Designing a Model-agnostic Cybersecurity and Functional Safety Risk Assessment Tool

Joel Gustafsson
joegu499@student.liu.se

Anton Hagel
antha627@student.liu.se

***Abstract*—Security and safety are two concerns within digital infrastructure that have thus far only been mapped out separately by tools that currently exist. Security revolves around the intended operation of software and hardware in accordance with the CIA attributes (Confidentiality, Integrity, and Availability). Safety, on the other hand, generally pertains to upholding the quality of life for the people depending on, operating, and using said systems. However, there are several sectors within critical infrastructure that need to consider both when conducting risk analysis. The purpose of this study is to propose a solution to the problems that arise when assessing risks associated with both cybersecurity and functional safety in parallel. Using the design science research method, we conducted the work in two cycles. The first cycle was focused on the gathering of information and forming a basis for the tool. The second was more focused on developing the actual tool. The output of the study results in the *HARA/TARA Tool*. It was designed to be flexible and lightweight. The flexibility comes from the user's ability to create methods with custom dimensions, parameters, and formulas. This allows for a diverse set of risk assessment methods to be crafted and used, and for functional safety to be integrated. The tool was evaluated by professionals, who thought the features showed promise. At the end of the study, the tool was a proof of concept in terms of its ability to incorporate functional safety, and will see further maintenance beyond the study.**

Svensk Sammanfattning - De verktyg som för närvarande existerar för att kartlägga säkerhets- och trygghetsaspekter inom digital infrastruktur har hittills behandlat dessa områden separat. Säkerhet avser systemens avsedda funktion i enlighet med CIA-principerna (Konfidentialitet, Integritet och Tillgänglighet). Trygghet relaterar däremot i högre grad till att upprätthålla livskvaliteten för de individer som är beroende av, använder eller opererar dessa system. Inom flera sektorer av den kritiska infrastrukturen föreligger dock ett behov av att beakta båda perspektiven vid genomförande av riskanalyser. Syftet med denna studie är att analysera tidigare lösningar för att identifiera hur de adresserat denna problematik, samt att tillämpa de insikter som genererats i utvecklingen av ett nytt verktyg. Studien genomfördes i två cykler: den första fokuserade på informationsinsamling och etablering av en konceptuell grund för verktyget, medan den andra ägnades åt utvecklingen av själva verktyget. Det resulterande HARA/TARA-verktyget utvecklades i Qt med hjälp av Qt-Creator-sviten och stöds av en databas med MySQL-motor. Resultaten av studien indikerar att befintliga industriverktyg har begränsad kapacitet att integrera funktionell säkerhet med cybersäkerhet. Inte heller existerande ramverk med sådan förmåga kunde på ett adekvat sätt representeras i tillgängliga industriverktyg. HARA/TARA-verktyget designades därför med hög flexibilitet, vilket möjliggör för användaren att skapa metoder med anpassningsbara dimensioner, parametrar och formler. Detta ger utrymme för att implementera metoder från avancerade ramverk och därigenom integrera funktionell säkerhet. Verktyget utvärderades av forskare från RISE, som bedömde att funktionaliteten uppvisade lovande potential. Även om verktyget inte var färdigställt vid studiens slut, är det väl dokumenterat för att underlätta fortsatt underhåll och vidareutveckling.

I. Introduction

A. Motivation

In the modern era, the susceptibility to cyberattacks grows larger and larger[1]. The digitalization of critical infrastructure has led our society to invest in more cybersecurity[2]. One example of such an area within critical infrastructure is the energy sector. As the internet and the need for electricity grow, the facilities producing and transporting it increasingly become targets for cyberattacks. The repercussions for interruptions in delivery also become larger and more severe. For most analyses of weaknesses in strictly Information Technology[3] (IT) based systems, considering only cyber threats is generally sufficient [1]. Critical infrastructure, however, presents a new challenge as there is a need to model their Operational Technology[4] (OT) components to account for the provision of vital resources and to guarantee the safety of its dependants. The neglect of functional safety can have severe consequences for critical infrastructure.

B. Aim

To tackle the rising threat to critical infrastructure, the aim of this study is to answer the demand, as expressed through the interests of RISE, for better cybersecurity within the domain. This must be done while not compromising functional safety in the systems. For this purpose, research into how other tools perform risk analysis is conducted. The findings are then examined to see if the tools consider functional safety. Any framework that supports both cybersecurity and functional safety analysis is included and presented in the study. A tool that answers these concerns is also developed during the study based on what is found to be desired for such a tool.

C. Research Questions

This study will revolve around these two questions:

[1]https://hbr.org/2023/04/cyber-risk-is-growing-heres-how-companies-can-keep-up

[2]https://www.ibm.com/think/insights/cybersecurity-dominates-concerns-c-suite-small-businesses-nation/

[3]https://csrc.nist.gov/glossary/term/information_technology

[4]https://csrc.nist.gov/glossary/term/operational_technology

RQ1 What support for risk analysis of cybersecurity and functional safety do modern risk assessment tools offer?
RQ2 How could a risk assessment tool be developed to address both cybersecurity and functional safety?

D. Approach

The scope of the study is limited due to time limitations within bachelor's thesis works. The study is also conducted in collaboration with the company RISE (explained more in Section II-A). In fact, it was RISE that came to us with the thesis topic. They also proposed to us the creation of an artifact based on our findings. Because of this, we concluded that design science research is the most appropriate method for a study of this type (explained more in chapter IV).

Since parts of the study are based on past research, we had to find an effective way of gathering research articles. To this end, we mainly used Linköping University's library search engine: UniSearch[5], with help from Google Scholar as it indexes more articles and is easier to work with. UniSearch is a search engine with a wide array of research articles stored in its database. This will make article retrieval faster and more practical than simply trying to find articles with more general-use search engines. The assumption is that Google Scholar has a stronger and broader engine, but could lead us to locked articles we need to access via UniSearch.

To be able to develop a User Interface (UI) that facilitates risk analysis of critical infrastructure, Qt version 6 will be used. Qt is a framework created in C++ that is specifically made for the development of UIs. Both authors of this thesis have experience with C++ and at least one of us has experience with Qt. This gives us reason to believe that working in this environment is efficient. C++ also makes the tool more lightweight and more portable, which is desired for a tool like this.

E. Limitations

Firstly, our choice of method creates some issues with regard to properly examining the current literature and the state of current solutions. This is because design science research generally concerns itself more with how things ought to be, as opposed to explaining how things are [2]. It is also a method that is more adapted to the collaboration between a research-conducting part and a company. This is why, although design decisions are based on research, the quality of our solutions cannot be determined in a more grounded scientific way. Instead, it is up to our judgement and the collaborating company to decide how the artifact is validated.

In the case of this thesis, RISE, specifically a few professionals working together with the CYREC project[6], assessed the results. This will limit how broad our conclusions can become, as we will only be able to evaluate how well we fit their particular use case. Design science research allows for more than one reliable approach to validation. However, due to time constraints, the thesis will stick to a verbal evaluation of the results from RISE. The time constraints limit the scope of this thesis.

Lastly, the time available for us to develop the artifact is limited and insufficient to create a production-ready application. All desired functionalities could therefore not be implemented in time.

II. Background

In this chapter, circumstances around the project and prerequisite knowledge are presented in parts. First, a short summary of the collaborating company of this study, RISE, is given. Afterwards, explanations of functional safety, cybersecurity, and standards related to both are given.

A. RISE

RISE Research Institutes of Sweden is a state-owned company that promotes the competitiveness of technical and sustainable solutions for academia, the public, and the private sector[7]. The company's stated mission is to "strive for sustainable growth in Sweden by strengthening the competitiveness and capacity for renewal of Swedish industry, as well as promoting the innovative development of society as a whole"[8]. They specialize in risk analysis of different kinds of transport, including automotive transport and the transport of electricity. The division, as does the entire company, consists of professionals working within a wide variety of fields related to cybersecurity and safety. Their expertise will be used during each part of the design science research method (this method is further explained in chapter IV).

B. Functional Safety and Cybersecurity

Cybersecurity has historically been focused on protecting assets that are valuable to the delivery and/or profitability of a service [1]. However, as OT systems become intertwined with IT, the need to account for cyber threats and mitigations becomes important. Any intervention from either one might harm the effectiveness of the OT system's ability to protect the public. Australian Signals Directorate's Australian Cyber Security Centre (ASD's ACSC) and collaborating agencies [1] concluded that any OT environment that is interconnected with IT requires a thorough understanding of both systems in order to successfully achieve both safety and security.

C. HARA, TARA, and ISO/SAE Standards

The ISO 26262 standard contains requirements for functional safety within electrical and electronic systems in road vehicles [3]. The standard provides metrics for the evaluation of risk in the form of Automotive Safety Integrity Levels (ASILs). The dimensions used in ASILs are severity, exposure, and controllability. Severity can be based on economic, physical, and property damage to individuals. Exposure is the probability of being in a hazardous event when a system fails, while controllability is the chance of avoiding the damages

[5]https://liu.se/en/article/biblioteket-online

[6]https://www.vinnova.se/en/p/cybersecurity-for-resilient-energy-communities-of-the-future/

[7]https://www.ri.se/en/about-rise/operations

[8]https://www.ri.se/en/about-rise/operations/mission-and-governance

from the event [4]. Dimensions are the categories of which calculations of risk are executed. Parameters are the metrics used to calculate the individual dimensions. The ISO 26262 standard is the basis of the *Hazard and Risk Assessment* (HARA) framework.

The ISO/SAE 21434 standard has similar aims as ISO 26262, but focuses on the cybersecurity of road vehicle systems instead of functional safety [3]. Threats are identified with the help of threat models, like the CIA model (Confidentiality, Integrity, and Availability) or the STRIDE model. ISO/SAE 21434 makes use of similar parameters for risk calculation as ISO 26262, but makes use of impact and likelihood in the calculation metrics instead of severity and exposure. Impact is based on two different categories: the damages to CIA (or other categories if other models are used), and the damages to business-related factors [5]. Business factors are usually confined to the following categories: financial damages, reputational damages, privacy violations, and non-compliance. The difference between likelihood and exposure is that likelihood is based on different parameters instead of being strictly statistical. In other words, instead of approximating a percentage value, the parameter is based on a scale (for example 0-4, low to severe). Likelihood parameters revolve around the capacity of threat agents and the severity of vulnerabilities in systems. Just like how HARA is based on ISO 26262, *Threat and Risk Assessment* (TARA) takes all of its guidelines from ISO/SAE 21434.

While both HARA and TARA provide a full lifecycle assessment of the electronics in road vehicles, neither of them is fully encompassing. HARA is focused solely on functional safety. TARA has aspects of both cybersecurity and functional safety, but does not meet the requirements of the ISO/SAE standards (more about this in Section III-A) [6]. This has led to cutting-edge studies being conducted around the design of new frameworks with both cybersecurity and functional safety in mind.

III. Related Work

In this chapter, relevant work related to the study is presented. Related work was studied during the problem investigation phase of cycle 1 (see Chapter IV).

A majority of the articles seeking to create methods to unify cyber security and functional safety focus on vehicular communications. This is still representative of critical infrastructure, as vehicular communications contain properties of both IT and OT. These methods and frameworks are compared in the first following section. In the section after, a short compilation of industry tools is displayed and compared to each other, and to risk assessment frameworks.

A. Comprehensive Risk Assessment Frameworks

A common thread in the design of the frameworks is the analysis of standards in comparison with existing risk assessment methods. The creation process of *TARA+AD* was based on the comparison of the ISO 21434 standard with existing frameworks [6]. Since the TARA+ and SARA methods exhibited the most amount of matching qualities with what is desired by the standard, they are used as the cornerstone of the framework design. Multiple articles take inspiration from the ISO 26262 standard for their framework design, which means that they are more based on HARA than TARA [4, 5]. That being said, both standards are used as inspiration in these frameworks, although the fundamental starting point is different [4, 5, 6]. All try to design comprehensive risk assessment methods to meet the requirements within not only systems in road vehicles, but critical infrastructure in general.

Three examples of such frameworks are *HEAVENS*, *TARA+AD* and *THARA* [4, 5, 6]. In Islam and Lautenbach [5] report about HEAVENS, Threat Level (TL), Impact Level (IL), and Security Level (SL) are described as different parameter categories that decide how the evaluation of risk is determined [5]. TL is the estimation of likelihood or probability, IL is the estimation of impact or exposure, and SL is the evaluation of the resulting product of TL and IL [3, 5]. TL, IL, and SL are in and of themselves dimensions, and are calculated by using their parameters. To determine the TL of a threat, HEAVENS makes use of four threat level parameters: expertise, knowledge about Target Of Evaluation (TOE), window of opportunity, and equipment [5]. IL parameters used in HEAVENS are the following four: safety, financial, operational, and privacy and legislation. These categories are similar to the ones used in the *ISO 21434* standard [6]. The categories safety and operational are explicitly part of the TARA attempt to integrate functional safety into cybersecurity risk analysis. At the time, only *ISO 26262* existed, so the attempt to integrate functional safety into cybersecurity predates *ISA/SAE 21434*.

TARA+AD makes use of similar terms and categories to *HEAVENS* [6]. Loskin [6] found in her study of the creation of the TARA+AD framework that methods created based on ISO standards do not necessarily follow them in a complete manner. In her study, she compares multiple methods with the *ISO/SAE 21434* cybersecurity standard. None of the frameworks, including HEAVENS, completely cover the requirements from the standard on their own. HEAVENS and SARA both need STRIFE to check all the requirement boxes. In the worst cases, for methods such as BRA and TARA Intel, half or more of the requirements are not covered. This is the motivation behind the cutting-edge framework TARA+AD. So the framework takes features from multiple different frameworks.

The result of this is a risk assessment method with similar categorization, only different in naming [6]. The real differences can be found in the parameters used for the dimensions, and in the calculations of the different levels. Instead of using the term Impact Level, the method splits IL into Impact Value (IV) and Impact Factor (IF). IV is the sum of the parameter values, while IF is an outputted value between 0-4. Both IV and IF are calculated via defined formulas that originate from the TARA+ method. The impact parameters used for the TARA+AD method are the same that are used for HEAVENS: safety, financial, operational, and privacy. However, in this case the formula used to sum up the values into IV weighs

the safety and operational dimensions higher than financial and privacy. This means that a higher value in the safety or operational parameters will lead to more severe results than higher values in the financial and privacy parameters. The version of threat level in this method sees a similar subdivision to that of impact level. Attack Potential (APo) is the sum of the parameter values, and Attack Probability (AP) is a calculated value between 0-4 via a formula. The threat level parameters used are elapsed time, expertise of attacker, knowledge of the asset, window of opportunity, and equipment used for attack. In the end, IF and AP are used together as an index to determine the security level, or risk value, in a risk value ranking table. The table is color-coded to determine how high the risk value is. The risk severity values are Quality Management (QM), low, medium, high, and critical.

THARA is a framework that attempts to integrate TARA considerations into HARA analysis, and vice versa [4]. For both cases, item and asset identification of hazards and threats is done in parallel. Then it depends on whether TARA is being integrated into HARA or HARA into TARA. In the former case, a comparison of items from HARA with assets from TARA is performed to find potential mappings of items and assets. If there are mappings, the attack vectors from TARA are used to calculate the risk levels. After hazard analysis has been performed on the items, the HARA parameters severity, exposure, and controllability are computed. If there were items associated with an asset-risk, the calculated risk levels will be used in the computation of controllability. There is a similar method when integrating HARA into TARA. If there are mappings of assets to items, the items' hazard scenarios are assessed and ASIL gradings are calculated. These gradings are then later used in the computation of safety severity for the associated asset. This framework still requires two independent risk assessments, but it makes a more concrete effort in integrating them than regular risk assessment methods. HEAVENS and TARA+AD both have better integration of cybersecurity and functional safety still, as relevant parameters and formulas for both are explicit parts of the methods [5, 6]. Even so, the concept of using the output of one method as the input of another for a more comprehensive risk analysis could be of use in future tools.

B. Industry Tools

We have selected a number of solutions identified by RISE to investigate more thoroughly. We observed that most of the tools focused mainly or wholly on cybersecurity with little to no functionality for using frameworks, other than the likes of STRIDE. Examples include: Microsoft Threat Modeling Tool[9], OWASP Threat Dragon[10], Threagile[11], IriusRisk[12] and PYTM[13]. All tools examined generate a graphical data flow diagram (*DFD*) with some being able to automatically assign risks based on internal rules. Most tools employ a kind of graphical interface for defining the DFD, while others, such as PYTM, rely on text input to define the assets and links that make them up. The interface usually takes the shape of a "drag and drop" canvas where a number of asset types are predefined. The tools can broadly be categorized as text-based or graphical, with varying methods for inputting risks and, when relevant, assets. Some tools have functionality for exporting the identified risks as a PDF report or even as a presentation. These kinds of features seem more common than the ability to customize the framework with which to perform the analysis itself. Of the tools we looked closer at, only IriusRisk has some support for multi-user concurrent usage. All other tools are strictly single-user applications. The quality of the multi-user experience is, however, out of the scope of this thesis for us to determine since it is paid software. IriusRisk also stands out in that it supports some AI integration for simplifying the definition of risks and assets. Again, this feature is difficult for us to assess fully because of the monetary requirements.

[9]https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool

[10]https://owasp.org/www-project-threat-dragon/

[11]https://threagile.io/

[12]https://www.iriusrisk.com/

[13]https://github.com/OWASP/pytm

IV. Design Science Research

The study is conducted according to the principles of design science research. Design science research is a qualitative method performed in iterations, or, as referred to in this thesis: cycles [7]. Each cycle follows five steps in consecutive order: problem investigation, solution design, design validation, implementation, and evaluation. In the case of this study, only two cycles were performed. The first cycle will serve to answer **RQ1**, while the second cycle gives answers to **RQ2**. This type of research fits well when the problem is of practical nature [2, 8]. This is because the method revolves around the iterative process of creating a product, which aligns with the goal of this study. The information gathered for each cycle phase also contributes to the answer of the research questions. Action research is a similar method that could also be used for this project. In the end, design science research was chosen as the more fitting method, as it aligns better with our goals. Action research requires observation of the routines of a workplace, followed by a change to that routine [9]. This makes action research unsuitable for this study. We also found that the information acquired from cycle phases in design science research fits better than what would be learned from action cycles.

A. Problem Investigation

In the first phase, our job is to learn more about the problem given to us [8]. For the first cycle, a smaller literature review is conducted, as well as a study of what risk analysis tools exist today. The second cycle takes the wisdom from the cycle before, in addition to an even shorter literature review.

B. Solution Design and Design Validation

After the problem investigation phase, a cycle enters the solution design phase [8]. An overall plan of the entire project

must be mapped out before starting the project properly. It is in this phase where a specific design of the artifact is mapped out. ”Artifact” is what the resulting product from design science research is called. After a design has been constructed, the cycle enters the design validation phase. In this phase, both researchers and supervisors of the project look critically at the design and assess its quality based on three categories: internal validity, trade-offs, and external validity. Internal validity revolves around the accuracy of the design in regard to the identified problem. Trade-offs consider what different approaches could be taken, and if those would be better than how the design looks now. External validity is the flexibility of the design in different contexts. The analysis of these categories is only done privately with the purpose of facilitating the creation of the artifact, and thus will not be included in this thesis.

C. Implementation and Evaluation

When a solution design has been constructed and validated, the practical part of the cycle begins [8]. The design constructed in the last two phases is implemented in practice with the tool of choice for the project. For this project, the Qt suite is the tool used to develop the artifact. What functions the artifact will have depends on the solution design phase. When the artifact has been developed, the evaluation phase commences. Because of scope concerns, the evaluation of each cycle will be limited to demonstrations of the tool and discussions with RISE about the accuracy of the implementation.

V. The HARA/TARA Tool

The following section describes the artifact developed within this study. The first subsection describes the workflow when using the program. The second goes into the design of the database behind the tool. The third subsection goes into how the database was implemented via MySQL and stored procedures. The fourth subsection explains how and why the *lua* language was integrated within the app. The fifth subsection explains how the Qt implementations interact with the backend. Finally, the sixth subsection gives an overview of the project organization.

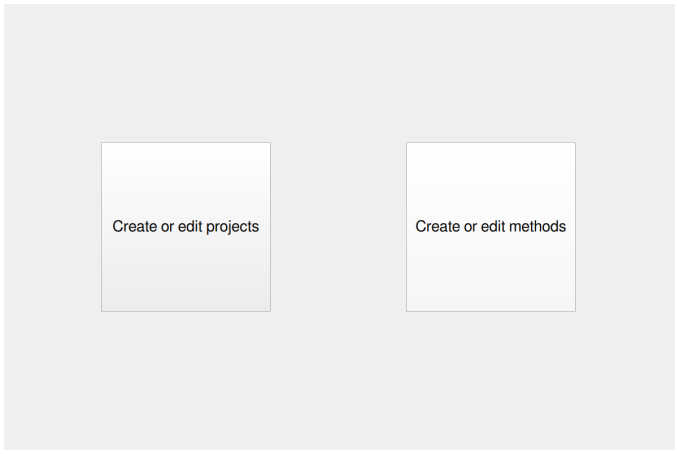

Fig. 1. Main page in created GUI. User chooses between creating/editing risk assessment methods, or creating/editing risk analysis projects.

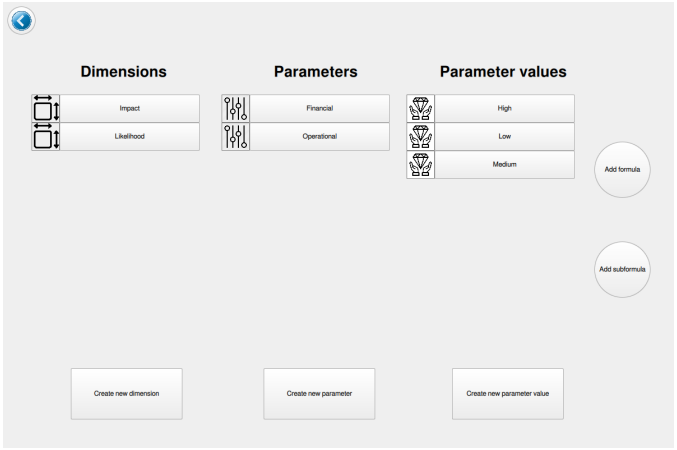

Fig. 2. Edit page for risk assessment method. Here the user can create dimensions, parameters, parameter values, formulas and subformulas for the method chosen.

A. Workflow Within the Application

A future plan is to make the tool available in two forms: as a script, and as a Graphical UI (GUI). At the time of the release of the study, only the GUI is implemented. The usage of the GUI is divided into two steps where the first takes one of three forms: Import, selection, or creation of a risk assessment method, and conducting risk assessments with a chosen method within a project. In the start page, the user gets to decide which of the two steps to enter, seen in figure 1. If the first step entails method creation: a user would begin

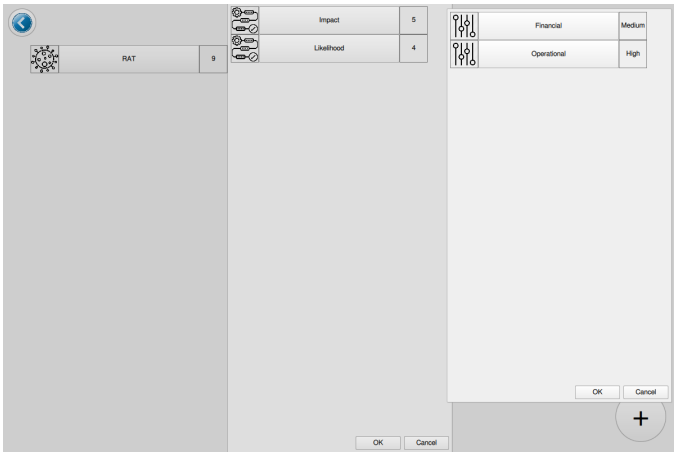

Fig. 3. Page for editing risks in a project. Risks are created with the plus button. When a risk is clicked, the middle window will show up with the method’s dimensions and the dimension levels set for the risk. To set a dimension level, the parameters must have a selected parameter value. The user can select the values of each parameter by clicking a dimension, which will open up the window to the far right.

by pressing the "Create or edit methods" button. This takes them to the list of currently created methods. Pressing the "plus" button leads to a popup window where the name of the method can be inputted. At this point, an empty method will have been created, which can now be clicked on in the list of available methods. Doing so would take the user to the "edit method" view, where the desired dimensions, parameters, parameter values, and formula can be added. This view can be seen in figure 2. Having created a method, the user can now move on to create a project based on this method. To create a project, the user must enter a project title, author name, and the risk assessment method used in the project. After that, the user can create and list several risks in the project by clicking it and opening the risk page (seen in figure 3). When the user is content with the risks, they can rate the dimension parameters for each risk. After this is done, the user can press "calculate security level". This makes the tool use the subformulas and formula to calculate the threat, impact, and security level for each dimension and risk. The result will be shown in a separate risk view. The concept of connecting countermeasures to risks in the risk view has backend but not frontend support.

B. HARA TARA Tool ER Model

During the first cycle of the project, an Entity Relationship (ER) diagram was constructed based on our findings (see section VI for said findings). After evaluations with RISE, a final ER diagram was used as the foundation for the database. The diagram can be seen in figure 4. All necessary objects to represent different methods and risk analysis projects can be seen in the diagram.

Risk assessment method represents the methods used to calculate risk. Some examples of methods that could be used are HARA, TARA, HEAVENS, and several more. Each method contains a name, a formula to calculate risk levels, and dimensions used in the formula. The *Dimensions* entity represents all the different dimensions that could be used in different risk assessment methods. HARA uses severity and exposure, for example, while TARA measures impact and likelihood. Connected to each individual dimension is its own subformula. This formula will make use of the *parameters* entity defined for the dimension to calculate the dimension's level, be that of TL or IL. The parameters are different criteria of which to value dimension levels. Examples of parameters for a severity dimension could be the financial damage caused by an incident or the threat to human life. Each parameter has predefined value and mapping pairs, so the user can evaluate the parameter. An example in the human life impact parameter could be to have the following scale (scaling of values does not necessarily need to be linear):

- Minor damage to people (value of 1)
- Moderate damage to people (value of 2)
- Severe damage to people (value of 4)
- Death (value of 8)

When there are well-defined methods to be utilized in the database, the *projects* entity can be used. Every created project must contain a method so it can conduct risk analysis. Each project also stores metadata, like a tag number, the author's name, and the date of creation. In a project, risks can be identified and stored in the *risks* entity. The records of risks will store a description of the risk, what types the risk is, countermeasures planned for the risk, and a risk level. The dimensions to be evaluated by the user are assigned by the chosen method to each risk. During the risk level calculation of a risk, the output of the subformulas for each dimension will be used as input to the method's formula. Its output will then be stored as the risk level for the risk.

C. The Database

Based on the ER diagram, an initialization schema was created for a database engine. The schema is written with *MySQL* syntax, so the server must support *MySQL*. The intended use case is for the database to be hosted remotely, so it is not included directly in the source code of the tool.

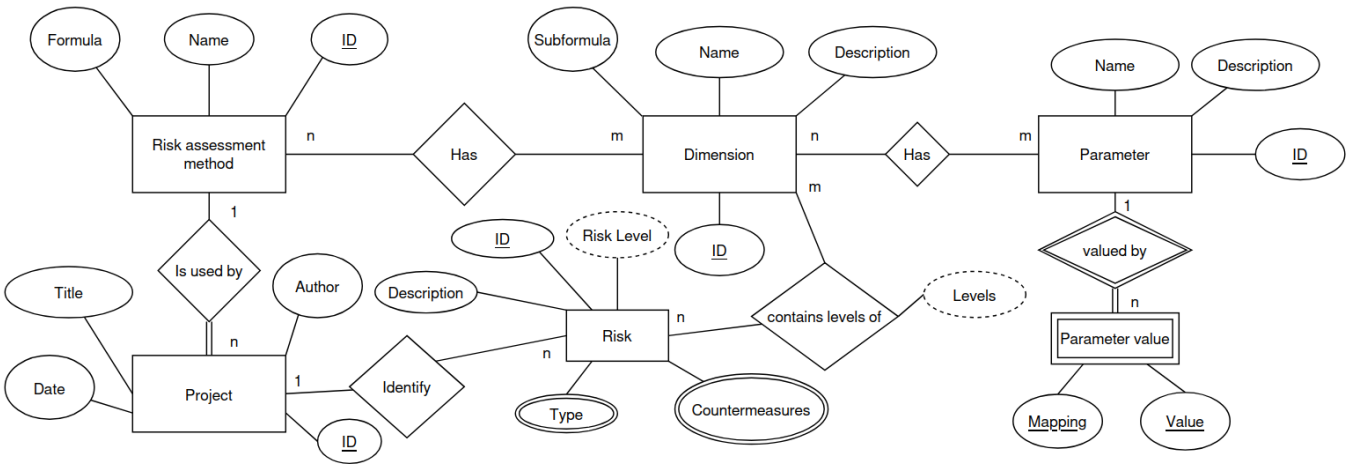

Fig. 4. Final ER diagram. Represents how the different entities within the database are related to each other.

In addition to tables representing the entities described in the diagram, a couple of complementary tables were created. Every table that has an attribute that can be multivalued has a connected table representing the values. And for each N:M-relation in the diagram, a complementary table exists to store which entities are related to each other. There is such a table for the relation between methods and dimensions, dimensions and parameters, and risks and dimensions.

To facilitate communication between the database server and the tool, several stored procedures were defined in the initialization schema. If the tool needs to send queries to the database, it can make use of these procedures instead of directly sending queries. The procedures guarantee that the necessary queries are executed atomically and without the underlying structure of the data. For example, when deleting a method, it is important to make sure that no dangling dimensions are left afterwards. Therefore, a procedure with transactions makes the process of deleting a method and its dimensions atomic. Methods, dimensions, parameters, parameter values, projects, and risks are all entities that the user needs to interact with. On top of that, there are attributes like formula, subformula, countermeasures, and levels that the user will edit. Most of these entities and attributes have add- and delete-functions that the tool will call upon request. Entities also have options to change name and give its ID to the software for proper display in the GUI. A few views are also defined in the schema to ease the display of specific items.

D. LUA Integration for Formula Evaluation

The aforementioned formulas stored within the database exist in the form of *LUA*[14] scripts stored simply as strings. These strings are retrieved from the database and are interpreted by a LUA virtual machine that is linked within the program. The choice to write the formulas in an interpreted language was motivated by the need for the ability to input them during runtime for method creation. The motivation behind choosing LUA in particular was threefold: its small code size, its easy syntax, and its simple integration into C++ programs. The small code size means we can build and link the entire language into our program, making it more portable. The simple syntax means that a user could input a formula without knowing virtually any LUA. A user with more knowledge about its syntax might instead use its powerful capabilities to, for example, run functions declared within the formula itself to simplify logic, or store intermediate data within locally declared structures.

E. Qt Quick and QML

In order to keep the project modern and extensible, we decided to write the graphical user interface (*GUI*) in QML through Qt Quick. QML is a declarative language which incorporates elements of Javascript, CSS and JSON[15]. Qt Quick contains the C++ API for the QML language, which allows for easy calling of C++ functions from QML signals. It also provides some primitive graphical types.

[14]https://lua.org/

[15]https://wiki.qt.io/Qt_Quick_Tutorial

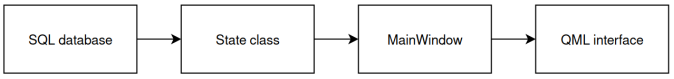

Fig. 5. The data flow through the different components of the program.

F. Project Organization

The application itself is fundamentally organized into three structures; the MainWindow class, which contains all the logic for connecting the front end to the backend. It inherits from the QObject class, which allows it to easily interface with the QML components. The State class, which handles all of the interfacing with the database as well as the execution of the formulas. An overview of the data flow within the application can be seen in figure 5. When running in terminal mode, only the State class is instantiated to not waste resources. Lastly, the QML code making up the GUI. This code defines the visual structure of the program and defines connections to functions made visible from the MainWindow class.

VI. Findings

RQ1

After examining several peer-reviewed studies on the subject of cohesive risk assessment methods, comparisons were made between the methods and some of the existing tools in the industry today. We observed that many of the peer-reviewed studies constructed state-of-the-art frameworks for critical architecture. This is due to their conformity to ISO/SAE standards and a wide range of parameters. We have additionally concluded that many of the currently available tools are inadequate to perform risk analysis with comprehensive methods like HEAVENS, because of their inability to take the required parameters and dimensions into account. The current tools tend to use simpler methods which are hardcoded into the application. All of the tools we've observed have failed to incorporate necessary TL, IL, and SL parameters to incorporate both cybersecurity and functional safety concurrently. Most of them are based on methods like CIA and STRIDE, and are usually missing parameters like hazard factors, safety and operational concerns, and metrics like controllability and severity. However, because of the limitations to the scope of the study, far from all industry-used tools were analyzed. There could exist tools that consider functional security, but either they were not found, or they were behind a paywall.

RQ2

Based on the problem investigation and multiple discussions with experts from RISE, we reasoned that a solution might be to implement the HARA/TARA tool. The aim of this tool is to be as flexible as possible. This is inspired by the different state-of-the-art frameworks that emphasize the importance of varying parameters based on what is needed within the current

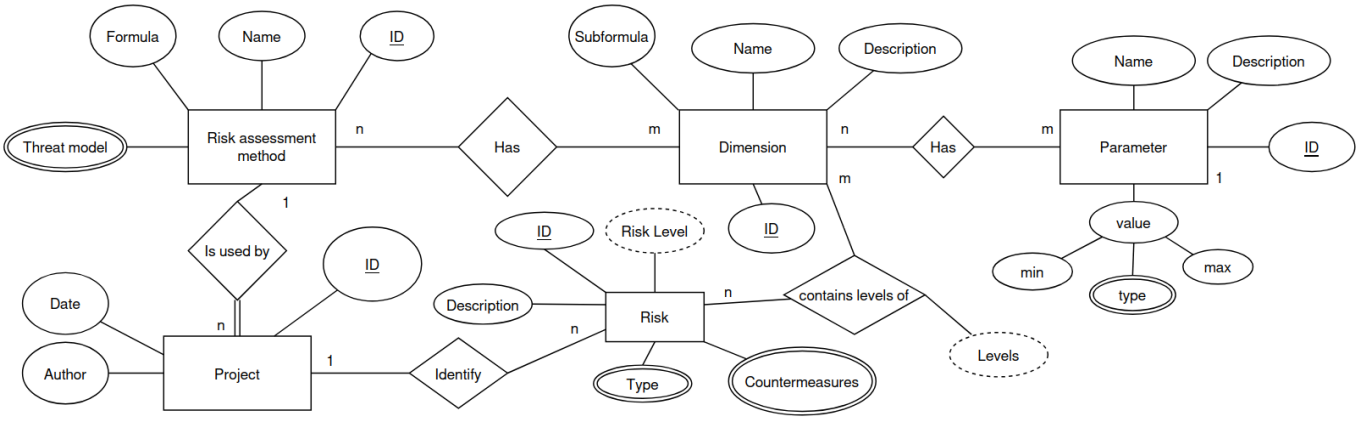

Fig. 6. First ER diagram.

domain of evaluation. To keep performance high and compatibility with different operative systems (OS's) wide, C++ and the Qt-Creator suite were utilized during the development of the tool. The structure of the underlying data was planned out together with RISE scientists before a proper implementation of the tool was attempted. An ER diagram was constructed to serve as a visual representation of the data and as the basis for an SQL server during cycle 1. The diagram was shown to experts from RISE in a presentation held over TEAMS at the end of cycle 1. After the presentation, an evaluation was held with the experts about the quality of the diagram. Before the evaluation, the diagram was structured in a different way, seen in figure 6. During the evaluation, RISE representatives noted that parameter values should be represented in a more flexible way. The old version of parameter values only allowed for a specified range of integers or floats as the value. The definition of value type was also unclear. To remedy this, we changed it so that parameter values are a weak entity type instead, so each parameter can be related to an arbitrary amount of parameter values. The current version of parameter values is explained in Section V-B. The scientists also felt that the attribute *threat model* was unnecessary, and that a title was needed for *project*. In response, we removed and added the attributes, respectively. RISE then confirmed the model to be representative of all that is needed for a flexible risk assessment method.

The HARA/TARA tool was then developed during cycle 2. As explained in Chapter V, the tool consists of a UI and an SQL database. Both are explained in the formerly mentioned Chapter. At the end of cycle 2, an evaluation of the constructed database and the UI was conducted with RISE. They noted that our approach of keeping most of the logic controlling the database storage within SQL procedures was a good one. It allows different tools to use the same database by simply calling said procedures from their respective MySQL bindings. It also helps with the separation of concerns. The communication between the *State* class and the database is also streamlined, by only needing to send a call query.

There were some critiques of the database as well. First off, while each individual procedure has transactions as to make the command atomic, complete creations of methods are not. This allows for two users to create and potentially clash with dimensions at the same time. Methods also become visible before they are fully designed. The scientists noted that these features are not necessarily negative, but could require change depending on design choices in the future. Some checks to make sure no duplicate objects are created were deemed unnecessary as well. The usage of well-placed constrictions in the database made the extra checks redundant.

Since most of the research-based functionality resides in the database, the RISE scientists did not have too much to say about the UI. It does not currently support multi-user handling, but the code is commented well enough to show how one could give the code the proper functionality. The script-based UI is left open-ended to allow for features to be implemented in the future. The GUI was seen as appropriate for the task of streamlining the process of creating methods and conducting risk analysis. With regard to the choice of having formulas represented by LUA scripts, they were similarly supportive, stating that LUA has been time-tested and is simple to incorporate into C and C++ projects.

Based on the previously mentioned observations, there is adequate support to say that the HARA/TARA Tool is one such solution that might streamline the process of risk assessment in critical infrastructure.

VII. Discussion

In this section, we will first argue around the validity of the results. After that, the implications of this work for the industry in practice and academia in research will be discussed.

A. Validity

The validity of the results and conclusions is based on two factors: the validity of the method used, and the validity of RISE as an authoritative source in the subject. As explained in the background, RISE conducts research in the domain of cybersecurity within critical infrastructure. Therefore, although

we also became familiar with the needs within the domain, a lot of design decisions were based on input from RISE. They helped us in the design of the ER-diagram, especially when it comes to what entities would be relevant in a tool like this. This, in combination with our newly found knowledge, led to the creation of the HARA/TARA tool. A typical scientific method of showing quantitative or qualitative results was not used in this study. A survey, benchmarking, interviews or similar methods were deemed too slow for the period of this study. What we ended up basing the results on was the evaluation of the tool by RISE. Because RISE is only one source of authority, and nothing concrete was used to assess the results other than their word, we are limited in our conclusions. A generalized statement cannot be made based on the results. It can, however, be deemed a proof-of-concept since there are professionals in approval of this tool. Whether their approval means anything to research is up to scrutiny of the company.

B. Implications for Research

The interest from RISE in a tool with the capability to analyze functional safety and cybersecurity concurrently gives us some proof that such a tool could be well received in the industry. Current tools are usually limited in their choice of parameters, especially when it comes to functional safety. While nothing definitive can be said with the method used in this study, we can at least reason that the examined open-source cybersecurity tools are lacking support for functional safety. Although the created tool cannot be proven to exhibit the properties that solve the problems of tools past, it at least has support from professionals at RISE. A more in-depth study of frameworks and ISO/SAE standards could be conducted to reach a better understanding of the features needed in future risk assessment tools.

All tools in the industry were not analyzed in the study. To make a generalized statement about risk analysis tools, a study of larger scale would have to be conducted. The tools compared in this study are just a fraction of the tools in use in the industry, so the conclusions only revolve around them. Hopefully, more studies are conducted to analyze the tools that were not included here. We also encourage other researchers to expand on the ideas presented in the tool. The implementations are open source and thus free to the public to observe and scrutinize.

C. Implications for Practice and Wider Context

This tool might streamline the professional work related to the securing of complex systems, combining cybersecurity and functional safety. As cybersecurity and functional safety become linked, a mix of people with backgrounds in both must form the teams responsible for ensuring safety and security [1]. Not having a tool capable of representing both paradigms at the same time could complicate the collaboration between the said members. The goal of the tool is to streamline the risk analysis process in critical infrastructure. If this goal can be reached, it could significantly increase security and safety in these sectors and be a net positive to society. However, the effectiveness of the tool is yet to be confirmed, as only the concept has been constructed and presented to RISE researchers. Only after complete development and a full release of the tool can results be measured. An attempt has been made to design the code in order for it to be maintainable at delivery, and the next implementation steps are suggested via documentation. The entirety of the source code will be kept open source which aids in the inclusivity of the development process. Risk analysis is an endeavor that should be available to everyone seeking to improve their security, and being able to acquire the software freely aids in this aspect. We hope that this study helps spread awareness of the problems surrounding risk assessment within critical infrastructure. We also hope that more tools start integrating functional security into their cybersecurity analysis.

VIII. Conclusion

The goal of the study was to depict how solutions look now and to show how new solutions might improve risk assessment in critical infrastructure. After comparing multiple industry-standard tools with each other and to state-of-the-art frameworks, the flaws became apparent. Flexibility is missing in most tools, and functional safety is never considered. The HARA/TARA Tool was designed to try and make it possible for the user to implement standard-compliant risk assessment methods. A generalized statement cannot be made regarding the quality of this tool. Nonetheless, the tool shows promise in evaluations with RISE scientists. Being based on the needs of risk assessment frameworks and input from RISE, this tool shows potential in its ability to assess risk in critical infrastructure.

References

[1] Australian Signals Directorate's Australian Cyber Security Centre (ASD's ACSC) and collaborating agencies. *Principles of operational technology cyber security*. Tech. rep. Developed in collaboration with CISA, NSA, FBI, MS-ISAC, NCSC-UK, Cyber Centre Canada, NCSC-NZ, BSI Germany, NCSC-NL, NISC Japan, NPA Japan, and NIS Korea. Accessed: 28 April 2025. Australian Signals Directorate, 2024. URL: https://www.cyber.gov.au/resources-business-and-government/maintaining-devices-and-systems/critical-infrastructure/principles-operational-technology-cybersecurity.

[2] Eric H. Kessler. *Encyclopedia of Management Theory*. Thousand Oaks, Calif.: SAGE, 2013, p. 2. ISBN: 9781412997829.

[3] Jherrod Thomas. "HATARA: A Novel Approach by Fusion of HARA and TARA for System Safety and Security Analysis". In: *International Journal of Innovative Science and Research Technology* 9.2 (2024). URL: https://ijisrt.com/assets/upload/files/IJISRT24FEB335.pdf.

[4] Vivek Agrawal et al. “Threat/Hazard Analysis and Risk Assessment: A Framework to Align the Functional Safety and Security Process in Automotive Domain”. In: *SAE Technical Paper Series* 4.2 (Mar. 2021). DOI: https://doi.org/10.4271/2021-01-0148.

[5] Mafijul Islam and Aljoscha Lautenbach. *HEAVENS - HEAling Vulnerabilities to ENhance Software Security and Safety Dnr 2012-04625*. Volvo AB, Mar. 2016. URL: https : / / autosec . se / wp - content / uploads / 2018 / 03 / HEAVENS_D2_v2.0.pdf.

[6] Ilona Loskin. *TARA+AD: Threat Analysis and Risk Assessment for Automated Driving : cybersecurity of road vehicles*. University of Jyväskylä, 2023. URL: https://urn.fi/URN:NBN:fi:jyu-202306193943.

[7] Eric Knauss. “Constructive Master’s Thesis Work in Industry: Guidelines for Applying Design Science Research”. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. 2021, pp. 110–121. DOI: 10.1109/ICSE-SEET52601.2021.00021.

[8] Roel Wieringa. “design science as nested problem solving”. In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST ’09* (2009). DOI: 10.1145/1555619.1555630.

[9] Miroslaw Staron. *Action research in software engineering: Theory and applications*. 1st ed. Springer, 2020.