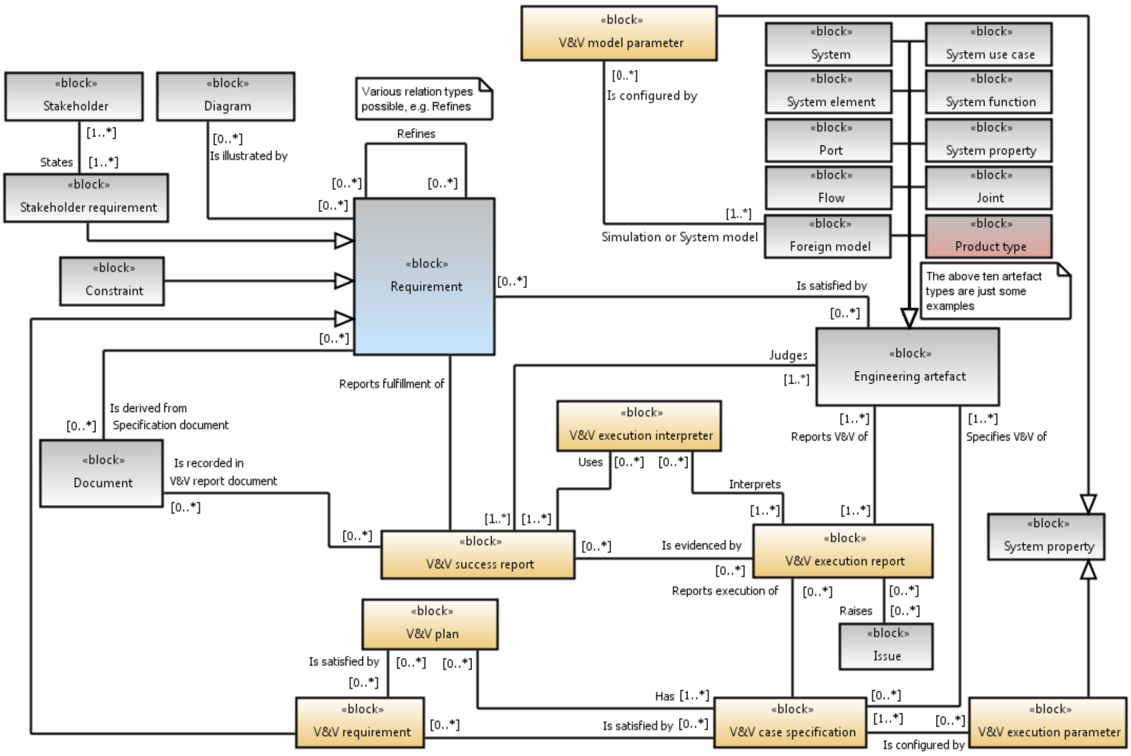
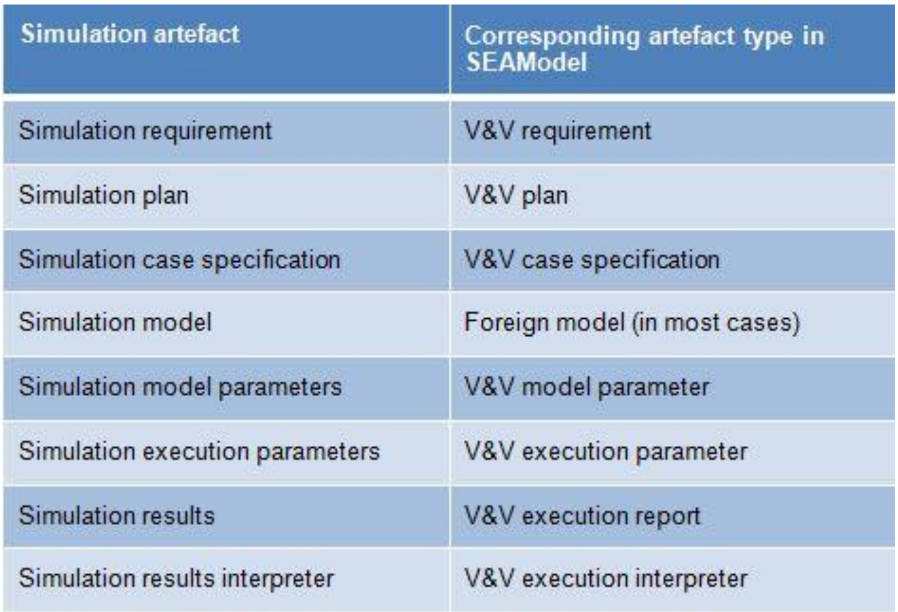


Figure 3.2. System engineering artefacts model for Requirements and V&V artefacts (Alanen, et al., 2015).

The Requirements and V&V model centralizes around the Requirement artefact that stores both the stakeholder and system requirements. The top level system requirements are derived from the stakeholder requirements. Other engineering artefacts that satisfy these requirements are verified and validated during the development work. For this process a special set of artefact types are provided: V&V requirement, V&V plan, V&V case specification, V&V model parameter, V&V execution parameter, V&V execution report, V&V execution interpreter and V&V success report. The interconnection of the before mentioned artefact types is depicted in Figure 3.2. Artefact types in turn are described in more detail in the Chapter 4. In regards to the simulation process presented in this thesis, simulation artefacts are mapped onto SEAModel according to these V&V artefact types. Table 3.1 showcases the simulation artefacts constituting the demonstration case and where they are stored among V&V artefacts.

Table 3.1. *Mapping of simulation related artefacts onto SEAModel. (Alanen, et al., 2015).*



Simulation artefact	Corresponding artefact type in SEAModel
Simulation requirement	V&V requirement
Simulation plan	V&V plan
Simulation case specification	V&V case specification
Simulation model	Foreign model (in most cases)
Simulation model parameters	V&V model parameter
Simulation execution parameters	V&V execution parameter
Simulation results	V&V execution report
Simulation results interpreter	V&V execution interpreter

Besides simulation artefacts, common artefacts like requirements and design artefacts are also relevant. The design artefacts are typically design and simulation models that are stored as foreign models in SEAModel data repository. Hence, foreign models are not stored as structured model elements, but as separate files. The foreign model artefact can be linked to several artefacts such as System or System element. In this thesis foreign models are incorporated to Simulation model artefact as it is in most cases. The design artefacts are verified and validated by the simulation cases and their correct design, based on the simulation results, is justified against the requirements.

Tracing of the engineering artefacts is manifested with a Traceability Information Model. TIM is easy to convert from SEAModel, since most of the artefacts relations can be considered as trace relations.

3.2 Traceability Information Model – TIM

Traceability Information Models (TIM) provides guidance to software and engineering artefacts to form traces through established relations. Essentially traceability information model consists of traceable artefacts and traceability relations between these artefacts. To manifest traceability, TIM needs to define the artefacts intended for tracing with related artefact types. Also the type of traceability relations needs to be defined. With semantics given to these relations, validation of the relations is enabled. Ultimately TIM is designed to support required project analyses by depicting how different engineering artefacts trace to one another. Similarly to SEAModel, TIM is applied to every system in the project, whether a system-of-interest or a subsystem. If TIM is not used in some specific system element, then a seamless traceability of requirements throughout the system may be difficult to arrange.

Traceability makes it possible to indicate how certain design solutions have come to be and how they are derived. This is a complex process which requires information to be created and maintained as part of the design data. TIM eases the set up with a guideline that allows the validation of changes. To accomplish this, engineering tools needs to support traceability by showing the dependencies and indicating potential impacts of the change occurred to a specific artefact.

In the context of this thesis, artefacts tracing is illustrated in Figure 3.3 where a traceability information model for Requirements and V&V artefacts is derived from the equal SEAModel representation depicted in Figure 3.2.

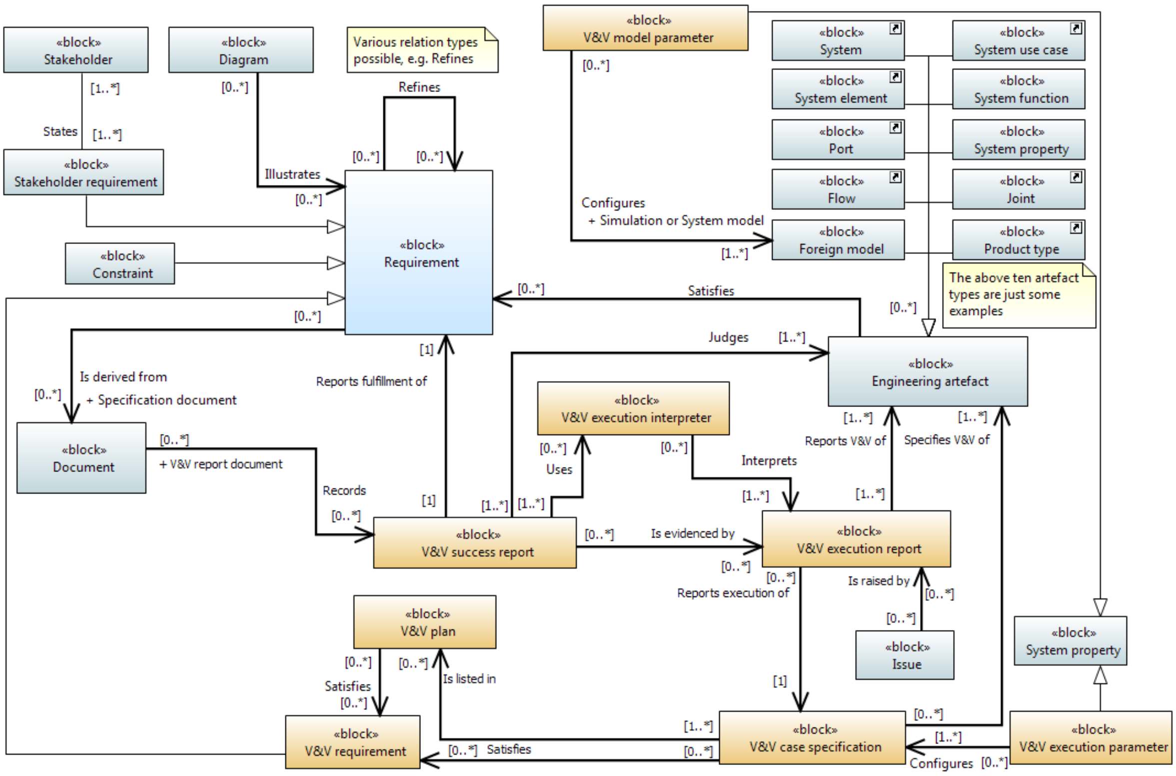


Figure 3.3. Traceability information model for Requirements and V&V artefacts (Alanen, et al., 2015).

The difference between TIM and SEAModel is that not all relations in SEAModel need to be trace links. The amount of trace links depends on the situation and the demanded level of traceability. For instance, as it can be seen from Figure 3.3 compared to Figure 3.2, the relation between Stakeholder and Stakeholder requirement is not considered as a trace link since it does not affect the traceability of engineering artefacts to requirements. Other differences between the models are the relationship names that, in some cases, are updated to their inverse versions from SEAModel to TIM in order to provide reading order from the source to the target. Also the link directions do not necessarily reflect the trace direction in SEAModel representation. The direction of the arrowed trace links is from the younger information to the older. Therefore, if the artefact is changed in the target end the source end artefact becomes suspect, and has to be checked for a possible update. Consequently, if the artefact in the source end is changed it has to be checked whether it still is consistent with the unchanged target end artefact that has now become under suspect.

The implementation of TIM requires the data repository platform to support traceability management which includes impact analysis. To demonstrate traceability according to SEAModel and TIM, a traceability demonstration model is implemented to illustrate the systems engineering workflow.

3.3 Traceability Demonstration Model

The traceability demonstration model is a case dependent representation of the artefacts framework set up by TIM. Traceability demonstration model is built onto a fictional case study presented in more detail in Section 4.2. The fundamentals of the demonstration model are defined by the ISO/IEC/IEEE 15288, hence the model follows the work flow presented in Section 2.3.1.

The traceability demonstration model is introduced in Appendix A: Traceability demonstration model. The model represents the three levels of the demonstration: Scissors-elevating platform (MEWP) which is the system of interest, Mechanical subsystem of the MEWP, and Platform mechanical subsystem. Engineering artefacts are divided into Stakeholder requirement, System requirement and Other artefacts categories. Connections between artefacts define the relation type and the inheritance direction of the artefacts. Traceability of the artefacts is presented in more detail in Section 4.7 as is the case study in general in the upcoming Chapter 4.

4. STUDY OF THE TRACING REQUIREMENTS

This chapter demonstrates how SEAModel based TIM is designed to work with a mechanical industry related case. TIM influences the specific case by creating a traceability demonstration model that represents the case. The company depicted in the demonstration scenario is fictitious, but the situation of engineering a feature request is applicable to machine industry in general. The demonstration case is explained by first introducing how different actors are related to the development process of the example case and how a systems engineering workflow scenario is created to trace requirements. Afterwards, the workflow is carried out by introducing tools to implement different models, requirements management, integration and traceability. Eventually, requirements traceability is studied with chosen tools and methods.

4.1 Initial Systems Engineering workflow

A fictitious company is developing mobile elevating work platforms (MEWP) of scissors type when they receive a new customer feature request to modify the platform movement. The customer requirement affects various staff members in the company, and their relation to the new feature request can be depicted with a use case diagram shown in Figure 4.1.

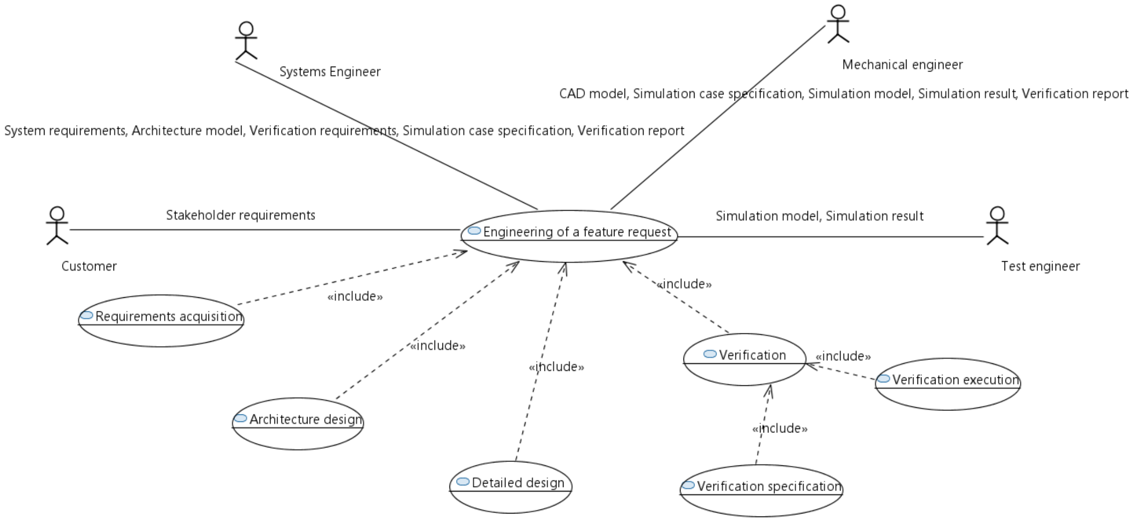


Figure 4.1. *The main use case (Alanen, et al., 2015).*

The related actors are Customer, Systems engineer, Mechanical engineer and Test engineer. Actors are interacting with the engineering artefacts via tools; Requirements Management Tool, SysML Tool, CAD Tool, Simulation Tool, Verification and Validation (V&V) Tool and Integration and Traceability Platform (ITP). The engineering artefacts are: Stakeholder requirements, System requirements, Architecture model, CAD model, Verification requirements, Simulation case specification, Simulation model, Simulation results, and Verification report. Interaction process and a more detailed analysis of storing and using of engineering artefacts in the integration and Traceability Platform (ITP) is shown in Figure 4.2. ITP is represented as a data store.

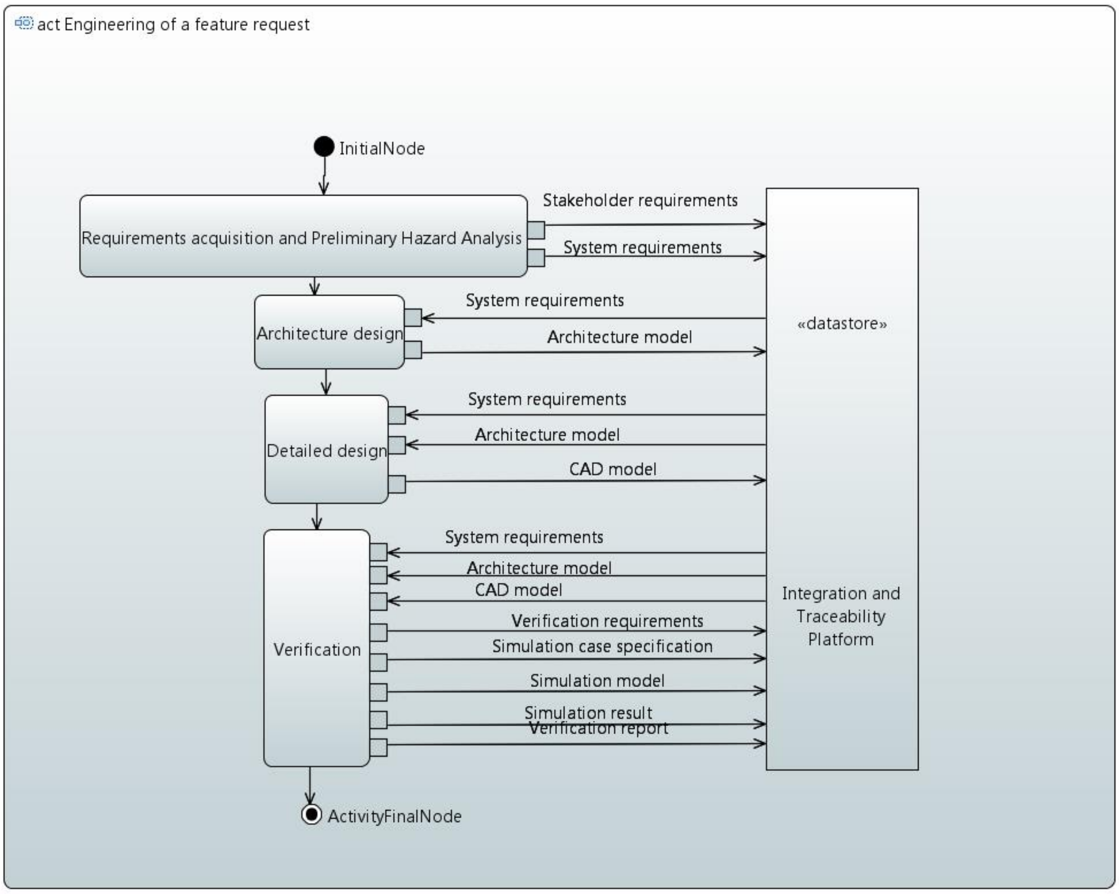


Figure 4.2. *The overall workflow and use of engineering artefacts in ITP (Alanen, et al., 2015).*

The customer request is analysed by the systems engineer who captures and creates stakeholder requirements and the consequent system requirements within the requirements management tool. The systems engineer also designs an architecture model of the updated mechanical design of the platform. According to the architecture model, a mechanical engineer creates a CAD model implementation, which is simulated according to a simulation case specification specified by the systems or mechanical engineer. The simulation case specification is created according to verification requirements defined by the systems engineer. The simulation is realised with the help of a simulation model, which is created out of the CAD model by the mechanical designer or a test engineer. The conformity of the CAD model is evaluated against the relevant requirements based on the simulation results. Finally, a verification report is issued by the systems or mechanical engineer.

The requirements traceability is demonstrated with a case study that follows the structure of the traceability demonstration model presented in Section 3.3. The case is divided according to engineering artefacts and the motion of this study follows the workflow introduced in Figure 4.2.

4.2 System identification – Mobile Elevating Work Platform

A MEWP was chosen as the demonstration platform, because the subject is familiar to VTT from previous work, and MEWP has been used as a systems engineering research platform before. The current design and concept for MEWP, depicted in Figure 4.3, descends from a VTT demonstration system for systems engineering platform of a fictitious Scissors Platform.

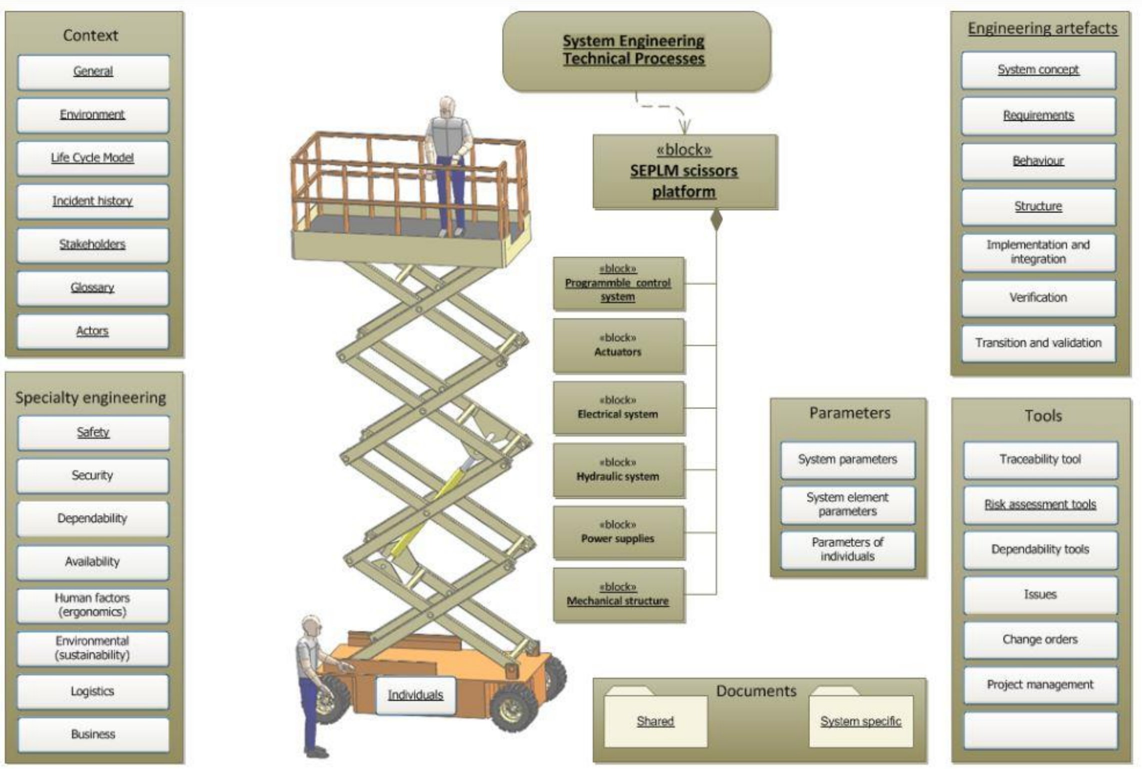


Figure 4.3. Scissors platform - systems engineering platform user interface (Alanen & Valkama, 2014)**.**

The MEWP used in the demonstration is a rubber wheel based scissors platform that can be moved sideways via pivoting axle of wheels. Machine is meant for industrial use both inside and outside environments. The main components of the structure are; Chassis, Scissors and Platform. MEWP is equipped with a diesel engine that powers hydraulic motors. Hydraulic power is used to accomplish movement and electric power for the electric devices and programmable automation system. The engine of the MEWP is controllable from both the chassis and platform, although the ignition is handled only through a control panel at the chassis. The platform can be lifted up to eight meters, and the maximum workload is 900 kg. Dimensions of the machine are roughly 1,6 m in width and 2,9 m in length. A more thorough view into the characteristics of the MEWP is presented in Sections 4.5 and 4.6.2.

4.2.1 Standards SFS-EN ISO 12100 and SFS-EN 280

The risk assessment of the MEWP follows the machine safety standard SFS-EN ISO 12100 (SFS-EN ISO 12100, 2010). The primary function of the standard is to offer general instructions and methodology to decision-making related to the safety of machinery. The aim is to support designers to develop machines, which are safe to use in such an environment where they are designed to work. SFS-EN ISO 12100 lists safety standards of three types: A-type standards include the basic safety standards that express the basic concepts, principals for design and general aspects to be applied to machinery. B-type standards are generic safety standards that handle one safety aspect or one type of safeguard that can be used in various machinery. C-type standards are machine specific safety standards that describe in detail safety requirements for a particular machine or group of machines. MEWP:s can be designed based on SFS-EN 280:2013 which is a type-C standard formulated for MEWP:s (SFS-EN 280, 2013).

The SFS-EN 280 standard includes design calculations, stability criteria, construction directives, safety examinations and tests of mobile elevating work platforms. EN 280 divides MEWP:s into three types and two groups (A and B). Type is decided according to how MEWP is controlled while travelling:

- Type 1: Travelling is only allowed with the MEWP in its transport configuration;
- Type 2: Travelling with raised work platform is controlled from a point of control at the chassis;
- Type 3: Travelling with raised work platform is controlled from a point of control at the work platform.

Division into a specific group is determined according to where the vertical projection of the platform is with respect to the tipping lines:

- Group A: MEWPs where the vertical projection of the centre of the area of the platform in all platform configurations at the maximum chassis inclination specified by the manufacturer is always inside the tipping lines;
- Group B: All other MEWPs.

The original demonstration MEWP is classified in group A, in which the vertical projection of the centre of the area of the platform in all platform configurations is always inside the tipping lines. A notable point in the standard is that we are talking about the centre of the platform area not the centre of the mass. Additionally, the demonstration MEWP is categorised to types 2 and 3, meaning that travelling with a raised work platform is controlled either from a point of control at the chassis or the work platform. (SFS-EN 280, 2013, pp. 7-8)

The design of the demonstration MEWP has adopted the ergonomic principles mentioned in Chapter 6.2.3 of the SFS-EN ISO 12100. Chapters 5.6.6, 5.6.7, 5.7.1 and 5.7.3 of SFS-EN 280 are followed in respect to work platform and controls.

4.2.2 Specifications of a fictitious customer feature request

A fictitious customer wants to upgrade the original MEWP model by requesting a new feature: It has to be able to move the platform of the machine horizontally, sideways, in order to reach closer to a wall in case the chassis cannot be parked close to a wall. The feature request is illustrated in Figure 4.4. The horizontal movement is specified to be within the range of 0…75 cm. The direction is to the right of the main driving direction (positive y-axis direction). At this point, it is not defined how the horizontal movement should be technically executed. A thorough definition of the technical execution of the customer feature is left outside of this thesis. A proposal for the implementation of the movement is presented by the systems engineer. The proposal is presented in Section 4.4.

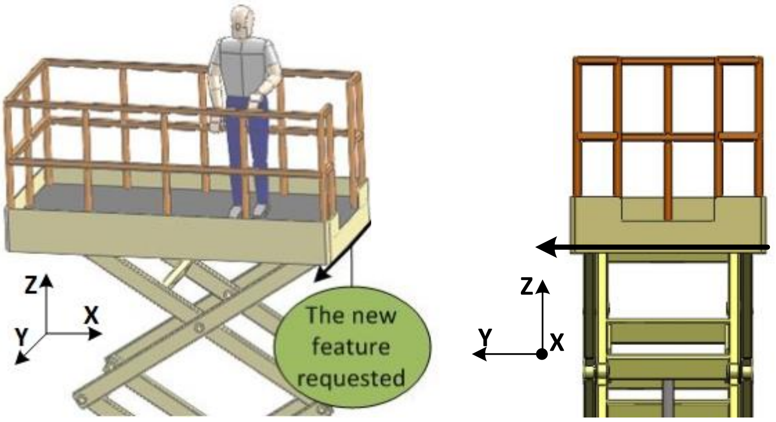


Figure 4.4. *New customer feature request (Altered from Alanen, et al., 2015).*

The new feature request causes the vertical projection of the centre of the area of the platform to exceed the tipping lines by around 5 cm. This leads to re-evaluation of the previous design and safety standards. Because the vertical projection of the platform is no longer always inside the tipping lines, the designed MEWP is classified to group B according to EN 280 (the original design was not supposed to have a moving platform). After risk assessment, it is specified that the horizontal movement of the platform shall not cause loosing of the stability with the maximum platform load with the worst case load location (SFS-EN 280, 2013). The requested feature is added among the customer requirements, and its impact on other stakeholder and system requirements is analysed. Furthermore, new system requirements are created accordingly to form the basis for the design work.

4.3 Possible Integration and Traceability Platform implementations

A study of plausible requirements and traceability tools used in the integration and traceability platform is based on previous research by VTT. The study was done outside of this thesis and it can be revised from the VTT publication Requirements traceability in simulation driven development (Alanen, et al., 2015). Hence the subject is handled on a superficial level concentrating mainly on the results.

Characteristics of the chosen platform were required to be such that SEAModel introduced in Chapter 3.1 can be followed. Therefore, implementable integration and traceability software platform would require the following features:

- Structured artefact repository for creation and maintaining of relations between artefacts
- Artefacts traceability with impact analysis
- Version control of individual and a set of artefacts
- Modification control
- The level of automatic document generation high as possible
- Document management
- Integration possibility with systems engineering tools
- Concurrent engineering capabilities
- Collaboration features
- Metrics of various system engineering issues

Other practical issues considered in the platform selection are cost, responsiveness and usability. Preliminary tools that were taken under consideration were CAD-oriented product life management tools such as Catia/Enovia, PTC windchill, ARAS PLM and Siemens Teamcenter. Although Product Life Management (PLM) tools typically cover at least the core set of the mentioned features, they do not completely support SEAModel and might need a lot of tailoring.

The second ITP implementation to be evaluated was a combination of integration platform ModelBus by Fraunhofer FOKUS and traceability tool Traceino. ModelBus supports a variety of plug-ins that connect the specific tools' internal data representation and that of ModelBus. However, ModelBus does not have CAD tool adapter available meaning one would have to be custom developed. Yet, ModelBus and the client tools concept look promising from the point of view of model based systems engineering. Hence a decision was made to go forward with a demonstration, Figure 4.5.

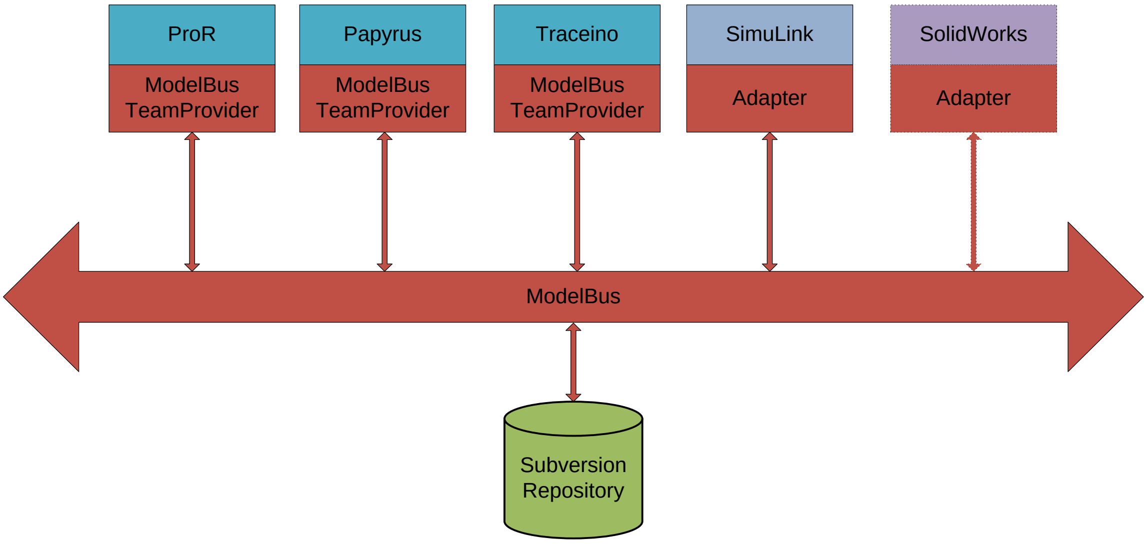


Figure 4.5. *Architecture for ModelBus and Traceino (Alanen, et al., 2015).*

Papyrus was used for SysML modelling via integrated development environment Eclipse, ProR for requirements engineering, and Traceino for traceability between requirements and SysML model. CAD and simulation models were left outside of the demonstration, because of lack of available adapters for CAD and simulation tools such as SolidWorks and MATLAB.

The conclusion of the demonstration noted that the implementation level of ModelBus concept is not mature enough to support research work let alone commercial work. Chosen software solutions would need high level of information and communication technology (ICT) proficiency to manage installations and work around the issues. Such skills are not often available in typical small- and medium-sized mechanical engineering enterprises (Alanen, et al., 2015).

ModelBus and Traceino combination showed that with a random set of systems engineering software tools, a tailored approach might be needed. This in mind the look for requirements and traceability platform tools turned to requirements management software and database oriented management systems. IBM Rational DOORS 9.5 and MS SharePoint were taken under consideration, because both of them were easily available and familiar to VTT. In the end DOORS was chosen for the demonstration case owing to its promoted integration possibilities with MATLAB.

4.3.1 Requirements Management in DOORS

The task of tracing dependencies and the relations created during the verification of a product requirement is essential for a requirements management tool. Requirements management is relatively well supported in the field of systems engineering. An example of this is IBM Rational DOORS which is a requirements management platform for requirements collaboration, communication and verification. Requirements management tool is integrated to other engineering tools through the ITP. DOORS stores stakeholder requirements such as the new customer feature request about the platform horizontal movement. The customer request is recorder as a stakeholder requirement of which a corresponding system requirement is created after analysis, and also stored into DOORS. The requirements acquisition workflow is presented in Figure 4.6. The created system requirements can be categorized for instance as functional, performance, maintenance or safety requirements depending on their nature.

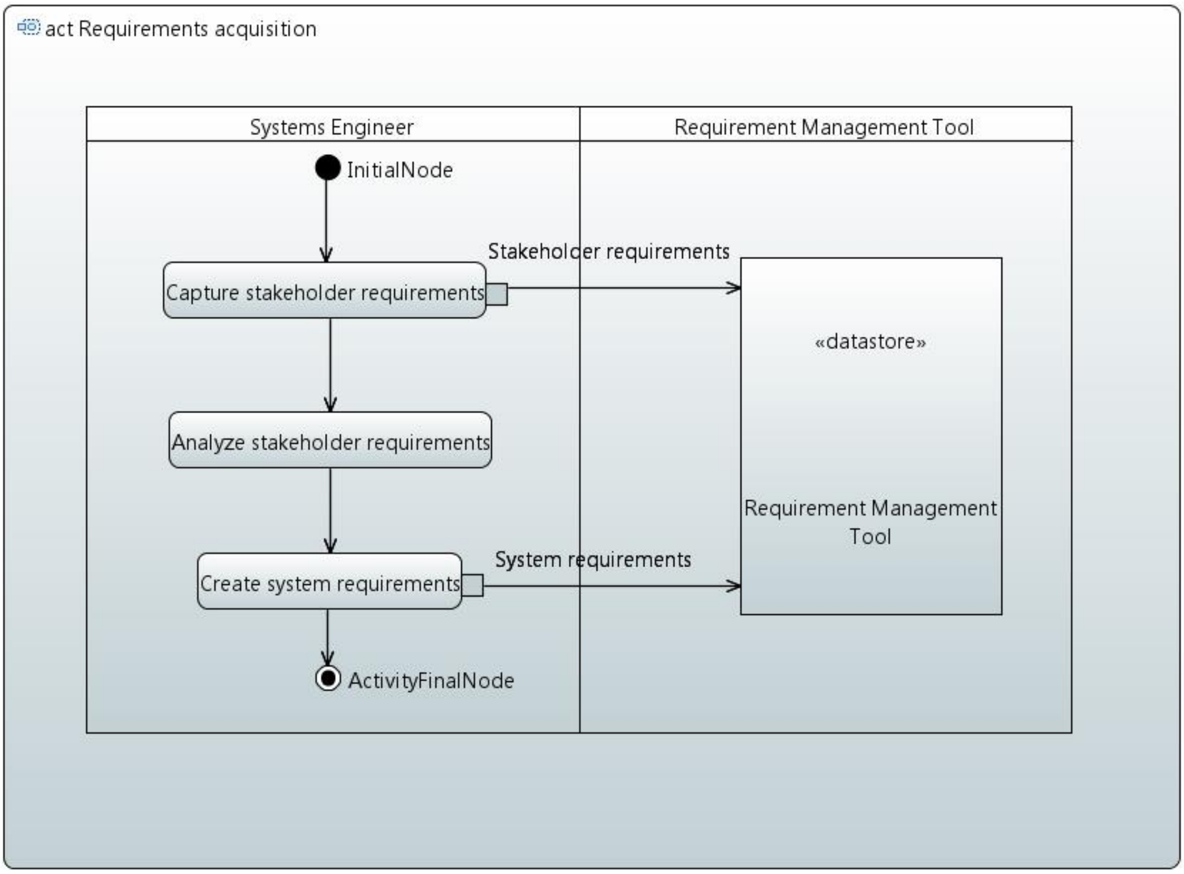


Figure 4.6. *Requirements acquisition workflow (Alanen, et al., 2015).*

In addition to the mentioned customer requirement, a set of requirements for the case study is derived from Machinery Directive 2006/42/EC and from the harmonised C-type standard SFS-EN 280 presented in Section 4.2.1. Introduction of all the requirements included in the case study is irrelevant considering the context of this thesis; hence only the most relevant requirements associated with the new customer feature are presented in Figure 4.7.

ID	
SPRO-REQ-356	**1 Customer requirements**
SPRO-REQ-357	**1.1 Platform horizontal movement** It shall be possible to move the platform in horizontal direction
SPRO-REQ-358	**1.2 Platform horizontal movement reach** The horizontal movement of the platform should reach up to a near by wall at a distance of less than 1 m
SPRO-REQ-359	**1.3 Platform horizontal movement direction** The horizontal movement should be sideways to the right of the main driving direction
SPRO-REQ-360	**2 Functional requirements**
SPRO-REQ-361	**2.1 Platform sideways movement** It shall be possible to move the platform sideways
SPRO-REQ-363	**3 Performance requirements**
SPRO-REQ-362	**3.1 Platform sideways movement reach** The sideways horizontal reach should be should be in maximum -0 ... +75 cm more than with static platform (where + direction is to the right of the main driving direction)
SPRO-REQ-353	**4 Safety requirements**
	⋮
SPRO-REQ-365	**4.15.1 Stability shall not be lost** Movement of the platform to sideways direction shall not cause loosing of the stability with the maximum platform load with the worst case load location

Figure 4.7. *A modified view of Requirements management in Doors.*

The original customer request is expanded to three customer requirements that explain the request in detail. The platform shall move in horizontal direction and the movement of the platform should reach a distance less than 1 m. Movement needs to have a direction, and it is decided to be to the right of the main driving direction. A matching functional requirement for the requested feature is; “Platform sideways movement: it shall be possible to move the platform sideways”, which emphasizes the possibility to move platform. A technical request for the performance of the MEWP is the platform sideways movement reach, which completes the previous functional requirement. An important aspect for the design and manufacturing of the MEWP is safety. The new customer feature should not jeopardize structural stability while the platform is horizontally moving. Therefore, the stability of the MEWP is simulated to inspect whether the stated safety requirement is compromised and possible stabilizers are needed to be added to the chassis. The simulation model is introduced and studied later in Section 4.6.

Engineering artefacts like the requirements mentioned above need to be transferred from requirements management tool to the traceability platform in order to trace relevant artefacts to each other and to enable impact analysis. Because ModelBus and other integration and traceability tools were discovered in Section 4.3 to be unsatisfactory for requirements traceability according to SEAModel, DOORS was chosen to fill this spot instead. Therefore, the ITP considered for the demonstration case was finally established as a collaboration of DOORS and SVN. DOORS functions as a traceability management tool while SVN stores the artefacts information into one repository and functions as an integration platform tool. Traceability management is demonstrated later in Section 4.7 after the traced engineering artefacts are disclosed and implemented to TIM.

4.4 Architecture design – logical model

The logical architecture model indicates information about the system physical structure seen on macroscopic level. The system is decomposed into logical components or subsystems that interact to satisfy the system requirements. Interactions realise dependencies and operations between logical elements. Logical components in turn are abstractions of the physical components that execute the system functionality without imposing implementation constraints (Friedenthal, et al., 2012). Subsystems in turn are a set of interacting components. Furthermore, logical components and subsystems are specialisations of system elements (see Section 2.3.1).

A new customer feature conveys to updating the former MEWP model. According to the requirements, a new design is drafted by designing a logical architecture of the MEWP system. This is accomplished by opening the original architecture model from the ITP. The logical architecture model contains the initial physical structure of which a new CAD model is later produced. With ITP, the architecture model elements are traced to the relevant system requirements; see Figure 4.2 and Figure 4.8 for a more detailed view.

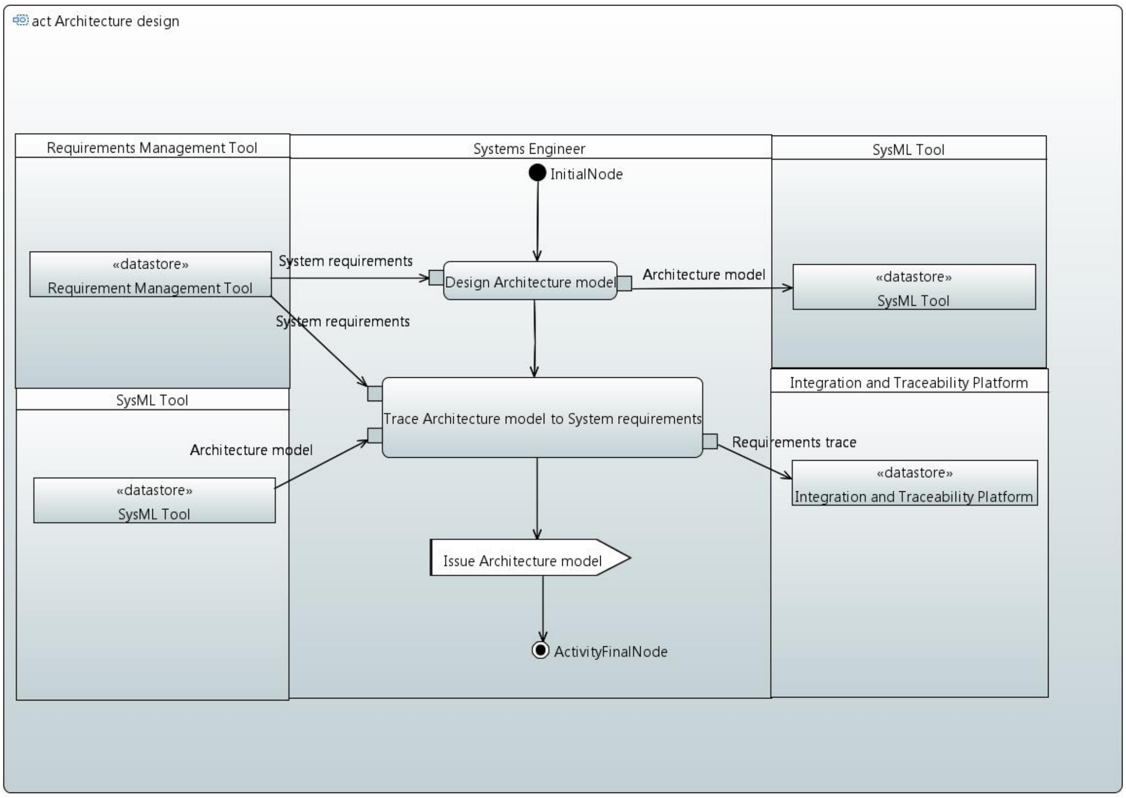


Figure 4.8. *Architecture design workflow (Alanen, et al., 2015).*

After the tracing is finished, architecture model is issued within ITP. The SysML tool used for designing the architecture model is Papyrus, which is accessed through Eclipse Mars (Eclipse, 2016). Papyrus is used because it offers support for modelling languages such as SysML. The designed main subsystems of the MEWP are shown in a package diagram in Figure 4.9.

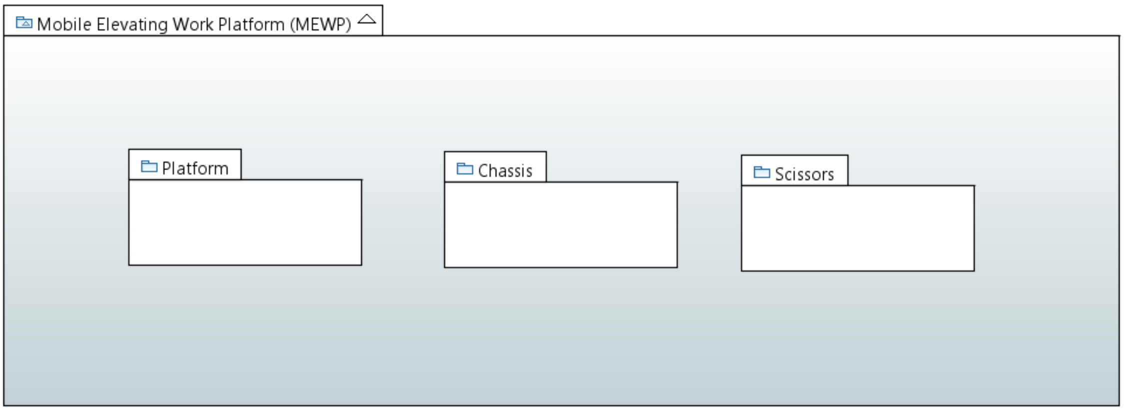


Figure 4.9. *Package diagram of the main components of MEWP.*

The main focus of the thesis is on the Platform subsystem, which the new customer feature is allocated to. The other two subsystems are Chassis and Scissors. Together these subsystems represent the whole mechanical structure of the MEWP. Naming of the subsystems stays the same throughout the thesis in order to make requirements traceability favourable and easy to follow. The logical architecture of the Platform is conducted from the package diagram and is captured in the coming sections by a block definition diagram and use case diagrams. After the logical architecture is defined and the appropriate requirements are satisfied, a logical model is created with the help of SysML block definition diagram. The functionality of the logical model is explained with SysML use case diagrams.

4.4.1 Block definition diagram

A block definition diagram (BDD) offers a decomposition representation of a system. The structural information of a system is communicated by realising a structural aspect of a model of a system, and illustrating the conceptual blocks and the relationships between them. In this thesis, the logical model is represented as a system decomposition diagram with BDD notation.

Blocks can define a type of logical or conceptual entity for instance; a physical entity, a person, a facility, or an entity in the natural environment. A block includes a description of a set of similar objects, or instances, all of which possess common characteristics. A set of features is included in a block that describes the characteristics of its instances. Behavioural features define how a block interacts with its environment, and structural features define the internal structure and properties of a block. BDD contains the contents inventory of the system and quantity of each element. The model element type can be a block, a package, or a constraint block. (Friedenthal, et al., 2012).

BDD notations are used to represent the changed platform mechanics of the MEWP. Figure 4.10 shows the decomposition diagram of the updated platform mechanics that satisfies the new customer feature.

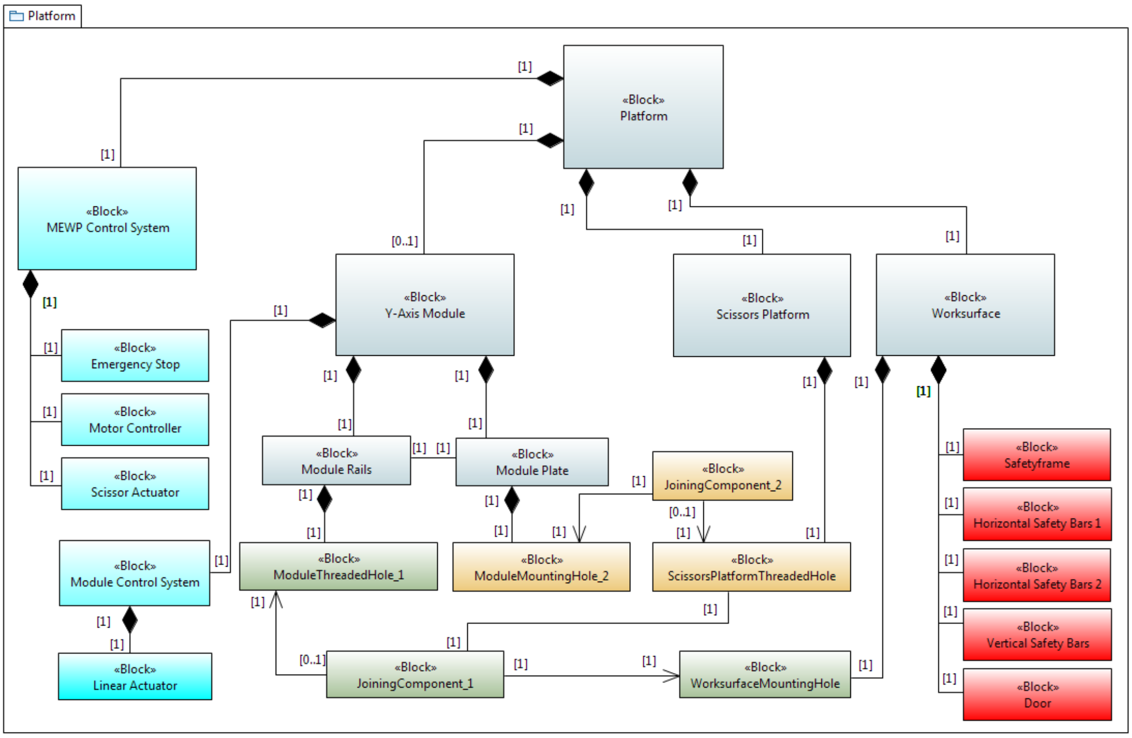


Figure 4.10. Decomposition of updated platform mechanics.

Blocks are included in the feature model, called Platform. The block symbol is a rectangle that is divided into a set of compartments. However, the name compartment at the top of the symbol is the only mandatory one. Other block features such as operations, value properties, and ports, are optional and can be represented in other compartments. To reduce complexity and to make the diagram more suited for the consensus of this thesis, the compartments of the block definition diagram were reduced to the name compartment.

The new platform covers an approximation for the new technical feature and the external systems that the feature interacts with. Platform is the top-level block and it provides the context for the new feature. Platform is composed of four main components; Y-Axis Module, Scissors Platform, Work surface, and MEWP Control System. Composition of blocks is indicated by the black diamond symbol pointing outwards of the block that composes it. The relationship between two blocks is a whole-part association; diamond end of the line describes the whole and the other end the part. The part end of the association shows the part property owned by the block in whole end of the association. Numbers on the line represents multiplicities that indicate existence of whole and part instances in the other ones end. A value of 1 at the whole end means that an instance of a part may only exist in one whole at any time. The other way around, a value of 1 at the part end means that an instance of the block at the part end cannot exist if no whole exists. The logical model in Figure 4.10 follows the mentioned use of associations and multiplicities.

The horizontal movement feature of the platform is made possible by a modular solution. The feature can be added according to customer requirements to the structure of the MEWP by including the y-axis module between the sub platform (Scissors Platform), where the scissors are attached to, and the work surface. The module includes a module plate that moves on a frame of rails. Actuation of the module is executed with a linear actuator, but the type and more specific method is out of the scope of this thesis. Because the module is optional, multiplicity at the part end of the composite association is 0...1. Also the associations regarding the attachment of the module are marked as 0...1. These connections are unidirectional directed associations which create a reference property, typed by the target block, in the source end. Joining of the module is achieved with joining components that will be mounted to the mounting holes and threaded holes in work surface and scissors platform. If the module is not used, the scissors platform is directly connected to the work surface. How joining is accomplished and what kind of a joining component would be used is not determined in this thesis.

MEWP Control System composes of Emergency Stop, Motor Controller, and Scissors Actuator, which are needed regardless is the MEWP controlled from the platform or the chassis. The horizontal movement of the platform is managed through Module Control System composing of Linear Actuator. Because this thesis concentrates on simulation driven mechanical designing, the logical architecture model is mainly created to support CAD modelling and mechanical designing. Therefore, the cyan coloured control systems, in Figure 4.10, are left outside of the acute scope.

An important aspect of designing the MEWP is work safety. Special attention is put to Work surface where the relevant operators are working. Figure 4.10 shows Work surface consisting of Safety frame, Horizontal Safety Bars 1 and 2, Vertical Safety Bars, and Door. The above mentioned components are marked as red in BDD, because their structure is considered here as atomic: these components, or atomic blocks, cannot be decomposed and they do not have an internal structure. The atomic nature is arguable also from the viewpoint of the MEWP manufacturer, because these components are bought as whole from the subcontractor and they have only one spare-part number.

However, the properties of an atomic block are still relevant for the description of the behaviour of a block (Matei & Bock, 2012). For instance, Door is defined as a component that does not compose of any other components. In reality, door would have a locking mechanism etc., but for the sake of this thesis a simplification was made to ease the CAD designing.

4.4.2 Use cases diagram

Use case modelling was chosen as the approach to analyse the customer feature request. Although use cases are not the best way to demonstrate a hierarchical structure of a machine, since use cases do not include hierarchy, the modelling method was used here to illustrate the different functional levels of the MEWP. Different functionalities can be seen for instance as a specific use case for a specific subcontractor. Practices applied for use case modelling were inherited from UML and SysML.

Use case diagram defines the actors and use cases from a usability perspective. Diagram identifies interactions between actors and the system. Users of the system are identified as actors which are used to represent human, organization, or any external systems that interact with the use case diagram system. The main purpose is to define the system functional scope among its various users. Use case description identifies goals for the use case, number of variant use, and a main pattern of use. Typically, a use case covers many scenarios that vary because of different circumstances. Presented use cases and operations denote captured system requirements referred in Section 4.3.1. In other words, system requirements are depicted in terms of the uses of the system. (Friedenthal, et al., 2012)

In this thesis use case diagrams are used to represent interactions between the actors and MEWP, as shown in Figure 4.11.

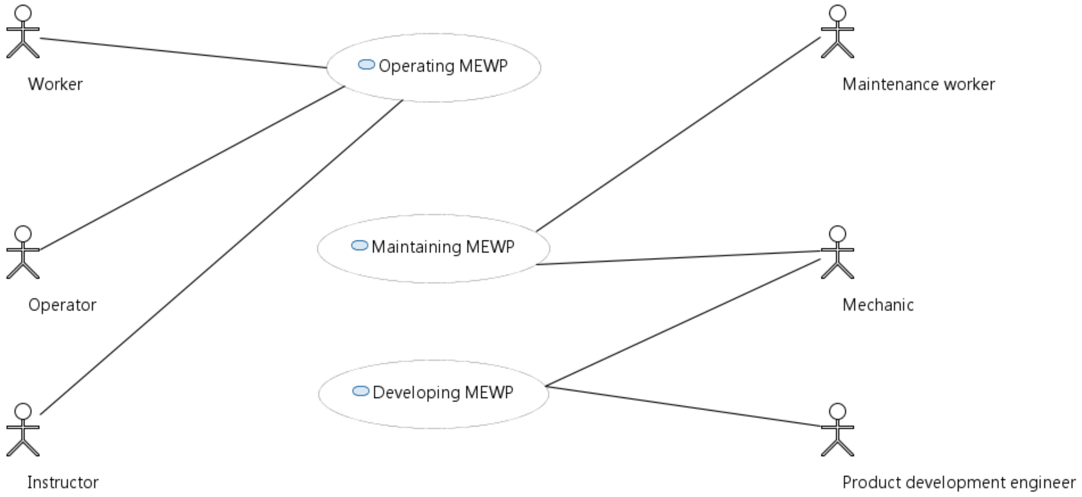


Figure 4.11. *Representation of the relationship between different actors and the MEWP System.*

Actors related to the MEWP System use case are Worker, Operator, Instructor, Maintenance worker, Mechanic, and Product development engineer. Interactions between actors and the use case are called communication paths which represent associations with some restrictions. For instance, association cannot be a composite type, since actors and use cases are regarded at the same level. Neither use cases nor actors may own properties, which is why associations do not have arrows to point the direction of correlation. Multiplicity of the association end is by default 0...1, if not shown. In this thesis, multiplicities are not used among the use cases to describe the number of actors associated with each use case or to describe the number of use cases the actors can be involved with. This was estimated as an unnecessary task regarding the scope of the thesis.

MEWP System has three use cases; Operating MEWP, Maintaining MEWP, and Developing MEWP. Each of them represents an activity during the life cycle of MEWP and could be regarded as individual processes (Weilkiens, 2006). The main focus is set on Operating MEWP, because it brings forward the main functionalities involved with the platform and the new required feature. Operation of the MEWP is carried out by worker, operator and possible instructors. Difference between operator and worker is that worker operates close to the MEWP and might be involved with the use of MEWP even though he is not the primary operator. In SEAModel Worker is not directly linked to the use case, but through a use case supplement. Maintenance and repair of MEWP is associated with maintenance worker and mechanic. Although, mechanic is involved with maintenance work his main responsibility is the assembly of MEWP and testing of the final product. Thereby, he is also concerned with the development of MEWP with the product development engineer.

Use case Operating MEWP, depicted in Figure 4.12, can be considered as a general use case that has two special cases; Operate MEWP from Platform and Operate MEWP from Chassis.

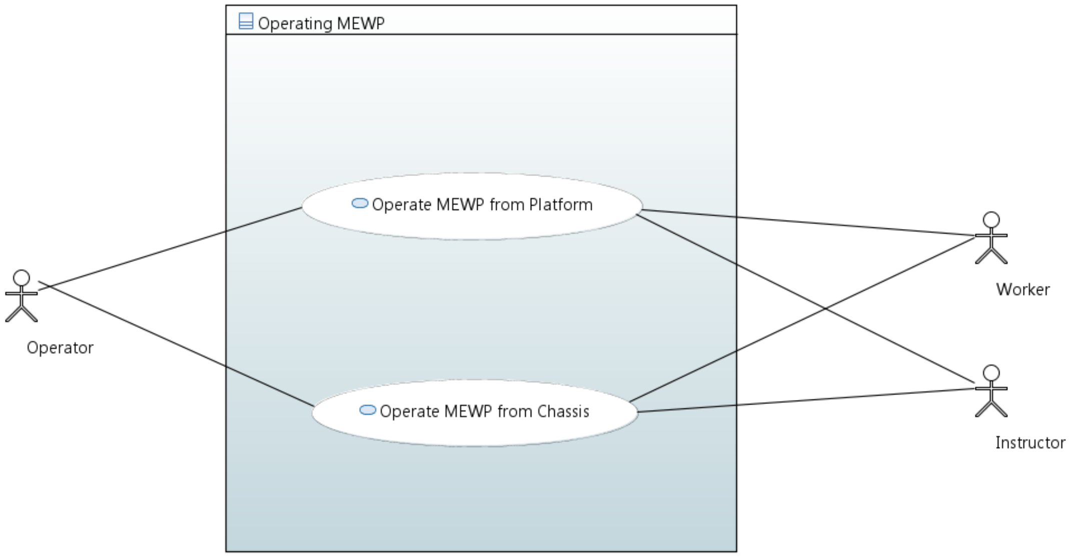


Figure 4.12. *A use case for describing Operating MEWP.*

MEWP can be controlled from platform and chassis as mentioned in Section 4.2. The use cases in Figure 4.12 are meant to define the level of control on a more specific level by visualising operations each actor can accomplish. Diagram in Figure 4.13 explains the control options possible to carry out from the platform.

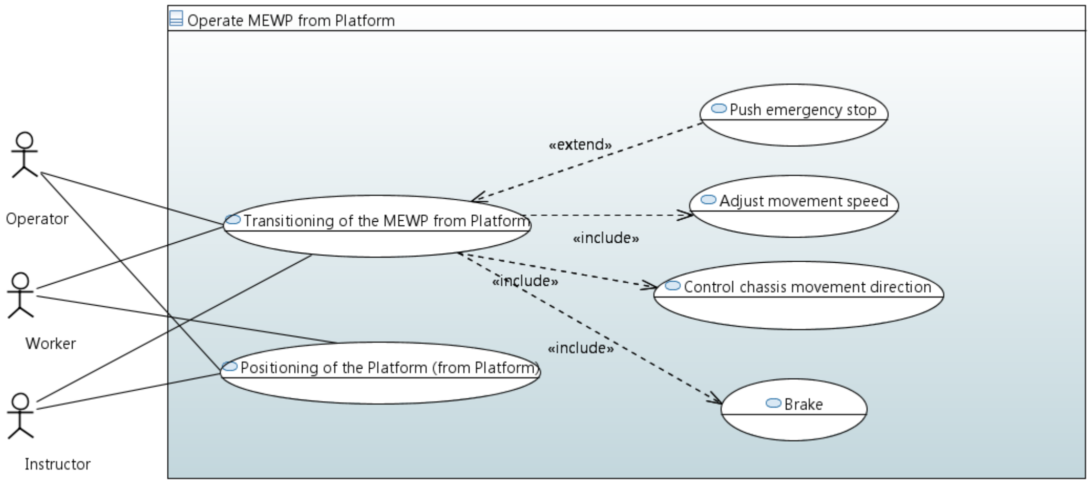


Figure 4.13. *A set of use cases for Operate MEWP from Platform.*

The relevant actors can control either the transition of the MEWP structure or vertical position of the scissors platform through included functionalities. Adjust movement speed, Control chassis movement direction and Brake are called included use cases that allow Transitioning of the MEWP from Platform, the base use case, to include their functionalities as part of the base use case when performed. Included use cases are always performed when the base use case is performed. Actors related with a base use case need not to be separately associated to any included use cases, because of the implicity of the inclusion. Instead of representing a functional decomposition of the base use case, included use cases describe common functionalities that other use cases may include. A use case can also be extended from the base use case with an extension relationship. (Friedenthal, et al., 2012)

Extended use case represents a fragment of functionality that does not contribute directly to the goal or outcome of the base use case, thereby not being part of the core functionality. For example, in the case of Operate MEWP from Platform in Figure 4.13, the extension case Push emergency stop describes a behaviour that is not commonly applied in the base use case, Transitioning of the MEWP from platform, in order to affect the movement of the MEWP. The base use case can be extended to a set of extension points which support the extensions and indicate to the extensions where in the base use case it can occur.

The use case Operate MEWP from Chassis is similar to use case Operate MEWP from Platform. Chassis controller is used for transitioning the MEWP and for controlling the motor as shown in Figure 4.14.

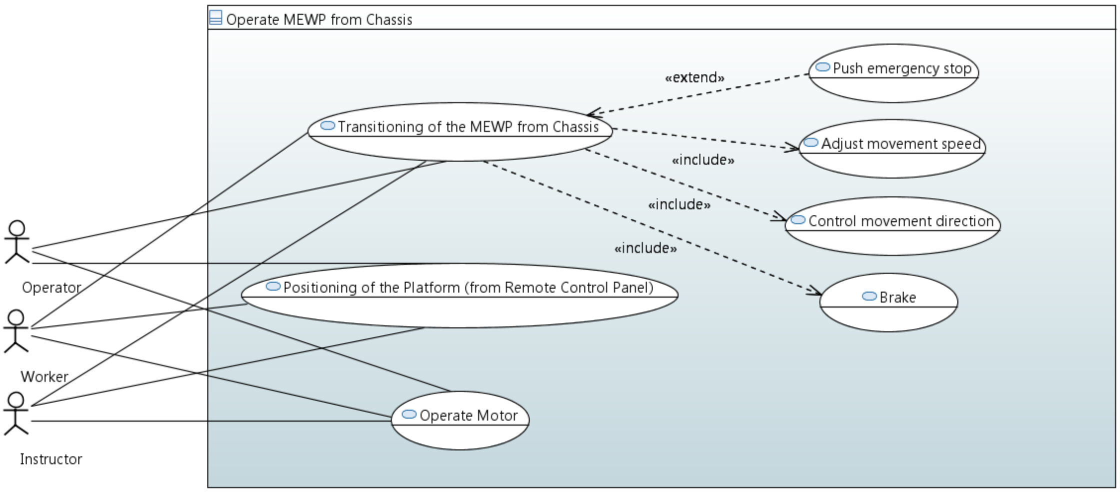


Figure 4.14. *A set of use cases for Operate MEWP from Chassis.*

The use case Operate Motor includes the basic functionalities of a motor; Turn motor On and Turn motor Off. These cases are depicted in Figure 4.15 and are only included in the chassis controller.

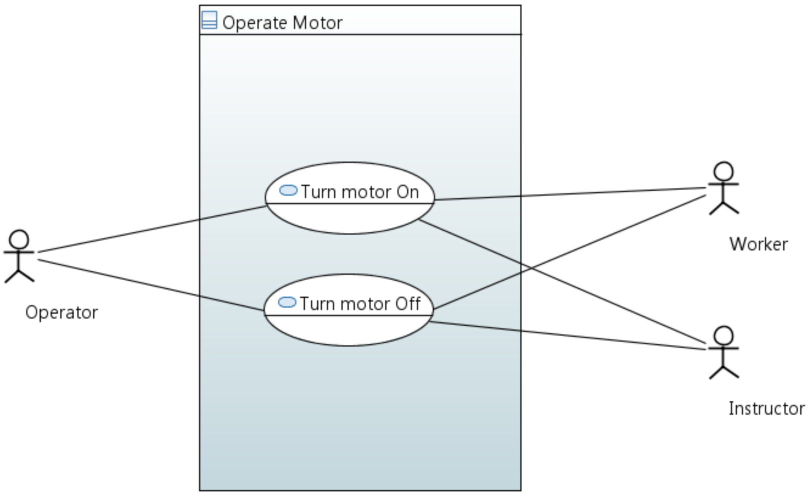


Figure 4.15. *A use case for operating motor.*

The position of the platform is possible to be controlled both from the platform and chassis. From the platform this is done directly with a control panel and from the chassis with a remote controller. The diagram of the use case Positioning of the Platform is shown in the Figure 4.16.

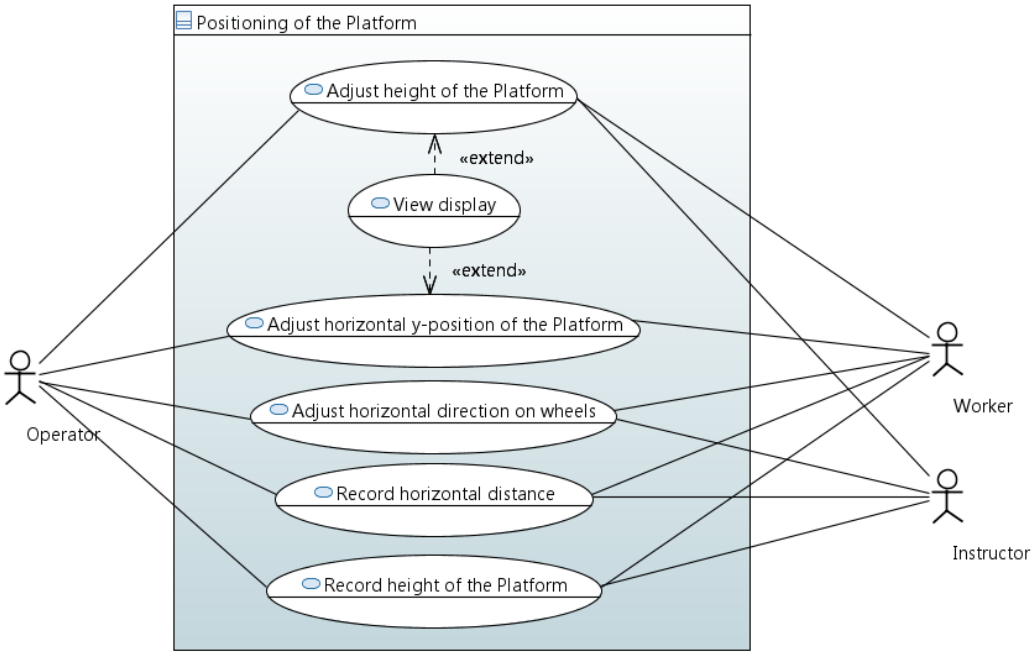


Figure 4.16. *A set of use cases for Positioning of the Platform.*

Positioning of the Platform represents a set of use cases that include; Adjust height of the Platform, Adjust horizontal y-position of the Platform, Adjust horizontal direction on wheels, Record horizontal distance and Record height of the Platform. The first two use cases explain how actors can adjust the height and the horizontal position of the platform by controlling the scissors and linear actuators. Positions and distances can be monitored with a display. View display is an extended use case of the base use cases Adjust height of the Platform and Adjust horizontal y-position of the Platform. If the MEWP is incorrectly situated in horizontal direction regarding the object of work, actor can adjust the position with precise micro movements. Finally, the last two use cases concern recording of the height and horizontal distance of the platform in x and y direction. The recorded parameters enable MEWP to be automatically operated to the same position in the future.

4.5 Detailed design – physical model

The physical model is created by a mechanical engineer according to the logical architecture model and the system requirements provided by the systems engineer. The system requirements are acquired from the DOORS requirements management tool. Actual CAD modelling starts after the logical architectural model is received. The goal of the CAD model is to satisfy with a physical representation the specifications and demands set by the customer and systems engineer. The model is produced with a CAD tool, which is chosen to be SolidWorks for the demonstration. SolidWorks was chosen because of its integration possibilities with MATLAB and Simulink. The CAD model elements are traced with ITP to the corresponding logical architecture model and system requirements before the model is issued. The workflow of the detailed physical model design is shown in Figure 4.17.

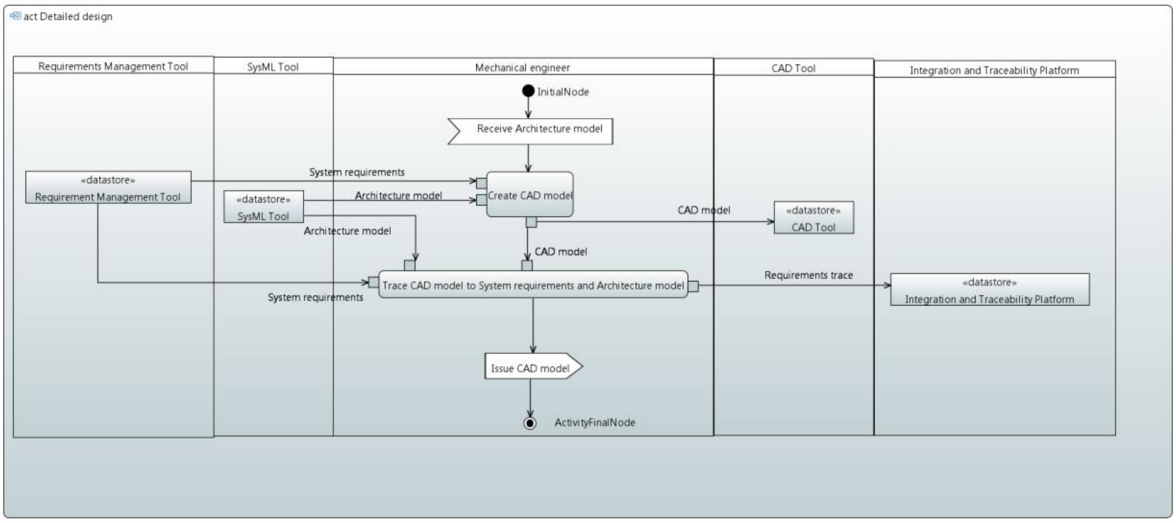


Figure 4.17. *Workflow of the physical model design (Alanen, et al., 2015).*

The study of the MEWP structure is based on the previous work of the SIMPRO project in which the first version of the CAD model was developed. The model presented in Figure 4.18 was created for the visualization purposes of the Scissors platform – systems engineering desktop (see Section 4.2), and the original model followed stakeholder requirements derived from standards SFS 12100 and EN 280 accordingly.

Figure 4.18. *Original design for the MEWP.*

However, the original model lacked correct dimensions and inertia characteristics, which are critical not only for the realistic and plausible presentation but also for the simulation model. For instance, the diameter of the hollow scissor beams was too small and the wall thickness too large. The hydraulic cylinder used to lift the scissors structure was also inadequate to lift the platform to the demanded 8 meters. Additionally, the overall mass of the MEWP was over 3 700 kg, which is relatively high compared to similar lifts with the same qualities on the market (Genie lift , 2014), (Ranolift Oy, 2016) (JLG, 2016).

4.5.1 Updating the physical model design

According to the specifications mentioned in Sections 4.2.2, 4.4.1and 4.4.2, the updated mechanical structure is a modular ensemble. The original platform is modified into two different sub platform structures that enable the new add-on. The new feature that enables horizontal y-axis direction shown in Figure 4.19 is a frame component that can be connected between the sub platforms. The main scope of the new design was to serve the requirements traceability, which is why the frame component was done with a simplified design.

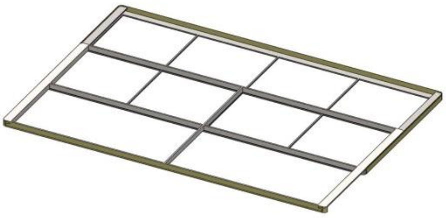


Figure 4.19. *The y-axis module for enabling the horizontal movement.*

The actuation and control systems of the module are situated in the scissors sub platform and work surface making the preliminary dimensions of the module 2 640 x 1 220 x 50 mm. A more detailed design and actuator selection was out of the main scope of this thesis as was the strength calculus for the frame component. Furthermore the objective was to create a plain component that could provide the y-axis motion. The y-axis module consists of plate and railing systems as was defined in the BDD in Section 4.4.1. The components are designed to fit within the platform structure.

To satisfy all the stakeholder and system requirements, the mechanical structure of the MEWP needed modification. Hydraulic cylinder of the scissors structure was redesigned to be a two-stage telescopic cylinder and the structure of the scissors was designed to match inertia of about 600 kg. Whether the designed cylinder was powerful enough for the 900 kg workload or not, was not considered into the scope of this thesis.

The platform level including the scissors platform, y-axis module and work surface were designed to match inertia of about 460 kg. The chassis of the MEWP was not modified except its inertia was overridden to 1 320 kg to match an approximation of the case structure, diesel aggregate, hydraulic motors and electronic system. Thereby, the overall weight of the system is around 2 400 kg. The weight of the mechanical structure was calculated with the material characteristics of plain carbon steel, which was the main material of the structure. Updated version of the mechanical structure is depicted in Figure 4.20.

Figure 4.20. *The updated mechanical structure of the MEWP.*

Some aspects of the architectural model were left outside of the physical model. Specifications like door and joining mechanism between platforms were not modelled because they were not critical for the scope of this thesis. However, the methods used in SolidWorks to constraint (aka. a mate) platforms and other assemblies together are important. Each constraint defines a kinematic relationship between connected parts. Examples of constraints are angle, mate and insert, which specify degrees of freedom (DoF) between parts. DoFs enable parts, for instance, to rotate with respect to each other about a certain axis. Well defined constraints and rigid connections in the CAD model make the simulation modelling easier to manage as decisions in CAD modelling are reflected to the simulation model. Whether the simulation model satisfies the desired needs or not and how much additional work needs to be done in the simulation case, depends on the decisions done in CAD modelling. Therefore, CAD modelling raises simulation requirements (see Table 3.1).

4.6 Simulation model of the physical model

The characteristics and details of the CAD model are inspected before further actions are carried out. The CAD model is verified according to the simulation requirements that are derived from the corresponding system requirements and architecture model elements. Simulation requirements are a special case of the verification requirements, because the verification process in this thesis is managed with simulations. Simulation requirements are created by the systems engineer for the verification of the CAD model not for the actual CAD model itself. Hence the simulation requirements are classified as process requirements. For instance, as loss of stability is identified as a potential hazard for the MEWP during the verification process, it poses a need for a verification requirement to check the stability of the structure with a simulation tool. Before issuing, the simulation requirements are traced with ITP to their relative system requirements and architecture model elements. The work flow for the verification process is shown in Figure 4.21.

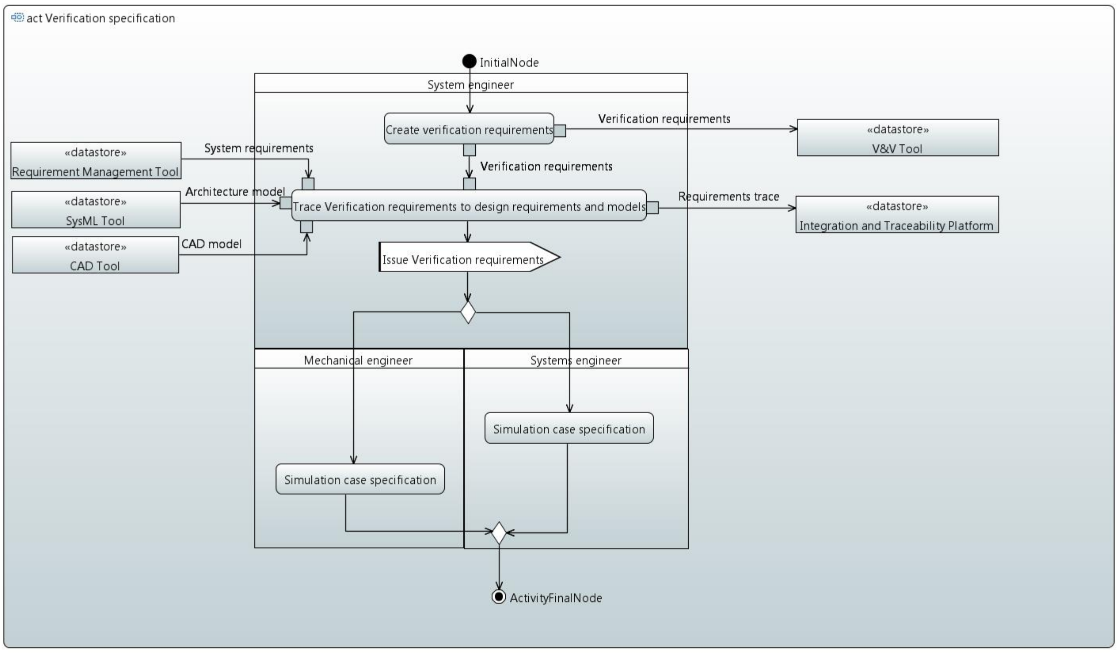


Figure 4.21. *Workflow of the Verification specification (Alanen, et al., 2015).*

The issued simulation requirements are used to create the simulation case specifications that define the rationale for the simulation case, the exact specification of the simulation steps, list of tools to perform the simulation case, circumstances of the simulation and expected results of the simulation. These specifications are set to test the model in the environment or situation, in which the observed feature can manifest. In this thesis, the simulation case specification defines the height and distance of the platform, general position of the MEWP and workload. Furthermore, standard SFS-EN 280 defines a set of wind conditions, manual forces and distributions of loads and forces that together create the conditions of minimum stability and unfavourable stresses. The simulation case specifications are stored to the verification and validation tool by the mechanical or systems engineer before tracing the simulation case specification at ITP to the verification requirements and to the CAD model elements under simulation. Finally, the simulation case specifications are issued in the end of the verification process workflow as shown in Figure 4.22.

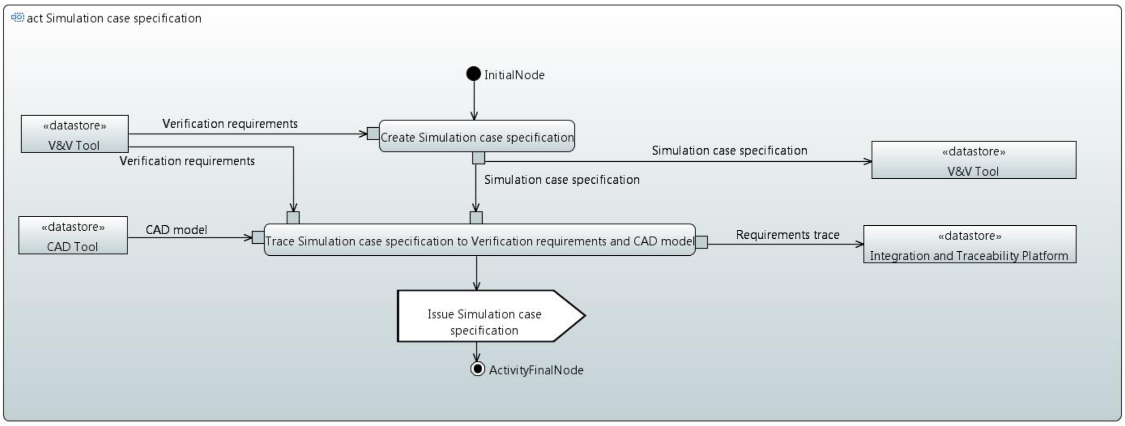


Figure 4.22. *Details of the simulation case specification presented in Figure 4.21 (Alanen, et al., 2015).*

The simulation case specification completes the verification specification of the CAD model. Because of The CAD model, the simulation case specification and the system requirements the verification process is able to be executed in the form of a simulation model. After the simulation model is verified the updated CAD model the results are additionally confirmed with static calculations. The software used for the calculations and the process itself is external to SEAModel and is only included in the thesis to confirm the results of the simulation.

4.6.1 Stability of the design – simulation analysis

The simulation model is developed from the CAD model by a mechanical or a test engineer to verify and validate the requirements. The model is done according to the simulation case specification and traced with ITP to the CAD model and to the simulation case specification. After being issued, the simulation model is executed and the results are recorded to the verification and validation tool within the database or ITP. Simulation results are traced with ITP to the simulation case specification and to the simulation model before being issued. Verification execution follows the Create and run simulation work flow presented in Figure 4.23.

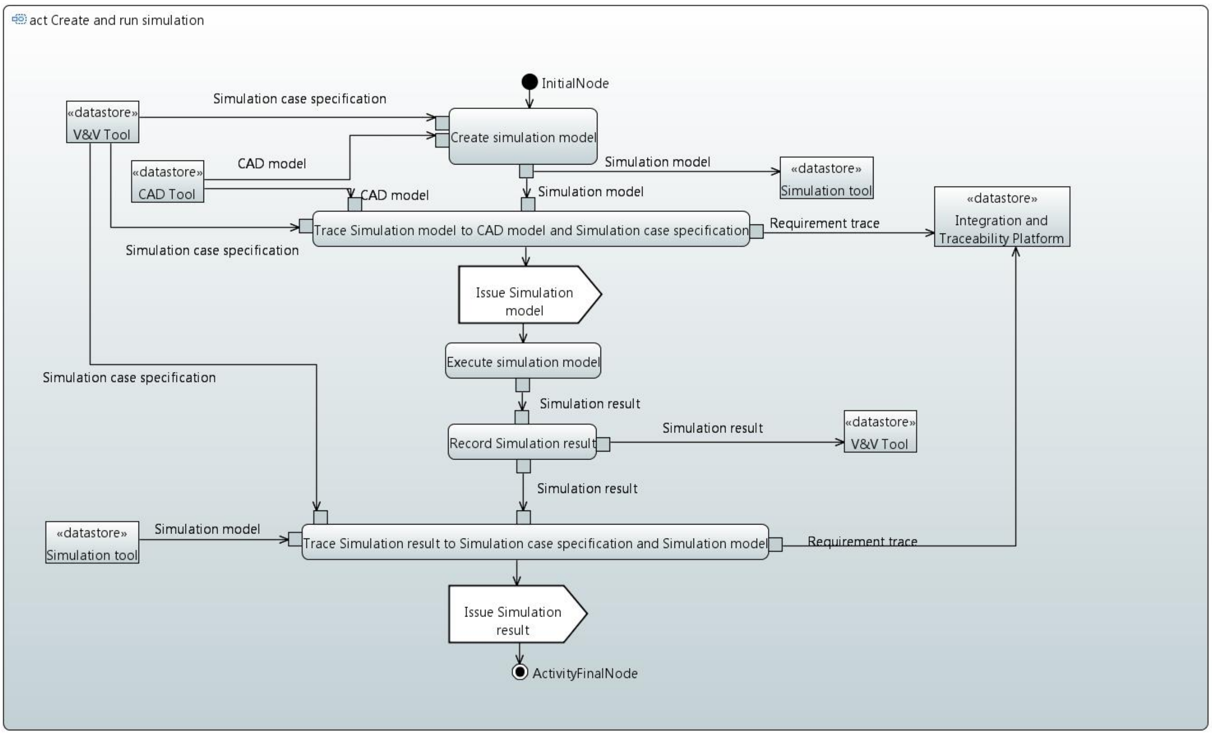


Figure 4.23. *Verification execution workflow for the Create and run simulation (Alanen, et al., 2015).*

The simulation software used for the model is SimMechanics second-generation. Compared to the first-generation, the second-generation technology has a simpler modelling paradigm with a new block library and more advanced visualisation utilities. Visualisation capabilities were considered important regarding this thesis, hence second-generation technology was preferred. However, the first-generation software should be used if the model requires time-varying constraints or variable mass. Both modelling technologies run within the Simulink environment and interfaces with MATLAB. SimMechanics applies the standard Newtonian dynamics of forces and torques in the mechanical systems of rigid bodies connected by joints. A system is represented by a connected block diagram in which translational and rotational motion is simulated in three dimensions.

Similar to Simulink, mechanical structures can be represented with components organized into hierarchical subsystems. SimMechanics has a plug-in possibility with SolidWorks providing an interface for direct exporting of CAD assemblies into SimMechanics. The plug-in enables one to use the body specifications like inertial properties, constraints and coordinate systems defined in SolidWorks within SimMechanics. Although sensors and actuators are automatically provided according to the CAD model, they might be wrongly situated or chosen if constraints are not accurately defined in the CAD model. This is especially difficult with complex models that include many different constraints. SimMechanics can successfully import angle offset, mate and insert constraints. However, if they are used to restrict motion between specific distances SimMechanics does not understand how to interpret this as a joint. For instance, in

SolidWorks if one limits the translational movement of the MEWP between 0…75 cm and exports the model into SimMechanics, the constraint is replaced by unnecessary joints that merely increase the complexity of the simulation model. Therefore the distance restriction should be added on the SimMechanics side.

Stability loss of the MEWP is simulated with a model in which the observed tipping direction is the direction of the extended work platform. The simulation model consists on the top level of three main subsystems, Chassis, Scissors and Platform. Additionally, there are two subsystems for two toolkits and two operators, which together compose the maximum 900 kg workload, set as a requirement. More precise calculations as to how the workload is distributed are presented in Section 4.6.2. Figure 4.24 shows the top level of the simulation model with the before mentioned subsystems. With the exception of Platform, all the subsystems are made rigid to measure only the stability of the MEWP while the platform is moving and to keep the model complexity as low as possible. Rigid components do not have joints or sensors, and they merely support the integrity of the ensemble.

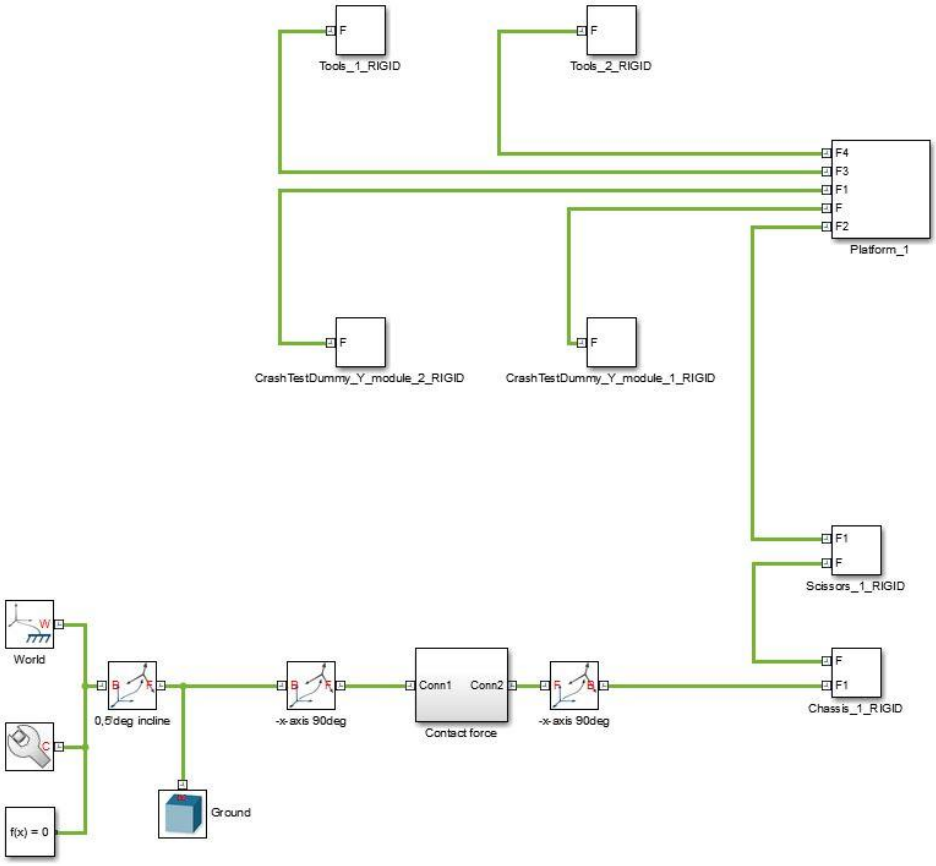


Figure 4.24. The main level of the simulation model for the MEWP.

The first three blocks in the left corner of Figure 4.24 are common in most SimMechanics second-generation models. World frame, mechanism configuration and solver configuration provide the basic information for the mechanical model, such as the predefined right-handed coordinate frame, uniform gravity and model solver instructions.

These blocks help set up the coordinate systems of different points on various bodies, which specify the local axis and origins for actuation and sensing.

The MEWP is positioned according to the SFS-EN 280 in the most unfavourable position with the maximum allowed inclination of the chassis. The machine is also under wind loads and side forces like manual force, which try to tip the structure. The tipping lines are determined to be at ¼ of the tyre ground contact width from the outside of the ground contact width (SFS-EN 280, 2013). The wind forces are assumed to act horizontally, and they are applied at the centre of area of each structural component, person and equipment with a mechanism shown in Figure 4.25. The external force subsystem is located in the immediacy of each component, person and equipment subsystem.

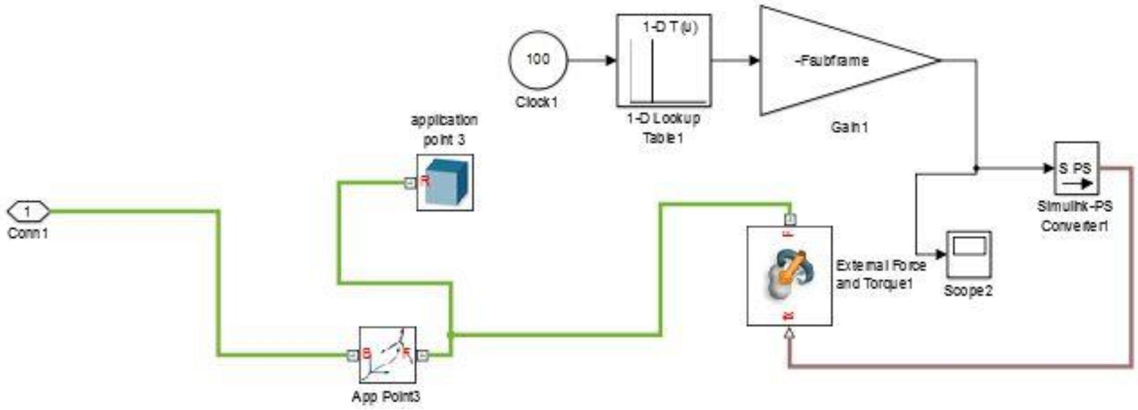


Figure 4.25. *Subsystem for the external force.*

The external force is added to an application point with an interpolated lookup table set by a timer. Depending on the shape exposed to the wind, the force is multiplied with a shape factor. Manual forces applied by persons on the platform are also calculated with the same mechanism, but multiplied with a predefined factor of 1,1. Compositions of external forces are specified in Section 4.6.2.

In order to simulate the possibility of MEWP to loose stability and tip over, the structure needs to have a connection with a fixed reference frame e.g. ground. Without any motion resistance, the structure would float in a void where the effects of gravitational force are hard to measure. Also, to measure the point of stability loss, the structure first has to be realistically supported. The contact force between the wheels and the ground is achieved by the coefficient static friction, which ties the model in the horizontal direction. In the vertical direction, contact is treated as a spring-damper force that affects when two frames reaches a specific threshold. Friction and damping force together create a static equilibrium, which generates a sense of support.

In Figure 4.24, the contact force is adapted to the simulation model with a “Contact force” subsystem that utilises hard-stop blocks by varying the spring and damper coefficients, and a friction element coupled to the respective degrees of freedom. The joint used between the contact frames, the tipping line of the chassis and the ground, is a planar joint that works similar to a hinge. The planar joint allows translation in 2D plane and rotation around z-axis. The revolute z-axis simulates the tipping possibility of the

MEWP. The y- and x-axis are prismatic, but the y-axis, which is parallel to the MEWP, is equipped with a hard stop block whereas the perpendicular X-axis is with a translational friction block. The mechanism of the contact force with the hard stop and friction subsystems is depicted in Figure 4.26.

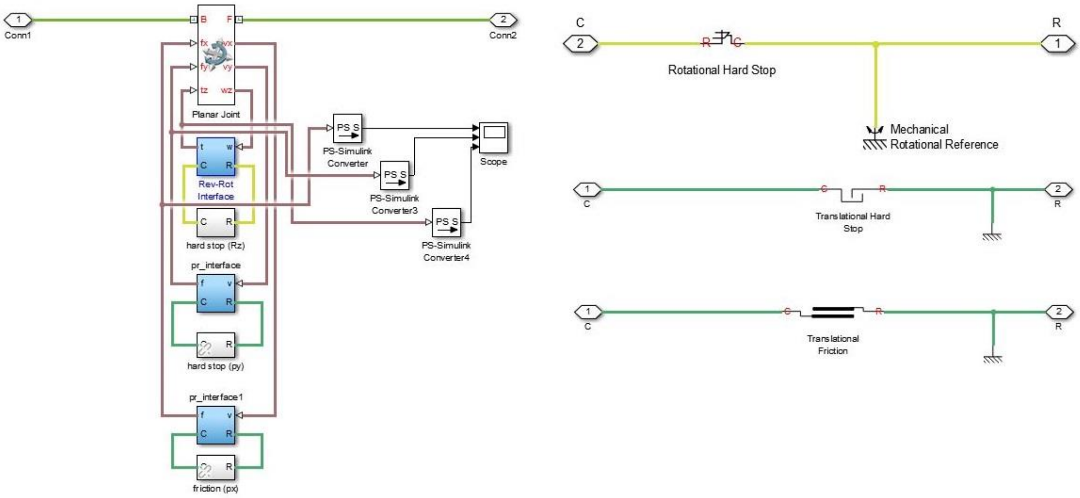


Figure 4.26. *Subsystem Contact force (left) with the hard-stop and friction subsystems opened up (right).*

The translational hard-stop block associated with the y-axis is parametrised to prevent the structure from moving in y-axis direction. Although this is a simplification of the model, it helps determine the stability in z-axis direction more easily with the current contact force mechanism. In an optimal scenario, the outer corners of the chassis would be conjoined with the ground by a 6-Degree of Freedom (DoF) joint enabling tipping into each direction. However, pursuing a more realistic contact force mechanism proved to be excessively difficult with the second generation technology of the SimMechanics software. The first-generation technology can define a model with 6-DoF and calculate the necessary support and contact forces independently. The calculation is done outside the SimMechanics solver components by individual math components. In second generation software, this was first attempted with its renewed components that executed calculations within themselves. However, these solver components lacked the necessary sub-blocks to support contact force calculations. To avoid this problem, the second attempt involved taking the position data outside the solver components in the same way it was done with the first generation technology. For reasons unknown, doing the calculations outside the SimMechanics solver with math components lead to unstable results.

The stability is studied while the work platform is actuated to the required distance. The work platform is the only moving element during the simulation regarding the mechanical structure. In SimMechanics, manipulation of the movement occurs in the y-Axis Module subsystem within Platform subsystem. Figure 4.27 shows how the actuation is accomplished with a prismatic joint between the module rails and plate.

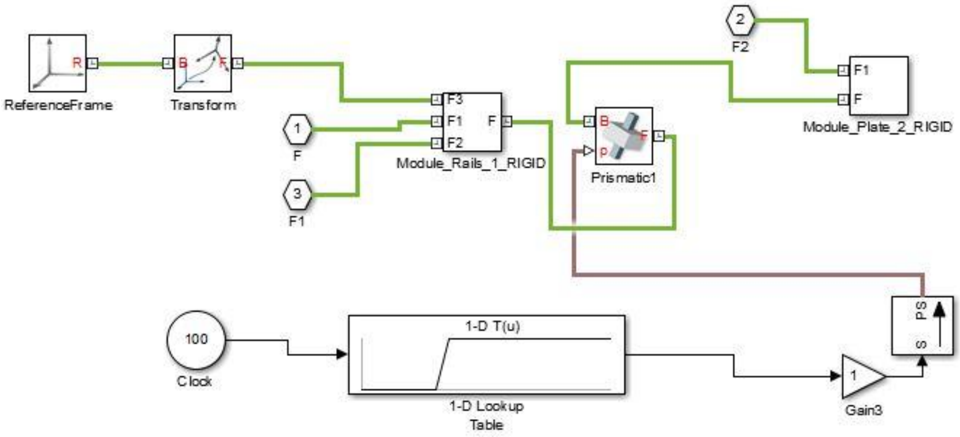


Figure 4.27. *Platform movement mechanism within the Y-Axis Module subsystem.*

The actuation is accomplished in a similar way as in the external force subsystems. The lookup table sets platform to move 75 cm from its initial position according to breakpoints that are initiated by a timer. Actuation initiates after 60 seconds, because the MEWP needs time to stabilise. Stabilisation is required because in the beginning of the simulation, contact forces cause the structure to shake as they take hold of the MEWP. When stabilisation is achieved, the external forces are added to the application points, and the work platform moves to the required position. The application points are depicted as red balls in Figure 4.28.

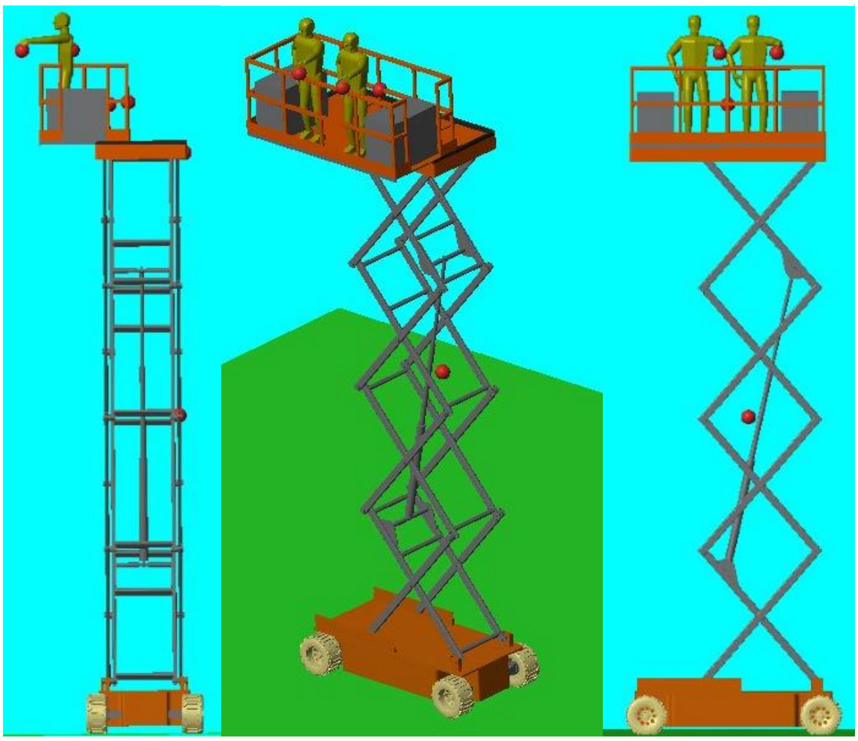


Figure 4.28. *MEWP with work surface in its final position in the simulation.*

Figure 4.28 depicts the MEWP in its final position at the end of the simulation. The simulation model proved the mechanical structure is stabile under the circumstances set by SFS-EN 280. The results are verified in the next Section 4.6.2.

4.6.2 Stability of the design – statics analysis

The statics analysis presented in this section refers to the Appendix B: Stability analysis of the MEWP. Characteristics of the MEWP were introduced in Section 4.2 in order to illustrate structural changes of the original MEWP by the new customer feature, and this section expands upon that information. This section covers stability calculus for the simulation model presented in previous Section 4.6.1.

The analysis expects the MEWP to be at its most unfavourable position with scissors and platform extended to the maximum distance. Furthermore, the overturning and corresponding stabilising moments are calculated according to the most adverse tipping lines. The applied tipping lines were introduced in section 3.5.1, and they were determined in accordance with EN 280 and ISO 4305. The maximum allowed inclination of the chassis was set to 0,5 degrees, which is the inaccuracy value normally added to the maximum allowable inclination. The set up for the static analysis is shown in Figure 4.29 in which Point A refers to the tipping line, M_p to vertical moment of the persons, M_e to vertical moment of the equipment, M_{ws} to vertical moment of the work surface, M_{sf} to vertical moment of the (scissors) subframe, M_{es} to vertical moment of the extending structure, M_{wp} to the horizontal moment of the persons, M_{we} to horizontal moment of the equipment, Ms to horizontal moment of the manual forces, and $M_{ws\_sf\_es}$ to the combined horizontal moment of the work surface, subframe and extending structure.

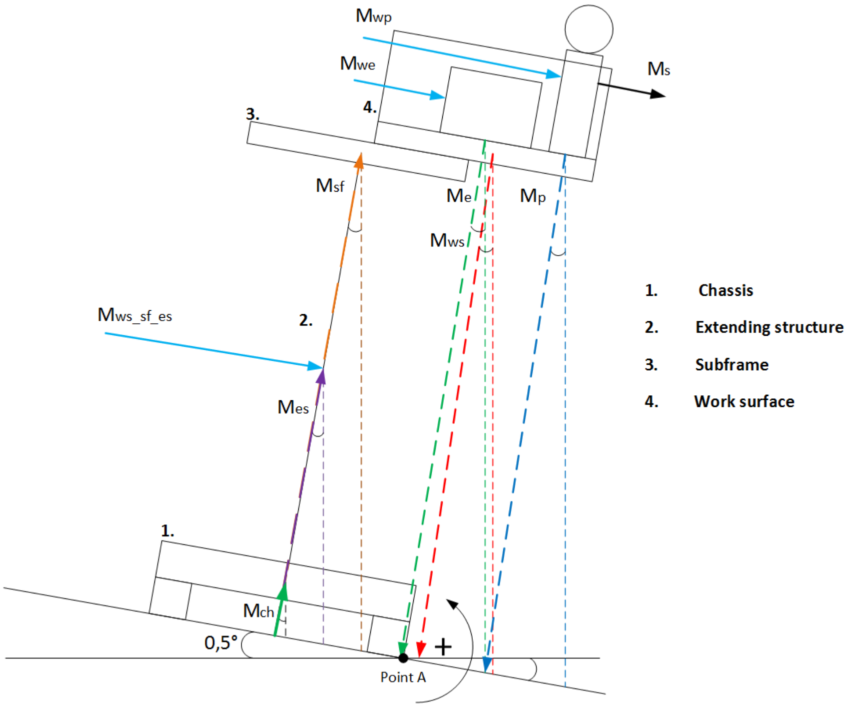


Figure 4.29. *Overturning and stabilising moments of the MEWP.*

The positive rotation direction around Point A is determined to be counter clockwise. The moments are calculated according to the masses or forces of elements.

The mass of a person is acquired from SFS-EN 280, which defines the standard weight to be 80 kg. Hence, the weight of a single toolbox is 370 kg if the whole workload is 900 kg (consisting of two person and two toolkits). Distribution of the weight and the location of the centre of mass of the toolboxes and persons are calculated according to SFS-EN 280. The mass of each person acts as a point load on the platform. Point loads are situated at a horizontal distance of 0,1 m from the upper inside edge of the top rail, and the distance between the point loads is 0,5 m. The mass of the equipment is also an evenly distributed point load that covers 76% of the floor area of the platform. 76% of surface area is equivalent to a pressure of 3 kN/m^2 allocated to the work surface. All the loads are assumed to be located in the most unstable positions where they result the most severe results. This assumption is done under the assumption that the masses of the components of the MEWP are static structural loads and are not moving. The rated load of persons and equipment are shown in Figure 4.30. The letter *n* in the figure refers to the number of persons and m$_p$ to the mass of a person.

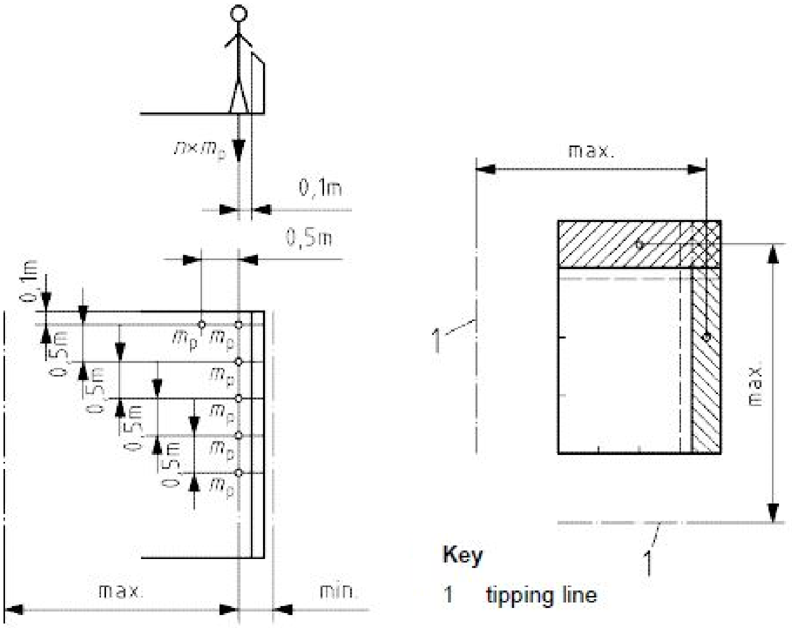


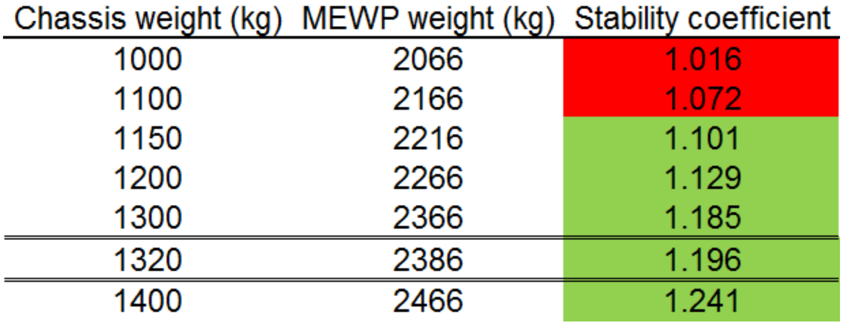
Figure 4.30. Rated loads of persons (left) and equipment (right) (Modified from SFS-EN 280).

The horizontal moments in Figure 4.29 are derived from the wind loads and manual forces. The manual forces are applied by persons working on the work platform while reaching out the platform or otherwise causing pulling or pushing horizontal forces. The inflicted manual force is 400 N for MEWPs designed to carry more than one person. The force is applied 1,1 m above the work platform floor and is communal. The manual forces are multiplied by a factor of 1,1 and are expected to act in the direction, which creates the greatest overturning moment.

The wind loads are assessed according to ISO 4302 and SFS-EN 280. The affecting wind pressure is 100 N/m^2, which is equivalent to a wind speed of 12,5 m/s. The created wind forces are assumed to be dynamic and act horizontally at the centre of area of components of the MEWP, persons and equipment. The area of persons affected by the wind is 1,0 m above the work platform floor. For equipment, the wind force is calculated as 3 % of their mass acting at 0,5 m above the work platform floor. The wind forces are multiplied by a factor of 1,1, and a special shape factor is applied according to the areas exposed to wind. With respect to the MEWP at hand, the used shape factors were 1,6 for L-, U-, T-, I-sections, which represent the shape of the whole structure, and 1,2 for large flat sections, which were applied to the toolkits. Shape factors were chosen in collaboration with the VTT researchers.

The overturning and stabilising moments are studied according to a stability coefficient. The stability coefficient is the relation of supporting and tipping moments. The lower the coefficient value is, the more unstable the system. According to the researchers of VTT an acceptable limit for a stabile system is 1,1. In Table 4.1 different stability coefficient values are collected. The calculations are done by varying the mass of the chassis, which was chosen to be the main variant (Appendix B).

Table 4.1. *Stability coefficient values by varying the mass of the chassis (Appendix B).*



Chassis weight (kg)	MEWP weight (kg)	Stability coefficient
1000	2066	1.016
1100	2166	1.072
1150	2216	1.101
1200	2266	1.129
1300	2366	1.185
1320	2386	1.196
1400	2466	1.241

Table 4.1 indicates the MEWP to be stabile when the chassis mass is over 1150 kg. In Section 4.6.1 the simulation model utilised the chassis mass of 1320 kg, which provides an acceptable stability coefficient value of ~1.2. The calculated total masses are equivalent to the masses of commercial MEWPs.

4.7 Traceability management in DOORS

Traceability is one of the essential aspects of satisfactory requirements management. In this thesis traceability of engineering artefacts is provided by Rational DOORS. Traceability management includes requirements traceability, which is a sub-discipline of requirements engineering. The traceability management tool ensures the origin of each requirement and provides bi-directional traceability between associated elements, such as simulation results and system requirements. In this thesis traceability is accomplished with the help of SVN tool that together with DOORS implement the Integration and Traceability Platform (ITP). Another considered alternative for SVN was SharePoint.

Most of the foreign tools such as SysML, CAD and simulation tools, used in this thesis, produce files only in a local workspace. To make the files available to all relevant participants and tools, files are committed to a version control platform. SVN keeps track of various files and document versions. The necessary documents from the different software tools such as SolidWorks, Papyrus and Simulink are brought together, so-to-speak, in SVN to enable the tracing process with DOORS.

DOORS implements tracing of engineering artefacts by linking the artefacts together. The links are trace relations derived from the artefacts relations of SEAModel. However, not all artefact relations can be considered as trace relations, as was noted in Section 3.2. The actual tracing of engineering artefacts to one another is depicted in Appendix A. To make the diagram more comprehensive, the traceability diagram can be simplified to include only the core systems engineering artefacts shown in Figure 4.31. The simplified model depicts as to how the validity of the safety requirement is studied with the CAD and Simulation models in relation to the new customer requirement. Notable aspect here is that all the artefacts are implemented as basic DOORS objects, except the model artefacts, which include external model files represented by surrogate objects in Doors. Surrogate object method is explained later in Sections 4.7.1and 4.7.3. Artefacts traced in the demonstration case are depicted with a red line that indicates the main loop of the tracing process.

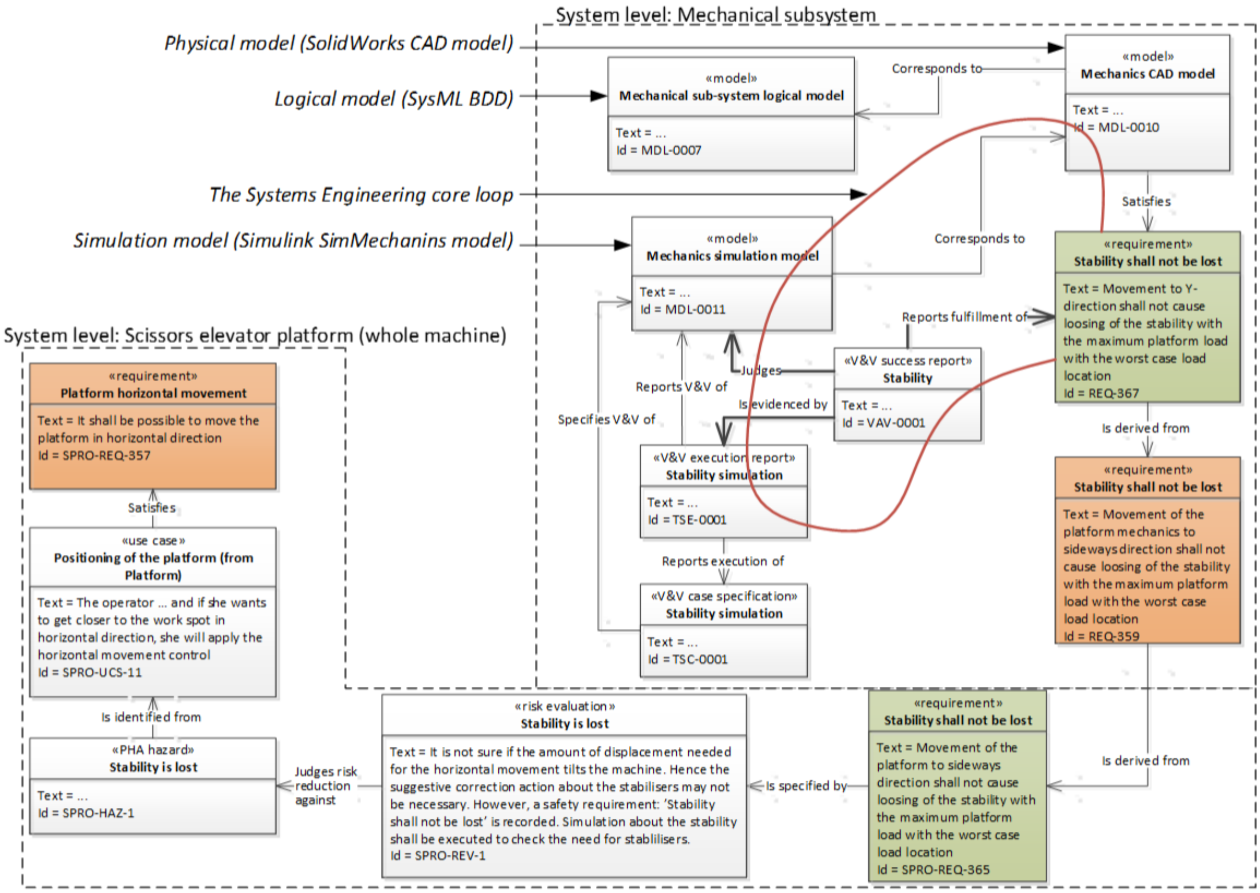
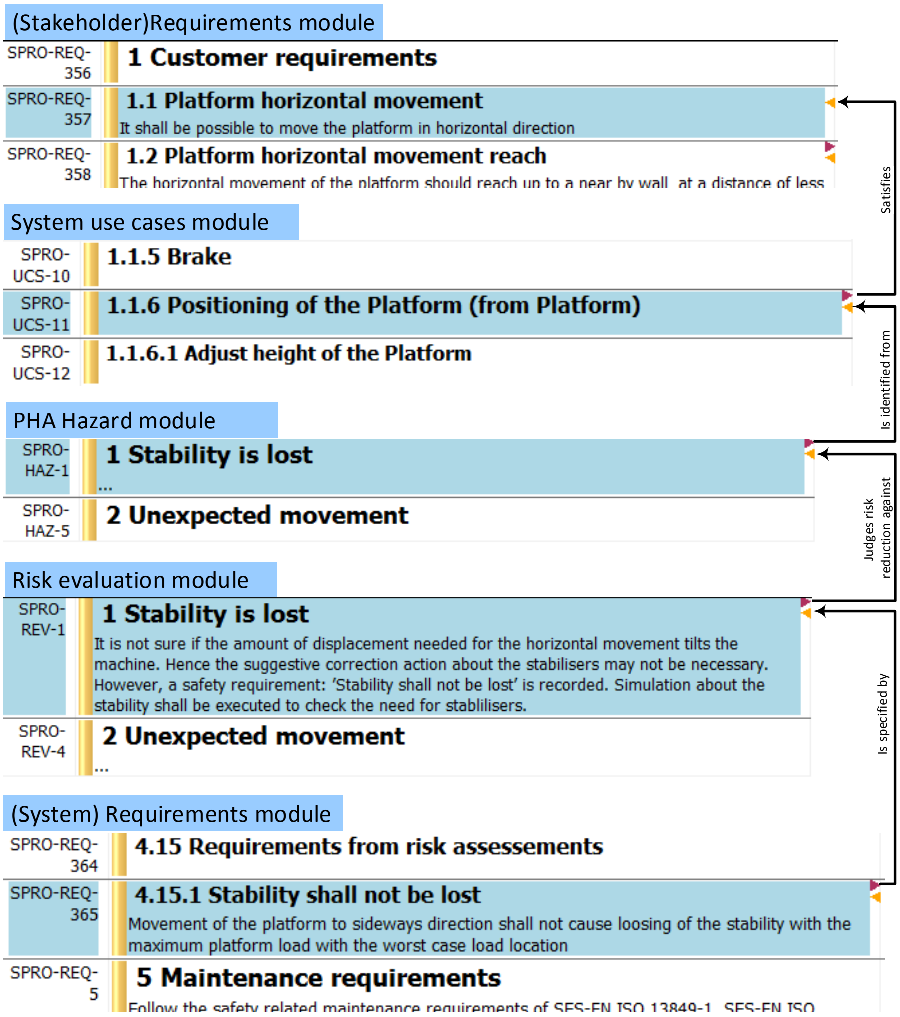


Figure 4.31. *Simplified Traceability demonstration model.*

The trace relations are implemented in Doors with a link module that is part of the inner tracing mechanism of DOORS. For instance, artefacts on the Scissors elevator platform level on Figure 4.31 are connected by relations; Satisfies, Is identified from, Judges risk reduction against, Is specified by, and Is derived from. In DOORS, these relations are trace links that connect objects of the corresponding formal modules together.

Figure 4.32 depicts how the Scissors elevator platform level of the simplified traceability demonstration model is implemented in DOORS with trace links. The trace chain is made visible from the original stakeholder requirement of the platform movement to the system requirement stating how the movement should not jeopardize the stability of the MEWP. Notable aspect here is that both stakeholder and system requirements are depicted with a Requirements artefact in the traceability demonstration model, and are implemented as different objects under Requirements module in DOORS. However, to emphasize difference, in Figure 4.34 and in Appendix A, stakeholder requirements are illustrated with colour red and system requirements with colour green.



(Stakeholder)Requirements module

SPRO-REQ-356 **1 Customer requirements**

SPRO-REQ-357 **1.1 Platform horizontal movement**
It shall be possible to move the platform in horizontal direction

SPRO-REQ-358 **1.2 Platform horizontal movement reach**
The horizontal movement of the platform should reach up to a near by wall, at a distance of less

System use cases module

SPRO-UCS-10 **1.1.5 Brake**

SPRO-UCS-11 **1.1.6 Positioning of the Platform (from Platform)**

SPRO-UCS-12 **1.1.6.1 Adjust height of the Platform**

PHA Hazard module

SPRO-HAZ-1 **1 Stability is lost**
...

SPRO-HAZ-5 **2 Unexpected movement**

Risk evaluation module

SPRO-REV-1 **1 Stability is lost**
It is not sure if the amount of displacement needed for the horizontal movement tilts the machine. Hence the suggestive correction action about the stabilisers may not be necessary. However, a safety requirement: 'Stability shall not be lost' is recorded. Simulation about the stability shall be executed to check the need for stablilisers.

SPRO-REV-4 **2 Unexpected movement**
...

(System) Requirements module

SPRO-REQ-364 **4.15 Requirements from risk assessements**

SPRO-REQ-365 **4.15.1 Stability shall not be lost**
Movement of the platform to sideways direction shall not cause loosing of the stability with the maximum platform load with the worst case load location

SPRO-REQ-5 **5 Maintenance requirements**
Follow the safety related maintenance requirements of SFS-EN ISO 13849-1, SFS-EN ISO

Figure 4.32. *Doors implementation of the simplified Scissors elevator platform level artefacts with trace links.*

DOORS utilises trace links through red and yellow link navigation triangles. These triangles indicate links going outside (red) or inside (yellow) of an artefact and what artefacts are linked to each other. In this way, links can be traced backwards and forwards as they provide bi-directional linking to see impacts of a possible change in a specific artefact, and how the change corresponds to other connected artefacts such as the requirements.

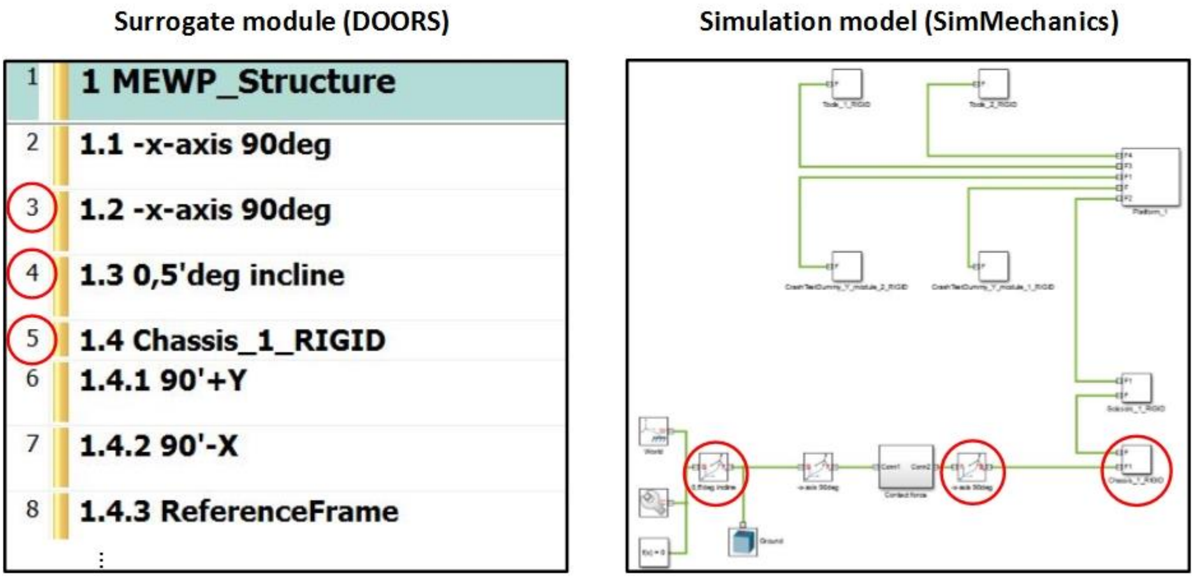
Files from external source are typically linked to DOORS with external links. External files are marked as URLs and they are originally one-way links to the resource they represent. URLs can be inserted in the resource to trace back into DOORS object, but to harmonize the tracing process external source linking is implemented in this thesis with the use of so called surrogate object method. More specifically, tracing process is done by linking external files from SVN to DOORS with a DXL script based surrogate object method which is described in details in Section 4.7.3

4.7.1 Implementing surrogate object method

The surrogate object method utilises a surrogate module to represent the artefacts of a foreign tool (such as a CAD tool). For each mapped element (element of a model or a full model), a surrogate object is created in the surrogate module, which is linked to the external file the surrogate object represents. The surrogate objects are also linked to the corresponding requirement objects (or other objects according to TIM). In this thesis surrogate object method is used to trace the logical architecture diagram, CAD model and simulation model artefacts between DOORS and SVN. DOORS does not inwardly know about the surrogate object method, but the method can be applied to DOORS environment as it is shown in Section 4.7.2 with MathWorks.

4.7.2 Surrogate object method implementation by MathWorks

The first surrogate module was created out of the SimMechanics model to allow tracing of the simulation model elements within the DOORS tool. MathWorks enables the use of surrogate object method by integrating DOORS and SimMechanics together with a plug-in. Figure 4.33 displays how the plug-in enables a representation of a DOORS surrogate module to be created from the SimMechanics simulation model. Even though, Figure 4.33 shows only the first eight objects of the surrogate module, note that the DOORS hierarchy can reflect the whole structure of the simulation model even representing the inner structures of the subsystems (see chassis in Figure 4.33). The level of mapped objects can be chosen whether it is only those with links to DOORS, or all of the simulation model objects. In Figure 4.33 all the simulation model objects are mapped to DOORS.



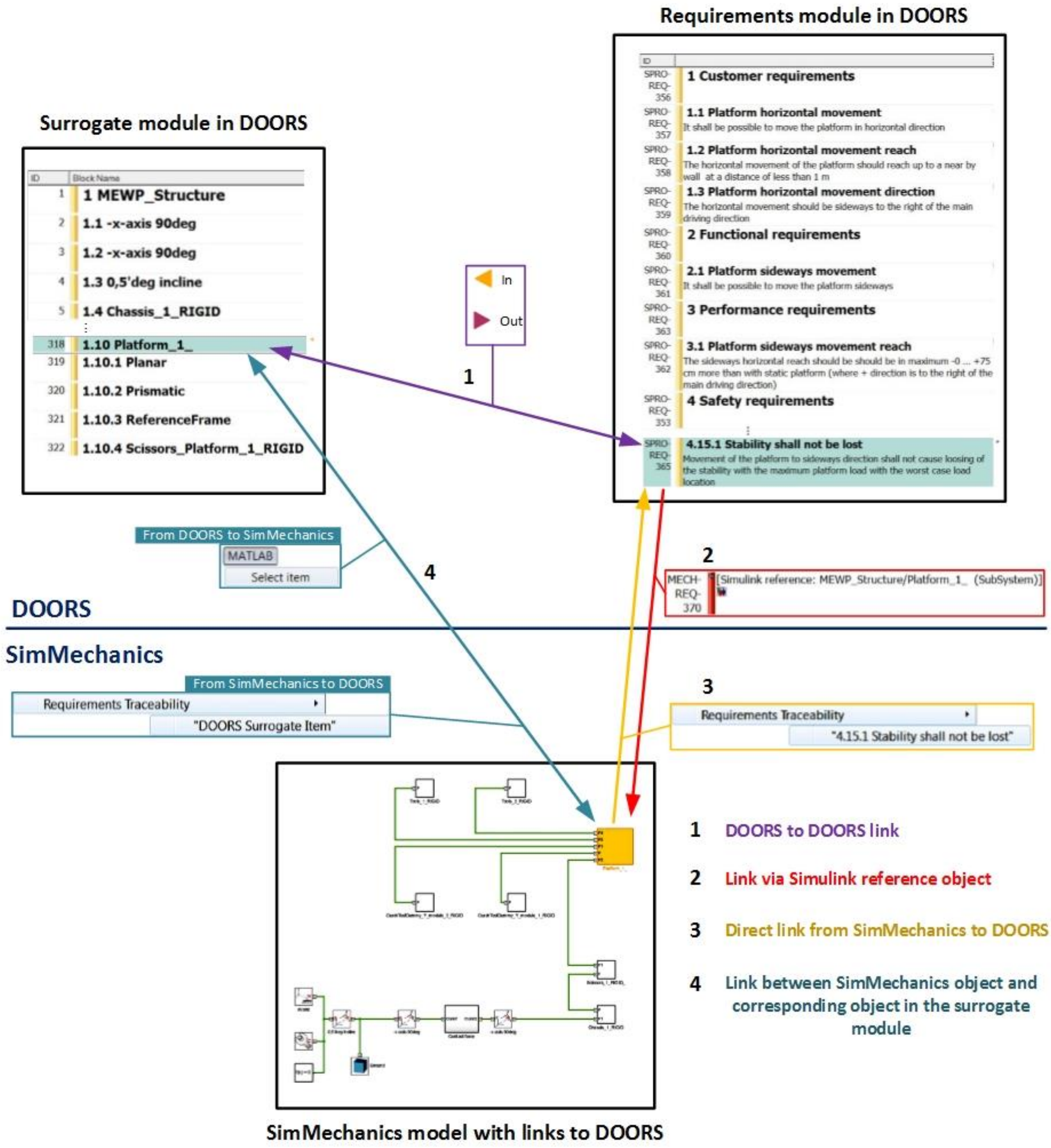
Surrogate module (DOORS)

1 1 MEWP_Structure
2 1.1 -x-axis 90deg
3 1.2 -x-axis 90deg
4 1.3 0,5'deg incline
5 1.4 Chassis_1_RIGID
6 1.4.1 90'+Y
7 1.4.2 90'-X
8 1.4.3 ReferenceFrame
⋮

Simulation model (SimMechanics)

Figure 4.33. *Doors implementation of the surrogate module (left) created from the SimMechanics model (right).*

The surrogate object method enables bi-directional linking without the need to modify objects in DOORS if a change occurs in the mapped object. All the required information is stored in the surrogate modules and link modules when the surrogate module is automatically generated by SimMechanics as a DOORS representation of the SimMechanics model. Links can be managed in the DOORS environment without the necessity to run foreign tools. An example of the linking methodology between the surrogate object, requirement and simulation model element is illustrated in Figure 4.34.



ID | Block Name
1 1 MEWP_Structure
2 1.1 -x-axis 90deg
3 1.2 -x-axis 90deg
4 1.3 0,5'deg incline
5 1.4 Chassis_1_RIGID
⋮
318 1.10 Platform_1_
319 1.10.1 Planar
320 1.10.2 Prismatic
321 1.10.3 ReferenceFrame
322 1.10.4 Scissors_Platform_1_RIGID

SPRO-REQ-356 1 Customer requirements
SPRO-REQ-357 1.1 Platform horizontal movement
It shall be possible to move the platform in horizontal direction
SPRO-REQ-358 1.2 Platform horizontal movement reach
The horizontal movement of the platform should reach up to a near by wall at a distance of less than 1 m
SPRO-REQ-359 1.3 Platform horizontal movement direction
The horizontal movement should be sideways to the right of the main driving direction
SPRO-REQ-360 2 Functional requirements
SPRO-REQ-361 2.1 Platform sideways movement
It shall be possible to move the platform sideways
SPRO-REQ-363 3 Performance requirements
SPRO-REQ-362 3.1 Platform sideways movement reach
The sideways horizontal reach should be should be in maximum -0 ... +75 cm more than with static platform (where + direction is to the right of the main driving direction)
SPRO-REQ-353 4 Safety requirements
⋮
SPRO-REQ-365 4.15.1 Stability shall not be lost
Movement of the platform to sideways direction shall not cause loosing of the stability with the maximum platform load with the worst case load location

MECH-REQ-370 [Simulink reference: MEWP_Structure/Platform_1_ (SubSystem)]

Figure 4.34. *An example of linking methodology with surrogate module (Figure modified from MathWorks, 2015).*

Surrogate module establishes a way to manage links within DOORS between objects in surrogate module (Platform_1_) and requirements (Stability shall not be lost) with DOORS to DOORS links (marked 1 in the Figure 4.34). These links are an alternative all-in DOORS representation to direct links from SimMechanics (Platform_1_) to artefacts such as requirements (Stability shall not be lost) in DOORS (marked 3 in the Figure 4.34). SimMechanics also offers a built-in selection linking with DOORS that enables navigation between artefacts in DOORS and simulation objects in SimMechanics with so-called reference objects (marked 2 in the Figure 4.34). However, this kind of practice modifies DOORS objects. Surrogate module establishes a way to surround the issue with a combination of DOORS to DOORS links between surrogate module and

DOORS artefacts, and links between SimMechanics objects and corresponding objects in the surrogate module (marked 4 in the Figure 4.34). Regarding this thesis, the before mentioned links 1 and 4 were used for studying trace analysis, and the realisation of impact analysis.

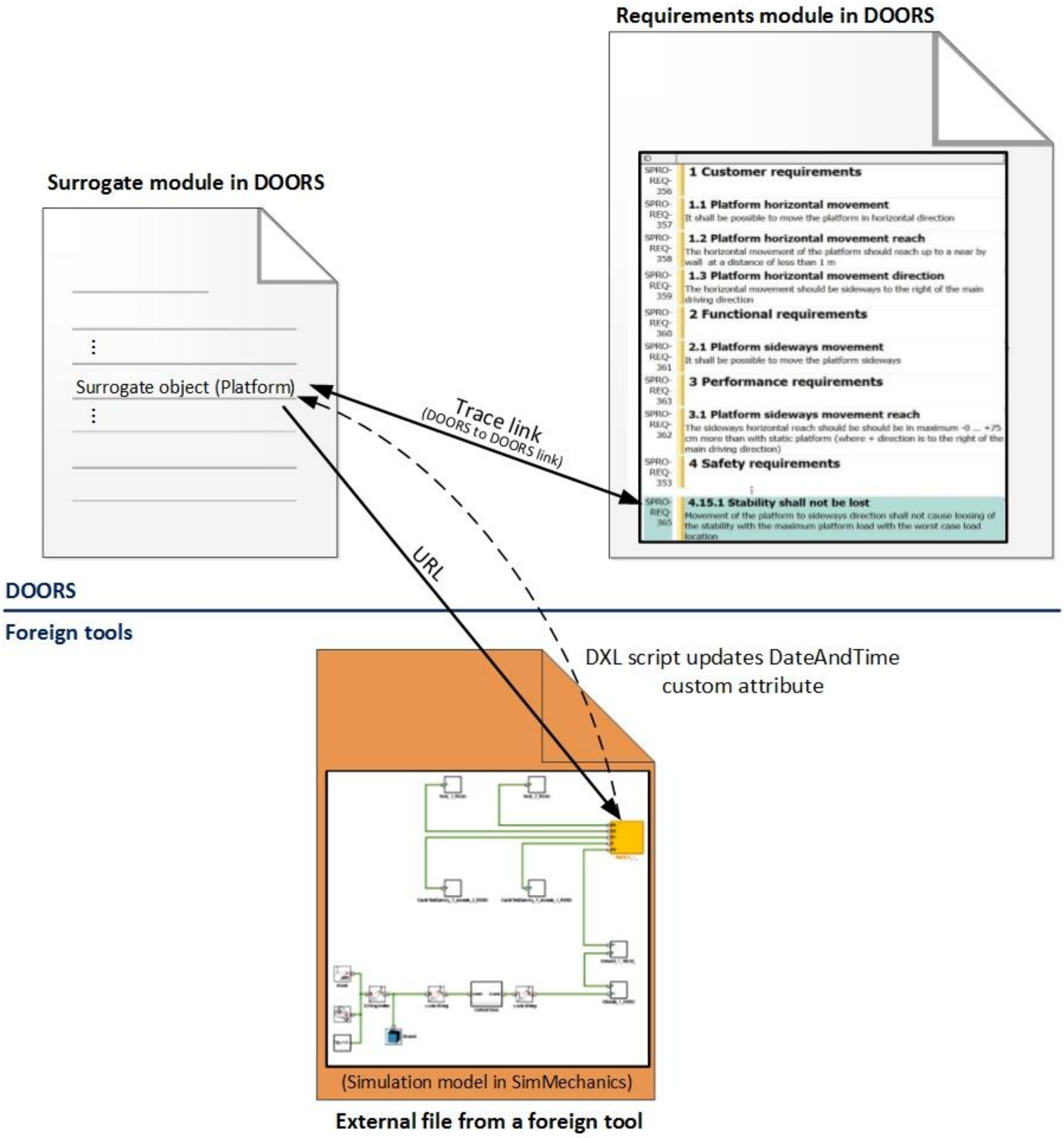
The impact analysis belongs, among trace analysis, to traceability functions, which facilitate the iterative systems development. DOORS/MATLAB integration provides impact analysis by reflecting model changes with a “Block deleted” attribute or with a suspect flag. The “Block deleted” attribute is a DXL programmed attribute by MathWorks, which is automatically included in the surrogate modules. If a block is deleted in the simulation model the Boolean value of the attribute changes to “True” in the surrogate module. Because the attribute provides information only about the existence of a specific block it leaves the granularity of impact study very low.

A suspect flag in turn, utilises flagging to indicate if an engineering artefact is edited at the one end of a trace link. When studied with DOORS and SimMechanics the suspect flag of a trace link, such as link 4 in Figure 4.34, was raised in the surrogate object end only when the name of the corresponding simulation model block was touched. However, suspect flags between trace links such as link 1 in Figure 4.34 within DOORS worked accordingly. The reason why MathWorks provided surrogate module implementation did not support impact analysis sufficiently is because surrogate objects do not reflect change albeit the corresponding simulation model element is changed. According to conversations with MathWorks, a deeper level of impact analysis is not supported in Simulink (hence also in SimMechanics) as the DOORS synchronization with the surrogate module is supposed to track only changes in block names and existence. Hence, the impact analysis provided by the surrogate object method of MathWorks was unable to cover changes in simulation model such as inertia and geometry changes of SimMechanics blocks.

The surrogate method is intended for tracing the simulation model elements to the physical structure elements and the requirements in order to validate the results. However, the insufficient impact analysis by MathWorks and the lack of granularity with external source linking of DOORS tool prevented the traceability functions to reach the wanted level. Additionally other external tools such as SolidWorks did not provide ways to integrate themselves with DOORS.

4.7.3 DXL script based surrogate object method

Implementation of SEAModel requires impact analysis based traceability, which can be achieved with different variations. Even though the surrogate method provided by MathWorks lacked proper way to achieve impact analysis, the method on itself proved to be a proficient way to display elements of external files as surrogate objects in one location (DOORS). Proceeding with surrogate method ideology required replacing the MathWorks based method with a tailored solution, so that changes in the external files would be covered by the impact analysis and the method could be applied to other external tools as well. Tailored solution included programming a DXL script for DOORS that updates a custom attribute, called *DateAndTime*. The new custom attribute is updated in the surrogate object if the comparison of the date and time stamps in the surrogate object and corresponding model file results in divergence. Figure 4.35 shows how the DXL script is implemented among the surrogate method. The presented trace linking between requirements and surrogate module is the same as in Figure 4.34 with link 1, but the link 4 between surrogate module and external file (SimMechanics model) is replaced with the DXL script application. Note here that link 4, which refers to a link between SimMechanics object and corresponding surrogate module object, is not meant to work with other external files than Simulink, and it is referred here only to ease the understanding of URL links position compared to the links presented by MathWorks.

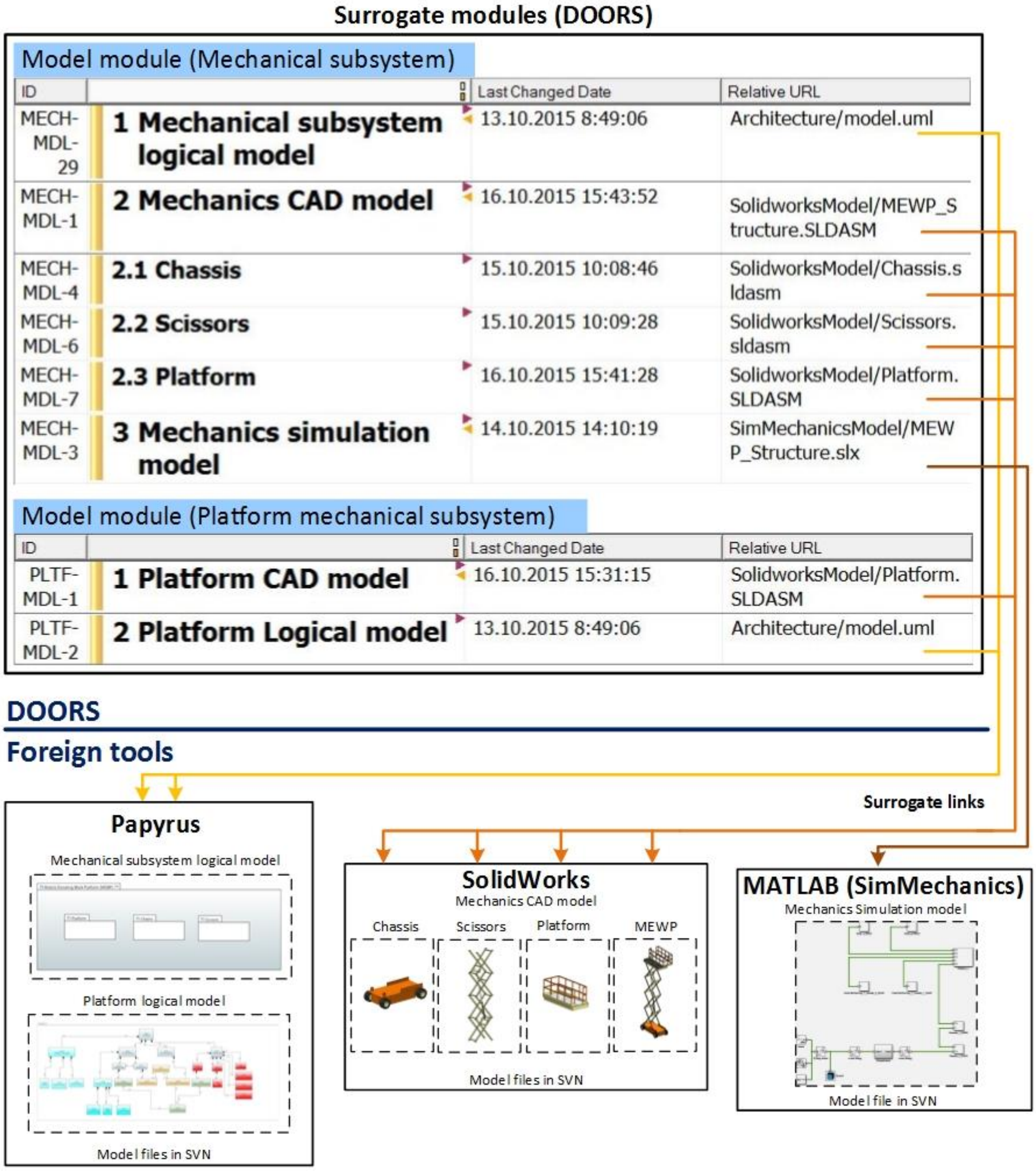


ID	
SPRO-REQ-356	**1 Customer requirements**
SPRO-REQ-357	**1.1 Platform horizontal movement** It shall be possible to move the platform in horizontal direction
SPRO-REQ-358	**1.2 Platform horizontal movement reach** The horizontal movement of the platform should reach up to a near by wall at a distance of less than 1 m
SPRO-REQ-359	**1.3 Platform horizontal movement direction** The horizontal movement should be sideways to the right of the main driving direction
SPRO-REQ-360	**2 Functional requirements**
SPRO-REQ-361	**2.1 Platform sideways movement** It shall be possible to move the platform sideways
SPRO-REQ-363	**3 Performance requirements**
SPRO-REQ-362	**3.1 Platform sideways movement reach** The sideways horizontal reach should be should be in maximum -0 ... +75 cm more than with static platform (where + direction is to the right of the main driving direction)
SPRO-REQ-353	**4 Safety requirements**
⋮	⋮
SPRO-REQ-365	**4.15.1 Stability shall not be lost** Movement of the platform to sideways direction shall not cause loosing of the stability with the maximum platform load with the worst case load location

Figure 4.35. *An example of a DXL script based surrogate method.*

The implemented DXL script causes touching the surrogate object to raise the suspect flag of a trace link if a requirement is linked to the surrogate object that represents, for instance, the model file of the changed simulation model. Furthermore the script offers a function to command a check for the model files such as Papyrus, CAD and Simulink files when a model element and hence the file is updated. This command is executed through a new custom feature to the user interface (UI) of the Doors called “User”, which makes possible the management of the surrogate module. With the new feature user can create either an empty surrogate module or create one from an existing formal module.

Creation of surrogate module includes additional attributes; a view and a trigger for integration with SVN. One of the attributes is “Repository Root” which is the URL for the SVN repository where the surrogate files such as objects are stored. The objects will include an attribute “Relative URL” where the relative URL of the file surrogated by the object in the SVN repository is contained. The URL is given in “Repository Root”. Other functions of the “User” tool relate to management of updating surrogate objects or surrogate object modules of the current project from the SVN repository information. Figure 4.36 depicts the structure of the surrogate modules constructed in the demonstration with relative URLs to the external files presented on the right.



Model module (Mechanical subsystem)

ID		Last Changed Date	Relative URL
MECH-MDL-29	1 Mechanical subsystem logical model	13.10.2015 8:49:06	Architecture/model.uml
MECH-MDL-1	2 Mechanics CAD model	16.10.2015 15:43:52	SolidworksModel/MEWP_Structure.SLDASM
MECH-MDL-4	2.1 Chassis	15.10.2015 10:08:46	SolidworksModel/Chassis.sldasm
MECH-MDL-6	2.2 Scissors	15.10.2015 10:09:28	SolidworksModel/Scissors.sldasm
MECH-MDL-7	2.3 Platform	16.10.2015 15:41:28	SolidworksModel/Platform.SLDASM
MECH-MDL-3	3 Mechanics simulation model	14.10.2015 14:10:19	SimMechanicsModel/MEWP_Structure.slx

Model module (Platform mechanical subsystem)

ID		Last Changed Date	Relative URL
PLTF-MDL-1	1 Platform CAD model	16.10.2015 15:31:15	SolidworksModel/Platform.SLDASM
PLTF-MDL-2	2 Platform Logical model	13.10.2015 8:49:06	Architecture/model.uml

Figure 4.36. *Surrogate module representations of Mechanical system level (top) and Platform Subsystem level (bottom) of the external files.*

Surrogate modules are created from the Model artefacts located at Mechanical System level and Platform Subsystem level of the demonstration architecture (see Appendix A). On the Mechanical System level, the surrogate module includes the Mechanical simulation model, Mechanics CAD model representations and Mechanics sub-system logical model. Unlike the CAD model, the simulation model is located in one file. Thus, a versatile impact analysis is difficult to arrange, because the granularity of traceability in a single file is poor. This was an acceptable drawback in the demonstration, as a change in any simulation element leads to rising of the suspect flag of the trace links linked to the surrogate object of the entire model. Location of the exact change needs to be determinate manually by following possible revision documentary or comments. At the moment these aspects are not included in the surrogate module script, but it is possible for an individual to follow revision commentary from the SVN.

On the platform subsystem level, the surrogate module includes representations of Platform CAD model and Platform Logical model. The logical model and simulation model both consist of single files, but the CAD model is a granular entity consisting of a variety of assembly and part files. The most relevant assembly files, Chassis, Platform and Scissors are represented at the Mechanical system level, but the Platform assembly is also included in the Platform Subsystem surrogate module on the Platform mechanical subsystem level. This practice enables platform to be an individual entity at the demonstration architecture.

In an actual representation all the relevant subassemblies (chassis, scissors, and platform) would have their own specific mechanical subsystem level, and they would not be included beneath Mechanics CAD model. However, in this thesis they are included in order to clarify the structure of the demonstration, and because only the platform had its own model module on its own specific subsystem level. From the research point of view it was also interesting to see how the traceability would work with the before mentioned representation as the practice increases the granularity of traceability. On a side note, because a model is represented in two different surrogate modules as the platform is in Figure 4.36, it is possible that the attribute Last Changed Date could differ between the surrogate modules, since they are updated separately. For instance, in Figure 4.36 Model module on Platform mechanical subsystem level is not updated, hence the different date.

4.8 Concept of arranging traceability in the case study

The previous sections have introduced the tools and methods to implement SEAModel in a mechanical engineering demonstration case. Requirements engineering have been extended to cover simulation artefacts via the surrogate object method, and the tracing linkage between requirements management (DOORS), surrogate modules (DOORS), modelling and simulation tools (Papyrus, SolidWorks, MATLAB/SimMechanics) have been enabled by the version control platform (SVN). Additionally the arrangement of traceability required sufficient harmonisation of the tools and impact analysis to assess the level of change in the system. Therefore, a custom DXL script was written to facilitate analysis of the impact of changes from foreign tools to requirements via a surrogate module.

The demonstration of requirements traceability concentrates on impact analysis by checking the suspect flags between artefacts after a change has occurred in 1) external files and 2) requirements. The area of interest is limited to the simplified traceability model depicted in Figure 4.31, restricting the study to pertain around the safety requirement for the stability of the MEWP. Fulfilment of the requirements and validation of simulation results are studied in a systems engineering loop. The loop goes from a system requirement to design artefacts (in this case SysML, CAD and simulation models), from design artefacts to the simulation results (recorded to a V&V execution report), from the V&V execution report to a V&V success report and from there back to the original system requirement. This process is implemented in Figure 4.37, in which the trace and impact analysis is demonstrated in the context of previously mentioned simplified traceability demonstration model.

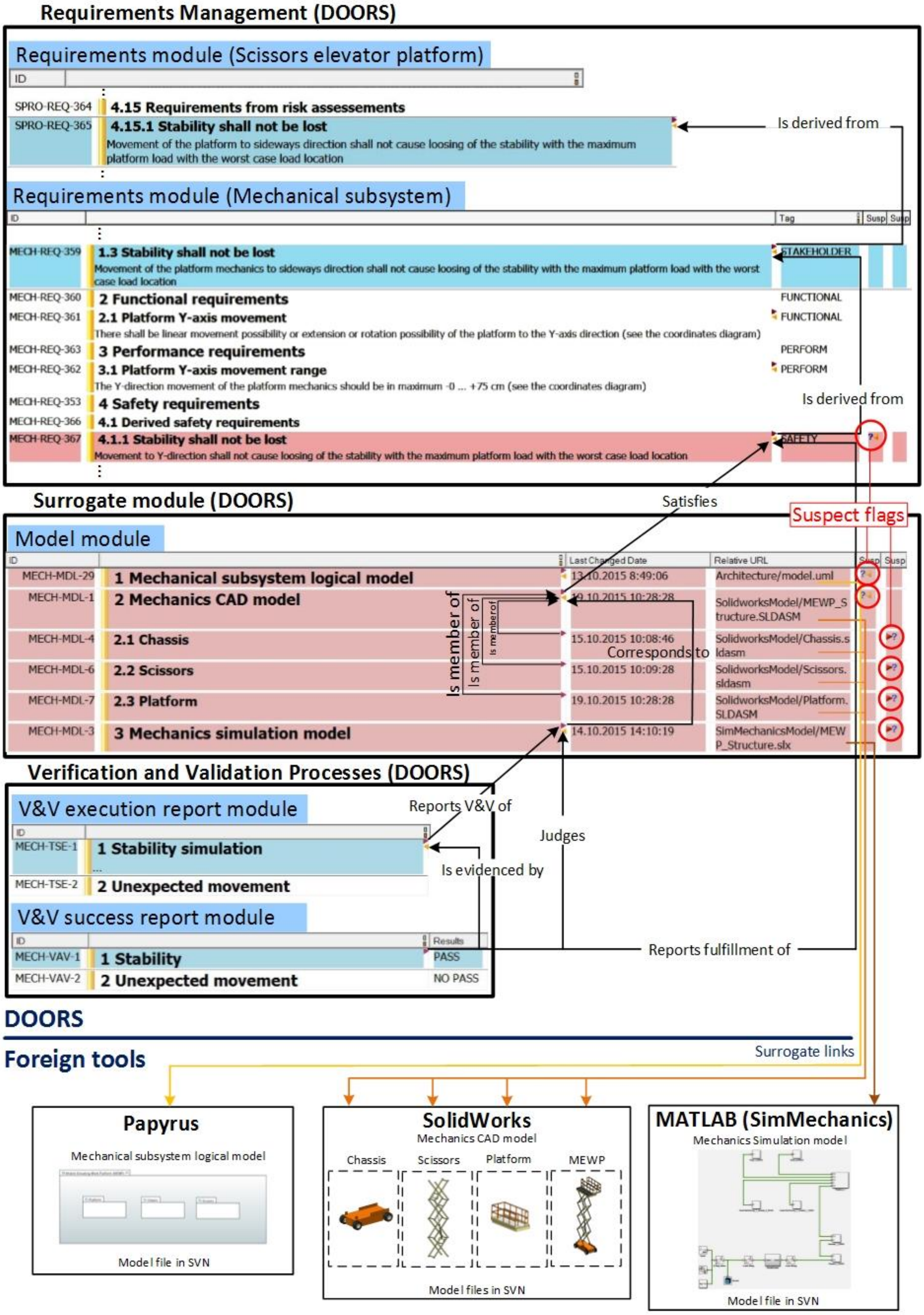


Figure 4.37. *Implementation of trace and impact analysis in the simplified traceability demonstration (see Figure 4.32).*

The simplified traceability model does not include artefacts from the Platform Subsystem level including surrogate object representations of the Platform CAD model and Platform Logical model. Therefore the surrogate module in Figure 4.37 is the one on Mechanical subsystem level. By focusing to the core loop, the effect of a change and the functionality of SEAModel are more easily demonstrated. Moreover, if functionality of SEAModel can be approved with the simplified model, functionality of the entire traceability model can be claimed.

4.8.1 Trace and impact analysis in the case study

The process of trace and impact analysis study is carried out by modifying Mechanics CAD model (MEWP main assembly). In Figure 4.37 suspect flags are highlighted with red circles, and the related DOORS objects, which are impacted by the changed CAD model (updated Mechanics CAD model), are noted with a red background. A blue background depicts the start or end of a trace link to or from an artefact that is part of the systems engineering core loop, but not affected by the immediate change. Note here that, even though Mechanical subsystem logical model is impacted by the change, and it is part of the surrogate module at Mechanical subsystem level, it is beyond the systems engineering core loop having no direct contact with the confirmation of the stability of the MEWP.

The modification of Mechanics Cad model is implemented by changing the dimensions of Platform in SolidWorks. The corresponding impact is studied with suspect flags among the traced artefacts. The suspect flags indicate that it has to be checked whether Mechanics simulation model still correspond to the changed platform object (CAD model), and furthermore, does the platform object (CAD model) still satisfy the system requirement of “Stability shall not be lost”. Platform object is located underneath Mechanics CAD model object which includes the whole assembly of MEWP. Therefore Mechanics CAD model needs to be updated in SolidWorks whenever one of its subassemblies is changed. Otherwise change can be only detected on Mechanics CAD model level between the main -and subassembly. This could be avoided by implementing trace links from each subassembly CAD object to the artefacts Mechanics CAD model (MEWP main assembly) is connected to, but it would increase the complexity of the traceability demonstration model. Increased complexity can be a serious problem in a project with a large number of artefacts.

Updated Mechanics CAD model causes the suspect flags of the trace links to rise at the requirements module of Mechanical subsystem level. This happens when the timestamps on the surrogate CAD objects are updated. The suspect flag notification is also noted at Mechanics simulation model, Mechanical subsystem logical model and in Chassis, Scissors and of course Platform. However the suspect notifications caused by the updates of the subassemblies are not informative in Figure 4.37 because the change is traced onwards only to Mechanics CAD model object (MEWP main assembly). If a change occurs in any of the subassemblies it makes all subassemblies suspect through their trace link to the Mechanics CAD model, even though the change would not actually influence other CAD assemblies. Yet, these subassemblies are justified as it was mentioned in the end of Section 4.7.3 and hence the incorrect suspect flagging in this regard is considered as a marginal flaw. Furthermore, there is no actual problem with the suspect notification that would question the functionality of SEAModel, because the change can be traced from a specific subassembly to relevant requirements through the surrogate object module on the specific mechanics subsystem level. In this case study the arrangement was only done for the platform model (see Figure 4.36), but for the sake of proving the functionality of SEAModel the approach in Figure 4.37 was chosen.

The impact analysis can be followed through the black arrowed trace links from the requirements to the V&V success report. The trace links and their types are the same as in the traceability demonstration model (see Appendix A) and in the simplified traceability demonstration model (see Figure 4.31), accordingly. The success report indicates the approval of the Mechanics CAD model with a “Results” attribute. This attribute has value “pass” or “no pass” depending on the simulation results. The attribute value is edited by the user and is not automatic. In Figure 4.37, the value is “pass”, since the simulation model has met the stability criteria. However, when Mechanics CAD model is changed the suspect rises in Mechanics simulation model. If the simulation model needs to be modified the suspicions are propagated, i.e. the V&V success report (i.e. the Stability object and the “results” attribute) need to be re-evaluated based upon a new simulation run with the new simulation model. The re-evaluation needs to be done also when a change occurs in the system requirement “Stability shall not be lost”. In the case of changing the system requirement suspects appear at the corresponding stakeholder requirement, Mechanics CAD model and naturally at V&V success report object.

5. ANALYSIS OF THE RESULTS

Chapter 5 concentrates on analysing the results of the case study presented in Chapter 4. The traceability case demonstration is assessed and the efficiency of the used traceability information model with the chosen tools and methods is discussed. The critical part of achieving sufficient traceability was the accomplishment of impact analysis. Hence the level of achieved impact analysis is also reviewed. The results address research questions presented in Section 1.4. Research questions are answered in non-chronological order to fit better with the analysis. Finally, recommendations are given concerning the implemented concept for requirements traceability.

5.1 Applicability of SEAModel in simulation driven mechanical engineering

SEAModel and TIM specified by SIMPRO project are evaluated in this thesis according to their capability to manage and offer traceability. An optimal solution for traceability would include fine-grained granularity, bi-directionality and commitment of engineers. Fine-grained granularity makes the process more transparent as the system is described with enough of different levels of traceable units. The traceable artefacts need to be bi-directionally connected to enable proficient impact analysis and to track the artefacts to and from the set requirements. In the end, all traceability implementations require the commitment of engineers from different disciplines to create and maintain these linkages.

SEAModel was created to complement the systems engineering standard ISO/IEC/IEEE15288 to follow the systems life cycle processes while concentrating on the technical processes. Figure 5.1 illustrates the systems engineering core artefacts of SEAModel that are produced by the technical processes.

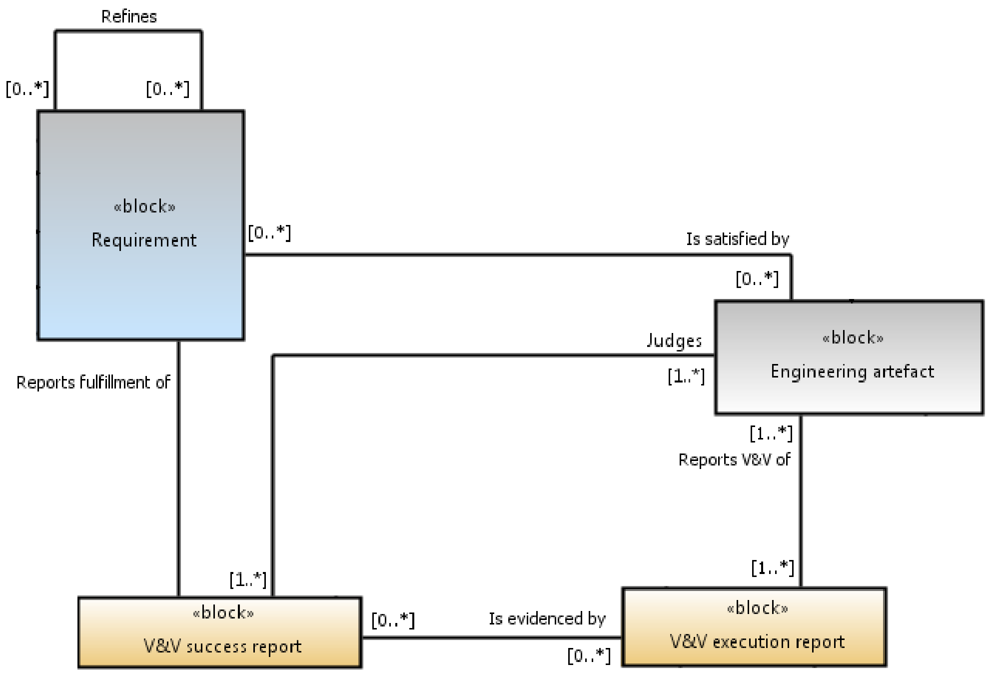


Figure 5.1. *The core loop of SEAModel (modified from Figure 3.2).*

The artefacts in the core loop; Requirement, Engineering artefact (any), V&V execution report and V&V success report provide a frame for indicating conformity to the Machinery Directive 2006/42/EY Annex I requirements. The directive concerns the safety issues relating to the product requirements of machines in general. These requirements are noted in the design of machines such as MEWP, accordingly.

However, the Machinery Directive does not recognise term assurance case which generally supports claims in areas such as safety and reliability. Instead Annex VII of the machinery directive mentions a technical file to be used, if required, for providing compliance of the requirements of the machine to the authorities (Directive on machinery 2006/42/EC, 2006). Although technical file for machinery is not the same as assurance case they both reflect information from validation record. Recording demonstrates how safety requirements are validated by testing and analysis during the validation process. Because of this conjunctive factor between assurance case and technical file, and the lack of assurance case in the directive, the contents of SEAModel are compared against the contents of the technical file. Contents of the technical file that are chosen for the comparison, in turn, comprise from the contents of a construction file, which namely provides means to demonstrate the conformity of machinery (Guide to application of the Machinery Directive 2006/42/EC, 2010). The comparison is illustrated in Table 5.1.

Table 5.1. *Comparing contents between SEAModel and Technical file (SFS-EN ISO 13849-2, 2012).*

Chosen contents of Technical File	Corresponding content in SEA-Model
General description of the machinery	System
Overall drawing of the machinery and drawings of the control circuits, as well as the pertinent descriptions and explanations necessary for understanding the operation of the machinery	Behaviour package; Structure package; Foreign model
Full detailed drawings, calculation notes, test results, certificates etc. required to check the conformity of the machinery with essential health and safety requirements	All V&V artefacts and requirements; System model; Foreign model
Documentation of risk assessment	Risk assessment package
List of the essential health and safety requirements which apply to the machinery	Requirement
Description of the protective measures implemented to eliminate identified hazards or to reduce risks and, when appropriate, the indication of the residual risks associated with the machinery	Risk evaluation
Standards and other technical specifications used, indicating the essential health and safety requirements covered by these standards	Black box artefacts
Any technical report giving the results of the tests carried out either by the manufacturer or by a body chosen by the manufacturer or his authorised representative	All V&V artefacts
Copy of the instructions for the machinery	Document
Where appropriate, the declaration of incorporation for included partly completed machinery and the relevant assembly instructions for such machinery	Document
Where appropriate, copies of the EC (European Commission) declaration of conformity of machinery or other products incorporated into the machinery	Document
Copy of the EC declaration of conformity	Document

Content comparison between technical file and SEAModel requires SEAModel to be studied more thoroughly than the scope of this thesis would allow. In addition to artefacts claimed from TIM the technical file also reflects characteristics of the behaviour package (see Figure 3.1 and Alanen, et al., 2015, pp 29), Structure package (Alanen, et al., 2015, pp 21), Data repository model (Alanen, et al., 2015, pp 15), and Risk assessment artefacts model (Alanen, et al., 2015, pp.46). Files such as copies of instructions, declarations and standards that do not have immediate correspondence in SEAModel, can be reflected as black box artefacts. Data repository model includes black box artefacts that are either without modelled internal structure or their structure is unknown by the data repository model. Black box artefacts in turn include foreign model and document as its main types. With these models and artefacts the principle of an assurance case is claimed by SEAModel.

To implement SEAModel in practice, the mechanical engineering tools require interfaces that enable and ease the requirements traceability. For instance, interfaces between SolidWorks and Simulink, and Simulink and DOORS had plug-ins to provide data over the software interfaces. In an optimal situation, this data and the derived design models would consist of multiple files providing fine-grained granularity. This way the depth of the occurring change could be pointed out more efficiently during the impact analysis. Additionally, the data structure of the mechanical engineering tools would consists of the same format that could also be easily handled during the systems engineering process. The best option considered for being incorporated into SEAModel is the XML format, since the structure of XML can be regarded as if a database. This works for SEAModel which is a data model that can be implemented in different structured ways such as XML-files. Yet, in the demonstration case XML format was not exploited, since the used software were chosen on the grounds of how known they were to the project researchers and how commonly used they were.

Nevertheless, the demonstration case indicated that TIM and SEAModel can offer sufficient traceability and their applicability in simulation driven mechanical design can be achieved with a set of commonly used modelling tools. However, traceability of artefacts with the optimal granularity and visibility of data is challenging to arrange from a set of heterogeneous modelling tools and cannot be guaranteed. For instance, the demonstrated ITP implementation does not offer optimal granularity of traceability because model elements (stored in SVN) from Papyrus, SolidWorks and MATLAB can only be traced in DOORS at the file level. Tracing of a changed model element within a model file to another model file was not achieved, which meant that the whole model needed to be checked in order to determine the impact of the change. This can prove to be arduous if the file is, for instance, a simulation model that consists of various different sub-systems and blocks. An exception regarding traceability was the connection between DOORS and Simulink, which offered a function to visually trace an object from DOORS surrogate module to the simulation model in Simulink and back (see Figure 4.34). Nevertheless, this connection did not offer sufficient impact analysis as it was mentioned in Section 4.7.2 regarding the surrogate object method by MathWorks.

5.2 Feasibility of SEAModel implementation

At the moment, SEAModel contradicts with the system used in the demonstration case (see Appendix A) on a system element level. Referred in Section 4.8.1, the decomposition of artefacts into sub-level artefacts on a specific system level is not well supported in SEAModel. The lack of support supervenes from the fact that SEAModel does not introduce an explicit way to link a system element to another system element (although, implicitly it can be done). Regarding the demonstration case, this means that the machine designer can only see the elements chassis, scissors and platform, of which the system-of-interest MEWP consists of and not what the individual elements of the MEWP consist of.

Traceability demonstration model (see Appendix A) depicts this problematic aspect as artefacts are located on different system levels: Scissors elevator platform, Mechanical subsystem and Platform mechanical subsystem of which the last one includes system elements of the platform artefact that again is a system element at the Mechanical subsystem level for the MEWP. Thereby, at the Scissors elevator platform system level, the mechanical designer cannot influence through the system design to the internal decomposition of the system elements acquired from a sub-contractor Yet, the demonstration case proved that representing the model files as surrogate objects on the traceability platform assists SEAModel to support tracing of engineering artefacts in requirements aware simulation engineering even over organisational borders. The information flows bi-directionally between the system levels from machine developer to the sub-contractors and back. The information between the system levels is just not as visible as it was pursued to be.

The demonstration case showcased some commonly used tools for mechanical engineering and requirements management. The tools were integrated together by placing the tools data into a SVN repository from which the timestamps of the files could be reached by all participants and tools with some additional scripting. However, the data does not automatically travel between the tools when change occurs in a linked model; the tools could not directly utilise a model created with a different tool. Simulink, however, could create a simulation model out of the physical model created with SolidWorks, though it may require modifying depending on the testing requirements. These requirements are set by the simulation type and the wanted outcome. In the demonstration case simulation requirements led to enhancing the simulation model with contact force systems and dynamic motion mechanisms to conduct the stability of the system.

In other words, true integration of the tools data was not achieved. Since changes are not reflected automatically from one model to another, engineers need to manually input the data leaving a possibility for a human mistake. For instance, when a mechanical engineer inputs changed data from the logical architecture model to the CAD model there may emerge parameter errors or misprints during the process. The problem could propagate from the very beginning when the original physical CAD model is manually created according to the logical architecture model. Even though it may be challenging to automatically reflect structural changes from a logical model to its corresponding CAD model, the demonstration case managed this by prompting the engineer with the previously mentioned suspect flags to check the child models for possible needs for changes due to the changes in the father model.

Impact analysis was made possible from requirements to design models, to V&V and through the system sub-system hierarchy by the DXL script created for the demonstration. The script facilitated the file level traceability and impact analysis when other current concepts did not offer necessary tracing of modelling artefacts. Impact analysis was managed in the DOORS environment with suspect flags that react to change in the corresponding linked artefacts. This functions basically automatically after the trace links are set up requiring only refreshing of the surrogate modules. However, as it was with the traceability in the demonstration case, impact analysis can be only done at the file level.

The complexity of a SEAModel based workflow may be difficult to grasp in the beginning without a personnel with systems engineering background. Nevertheless, SEAModel facilitates traceability models and an organised data repository that the mechanical designers could use for the artefacts they need as the input for their work and for artefacts their work produces. In model-based systems engineering, engineering tools that are aware of the artefacts model and provide good user experience are vital in decreasing the feeling of complexity after the engineers are comfortable with the tools. With the commonly accepted engineering and requirements management tools used in the demonstration case, SEAModel satisfactorily supports at least the basic needs of simulation artefacts tracing.

Artefacts traceability includes a time investment that is required upfront to implement SEAModel into a database oriented software platform, to set up the trace links and configure the traceability information model to work with chosen engineering tools. The systematic approach minimises chaos but it may increase costs as shown in Figure 5.2.

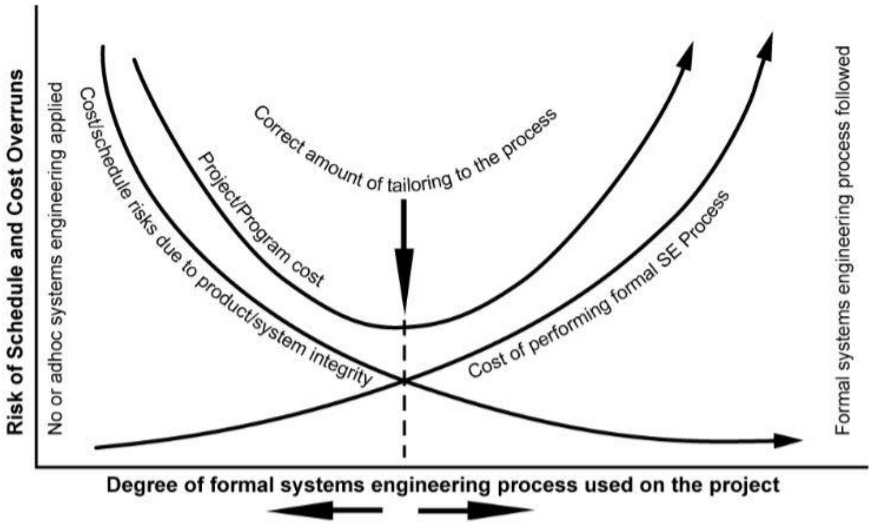


Figure 5.2. *Balance between risk and process (INCOSE, 2006).*

The increase in costs is due to rising level of formalism that cannot be satisfied accordingly with the ideal amount of tailoring to the process. By increasing the degree of formal systems engineering and making the process more complex, Systems Engineering process costs will increase, but the necessary requirements such as safety standards can be satisfied.

However, it can be claimed that the incremental resources (time, financial) consumed while using SEAModel are outweighed by the resources saved further along in the development process. Saved development costs are due to benefits such as increased quality of safety engineering that SEAModel supports. Quality of the safety engineering is increased because SEAModel manages the risk assessment systematically, and traceability of the safety requirements is supported according to the demonstration results.

Yet, the use of SEAModel should be optimised according to the size of the project. If the project is small in size, the time and financial investments of a full SEAModel implementation may be too high to be profitable considering the effects of psychological inertia among the engineering personnel. Though psychological inertia appears regardless of the size of the project, larger projects may have more resources to overcome the factor.

5.3 Discussions of the chosen methods and the relevance of the thesis

Section 5.3 discusses the chosen methods presented in the thesis as well as the significance of the research. The relevance of the subject is discussed in accordance with the state-of-the-art in mechanical engineering industry.

This thesis was conducted in the context of mechanical engineering industry with a case study focusing on the traceability of the engineering artefacts in simulation driven systems development. Specifically, the case study examined challenges surrounding the requirements traceability in a simulation driven validation and verification process while implementing a systematic model, SEAModel for the engineering artefacts. The significance of a systematic model was one of the main success factors noted in development of complex systems. Other factors included systematic processes and life cycle model (ISO/IEC/IEEE 15288), a well-defined and effective organisation model, well-chosen systems engineering tools and effective use of project management and finally an integration and traceability platform to bring everything together. A specific notation was put on the analysis of engineering tools that would support version control and impact analysis as the relevant engineering artefacts were traced to each other.

Concepts of Systems Engineering, such as requirements traceability, are commonly allocated to the software industry rather than the machine and mechanical engineering industry. Yet, the novelty of this thesis showcases how Systems Engineering is implemented in a mechanical engineering case to produce an efficient and streamlined process from stakeholder requirements to design and verification of a product via simulation. Current practices often take lightly the simulation process and easily leave simulation as a separated practice without consultation of the original stakeholder requirements, and the consequent system requirements. Furthermore this kind of practise leaves simulation engineers without knowing the reasons or justifications behind a specific simulation task. The case illustrated in the thesis provides a possible way to bring the simulation domain closer to other practises by integrating artefacts produced by the simulation engineers and their software tools with the other artefacts of the development processes.

The case study brought up the relevance of developing a versatile TIM for establishing sufficient traceability of engineering artefacts. SEAModel is required to explain how the physical structure of a system is defined by its system elements and their interfaces in order to provide a proper TIM. Therefore, SEAModel adapts the system structure concept model of AP233 (ISO/DIS 10303-233, 2009). The systems engineering workflow of SEAModel, however follows the standard ISO/IEC/IEEE 15288, which defines the systems life cycle processes. SEAModel, which specifically follows the technical processes of the systems life cycle processes, defines the core engineering artefact types and their relations that are presented in TIM, and eventually in the traceability demonstration model. Even though the contents of the demonstration model varied throughout the case study the model aided to provide together with data models (SEAModel and TIM) a valid structure for the artefacts traceability.

Application of the models was demonstrated in the case study with a set of engineering tools typical for the industry. The main reasons for the tool selection were that the tools were commonly used and they were easily attainable for the case study. Because of the limited resources apprehended to the thesis, the licensing policy of the tools was also a notable factor. Additional reason why the preliminary chosen tools were used throughout the case was that SEAModel is implementable with different kinds of database oriented platforms allowing engineering tools to be selected from a heterogeneous kit. Flexibility among tool selection is a notable factor since it is unlikely that all collaboration partners within a project utilise the same tools. This also means that optimal solution for SEAModel implementation is difficult to find. However, the tools have to provide a structured data repository like SVN in the case study in order to capture dependencies between the engineering artefacts and trace the impacts of a change among the artefacts.

Integration of engineering tools within the ITP was approached by adopting the surrogate object method. Surrogate object method is also a mean to implement traceability with impact analysis which can be done with different variations from sophisticated graphical diagrams to tailored solutions. The limitation of tracing modelling artefacts in current practices was noted during the case study and consequently a tailored DXL script was created for DOORS to diversify the surrogate object method. The tailored method facilitated a file level traceability between DOORS and SVN which was not only satisfactory for the SIMPRO project, but also a significant stride from neglected requirements traceability towards proper management of shared engineering artefacts. Furthermore the presented data models, traceability process model and surrogate object method together display the potential for the mechanical industry to evolve towards utilisation of model-based systems engineering over conventional methods that use document based repository of systems engineering data.

Software updates can cause malfunctions in the intended workflow and change the outcome of the study. This issue needs to be highly noted in requirements traceability, because the process includes a variety of software interfacing with each other. For instance, the tailored solution of DXL script did not function after DOORS software was updated. Update changed the file structure of DOORS which resulted in disabling the script and the possibility to produce impact analysis. Resolving the problem highlighted the importance to check all the characteristics of the requirements traceability process when software updates have occurred. Although this may seem time consuming it is a necessity as the overall effect of software updates is difficult to predict.

The research process included an inherent limitation of relying on a single case study. Even though the study was carefully planned and it derived from previous research at the VTT it is not possible to make generalized statements of the research. The goal of the case was to verify the functionality of SEAModel with chosen engineering tools, but to do this profoundly would require larger variation of cases. Furthermore the exact outlook and far-reaching impact of the results was narrowed down to speculation, because there was not a specific point of comparison from the industry. This was due to limiting the research methods to constructive and quantitative methods revolving around the case study while leaving the empirical interview study out of the scope. The research methods were exactly defined, because of the limited time resources. Furthermore additional methods would not have fit within the extent of the thesis.

In retrospect, an ethnographic study with the industry could have clarified how the cooperation among engineers supports Systems Engineering. Also, this kind of a study could have shown how the results of the thesis actually compare against the state-of-the-art methods and tools in mechanical industry. For instance by following the procedures of a client organization during a regular workday could have provided better insight on how engineers work with the given data, data they produce and how aware they are of the stakeholder requirements. However, as an initial demonstration of SEAModel implementation the thesis succeeded in providing a practical scenario for requirements traceability in simulation driven design.

5.4 Recommendations for the future work

Section 5.4 discusses recommendations that are based on the conclusions of the case study and its results. The recommendations for further research pertain to the present situation of the SIMPRO project, and also cover the implementation of SEAModel.

The case study brought forward the difficulty to decide which artefacts should be traced to a specific artefact. Even with the data models SEAModel and TIM, tracing was a complex process because the demonstration had artefacts spread on three system levels, Scissors elevator platform level, Mechanical subsystem level, and Platform subsystem level. To ease the complexity, the optimal traceability tool should be aware of the used artefacts data models. By knowing how the artefacts are distributed among possible subsystem levels, traceability tool could support the user in creation of the trace links and minimise the level of confusion regarding the placement of the artefacts.

An example of the traceability issue with system levels is the architectural structure of MEWP that was decomposed to three components; chassis, scissors, and platform. The problem emerged due to the fundamentals of SolidWorks that allowed subsystem such as the platform to be updated without updating the upper level element MEWP. Because the version number of the upper level element is not touched, the impact analysis will fail. One possible solution is the one exhibited in the thesis i.e. to trace all the lower level elements to the upper level element. This way the information of a change in any of the elements is indicated to others as a risen suspect flag. The method, however, forces the user to check all the elements (MEWP, chassis, scissors and platform) even though not all of them are necessarily affected by the change (see Figure 4.37 where change in platform is seen in every element of Mechanics CAD model as it is updated).

Another solution could be to link the lower level elements to the same artefacts as the upper level element thereby allowing propagation to bypass straight to the relevant artefacts. However, the creation of trace links at this magnitude would require a lot of work and it would increase the complexity of the requirements management and the traceability demonstration model. Furthermore, the increased number of connections would require the user to be even more alert and careful while arranging the links.

Other possibility could be to design a new attribute e.g. “Touched” that could be implemented to the requirements management tool. The one who modifies the lower level element would enter a notification about the change into the attribute. Impact of the change in the new attribute would inform the element to have been changed and prompt the user to save the upper level element. Saving the upper level element would allow indication of the change to propagate forward. However, the whole problem with the decomposed components can be avoided by situating all the relevant subassemblies on their own mechanical subsystem levels as individual model modules. This practice would allow tracing to happen directly from the specific model module to the relevant requirements etc.

The impact analysis on itself could be done in finer granularity with a different script than the one used in the case study. A more elegant approach would be to save the simulation e.g. SimMechanics model files into XML format based text files and compare the current model revision to the previous one to find the exact elements that have been changed. The same would apply for the SysML models.

6. SUMMARY AND CONCLUSIONS

System development, being either software or mechanical based, is built upon requirements regardless of how well the discipline of requirements engineering is recognised. Requirements engineering aims to help organisations reach their goals, but still it is perceived as an uncomprehending task by many. The purpose of this thesis was to study, could the requirements engineering, which is a necessary part of the overall systems engineering process, be extended to cover simulation artefacts with the use of Systems Engineering Artefacts Model (SEAModel). The context was a fictitious simulation driven mechanical engineering case study. Furthermore, the goal in the background was to demonstrate how to apply Systems Engineering as a multidisciplinary approach for developing solutions to complex engineering problems. More strictly, the core of this thesis concentrated on building a traceability chain between a set of heterogeneous engineering tools and demonstrating a method for sufficient impact analysis.

A noticeable drawback of traceability is the high effort of arranging and maintaining traceability relations. To change the attitudes of the engineers and to make requirements aware simulation engineering more attractive for its practitioners, the traceability tools need to be easy and effortless to use, and the benefits of traceability need to be demonstrated to compensate for its costs. SEAModel and the derived Traceability Information Model (TIM) demonstrated in this thesis are created to facilitate possible answers for these demands.

The development of requirements traceability in simulation driven design was approached with an insight of how the data models were created and how the workflow of artefacts traceability was thought through. The theory basis leaned on the Total Theory of Technical Systems which guided towards the theories of Design Science. The workflow of the requirements and systems engineering activities was concluded from the systems life cycle processes standard ISO/IEC/IEEE 15288 and its daughter standards. One of the results of this thesis was to create a traceability demonstration model according to the data models. Many potential design theories such as Property-Driven development could have been used to provide the demonstration model, but the technical processes of ISO/IEC/IEEE 15288 provided the most comfortable solution to be implemented in the case study.

The case study involved studying traceability of requirements of a mobile elevating working platform (MEWP) of scissors type, by viewing it through the traceability demonstration model. The preparatory work consisted of providing logical architecture model, CAD model and finally a simulation model, which were built upon the stakeholder and system requirements captured into the requirements management tool. Requirements were based on a fictitious customer request for a new feature for the machine. The created model artefacts were mapped to the traceability demonstration model, which defined the traceability chain to trace artefacts in the requirements management tool. A special concern was put on the safety requirements that promoted the need to validate and verify the CAD model through simulation to insure the stability of the machine when executing the new requested feature.

To trace foreign models within the requirements management tool, a surrogate object method was introduced. The method provided means to present foreign model elements as surrogate objects in the requirements management tool. Originally, the surrogate object method was utilised to trace model elements from the simulation tool to requirements, but later it was expanded to include other chosen engineering tools as well.

The case study confirmed that artefacts traceability with impact analysis from stakeholder requirements to verification and validation reporting could be arranged with a selected combination of state-of-the-art engineering tools. The integration and traceability platform (ITP) that connected the tools together and facilitated navigation of the information flow consisted of the Subversion version control software (SVN) and IBM DOORS requirements management tool. To connect DOORS objects to files in SVN, the ITP was tailored by implementing a DXL based script to aid impact analysis to cover changes in external files. Although such a method did not provide optimal granularity and full visibility of data in all tools, a file level granularity of model elements was accomplished. The file level granularity was considered to provide a satisfactory solution for traceability. Furthermore the results and the chosen methods were recognised as a step forward from otherwise neglected requirements traceability.

The results of this thesis can be seen as a demonstration of how SEAModel can be utilised with a set of commonly used heterogeneous state-of-the-art systems engineering software tool brands. The implementation work of SEAModel can be realised on different database oriented software platforms, which suits well for small- and medium-sized enterprises that cannot afford a complete product life cycle management tool.

REFERENCES

Alanen, J., 2015. Tracing artefacts according to SEAModel in simulation oriented product development. Tampere, VTT.

Alanen, J., Isto, P., Tikka, P. & Tommila, T., 2015. Requirements traceability in simulation driven development, Espoo: VTT Technical Research Centre of Finland Ltd.

Alanen, J. & Valkama, P., 2014. SEPLM Scissors Platform Systems Engineering - desktop. [Online]
Available at: https://workspace.vtt.fi/sites/se_plm/saksinostin/default.aspx
[Accessed 15.8.2015].

Checkland, P. B. & Haynes, M. G., 1994. Varieties of Systems Thinking: the case of soft systems methodology. System Dynamics Review, 10(2-3), pp. 189-197.

Directive on machinery 2006/42/EC, 2006. Directive 2006/42/EC of The European Parliament and of The Council on machinery, and amending Directive 95/16/EC. Official Journal of the European Union, pp. 24-86.

Eclipse, 2016. Mars. [Online]
Available at: https://eclipse.org/mars/
[Accessed 1.10.2015].

Friedenthal, S., Moore, A. & Steiner, R., 2012. A Practical Guide to SysML - The Systems Modelling Language. 2nd ed. Waltham: Morgan Kaufmann, Object Management Group.

Genie lift , 2014. ROSTEK-Tekniikka Oy. [Online]
Available at: http://www.rostek-tekniikka.fi/images/esitteet/gs-sarja/Genie_GS-1530-1532-1930-1932_-esite_14.pdf
[Accessed 25.5.2015].

Guide to application of the Machinery Directive 2006/42/EC, 2010. Guide to application of the Machinery Directive 2006/42/EC. 2nd ed. Brussels: European Commission Enterprise and Industry.

Hubka, V. & Eder, W. E., 1988. Theory of Technical Systems, A Total Concepts Theory of Engineering Design. Berlin: Springer-Verlag.

INCOSE, 2006. INCOSE Systems Engineering Handbook, A Guide for System Life Cycle Processes and Activities. v.3 ed. International Council on Systems Engineering (INCOSE): INCOSE-TP-2003-002-03.

ISO/DIS 10303-233, 2009. Industrial automation systems and integration - Product data representation and exchange. Part 233: Application protocol: Systems engineering ed. Geneva: International Organization for Standardization (ISO). pp. 800.

ISO/IEC/IEEE 15288, 2015. Systems and software engineering - System life cycle processes. 1st ed. Geneva: ISO/IEC/IEEE.

JLG, 2016. JLG. [Online]
Available at: www.jlg.com/en-gb/equipment/scissor-lifts/electric/le-series/m3369le
[Accessed 25.5.2015].

Juuti, T., 2008. Design Management of Products with Variability and Commonality – Contribution to the Design Science by elaborating the fit needed between Product Structure , Design Process , Design Goals , and Design Organisation for Improved R&D Efficiency. Tampere, Tampere University of Technology.

Lamb, C. T. & Rhodes, D. H., 2009. Collaborative Systems Thinking: Uncovering the rules of team-level Systems Thinking, Systems Conference, 2009 3rd Annual IEEE, pp. 413-418: Institute of Electrical and Electronics Engineers,

Matei, I. & Bock, C., 2012. National Insitute of Standards and Technology. [Online]
Available at: http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7888.pdf
[Accessed 20.8.2015].

MathWorks, 2015. Managing Requirements for Fault-Tolerant Fuel Control System (IBM Rational DOORS). [Online]
Available at: http://se.mathworks.com/help/slvnv/examples/managing-requirements-for-fault-tolerant-fuel-control-system-ibm-rational-doors.html
[Accessed 10.9.2015].

Olesen, J., 1992. Concurrent Development in Manufacturing - Based on Dispositional Mechanisms. Institute for Engineering Design: Technical University of Denmark (DTH).

Ranolift Oy, 2016. Nostinvuokraamo. [Online]
Available at: http://www.nostinvuokraamo.fi/saksityolavat.html
[Accessed 25.5.2015].

SFS-EN 280, 2013. Mobile elevating work platforms – Design calculations – Stability criteria – Construction – Safety – Examinations and tests. 3rd ed. Brussels: European Committee for Standardization.

SFS-EN ISO 10007, 1996. Quality management - Guidelines for configuration management. Brussels: European Committee for Standardization.

SFS-EN ISO 12100, 2010. Safety of machinery - General principles for design - Risk assessment and risk reduction. 1st ed. Brussels: International Organisation for Standardization.

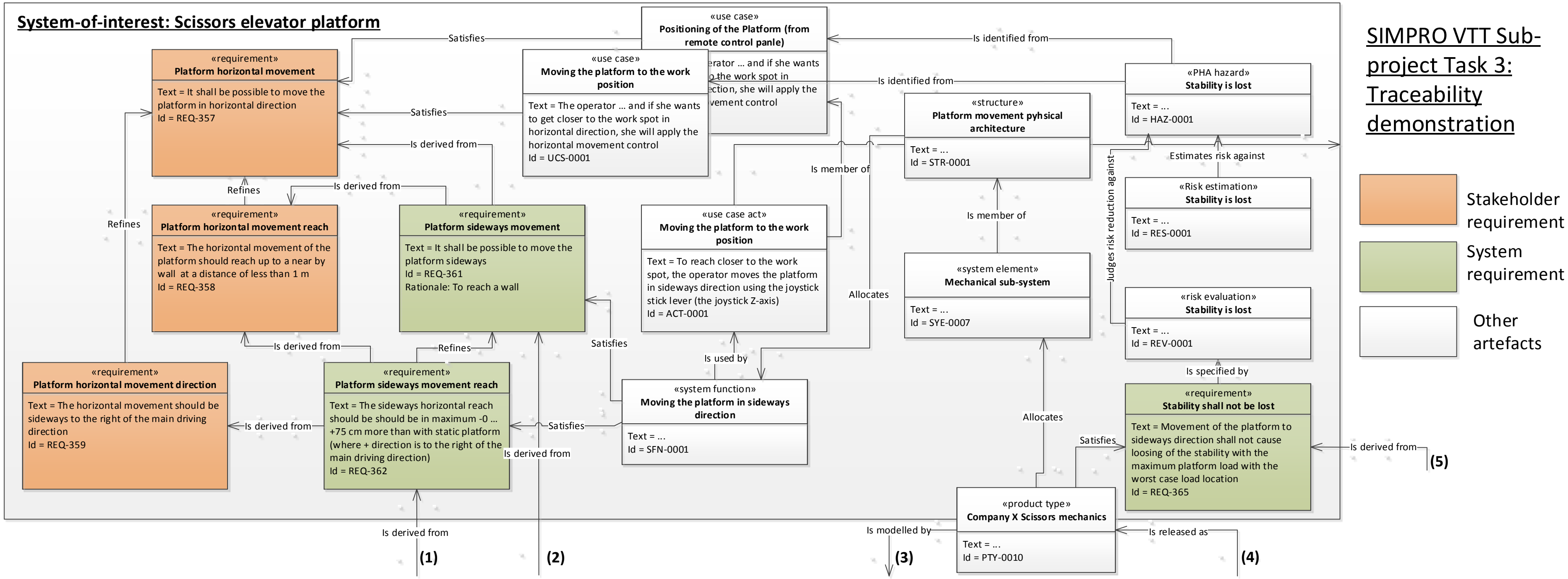
SFS-EN ISO 13849-2, 2012. Safety-Related Parts of Control Systems. Part 2: Validation (ISO 13849-2:2012). 2nd ed. Geneva: ISO copyright office.

Weber, C., 2012. Idea - Invention - Innovation: Stategies, Approaches, Research Challenges. In Proceedings of the 12th International Design Conference, Dubrovnik: University of Zagreb, pp. 1265–1274.

Weilkiens, T., 2006. Systems Engineering with SysML/UML - Modeling, Analysis, Design. 1st ed. Burlington: Morgan Kaufmann.

VTT, 2015. SIMPRO. [Online]
Available at: http://www.vtt.fi/sites/SIMPRO
[Accessed 16.3.2016].

APPENDIX A: TRACEABILITY DEMONSTRATION MODEL (1/3)



System-of-interest: Scissors elevator platform

«use case»
Positioning of the Platform (from remote control panle)
…erator … and if she wants …o the work spot in …ection, she will apply the …vement control

«use case»
Moving the platform to the work position
Text = The operator … and if she wants to get closer to the work spot in horizontal direction, she will apply the horizontal movement control
Id = UCS-0001

«requirement»
Platform horizontal movement
Text = It shall be possible to move the platform in horizontal direction
Id = REQ-357

«requirement»
Platform horizontal movement reach
Text = The horizontal movement of the platform should reach up to a near by wall at a distance of less than 1 m
Id = REQ-358

«requirement»
Platform sideways movement
Text = It shall be possible to move the platform sideways
Id = REQ-361
Rationale: To reach a wall

«requirement»
Platform horizontal movement direction
Text = The horizontal movement should be sideways to the right of the main driving direction
Id = REQ-359

«requirement»
Platform sideways movement reach
Text = The sideways horizontal reach should be should be in maximum -0 … +75 cm more than with static platform (where + direction is to the right of the main driving direction)
Id = REQ-362

«use case act»
Moving the platform to the work position
Text = To reach closer to the work spot, the operator moves the platform in sideways direction using the joystick stick lever (the joystick Z-axis)
Id = ACT-0001

«system function»
Moving the platform in sideways direction
Text = …
Id = SFN-0001

«structure»
Platform movement pyhsical architecture
Text = …
Id = STR-0001

«system element»
Mechanical sub-system
Text = …
Id = SYE-0007

«product type»
Company X Scissors mechanics
Text = …
Id = PTY-0010

«PHA hazard»
Stability is lost
Text = …
Id = HAZ-0001

«Risk estimation»
Stability is lost
Text = …
Id = RES-0001

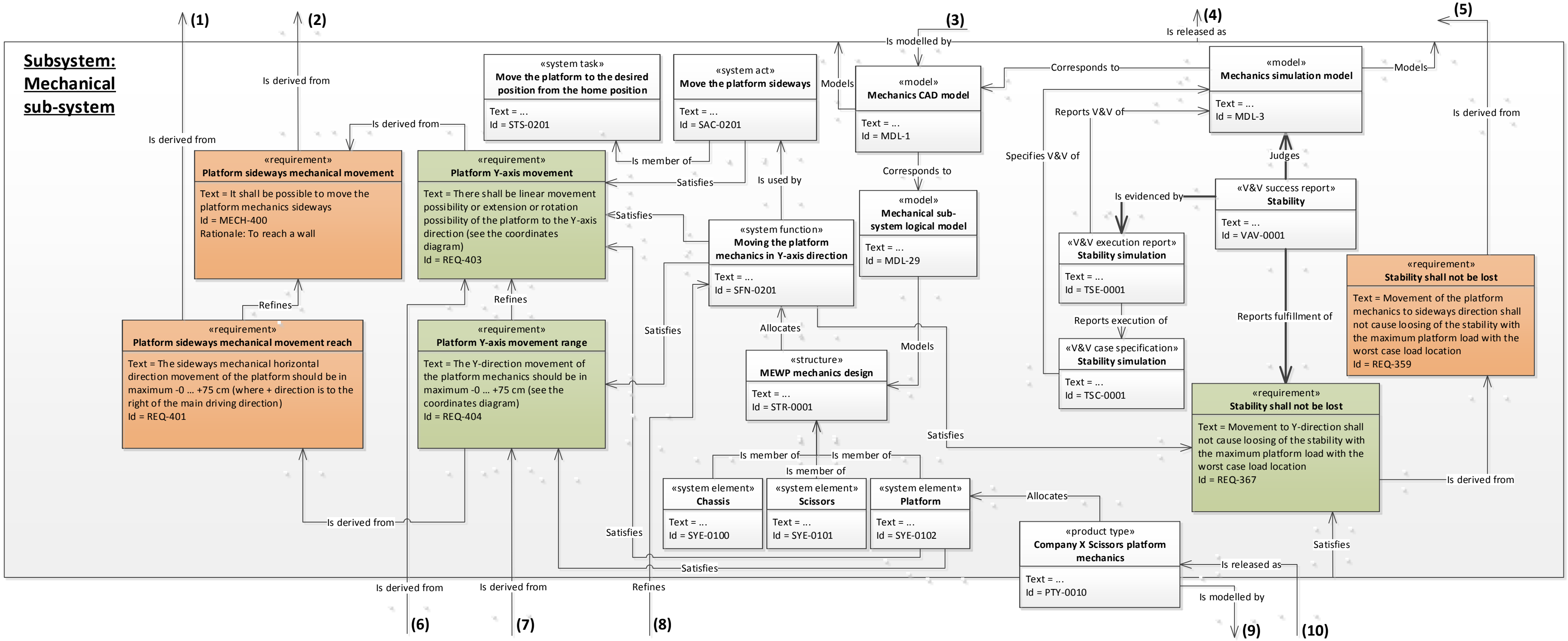
«risk evaluation»
Stability is lost
Text = …
Id = REV-0001

«requirement»
Stability shall not be lost
Text = Movement of the platform to sideways direction shall not cause loosing of the stability with the maximum platform load with the worst case load location
Id = REQ-365

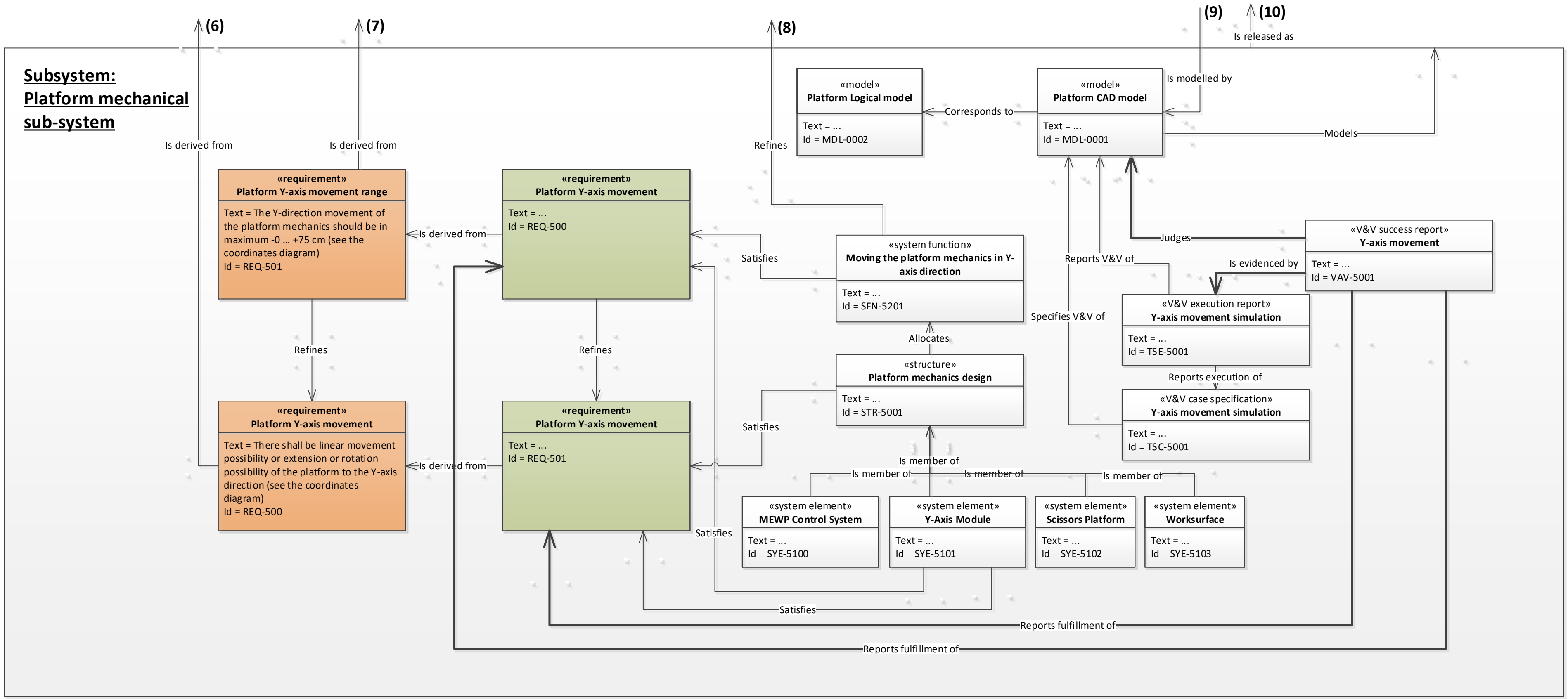
SIMPRO VTT Sub-project Task 3: Traceability demonstration

🟧 Stakeholder requirement
🟩 System requirement
⬜ Other artefacts

APPENDIX A: TRACEABILITY DEMONSTRATION MODEL (2/3)



APPENDIX A: TRACEABILITY DEMONSTRATION MODEL (3/3)

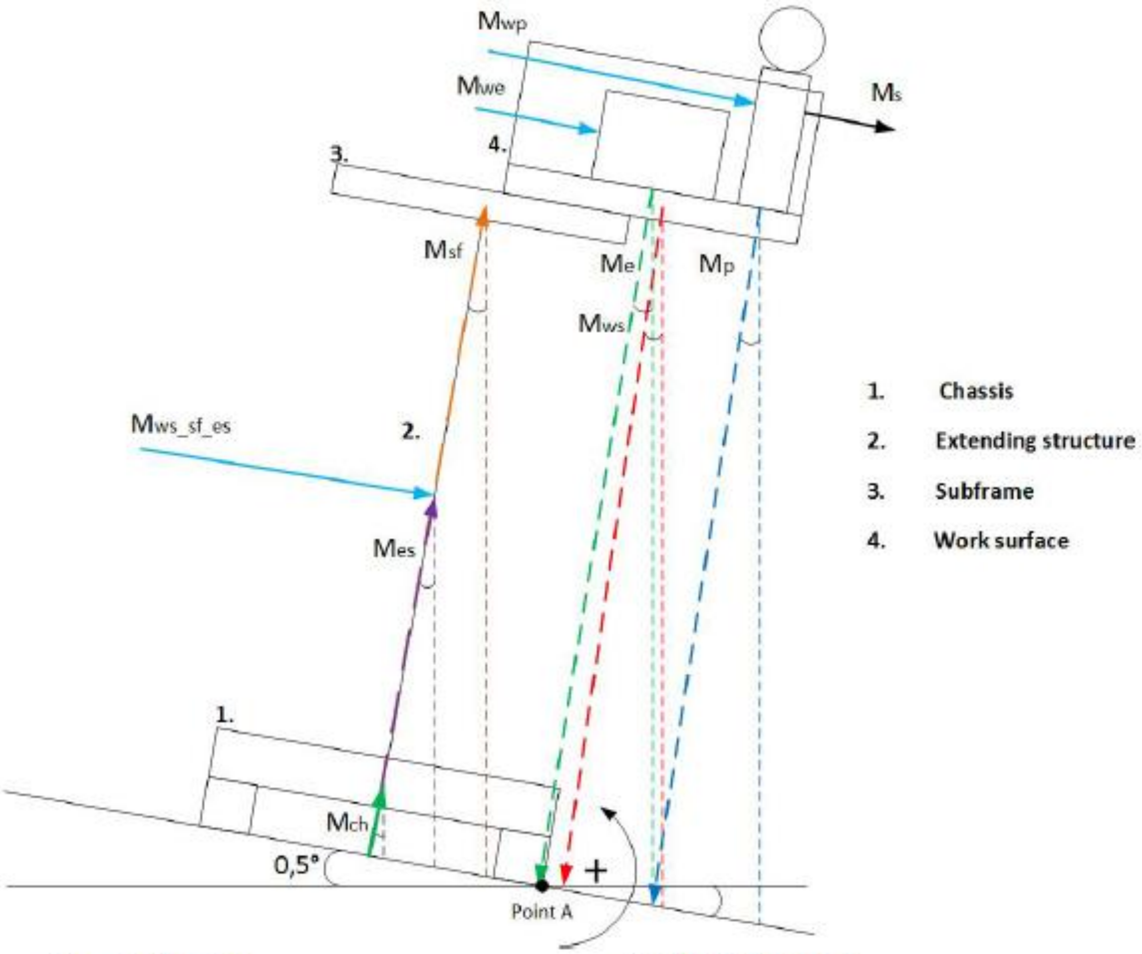


APPENDIX B: STABILITY ANALYSIS OF THE MEWP

Stability analysis of the MEWP

These calculations are done to verify the results of the SimMechanics simulation model regarding the stability of the mobile elevating work platform (MEWP). Calculation include simplifications, but they give a reference to the simulation model. Used parameters are derived from the CAD model designed for the MEWP. Stability of the MEWP is calculated by defining the minimun mass of the chassis, which prevents the structure from falling under the conditions set by SFS-EN 280.

Text highlighted in RED indicates a simplification or a notable value that has been used.



$m_{max} := 900\ \boldsymbol{kg}$	maximum workload
$m_p := 80\ \boldsymbol{kg}$	mass of a person
$n_h := 2$	number of persons
$m_e := m_{max} - \left(n_h \cdot m_p\right) = 740\ \boldsymbol{kg}$	mass of equipment

Mass of the main components of the MEWP

$m_{es} := 604\ kg$ extending structure

$m_{sf} := (203 + 52)\ kg = 255\ kg$ subframe + rails of the module

$m_{ws} := (152 + 55)\ kg = 207\ kg$ work surface + plate of the module

Dimensioning of the main components and tensions indicators

Tension levers are dimensioned according to the vertical projection of the MEWP.

$A_{ws} := (2628.9\ mm \cdot 1206.5\ mm) = 3.172\ m^2$ square area of the work surface

$sf_{korkeus} := 198.2\ mm = 0.198\ m$ height of the subframe

$ws_{korkeus} := 53.56\ mm = 0.054\ m$ height of the work surface

$np_{korkeus} := 8\ m$ maximun lift height of the work surface

$es_y := \frac{np_{korkeus}}{2} = 4\ m$ force moment indicator in y-axis direction of the centre of mass of the extending structure

$sf_y := np_{korkeus} - \frac{sf_{korkeus}}{2} = 7.901\ m$ force moment indicator in y-axis direction of the center of mass of the subframe

$ws_y := np_{korkeus} - \frac{ws_{korkeus}}{2} = 7.973\ m$ force moment indicator in y-axis direction of the centre of mass of the work surface

Tipping lines shall be determined in accordance with ISO 4305 but for solid and foam-filled tyres the tipping lines may be taken at 1/4 of the tyre ground contact width from the outside of the ground contact width. (SFS-EN 280)

$r := 248.33\ mm \cdot 0.25 = 0.062\ m$

$ch_{leveys} := 761.98\ mm = 0.762\ m$ distance from the centre of mass of the chassis to the outer edge of the chassis

$r_t := ch_{leveys} - r = 0.7\ m$ tipping line of the MEWP measured from the centre of mass of the chassis (Point A)

Work surface moves 75 cm to the horizontal y-axis direction

$y_h := 750\ mm = 0.75\ m$

$ws_x := y_h = 0.75\ m$ force moment indicator in x-axis direction of the centre of mass of the work surface (approximation to ease the calculus)

Note. The vertical projection of the centre point of the work surface is outside of the tipping line

$$r_{työtason\_pinnan\_keskipisteen\_sijainti} := r_t - ws_x = -0.05\ m$$

The mass of each person is assumed to act as a point load on the work surface and any platform extension at a horizontal distance of 0,1 m from the upper inside edge of the top rail. The distance between the point loads shall be 0,5 m. (SFS-EN 280)

The people and equipment loads are assumed to be point loads on the work surface. Therefore, the force moment indicator in y-axis direction is the same as is the maximum height of the work surface.

$$p_y := np_{korkeus} = 8\ m$$

force moment indicator of a person's point load in y-axis direction

$$p_x := \frac{1221.2\ mm}{2} + y_h - 100\ mm = 1.261\ m$$

force moment indicator of a person's point load in x-axis direction

The mass of equipment is assumed to act as an evenly distributed load on 25 % of the floor of the work surface. If the resulting pressure exceeds 3 kN/m2 the figure of 25 % may be increased to a figure giving a pressure of 3 kN/m2.

Pressure caused by equipment:

$$A_{ws} := (2628.9 \cdot 1206.5 \cdot 0.25)\ mm^2 = 0.793\ m^2$$

$$p_e := \frac{m_e}{A_{ws}} = 933.233\ \frac{kg}{m^2} \qquad p_e \cdot g = 9.152\ \frac{kN}{m^2}$$

Pressure exceeds 3 kN/m2 aka the value of 25% needs to be raised.

$$p_{e\_2} := 3\ \frac{\frac{kN}{m^2}}{g} = 305.915\ \frac{kg}{m^2}$$

$$A_{ws\_2} := \frac{m_e}{p_{e\_2}} = 2.419\ m^2$$

new surface area of the equipment

The mass of equipment is evenly distributed:

$$E_m := \frac{A_{ws\_2}}{(2629.9 \cdot 1206.5)\ mm^2} = 0.762$$

The mass of equipment is assumed to act as an evenly distributed load on ~76% of the floor of the work surface.

$$E_x := \left(750\ mm + \frac{1206.5\ mm}{2}\right) - \frac{\left(\left(750\ mm + \frac{1206.5\ mm}{2}\right) \cdot E_m\right)}{2} = 0.837\ m$$

force moment indicator of equipment's centre of mass in x-axis direction

$E_y := np_{korkeus} = 8\ m$ force moment indicator of equipment's centre of mass in y-axis direction

An allowance of 0,5° for inaccuracy in setting-up the MEWP shall be added to the maximum allowable inclination of the chassis permitted by the manufacturer. (SFS-EN 280)

$ang := 0.5\ deg$ allowance for inaccuracy

Force moments acting in vertical direction in terms of tipping line aka Point A:

$$M_p := \left(n_h \cdot m_p \cdot g\right) \cdot \left(p_x - r_t + p_y \cdot \tan(ang)\right) \cdot \cos(ang) = 989.285\ N \cdot m$$

$$M_e := \left(m_e \cdot g\right) \cdot \left(E_x - r_t + E_y \cdot \tan(ang)\right) \cdot \cos(ang) = \left(1.505 \cdot 10^3\right)\ N \cdot m$$

$$M_{ws} := \left(m_{ws} \cdot g\right) \cdot \left(ws_x - r_t + ws_y \cdot \tan(ang)\right) \cdot \cos(ang) = 242.946\ N \cdot m$$

$$M_{sf} := \left(m_{sf} \cdot g\right) \cdot \left(-r_t + sf_y \cdot \tan(ang)\right) \cdot \cos(ang) = -1.578 \cdot 10^3\ N \cdot m$$

$$M_{es} := \left(m_{es} \cdot g\right) \cdot \left(-r_t + es_y \cdot \tan(ang)\right) \cdot \cos(ang) = -3.939 \cdot 10^3\ N \cdot m$$

Forces acting in horizontal direction:

The full area of one person shall be 0,7 m2 (0,4 m average width × 1,75 m height) with the centre of area 1,0 m above the work surface floor. All MEWPs used out-of-doors are regarded as being affected by wind at a pressure of 100 N/m2, equivalent to a wind speed of 12,5 m/s (Beaufort Scale 6). (SFS-EN 280)

Wind forces shall be multiplied by a factor of 1,1 and taken to be acting horizontally. Manual forces applied by persons on the work surface shall be multiplied by a factor of 1,1 and taken to be acting in the direction creating the greatest overturning moment. (SFS-EN 280)

Shape factors applied to areas exposed to wind:
a) L-, U-, T-, I-sections: 1,6;
b) box sections: 1,4;
c) large flat areas: 1,2;
d) circular sections, according to size: 0,8/1,2;
e) persons directly exposed: 1,0.

$$p_w := 100\ \frac{N}{m^2}$$

$p_{y\_tuuli} := np_{korkeus} + 1.0\ m = 9\ m$ force moment indicator of a person's wind surface area in y-axis direction

$F_{wp} := \left(0.7\ m^2 \cdot p_w\right) \cdot n_h \cdot 1.0 = 140\ N$

wind force affecting a person

The wind force on exposed tools and materials on the work surface shall be calculated as 3 % of their mass, acting horizontally at a height of 0,5 m above the work surface floor. (SFS-EN 280)

$E_{y\_tuuli} := np_{korkeus} + 0.5\ m = 8.5\ m$

force moment indicator of an equipment's wind surface area in y-axis direction

$F_{we} := 0.03 \cdot m_e \cdot g \cdot 1.2 = 261.249\ N$

wind force affecting equipment

The minimum value for the manual force M shall be taken as 200 N for MEWPs designed to carry only one person and 400 N for MEWPs designed to carry more than one person, applied at a height of 1,1 m above the work surface floor. Any greater force permitted shall be stated by the manufacturer. (EN 280)

$F_S := 400\ N \cdot 1.1 = 440\ N$

manual force produced by the people

$S_y := np_{korkeus} + 1.1\ m = 9.1\ m$

force moment indicator of manual force in y-axis direction

Wind forces affecting the structure of the MEWP:

Calculations for the wind forces include simplifications and the results are suggestive.

$A_{ws\_sf\_es} := (2643.60\ mm \cdot 404.10\ mm) + (2438.40\ mm \cdot 100\ mm \cdot 8) = 3.019\ m^2$

wind surface area for the extending structure + subframe + work surface

$F_{ws\_sf\_es} := p_w \cdot A_{ws\_sf\_es} \cdot 1.6 = 483.04\ N$

wind force affecting the extending structure + subframe + work surface

Force moments acting in horizontal direction in terms of tipping line aka Point A:

To simplify the calculus, the horizontal force moment indicators are presumed to consis only from the distance in y-axis direction measured from Point A.

$M_{wp} := F_{wp} \cdot P_{y\_tuuli} = \left(1.26 \cdot 10^3\right)\ N \cdot m$

force moment caused the people

$M_{we} := F_{we} \cdot E_{y\_tuuli} = \left(2.221 \cdot 10^3\right)\ N \cdot m$

force moment caused by the equipment

$$M_S := F_S \cdot S_y = \left(4.004 \cdot 10^3\right)\ N \cdot m$$

force moment caused by the manual force of the people

$$M_{ws\_sf\_es} := F_{ws\_sf\_es} \cdot \frac{np_{korkeus}}{2} = \left(1.932 \cdot 10^3\right)\ N \cdot m$$

force moment caused by the extending structure + subframe + work surface

Mass of the chassis according to the force moments:

The rotation axis of Point A is chose to be counter clockwise. The MEWP is stabile, when the sum of force moments affecting Point A is zero.

$$M_{ch} := (-M_p) + (-M_e) + (-M_{ws}) + (-M_{sf}) + (-M_{es}) + (-M_{wp}) + (-M_{we}) + (-M_S) + (-M_{ws\_sf\_es}) = -6.637 \cdot 10^3\ J$$

The Mass of the chassis resolved according to the force moments:
M_ch = (m_ch*g)*(-r_t + ch_y*tan(0.5))*cos(0.5)

$ch_y := 343.7\ mm$ center of mass of the chassis

$$m_{ch} := \frac{(M_{ch})}{g \cdot \left(-r_t + ch_y \cdot \tan(ang)\right) \cdot \cos(ang)} = 971.185\ kg$$

mass of the chassis

Safety margin is regarded in the mass of the chassis:

$m_{ch2} := 1000\ kg$ mass of the chassis with a safety margin

Total mass of the MEWP:

$$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch2} = \left(2.066 \cdot 10^3\right)\ kg$$

mass of the MEWP

Stability factor of the MEWP:

Stability factor regards the relation of overturning and stabilising moments. The point of momentum is the previously defined Point A, which is the tipping line of the MEWP. The direction rotation axis counter clockwise.

$$M_{ch2} := \left(m_{ch2} \cdot g\right) \cdot \left(-r_t + ch_y \cdot \tan(ang)\right) \cdot \cos(ang) = -6.834 \cdot 10^3\ N \cdot m$$

force moment of the chassis in relevance to Point A

Overturning moment:

$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = \left(1.215 \cdot 10^4\right) N \cdot m$$

Stabilising moment:

$$M_u := \left(-M_{sf}\right) + \left(-M_{es}\right) + \left(-M_{ch2}\right) = \left(1.235 \cdot 10^4\right) N \cdot m$$

Stability factor:

$$S_{stb} := \frac{M_u}{M_f} = 1.016$$

The lower the value of the stability factor, the more unstablile the system. A sufficient treshold for a stabile system is considered here to be stability factor value 1,1.

Different stability factor values in accordance to different masses of the chassis:

$m_{ch\_1} := 1100\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_1} = \left(2.166 \cdot 10^3\right) kg$ mass of the MEWP

$$M_{ch\_1} := \left(m_{ch\_1} \cdot g\right) \cdot \left(-r_t + ch_y \cdot \tan(ang)\right) \cdot \cos(ang) = -7.517 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = \left(1.215 \cdot 10^4\right) N \cdot m$$
$$M_u := \left(-M_{sf}\right) + \left(-M_{es}\right) + \left(-M_{ch\_1}\right) = \left(1.303 \cdot 10^4\right) N \cdot m$$

$S_{stb\_1} := \dfrac{M_u}{M_f} = 1.072$ stability factor

$m_{ch\_2} := 1150\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_2} = \left(2.216 \cdot 10^3\right) kg$ mass of the MEWP

$$M_{ch\_2} := \left(m_{ch\_2} \cdot g\right) \cdot \left(-r_t + ch_y \cdot \tan(ang)\right) \cdot \cos(ang) = -7.859 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = \left(1.215 \cdot 10^4\right) N \cdot m$$
$$M_u := \left(-M_{sf}\right) + \left(-M_{es}\right) + \left(-M_{ch\_2}\right) = \left(1.338 \cdot 10^4\right) N \cdot m$$

$S_{stb\_2} := \dfrac{M_u}{M_f} = 1.101$ stability factor

$m_{ch\_3} := 1200\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_3} = \left(2.266 \cdot 10^3\right) kg$ mass of the MEWP

$$M_{ch\_3} := (m_{ch\_3} \cdot g) \cdot (-r_t + ch_y \cdot \tan(ang)) \cdot \cos(ang) = -8.201 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = (1.215 \cdot 10^4)\ N \cdot m$$
$$M_u := (-M_{sf}) + (-M_{es}) + (-M_{ch\_3}) = (1.372 \cdot 10^4)\ N \cdot m$$

$S_{stb\_3} := \frac{M_u}{M_f} = 1.129$ stability factor

$m_{ch\_4} := 1300\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_4} = (2.366 \cdot 10^3)\ kg$ mass of the MEWP

$$M_{ch\_4} := (m_{ch\_4} \cdot g) \cdot (-r_t + ch_y \cdot \tan(ang)) \cdot \cos(ang) = -8.884 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = (1.215 \cdot 10^4)\ N \cdot m$$
$$M_u := (-M_{sf}) + (-M_{es}) + (-M_{ch\_4}) = (1.44 \cdot 10^4)\ N \cdot m$$

$S_{stb\_4} := \frac{M_u}{M_f} = 1.185$ stability factor

$m_{ch\_5} := 1320\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_5} = (2.386 \cdot 10^3)\ kg$ mass of the MEWP

$$M_{ch\_5} := (m_{ch\_5} \cdot g) \cdot (-r_t + ch_y \cdot \tan(ang)) \cdot \cos(ang) = -9.021 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = (1.215 \cdot 10^4)\ N \cdot m$$
$$M_u := (-M_{sf}) + (-M_{es}) + (-M_{ch\_5}) = (1.454 \cdot 10^4)\ N \cdot m$$

$S_{stb\_5} := \frac{M_u}{M_f} = 1.196$ stability factor

$m_{ch\_6} := 1400\ kg$ mass of the chassis

$m_{tot} := m_{ws} + m_{sf} + m_{es} + m_{ch\_6} = (2.466 \cdot 10^3)\ kg$ mass of the MEWP

$$M_{ch\_6} := (m_{ch\_6} \cdot g) \cdot (-r_t + ch_y \cdot \tan(ang)) \cdot \cos(ang) = -9.568 \cdot 10^3\ N \cdot m$$
$$M_f := M_p + M_e + M_{ws} + M_{wp} + M_{we} + M_S + M_{ws\_sf\_es} = (1.215 \cdot 10^4)\ N \cdot m$$
$$M_u := (-M_{sf}) + (-M_{es}) + (-M_{ch\_6}) = (1.508 \cdot 10^4)\ N \cdot m$$

$S_{stb\_6} := \frac{M_u}{M_f} = 1.241$ stability factor

By comparing total masses of the MEWP:s to equivalent commercial products, and noting the calculated values of the stability factors, it can be stated that chassis 1150kg (S_stb=1.101) and 1320kg (S_stb=1.196) are the most respectable choises.