Aalto University
School of Electrical
Engineering

Master's Programme in Automation and Electrical Engineering

Assessing the Suitability of Software Tools for the System-Theoretic Process Analysis of Nuclear Instrumentation and Control Systems

Carl Akira King

Master's Thesis
2024

Aalto University
School of Electrical
Engineering

Author Carl Akira King

Title Assessing the Suitability of Software Tools for the System-Theoretic Process Analysis of Nuclear Instrumentation and Control Systems

Degree programme Automation and Electrical Engineering

Major Control, Robotics and Autonomous Systems

Supervisor Prof. Valeriy Vyatkin

Advisor Dr Polina Ovsiannikova

Collaborative partner VTT

Date 30 September 2024 **Number of pages** 80+23 **Language** English

Abstract

System-Theoretic Process Analysis (STPA) is a promising, novel hazard analysis method that is capable of analyzing modern software-intensive systems, such as Instrumentation & Control (I&C) systems used in nuclear power plant modernization efforts. However, the method has yet to be fully adopted in the nuclear industry. One step toward the adoption of STPA is a suitable software tool for conducting STPA analyses in the industrial domain. In order to determine such a tool, this thesis evaluates the available software tools against tool requirements generated following requirements engineering principles. Two focus group sessions were conducted as a part of the requirements engineering process, during which requirements generated based on findings in the research domain were discussed and assessed together with STPA practitioners from both the industrial and research domains. The evaluation of STPA software tools was undertaken using case study data from the STPA analysis of a nuclear reactor feedwater system.

The thesis results include both the results of the requirements engineering process and the results of the software tool evaluation. The former discovered that robust traceability and Control Structure features are paramount for a suitable software tool, while the latter determined that many software tools still lack in these aspects as well as in their documentation. However, many tools presented promising individual features, and one tool showed promise for adoption in the nuclear domain.

Keywords I&C, nuclear power plant, software tools, STPA, process automation

Aalto-yliopisto
Sähkötekniikan
korkeakoulu

Tekijä Carl Akira King

Työn nimi Ohjelmistotyökalujen soveltuvuuden arvionti ydinalan automaatiojärjestelmien systeemiteoreettiseen prosessianalyysiin

Koulutusohjelma Automation and Electrical Engineering

Pääaine Control, Robotics and Autonomous Systems

Työn valvoja Prof. Valeriy Vyatkin

Työn ohjaaja Dr Polina Ovsiannikova

Yhteistyötaho VTT

Päivämäärä 30.9.2024 **Sivumäärä** 80+23 **Kieli** englanti

Tiivistelmä

Systeemiteoreettinen prosessianalyysi (STPA) on lupaava, uusi hasardianalyysimenetelmä, joka kykenee analysoimaan moderneja ohjelmisto-painotteisia järjestelmiä, kuten ydinvoimaloiden modernisoinnissa hyödynnettäviä automaatiojärjestelmiä. STPA ei ole kuitenkaan vielä vakiintunut teollisuuden menetelmiin. Yksi edellytys STPA-menetelmän vakiinnuttamiseksi on menetelmän teollisuudessa hyödyntämistä edesauttava ohjelmistotyökalu. Tämä opinnäytetyö arvioi olemassa olevia ohjelmistotyökaluja verraten niitä vaatimusten määrittely (engl. Requirements Engineering) prosessin avulla kehitettyihin vaatimuksiin. Osana vaatimusten määrittely prosessia järjestettiin kaksi työpajaa, joissa keskusteltiin ja arvioitiin tutkimuskirjallisuuteen perustuvia alustavia työkaluvaatimuksia yhdessä sekä teollisuuden STPA osaajien, että STPA tutkijoiden kanssa. Ohjelmistotyökalut arvioitiin arvioitiin syöttämällä niihin tapaustutkimuksen tuloksia ydinvoimalan reaktorin syöttövesijärjestelmän STPA analyysistä.

Opinnäytetyön tulokset koostuvat sekä vaatimusten määrittely prosessin, että ohjelmistotyökalujen arvioinnin tuloksista. Vaatimusten määrittely prosessin tärkeimpiä havaintoja olivat jäljitettävyyteen ja STPA ohjausrakenteisiin liittyvien ominaisuuksien tärkeys ohjelmistotyökalun soveltuvuuden kannalta. Arvioinnin tuloksissa ilmeni monien ohjelmistotyökalujen toimimattomuus näillä osa-alueilla, sekä useiden ohjelmistotyökalujen käyttöä tukevan dokumentaation puute. Arvioinnissa kuitenkin myös korostuivat eri ohjelmistotyökalujen yksittäiset, lupaavat ominaisuudet. Yksi arvioiduista työkaluista on lupaava ydinvoimateollisuuden käyttöön.

Avainsanat automaatiojärjestelmät, ohjelmistotyökalut, prosessi automaatio, STPA, ydinvoimala

Preface

As I began this enormous task, I realized that everything I have learnt in my years at Aalto University was coming together in this one job. In determining requirements I got to utilize my background in Aalto University's Design minor, where a special emphasis is placed on human-centric design and understanding the needs of different stakeholders. In understanding STPA and applying STPA in the software tools, I drew from all my past courses in automation both in my Bachelor's and Master's studies. As someone with this background, understanding the STAMP model, Control Structures and control loops was intuitive.

While all the skills and knowledge accrued over the years were certainly required to undertake the challenges of determining requirements and evaluating software tools, the real challenges of this thesis came from within. As an uncompromising individual with a detail-oriented mindset and a rigid set of self-imposed standards, learning when to settle was one of the biggest challenges in this work. Even now, as I look at the final PDF of the thesis, I see parts of the thesis I wish I had done differently, or parts where I hoped to have been more thorough. One such regret is that I didn't get to use non-participant observation as a method for generating requirements. Observing an STPA analysis being conducted, especially the meetings with system experts, would have surely provided invaluable insights towards the generation of requirements.

First and foremost, I would like to thank Professor Valeriy Vyatkin for providing this opportunity for me to develop both as an individual and in an academic capacity. In addition to his support in this work, his lectures over the years have been some of the highlights of my time at Aalto University.

In their continuous and insightful support, Dr. Polina Ovsiannikova from Aalto University, as well as Josepha Berger and Antti Pakonen from VTT deserve the most sincere expression of gratitude. In addition to their unwavering support, their company ensured that I enjoyed every day of working on this thesis.

One of the activities that kept me sane this summer was track cycling with my close friends. On most dry days, quick morning sessions at the Käpylä velodrome ensured I had the capacity to sit still and focus for the rest of the day, working on the thesis. At times when the thesis work was too much, these sessions provided a meaningful alternative focus point. To my close friends, I'm grateful for every moment I spent with you this summer, thank you.

Finally, two people deserve a special mention. Both my partner and my brother have been the support one can only dream of. On countless instances during the past 6 months I relied on their kind and supportive words for motivation to continue on. I continue to be surprised by the strength of the connection I feel with you. So I guess it comes as no suprise, that I'm beyond grateful for everything you've been to me this summer. Thank you.

Otaniemi, 30 September 2024

Carl Akira King

Contents

Abbreviations

CA	Control Action
CAST	Causal Analysis based on System Theory
FMEA	Failure Modes and Effects Analysis
FTA	Fault Tree Analysis
HAZOP	Hazard and Operability Analysis
IAEA	International Atomic Energy Agency
I&C	Instrumentation and Control
NPP	Nuclear Power Plant
PRA	Probabilistic Risk Assessment
STAMP	System-Theoretic Accident Model and Processes
STPA	System-Theoretic Process Analysis
STUK	Radiation and Nuclear Safety Authority
UCA	Unsafe Control Action

1 Introduction

Instrumentation and control (I&C) systems form the backbone of many industrial processes from the production of commercial goods to power generation. These systems play a key role in defining how industrial processes behave and whether they are, for example, efficient or safe. As with other industrial processes, power generation in nuclear power plants relies heavily on I&C systems performing as intended. Most operational power plants, however, were designed and built decades ago and operate with equally old I&C systems [1]. These aging I&C systems are becoming increasingly difficult to maintain and may result in compromises in performance and safety [2].

To address the issue of aging I&C systems, many nuclear power plants are undergoing modernization, or nuclear power plant licensees are investigating their modernization. Modernization, however, is not always as straightforward as replacing the old components with new ones. The older analog and electromechanical components may have behavioral characteristics known to the original system designers but unknown to those modernizing the system with newer digital components, resulting in unintended consequences in the system's behavior [1]. Additionally, when simply replacing older components, many of the benefits of newer digital components remain unutilized. These benefits include increased diagnostic or monitoring capabilities [1]. Modernization, therefore, often requires redesigning the system at least partially [2].

While modern digital components and their software-intense approaches to I&C systems are used in many industries, the nuclear domain requires these systems to be held to a higher standard, especially with regard to safety. Safety is often assessed using well-established hazard analysis methods such as FMEA or PRA. These methods are suitable for the analysis of older analog and electromechanical systems, where component reliability plays a significant role in safety. However, these methods do not address many of the safety issues common in modern software-intense systems [3]. In these systems, safety is often compromised due to the unintended behavior of the software stemming from flaws in their design or implementation. Hence, alternative hazard analysis methods are required for effectively evaluating the safety of modern software-intensive systems.

One potential alternative to the conventional hazard analysis methods is System-Theoretic Process Analysis (STPA) [3]. It is a hazard analysis method that has been studied extensively in the context of nuclear and safety-critical systems and has been shown to provide valuable information to aid system design. It is capable of exposing issues in systems that conventional methods are unable to, such as issues arising from complex interactions between system components [3]. While STPA has proven its capabilities in identifying these issues in the research domain, it is still fairly novel to industrial domains [4]. Its practical implementation in analyzing systems in the nuclear context remains unestablished, as regulators and nuclear power plant licensees have yet to fully adopt the method. Contributing to the method's novelty, it is being actively researched with a significant revision to the method introduced in 2018 [3] and extensions to the method developed and proposed since [4] [5] [6].

One obstacle to the adoption of STPA in the nuclear industry is the lack of suitable software tools for the method. The STPA method is often employed using rudimentary

approaches, such as simple Excel spreadsheets and Microsoft Visio. While these approaches satisfy the need to conduct isolated STPA analyses with narrower scopes or higher levels of abstraction, they quickly become inadequate in larger analyses. Conducting the analysis requires referring to items produced earlier in the analysis, and the number of such items can quickly rise to hundreds and thousands. As the analysis progresses, it becomes increasingly difficult to manage the amount of information in Excel spreadsheets [7]. Consequently, prior research has identified the need to determine suitable alternative software tools for conducting STPA analyses [8] [9].

While many STPA software tools have been developed, and their suitability in other contexts has been evaluated [10], their suitability in the context of the nuclear domain has not yet been evaluated. The software tools also vary greatly in their capabilities and features with some of the tools being able to help in just parts of the STPA analysis while others are capable of integrating the method into the larger systems engineering processes. Some of the available tools may inhibit learning the STPA method as identified in prior research [11], while ideally, the software tool should ease the learning curve. This could be due to, for example, an unintuitive user interface combined with a lack of experience with the STPA method. These factors together call for a comprehensive evaluation of the existing software tools in the nuclear context.

In order to evaluate available software tools for conducting STPA analyses in the nuclear domain, clear criteria for these tools need to be determined. Defining clear criteria for an STPA software tool provides the additional benefit of laying the foundation for developing a suitable software tool or guiding the further development of existing software tools. The aim of this thesis is to determine the requirements for STPA software tools in the context of the Finnish nuclear industry and to evaluate the existing software tools against these requirements. The requirements for these software tools will be generated through cooperation with STPA practitioners in the Finnish nuclear industry and STPA experts from the research domain. The tools will then be evaluated against these requirements using the data from a case study analysis conducted as a part of prior research [11]. This thesis is a part of the SEAMLES project, which is funded by the National Nuclear Safety and Waste Management Research Programme 2023-2028 (SAFER2028).

The remainder of the thesis is structured as follows. Chapter 2 presents the background for the study, including an overview of STPA and the nuclear domain. Chapter 3 identifies preliminary requirements through a review of the existing literature and presents the final requirements revised together with practitioners and STPA experts. Chapter 4 presents an overview of the existing software tools and an evaluation of the tools against the requirements. Chapter 5 concludes this work and presents a discussion of the results.

2 Background

This chapter introduces the background relevant to the topics covered in this thesis, such as STAMP and STPA, safety critical systems, the reactor feedwater case study, requirements engineering, and STPA related issues identified in existing research literature.

2.1 System-Theoretic Process Analysis

System-Theoretic Process Analysis (STPA) is a hazard analysis method, based on the System Theoretic Accident Model and Processes (STAMP). The latter was introduced in 2004 [12], and has since served as the foundation for STPA and Causal Analysis using System Theory (CAST) used for post accident analysis. This section will briefly introduce the STAMP accident model and present in further detail the STPA hazard analysis method.

2.1.1 STAMP

STAMP is an accident model, which, due to being based on a system-theoretic approach, treats accidents as a result of inadequate control. In STAMP, systems are seen as consisting of subsystems interacting with each other through relations of control and feedback. These subsystems and their interactions are responsible for ensuring that the system as a whole operates under the defined constraints. These constraints should limit the system states to those that are considered safe. [12]

An example of such a system is a logistics operation, where a dispatcher and delivery driver are responsible for completing deliveries. On a highly abstracted level, such a system could be composed of a customer, dispatcher, delivery driver, and the delivery process. From a system-theoretic approach, this system could be constrained such that the delivery driver is safe, and the delivery process is punctual. Interactions between each of these subsystems determine whether the system operates within the defined constraints. For example, ineffective communication between the dispatcher and delivery driver may result in a misunderstanding regarding a delivery, leading to a violation of the system's constraint regarding punctuality. The hierarchical view of the system described here, is referred to as a Control Structure, and is visualized in Figure 1.

A central concept within STAMP is that of control loops. The first aspect of a control loop is control asserted by entities higher up in the hierarchical representation of a system, such as a dispatcher in the previous example of a logistics operation. The second aspect is feedback provided by entities lower in the hierarchy, such as the delivery process to a delivery driver, or a delivery driver to a dispatcher in the previous example. Adequate feedback is essential to the control loop, ensuring that the entity conducting the decision making and asserting control are informed of the states within the system that they are responsible for constraining. For example, by informing the dispatcher of an issue they have faced, the delivery driver can ensure that they are given more time to complete the deliveries, and the dispatcher can assign orders

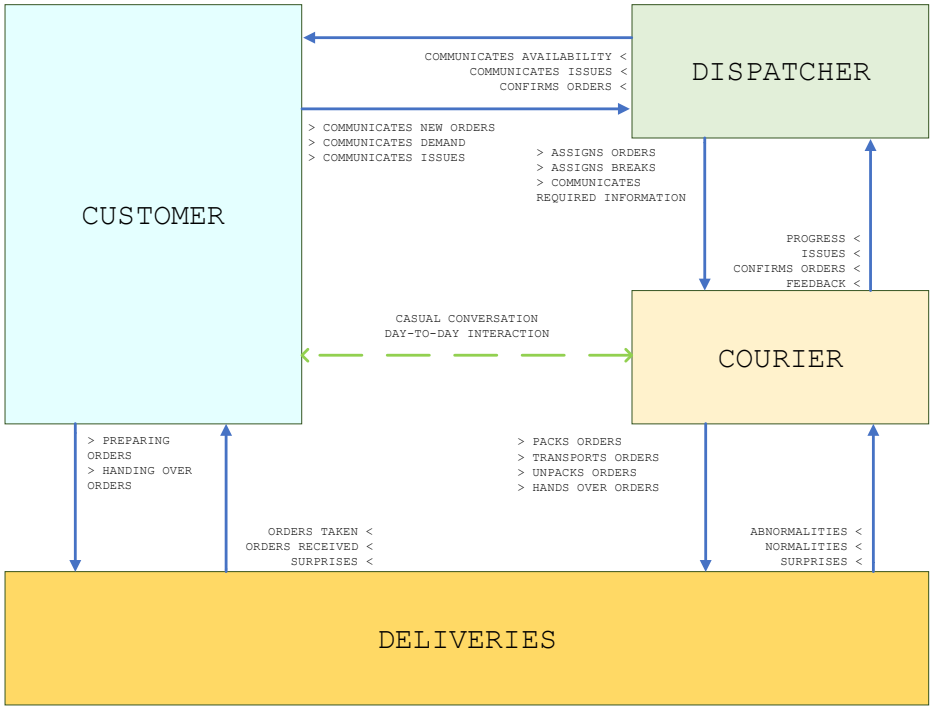

Figure 1: An example of a Control Structure of a logistics operation.

to another delivery driver. Similarly, the dispatcher may communicate an increased demand to the customer, prompting the customer to plan for potential delays in the delivery process.

2.1.2 STPA

STPA builds on the model defined in STAMP and provides a method of systematically assessing threats to the desired emergent properties of a system, usually safety. The analysis ends in a list of scenarios which outline how a system's state could compromise the desired emergent properties of the system. However, the analysis provides useful intermediate results during the analysis process as well, such as conditions in which each control action identified in the system becomes hazardous, or responsibilities for controllers and safety constraints for the system.

The STPA method consists of 4 general steps, each of which consist of smaller tasks. Each of these steps produces outcomes that can be considered results of the analysis. In this work, these outcomes are referred to as results. The most important aspects of these steps and their results are explained in further detail in this section, as well as in Figure 2.

In Step 1, the purpose of the analysis is defined. This involves determining the boundaries of the system to be analysed, identifying losses and hazards, and defining system-level safety constraints [13]. Losses are the consequences of a hazardously behaving system that are of utmost importance to avoid. Examples of Losses include,

loss of life, injury to humans, or for example, a significant financial loss. Hazards are defined as the system states which may potentially lead to a Loss. A system state concerns the entire system, rather than the state of individual component in a system. For example, in a nuclear power plant Hazards could be defined by the temperature of the reactor core. The reactor core exceeding its maximum allowed temperature is one example of such a Hazard. System-level safety constraints are conditions which forbid the Hazardous system states: "The reactor core must not exceed its maximum allowed temperature".

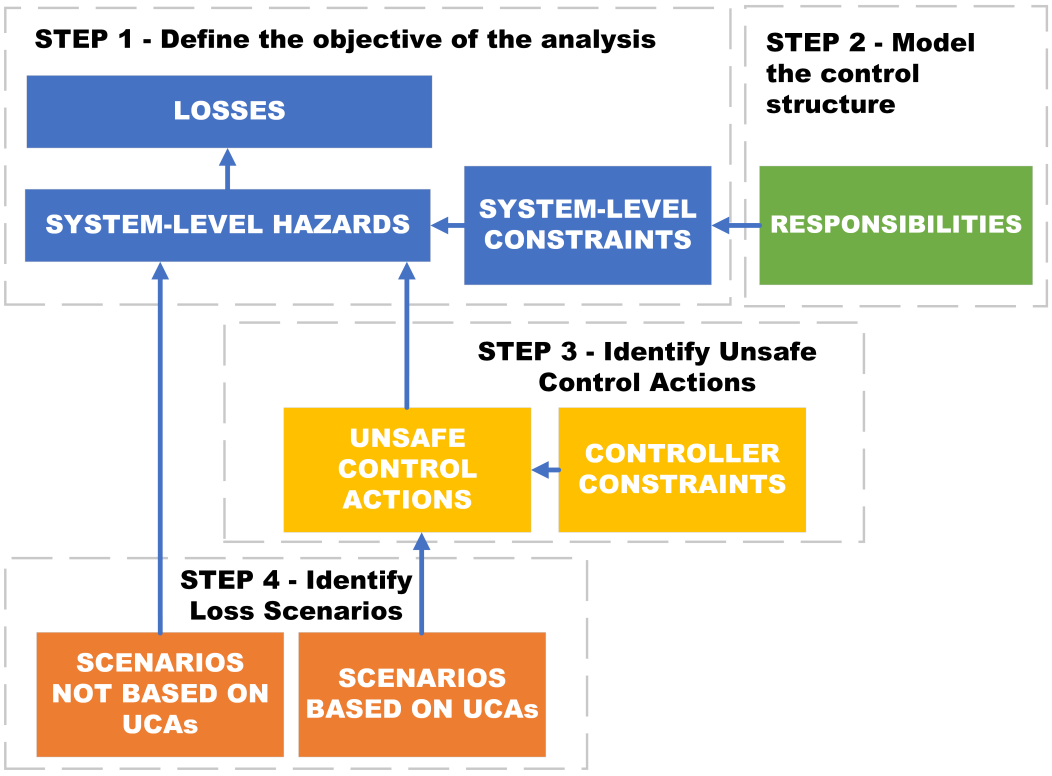

Figure 2: The traceability of various STPA results and the steps in which they are generated. In addition to traceability demonstrated here, STPA uses traceability between these results and the Control Structure. From [14].

In Step 2, the Control Structure is modeled. This is a key part of the analysis, which defines much of the workload and comprehensiveness of the analysis. The Control Structure identifies the controllers, controlled processes and the communication between them, and can be expanded to include actuators and sensors depending on the level of detail required in the analysis. The Control Structure presented in Figure 1 demonstrates a system defined with a specific boundary and level of abstraction, however it could be further expanded or decreased in both of these aspects. For example, the boundary of the system to be analyzed could be scaled down to omit the customer: requests for deliveries just become known by the dispatcher and the Control Structure does not clarify how. Similarly, the system boundaries could be expanded to include the customer's customers, as they are likely the recipients of

the deliveries. The level of abstraction could be increased by, for example, reducing the number of Control Actions (CA) and feedback. Instead of outlining all the types of communication between the dispatcher and courier, they could simply be defined as "requests" and "updates". Decreasing the level of abstraction could be done by similarly increasing the number of communication types, or by defining more entities in the Control Structure. For example, what types of devices are used to communicate between customer, dispatcher, and courier?

Step 3 involves identifying Unsafe Control Actions (UCAs). These are Control Actions in specific contexts which may lead to a Hazard. UCAs are required to document the following information: Source, Type, Control Action, Context, and link to Hazard. In addition, an STPA-ID is determined for each UCA, in order to ensure traceability. The term "Type" refers to the guidewords provided for UCA: "provided", "not provided", "provided at the wrong time", and "provided for an incorrect duration". These guidewords are applied to a specific Control Action combined with the context in which this type of Control Action becomes hazardous. This context may, for example, be an operational context such as the different operational modes the power plant may be in, or an environmental condition, such as an heavy rainfall or an earthquake. Examples of UCAs are presented in Figure 2 in Section 2.2.3.

Step 4 of the STPA analysis uses the UCAs and Control Actions as a basis for determining Loss Scenarios. Loss Scenarios document why a certain UCA would take place, or why a CA would be improperly executed. These reasons are referred to as Causal Factors. Some examples of Causal Factors are inadequate or missing feedback, inaccurate or flawed process models, and flawed control algorithms [3]. In cases when a controller is human, control algorithms may be referred to as "decision making", and process models as "mental models". Extensions to STPA which further expand these concepts have been proposed to better address the human component in STPA analyses [15].

In addition to the main outputs of each step described here, steps may also produce outputs that can be used for system development, for example controller constraints or refined hazards. These are not discussed in detail here, due to not being generated in the reactor feedwater case study. Some guides on STPA also take an alternative approach to creating controller constraints, as they are not pertinent to other results produced in the analysis [13].

Many of the steps described here are often repeated or revisited during the analysis process. For example, in Step 1 the boundaries of the system being analyzed are determined. However, upon modeling the Control Structure it may become apparent that a different boundary would be more suitable to the analysis of the system. This may require revisiting Step 1 and redefining the system boundaries. A similar iterative workflow often applies during the creation of the Control Structure itself. The Control Structure can be generated by starting at a more general level, and it may be expanded and detail may be added as the analysis progresses. Further, subsequent analyses may be conducted of individual subsystems of the system represented in the Control Structure.

2.2 Safety Critical systems

A safety critical system, is defined as a system which upon failing, could result in serious consequences such as the loss of life or damages to property of the environment [16]. In addition to the systems in the primary focus of this thesis, nuclear power plants, safety critical systems include most forms of aviation, weapons, and medical devices such as pacemakers or insulin pumps. These systems are what are traditionally considered safety critical systems, however, as our society has shifted to increasingly rely on digital information systems, an increasing number of systems can be considered safety critical. For example, in a recent data breach case in Finland the private information of approximately 30 000 therapy patients was stolen by a perpetrator, who then proceeded to use the data for blackmailing purposes. This incident reportedly led to a loss of life in multiple cases [17]. Such cases emphasize the safety-critical nature of systems traditionally not considered as such, in this case the information system utilized by the therapy provider.

Though a large variety of systems can be categorized as safety critical systems, they may vary significantly by design and implementation. For example, how these systems implement redundancy or approach resilience may be entirely different. For the purpose of this thesis, a more detailed view of nuclear power plants as safety critical systems is presented in the following section.

2.2.1 Nuclear Power Plants

Nuclear Power Plants (NPPs) produce a significant share of the electricity in Finland, approximately 42% of the electricity production in Finland in 2023 was produced in nuclear reactors. This share in electricity production has been steadily increasing in Finland over the past 20 years. Part of this increase can likely be attributed to the newly operational Olkiluoto-3 reactor. [18]

NPPs are commonly categorized by the type of reactor they employ for power generation. The most common reactor type used in NPPs globally are of the Pressurized Water Reactor (PWR) type, followed by the Boiling Water Reactor (BWR) type and Pressurized Heavy Water (PHWR) type reactors [19]. While globally the PWR type reactors vastly outnumber the BWR type reactors at 306 PWR reactors compared to 41 BWR reactors, the distribution is significantly more balanced in Finland, with 3 operational PWR reactors and 2 operational BWR reactors [18]. Hence, this section will briefly present both reactor types.

The Pressurized Water Reactor consists of a reactor core with two cooling circuits. The primary cooling circuit includes the reactor core, and operates using pressurized water. The heat from the primary cooling circuit is transferred to the secondary cooling circuit using a heat exchanger. In the secondary cooling circuit, water is allowed to evaporate into steam, which is used to drive the steam turbines. The power generation of the reactor happens in these turbines. [20]

A Boiling Water Reactor consists of just one cooling circuit, the primary cooling circuit. This circuit transfers water to the reactor core, which heats the water into steam. This steam is used to drive steam turbines. The operating principles of both reactor

types are therefore very similar; both reactor types produce heat in the reactor core, which is eventually turned into steam that is used to drive the electricity producing turbines. [20]

While the differences of the reactor types may seem trivial on-paper, the type of reactor defines many of the constraints on system design. For example, in BWRs the cooling circuit always carries traces of radionuclides, requiring the turbine to be shielded and NPP maintenance personnel to wear protective equipment while working near it [20]. Similarly, the pressures in the cooling circuits differ between the reactor types: the cooling circuit of a BWR type reactor is approximately 75 times atmospheric pressure, while the primary cooling circuit in a PWR is approximately 150 times atmospheric pressure [20].

2.2.2 Governing bodies

The nuclear domain is strictly regulated by both international and national governing bodies. On the international level, the International Atomic Energy Agency (IAEA) publishes Safety Standards, which provide "the fundamental principles, requirements and recommendations for nuclear safety" [21]. On a national level, The Radiation and Nuclear Safety Authority (STUK) defines guidelines for the nuclear power plant licensees to follow. Most pertinently to this thesis, the guidelines define the hazard analysis methods required to demonstrate the safety of systems in nuclear power plants. Currently, STPA is not required by STUK.

2.2.3 Reactor feedwater case study

The Reactor Feedwater Case Study [11] is an STPA analysis conducted in a prior Master's thesis of a reactor feedwater system for a BWR nuclear reactor in Finland, specifically the OL1 and OL2 reactors. In addition to providing valuable insights into the application of STPA on an I&C system in the nuclear domain, the case study provides the sample data used in this thesis to evaluate STPA software tools. This sample data includes, the Control Structure, Losses, Hazards, System Constraints, Control Actions, Unsafe Control Actions and Loss Scenarios produced in the analysis. The sample data is to be accurately replicated in each of the evaluated software tools, ensuring that they are capable of managing and representing the data.

The reactor feedwater system forms a part of the larger cooling circuit of the OL1 and OL2 BWR reactors. In Figure 3 the feedwater system can be identified as the systems labeled "445" and "445P". Though Figure 3 presents one model of the reactor cooling circuit, it omits the perspective of the controllers and control hierarchy used to control each of the valves and pumps in the reactor feedwater system. Such a model is presented in Figure 1, which presents a Hierarchical Control Structure similar to that of the case study.

The reactor feedwater case study began by determining the system boundaries and the objectives of the analysis. After discussions with stakeholders, the boundaries for the system were agreed upon and analysing system safety was determined as the

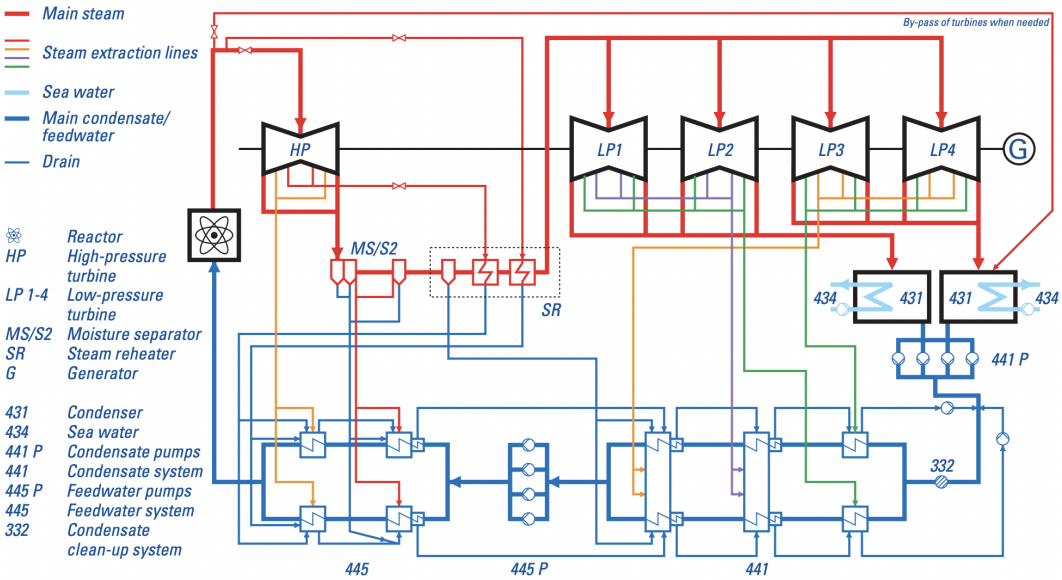

Figure 3: The reactor cooling circuit. The reactor feedwater system is a part of the larger cooling circuit. [22]

objective of the analysis. This was followed by identifying the Losses, Hazards, and System Constraints of the STPA analysis. These are presented in Table 1. [11]

After completing Step 1 of the STPA analysis, the case study proceeded to Step 2: Modeling the Control Structure. Modeling started from simple models which identified the main components and their relations to each other. Eventually after repeated refining, a Control Structure such that all stakeholders could agree on its sufficient accuracy was defined. A version of this Control Structure replicated in an STPA software tool is presented in Figure 4. A notable difference to the original Control Structure from the case study is a lack of color coding. [11]

The modeling of the Control Structure was followed by the identification of UCAs Step 3. These were generated for each of the Control Actions identified in the Control Structure. All together, a total of 146 UCAs were identified for 18 Control Actions. While the sheer number of the UCAs makes their presentation in this thesis unfavorable, a list of UCAs for one Control Action is presented in Table 2 in order to demonstrate the type of data these UCAs represent.

Following the identification of UCAs, Loss Scenarios were generated. The Loss Scenarios were identified for UCAs, CAs, and individual components. The total number of Loss Scenarios generated in the case was approximately 400, with some Loss Scenarios related to multiple UCAs. An example of a Loss Scenario from the case study is presented in Table 3.

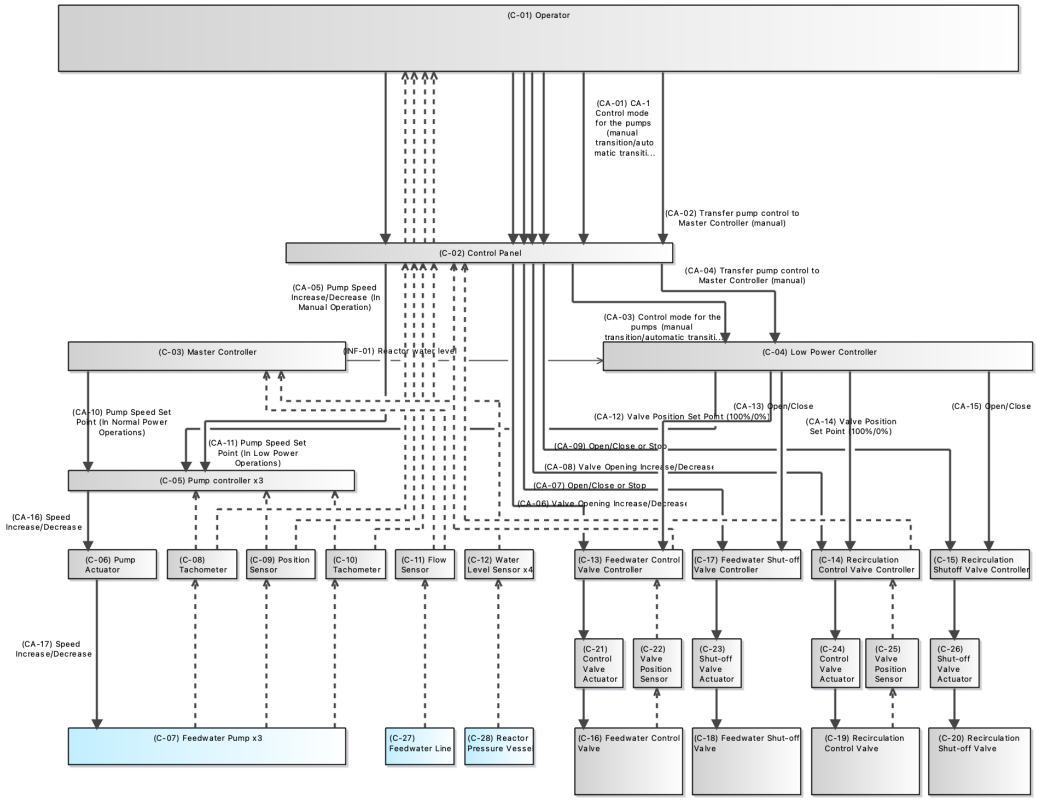

Figure 4: This figure represents the Hierarchical Control Structure of the reactor feedwater system, developed as a part of the case study. Due to being replicated in a software tool, this version of the Control Structure has slight differences to the Control Structure originally presented in [11].

Table 1: A table presenting the results of Step 1 in the reactor feedwater case study. Adapted from [11].

Result	Description
Loss L-1:	Injuries to humans
Loss L-2:	Exposure to radiation
Loss L-3	Damages to plant equipment and components
Hazard H-1	Reactor Water level falls below the minimum required level [L-1] [L-2] [L-3]
Hazard H-2	Reactor Water level exceeding the maximum allowed level [L-1] [L-2] [L-3]
Constraint SC-1	Reactor vessel Water level should be maintained above the minimum required level
Constraint SC-2	Reactor water level should be maintained below the maximum allowed level

Table 2: A table presenting UCAs derived from CA-3. Adapted from [11].

UCA Guideword	UCA-ID, description, and linked Hazards
Provided:	UCA-3-1 : The master controller provides the pump speed set point to the Pump controller during the Low power operation. [H-1, H-2]
Provided:	UCA-3-2: The Master controller provides an incorrect pump speed set point to the Pump controller during Normal operation. [H-1, H-2]
Provided:	UCA-3-3: The Master controller provides an incorrect pump speed set point to the Pump controller during a Scram event. [H-2]
Not provided:	UCA-3-4: Master controller does not provide the pump speed set point to the Pump controller point during Normal operation. [H-1, H-2]
Not provided:	UCA-3-5: Master controller does not pro- vide the pump speed set point to the Pump controller point during a Scram event. [H-2]
Provided at wrong time:	UCA-3-6: The Master controller provides the pump speed set point to the Pump controller too late after a Scram event is initialized. [H-2]
Provided for an incorrect duration	N/A

Table 3: A table presenting a Loss Scenario based on UCAs 6-1, 6-3 and 6-9 from the case study data. Adapted from [11].

ID	Loss Scenario
Scenario-94:	The physical pump controller malfunctions and provides the signal to Increase the speed continuously [UCA-6-1, UCA-6-3, UCA-6-9], causing the pumps to go over the speed set point. As a result the water level in the reactor may be too high [H-2].

2.3 Requirements engineering

Requirements engineering is a form of engineering concerned with determining and managing criteria for a system being developed [23]. While this thesis is not directly concerned with determining requirements for a system being developed, determining requirements is an integral part of the work required to evaluate existing software tools. In this context, requirements engineering provides many guidelines for accurately and effectively translating the needs of stakeholders into requirements. Therefore, this section will present a brief overview of requirements engineering with a focus on aspects especially relevant to the work conducted in this thesis.

2.3.1 Requirements elicitation

Requirements elicitation refers to the techniques used to elicitate the needs of stakeholders with regard to a system being developed. In the context of this work stakeholders could refer to, for example, the users of a software tool, the company employing these users, and the employees utilizing the results of the analysis. Stakeholders, and people in general, often have difficulties communicating exactly what they mean, and this issue is amplified when the communication is the basis for developing large, complex systems. The method by which this communication is facilitated, requirements elicitation, plays a critical role in determining accurate requirements that describe a system matching the needs of stakeholders. [24]

Some of the techniques used to elicitate requirements are different kinds of interviews. These may have a stricter format such as in the case of questionnaire interviews, or have a more relaxed structure such as in open-ended interviews or focus groups. Each of these methods have to be applied with an understanding and consideration of their respective strengths and limitations [24].

In a questionnaire interview, each interviewee is asked the same questions with the same wording. The questions are formatted such that they have a predefined set of possible answers. The method has the benefit of easily quantifiable results. For example, statistical methods can be utilized with the results, revealing responses to the interviewer's questions which correlate with each other. The predefined, strict format is, however, limited by the preconceptions and biases of the person deciding the questions to be asked [24]; an interviewer may not realize a certain factor to be important when creating and asking the questions, leaving these aspects undiscovered by the questionnaire interview. Similarly, a question in the interview may be created with little consideration towards the differing perspectives of the interviewees, such that they assume the interviewee has a similar background to the interviewer or the majority of individuals in a certain environment.

Open-ended interviews enable the interviewee to answer the questions more freely, in their own words, rather than having to select from a predefined selection of answers. Interviewees are free to express their thoughts with regard to the interview questions more expansively, and therefore, interviewees are more likely to communicate thoughts they consider important to the subject matter. By having a less strict approach, open-ended interviews address many of the limitations of the stricter

questionnaire interviews, however, the data gathered through an open-ended interview is less quantifiable compared to the data produced by questionnaire interviews [24]. The questions presented to the interviewees are also still subject to the biases and preconceptions of the person creating the questions.

Focus groups are an interview technique in which a group of people are presented with a topic, which the group is encouraged to discuss [24]. This discussion may be aided by different forms of stimulus such as images or videos. The method is suitable for gathering the thoughts of stakeholders who are knowledgeable on the subject matter of the interview. This form of requirements elicitation is employed in this thesis for revising the requirements due to the availability of prospective users and STPA practitioners. The method is also suitable for situations in which the group can be presented with some form of stimulus to encourage a discussion, in the case of this thesis, a preliminary set of requirements.

In addition to the different kinds of interviews, requirements may be elicited using brainstorming, mindmapping, workshops, user stories, prototyping, and many other techniques [25]. These techniques each have their distinct benefits and limitations, for example, prototyping is often suitable for situations when the users are unable to express their needs [25].

2.3.2 Types of requirements

A common division is made in requirements engineering regarding the categories of requirements. Namely, two categories of requirements concerning the system being developed are prominent in requirements engineering literature: functional and non-functional requirements. This division in requirements is also made in this thesis.

The general definition of functional requirements has largely been agreed upon in requirements engineering: they are requirements that define the functions that a system is required to perform [26]. For example, a calculator could be required to be able to calculate the root of numbers. Similarly, a hazard analysis software tool may be required to present an overview of the analysis status to the user.

Non-functional requirements are those not concerned with the specific functions a system must be able to perform. However, what aspects of a system this exactly entails, or an exact definition have not been agreed upon [26]. One perspective is to view software tools as systems with emergent properties, these properties being those that non-functional requirements are concerned with. For example, a software tool may be, reliable, user-friendly, or efficient at a specific task.

In this thesis, requirements are largely functional requirements, though they are constructed with a consideration towards broader goals for the software tool, such as convenience to the user. This is due to the approaches required for evaluating the non-functional aspects of a software tool in comparison with the approaches required to evaluate functional aspects. In the latter, given that a set of functional requirements are agreed upon with the stakeholders, a single evaluator can handle the evaluation process. However, in the former, a suitable approach to the evaluation process would involve multiple users testing each software tool, and the result of the evaluation process as a whole would take into account each reviewer's subjective assessment of

the software tools with regard to the non-functional requirements.

2.4 Issues identified in existing literature

An important part of defining requirements is taking into account the needs and concerns of potential users of the software tool being evaluated or developed. In the research domain, many papers discuss or mention issues with either software tools that have been tested, or the STPA method itself. While the topic of suitable software tools, or the lack thereof, is often briefly discussed in conjuction to another, primary focus of a scientific publication, these mentions of STPA related issues experienced in the research domain provide some insight to aspects, which software tools could address, and areas in which current software tools may be lacking. In this thesis, these issues raised in the research domain will form the basis of a preliminary set of requirements, which are designed to address these concerns. While requirements generated in this way will likely vary significantly by author, these preliminary requirements serve only as a starting point for collaborative work with STPA practitioners in industry and STPA experts from the research domain.

A prior evaluation of existing software tools presented in [10], though limited in its depth, highlighted many issues currently available software tools may have. Many of the tools covered in the evaluation were not publicly available, were text-heavy, or provided little apparent value over using rudimentary approaches such as Microsoft Excel and Visio. In a Master's thesis published in 2023 [11], a brief overview of available STPA tools notes that due to not supporting the newer STPA revision, rudimentary approaches were favoured for the case study conducted in the thesis instead of the STPA specific software tools. The thesis also notes that some of the software tools are suitable for experienced STPA users, but that a user new to the method would rather benefit from rudimentary approaches. The lack of customizability of the existing software tools was also mentioned [11].

Concerns expressed in the research literature over the STPA method itself were often those concerning the difficulty in identifying key results produced by the analysis. For example, a comparison between HAZOP and STPA notes that the text-heavy nature of the results produced by an STPA analysis make it difficult to acquire a quick overview of the results [27]. In an STPA Guide published by VTT [13], prioritization is suggested as a form of guiding the analysis and design decisions made in response to the analysis results. Prioritization is also identified as a potential avenue for improving STPA in [11].

The text-heavy nature of the analysis has also been identified as a factor that inhibits conducting the analysis, especialy group work [27]. Though based on an earlier version of STPA, a techincal report also highlights that the analysis produces large tables of results, which may be hard to work with [7]. However, group work is essential to the method, as each step of the analysis requires the STPA facilitator to work with system experts [3]. Discussions with STPA experts at the outset of this work highlighted similar issues regarding the group work aspects of the method. Most notably, the difficulties arise from having to use multiple software tools to handle both the text-heavy tables of results as well as diagrams such as the Control Structure in

order to communicate the analysis to system experts, who may not be familiar with STPA. For example, a system expert may be able to advise on the function of a pump controller, but have little prior knowledge on STPA's Control Structures. In such a situation, the STPA facilitator may have to be able to switch back and forth between the relevant parts of the Control Structure and other analysis results or documents pertinent to the analysis.

One aspect of the STPA method that is often mentioned in the research literature explored for this work, is the steep learning curve of the method [7] [27]. Employing the method requires understanding the underlying accident model STAMP as well as familiarity with the analysis process. Especially those familiar or experienced with reliability based hazard analysis methods such as FMEA may find it hard to employ the method [7]. In these cases, it may be hard to distinguish a system state from the causes of the state [3]. For example, a nuclear power plant may be in a state where the reactor water level is too low. This could be caused by the failure of a feedwater pump. In STPA the former is documented as a Hazard, while the latter is documented as a Loss Scenario. For someone inexperienced with STPA, it may be easy to confuse the two and hence incorrectly document results in the analysis. The steep learning curve of the method could be addressed by a software tool, which could provide guidance on employing the method, for example through providing examples or guide words for documenting analysis results.

A literature review exploring the literature regarding the STAMP model and STAMP based methods suggests STAMP could be enhanced by the addition of a quantitative analysis method to the model [4]. In practice, this could be implemented through, for example, an extension to the STPA method or integration of STPA with existing quantitative analysis methods. The combined use of STPA with other hazard analysis methods has been proposed in multiple papers, including a paper specifically investigating the use of STPA in the nuclear I&C system context [9]. Such integration has been proposed in the form of the Risk Analysis And Assessment Modeling Language (RAAML), an extension to the Systems Modeling Language (SysML). RAAML defines how different analysis results can be utilized as a part of the same systems engineering effort, allowing STPA to be used in conjunction with more conventional hazard analysis methods, such as FMEA or FTA [28].

A scoping review from published in 2022 highlights the flexible interpretations STAMP and STPA have, especially in their high number of proposed modifications and extensions in the reviewed journal articles and conference papers [29]. In total, approximately half of the reviewed publications suggested modifications to STAMP, STPA or CAST. These findings suggest a software tool should be able to support analysis work with flexible interpretations of the methods.

The need for integrating STPA effectively into all phases of the system engineering effort is also identified in the STPA Handbook [3]. By integrating STPA into the system engineering life cycle, and especially by utilizing the method in earlier phases of the life cycle, the cost of safety is reduced [3]. Essentially in this approach, safety is addressed through system design rather than "bolt-on" safety fixes. The need for integration is also identified in other research literature, for example in a research paper concerned with the developmpent of the STPA software tool SAHRA, it is proposed

that software tools should be integrated into the toolchain to reduce human error [30].

In a conference paper presenting the development of WebSTAMP, an STPA software tool, requirements for a software tool supporting STPA and STPA Safe-Sec were also outlined [31]. Unfortunately, the tool was developed for the earlier version of STPA, resulting in many of the requirements presented in the paper to decrease in significance to the evaluation conducted in this thesis. Another unfortunate aspect of the requirements presented in the paper is the lack of an in-depth requirements elicitation process used to generate the requirements. Though a mention of such processes is made in the paper, ultimately, it states that the essential requirements presented in the paper are based on the authors' experiences of STPA analysis. The requirements presented in the paper do however align to a certain extent with issues already discussed in this section. For example, traceability between all elements of the analysis is identified as important, as well as the clearer approach to generating each result, i.e. systematizing the generation of results [31]. The paper also suggests the automating certain aspects of the analysis, and reusing parts of previous analyses, likely to lessen repetitive work for the user.

Preliminary discussions with industry practitioners also highlighted the need for a STPA specific software tool. Currently, the use of STPA by the practitioners is supported by an Application Lifecycle Management tool. While this tool provides many benefits to simply conducting an analysis in Excel, such as the capability to generate reports and an overview of the analysis, it can still be time consuming to use due to several omissions in the software tool, most notably the lack of built-in Control Structure features.

The findings and proposals discovered in the review of existing research literature and discussed here are summarized in Table 4. The table highlights the issues discussed in the existing research literature, which could have implications on STPA software tool requirements.

Table 4: A table demonstrating findings and proposals related to the requirements for an STPA software tool and their sources. Adapted from [14]

Requirement related findings and proposals	**Source**
Traceability: Robust traceability is needed to ease the documentation of UCAs and Loss scenarios	[8]
Traceability, Prioritization: It is hard to get a quick overview of the most important hazards	[27]
Traceability, Prioritization: Filtering for example by priority could be beneficial	[11]
Prioritization: Prioritization benefits using STPA results to formulate system safety constraints, safety goals and to influence system design	[13]
Prioritization: Prioritization could help address and use the results of the analysis	[11]
Customizability: Current software tools suffer from limited customizability	[29][11]
Integration: Software tools should be integrated into the toolchain to reduce human error	[30]
Integration: The optimum benefit from STPA is generated through a combination with other methods.	[9]
Guidance: Some software tools inhibit learning STPA rather than support it.	[11]
Guidance: The STPA method has a steep learning curve, and may be difficult to learn for those familiar with conventional hazard analysis methods.	[27][7]
Group work: The text-heavy nature of the analysis hinders group work.	[27]
Group work: The large tables produced by the analysis can be hard to work with	[7]

3 Software tool requirements

Evaluating software tools requires the evaluator to know the characteristics of the tool they are looking for. The tools can be required to satisfy specific goals, such as convenience to the user, or to provide certain features, such as visualizing a specific type of information. The former type of requirements is often referred to as a non-functional requirement, while the latter type is referred to as a functional requirement. In this thesis, requirements define the features required of the software tools rather than the goals a software tool should satisfy. This is due to the features of a software tool being simpler to validate compared to software tool goals, such as convenience to the user. The latter would preferably include user research taking into account the opinions of at least a few users.

While the non-functional requirements will not be formulated or evaluated against in this work, they will be discussed on a general level, as the functional requirements are defined such that they work to ensure many of the goals are met. For example, a requirement to be able to import a certain file type may work towards a larger goal of convenience to the user, or compatibility with existing systems. This work will hence document the goals and basis upon which the requirement was developed.

This chapter presents in detail the requirements used to evaluate the software tools and a detailed overview of the process which these requirements are a result of.

3.1 Goals for the requirements

On a very general level, the goal of the software tool requirements is to define such a software tool, which would address the concerns presented in the research literature as well as the issues industry practitioners have faced with their current approaches. The former was condensed into the goals presented in Table 5, while the concerns of industry practitioners are used to inform the process of revising the preliminary set of requirements generated in line with these goals.

Table 5: A table demonstrating the goals for the software tool requirements.

Goal number	Description
G1	Ensuring the convenient documentation of results.
G2	Ensuring robust traceability.
G3	Enabling the effective utilization of analysis results.
G4	Enabling effective cooperation with system experts during the analysis.
G5	Enabling a more gentle learning curve for STPA.

3.2 Methods

One source for generating software tool requirements is the existing research literature on the STPA method. As demonstrated in Chapter 2, the findings and discussions presented in existing research literature emphasize the many aspects of conducting an STPA analysis that can make the method more labor-intensive, time-consuming, or harder to learn. Some of the research literature provides direct suggestions, such as implementing STPA result prioritization to manage the expansive and time-consuming nature of the analysis [8], while other research literature may simply highlight an issue, such as the lack of customizability in existing software tools [11]. Some papers have determined requirements for an STPA software tool [31][10], however they may not be the results of a thorough requirements engineering process, rather they may reflect individual preferences and experiences with STPA.

Ideally, in order to determine what kind of requirements the software tools should be evaluated against, the potential users of the software tools should be consulted. They may provide important insights into, for example, which tasks could benefit the most from the aid of a software tool, what kind of experiences they have had with current software tools, or how they would like the software tool to be implemented in their work. In the context of this work, STPA practitioners, especially those in the nuclear industry are of special interest, as the software tool is evaluated in the context of performing STPA analyses for nuclear I&C systems.

In this work, the software tool requirements are formulated in a way that takes into account both the prospective users of the software tools and the existing research literature. First, an initial draft of the requirements will be made based on the existing research literature. Next, these requirements will be discussed with industry practitioners and feedback will be collected in focus group sessions. In this work, focus groups consist of a total of 7 individuals. The feedback gathered in these sessions is then utilized to revise the requirements. Finally, the revised requirements will be reviewed together with an STPA expert to ensure they are suitable to be used in the evaluation of STPA software tools.

3.3 Preliminary requirements

Based on the issues identified in existing research literature, a preliminary set of requirements for an STPA software tool was drafted. This set of requirements was categorized by the aspects of the STPA method and software tool they concerned: Traceability, Prioritization, Integration, Guidance, and Customizability. In addition, a few uncategorized requirements are presented in the "Other" category. This preliminary set of requirements is presented in Tables 6-11. The requirements are documented together with numbers 1, 2, and 3 which represent the use cases they would benefit the most. Use case number 1 was defined as an STPA expert working alone on the analysis, while use case number 2 was defined as cooperating with system experts as a part of the analysis. This would include situations such as discussing parts of the analysis with people familiar with the system, but not necessarily familiar with STPA. Finally, use case number 3 was defined as utilizing the analysis after its completion.

This use case is concerned with the effective utilization of the results produced by the analysis. While use cases are not limited to this set of 3 use cases, these use cases together are expected to cover most practical situations in which the STPA software tool would be utilized. One use case that was not explored, is the collaboration of two or more STPA facilitators. This type of work on the STPA analysis appears to be largely unaddressed in the research literature, however, as the scopes of analyses expand and the workloads increase, this use case may need to be further investigated.

During the drafting of the preliminary STPA software tool requirements, one STPA expert from VTT was consulted on several occasions, and requirements were adjusted accordingly. During these meetings, thoughts on what kind of information is required to undertake each step of the STPA analysis were discussed, and the division of requirements based on the use cases they were estimated to impact the most was devised. Another topic that was discussed in detail during the meetings was how each use case would apply to each step and the gathering of information for each step. For example, generating the Control Structure is likely often undertaken together with system experts, as their input is critical to the generation of a Control Structure that can adequately model the system in question. These discussions had a significant impact on the what kinds of preliminary requirements were generated, and their details such as how they were worded.

Table 6: Requirements determining the features concerning the traceability of the STPA analysis. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	Requirement	Use cases
Traceability	The software tool can visualize traceability from loss scenario to UCA to hazard to loss (for each result its entire traceability).	1,2,3
Traceability	The software tool can conveniently visualize traceability from the Control Structure to results and vice versa.	1,2
Traceability	The software tool supports traceability between the Control Structure and results in general.	1,2
Traceability	The software tool supports filtering items based on traceability such as showing items that trace back to a specific Control Structure entity or showing all scenarios related to a specific control action or hazard.	1,3
Traceability	The software tool supports traceability from one STPA analysis to another, for example when conducting analyses at different levels of abstraction.	1,3

While traceability is one of the strengths of STPA [3], much of the research literature emphasized a need for more robust approaches [8],[27]. Though the built-in

traceability documents the relations of each result with regard to each other, forming a mental model of the analysis could likely be improved by representing these relations in different ways to the user. Based on such a perspective, requirements for the software tool to be able to visualize traceability were generated. In the context of this work, the term "visualize" is used to refer to the software representing data in visual formats more elaborate than rows on a spreadsheet. The Control Structure is commonly identified as a key part of the analysis [29][3], and hence the traceability of results with regard to their relations to the Control Structure was also emphasized in the requirements. Finally, in order to ensure that the software tool supports an iterative analysis approach, traceability between analyses is required to be supported.

Table 7: Requirements determining the features concerning the prioritization in the STPA analysis. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	Requirement	Use cases
Prioritization	The software tool supports prioritizing Loss Scenarios: a) Associating RPN numbers to results. b) The possibility of associating results of other hazard analysis methods with STPA results.	1,3
Prioritization	The software tool supports filtering/sorting STPA results by priority.	1,3
Prioritization	The software tool can visualize STPA results in order of priority.	1,2,3
Prioritization	The software tool supports systematically documenting causal factors: a) Associating causal factors to loss scenarios. b) Filtering/sorting/grouping loss scenarios by their causal factors.	1,2,3

Many sources suggest prioritization as a solution to handling the time-consuming and expansive nature of the analysis [8],[32]. By prioritizing analysis results during the analysis, the analysis can be focused on the more critical parts of the system and results could also become easier to utilize. Some of the approaches to prioritization include assigning a Risk Priority Number (RPN) to the results, namely UCAs and Loss Scenarios [13]. In one proposal assigning RPNs to results involves consulting system experts on their estimate of how severe consequences a UCA has, likelihood of a loss scenario, strength of knowledge on the UCA and loss scenario, and available time to respond. These values are multiplied to return a value for each loss scenario, a higher value indicating a higher priority. [8]

The prioritization of results was seen as something that a software tool would benefit from supporting by the author. Successfully implementing a prioritization feature into a software tool could help especially in a systems engineering context, where STPA results may need to be used to guide decisions concerning system design.

Prioritized results in this context could inform the user of the most important aspects of the system address.

Prior to drafting the preliminary requirements, a quick overview of some available software tools was conducted. One of the tools, STPA Viewpoint for Capella, demonstrated an interesting approach to loss scenarios, in which the user was first required to generate causal factors for each UCA. These causal factors are, in essence, the reasons why a UCA may occur. Examples of these include, inadequate control algorithms, inaccurate process models, or incorrect feedback. Causal factors were identified by the author as something that could be used to prioritize results. For example, if it were convenient to identify that many loss scenarios stem from the same causal factor, addressing the causal factor could be prioritized. Hence, a requirement concerning causal factors was added to the prioritization category.

Table 8: Requirements determining the features specifically concerned with easing the STPA analysis for the user. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	**Requirement**	**Use cases**
Guidance	The software tool uses language consistent with the STPA method: i.e. Hazards are labeled as hazards, loss scenarios as loss scenarios, etc.	1,2
Guidance	The software tool provides guidewords for UCAs.	1,2
Guidance	The software tool provides guiding questions for Loss Scenarios.	1,2
Guidance	The software tool guides the user to document results in the correct syntax.	1,3
Guidance	The software tool enables conveniently documenting and viewing material related to each result: standards, operating manuals, layouts and process plans, etc.	1,2,3

The guidance category of requirements was created to include requirements that guide the user toward conducting the analysis efficiently and correctly. Guidewords and guiding questions were quickly noticed as something that could benefit users, and hence they were quickly transformed into requirements. The requirement for consistent language was created after initial tests of software tools revealed that some tools had a much steeper learning curve due to the naming of the results conflicting with how they are defined in the STPA Handbook.

Conducting an STPA analysis involves handling and consulting documentation that may differ significantly from one analysis to another. For example, process plans, system models, standards, and many other types of documentation can be used to inform the analysis. Considering that much of the analysis may be based on information from these kinds of documentation, handling and linking the analysis directly to the documentation within the software tool was seen as potentially very

important and worth discussing with industry practitioners. Hence, a requirement concerning handling material required to conduct the analysis was added.

Table 9: Requirements determining the features concerned with integrating the STPA analysis to the larger system engineering effort. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	Requirement	Use cases
Integration	The software tool can export analysis results in formats accepted by other system/requirements engineering tools.	3
Integration	The software tool supports Risk Analysis and Assessment Modeling Language (RAAML).	1,2,3
Integration	The software tool has the ability to trace decisions in system design and implementation back to analysis results, for example, a certain component could be traced back to specific control actions, loss scenarios, or a countermeasure.	3
Integration	The software tool can visualize the traceability from system design and implementation to the analysis results.	3
Integration	The software tool can highlight changes made to the design and implementations that require re-analysis.	1,3

Integration of the software tool to the larger systems engineering effort was generally considered through multiple levels integration. The highest level is that of total integration; the STPA analysis is integrated into a systems engineering tool, and activities in the systems engineering tool can be directly traced back to the analysis results within the tool. An intermediate level is that of compatibility between the STPA software tool and a systems engineering tool, or tool used by the nuclear power plant licensee for managing requirements, for example. The outputs of the STPA software tool need to be conveniently and efficiently utilized in other systems/requirements engineering tools. On the lower levels of integration, there is little direct interplay between the STPA software tool and other software tools, however decisions based on the analysis can still be traced within the STPA software tool. This can be done for example through the documentation and traceability of countermeasures in the STPA software tool.

One approach to integration is the Risk Analysis and Assessment Modeling Language (RAAML) [28]. It is an extension to the SysML modeling language, and aims to connect various types of risk analysis and assessment activities to each other. It enables traceability between analyses and other model-based systems engineering activities. While the novelty of RAAML may prevent it from being widely represented, or represented at all, in the tools evaluated in this thesis, the requirement to support

RAAML was included at this stage in order for the requirements to potentially better serve as the foundation for developing a software tool. Including RAAML in the preliminary requirements is also intended to generate discussion on the topic during the focus group sessions.

Requirements concerning the customizability of the STPA tool focused on supporting the different approaches users may have to conducting an STPA analysis. To this end, preliminary requirements for the tool require it to allow flexible interpretations of the STPA method, i.e. the user not being forced to follow the syntax word-to-word. Similarly, alternative approaches to STPA were considered beneficial to support, for example STPA - Engineering for humans and STPA-SafeSec. The color coding of the Control Structure was also identified as a customizability feature that could improve the legibility and readability of the Control Structure.

Table 10: Requirements determining the features concerned with adapting the software tool for different types of approaches to STPA. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	**Requirement**	**Use cases**
Customizability	The software tool supports flexible interpretations of STPA, i.e. ways of conducting the analysis that are not outlined in the STPA Handbook.	1,3
Customizability	The software tool supports known alternative interpretations of STPA, such as STPA-SafeSec, or STPA-Engineering for humans.	1,3
Customizability	The software tool allows color coding of the Control Structure.	1,2,3

Finally, some preliminary requirements that are not categorized are presented in Table 11. These may be categorized in further revisions of the requirements, however, in the focus group sessions they were presented in the format presented in Table 11. These requirements are concerned with individual findings or ideas the author made while conducting the background research for these requirements. One such finding was the importance of supporting the 2018 revision of STPA, as some of the tools may be based on the older versions of STPA and the reactor feedwater case study was conducted folowing the 2018 revision of STPA. In order to ensure the completeness and correctness of the analysis and its results, requirements were generated with the aim of directing the user's attention to which items of the analysis need to be addressed and how to address them correctly.

Table 11: Uncategorized requirements for the STPA software tool. Use case number 1 = an STPA expert working alone, use case number 2 = an STPA expert working together with system experts, and use case number 3 = utilizing the STPA analysis after its completion.

Category	Requirement	Use cases
Other	The software tool supports the 2018 revision of STPA.	-
Other	The software tool supports adding notes/comments.	1,2,3
Other	The software tool can warn the user of incorrectly formatted results, such as a UCA that is missing context.	1
Other	The software tool can highlight items that need to be reviewed after part of the analysis has been modified, for example, a control action is added $\rightarrow$ review controller constraints, and generate UCAs and loss scenarios.	1,3

3.4 Results of the review focus groups

During the focus group meetings industry practitioners confirmed many of the existing requirements as important, but also had many thoughts and ideas on additional aspects that need to be considered, and how the requirements may need to be revised. The overall reception to the preliminary requirements was positive. The requirements were seen as providing a good basis for developing a software tool, and one industry practitioner noted that the current tools he had tried had lacked the kind of systematic approach demonstrated in the preliminary requirements.

One aspect of the STPA analysis was highlighted by all industry practitioners who participated in the meetings: the Control Structure. The importance of this aspect of the STPA analysis was emphasized multiple times, and in contrast, the Control Structure was perhaps slightly underrepresented in the software tool requirements. Industry practitioners especially noted that the Control Structure should be easy to create, modify, and work with during the analysis, and the iterative nature of the analysis should be supported by the software tool in aspects concerning the Control Structure as well. Industry practitioners emphasized that the Control Structure may change significantly as the analysis progresses, making the ability to easily edit the Control Structure important. A good way of integrating the Control Structure into the software tools was also noted by the industry practitioners as lacking in most tools, and two industry practitioners mentioned that the ability to create or edit the Control Structure was entirely missing from the software tool they currently use for STPA.

Common themes during the discussions regarding the Control Structure and its implementation in the software tool were the ability to iteratively add detail to the Control Structure and the ability to view the Control Structure at various levels of abstraction. In the former, industry practitioners wished they could start building the

Control Structure from a broader level and add details in the future. This could be during a single analysis or over the course of many analyses. In the former, industry practitioners noted that being able to adjust the level of abstraction for viewing purposes would be very helpful.

During the meetings, a point was made regarding Control Structures by two industry practitioners with experience applying STPA to nuclear power applications. According to the industry practitioners, STPA may be applied to specific plant states at a time and then repeated with the same Control Structure to other plant states. In this context being able to copy a part of the analysis to another analysis would save a lot of time. Especially beneficial would be the ability to copy the Control Structure, but being able to copy other parts of the analysis such as losses or hazards may be beneficial as well.

Industry practitioners also hoped to remove some of the more repetitive tasks of the STPA analysis. One example was the creation of UCAs. In general, UCAs follow a strict format, which defines the control action, how it is provided, where it is sent from, where it is received, the hazards it is related to, and the context in which the control action becomes unsafe. Most of these fields could be generated automatically, leaving the context and hazards for the user to determine. These automatically generated UCAs could then be reviewed by the user to see if they are applicable to the specific Control Action.

Requirements regarding prioritization generated significant discussion in both meetings and were perhaps the most in need of revision. Much of the discussion was concerned with the correct approach towards prioritization. The preliminary requirements regarding prioritization were largely based on research on how risk priority numbers could be applied to the STPA method. While the risk priority numbers have been identified as providing some benefit to conducting the analysis and utilizing its results, industry practitioners identified the method as too time-consuming. Instead, practitioners suggested alternative forms of prioritization, such as prioritizing by the traceability of the result, or estimating a single priority value, such as “high”, “medium” and “low”. One industry practitioner considered prioritizing through traceability more “in the spirit of STPA” than assigning priority values. For example, a UCA that links to a more severe loss would be considered before a UCA that links to a less severe loss. Alternatively, prioritization by the number of hazards a UCA or loss scenario links to was proposed in the meeting.

The requirements concerning the traceability of the method were seen as very important criteria for evaluating the software tools. All industry practitioners emphasized the importance of well-functioning, robust traceability in the software tool. This is also corroborated by the research literature, with one paper stating that the lack of robust traceability built into the STPA method hinders the creation of UCAs and Loss Scenarios [27]. While traceability applies to each result of the analysis, it may be hard to work with given the expansive and complex nature of the analysis. Without a specialized software tool, one may need to switch between spreadsheets, documents, and programs to generate a single result. Switching windows and navigating spreadsheets to find the specific piece of information required to produce the result does indeed hinder the documentation of results.

The traceability requirements are further emphasized by the views industry practitioners presented on result prioritization. In their view, the traceability of the analysis provides a simpler and more intuitive way to prioritize the results, than evaluating the priority of each analysis result separately. This, however, requires the features of the software tool concerning traceability to be robust and effective. For example, for prioritizing Loss Scenarios, the software would need to provide convenient ways of showing all Loss Scenarios related to a specific loss, hazard, control action, Control Structure entity, or any other result. In general, the results produced by the analysis need to be able to be visualized in flexible ways such that the person conducting the analysis sees exactly what they need to see. For example, visualizing results filtered by their relation to another result or visualizing these relations between results in ways such as a tree diagram.

The topic of different views of the results was discussed multiple times in each meeting as well. The use cases presented at the start of each meeting provided a basis for this discussion: conducting STPA alone as an expert, cooperating on STPA together with system experts, and utilizing the results of the STPA analysis. Throughout the meetings, the requirements were reflected through these use cases. Industry practitioners agreed that the example use cases presented at the beginning of the meetings were important. For example, having the Control Structure conveniently available when cooperating with system experts was immediately identified as a significant feature. An important point about utilizing the results of the analysis was also made; the results may need to be presented to decision-makers or other stakeholders. These people may know nothing about the STPA method but need to have actionable data from the results. Industry practitioners hoped that the software tool could have a separate view for presenting the results to stakeholders.

The customizability requirements generated significant discussion in the meetings. One industry practitioner was especially concerned with the need for supporting alternative forms of STPA, such as STPA-SafeSec or STPA: Engineering for humans. The alternative methods were viewed as issues STPA itself should solve. A single STPA method that incorporated parts of these alternative methods would be ideal, and a software tool would support this single STPA method rather than supporting multiple different methods. In terms of flexibility, industry practitioners voiced concerns over its balance with structure. Having structure may remove flexibility, but provide other benefits, such as ease of use.

The guidance requirements generated less discussion than the other categories, however, there were some ideas regarding these requirements. One industry practitioner suggested the guiding measures of the STPA software tool should be good enough to guide a user unfamiliar with the method through the process of conducting an analysis. Another practitioner suggested meetings with system experts needed more guidance. However, defining such guidance is beyond the scope of this work.

Lastly, there were some ideas presented in the meeting by the practitioners that do not directly fall into any category of requirements. One point was concerned with the ability to see one's progress throughout the analysis. In the software tool, this could look something like highlighting parts of the Control Structure that have complete results with a color or symbol, while other parts could be indicated with other colors

or symbols. Another point considered the language of the software tool important; industry practitioners in the Finnish nuclear industry would likely appreciate Finnish language support. Security was also considered an important aspect of the software tool, and in practice, a software tool should be available to host by the nuclear licensee themselves, rather than a third party.

The possibility for each STPA practitioner to approach the analysis in a way that suits them should be considered, and views should not be limited to those preconceived as useful by the designers of the software. Supporting this notion, a 2022 literature review suggests STAMP and STPA are viewed more as flexible guidelines for serving individual analysis needs rather than as a strict toolset that must be used according to original specification [29]. This finding could also be observed in the different approaches to STPA demonstrated by the practitioners even in such a small sample group. An STPA expert consulted during the drafting of the preliminary requirements emphasized different aspects of the analysis, such as managing the expansive results through prioritization, while STPA practitioners consulted in the meetings placed significant emphasis on the Control Structure and traceability. Individual practitioners may have difficulties in different parts of the analysis or appreciate different features in the software. The points brought up during the meetings, however, seemed to be largely agreed upon by all consulted practitioners.

3.5 Revised requirements

After documenting the discussions of the focus group meetings during which the preliminary software tool requirements were reviewed, the preliminary requirements were each carefully inspected to assess whether and how they may need to be modified. Major changes were not needed, however, due to the largely positive feedback on the preliminary feedback. These requirements are grouped differently from the preliminary requirements in order to accommodate the ideas discussed in the focus group meetings. All the revised requirements are presented in Table 12 along with their respective categories and reasoning. Documenting the reasoning for each requirement is measure intending to allow prospective use of the requirements beyond this work. In such a context the requirements should be assessed against their respective reasoning to determine whether the requirement still provides value. In addition, it is important to present this reasoning to demonstrate the need for each requirement within the context of this work.

An entirely new category of requirements was made to cover the features concerning the Control Structure. Control Structure requirements determine what features the software should provide concerning the Control Structure of an analysis. In hindsight, a glaring omission from the preliminary requirements, the possibility to create and modify the Control Structure has now been added as a requirement. In addition, requirements concerning modifying and viewing the levels of abstraction of the Control Structure were included.

The guidance requirements were also expanded with four entirely new requirements. The first of these requirements requires the software tool to provide views that support each of the three use cases: STPA alone, STPA with cooperation, and

utilizing/presenting STPA results. Two requirements define the features regarding generating and copying STPA analysis results. One requirement aims to clarify the analysis process to the user by suggesting that the software tool should be able to indicate which parts of the analysis are complete or incomplete. These requirements together with the preliminary guidance requirements intend to define features that guide the user into conducting the analysis effectively and conveniently.

Prioritization requirements were modified to accept a wider range of approaches to prioritization. RPNs are no longer specifically mentioned, rather just a general requirement for the software tools to be able associate priority values to results was made. The software tools are also required to be able to filter and sort by such priority. These changes were largely due to the feedback from the industry practitioners, who emphasized using traceability to prioritize, or alternatively using a less time-consuming approach to assigning priority than RPNs, such as a single value on a "high - medium - low" – scale.

Much of the other requirements remain largely unchanged. Some of the wording in the traceability requirements was changed to further emphasize the importance of the traceability requirement, however, no requirements were removed and none were added. These slightly revised traceability requirements are presented in Table 12. The uncategorized requirements were increased by one requirement, which is documented along with other requirements in Table 12. It addresses the security needs of the nuclear power industry by stating how the software tool should be possible to operate by the nuclear power licensee.

In creating the revised requirements a balance had to be considered. Requirements that are too detailed may limit the creativity in developing a potential software tool, while requirements that lack specificity may leave room for misunderstanding the goals of the requirement hence leading to poor choices in software design and further resulting in a dysfunctional software tool. Features identified as necessary or very important by industry practitioners were easily translated into requirements, but ideas presented during the focus group meetings as less important were harder to formulate into requirements. Requirements formulated from such ideas needed to be carefully considered to ensure they capture the essence of the idea rather than its specific implementation. However, they also need to be easy to verify when evaluating existing software tools.

An example of a requirement where such a delicate balance had to be considered – “The software tool should have views which support each of the three use cases. . . ” – does not state that the software tool must have a separate view for each use case, nor does it address how views should be available to the user. Similarly, the word “visualize” used in many requirements encompasses forms of visually communicating the data that go beyond simple rows in a table or spreadsheet, and it does not specify a certain type of view, such as a tree-type view. In other instances, words such as “ indicate” are used to allow for creative approaches to a feature, such as using haptic or auditive feedback to communicate information to the user, though these types of feedback are unlikely to be found in the software tools explored in this thesis based on the initial software tool tests at the outset of this work. By wording the requirements as demonstrated, the requirements are specific enough to capture the goal of the

requirement, while remaining relatively straightforward to validate.

Table 12: A table presenting the revised software tool requirements by category, along with their respective reasoning.

Beginning of Table 12		
Category	**Revised requirement**	**Reasoning**
Traceability	The software tool should be able to visualize traceability from loss scenario to UCA to hazard to loss (for each result its entire traceability).	Enables convenient and robust traceability of results.
Traceability	The software tool should be able to conveniently visualize traceability from the Control Structure to results and vice versa.	Enables convenient and robust traceability between the Control Structure and other results. Aids in communicating results to system experts.
Traceability	The software tool should support traceability between the Control Structure and results in general.	Enables traceability between the Control Structure and other results.
Traceability	The software tool should support filtering items based on traceability such as showing items that trace back to a specific Control Structure entity, or showing all scenarios related to a specific control action or hazard.	Enables getting a better understanding of the analysis as a whole. Enables prioritization through traceability.
Traceability	The software tool should support traceability from one STPA analysis to another, for example when conducting analyses at different levels of abstraction or on different levels of the Control Structure.	Enables working iteratively on STPA analyses. Enables traceability between STPA analyses.
Control Structure	The software tool should support creating and modifying the Control Structure of the analysis.	Ensures that the Control Structure can be worked with within the software tool.
Control Structure	The software tool should support viewing the Control Structure of an analysis at different levels of abstraction.	Enables getting a better understanding of the analysis as a whole. Aids in communicating the Control Structure to system experts.

Continuation of Table 12		
Control Structure	The software tool should support iteratively reducing the level of abstraction of the Control Structure over the course of the analysis.	Enables working iteratively on STPA analyses.
Control Structure	The software tool should support color coding the Control Structure.	Enables improving the clarity of the Control Structure.
Guidance	The software tool should use language consistent with the STPA method: i.e. Hazards are labeled as hazards, loss scenarios as loss scenarios, etc.	Lowers the threshold for learning the software tool and STPA method.
Guidance	The software tool should provide guidewords for UCAs.	Lowers the threshold for learning the STPA method and aids in generating UCAs.
Guidance	The software tool should provide guiding questions for Loss Scenarios.	Lowers the threshold for learning the STPA method and aids in generating Loss Scenarios.
Guidance	The software tool should guide the user to document results in the correct syntax.	Lowers the threshold for learning the STPA method and aids consistent and unambiguous documentation of results.
Guidance	The software tool should support copying parts of prior STPA analyses, such as Control Structures, hazards or losses.	Reduces repetitive tasks in the software tool.
Guidance	The software tool should enable conveniently documenting and viewing material related to each result: standards, operating manuals, layouts and process plans, etc.	Enables traceability between results and source material. Aids in accessing documents relevant to the analysis.
Guidance	The software tool should be able to generate some results semi-automatically, namely UCAs from CAs.	Reduces repetitive tasks in the software tool.

Continuation of Table 12		
Guidance	The software tool should have views which support each of the three use cases: STPA alone, STPA with cooperation, STPA result utilization and presentation.	Ensures that each use case is considered during the development of an STPA software tool.
Guidance	The software tool should be able to conveniently indicate the parts of the analysis which have been completed and which parts still need to be explored.	Aids in communicating analysis progress to the user.
Guidance	The software tool should be able to warn the user of incorrectly formatted results, such as a UCA that is missing context.	Aids consistent and unambiguous documentation of results.
Guidance	The software tool should be able to highlight items that need to be reviewed after part of the analysis has been modified, for example, a control action is added $\rightarrow$ review controller constraints and generate UCAs and loss scenarios.	Aids in working iteratively on STPA analyses.
Integration	The software tool should be able to export analysis results in formats accepted by other system/requirements engineering tools.	Enables integrating STPA to the larger system engineering effort.
Integration	The software tool support Risk Analysis and Assessment Modeling Language (RAAML).	Enables integrating STPA to the larger system engineering effort.
Integration	The software tool has the ability to trace decisions in system design and implementation back to analysis results, for example, a certain component could be traced back to specific control actions, loss scenarios, or a countermeasure.	Enables traceability between analysis results and decisions concerning the system being engineered; aids in understanding why certain decisions concerning the system were made.
Integration	The software tool can visualize the traceability from system design and implementation to the analysis results.	Aids in understanding the system engineering effort as a whole.

Continuation of Table 12		
Integration	The software tool can highlight changes made to the design and implementations that require re-analysis.	Enables effective utilization of STPA in the larger system engineering effort.
Prioritization	The software tool should support the prioritization of Loss Scenarios: a) by associating priority values to results, b) by associating results of other hazard analysis methods to the results	Enables flexible approaches to Loss Scenario prioritization in STPA. Aids in utilizing STPA results.
Prioritization	The software tool should support filtering/sorting STPA results by priority.	Enables flexible approaches to Loss Scenario prioritization in STPA. Aids in utilizing STPA results.
Prioritization	The software tool can visualize STPA results in order of priority.	Aids in understanding STPA results and their priority.
Prioritization	The software tool should support associating causal factors to loss scenarios.	Enables prioritization of loss scenarios by causal factors.
Prioritization	The software tool should support filtering/sorting/grouping loss scenarios by their causal factors.	Enables prioritization of loss scenarios by causal factors.
Other	The software tool should support the 2018 revision of STPA.	Ensures the software tool supports the 2018 revision of STPA.
Other	The software tool should support adding notes/comments.	Enables communicating information regarding the analysis.
Other	The software tool should be hostable by the nuclear power plant licensee rather than a third party.	Aids in the security of the software tool.
End of Table 12		

3.6 Finalized software tool requirements

The revised requirements were proofread and evaluated together with an STPA expert from VTT, and some final adjustments were made to the requirements accordingly. Changes were mostly made to the wording of the requirements, with the intention of clarifying them. The set of requirements discussed with the STPA expert were identical to those presented in Table 12, but with the omission of the reasoning column due to time constraints. The set of requirements was annotated by the STPA expert

alone, after which a meeting of approximately 1,5 hours was organized to discuss the annotations together. The resulting set of finalized software tool requirements is presented in Table 13. These requirements will be used in Chapter 4 to evaluate the STPA software tools.

Table 13: A table presenting the revised software tool requirements by category, along with their respective reasoning.

Beginning of Table 13		
Category	**Revised requirement**	**Reasoning**
Traceability	The software tool should be able to visualize traceability from Loss Scenario to UCA to CA to Safety Constraint to Hazard to Loss (for each result its entire traceability).	Enables convenient and robust traceability of results.
Traceability	The software tool should be able to conveniently visualize traceability from the Control Structure to results and vice versa. E.g. Clicking on a CA in the Control Structure shows UCAs, Loss Scenarios, etc., or other similar functionality.	Enables convenient and robust traceability between the Control Structure and other results. Aids in communicating results to system experts.
Traceability	The software tool should support traceability between the Control Structure and results in general.	Enables basic traceability between the Control Structure and other results.
Traceability	The software tool should support filtering items based on traceability such as showing items that trace back to a specific Control Structure entity, or showing all scenarios related to a specific control action or hazard.	Enables getting a better understanding of the analysis as a whole. Enables prioritization through traceability.
Traceability	The software tool should support traceability from one STPA analysis to another, for example when conducting analyses at different levels of abstraction or on different sections of the Control Structure.	Enables working iteratively on STPA analyses. Enables traceability between STPA analyses.
Control Structure	The software tool should support creating and modifying the Control Structure of the analysis.	Ensures that the Control Structure can be worked with within the software tool.

Continuation of Table 13		
Control Structure	The software tool should support viewing the Control Structure of an analysis at different levels of abstraction (zooming in/out).	Enables getting a better understanding of the analysis as a whole. Aids in communicating the Control Structure to system experts.
Control Structure	The software tool should support iteratively reducing the level of abstraction of the Control Structure over the course of the analysis.	Enables working iteratively on STPA analyses.
Control Structure	The software tool should support color coding the Control Structure.	Enables improving the clarity of the Control Structure.
Guidance	The software tool should use language consistent with the STPA method: i.e. Hazards are labeled as hazards, loss scenarios as loss scenarios, etc.	Lowers the threshold for learning the software tool and STPA method.
Guidance	The software tool should provide the 4 guidewords for UCAs: "Not providing", "Providing", "Too late/early/wrong order", and "Too long/short"	Lowers the threshold for learning the STPA method and aids in generating UCAs.
Guidance	The software tool should provide guiding questions for Loss Scenarios.	Lowers the threshold for learning the STPA method and aids in generating Loss Scenarios.
Guidance	The software tool should guide the user to document results in the correct syntax. For example for UCAs, five items need to be included: "Source", "type", "Control Action", "context", and "link to hazards".	Lowers the threshold for learning the STPA method and aids consistent and unambiguous documentation of results.
Guidance	The software tool should support copying parts of prior STPA analyses, such as Control Structures, hazards or losses.	Reduces repetitive tasks in the software tool.

Continuation of Table 13		
Guidance	The software tool should enable conveniently documenting, viewing, and linking material related to each result: standards, operating manuals, layouts and process plans, etc.	Enables traceability between results and source material. Aids in accessing documents relevant to the analysis.
Guidance	The software tool should be able to generate some results semi-automatically, namely UCAs from CAs.	Reduces repetitive tasks in the software tool.
Guidance	The software tool should have views which support each of the three use cases: STPA alone, STPA with cooperation, STPA result utilization and presentation.	Ensures that each use case is considered during the development of an STPA software tool.
Guidance	The software tool should be able to conveniently indicate the parts of the analysis which have been completed and which parts still need to be explored.	Aids in communicating analysis progress to the user.
Guidance	The software tool should be able to warn the user of incorrectly formatted results, such as a UCA that is missing context.	Aids consistent and unambiguous documentation of results.
Guidance	The software tool should be able to highlight items that need to be reviewed after part of the analysis has been modified, for example, a control action is added $\rightarrow$ review controller constraints and generate UCAs and loss scenarios.	Aids in working iteratively on STPA analyses.
Integration	The software tool should be able to export analysis results in formats accepted by other system/requirements engineering tools.	Enables integrating STPA to the larger system engineering effort.
Integration	The software tool should support Risk Analysis and Assessment Modeling Language (RAAML).	Enables integrating STPA to the larger system engineering effort.

Continuation of Table 13		
Integration	The software tool has the ability to trace decisions in system design and implementation back to analysis results, for example, a certain component could be traced back to system requirements informed by specific loss scenarios, or a countermeasure.	Enables traceability between analysis results and decisions concerning the system being engineered; aids in understanding why certain decisions concerning the system were made.
Integration	The software tool can visualize the traceability from system design and implementation to the analysis results.	Aids in understanding the system engineering effort as a whole.
Integration	The software tool can highlight changes made to the design and implementations that require reanalysis.	Enables effective utilization of STPA in the larger system engineering effort.
Prioritization	The software tool should support the prioritization of Loss Scenarios: a) by associating priority values to results, b) by associating results of other hazard analysis methods to the results	Enables flexible approaches to Loss Scenario prioritization in STPA. Aids in utilizing STPA results.
Prioritization	The software tool should support filtering/sorting STPA results by priority.	Enables flexible approaches to Loss Scenario prioritization in STPA. Aids in utilizing STPA results.
Prioritization	The software tool can visualize STPA results in order of priority.	Aids in understanding STPA results and their priority.
Prioritization	The software tool should support associating causal factors to Loss Scenarios and filtering/sorting/grouping Loss Scenarios by their causal factors.	Enables prioritization of loss scenarios by causal factors.
Other	The software tool should support the 2018 revision of STPA.	Ensures the software tool supports the 2018 revision of STPA.
Other	The software tool should support adding notes/comments.	Enables communicating information regarding the analysis.

Continuation of Table 13		
Other	The software tool should be hostable by the nuclear power plant licensee rather than a third party.	Aids in the security of the software tool.
End of Table 13		

4 Evaluating software tools

After determining the software tool requirements, available software tools were explored and evaluated against these requirements. The software tools were installed on a Mac OS computer if supported by the software tool, or on a Microsoft Windows computer otherwise. During the evaluations, the MacOS version was 12.7.2 and the Windows computer was operating on Windows 10 OS build 19045.4651 unless otherwise specified. Once installed, the software tools were input with the results produced in the case study STPA analysis of a reactor feedwater system, also outlined in Chapter 2. Finally, each software tool was tested extensively to ensure they are evaluated true to their actual features.

This chapter outlines the scope of the evaluation and presents a brief overview of each tool's installation process, the results of each software tool's evaluation, as well as a comparison of the software tool evaluations.

4.1 Scope of the evaluation

The software tools included in this evaluation are those listed on the MIT website "MIT Partnership for Systems Approaches to Safety and Security (PSASS)" [33]. The number of available STPA software tools is large, and adding the case study data to each software tool is extremely time-consuming. However, the comprehensiveness of these tools varies significantly, from simple tools that handle one aspect of the STPA method to tools that are capable of integrating STPA with other system engineering activities. For example, the software tool "Depict" by Michael Stone, is listed as a STAMP tool on the aforementioned MIT PSASS website [33]. The tool's functionality is limited to generating the Control Structure using a proprietary syntax. While generating a Control Structure is a key part of the STPA analysis, the feature alone does not make the software tool a viable option in the context of the nuclear industry in Finland and the analysis of nuclear I&C systems.

In order to address the time-consuming nature of the evaluation of the available software tools, a set of requirements that were identified as important and easy to verify was used as a basis for excluding some software tools from the evaluation. These requirements are presented in Table 14. Table 15 in turn, lists the tools which were excluded from the in-depth evaluation and which of the criteria presented in Table 14 they did not satisfy. In addition, the software tool must be readily available to be tested, as otherwise the analysis data from the reactor feedwater system case study cannot be used to evaluate the software tool. This, unfortunately, excludes many tools listed on on the MIT PSASS website [33].

Table 14: The minimum conditions software tools need to meet to be considered in this evaluation.

Category	Requirement
Control Structure	The software tool should supports creating and modifying the Control Structure of the analysis.
Other	The software tool should support the 2018 revision of STPA.
Other	The software tool should be hostable by the nuclear power plant licensee rather than a third party.

Table 15: This table presents the tools which were excluded from the evaluation, and the reasoning for their exclusion.

Tool name	Reasoning
An STPA tool	The software tool is not publicly available.
Depict	The software tool does not fully support the 2018 revision of STPA (only supports modeling the Control Structure).
SafetyHAT	The software tool does not support creating or modifying the Control Structure of the analysis.
SAHRA	The software tool is not publicly available.
SpecTRM	The software tool is not publicly available.
STAMP Workbench	The software tool doesn't support the 2018 revision of STPA.
STPA Automation tool	Can not be hosted by the nuclear power plant licensee as the tool is accessed used in Google Sheets.
STPAMaster Lite	Can not be hosted by the nuclear power plant licensee as the tool is accessed used in Google Sheets.

4.2 Tool evaluation 1: STPA Viewpoint for Capella

"STPA Viewpoint for Capella" is an add-on for the Model-Based Systems Engineering software tool Capella. The Capella website lists multiple large companies as adopters of the software tool, and also highlights the numerous case-studies involving the tool from industries ranging from aviation to nuclear energy. Both Capella and the STPA add-on are open source, and the latter can be installed using instructions available on GitHub, to which a link is available on MIT website listing STAMP software tools [33].

The software tool was installed on a Mac OS computer, a M1 Macbook Air running Mac OS 12.7.2 at the time of testing. Data from the reactor feedwater case study was fed into the software tool to an extent that would allow verifying the tool's features. Results were inputted in the format they were originally produced in, e.g. a result would have the same ID and name in both the original case study and the one recreated in the software tool. The case study was recreated in the tool following the software tool's step-by-step user guide.

Of the 32 software tool requirements, the software tool fully satisfied 11 requirements, partially satisfied 12 requirements, and didn't satisfy 9 requirements. A general

overview of this evaluation is presented in Table 16, and a more detailed evaluation with comments concerning the software tool with regard to each requirement is available in Appendix A.

Many of the key findings made regarding the software tool during its evaluation concern the tool's usability. Issues with the software tool that are not demonstrated in the evaluation Table 16 include issues with legibility, ease of navigating the user interface, and ease of accessing the features defined in the set of requirements. For example, text in the different tables of STPA results were often hard to read, simply due to a total lack of contrast between the text and its background. This may be due to the software tool adopting the "dark mode" theme of the operating system incorrectly. Another issue concerned the STPA-IDs of each result: the IDs reset to their default values each time a new result was added. For example, the tool would format the STPA-IDs of UCAs as "UCA-01", "UCA-02", etc, by default. If UCAs were added in a alternative format such as "UCA-2.1", "UCA-2.2", etc, the tool would reformat the results into its default format each time a new result was added, resulting in confused attempts at understanding the tool's behavior. Other similar minor issues in the tool together with the author's lack of familiarity with the tool's foundation, Capella, compounded to a frustrating and confusing user experience.

The software tool also relies heavily on tables to present the results, and while generating results was less mentally tasking due to the guidance the tool provided, visual representations of the results were lacking. Visualizing links between results was not straightforward, nor was it very useful. For example, the author was unable to visualize UCAs related to a certain hazard, or visualize the entire chain of such relations from Loss Scenario to Loss. The software tool was also unable to sort or filter results by their relation to other results. For example, a table of UCAs could not be arranged to show just results tied to a certain Hazard.

There are aspects of the STPA analysis that the tool handles well, however. For example, the generation of Loss Scenarios is more approachable due to the generation of "Causal Factor Diagrams", which are visual representations of the factors that contribute to an UCA occurring. These causal factors are then documented in each Loss Scenario. An example of a Causal Factor Diagram generated by STPA Viewpoint for Capella is presented in Figure 5. This systematic approach to generating Loss Scenarios guides the user to thoroughly investigate the possibile Scenarios a UCA could contribute to. The tool is also capable of integrating STPA as a part of the system engineering whole, a merit which many tools lack.

Overall, STPA Viewpoint for Capella demonstrates many useful features. Of the 32 functional requirements an ideal STPA software tool would satisfy STPA Viewpoint for Capella satisfies or partially satisfies 23. However the implementation of these features lacks in practical usability, especially for those lacking familiarity with Capella and those wanting to conduct isolated STPA analyses. In the latter scenario, conventional approaches to STPA would likely be preferrable over the software tool. However, these conclusions are limited in multiple aspects: the tool was not tested as a part of a larger system engineering effort as likely intended by its creators and the conclusions were drawn based on the observations of one tester.

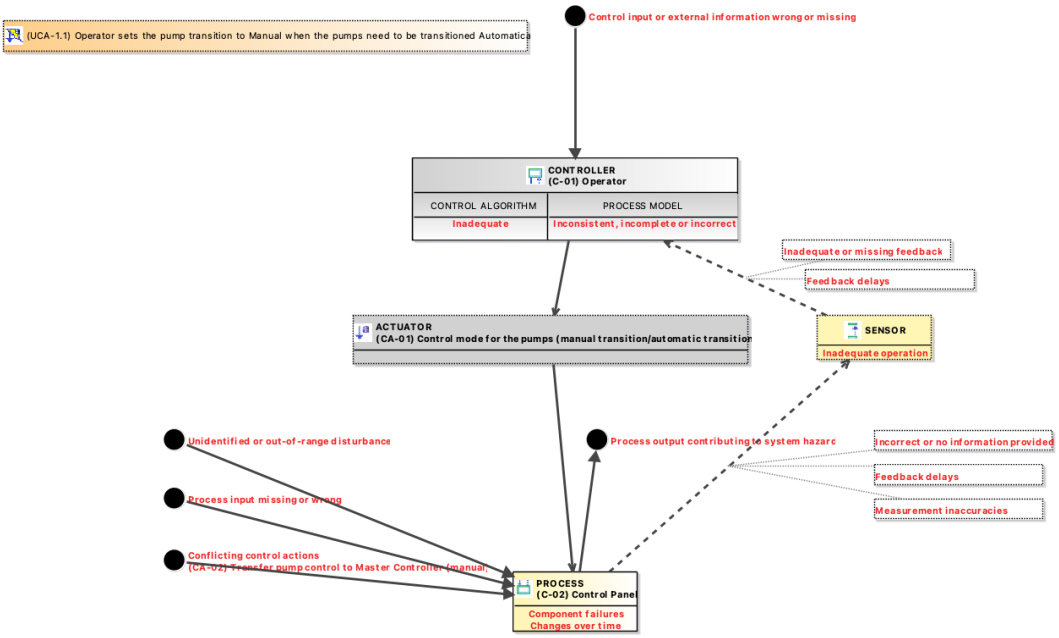

Figure 5: An image exported from STPA Viewpoint for Capella, demonstrating how a Causal Factor Diagram is presented in the tool for a UCA.

Table 16: A table showing the results of the evaluation for STPA Viewpoint for Capella.

Beginning of Table 16		
Category	**Requirement**	**Satisfies Requirement**
Traceability:	Visualizing result traceability.	Partially
Traceability:	Visualizing Control Structure to result traceability.	Partially
Traceability:	Support for Control Structure to result traceability in general.	Yes
Traceability:	Filtering results by traceability.	No
Traceability:	Traceability between different analysis/levels of abstraction.	No
Control Structure:	Creating and modifying the Control Structure.	Yes
Control Structure:	Viewing the Control Structure at different levels of abstraction.	No
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	Partially
Control Structure:	Color-coding the Control Structure.	Yes
Guidance:	Language consistent with the STPA Handbook.	Yes
Guidance:	4 guidewords for UCAs.	Yes
Guidance:	Guiding questions for Loss Scenarios.	Partially
Guidance:	Enforcing correct syntax for results.	Yes
Guidance:	Documenting, viewing, and linking related material.	Partially

Continuation of Table 16		
Guidance:	Copying parts of a prior analysis.	Partially
Guidance:	Partial auto-generation of results (specifically UCAs important).	No
Guidance:	Views supporting each use case.	Partially
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Yes
Guidance:	Warning of incorrectly formatted results.	No
Guidance:	Highlight results to be reviewed after analysis modification.	Partially
Integration:	Export results in formats accepted by other systems engineering tools.	Partially
Integration:	Support for RAAML.	No
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	Yes
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	Partially
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	Partially
Prioritization:	Prioritizing Loss Scenarios/results.	No
Prioritization:	Filtering/sorting results by priority.	No
Prioritization:	Visualizing results by priority.	No
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Partially
Other:	Support for 2018 STPA revision.	Yes
Other:	Support for adding notes/comments.	Yes
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Yes
End of Table 16		

4.3 Tool evaluation 2: VisualPro STPA

"VisualPro STPA" is a software tool developed by VWAY a company, that focuses on creating software for engineering processes. For the purpose of this evaluation, a trial version was requested through VWAY's website. The request was followed by an email from a representative of VWAY offering a link to the installation files for the tool. The tool was installed on a Windows PC running the Windows 10 Home operating system OS build 19045.3448. At the time of writing this evaluation, the tool is under active development, and features such as co-working on the analysis and integration into the larger system engineering whole are being developed. The company's website also mentions a possibility to integrate the tool with other software such as Polarion by Siemens for application lifecycle management. Testing integration

with other software is, however, beyond the scope of this work, and therefore remains unverified in this evaluation.

Overall, the software tool satisfied 14 requirements and partially satisfied 11 requirements, while 7 requirements were left unsatisfied by the tool. The results of the evaluation are presented in Table 17, and a more detailed evaluation with comments on how the software tool satisfied each requirement is provided in Appendix A. Notably, the software tool satisfies 4 out of 5 Traceability requirements, and either entirely or partially satisfies 7 out of 11 Guidance requirements.

Basic traceability features are intuitively and conveniently accessed in the tool. For example, the user can search all results produced during the analysis for keywords and show only those results pertaining to a certain Control Structure entity. For example, the author was able to visualize the traceability of results which mentioned the system state "Scram". In this case UCAs which mentioned the keyword, and results related to these UCAs such as Losses, Hazards and Loss Scenarios were shown. For evaluation purposes, a countermeasure was added to the analysis as well, namely one which proposed redundant control panel indicators to prevent certain Loss Scenarios. Upon searching for "Redundant" in the "Trace View" of the tool, the tool visualized all the results related to the countermeasure. Searching by the STPA-ID of a result produces a similar view. The search feature in the tool's "Trace View" relies on results being documented with the same keywords each time. For example, searching for "Redundancy" will not return for results containing the word "Redundant". The tool does, however, encourage users to use the same words when documenting results. For example, the names of Control Structure entities such as controllers, processes, and feedback are suggested when typing into the text fields of the results.

In addition to filtering results by keywords, the Trace View in the tool provides an easy way of navigating the results. For example, double-clicking a result highlights the results along with all the results linked to it from Countermeasure to Loss. Upon right-clicking any result in the Trace View, the user is prompted to visit the definition of the result, a page where you can edit the result and view it in more detail. This is useful, as the Trace View easily reveals errors in the documentation of results. For example, some of the UCAs in the case study data weren't linked to any Hazard, despite the Loss Scenarios generated from them being documented together with Hazards. In the Trace View this quickly became apparent, as these results were visualized separate from the other results. A screenshot of the software tool's Trace View, which demonstrates this functionality is presented in Figure 6.

In terms of the use cases identified during the generation of the software tool requirements, the tool implements features which address these use cases well. The first use case, an STPA practitioner working alone on an analysis, requires that the very basic features are met at the minimum. However, the second and third use case require some further consideration. In the second use case, "Cooperating on STPA with system experts", the software tool needs to be able to convey the analysis effectively to people who may not be familiar with STPA. During the requirements review focus groups, industry practitioners identified the need to simultaneously display the Control Structure and other analysis results as very important, a feature that VisualPro STPA does by default for UCAs and Loss Scenarios. This feature is presented in

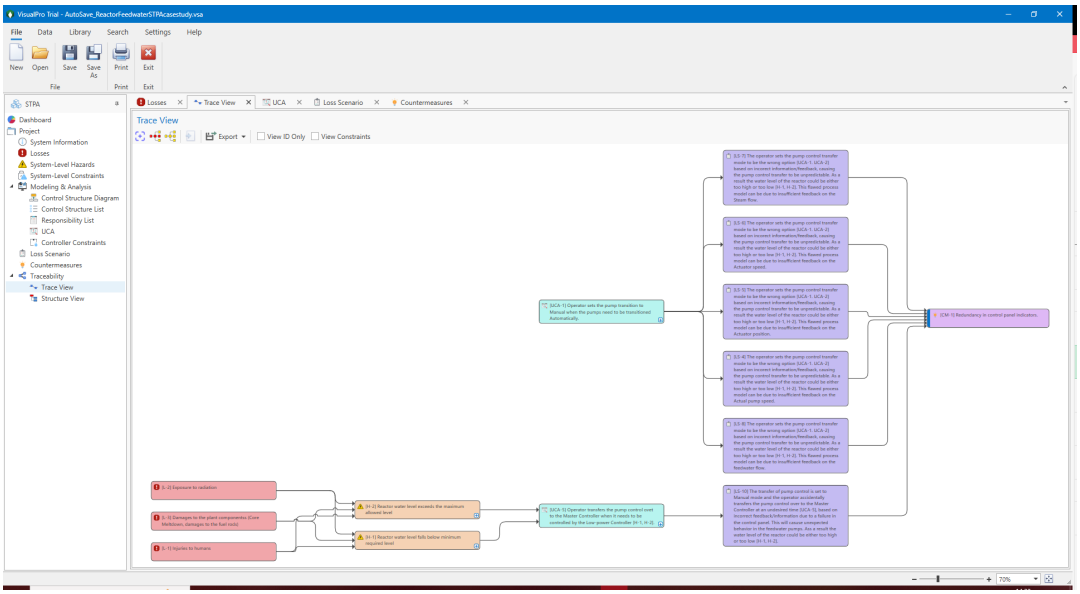

Figure 6: A screenshot demonstrating how missing traceability in results is highlighted in VisualPro SA's Trace View. A UCA which does not link to any Hazard can be observed in the top center of the image.

Figure 7 along with the typing suggestion feature described earlier in this evaluation. For the third use case, "Utilizing the results of the STPA analysis", conveying the results to stakeholders such as foremen or decision makers was considered important. The software tool enables generating reports in formats such as DOCX, PDF, and Powerpoint. In addition it provides a view within the software tool showing statistics on the analysis. While the value of the statistics presented in the tool remain unconfirmed, a consideration towards this use case is apparent in the software tool.

Many significant observations of the tool's performance beyond the functions specified in the requirements were made during the evaluation of the tool. The tool was straightforward to install, and the tool's learning curve is shallow; The author could start feeding the data from the case study within minutes of launching the tool for the first time. From the perspective of user-friendliness, the tool was easy to navigate and the author felt comfortable clicking the various icons, i.e. there was little fear of causing irreversible changes to the project through such actions.

Though the overall experience of using the tool was very positive, some minor issues were discovered during use. One issue related to the format of the STPA-ID. Similar to the Capella-based software tool, VisualPro STPA enforced a format which differed from how the results were originally documented. This made the process of transferring the data into the tool confusing as the amount of case study data replicated in the tool increased. Another issue concerned the naming of Control Actions in the Control Structure. Specifically, where the original case study data often had one Control Action referring to multiple instances of control in the Control Structure, the tool treated each instance as separate control actions. Hence, in the case of this evaluation, it was not possible to replicate the entirety of the case study data in the

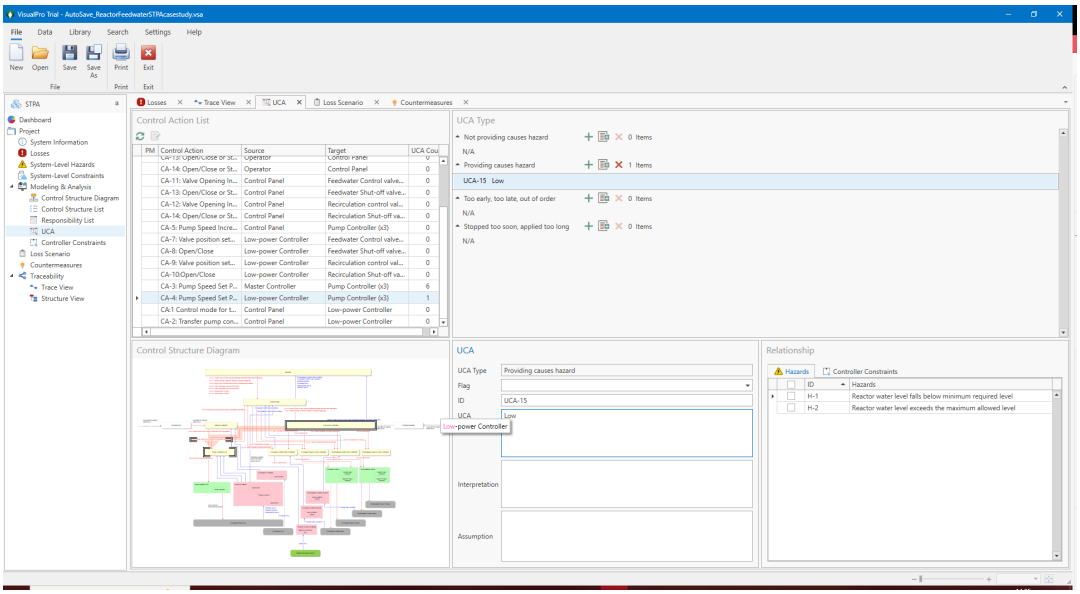

Figure 7: A screenshot of VisualPro SA demonstrating how the tool presents the Control Structure with UCAs and how the tool utilizes traceability to the Control Structure in typing suggestions during the creation of UCAs.

software tool. Similarly, single Loss Scenarios concerning multiple UCAs were not possible to document as such in the software tool, rather, each UCA had to have its own set of Loss Scenarios leading to more inconsistencies between the original case study data and the data replicated in the software tool. The tool also supports only a single Control Structure per analysis, and iteratively working on the Control Structure doesn't seem to be considered by the developers.

Overall, the software tool demonstrated many of the required features in a well thought-out manner. Many aspects identified as very important by industry practitioners during the review focus groups such as the traceability and Control Structure features were implemented such that the features are convenient and easy to understand. In these aspects, the software tool makes a good case for itself as a standalone STPA analysis tool, as well as a highly promising tool for utilizing STPA in a larger system engineering context, though further assessment of its suitability for this purpose is required.

Table 17: A table showing the results of the evaluation for VisualPro SA.

Beginning of Table 17		
Category	**Requirement**	**Satisfies Requirement**
Traceability:	Visualizing result traceability.	Yes
Traceability:	Visualizing Control Structure to result traceability.	Yes
Traceability:	Support for Control Structure to result traceability in general.	Yes
Traceability:	Filtering results by traceability.	Yes
Traceability:	Traceability between different analysis/levels of abstraction.	No
Control Structure:	Creating and modifying the Control Structure.	Yes
Control Structure:	Viewing the Control Structure at different levels of abstraction.	Partially
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	No
Control Structure:	Color-coding the Control Structure.	Yes
Guidance:	Language consistent with the STPA Handbook.	Yes
Guidance:	4 guidewords for UCAs.	Yes
Guidance:	Guiding questions for Loss Scenarios.	Yes
Guidance:	Enforcing correct syntax for results.	No
Guidance:	Documenting, viewing, and linking related material.	Partially
Guidance:	Copying parts of a prior analysis.	No
Guidance:	Partial auto-generation of results (specifically UCAs important).	Partially
Guidance:	Views supporting each use case.	Yes
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Yes
Guidance:	Warning of incorrectly formatted results.	No
Guidance:	Highlight results to be reviewed after analysis modification.	Partially
Integration:	Export results in formats accepted by other systems engineering tools.	Partially
Integration:	Support for RAAML.	No
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	Partially
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	Partially

Continuation of Table 17		
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	Partially
Prioritization:	Prioritizing Loss Scenarios/results.	Partially
Prioritization:	Filtering/sorting results by priority.	Partially
Prioritization:	Visualizing results by priority.	No
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Partially
Other:	Support for 2018 STPA revision.	Yes
Other:	Support for adding notes/comments.	Yes
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Yes
End of Table 17		

4.4 Tool evaluation 3: RMStudio

"RMStudio" is a risk management tool developed by "Stiki - Information Security" and is available as both a browser-based tool as well as a desktop program. For the purpose of this evaluation, a trial was requested through the RMStudio website. Instructions and a trial license for both the Web and Desktop versions of the software tool were received through email. The web version was evaluated on a Google Chrome browser version 126.0.6478.182 using a Windows 10 Home desktop computer. The latter was running OS build 19045.4651 at the time of the evaluation, and also served as the device used for the evaluation of the desktop version of the software tool.

The web and desktop versions were very similar to each other, and were considered to be one software tool for the purpose of this evaluation. However, a separate evaluation of each version would have yielded lower scores for each version, as some of the differences concerned features defined in the requirements. Overall, RMStudio satisfied 13 requirements, partially satisfied 9 requirements, and did not satisfy 10 requirements across the two versions. These results are presented in Table 18, as well as in Appendix A, which also presents the reasoning for the evaluation of the tool with regard to each requirement.

As with VisualPro SA, getting started with RMStudio was fairly straightforward, especially with the web based version. STPA is one aspect of the software tool, and there are features beyond the analysis provided in the tool, though these features appear to cater more towards managing security than, for example, supporting a Model-Based Systems Engineering process.

While the tool satisfies some of the requirements concerning traceability, it lacks in features that visualize the relationships between results produced by an analysis. At the time of evaluation, the tool could only visualize the connections between losses and hazards. A connection between the results and the Control Structure could not be visualized by the tool either. However, basic traceability features were well implemented, such as the possibility to create sub-diagrams for the Control

Structure combined with the possibility to associate UCAs derived from many Control Structure diagrams to each Loss Scenario, the latter feature being available only on the Web-based version of the tool. These features enable iteratively adding detail to the system model and being able to keep earlier, more abstract versions of the Control Structures and results generated from them.

In terms of documenting results, one feature offered by the tool was the partial automatic generation of UCAs. Based on the Control Action serving as the basis for the UCA and the latter's type, the tool automatically writes the source, type, and control action to the description of the UCA, leaving the user to write the context and tick the boxes for the Hazards the UCA is linked to. An example of the partial automatic generation of UCAs is presented in Figure 8. The partial automatic generation of UCAs was specifically identified by industry practitioners during the focus group meetings as something that would relieve them of a significant amount of repetitive work in analyses. The automatically generated parts of the UCA description could also potentially serve to lower the learning curve for the STPA method.

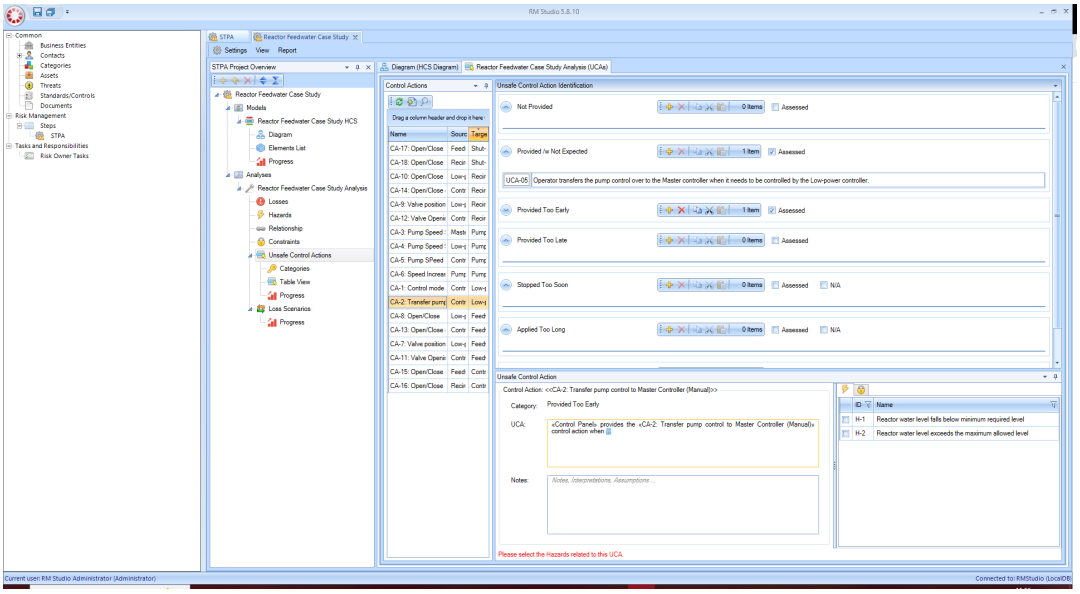

Figure 8: A screenshot demonstrating RMStudio's UCA autogeneration functionality.

Another idea presented by the industry practitioners during the meetings was the ability to see the progress of the analysis in the software tool. In the desktop version of RMStudio, this feature is implemented through "Progress Views", which show a progress bar and items that need to be resolved for the progress bar to advance. These items include unlabeled control actions or feedback in the Control Structure, and for example Control Actions which are missing UCAs. Clicking on an item that needs to be resolved will lead to the view where it can be addressed. After the creation of a Control Structure, most of the analysis could be conducted by simply clicking on each item that needs to be resolved, creating the result, and moving onto the next item to resolve. However, the Web-based version does not include the Progress View feature.

The software tool's functionality beyond the requirements used for the evaluation,

such as the general user experience, left a mixed impression on the author. Some of the basic and necessary functionality in the tool was hard to use. For example, when editing the Control Structure in the desktop version of tool, adjusting the sizes and locations of controllers would often result in the connectors, such as control action arrows, being contorted beyond repair. This made adjusting the Control Structure diagram into a legible format more labour-intensive than it needed to be. However, the Web-based version was faster to navigate and use, but instead lacked in features such as the Progress View.

Overall, RMStudio demonstrates many well implemented features: UCA autogeneration, features that support iterative Control Structure approaches, and the Progress View. These features are contrasted by aspects of the tool that, in their current state, would require further refinement, such as the visualization of results and glitchy performance of the desktop version. A possibility to associate potential and implemented countermeasures to the analysis results could further improve the tool's usability for systems engineering tasks.

Table 18: A table showing the results of the evaluation for RMStudio.

Beginning of Table 18		
Category	**Requirement**	**Satisfies Requirement**
Traceability:	Visualizing result traceability.	Partially
Traceability:	Visualizing Control Structure to result traceability.	No
Traceability:	Support for Control Structure to result traceability in general.	Yes
Traceability:	Filtering results by traceability.	Yes
Traceability:	Traceability between different analysis/levels of abstraction.	Yes
Control Structure:	Creating and modifying the Control Structure.	Yes
Control Structure:	Viewing the Control Structure at different levels of abstraction.	Partially
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	Partially
Control Structure:	Color-coding the Control Structure.	Yes
Guidance:	Language consistent with the STPA Handbook.	Yes
Guidance:	4 guidewords for UCAs.	Yes
Guidance:	Guiding questions for Loss Scenarios.	Partially
Guidance:	Enforcing correct syntax for results.	Partially
Guidance:	Documenting, viewing, and linking related material.	Partially
Guidance:	Copying parts of a prior analysis.	No

Continuation of Table 18		
Guidance:	Partial auto-generation of results (specifically UCAs important).	Yes
Guidance:	Views supporting each use case.	Partially
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Yes
Guidance:	Warning of incorrectly formatted results.	No
Guidance:	Highlight results to be reviewed after analysis modification.	Yes
Integration:	Export results in formats accepted by other systems engineering tools.	Partially
Integration:	Support for RAAML.	No
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	No
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	No
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	No
Prioritization:	Prioritizing Loss Scenarios/results.	No
Prioritization:	Filtering/sorting results by priority.	No
Prioritization:	Visualizing results by priority.	No
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Partially
Other:	Support for 2018 STPA revision.	Yes
Other:	Support for adding notes/comments.	Yes
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Yes
End of Table 18		

4.5 Tool evaluation 4: CAIRIS Support

"CAIRIS support" is a software tool created by Dr. Shamal Faily from Bournemouth University. The software tool can be hosted on a virtual machine running on the user's computer, and accessed through the user's browser. To this end, the user has to use other tools such as Vagrant and Virtualbox to setup and run the tool on their computer. This approach to installing CAIRIS was attempted on both a M1 Macbook Air and a Windows 10 Home desktop computer. On the former device, the author was unable to get the software tool to function, likely due to an incompatibility of the virtual machines with the M1 System-on-Chip in the Macbook Air. On the latter device, however, the author managed to setup the virtual machine and CAIRIS to display the tool's login screen in their browser. However, any attempt at logging in with the provided credentials was thwarted by an undefined server error from the software

tool. Due to the unsuccessful attempts at installing the software tool, this evaluation will be based on the live demo website. The live demo website is rebuilt once a day, erasing any progress made during the preceding day, hence only a partial evaluation is presented here. This evaluation should be considered a subjective account of the author's experience with the tool, as the set of requirements used to evaluate other tools will not be assessed against here due to the constraints set by the live demo.

Compared to many of the other tools evaluated in this thesis, installing CAIRIS is slightly less straightforward, and requires a deeper understanding of operating systems, namely in using the Terminal or Command Line interfaces. While these interfaces are very familiar to the author, they do provide an additional challenge to those used to installing computer programs using an installer. In order to install CAIRIS on their own computer, the user first needs to install a command line tool called Vagrant, and a virtualizer called Virtualbox. The instructions provided on the CAIRIS website are clearly written for those already familiar with these tools, and provide no guidance on installing them. Ultimately, though all instructions were followed, the author was unable to get the software tool to function on their computers.

Another key finding about the software tool concerns its approach to supporting STPA. In order to conduct an STPA analysis in the tool, the user is required to utilize features which are likely originally intended for another form of analysis or modeling. For example, the tool considers Losses, Hazards and UCAs as types of "Obstacles", and this is not immediately apparent to the user through the user interface, rather it becomes evident through the guide to STPA provided on the CAIRIS website. Similarly, creating Control Structures requires following steps which the author was unable to complete. According to the CAIRIS website, creating Control Structures needs to be preceded by the creation of processes, data stores, use cases, information assets, data flow diagrams and trust boundaries. In order to create these items, other items were required, such as "Actors", an item which the author was unable to create in the live demo website provided by the developers. In contrast, other tools have enabled the creation of a Control Structure in just a couple of clicks, without the need to define additional items first. This issue is demonstrated in Figure 9.

Many of these issues are captured in the requirement "The software tool should use language consistent with the STPA method". The reasoning for the requirement being, that learning a new software tool becomes much easier when it relies on concepts already familiar to the user. In addition, if a user is also unfamiliar to the STPA method, a software tool consistent with the method can be used to learn STPA. However, a software tool using language inconsistent with the STPA method could lead to an additional burden in learning the method.

In the live demo of CAIRIS, many items required knowing answers to questions that would normally not be necessitated by the STPA method. For example, creating a Loss using the "Obstacle" feature of the tool leads to the user being prompted to fill in details such as the originator, category, definition, probability and reasoning. Similar to the creation of a Control Structure in the tool, there are too many new and uncertain concepts to input during the initial phases of the analysis, and the tool seems to encourage filling in all the details of the analysis upfront, rather than working down from a higher level of abstraction to a lower one.

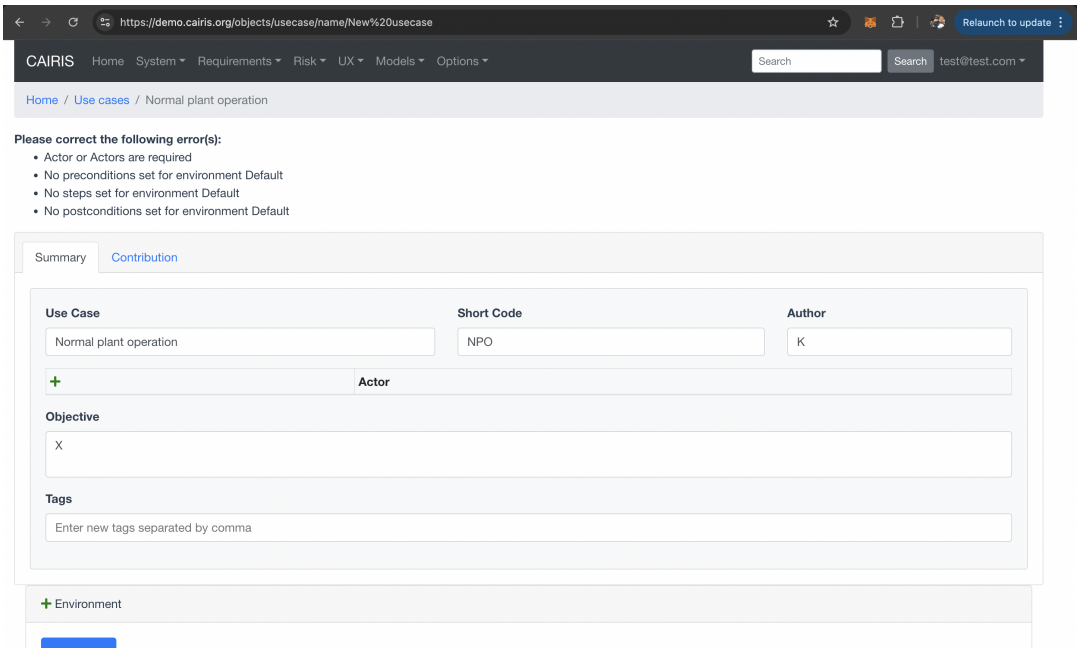

Figure 9: A screenshot of the CAIRIS software tool in a Google Chrome browser taken during one of multiple attempts to create a Control Structure diagram in the tool. The tool's guide to conducting an STPA analysis in the tool has no mention of the items displayed in the error messages, and how to address the issues presented in the error messages did not become apparent to the author even after a thorough investigation of the software tool.

Despite the many weaknesses of the tool described above, the tool can be commended for its browser-friendly implementation and smooth performance. Most actions in the tool were quickly followed by the intended action, or a mostly informative error message, such as a field that was left unfilled. However, as demonstrated in Figure 9, these error messages were not always clear in indicating the actions a user must take to resolve an issue.

Based on the experiences of the author presented here, the tool cannot be recommended for STPA analyses in the nuclear context. A large amount of time, approximately 30 hours was spent with just the initial setup of the tool, and a similarly large amount of time was spent attempting to input the case study data into the live demo version of the tool. In this time the author could have replicated much of the case study data in most other software tools evaluated in this work, or even tools such as Excel and Visio. Both the tool's installation and learning processes could be expedited significantly by providing more detailed, step-by-step instructions. Perhaps in the tool itself there could be a separate view for conducting an STPA analysis, with only the features relevant to the analysis process. Alternatively a sample STPA project could be provided in the tool. Despite the software tool not performing well in the context this evaluation, it should not be discounted in its ability to provide a usable platform for analysis activities beyond STPA.

4.6 Tool evaluation 5: XSTAMPP

XSTAMPP is a STAMP tool which supports STPA, STPA-Sec, STPA-Priv and CAST analyses. It was released in March 2015 as a continuation to the A-STPA software tool project [34]. For the purpose of this evaluation, it was installed on a Windows 10 Home PC running OS build 19045.4780. In this case, installing the software tool required reinstalling Java and installing Maven. About 90 minutes were spent installing and verifying each of the steps for the installation were completed as specified.

Immediately at the beginning of the evaluation process, it became apparent to the author that the software tool is no longer being worked on. The last update to the tool was posted to its GitHub page approximately 5 years before this evaluation took place, and the website for the tool is no longer available. After experimenting with the tool for roughly 15 hours, the author was forced to look for documentation online on how to proceed with the analysis in the tool. While an archived version of the XSTAMPP website [35] was able to reveal that documentation used to be available on the website, they were not available through the archived website. A third party website, however, still hosted a few of the tutorial presentation slides [36]. These slides were used as the main source of information on how to conduct an STPA analysis with the software tool.

Based on a time-intensive, but incomplete replication of the case study data in the software tool, XSTAMPP satisfies 7 requirements, partially satisfies 10 requirements, and does not satisfy 15 requirements. These results are documented in Table 19. Many of the requirements would likely have been satisfied if the tool had displayed less unreliable behavior. It appeared that the prerequisites for certain features were often already implemented into the software tool, however their implementations were often dysfunctional. For example, the feature enabling the user to filter certain results of the analysis enabled the user to use the Causal Factor field of a Causal Scenario (i.e. Loss Scenario) to filter out results, however keywords only limited the results to scenarios with defined Causal Factors, rather than scenarios with the specific keyword in the Causal Factors field of the scenario. Similarly, sometimes features would not work at first, but worked after restarting the software tool or computer. Oftentimes, it appeared that there was a certain sequence to how the results must be input into the software tool, but this sequence was not communicated in any way to the user, other than through the inability to enter information into a field or create a result.

The most immediate disadvantage to the software tool is in its lack of documentation. Official documentation is no longer available, and the tool's GitHub page only hosts a limited set of instructions for those unacquainted with the tool to follow. Combined with features implemented in ways that do not entirely align with the 2018 STPA Handbook [3], the software tool is hard to learn and utilize. Up-to-date, step-by-step, tutorials would provide immense help in utilizing the tool, however some of these issues likely also stem from design issues. The original tool predates the 2018 STPA revision, and therefore was likely designed with the workflow of the earlier STPA version in mind despite being updated to accommodate the 2018 STPA revision since. This is further highlighted by the absence of some of the key terminology presented in the 2018 Handbook, for example there are no mentions of Loss Scenarios, and Losses

are still labeled "Accidents".

One feature the tool offers that has not been implemented in other tools is the ability to use Linear Temporal Logic (LTL) for the formal verification of the STPA analysis. While this feature was not tested by the author due to lacking documentation and time constraints, such a feature could bring additional value to a software tool especially in the nuclear domain.

Overall, the tool cannot be recommended for conducting neither standalone STPA analyses nor STPA analyses in the nuclear I&C system context. Multiple factors contribute to this assessment, including outdated and unavailable documentation, discontinued development of the software tool, unreliable performance and language inconsistent with the STPA Handbook.

Table 19: A table showing the results of the evaluation for XSTAMPP.

Beginning of Table 19		
Category	**Requirement**	**Satisfies Requirement**
Traceability:	Visualizing result traceability.	No
Traceability:	Visualizing Control Structure to result traceability.	No
Traceability:	Support for Control Structure to result traceability in general.	Yes
Traceability:	Filtering results by traceability.	Partially
Traceability:	Traceability between different analysis/levels of abstraction.	No
Control Structure:	Creating and modifying the Control Structure.	Yes
Control Structure:	Viewing the Control Structure at different levels of abstraction.	Partially
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	Partially
Control Structure:	Color-coding the Control Structure.	Yes
Guidance:	Language consistent with the STPA Handbook.	No
Guidance:	4 guidewords for UCAs.	Yes
Guidance:	Guiding questions for Loss Scenarios.	No
Guidance:	Enforcing correct syntax for results.	Yes
Guidance:	Documenting, viewing, and linking related material.	No
Guidance:	Copying parts of a prior analysis.	No
Guidance:	Partial auto-generation of results (specifically UCAs important).	No
Guidance:	Views supporting each use case.	Partially
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Partially

Continuation of Table 19		
Guidance:	Warning of incorrectly formatted results.	No
Guidance:	Highlight results to be reviewed after analysis modification.	No
Integration:	Export results in formats accepted by other systems engineering tools.	Partially
Integration:	Support for RAAML.	No
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	Partially
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	No
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	No
Prioritization:	Prioritizing Loss Scenarios/results.	Partially
Prioritization:	Filtering/sorting results by priority.	No
Prioritization:	Visualizing results by priority.	No
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Partially
Other:	Support for 2018 STPA revision.	Partially
Other:	Support for adding notes/comments.	Yes
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Yes
End of Table 19		

4.7 Tool evaluation 6: astah System Safety

Astah System Safety is the commercial version of STAMP Workbench. The tool is intended to support modeling and safety assessment activies, and is available for both Windows and MacOS. For the purpose of this evaluation, the software tool with a trial license was installed on a M1 MacBook Air, running OS version 12.7.2. Installation was straightforward and within two hours of downloading the tool, the Steps 1 and 2 of the case study's STPA analysis had been replicated in the tool.

Overall, the software tool satisfied 12 requirements, partially satisfied 8 requirements, and did not satisfy 12 requirements of the 32 total requirements. Multiple requirements were not satisfied in some of the key areas, such as traceability and guidance, but a novel hyperlinking feature enabled the tool to satisfy many requirements other tools could not address as effectively. These requirements and the tool's performance against them is presented in Table 20.

One significant finding made during the evaluation is the tool's performance with regards to the traceability requirements. The tool had no traceability visualization features, nor was the traceability between all results implemented. Most notably, the UCAs are not connected to Hazards, and therefore, Losses, in any way. For example,

in order to filter results by Hazard, the user must make a mention of the Hazard in the UCA description, which can then be found by a matching keyword. However, the user is not guided in any way to document the Hazard into the descriptions of UCAs.

The tool performs well with regard to Control Structure requirements. It offers the ability to customize the Control Structure with colors and notation the user prefers, and also offers the ability to add sub-Control Structure diagrams within controllers, in order to iteratively add detail. It also allows the user to start from more abstract Control Structures to less abstract Control Structures to a certain extent. However, there are some limitations to the Control Structure features. For example, the sub-Control Structure diagrams do not produce Control Actions, and their interface with the main Control Structure diagram is not defined beyond the sub-Control Structure being contained within a controller in the main diagram. Therefore, the sub-Control Structures cannot decrease the level of abstraction for the entire analysis (e.g. results such as UCAs or Loss Scenarios), rather it can just add detail to the main Control Structure.

The guidance features offered by the tool present many weaknesses, but also some strengths of the tool. The tool enforces little constraints over the documentation of some results, such as UCAs and Loss Scenarios, yet it does offer guidance for creating Loss Scenarios by offering hint words and actively encouraging the user to consider the Causal Factors for Loss Scenarios. However, by not enforcing the documentation of Hazards with Loss Scenarios or UCAs, the performance of the tool's result filtering features are limited by the user's ability to remember to document results and type them correctly. One aspect which was not captured by the requirements, though related to guidance, is the availability of a user manual for the tool. Astah System Safety offers a reference manual with a tutorial on STPA in the tool and comprehensive guidance on the usage of the tool. The extent of the availability of such support material is beyond other tools evaluated in this thesis.

One key feature of the software tool is its ability to associate results to almost any other type of data, from files on the user's computer, to URLs, to other models in the software tool. Using the "hyperlink"-feature presented in Figure 10, the user can document and access items they want to associate with almost any result. For example, a user may want to associate a controller in the Control Structure with a certain datasheet or manufacturer website, or perhaps a document outlining the project management aspects related to the controller. By using the hyperlinking feature such items can be viewed whenever the controller is selected. There are some limitations to the hyperlinking feature, however. Most importantly, results cannot be associated with a specific Loss or Hazard using the hyperlinking feature, rather they can be associated to the table containing the Losses and Hazards. The hyperlinking feature could also present the associated items more prominently, as currently the items need to be found in a sub-menu of a menu that can be accessed through the result they are associated to. Reducing the steps required to access the associated items could prove useful in the second use case, cooperating on STPA with system experts, as diagrams and websites could be associated to results using the hyperlinking feature, enabling less time to be spent on searching for the suitable supplementary material during focus group meetings.

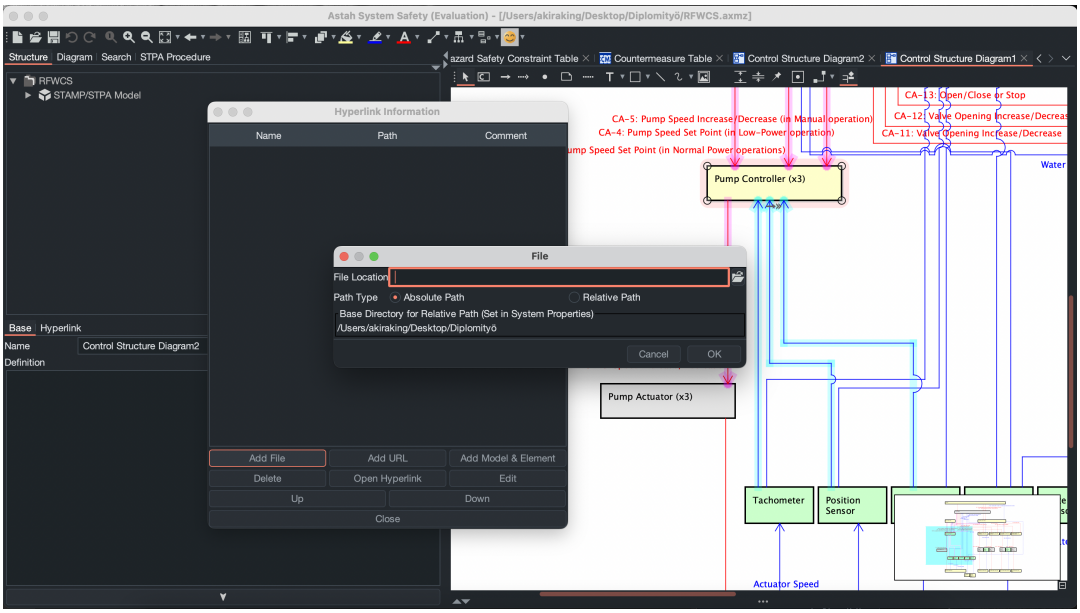

Figure 10: This screenshot of astah System Safety demonstrates the tool's hyperlinking feature being used for linking a file to the "Pump Controller (x3)"-controller in the Control Structure diagram.

A notable omission in the software tool are traceability visualization features. Both the visualization of traceability within the analysis and beyond the analysis, as well as the visualization of associations made using hyperlinks are not possible with the tool. Therefore a quick understanding of how results connect to the analysis as a whole is difficult to achieve. Fortunately the tool supports custom plugins, and several plugins were listed as freely available on the software tool's website. A plugin enabling the visualization of interdependencies between model elements was installed, however, possibly due to a lack of support for the STPA analysis features in the tool, the plugin's functionality could not be confirmed as a part of this evaluation.

During testing, the tool would occasionally suffer performance issues. For example, when editing the Control Structure the performance of the tool would decrease as the Control Structure became more detailed, resulting in noticeable input lag and several crashes of the tool. In order to mitigate this issue, the author had to save the project frequently to avoid having to having to repeat any work. Other performance issues were observed during the testing of PDF generation within the tool. A PDF of a Loss-Hazard-Safety Constraint table generated by the tool could not be adjusted to include any of the table contents, rather just the table header was present in the PDFs.

Overall, astah System Safety presents a thorough implementation of many useful features, and provides a solid base for conducting STPA analyses. However, the tool does have significant drawbacks in key areas regarding traceability, guidance, and the visualization of results and their traceability. Despite these weaknesses, the possibility and comprehensive support for expanding the tool's functionality through the use of plugins and the readily available documentation make the tool one of the more suitable tools currently available for conducting STPA analyses.

Table 20: A table showing the results of the evaluation for astah System Safety.

Beginning of Table 20		
Category	**Requirement**	**Satisfies Requirement**
Traceability:	Visualizing result traceability.	No
Traceability:	Visualizing Control Structure to result traceability.	No
Traceability:	Support for Control Structure to result traceability in general.	Yes
Traceability:	Filtering results by traceability.	Partially
Traceability:	Traceability between different analysis/levels of abstraction.	Partially
Control Structure:	Creating and modifying the Control Structure.	Yes
Control Structure:	Viewing the Control Structure at different levels of abstraction.	Yes
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	Partially
Control Structure:	Color-coding the Control Structure.	Yes
Guidance:	Language consistent with the STPA Handbook.	Yes
Guidance:	4 guidewords for UCAs.	Yes
Guidance:	Guiding questions for Loss Scenarios.	Yes
Guidance:	Enforcing correct syntax for results.	Partially
Guidance:	Documenting, viewing, and linking related material.	Yes
Guidance:	Copying parts of a prior analysis.	No
Guidance:	Partial auto-generation of results (specifically UCAs important).	No
Guidance:	Views supporting each use case.	Partially
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Partially
Guidance:	Warning of incorrectly formatted results.	No
Guidance:	Highlight results to be reviewed after analysis modification.	No
Integration:	Export results in formats accepted by other systems engineering tools.	Partially
Integration:	Support for RAAML.	No
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	Partially
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	No

Continuation of Table 20		
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	No
Prioritization:	Prioritizing Loss Scenarios/results.	No
Prioritization:	Filtering/sorting results by priority.	No
Prioritization:	Visualizing results by priority.	No
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Yes
Other:	Support for 2018 STPA revision.	Yes
Other:	Support for adding notes/comments.	Yes
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Yes
End of Table 20		

4.8 Tool evaluation overview

In total, 6 tools were evaluated in this thesis, of which 5 were evaluated against the requirements developed with STPA practitioners and experts. In addition to the tools presented in the evaluations many other tools were tested to varying extents to ascertain whether or not they were suitable to be further examined. These other tools include Depict, IBM Rhapsody, Moose, SafetyHAT, Stamp Workbench and STPA Automation tool. While these tools may be useful in their specific intended use cases, they were not selected for further analysis due to usability issues or a failure to meet key requirements outlined in Section 4.1. IBM Rhapsody was excluded due to its STPA features being currently under development, though parts of the STPA profile in the tool were available prior to the evaluation. The STPA support in IBM Rhapsody is being developed according to the RAAML specification. A comparison of the 5 tools evaluated against the requirements is presented in Table 21. CAIRIS Support is not represented in this table, as it was not evaluated against the requirements, rather only a subjective account of the evaluation process is provided in this thesis.

Table 21: A table presenting a comparison of software tool performance against requirements. Y = Satisfies requirement, P = Partially satisfies requirement, N = Does not satisfy requirement.

Beginning of Table 21					
Requirement	**STPA Viewpoint for Capella**	**Visual Pro SA**	**RMStudio**	**XSTAMPP**	**astah System Safety**
Traceability: Visualizing result traceability.	P	Y	P	N	N
Traceability: Visualizing Control Structure to result traceability.	P	Y	N	N	N
Traceability: Support for Control Structure to result traceability in general.	Y	Y	Y	Y	Y
Traceability: Filtering results by traceability.	N	Y	Y	P	P
Traceability: Traceability between different analysis/levels of abstraction.	N	N	Y	N	P
Control Structure: Creating and modifying the Control Structure.	Y	Y	Y	Y	Y
Control Structure: Viewing the Control Structure at different levels of abstraction.	N	P	P	P	Y
Control Structure: Iteratively reducing the level of abstraction of the Control Structure.	P	N	P	P	P
Control Structure: Color-coding the Control Structure.	Y	Y	Y	Y	Y
Guidance: Language consistent with the STPA Handbook.	Y	Y	Y	N	Y
Guidance: 4 guidewords for UCAs.	Y	Y	Y	Y	Y
Guidance: Guiding questions for Loss Scenarios.	P	Y	P	N	Y
Guidance: Enforcing correct syntax for results.	Y	N	P	Y	P
Guidance: Documenting, viewing, and linking related material.	P	P	P	N	Y
Guidance: Copying parts of a prior analysis.	P	N	N	N	N
Guidance: Partial auto-generation of results (specifically UCAs important).	N	P	Y	N	N
Guidance: Views supporting each use case.	P	Y	P	P	P
Guidance: Convenient indication of analysis work that is complete or needs to be explored.	Y	Y	Y	P	P
Guidance: Warning of incorrectly formatted results.	N	N	N	N	N

Continuation of Table 21					
Guidance: Highlight results to be reviewed after analysis modification.	P	P	Y	N	N
Integration: Export results in formats accepted by other systems engineering tools.	P	P	P	P	P
Integration: Support for RAAML.	N	N	N	N	N
Integration: Traceability between decisions in system design/implementation and STPA analysis results.	Y	P	N	P	P
Integration: Visualizing traceability between decisions in system design/implementation and STPA analysis results.	P	P	N	N	N
Integration: Highlighting items that require re-analysis after changes in related system design/implementation.	P	P	N	N	N
Prioritization: Prioritizing Loss Scenarios/results.	N	P	N	P	N
Prioritization: Filtering/sorting results by priority.	N	P	N	N	N
Prioritization: Visualizing results by priority.	N	N	N	N	N
Prioritization: Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	P	P	P	P	Y
Other: Support for 2018 STPA revision.	Y	Y	Y	P	Y
Other: Support for adding notes/comments.	Y	Y	Y	Y	Y
Other: Hostable by nuclear power plant licensee (rather than 3rd party cloud).	Y	Y	Y	Y	Y
End of Table 21					

The comparison of results presented in Table 21 reveals the common strengths and weaknesses of the software tools. The tools often satisfied or partially satisfied most of the Traceability and Control Structure requirements. However, requirements regarding Guidance, Integration and Prioritization were less often satisfied by the tools. These findings align with the discussions held with STPA practitioners and experts during the focus group meetings; traceability and the Control Structure are the key aspects of an STPA analysis. Each tool is focused on implementing both aspects of the STPA analysis with a unique approach. For example, by being integrated into a larger MBSE tool, STPA Viewpoint for Capella was able to provide traceability between the analysis and decisions related to the analysis results.

Requirements in the Guidance category often reflected thoughts STPA practitioners presented in the focus group meetings of what kinds of features would be helpful but not strictly necessary, such as features concerning result auto-generation and viewing analysis progress. Integration requirements were discussed more briefly in the focus group meetings, while Prioritization requirements generated significant discussion on preferred approaches to prioritization, and whether prioritization should be focused on, with the conversations emphasizing both sides of the latter point. These three categories also demonstrate significant differences in the performance of the software tools, with some requirements only being satisfied by one tool. It would appear that the developers of the software tools are similarly divided on which specific features are most worthwhile to implement.

While some tools are definitely presented significantly stronger in this evaluation, all evaluated tools had features aligning with the requirements that were innovative and unique to the tool. Examples of these features include the hyperlinking feature in astah System Safety, traceability visualization features in Visual Pro SA, UCA auto-generation in RMStudio, and traceability beyond the analysis in STPA Viewpoint for Capella. RMStudio and CAIRIS Support are both accessible through a web browser, allowing users to conduct analyses on the device of their choice. Given a different evaluation process, the usefulness of these features for STPA analyses in the nuclear domain could be verified. For example, conducting an analysis with a software tool rather than replicating the analysis data would reveal the usefulness of hyperlinking documentation to results, similarly, a case study exploring the prioritization of analysis results by traceability could assess the usefulness of Visual Pro SA's traceability visualization features.

One significant aspect of the software tools which is not reflected in the requirements is the availability of guiding material for the software tool itself. Tutorials, user manuals, videos, and built-in help menus are all useful in decreasing the learning curve for new users. In many cases, documentation was scarce, and this issue was amplified by an unintuitive user interface, or specific aspects of the user interface, which were hard to understand. For example, with tools such as XSTAMPP and CAIRIS Support, the evaluation process was hindered by a specific task of which the approach to undertaking in the tool was not apparent through the user interface. Examples of these include creating Loss Scenarios in XSTAMPP or the Control Structure in CAIRIS Support. In contrast, the guiding materials for some tools were not viewed at all during the initial phases of their evaluation processes, due to the tools being intuitive to use and aligning well with the STPA method described in the STPA Handbook [3]. This was the case in Visual Pro SA, RMStudio and astah System Safety.

While the requirements determined in this thesis attempt to capture the functionalities required for an STPA software tool for use in the context of nuclear I&C systems, over the course of the evaluation process it has become apparent that many more aspects of the software tools define their suitability or lack thereof. For example, some software tools could integrate the required features intuitively into the software tool, while other tools often satisfied requirements by means of a technicality, such as another feature being possible to use to satisfy a requirement. For example, many of the filtering and grouping requirements were often satisfied by a search feature in the software tools. In most cases, the search feature would seem to be intended for finding a specific result in order to edit or view it. However, given that the keywords would often narrow the results to those related to, for example, a specific Hazard, they would satisfy the related requirements. The goal of such requirements, however, is to support a deeper understanding of the analysis as a whole, and such search features would often fall short of reaching this goal, due to how the results were presented.

Overall, while each tool has its unique strengths described in their respective evaluation sections, only VisualPro SA can be recommended for a further assessment of suitability for the Finnish nuclear domain. The tool provided a robust implementation of the features at the foundation of an STPA software tool: traceability and Control Structure features. Additionally the tool has some useful features in the guidance

and integration categories, and performs well in facilitating the three different use cases, with regards to which the requirements were generated. Support for integration with Application Lifecycle Management tools also had an impact on this assessment, though not captured entirely in the requirements. While VisualPro SA performs well in multiple aspects, it does lack in some key areas, namely in better support for an iterative analysis process. However, considering the active development status of the tool, it remains possible for these issues in the tool to be addressed in the future.

5 Discussion and Conclusions

This section concludes the findings presented in this thesis, briefly discusses the threats to the validity of this work, and offers a brief discussion on STPA in general, mostly concerning the scope of STPA analyses and how STPA could be adopted beyond analyzing the safety of safety-critical systems.

5.1 Threats to validity

While the results presented in this thesis have provided an important insight into what constitute a suitable STPA software tool and the suitability of available software tools, the work presented here does have some significant limitations:

- The number of participants in the focus groups was limited.
- Focus group participants mostly consisted of those conducting STPA analyses, while other stakeholders could have been taken into account as well.
- The evaluation work was conducted by one evaluator.
- The requirements do not capture usability issues, for example the ease of use of the user interface.
- Case study data was replicated in the software tools, rather than the software tools being used to conduct a novel STPA analysis.
- Some requirements were challenging with regards to maintaining consistent evaluations across software tools.
- The software tools were evaluated on limited hardware, i.e. they may have performed better on a different operating system or computer.

Many of these limitations were known at the outset of this work, and were a result of limited resources. For example, the number of STPA practitioners in the nuclear industry in Finland are limited, and hence the number of participants in the focus groups was also similarly low. Conducting an STPA analysis with each software tool for evaluation purposes would have been challenging with regards to time and available cases to analyse. Similarly, due to not being required by STUK, applying STPA in the nuclear industry in Finland is fairly limited, resulting in ambiguity in how the method should be utilized. This issue was emphasized when considering how software tools should satisfy the third use case "Utilizing STPA results". Many tools offer the possibility to generate reports or provide overviews of the analysis, however whether or not these reports or overviews provide any value those utilizing STPA in the nuclear industry remains unclear.

Other limitations became apparent during the evaluation process, such as the challenges with regards to some requirements being more difficult to maintain a consistent assessment against, or the limited hardware the tools were tested on.

The latter would become apparent as some tools would perform significantly below expectations or were not able to be installed properly.

Many of the limitations of this work have, however, been at least partially addressed. For example, the limited size of the focus groups was mitigated to a degree with the extensive cooperation with an STPA expert from the research domain, who aided in the both the generation and finalization of the software tool requirements. This cooperation enabled the requirements to capture important aspects of conducting an STPA analysis, such as the various use cases, and how they are related to the features the software tools should provide. Similarly, the consistency of evaluations with regard to some specific requirements across software tools was address by cross-checking the evaluation results of each software tool and the reasoning forming the basis of each evaluation. The reasoning for each assessment against a requirement is also presented in Appendix A to further mitigate the limitations regarding the consistency of evaluations and the evaluation work being conducted by a single evaluator.

One use case which was not captured in the requirements, is that of multiple STPA facilitators or people working on the same STPA analysis using the same software tool. That is to say, collaborating with other users in the software tools was largely not considered in these requirements, although some requirements were created with considerations towards, for example, communicating changes to other users. As larger, more complex systems are analyzed and in increasing detail, multiple people may be needed to conduct the analysis. This realization was made after the requirements had been finalized and the evaluation process had already started, leaving collaborative work on STPA to be unaddressed in the requirements and evaluations.

All in all, the reproducibility of the requirements and evaluation conducted in this work is likely good. Though future work is unlikely to arrive at the exact same results, the broader whole of the work is likely highly reproducible. For example, while future work arriving at the exact same requirements is highly unlikely, the facets of traceability, the Control Structure, Guidance, and Integration, as well as their contents in general are likely to be reproduced given a similar requirements engineering approach. While slight differences in the evaluation of software tools are likely, especially given that their performance may vary depending on the device they are tested with, findings are likely to share many commonalities with those presented in this thesis.

5.2 Brief thoughts on STPA and future work

Throughout the process of conducting the work presented in this thesis, the topic of current issues with STPA and STPA software tools re-emerged frequently. The increasingly labor-intensive nature of the analysis as the level of detail captured by the analysis is increased was one aspect of the STPA analysis that was discussed on several occasions. While the STPA Handbook [3] states that the STPA method can be utilized for the analysis of systems with regards to other emergent properties, the labor-intensive nature of the analysis is likely one of the obstacles preventing the adoption of STPA in use cases other than analyzing the safety of safety-critical systems.

As can be seen in the results presented in this thesis, software tools are indeed one way of addressing these issues; software tools can provide a way of reducing repetitive work, aid in prioritizing analysis efforts towards more critical parts of a system, and potentially enable smoother co-operation between system experts and STPA facilitators. While all these benefits address the labor-intensive nature of the analysis, what other methods could be used to alleviate this issue?

One of the most significant factors impacting the time spent on conducting an STPA analysis, is the level of abstraction of the Control Structure of the analysis. While the level of abstraction of the Control Structure may not necessarily be compromised on when analyzing the safety of safety-critical systems, further work could investigate whether a higher level of abstraction in the Control Structures of analyses suffices for other emergent properties or systems in other domains that are not safety-critical. Improving knowledge on the required level of abstraction in each analysis use case could help avoid analyses being conducted with excessive levels of detail considering the goals of the analyses and available resources.

Demonstrating the feasibility of applying STPA in domains beyond those concerned with safety-critical systems could enable larger scale adoption of the method, hence increasing the amount of research conducted on the method. Similarly, larger scale adoption of the method could incentivize software tool developers in developing more comprehensive software tool support for STPA. Use cases in which STPA could potentially be utilized range from service design to addressing issues in human systems, such as large organizations, in addition to the analysis of highly technical systems such as those presented in the case study utilized in this thesis.

Future work should focus on exploring and comparing levels of abstraction in the analyses and alternative use cases in terms of both emergent properties and the types of systems being analyzed. With an improved understanding of STPA in these alternative contexts, improvements could potentially be seen in the application of STPA in the analysis of safety-critical systems. Additionally, through larger scale adoption, software tools and their application are likely to become more sophisticated allowing for the more effective utilization of the STPA method.

One aspect of STPA that was less explored in this thesis was that of how STPA should be used post analysis. While it was identified as an important factor, STPA has not yet been largely adopted in the Finnish nuclear domain, which leads to difficulties in drawing conclusions on this matter. However, how the analysis is utilized defines much of the value the analysis can provide. For example, by conveniently yielding suggestions on how to improve system design the analysis could provide more value. Providing this additional value to an STPA analysis could potentially be achieved using Artificial Intelligence (AI). While a very recent journal article explored the role of AI in STPA, the work mostly investigated ways in which Large Language Models (LLM) could expedite the analysis process, by for example generating results [37]. One aspect which was discussed less in the article was ways in which AI could aid in the utilization of analysis results. Could a complete analysis be given to a LLM, and realistic, useful suggestions be generated by the LLM in return? Future work could provide much needed answers in this regard.

5.3 Conclusions

This thesis has presented the requirements for STPA software tools for the analysis of nuclear I&C systems, and the evaluation of available STPA software tools against these requirements. Each of these results alone could have justified their own respective Master's theses, as the work could have been conducted with a much more detailed and expansive approach. As highlighted in the Discussions section, the small sample size of STPA practitioners and experts limits the reproducibility of the requirements produced in this work. Similarly, the evaluation of software tools could be more extensively undertaken by conducting an analysis with each software tool rather than replicating analysis results in the software tools. These analyses could be conducted by STPA practitioners, and their experiences could be compared to form an assessment of each software tool.

The approaches to the work conducted in this thesis reflect the amount of available resources, especially time and the availability of STPA practitioners. The results produced in this work are, however, also convincing in several aspects. During the requirements engineering process, notable omissions, such as Control Structure specific requirements, and needlessly specific requirements, such as those concerning Prioritization were quickly addressed by the participants of the focus group meetings. Participants were open and straightforward in communicating their ideas, thoughts and criticisms of the preliminary requirements, and thorough revisions were later conducted together with an STPA expert. Similarly, the quantitative results of the software tool evaluations generally correlated well with the qualitative results, i.e. how the tools satisfied the requirements reflected the author's descriptions of the user experience of each software tool.

Significant results from the requirements engineering process undertaken in this thesis include the requirements outlining the necessary functionality of the software tool's traceability and Control Structure features. The requirements defined in these categories often form the basis of higher level, "nice-to-have" features. For example, the auto-generation of UCAs requires traceability between the Control Structure and Control Actions, as the software tool needs information of both to auto-generate any meaningful amount of the UCA description. As highlighted by the participants of the focus group sessions, traceability is also important for result prioritization, and being able to highlight specific results and their traceability could be the key to utilizing such prioritization. An example of this is highlighting all CAs or UCAs related to a Hazard perceived as the most critical.

While requirements concerning the traceability and Control Structure features of the software tool are key findings, they do not diminish the value of the other requirements presented in this thesis. Requirements in the Guidance category capture ways in which the software tools can provide significant value to the user, and requirements in the Integration and Prioritization categories emphasize how a tool can support the utilization of STPA results for decisions concerning, for example, system design and implementation.

The results of the evaluation process emphasized the need for detailed and approachable documentation. Oftentimes, the documentation for a software tool was

either very limited, no longer available, or left critical parts of the analysis process with the tool unexplained. Depending on the tool, up to 30 hours were spent attempting to understand how to input a specific part of the case study data into the tool.

Another key finding of the evaluation process is how functional requirements do not capture all the relevant aspects of a software tool. While the tools which performed the best generally also satisfied the most requirements, the practical usability and performance of the software tools could have been better captured in the requirements. In many instances, requirements were satisfied or partially satisfied by features in the software tools, which were likely not intended to satisfy the intentions the requirements attempted to capture. In these cases, their usability was often poor when attempting to utilize the features as specified in the requirements. While this may point to a lack of specificity in the requirements, a fine balance had to be considered between specificity and enabling creative approaches. This is especially important in the case of this thesis, as the requirements are used to evaluate available tools, not in order to develop a software tool.

While most tools either satisfied or partially satisfied up to a third of the requirements determined in this work, only one tool can be recommended for a further assessment of suitability in the context of STPA analyses in the Finnish nuclear domain, VisualPro SA. With the most requirements satisfied of the tools evaluated, an intuitive user experience and robust traceability features, as well as the possibility for integration with common Application Lifecycle Management tools through an API, VisualPro SA appears to be the most suitable option of the available software tools. However, the tool would be further improved by applying the novel ideas presented in other software tools, most importantly by improving the support for iteratively working on STPA analyses.

References

[1] WNA, "I&C modernization: Current status and dfficulties," World Nuclear Association, Tech. Rep., 2020.

[2] IAEA, *Management of Ageing and Obsolescence of Instrumentation and Control Systems and Equipment in Nuclear Power Plants and Related Facilities Through Modernization*. International Atomic Energy Agency, 2022, no. NR-T-3.34.

[3] N. G. Leveson and J. P. Thomas, "STPA handbook," *Cambridge, MA, USA*, 2018.

[4] Y. Zhang, C. Dong, W. Guo, J. Dai, and Z. Zhao, "Systems theoretic accident model and process (STAMP): A literature review," *Safety Science*, vol. 152, p. 105596, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753521004367

[5] N. P. de Souza, C. de Azevedo Castro César, J. de Melo Bezerra, and C. M. Hirata, "Extending STPA with STRIDE to identify cybersecurity loss scenarios," *Journal of Information Security and Applications*, vol. 55, p. 102620, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212620307857

[6] S. I. Ahn, R. E. Kurt, and O. Turan, "The hybrid method combined STPA and SLIM to assess the reliability of the human interaction system to the emergency shutdown system of LNG ship-to-ship bunkering," *Ocean Engineering*, vol. 265, p. 112643, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822019266

[7] EPRI, "Hazard analysis methods for digital instrumentation and control systems," Electric Power Research Institute, Tech. Rep., 2013.

[8] J. Berger, R. Tiusanen, H. Kothalawala, and A. Pakonen, "Applying priority-informed STPA to a nuclear I&C system," in *Proc. ETFA*, 2024, submitted for publication.

[9] M. Rejzek and C. Hilbes, "Use of STPA as a diverse analysis method for optimization and design verification of digital instrumentation and control systems in nuclear power plants," *Nuclear Engineering and Design*, vol. 331, pp. 125–135, 2018.

[10] N. Ludvigsen, "Prototyping a digital support tool for an agile implementation of STPA," Master's thesis, NTNU, 2018.

[11] H. Kothalawala, "Application of system-theoretic process analysis (STPA) in nuclear instrumentation and control systems," Master's thesis, Aalto University School of Electrical Engineering, 2023.

[12] N. Leveson, "A new accident model for engineering safer systems," *Safety Science*, vol. 42, no. 4, pp. 237–270, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092575350300047X

[13] J. Berger, "STPA Guide," VTT Technical Research Centre of Finland Ltd., VTT Research Report VTT-R-00848-23, 2024. [Online]. Available: https://cris.vtt.fi/en/publications/stpa-guide

[14] C. A. King, P. Ovsiannikova, and V. Vyatkin, "Assessing the Suitability of Software Tools for System-Theoretic Process Analysis of Nuclear Instrumentation and Control Systems," in *Proc. ETFA*, 2024, submitted for publication.

[15] M. E. France, "Engineering for humans: A new extension to STPA," Ph.D. dissertation, Massachusetts Institute of Technology, 2017.

[16] J. C. Knight, "Safety critical systems: challenges and directions," in *Proceedings of the 24th International Conference on Software Engineering*, ser. ICSE '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 547–550. [Online]. Available: https://doi.org/10.1145/581339.581406

[17] STT, "Vastaamo-uhrien juristi: Ihmisiä on päätynyt itsemurhaan tietomurron ja kiristyksen takia," *YLE Uutiset*, 2024. [Online]. Available: https://yle.fi/a/74-20077270

[18] IAEA, "Country Statistics - Finland," Available at: https://pris.iaea.org/PRIS/CountryStatistics/CountryDetails.aspx?current=FI Accessed: 20.8.2024, 2024.

[19] IAEA, "In operation & suspended operation reactors," Available at: https://pris.iaea.org/PRIS/WorldStatistics/OperationalReactorsByType.aspx Accessed: 20.8.2024, 2024.

[20] WNA, "Nuclear Power Reactors," Available at: https://world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors Accessed: 20.8.2024, 2024.

[21] IAEA, "Safety Standards," Available at: https://www.iaea.org/resources/safety-standards Accessed: 21.8.2024, 2024.

[22] TVO, "Nuclear power plant units Olkiluoto 1 and Olkiluoto 2," Available at: https://www.tvo.fi/uploads/File/nuclear-power-plant-units.pdf Accessed: 21.8.2024, 2008.

[23] A. van Lamsweerde, "Requirements engineering in the year 00: a research perspective," in *Proceedings of the 22nd International Conference on Software Engineering*, ser. ICSE '00. New York, NY, USA: Association for Computing Machinery, 2000, p. 5–19. [Online]. Available: https://doi.org/10.1145/337180.337184

[24] J. Goguen and C. Linde, “Techniques for requirements elicitation,” in *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, 1993, pp. 152–164.

[25] C. Pacheco, I. García, and M. Reyes, “Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques,” *IET Software*, vol. 12, no. 4, pp. 365–378, 2018.

[26] M. Glinz, “On non-functional requirements,” in *15th IEEE International Requirements Engineering Conference (RE 2007)*, 2007, pp. 21–26.

[27] E. Heikkilä, T. Malm, R. Tiusanen, and T. Ahonen, “Hazard analysis of an autonomous container handling system–a comparison of STPA and HAZOP methods,” *Scientific Journal of Gdynia Maritime University*, no. 125, pp. 25–39, 2023.

[28] OMG, “About the risk anaysis and assessment modeling language specification version 1.0,” Available at: https://www.omg.org/spec/RAAML/1.0/About-RAAML Accessed: 6.5.2024.

[29] R. Patriarca, M. Chatzimichailidou, N. Karanikas, and G. Di Gravio, “The past and present of System-Theoretic Accident Model And Processes (STAMP) and its associated techniques: A scoping review,” *Safety Science*, vol. 146, p. 105566, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753521004082

[30] S. S. Krauss, M. Rejzek, and C. Hilbes, “Tool qualification considerations for tools supporting STPA,” *Procedia Engineering*, vol. 128, pp. 15–24, 2015.

[31] F. G. Souza, D. P. Pereira, R. M. Pagliares, S. Nadjm-Tehrani, and C. M. Hirata, “WebSTAMP: A web application for STPA & STPA-Sec,” in *MATEC Web of Conferences*, vol. 273. EDP Sciences, 2019, p. 02010.

[32] M. Tsuji, T. Takai, K. Kakimoto, N. Ishihama, M. Katahira, and H. Iida, “Prioritizing scenarios based on STAMP/STPA using statistical model checking,” in *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2020, pp. 124–132.

[33] MIT Partnership for Systems Approaches to Safety and Security (PSASS), “STAMP tools,” Available at: https://psas.scripts.mit.edu/home/stamp-tools/ Accessed: 20.8.2024, 2024.

[34] A. Abdulkhaleq and S. Wagner, “XSTAMPP: An eXtensible STAMP Platform As Tool Support for Safety Engineering,” 2015. [Online]. Available: https://www.researchgate.net/publication/271510990_XSTAMPP_An_eXtensible_STAMP_Platform_As_Tool_Support_for_Safety_Engineering

[35] A. Abdulkhaleq, "XSTAMPP For Safety Engineering of Software Intensive Systems," Available at: https://web.archive.org/web/20170926212328/http://www.xstampp.de/index.html Accessed 28.8.2024, 2017.

[36] A. Abdulkhaleq, "Tutorial 5 how to draw the process model in stpa project," Available at: https://www.slideshare.net/slideshow/tutorial-5-how-to-draw-the-process-model-in-stpa-project/76937162 Accessed 28.8.2024, 2017.

[37] S. Charalampidou, A. Zeleskidis, and I. M. Dokas, "Hazard analysis in the era of AI: Assessing the usefulness of ChatGPT4 in STPA hazard analysis," *Safety Science*, vol. 178, p. 106608, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092575352400198X

A Evaluation results with reasoning

This appendix provides the evaluation tables for all software tool evaluations with the reasoning for the assessments of the tool's performance against each requirement. In many instances, the features provided in a software tool could be interpreted to satisfy a requirement, when the implementation of the feature was misaligned with the goals of the requirement. In these instances, the assessments of each software tool against the specific requirement were cross-checked against each other to ensure a consistent evaluation across the software tools. However, due to the nature of the evaluation process, most importantly the fact that it is conducted by one evaluator, the impact of personal biases cannot be overlooked. This appendix is provided in an attempt to make the evaluation process and results more transparent to the reader, especially in light of instances of ambiguously assessable features being encountered on several occasions during the evaluation process. Additionally, no amount of proofreading can remove all possibility for mistakes, and by documenting the reasoning for each assessment, errors in these assessments are also likely captured, allowing readers to better adjust for possible inaccuracies in the main body of the thesis.

Table A1: A table showing the results of the evaluation for STPA Viewpoint for Capella along with the reasoning for each assessment. Colours green, yellow, and red correspond with an assessment of either satisfied, partially satisfied, or not satisfied, respectively.

Beginning of Table A1		
Category	**Requirement**	**Reasoning**
Traceability:	Visualizing result traceability.	The tool can visualize traceability partially, from Loss to Safety Constraint, and from Loss Scenario to Control Action.
Traceability:	Visualizing Control Structure to result traceability.	The tool can visualize traceability from the Control Structure to the results through "contextual traceability diagrams", however, the traceability visualized this way is limited.
Traceability:	Support for Control Structure to result traceability in general.	The tool links the results to the Control Structure and its elements such as controllers, control actions, feedback, and processes.
Traceability:	Filtering results by traceability.	The tool is unable to filter results based on traceability to another result. It is able to visualize some of the traceability, but no filtering can be applied to what is visualized.
Traceability:	Traceability between different analysis/levels of abstraction.	Based on the author's understanding, STPA analyses could be linked through the "Capella system elements" field of each result, however this feature remains unvalidated.
Control Structure:	Creating and modifying the Control Structure.	The tool has a built-in Control Structure creating/editing/viewing feature.

Continuation of Table A1		
Control Structure:	Viewing the Control Structure at different levels of abstraction.	The tool does not support viewing the Control Structure at different levels of abstraction unless you create multiple hierarchical Control Structure diagrams of the same Control Structure.
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	The tool allows the creating detailed controller diagrams of controllers identified in the Control Structure.
Control Structure:	Color-coding the Control Structure.	The tool supports color coding the Control Structure.
Guidance:	Language consistent with the STPA Handbook.	The tool uses language consistent with the STPA method.
Guidance:	4 guidewords for UCAs.	The tool prompts the users to create an "Unsafe Control Action Table" in which the guidewords are given by default.
Guidance:	Guiding questions for Loss Scenarios.	Guiding questions are not given, however a "Causal Factor Diagram is created for each UCA, which then informs the user in the creation of Loss Scenarios.
Guidance:	Enforcing correct syntax for results.	The tool documents all the required items for each UCA, however, it does not prompt or enforce any rules on naming the UCA according to the STPA method, rather the user can assign it any name.
Guidance:	Documenting, viewing, and linking related material.	The tool allows importing documents such as PDFs into the project, however, the author was unable to link STPA results with specific documents.

Continuation of Table A1		
Guidance:	Copying parts of a prior analysis.	The tool support copying any of the results produced in the analysis, however their transferability to another STPA analysis remains unconfirmed.
Guidance:	Partial auto-generation of results (specifically UCAs important).	The tool has no auto-fill for STPA results, such as naming UCAs.
Guidance:	Views supporting each use case.	The tool does not have separate views for each use case, however reports in the form of static webpages can be generated using an optional plugin. The author was unable to confirm this functionality.
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	Items in the analysis can be flagged by their status (e.g. "to be reviewed")
Guidance:	Warning of incorrectly formatted results.	The tool does not warn the user of incorrectly formatted results.
Guidance:	Highlight results to be reviewed after analysis modification.	The tool cannot automatically highlight items that need to be reviewed, however, users can highlight items manually.
Integration:	Export results in formats accepted by other systems engineering tools.	The software tool can export data as images and CSV file format. However, the tool is also a MBSE tool in itself.
Integration:	Support for RAAML.	The tool does not support for RAAML .
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	The tool allows associating STPA results with countermeasures and system elements modeled in Capella.

Continuation of Table A1		
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	The author was unable to confirm this functionality personally, however, the tool's user guide demonstrates some visualization of system elements to the STPA analysis.
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	The tool allows users to manually flag analysis items.
Prioritization:	Prioritizing Loss Scenarios/results.	The tool supports neither condition.
Prioritization:	Filtering/sorting results by priority.	The tool does not support filtering/sorting results by priority.
Prioritization:	Visualizing results by priority.	The tool cannot visualize results in order of priority.
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	The software tool can associate causal factors, however they cannot be filtered/sorted/grouped by their causal factors.
Other:	Support for 2018 STPA revision.	The tool supports the 2018 revision.
Other:	Support for adding notes/comments.	The tool supports adding comments/notes.
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	The tool can be installed on the user's computer and doesn't require a third party cloud to function correctly.
End of Table A1		

Table A2: A table showing the results of the evaluation for VisualPro SA along with the reasoning for each assessment. Colours green, yellow, and red correspond with an assessment of either satisfied, partially satisfied, or not satisfied, respectively.

Beginning of Table A2		
Category	**Requirement**	**Reasoning**
Traceability:	Visualizing result traceability.	The tool can visualize traceability from Loss Scenario to Loss, via UCA, Hazards and Safety Constraints.
Traceability:	Visualizing Control Structure to result traceability.	When creating UCAs or Loss Scenarios the Control Structure is shown adjacent to them and is highlighted in parts to which the results relate to (e.g. specific control loops).
Traceability:	Support for Control Structure to result traceability in general.	The tool links the results to the Control Structure and its elements such as controllers, control actions, feedback, and processes.
Traceability:	Filtering results by traceability.	The tool is able to filter results based on traceability to another result and visualize these relations. For example, a UCA can be highlighted to visualize all it's Loss Scenarios, Hazards and Losses.
Traceability:	Traceability between different analysis/levels of abstraction.	The tool does not appear to support such functionality.
Control Structure:	Creating and modifying the Control Structure.	The tool has a built-in Control Structure creating/editing/viewing feature.
Control Structure:	Viewing the Control Structure at different levels of abstraction.	The tool does not support viewing the Control Structure at different levels of abstraction, but has a overview window of the Control Structure, which shows the part of the Control Structure which is related to the result you are currently viewing.

Continuation of Table A2		
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	The tool does not appear to be designed for iteratively working on the Control Structure.
Control Structure:	Color-coding the Control Structure.	The tool supports coloring the Control Structure.
Guidance:	Language consistent with the STPA Handbook.	The tool uses language consistent with the STPA method.
Guidance:	4 guidewords for UCAs.	The tool allows users to create UCAs under four different categories corresponding with the guidewords.
Guidance:	Guiding questions for Loss Scenarios.	Guiding questions/causal factors are given based on those from the 2018 Handbook or alternatively a source chosen by the user.
Guidance:	Enforcing correct syntax for results.	The tool does not guide the user toward any syntax of the result descriptions.
Guidance:	Documenting, viewing, and linking related material.	The tool supports attaching files to countermeasures.
Guidance:	Copying parts of a prior analysis.	The tool support copying parts of the Control Structure within the analysis, but copying from one analysis to another does not seem possible in the tool.
Guidance:	Partial auto-generation of results (specifically UCAs important).	The tool suggests words from the Control Structure when typing results such as Loss Scenarios or UCAs.
Guidance:	Views supporting each use case.	The tool has a dashboard page showing an overview of the analysis, potentially useful for stakeholders. The simultaneous view of the results and Control Structure should support cooperating with system experts. The tool can generate reports of the analysis in docx/powepoint format.

Continuation of Table A2		
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	The tool conveniently shows how many UCAs a CA has, or how many UCAs a Loss Scenario has in their respective creation views, giving an indication of less complete areas of the analysis.
Guidance:	Warning of incorrectly formatted results.	The tool does not warn the user of incorrectly formatted results.
Guidance:	Highlight results to be reviewed after analysis modification.	The tool cannot automatically highlight items that need to be reviewed, however, users can highlight items manually.
Integration:	Export results in formats accepted by other systems engineering tools.	The software tool can export data as images and SVG file-format, in addition to generated reports in docx, ppt, and xlsx formats. The tool supports integration with Application Lifecycle Management software such as Polarion, though this aspects was not tested in this work.
Integration:	Support for RAAML.	The tool does not support for RAAML .
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	The tool can document and trace countermeasures in which documentation can be attached, but they do not connect to larger models of the system directly.
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	The tool can visualize traceability from countermeasures to other analysis results.
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	The tool allows the user to select whether a countermeasure is unsolved or solved.

Continuation of Table A2		
Prioritization:	Prioritizing Loss Scenarios/results.	The tool supports associating priority values (1-5) to countermeasures.
Prioritization:	Filtering/sorting results by priority.	The tool supports filtering countermeasures by priority.
Prioritization:	Visualizing results by priority.	The tool cannot visualize results in order of priority.
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	The software tool can associate causal factors, however sorting/grouping Loss Scenarios by their causal factors is limited to Loss Scenarios stemming from one UCA at a time. Only one causal factors can be selected per Loss Scenario.
Other:	Support for 2018 STPA revision.	The tool supports the 2018 revision.
Other:	Support for adding notes/comments.	The tool supports adding details to results and documents the author of each results. Results can also be flagged.
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	The tool can be configured to access the licensee's own server and the tool itself is installed locally.
End of Table A2		

Table A3: A table showing the results of the evaluation for RMStudio along with the reasoning for each assessment. Colours green, yellow, and red correspond with an assessment of either satisfied, partially satisfied, or not satisfied, respectively.

Beginning of Table A3		
Category	**Requirement**	**Reasoning**
Traceability:	Visualizing result traceability.	The tool can visualize the traceability of Losses to Hazards, however the traceability of other results is not possible.
Traceability:	Visualizing Control Structure to result traceability.	No visualization of traceability between the Control Structure and results produced in the analysis.
Traceability:	Support for Control Structure to result traceability in general.	Results are linked to entities in the Control Structure, such as controllers or controlled processes.
Traceability:	Filtering results by traceability.	The tool can sort UCAs in a spreadsheet view such that they are ordered by their relation to a controller or process in the Control Structure. Loss scenarios can be filtered by keywords appearing in the Loss Scenario, for example a term used in its description.
Traceability:	Traceability between different analysis/levels of abstraction.	Sub-Control Structure diagrams can be created within the analysis such that the control actions and feedback identified in the sub diagram are listed with those in from the main diagram. Loss Scenarios can simultaneously be associated with UCAs generated from both the original Control Structure diagram and the sub diagram.
Control Structure:	Creating and modifying the Control Structure.	The tool has a built-in feature for creating, modifying and viewing Control Structures.

Continuation of Table A3		
Control Structure:	Viewing the Control Structure at different levels of abstraction.	The tool allows creating sub Control Structure diagrams, which could be used to model part of the system in more detail.
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	The tool supports creating new Control Structure diagrams under the original diagram. Though the new diagram doesn't inherit any information from its parent diagram, the possibility to associate UCAs generated from it to the same Hazards and Loss Scenarios partially enables working iteratively. The latter functionality works on the Web-based version, but due to a different approach in to loss scenario creation, it is not possible on the desktop version!
Control Structure:	Color-coding the Control Structure.	The software tool allows the user the adjust the color of each item in the Control Structure diagram.
Guidance:	Language consistent with the STPA Handbook.	The tool uses language consistent with the STPA method.
Guidance:	4 guidewords for UCAs.	The software tool has categories based on the guidewords for UCAs, within which the user creates each UCA.
Guidance:	Guiding questions for Loss Scenarios.	The tool does not provide guiding questions for Loss Scenarios, however it displays a "Control Loops" view which partially satisfies this need.

Continuation of Table A3		
Guidance:	Enforcing correct syntax for results.	The tool autogenerates part of the UCA description, which can be used as a guide to formulate UCAs. For other results, little guidance is given.
Guidance:	Documenting, viewing, and linking related material.	The tool supports adding documentation, however linking or associating results to documentation was not able to be confirmed as possible.
Guidance:	Copying parts of a prior analysis.	Limited support for copying items from one analysis to another.
Guidance:	Partial auto-generation of results (specifically UCAs important).	The tool generates part of the UCA description automatically; the source, type and control action of each UCA are pre-entered for each UCA.
Guidance:	Views supporting each use case.	The tool supports generating a report of the analysis.
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	The web version of the tool does not communicate progress, however the desktop version communicates un-assesssed items very clearly (un-labeled Control Structure feedback/control, missing UCAs, and missing Loss Scenarios).
Guidance:	Warning of incorrectly formatted results.	The tool does not warn the user of incorrectly formatted results.
Guidance:	Highlight results to be reviewed after analysis modification.	The progress views for the diagrams, UCAs and Loss Scenarios highlight items that need to be assessed. However, this feature does produce some erroneous messages.

Continuation of Table A3		
Integration:	Export results in formats accepted by other systems engineering tools.	The tool is able to export spreadsheets of results and .PNG/.SVG files of diagrams. How data would be transferred from this tool to another remains unclear.
Integration:	Support for RAAML.	The tool does not support RAAML.
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	Tracing decisions in design and implementation to analysis results is not supported by the tool.
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	The tool cannot visualize traceability beyond the scope of the analysis results.
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	The tool does not offer features based on traceability beyond the analysis.
Prioritization:	Prioritizing Loss Scenarios/results.	Associating priority values or results of other hazard analysis methods is not supported by the tool.
Prioritization:	Filtering/sorting results by priority.	The software tool does not support filtering/sorting results by priority.
Prioritization:	Visualizing results by priority.	The software tool cannot visualize results in order of priority.
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	The tool can document the approximate source of the Loss Scenario together with the Loss Scenario (Controller/Feedback/Both). The tool also has a "Control Loops"-view, which displays factors that could contribute to loss scenarios, however these factors cannot be associated to Loss Scenarios.
Other:	Support for 2018 STPA revision.	The software tool supports the 2018 revision of STPA.

Continuation of Table A3		
Other:	Support for adding notes/comments.	The software tool supports adding notes to UCAs, as well as documenting additional details to Losses, Hazards, Constraints, and Control Structure entities.
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	The tool supports hosting your own database for the tool.
End of Table A3		

Table A4: A table showing the results of the evaluation for XSTAMPP along with the reasoning for each assessment. Colours green, yellow, and red correspond with an assessment of either satisfied, partially satisfied, or not satisfied, respectively.

Beginning of Table A4		
Category	**Requirement**	**Reasoning**
Traceability:	Visualizing result traceability.	The tool cannot visualize traceability, rather it relies on presenting results in tables.
Traceability:	Visualizing Control Structure to result traceability.	No visualization of traceability between the Control Structure and results produced in the analysis.
Traceability:	Support for Control Structure to result traceability in general.	Results are linked to entities in the Control Structure. Control Actions are created in the HCS diagram.
Traceability:	Filtering results by traceability.	The tool can filter some analysis results by key words that appear in the descriptions of CAs, for example. However, this filtering cannot be applied directly to the traceability of results, rather it only scans the description. Hence the traceability must be documented in the descriptions of the results for this feature to function.
Traceability:	Traceability between different analysis/levels of abstraction.	The tool does not appear to support such functionality.
Control Structure:	Creating and modifying the Control Structure.	The tool has a built-in feature for creating, modifying and viewing Control Structures.
Control Structure:	Viewing the Control Structure at different levels of abstraction.	The tool allows adding process models, process model variables and process model values to refine the Control Structure. This provides two views, the view of the Control Structure without these items, and a view with these items.

Continuation of Table A4		
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	The tool supports adding some detail to the Control Structure to reduce the level of abstraction, namely adding the process models and related items to Control Structure entities.
Control Structure:	Color-coding the Control Structure.	The software tool allows the user the adjust the color of each item in the Control Structure diagram.
Guidance:	Language consistent with the STPA Handbook.	The tool follows the language used in the STPA method, however, the language is inconsistent with the method in key parts of the tool. For example, no mention of Loss Scenarios are made in neither the tool nor the Documentation it comes with, making learning the tool much harder.
Guidance:	4 guidewords for UCAs.	The software tool has categories based on the guidewords for UCAs, within which the user creates each UCA.
Guidance:	Guiding questions for Loss Scenarios.	The tool does not provide guiding questions for Loss Scenarios.
Guidance:	Enforcing correct syntax for results.	The tool encourages the user to document in all the necessary information per analysis result.
Guidance:	Documenting, viewing, and linking related material.	Adding documentation or linking to it does not appear to be possible in the tool.
Guidance:	Copying parts of a prior analysis.	Limited support for copying items from one analysis to another.

Continuation of Table A4		
Guidance:	Partial auto-generation of results (specifically UCAs important).	The author was unable to get the tool to autogenerate results, despite some mentions of such a feature in the GitHub page.
Guidance:	Views supporting each use case.	The tool supports generating a report of the analysis, and viewing progress of the analysis. However, the second use case does not appear to be too thoroughly considered.
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	The tool has a progress view, but in terms of what is left for the user to do and how the user should proceed in the analysis there is little help available in the tool.
Guidance:	Warning of incorrectly formatted results.	The tool does not warn of incorrectly formatted results.
Guidance:	Highlight results to be reviewed after analysis modification.	The software tool does not highlight the items that need to be reviewed after a change.
Integration:	Export results in formats accepted by other systems engineering tools.	The tool supports exporting data in multiple data formats such as csv and various image and document formats.
Integration:	Support for RAAML.	The tool does not support RAAML.
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	The tool documents Design Requirements and can link them to the analysis results, however they do not connect to larger systems engineering models directly.
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	The tool offers no visualization of traceability.
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	The tool is unable to highlight such changes, mostly due to not supporting such traceability beyond the design requirements.

Continuation of Table A4		
Prioritization:	Prioritizing Loss Scenarios/results.	The tool seems to support assigning severity to UCAs, but the author could not find the way to add this information to the UCAs or Loss Scenarios.
Prioritization:	Filtering/sorting results by priority.	The tool does not support filtering/sorting results by priority.
Prioritization:	Visualizing results by priority.	The tool is not able to visualize STPA result in order of priority.
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	The tool can asssociate causal factors to Loss Scenarios, however, filtering by the causal factors did not appear to function during this evaluation, as key words would not narrow down the results presented by the tool.
Other:	Support for 2018 STPA revision.	While the tool does support the 2018 revision of STPA, many of the features are inconsistent in labeling from the Handbook.
Other:	Support for adding notes/comments.	The software tool supports adding notes/comments to results.
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	The software tool can be installed on the users personal computer and is not hosted on an third party cloud.
End of Table A4		

Table A5: A table showing the results of the evaluation for astah System Safety along with the reasoning for each assessment. Colours green, yellow, and red correspond with an assessment of either satisfied, partially satisfied, or not satisfied, respectively.

Beginning of Table A5		
Category	**Requirement**	**Reasoning**
Traceability:	Visualizing result traceability.	The software tool cannot visualize traceability.
Traceability:	Visualizing Control Structure to result traceability.	The software tool cannot visualize traceability between the Control Structure and other results.
Traceability:	Support for Control Structure to result traceability in general.	The software tool generates CAs based on the Control Structure, and for example components. Right clicking a result within a table usually reveals a "Jump To Diagram"-option, which switches the view to the Control Structure and highlights the related item there.
Traceability:	Filtering results by traceability.	The software tool has a search feature, which can filter results by keyword. However, full traceability isn't built-in to the tool, rather the user must make sure they employ traceability when writing the descriptions for the results, so the correct results are shown when keywords are applied.
Traceability:	Traceability between different analysis/levels of abstraction.	The software tool allows creating sub-Control Structures for individual elements in the main Control Structure, and the tool is able to present a combined view of the main Control Structure with its sub-Control Structures. However, other results cannot be generated based on these Control Structures (UCAs, Loss Scenarios etc.)
Control Structure:	Creating and modifying the Control Structure.	The tool supports creating/modifying the Control Structure.

Continuation of Table A5		
Control Structure:	Viewing the Control Structure at different levels of abstraction.	The tool supports sub-Control Structures which can be viewed as a part of the main Control Structure if so wished. There is also a small view with an abstracted version of the Control Structure showing where in the Control Structure the user is currently viewing.
Control Structure:	Iteratively reducing the level of abstraction of the Control Structure.	The software tool supports iteratively reducing the level of abstraction through the sub-Control Structure diagrams. However, these sub-Control Structures do not connect to the rest of the analysis through, UCAs or Loss Scenarios, for example.
Control Structure:	Color-coding the Control Structure.	The Control Structure Diagram can be colored the the user liking in the tool.
Guidance:	Language consistent with the STPA Handbook.	The software tool uses language consistent with the method. However, in certain parts of the tool the results are documented differently than could be expected based on the Handbook. (Loss Scenarios are grouped by Hazardous Causal Factor which receive IDs, Scenarios under each Hazardous Causal Factor)
Guidance:	4 guidewords for UCAs.	The guidewords are provided.
Guidance:	Guiding questions for Loss Scenarios.	Guiding questions are provided in the tool.

Continuation of Table A5		
Guidance:	Enforcing correct syntax for results.	The tool does not guide the user to document the link to Hazards in UCAs or Loss Scenarios, though the user can include it in the description.
Guidance:	Documenting, viewing, and linking related material.	Most results can have a hyperlink added to them. This link can be configured to lead to a file, URL, or another model or element. Upon right clicking the result a menu is revealed, with a "Hyperlink" sub-menu. Hyperlinks are presented in the submenu and clicking the link opens the file, URL or directs the user to the element.
Guidance:	Copying parts of a prior analysis.	Beyond basic copy paste functionality within a diagram, there is no possiblity to copy parts of the analysis.
Guidance:	Partial auto-generation of results (specifically UCAs important).	No results can be automatically generated.
Guidance:	Views supporting each use case.	The tool mostly seems to be intended for the STPA alone use case, however the documentation hyperlinking, and fairly convenient Control Structure viewing features could make it useful for the second use case as well. However, no reports can be generated.
Guidance:	Convenient indication of analysis work that is complete or needs to be explored.	In the STPA Procedure menu on the left-hand side of the UI, parts which have been started turn green after an entry has been added to it. However, the tool cannot indicate any further detail.

Continuation of Table A5		
Guidance:	Warning of incorrectly formatted results.	The tool does not warn the user of missing details, or incorrectly formatted results. However the tables do have separate columns for most required fields.
Guidance:	Highlight results to be reviewed after analysis modification.	The tool cannot highlight items to be reviewed after a change has been made.
Integration:	Export results in formats accepted by other systems engineering tools.	The tool can export diagrams in various image formats including PNG and SVG, and tables to Excel-format. Results can be printed through the tool, and saved as PDFs, though some tables would show as empty in the PDFs, such as the Loss/Hazard table.
Integration:	Support for RAAML.	The tool does not support RAAML, though it is heavily built around SysML, with support for GSN which is also specified in RAAML. Whether or not the tool follows the same specification for GSN as is defined in RAAML is unclear to the author.
Integration:	Traceability between decisions in system design/implementation and STPA analysis results.	The tool can trace countermeasures to UCAs and their Hazardous Causal Factors, in addition countermeasures can be hyperlinked to other analysis elements, URLs or files. How these connect to implementations or specific components and decisions beyond the counter measures is unclear.

Continuation of Table A5		
Integration:	Visualizing traceability between decisions in system design/implementation and STPA analysis results.	Traceability cannot be visualized in the tool.
Integration:	Highlighting items that require re-analysis after changes in related system design/implementation.	The software tool cannot highlight changes.
Prioritization:	Prioritizing Loss Scenarios/results.	No specific fields in the results document priority.
Prioritization:	Filtering/sorting results by priority.	The tool does not support filtering/sorting by priority.
Prioritization:	Visualizing results by priority.	The tool cannot visualize results in order of priority.
Prioritization:	Associating Causal Factors to Loss Scenarios and filtering/grouping/sorting by them.	Causal Factors can be associated with Loss Scenarios and Loss Scenarios are grouped by their Causal Factors by default.
Other:	Support for 2018 STPA revision.	The tool supports the 2018 revision of STPA.
Other:	Support for adding notes/comments.	The tool supports comments in some places, and adding definitions in most places.
Other:	Hostable by nuclear power plant licensee (rather than 3rd party cloud).	The tool is installed locally on the user's computer.
End of Table A5		