FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Enterprise Architect Based Tool for Security Threats Identification

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Magdy El Sadany
Matrikelnummer 01229407

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dipl.-Ing. Dr.techn. Wolfgang Kastner
Mitwirkung: Mag. Christoph Schmittner, Austrian Institue of Technology

Wien, 30. April 2019

Magdy El Sadany	Wolfgang Kastner

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Enterprise Architect Based Tool for Security Threats Identification

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Magdy El Sadany
Registration Number 01229407

to the Faculty of Informatics

at the TU Wien

Advisor: Dipl.-Ing. Dr.techn. Wolfgang Kastner
Assistance: Mag. Christoph Schmittner, Austrian Institue of Technology

Vienna, 30th April, 2019

Magdy El Sadany	Wolfgang Kastner

Erklärung zur Verfassung der Arbeit

Magdy El Sadany
Linzer Straße 150-158/13/4, 1140 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. April 2019

Magdy El Sadany

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Dr. Wolfgang Kastner, der meine Bachelorarbeit betreut und begutachtet hat. Für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich herzlich bedanken.

Ich möchte mich bei dem Leiter der „Dependable Systems Engineering“ Abteilung des Austrian Institute of Technology Dr. Willibald Krenn, MSc. Christoph Schmittner, für die Unterstützung meiner praktischen Arbeit bedanken.

Ein besonderer Dank gilt allen Teilnehmern und Teilnehmerinnen meiner Befragung, ohne die diese Arbeit nicht hätte entstehen können. Mein Dank gilt ihrer Informationsbereitschaft und ihren interessanten Beiträgen und Antworten auf meine Fragen.

Ebenfalls möchte ich mich bei meinen Kollegen Msc. Shabaan Abdelkader bedanken, der mir mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite stand. Bedanken möchte ich mich für die zahlreichen interessanten Debatten und Ideen, die maßgeblich dazu beigetragen haben, dass diese Bachelorarbeit in dieser Form vorliegt.

Abschließend möchte ich mich bei meinen Eltern bedanken, die mir mein Studium durch ihre Unterstützung ermöglicht haben und stets ein offenes Ohr für mich hatten.

Acknowledgements

In this position, I want to thank all those who supported and motivated me during the preparation of this bachelor thesis.

First, my thanks go to Dr. Wolfgang Kastner, who supervised and examined my bachelor thesis. For the helpful suggestions and constructive criticism in the preparation of this work, I would like to thank.

I would like to thank the head of the Dependable Systems Engineering Department of the Austrian Institute of Technology Willibald Krenn, MSc. Christoph Schmittner, for the support of my practical work.

A special thanks to all participants of my survey, without whom this work could not have come about. My thanks go to her willingness to provide information and her interesting contributions and answers to my questions.

I would also like to thank my colleagues. Thanks to MSc. Shabaan Abdelkader, who helped me with patience, interest and helpfulness. I would like to thank you for the many interesting debates and ideas that have contributed significantly to the fact that this bachelor thesis is available in this form.

Finally, I would like to thank my parents, who made my studies possible with their support and always had an open ear for me.

Kurzfassung

Die Bedrohungsmodellierung beschreibt eine Sicherheitsanalysemethode, bei der ein abstraktes Bedrohungsmodell auf ein Systemmodell angewendet wird, um zu bestimmen, welche Bedrohungen in diesem System möglich sind. In dieser Bachelorarbeit wird eine Erweiterung auf dem Gebiet der Bedrohungsmodellierung vorgestellt. Begleitend zum theoretischen Teil zeigt die Arbeit praktische Erweiterungen zu einem bestehenden Bedrohungsmodellierungstool, die unter Verwendung von cSharp, SharpDevelop, Wixtools und Enterprise Architect durchgeführt wurden. Weiteres beinhaltet die Arbeit auch ein Überblick über die vier BSI Schutzprofile und eine konkrete Untersuchung des Entwurfs der autonomen Fahrzeuge nach Common Criteria, welcher im September 2018 veröffentlicht wurde. Für die Analyse des modellierten Systems wurde ThreatGet - ein PlugIn in Enterprise Architect - entwickelt. ThreatGet beinhaltet eine Datenbank, die sich durch ihre Skalierbarkeit auszeichnet.

Die Anwendung der Schutzprofile in ThreatGet wird in Zukunft eine große Rolle spielen. Die Arbeit zeigt exemplarisch, wie eine Regel aus einem Schutzprofil für autonome Fahrzeuge in ThreatGet modelliert werden kann.

Abstract

Threat Modeling describes a security analysis method in which an abstract threat model is applied to a system model to determine which threats are possible in that system.

This bachelor thesis takes into account the logical system structure and presents an extension in this area. Accompanying to this, a refinement of an existing threat modeling tool has been carried out using cSharp, SharpDevelop, Wixtools and Enterprise Architect. The work also includes an overview of the four BSI protection profiles and a study of the design of the autonomous vehicles according to Common Criteria, which was published in September 2018. To analyze the modeled system, ThreatGet - a plug-in in Enterprise Architect - was developed. ThreatGet also includes a database characterized by its scalability.

The application of the protection profiles in ThreatGet will play a major role in the future. Thus, the work includes the definition of a rule from a protection profile of autonomous vehicles, which was modelled in ThreatGet.

Contents

CHAPTER 1

Introduction

1.1 Motivation and Background

The Internet is the largest and most widely used communication network in the world. Since the beginning, it was developed by humans for humans. With the evolution of smart devices in our lives, communication and the Internet are changing for machines (Internet of Things - IoT). Data is collected and communicated to optimize efficiency, productivity or energy consumption [1].

At the IoT, the devices are enhanced with software and sensors for data storage and communication. The individual devices are often realized or interconnected via particular platforms. An IoT platform collects the data and informs consumers about the activities [1].

The principle of IoT also plays an essential role in the design of autonomous vehicles. Since vehicles will have to work without a human, there is a need for coordination between vehicles and infrastructure. Therefore, it is crucial to maintain communication. This includes the vehicles, the infrastructure, other road users, and all involved stakeholders. Autonomous vehicles are expected to have a central gateway that collects and filters information and data from all the sensors in the car, and then forwards it to a platform. The platform collects all information from all road users and then returns the corresponding answer. This enhances a single vehicle to have global awareness and allows vehicles to learn from data collected by other vehicles

Autonomous vehicles are no longer seen as an independent system but as a vast ecosystem. In such a system, communication and security must be given. The advantages of such a system lie in the dynamic development and expansion possibilities [9].

The security requirements need to ensure secure communication between the participants in the system and are based on the principle of "security by design." This means the

whole development includes risk management from the beginning, e.g., risk identification, evaluation, mitigation, and re-assessment. Modeling a system enables a systematic process, by which the system model can be analyzed, security measures added and the system can be re-analyzed whenever details are added. An example of a tool for this process is the Microsoft Threat Modelling Tool.

The tool offers so-called templates and the possibility to extend the tool. A template consists of entities, describing the available elements to model a system and threat types, describing potential threats [13].

Microsoft Threat Modeling Tool is available as a plugin for Visio. Managing and extending a template is difficult due to a minimal available interface. Visio is also a purely graphical modeling system, e.g., we cannot define relations or underlying behaviors. All in all, only the logical system structure is considered. These known issues led to the development of a ThreatGet threat-modeling prototype in Enterprise Architect (EA) by the Austrian Institute of Technology (AIT). ThreatGet uses different security parameters to understand and discover the threats in a system, taking into account the data flow between each element.

1.2 Structure of the Work

As part of this thesis, cybersecurity and security investigations are handled through the use of ThreatGet as a threat modeling tool. Chapter 2 of the thesis explains the Common Criteria (ISO / IEC 15408) as well as the security model and the standard ISO / SEA 21434. Besides, in Chapter 3, ThreatGet is analyzed, improved, and handled in a practical part of the thesis. The use of ThreatGet is also treated with the "Digital Tachograph - Vehicle Unit (VU PP)" given as an example of the use of a protection profile in EA [3].

The work is a collaboration between the Automation Systems group of the Institute for Computer Engineering of Vienna University of Technology and AIT.

CHAPTER 2

Threat Analysis

Threat analysis is an essential part of risk management. Generic risk management is defined by ISO 31000 "Risk Management - Principles and guidelines." Due to the evolutionary nature of risk, the generic model of risk management is a cyclical process. Figure 2.1 shows exemplary the main steps for the use case of autonomous vehicles.

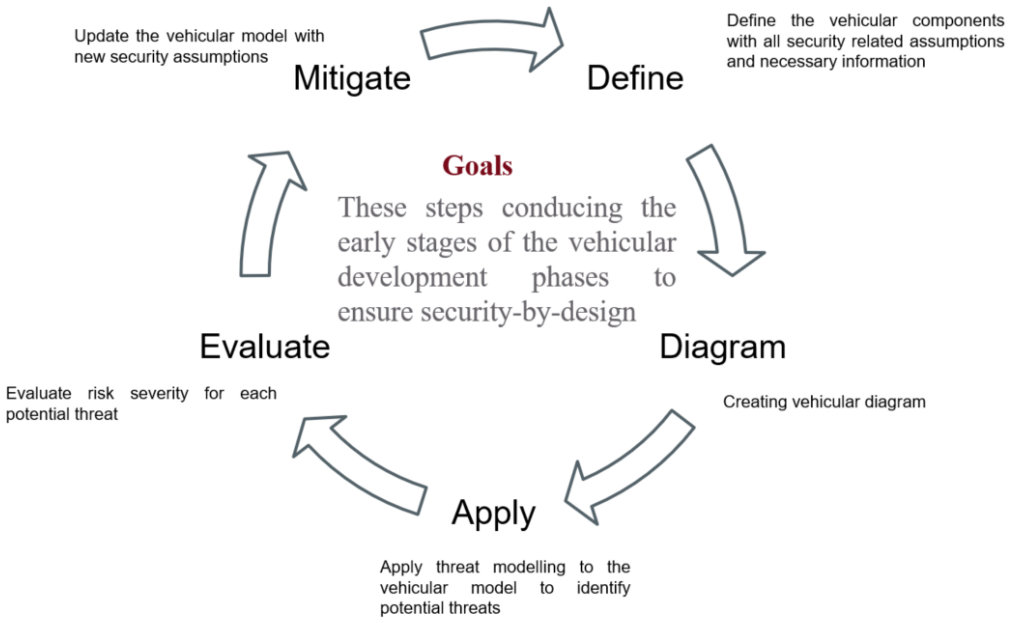

Figure 2.1: The Five Major Risk Management Steps

For IT and cyber systems, this process is described in ISO 27005 "Information technology - Security techniques - Information security risk management" [4].

ISO 31000 defines the generic level for risk management and is the highest high level ISO standard for risk management. It thus defines the foundations for domain specific models. Depending on the domain, it is used as a template. Based on the ISO 31000 defined principles, ISO 27005 for information technology and ISO / SAE 21434 for autonomous vehicles were established for risk management.

2.1 Road Vehicle - ISO/SAE 21434

The modern autonomous vehicles are no longer referred to as old cars, but are considered in many areas as mobile computers. For safety, there is the ISO 26262 standard [12]. Due to increased connectivity, security is a new topic, for which a new standard is in development.

The Cyber-Security and Information Standard ISO / SAE 21434 will demonstrate the commitment in all areas and is expected to be completed by the end of 2020.

Currently, the ISO / SAE 21434 has been in draft form since September 2018 in the Common Draft (CD), therefore the already existing information can change. As sector standard for automotive domains, the risk management process is based on ISO 31000. While ISO/SAE 21434 gives no detailed guidance on methods, previous approaches include attack trees, the eTRVA, EVITA and HEAVENS [13, 7, 6] [13, 7, 6].

2.1.1 Attack Trees

Attack trees represent multiple sequences of actions an attacker could take to reach a certain goal. The root node represents the goal of the attacker while the leaf nodes represent actions and attack steps. Leaves can be combined with “AND” and “OR”. “AND” represents multiple actions which are required in combination to reach the top node. “OR” represents multiple actions where one of them is required to reach the top node. Systems can have multiple goals which can be identified by a previous activity. On a high-level, the attack tree generation can be divided into three steps. Identification of attack goals identify potential attack sequences and rating nodes [13, 7, 6].

For the generation of an attack tree, partial trees from previous analyses can be reused and an attack tree can also include countermeasures. Although attack trees appear similar to fault trees for safety, they are difficult to combine due to different level of values which can be assigned to leaves. In a fault tree, a hardware failure represented in a leaf node can have a well-known and experience based failure probability, something which is difficult to enumerate for a security event. In security, assigned values can be costs, complexity or required time for an attack, which can be used to prune the tree by defining thresholds, but not for direct calculation. There are also some approaches to combine fault trees and attack trees to consider complex scenarios [13, 7, 6].

2.1.2 Framework

ETSI Threat Vulnerability and Risk Analysis (eTVRA)

eTVRA is a generic approach to threat, vulnerability and risk analysis. It was developed for the telecommunication sector and later applied to Intelligent Transportation Systems. The goal of eTRVA is a systematic identification and mitigation of unwanted incidents.

eTRVA starts with an identification of the assets followed by identifying of vulnerabilities, threats which can exploit these vulnerabilities and potential following system level impacts. The quantification of the threats is based on ISO/IEC 15408. Based on impact and quantification, risks are ranked. The method proposes a template to be used for recording threats, threat agents, weaknesses and vulnerabilities. The steps are:

1. Identification of security objectives.
2. Identification of the requirements, derived from the objectives from step 1.
3. Inventory of the assets.
4. Identification and classification of vulnerabilities, threats and unwanted incidents.
5. Quantifying the occurrence likelihood and impact of the threats.
6. Establishment of the risks.
7. Identification of countermeasures.

EVITA

The EVITA project aimed at securing vehicular on-board systems and developed besides security solutions also a methodology for threat and risk analysis. The EVITA methodology rates risks based on severity and attack potential. While EVITA defines a security engineering lifecycle, the focus here is on the steps for threat identification.

1. Develop view on system.
2. Describe relevant use cases.
3. Identify assets to be protected within the use cases.
4. Identify threats to the assets.
5. Evaluate and rank risks.
6. Identify security requirements for the threats based on risk analysis.

For the identification of threats (step 4), "dark-side scenarios" are used. This approach aims at identifying potential attacker motivation and capabilities, and based on this information to model the attacks. Based on attack goals that satisfy the motivation of the attacker, attack trees are developed to identify scenarios how an attack could be conducted. EVITA structures the attack trees in three major levels. Level 0 is the high-level goal of the attacker. Level 1 contains multiple objectives how an attacker could achieve the goal and have a negative impact on stakeholder. Level 2 and below model attack methods which can consist of multiple intermediate steps, connected with "AND" and "OR".

The risk analysis (step 5) is based on high-level security objectives (operational, safety, privacy and financial) where the severity of a threat is rated and the rating of attack potential of the identified scenarios.

HEAVENS

The HEAVENS project aimed at addressing software vulnerabilities which could impact safety and security in vehicles. It developed a method for threat analysis and risk assessment, contained in the HEAVENS security model, which was updated in the HOLISEC project. The HEAVENS workflow consists of three main phases, threat analysis, risk assessment and security requirements.

The threat analysis requires as input the functional use case and identifies threats for each asset involved in the use case. Threats are also mapped to security attributes, e.g. which security attribute is endangered by a threat. For the threat identification, STRIDE and threat modelling are used. The approach is aimed at the concept phase where vulnerabilities are not yet known, e.g. threats are identified independent from vulnerabilities.

Risk assessment is done by ranking the impact (Impact Level, IL) on an asset and the potential of a threat (Threat Level, TL) and defining the risk (Security Level, SL) based on this. Threat levels are based on Common Criteria, the impact is similar to EVITA but extended with impact on legal and regulatory assets.

Threat Modeling

Threat Modeling [13] is the theoretic foundation of ThreatGet and used in the EVITA and HEAVENS method. In addition, it has also some history of application to automotive [6].

2.2 Common Criteria

The Common Criteria for Information Technology Security Assessment is an international standard for various and general criteria to assess and verify the security features of IT products. The introduction of the protection profile puts the industry on cybersecurity and also sets a minimum level of required and agreed cybersecurity.

SDO	No.	Title
SAE	J3061	Cybersecurity Guidebook for Cyber-Physical Vehicle Systems
	J3138	Guidance for securing the Data Link Connector (DLC)
ITU-T SG17 Q13 (Security aspects for Intelligent Transport System)	X.1373	Secure software update capability for intelligent transportation system communication devices
	X.itssec-2	Security Guidelines for V2X Communication Systems
	X.itssec-3	Security requirements for vehicle accessible external devices
	X.itssec-4	Methodologies for intrusion detection system on in-vehicle systems
	X.itssec-5	Security guidelines for vehicular edge computing
ISO/SAE	ISO/SAE 21434	Road Vehicles — Cybersecurity Engineering

Figure 2.2: Overview about the security standards of the vehicular domain

The protection profile aims to define a certain set of security related requirements (evaluation and functional) for a certain system. This system is defined as Target of Evaluation (ToE).

Common Criteria is aimed at three groups of stakeholders, Consumers (supporting the decision if a ToE fullfils the security needs), Developers (identifing security requirements suitable for a ToE) and Evaluators (guiding on how to evaluate a ToE).

The model behind Common Criteria is that assets need to be safeguarded by countermeasures. In order to secure a system, the countermeasures need to be:

- correct: that the expected functionality is provided,
- sufficient: that the considered threats are covered.

While the evaluation criteria give guidance on how to analyze the countermeasures, it would be beneficial to analyze at an early stage if a system design complies with a protection profile. We present in the following chapter a tool which implements a novel approach to threat modeling and follows with a concept on how to assess a system for compliance with a protection profile.

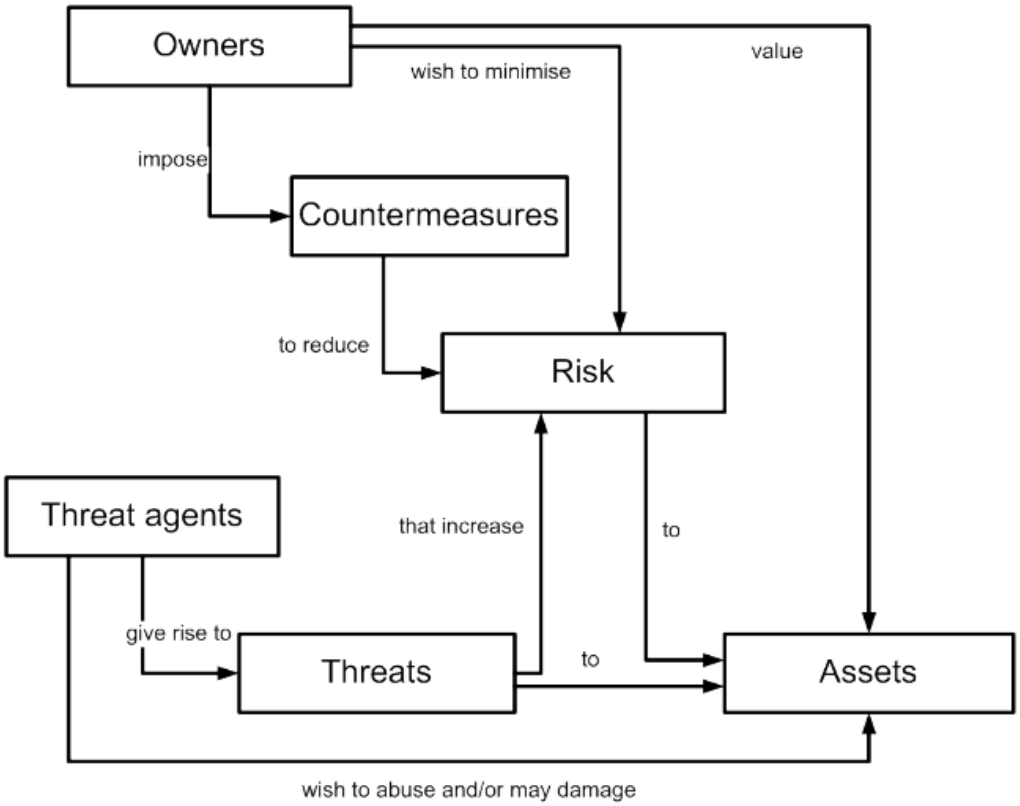

Figure 2.3: Structure of a CC Modell

CHAPTER 3

ThreatGet

ThreatGet is based on Enterprise Architect (EA), which makes use of a "complete" modeling tool. Therefore, after performing the operations, several charts presenting the result are available. The advantage of application-based research lies in dynamic expansion. Therefore, if there are changes in the original system modeling, the result already calculated will reflect the changes, and the threats will be updated accordingly. So, it is possible to proceed step by step and eliminate threats.[11].

3.1 Starting Status

As mentioned in Chapter 1, ThreatGet's mission is to investigate, identify, and reference possible threats that could disrupt systems or interaction. The lack of documentation of EA meant that the plug-in and thus the current situation is not performance-optemized. ThreatGet iterates over all existing objects and checks via connections if impending threats could be calculated. The projects in EA are subdivided into models, then into diagrams and further into objects. About the objects, it is then possible to address the connector. Thus, ThreatGet executes several loops to calculate the result which is not performance-based. In order to get to the objects and successfully iterate over the objects, an effort of $O(n^4)$ is needed. In addition to suboptimal performance, ThreatGet does not yet offer an extension of the database for elements and possible threats. Thus, the current version is based on a self-predefined database. The user has the possibility via EA to edit the properties of the objects and connections; however the insight and the processing of such switches is not possible due to the missing documentation.

3.1.1 Example

The system model is shown in Figure 3.1 It describes a generic interaction in an autonomous vehicle.

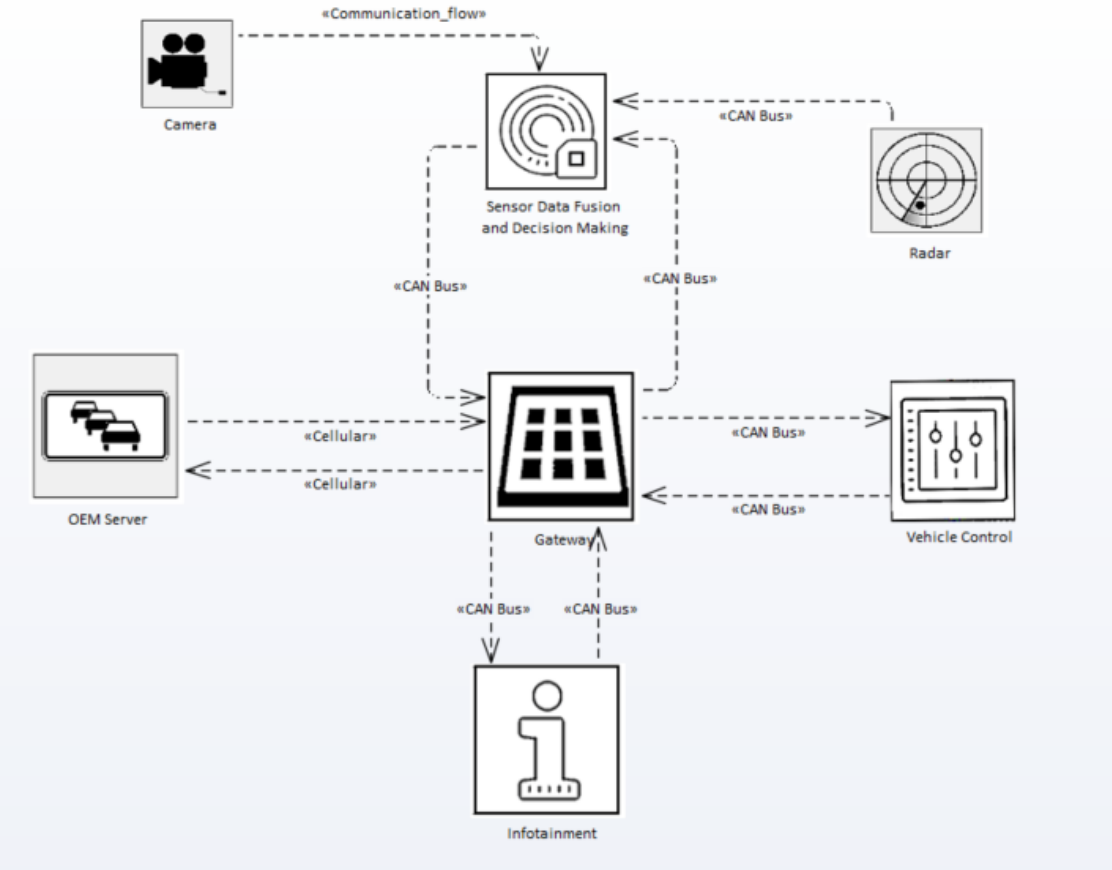

Figure 3.1: Generic interaction in an autonomous vehicle

Figure 3.2 shows the summarized result of the operation. On the left of Figure 3.2, the results are shown for an interaction. On the right side, the list of all found threats incl. the risk level (High, Medium and Low) is displayed. Furthermore, for each threat the "Impact" and the "Likelihood" are displayed, which is not yet implemented in the current state.

As Figure 3.3 shows, every interaction and their threats have their own EA diagrams, so the user has an insight into the individual diagrams and can also change the objects at any time if required.

After re-executing the operations, the created graphs of the first execution are deleted and replaced with the new graphs.

3.1.2 Development Point

The modeling of EA is based on several MDGs. The structure and details of an MDG technology are explained in more detail in Sections 4.1. Installing EA will install the

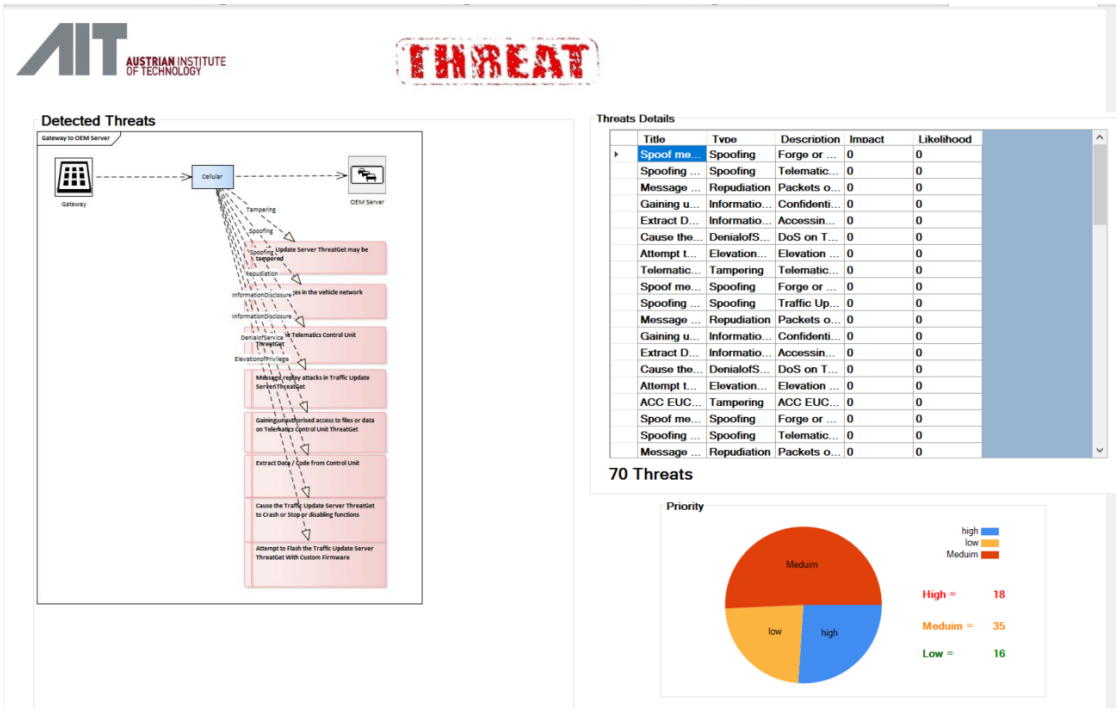

Figure 3.2: Summarized result of ThreatGet

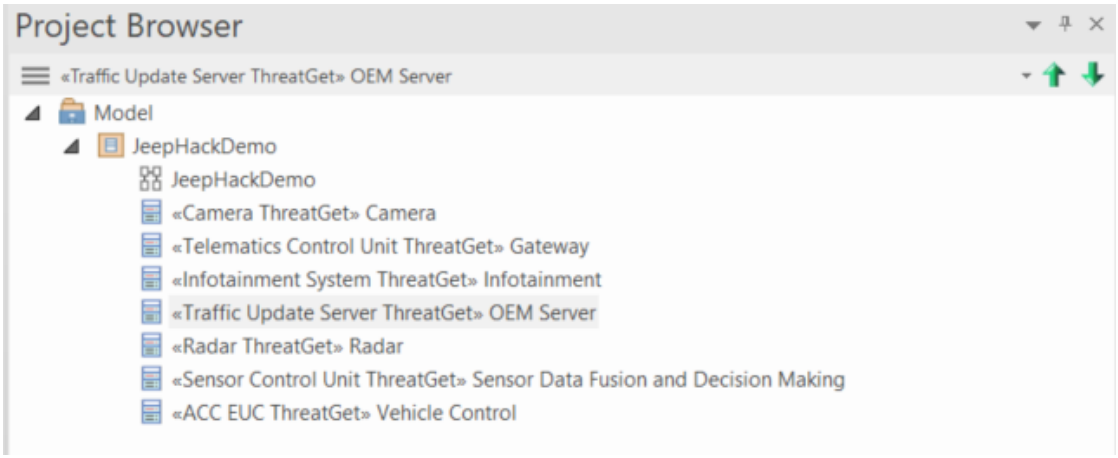

Figure 3.3: Individual diagrams of all identified threats

standard MDGs. These are partially encrypted XML files which are decrypted by EA and then displayed in the interface of the program the user. ThreatGet is also based on an MDG and therefore on an encrypted XML file. An installation package is needed to connect the plug-in to the default setting of EA.

The default performance of ThreatGet analysis in EA is $O(n^4)$. The poor performance

in EA is based on interactions which in EA consider models, diagrams, objects, and connections to find possible threats. In an example of four models, four diagrams, four objects with four connections each, EA needs a total of 256 iterations to calculate a result for each object. For performance optimization, the use of SQL is necessary. The selection methods of SQL make it possible to address specific objects or diagrams and then perform an iteration via the corresponding connection. The results of this procedure will be discussed in Section 4.2.

ThreatGet supports system analysis and risk evaluation. To enable this, the user must get complete access to the data flow of the modeled system. For this purpose, the ThreatGet MDG will give the user the ability to access the properties of the connections and their values. The properties make it possible to track the complete data flow through the connections and to control accordingly.

ThreatGet is not restricted to one domain and can be used in several areas. For this purpose, a database is created, and a suitable user interface should allow the modelers to create their threats. It is also possible to edit the existing threats or to delete them.

3.2 Planned Use Case

The basis - theoretical part - of the use case is dealt with in the subsection and the practical part is dealt with in chapter 5. The creation of the ThreatGet MDG and its details are covered in chapter 4.

The vehicle unit (VU) and its use in vehicles are discussed for the concept of the application case. VU's activities include recording, displaying, storing, printing and finally outputting vehicle activities and data. As in the aforementioned IoT example in Chapter 1, first of all, the data and activities of the users are recorded and stored in an internal memory and an internal tachograph card. The data to be displayed is then forwarded by the VU only to the corresponding devices.

As a basis for the threats description, the "UN Task Force on Cyber security and OTA issues (CS / OTA)" threats 2 "takes into account the focus is on the "Dependable Systems Action to circumvent monitoring systems (eg hacking / tampering / blocking of messages search as ODR Tracker data, or number of runs)"[8].

Another option would be to examine the "Safety and Security Co-Analyzes" from 2018, since the paper provides a very good and comprehensive summary of the possible threats. [5].

The mentioned references form a good basis for the test and creation of own threats and rules. The examples are then modeled in EA and the results are visible in Section 5.

CHAPTER 4

ThreatGet Development

4.1 MDG Technology

The user can extend the features of Enterprise Architect by defining a customized toolbox that fits the exact needs. The Model Driven Generation (MDG) technology enables the user to access and use pertaining resources to a specific technology in Enterprise Architect. This section describes the different phases of MDG to create a customized toolbox. There are three main phases in the MDG to create a user-defined toolbox:

- Profiles are collections of extensions, based on stereotypes that are applied to UML elements, connectors and features.
- Toolboxes are described as containers of the defined elements, connectors, and feature are defined as stereotypes in the profile phase. A toolbox consists of one or more expandable/collapsible regions, referred to as Toolbox Pages [10].
- Diagrams are generated by the MDG technology which contain all the pre-defined elements, connectors, and other features integrated with toolboxes.

4.1.1 ThreatGet Toolbox

This section describes the steps for creating stereotypes of elements and connectors of the ThreatGet toolbox. To do so, a new MDG project needs to be created from selecting "Basic Template" diagram as shown in Figure 4.1.

EA generates the primary three phases of the MDG technology as illustrated in Figure 4.2.

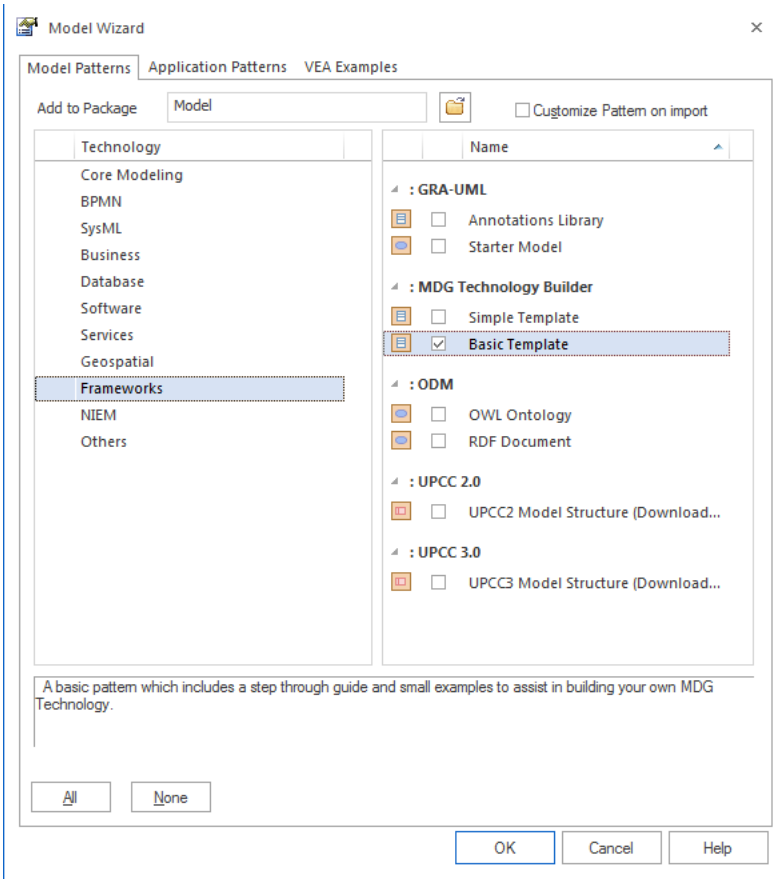

Figure 4.1: Creating a basic template MDG-Model

Before start creating the required stereotype, an icon of the underlying element needs to be defined. This step needs to be managed from the "Image Manager" as shown in Figure 4.3. Furthermore, all suitable images which are going to be used as unique shapes of the desired objects can be improved. As shown in Figure 4.3, the Image Manager aims to integrate all imported image(s) into the EA's image repository.

ThreatGet Profile Page

Profile(s) has/have a collection of user-defined stereotype based on the UML model. The stereotypes can be attributed to specifically define "tagged values" that further extend the characteristics of the stereotyped element or connector. Figure 4.4 shows the stereotypes of all objects and connectors in ThreatGet.

The stereotype can be created via drag and drop of the "Add Stereotype" from the Toolbox into the workspace. Some parameters need to be set to define the properties of the created stereotype. Figure 4.5 depicts the stereotype properties window.

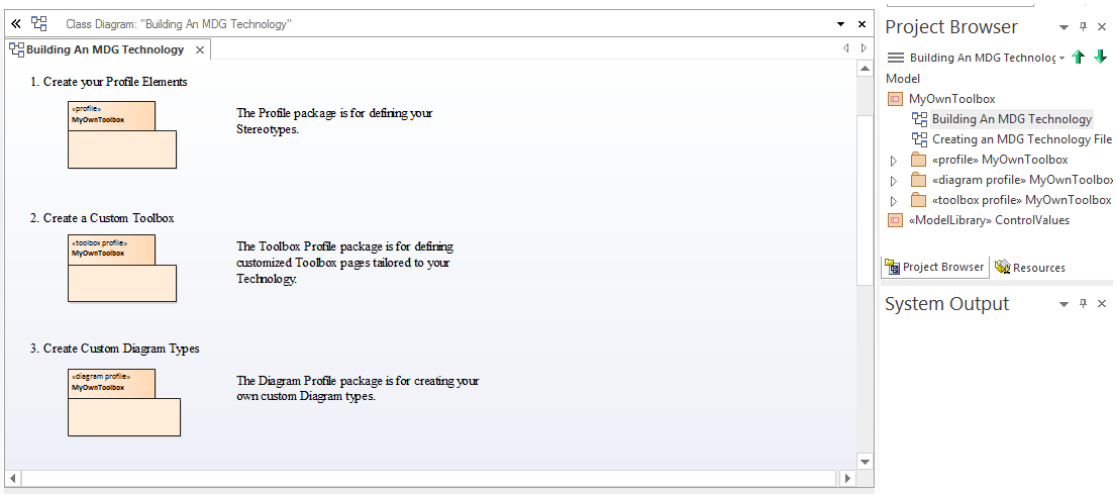

Figure 4.2: Contents of the created project

There are main stereotype parameters that have to be defined (i.e., the name, type, metatype). Also, there are additional properties of the stereotype, such as the tagged values and the shape of the unit which give additional properties to the defined unit. For example, the shape of the element can be set by selecting the "Shape Script" from the left side list. As discussed before, the EA's Image Manager is responsible for images in EA. Figure 4.6 illustrates how to set the sensor image to the sensor stereotype to be a unique shape of that element.

ThreatGet Toolbox Page

ThreatGet toolbox page gives access to all elements and connectors. ThreatGet classifies all the defined objects and connectors into eight main categories (i.e., Sensors, Actuators, Control Units, Interfaces, Communication Flows, Data Stores, Boundaries, User). As shown in Figure 4.7, these are the main toolbox pages of the ThreatGet objects. Each page has a list of the defined stereotype.

To create a new toolbox page, "Add Toolbox Page" from the Toolbox needs to be activated, then the elements are chosen which will be a part of this page. Figure 4.8 depicts all sensor objects which are listed under the Sensor toolbox page.

ThreatGet Diagram Page

After creating all the required elements and definition of the categories of these elements, there is only one step left which is to create the ThreatGet diagram page. To do that use the "Add Diagram Extension" from the toolbox and put it into the workspace via drag an drop, define the name and the toolbox page of ThreatGet which will contain, all of the defined elements. Figure 4.9 depicts the created ThreatGet diagram page using MDG technology.

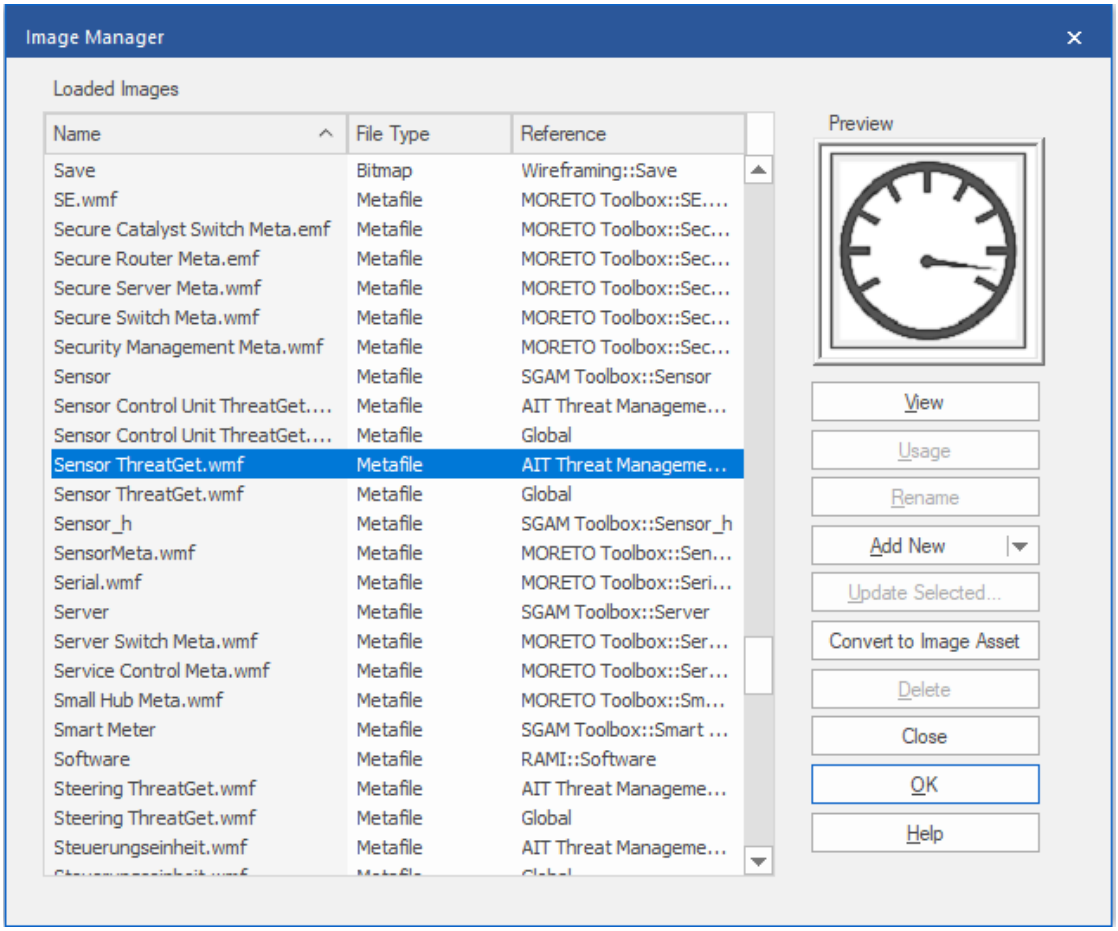

Figure 4.3: Image Manager

4.1.2 Generate ThreatGet package

The last step is to generate the ThreatGet package which can be integrated into any EA software. Choose “Generate MDG Technology” from the "Public" tab, the Generate MDG Technology window will be displayed as illustrated in Figure 4.10.

Afterward, the location paths of the ThreatGet contents such as profile, toolbox, diagram, images, and so on have to be defined. The MDG creation wizard generates the ThreatGet package in XML format.

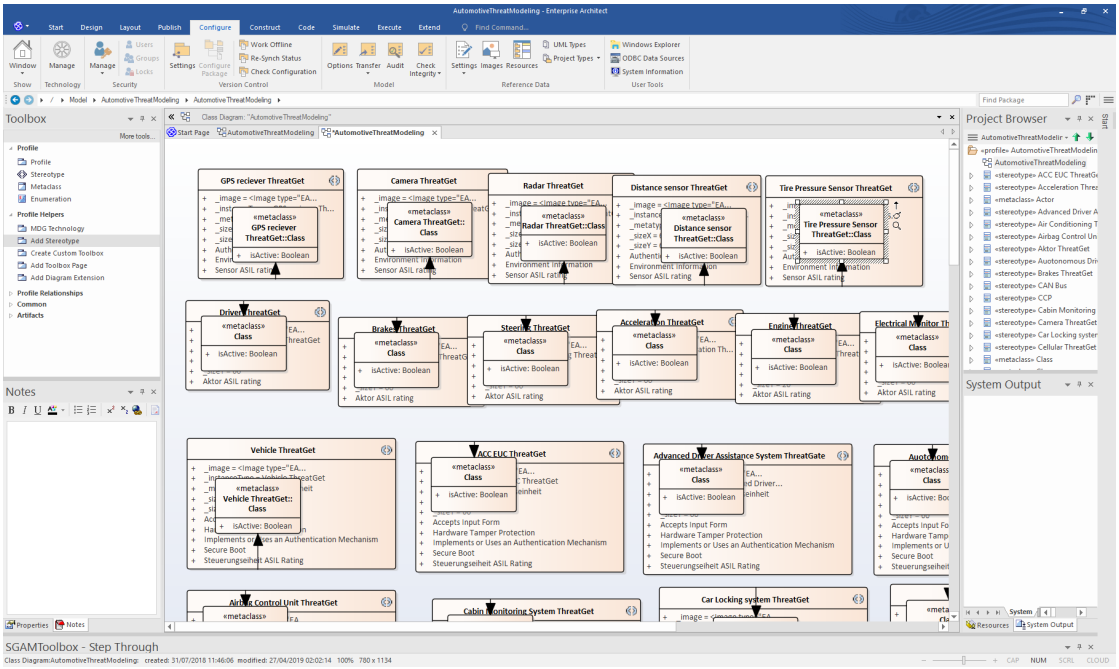

Figure 4.4: ThreatGet Stereotypes

4.2 Automation

SharpDevelop and Wixtools were used for the creation and completion of the ThreatGet MDG. The result of this process is an installation package, which then allows EA users to use ThreatGet. The installation package connects the created MDG with the responsible registry key and then binds it to the standard EA-MDG-Technology, so ThreatGet in EA is treated as any other MDG.

As mentioned in Section 3.1, ThreatGet's performance is a problem as it completes many different iterations to compute the final result. EA stores all models, packages, diagrams, objects, and connections in different databases. Each of these databases consists of different numbers of tables that have no documentation. The tables are nevertheless restricted to the "objects" and contain only a small amount of information. The missing documentation and the many tables presented a challenge, which will be addressed in Section 4.2.2.

The example mentioned in Section 3.1 presents the number of loops ThreatGet has to go through to get a stable result. As can be seen in the example, the number quickly scales to a high number, and the analysis can take up to several minutes. In order to solve the performance problem, a search function was implemented by the application of MySQL. The function consists of a connection between cSharp and MySQL and is executed only once if necessary. The performance of this search function is, therefore, $O(1)$. The iteration over the links in the found objects represents the performance of

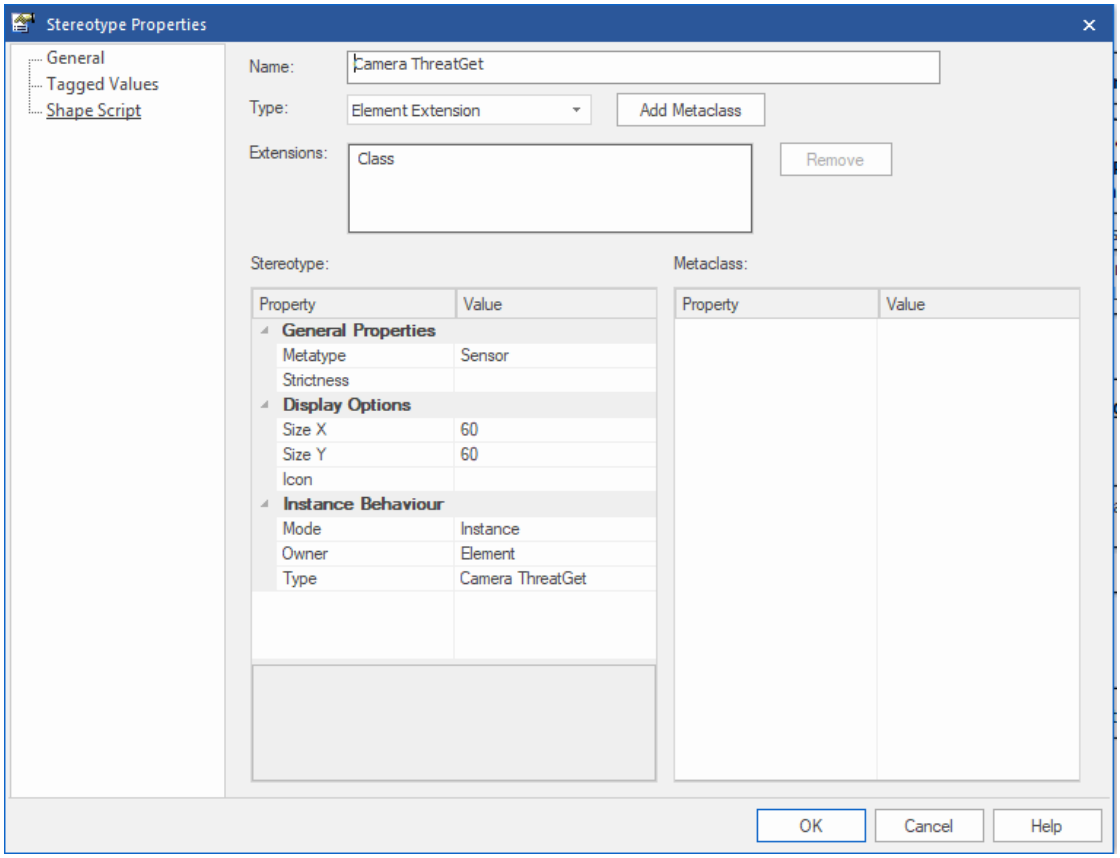

Figure 4.5: Stereotype propetrties of the camera unit in ThreatGet

$O(n)$. ThreatGet shows visible quality and performance recovery after the development point.

An essential part of automation is controlling the data flow. One of the ways to gain control is to start with an object and then track the connection, so it is possible to apply algorithms such as Dijkstra and Kruskal. Tracking was not possible until now because the standard rule of EA does not support that. Objects have been assigned a meaningful name and ID from a number when created, but the connection is created with a new name and a valid ID. The issue handler was used for this problem, which makes it possible to manipulate the elements or connections as they are created, so the connections are named "Connection from [Source Object Name] to [Targets Object Name] Figure 3.1 shows an example without an event handler. The EA event handler is visible in Figure 4.11.

Exporting charts as JSON or XML, therefore, provides an optimal way to connect the ThreatGet chart or the EA chart to other services. This will create an interface to perform future operations. Therefore, we use our example in Section 3 for other security models or protection profiles. Then we export the graph to test the graph according to

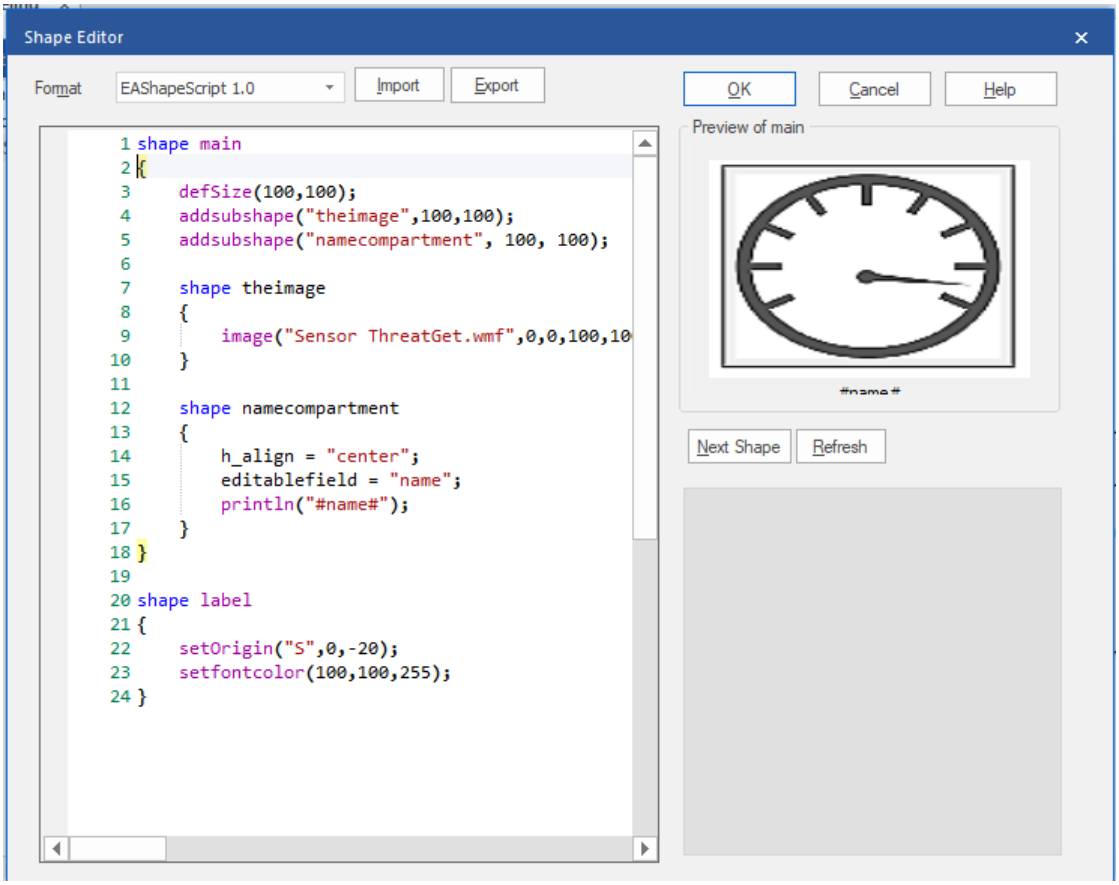

Figure 4.6: Set the Sensor Image to the Sensor Stereotype

the standards

ThreatGet divides the elements in EA into six categories.

- Sensor
- Actuator
- Control Unit
- Communication Interface
- Communication flow
- Datastore

Each of these categories contains certain security information that every element in this category must have in order to successfully perform a ThreatGet operation. An interface

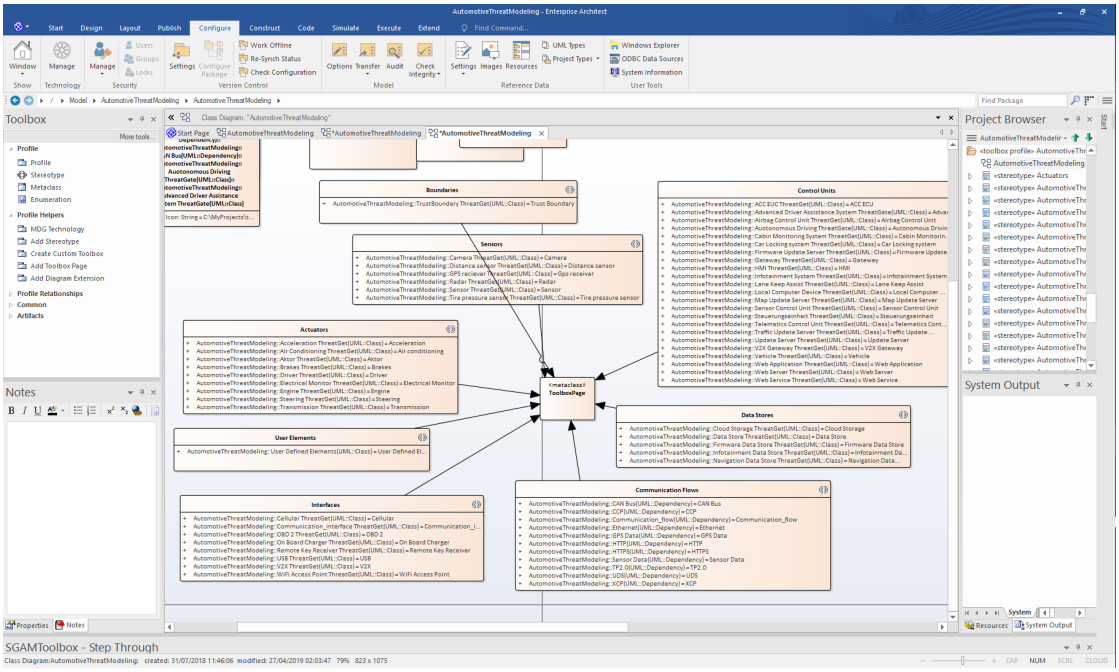

Figure 4.7: ThreatGet Toolbox Page

was created to let the user choose the name, category, and stereotype. The editing and extension of the XML database was a problem. The user input is adjusted in XML format and the database is thereby extended.

Figures 4.12-4.14 show the user interface to create an element and the new created element.

4.2.1 Results

The result of the improvement is most visible in the performance compared to the beginning to achieve precisely the same result. In addition to the performance gain, a more accurate solution has been created. By looking at the objects and connectors, it is now possible to determine their positions, but this helps in determining the additional physical layers. Therefore, if there is an element surrounded by another element, then this is taken into account in the threat analysis. The expansion of the database is an essential step in ThreatGet, ensuring that ThreatGet can be used for various topics in restricted areas. Finally, the user can expand the elements as desired, and thus the user is not restricted to the automotive sector only.

4.2.2 Challenges

The biggest challenge is the lack of documentation. As a result, for example, there is no insight into the existing tables, so the solution is laboriously found in the EA community

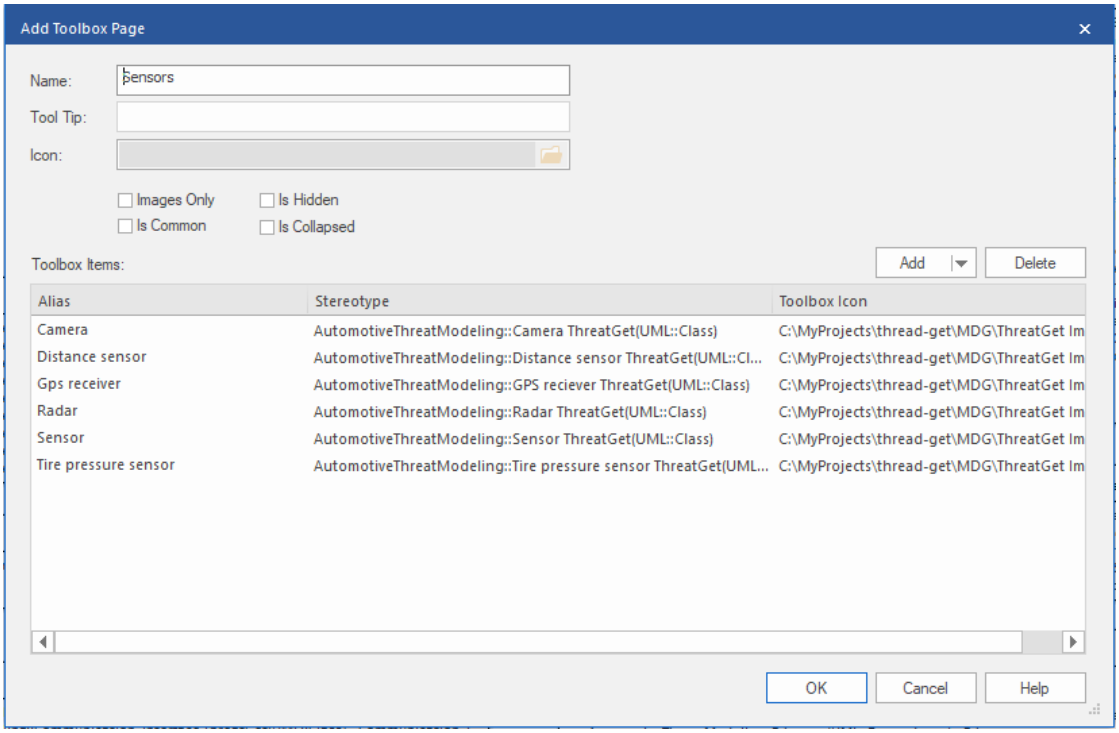

Figure 4.8: Sensor Toolbox Page

or by explanatory research in EA. As an example, consider the comparison between the connectors and elements, so you need the *t_object* table to get general information about the element, *t_objectproperties* to get the element's properties and *t_diagramobjects* to pick the element's positions. For the connectors, on the other hand, the naming is *t_connector* for the general information and *t_connectortag* for the properties table. The mentioned example should show the difficulty with missing documentation.

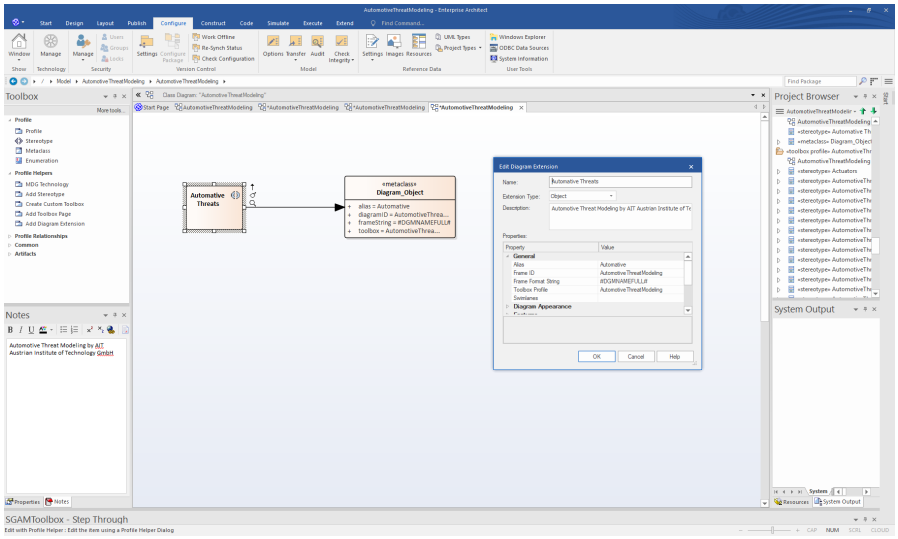

Figure 4.9: TheatGet Diagram Page

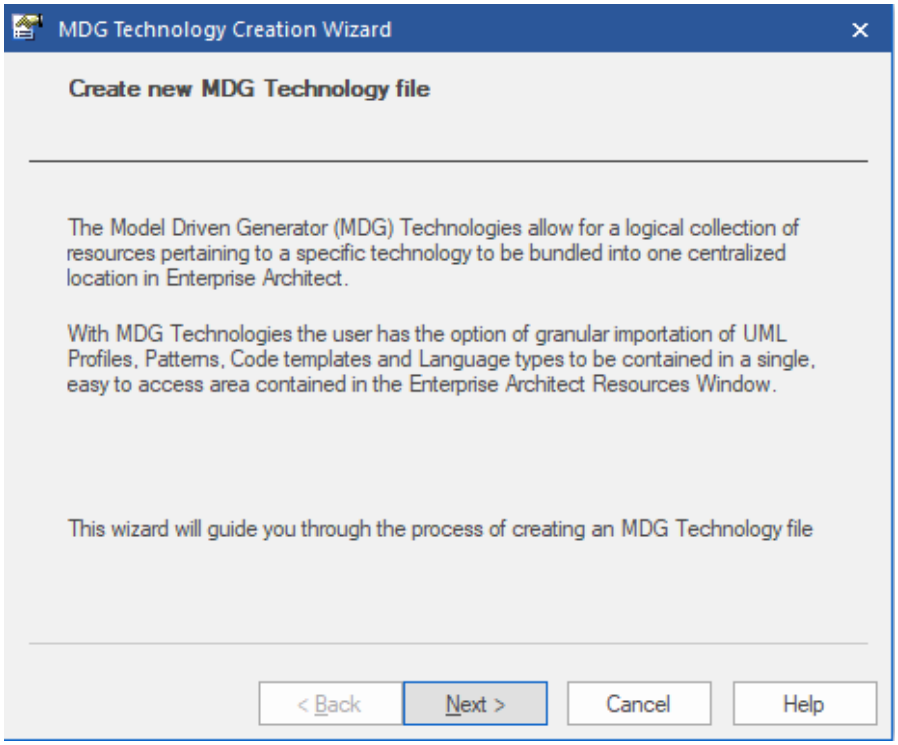

Figure 4.10: MDG Creation

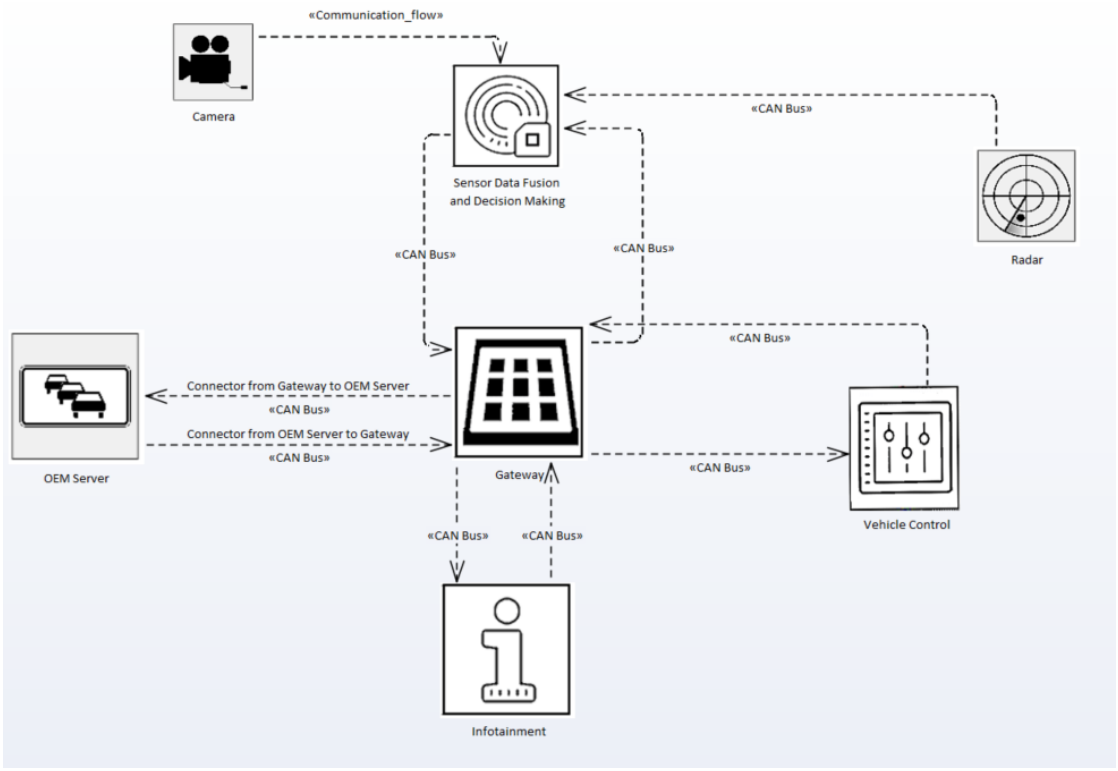

Figure 4.11: Example using an Event handler

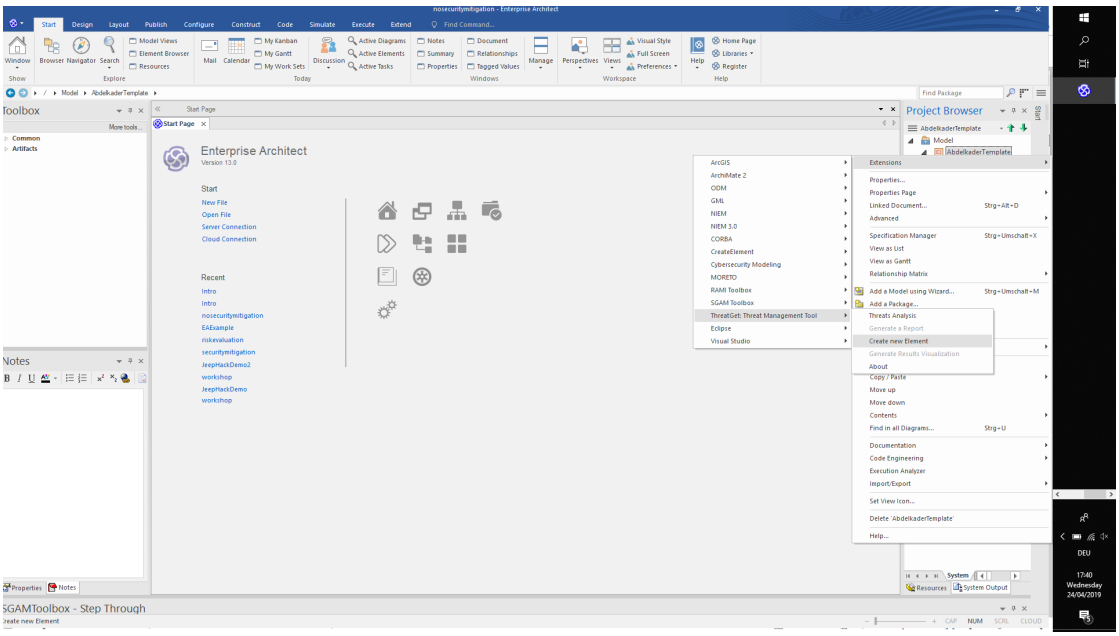

Figure 4.12: Created User-Interface

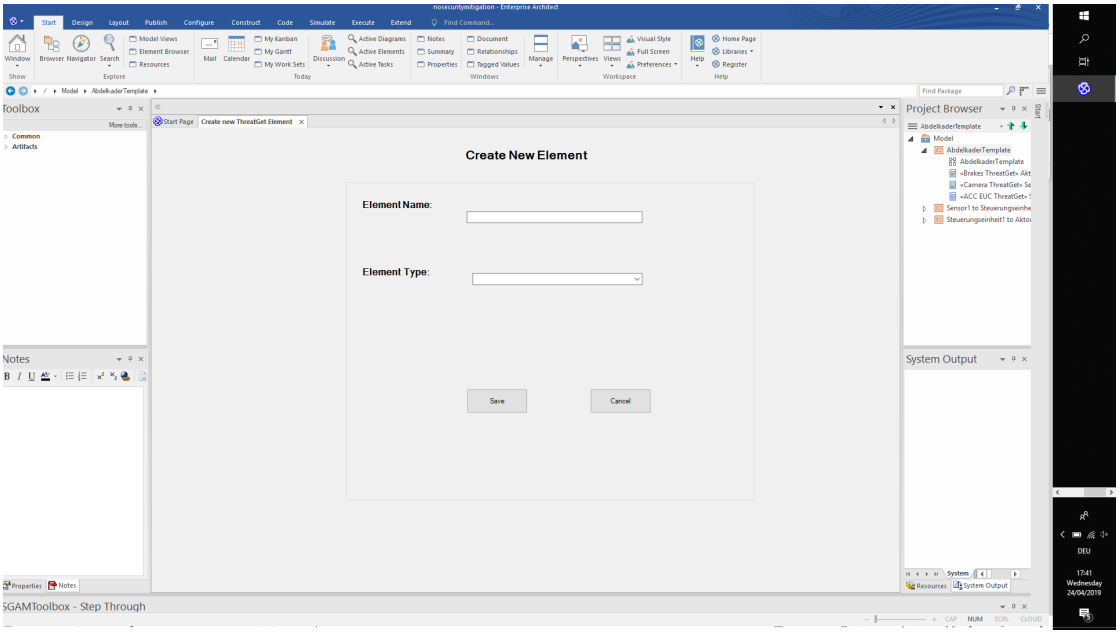

Figure 4.13: The created object is now seen as a normal element of EA

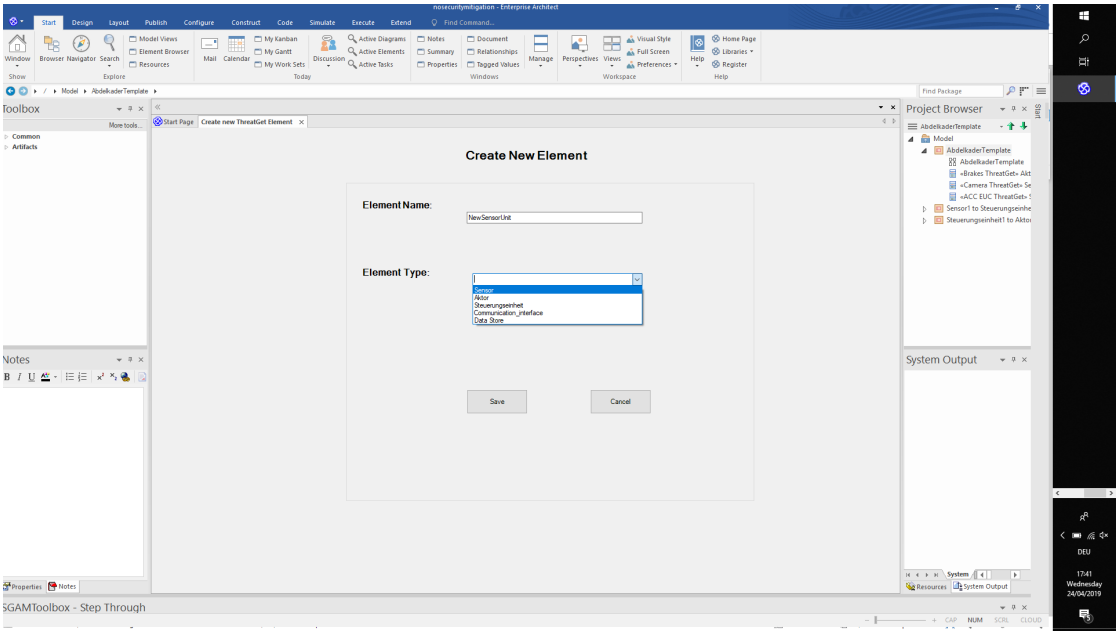

Figure 4.14: The created user-object is also seen as a ThreatGet Element

CHAPTER 5

Protection Profiles in ThreatGet

As already mentioned, the database is already dynamic and therefore extensible. The rules, on the other hand, are still static and are statically filled by the development team. Nevertheless, these two points can be used to declare a rule set which evaluate compliance to a Protection Profile in ThreatGet. A use case is given for the Digital Tachograph - Vehicle Unit (VU PP) from 2016 [2].

The protection profiles provide the security requirements at a very abstract level in order to gain a certain amount of security. For the protection profile, three rules are simulated by using ThreatGet.

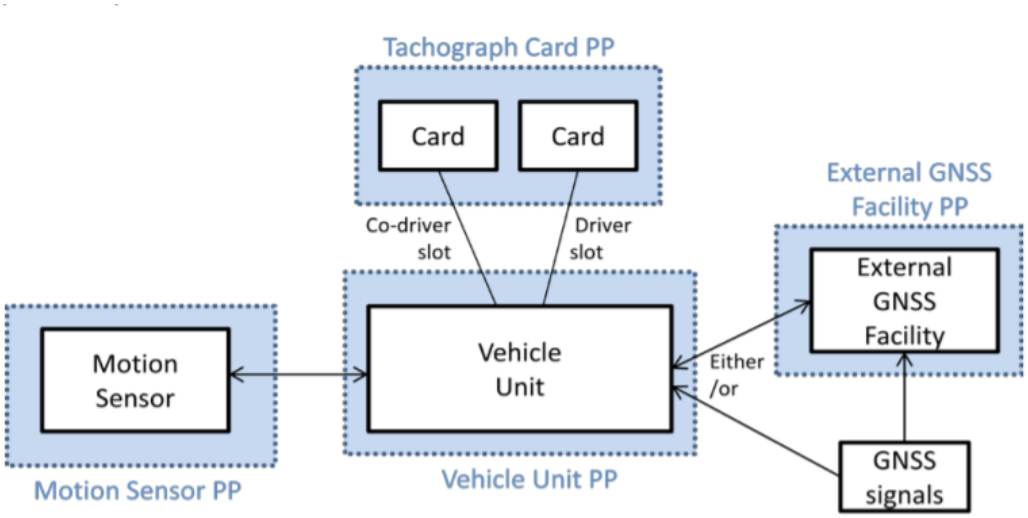

Figure 5.1: Protection Profiles Context[2]

Figure 5.1 shows the abstract version of the protection profile context. Figure 5.2. shows the modeling of Figure 5.1 in EA.

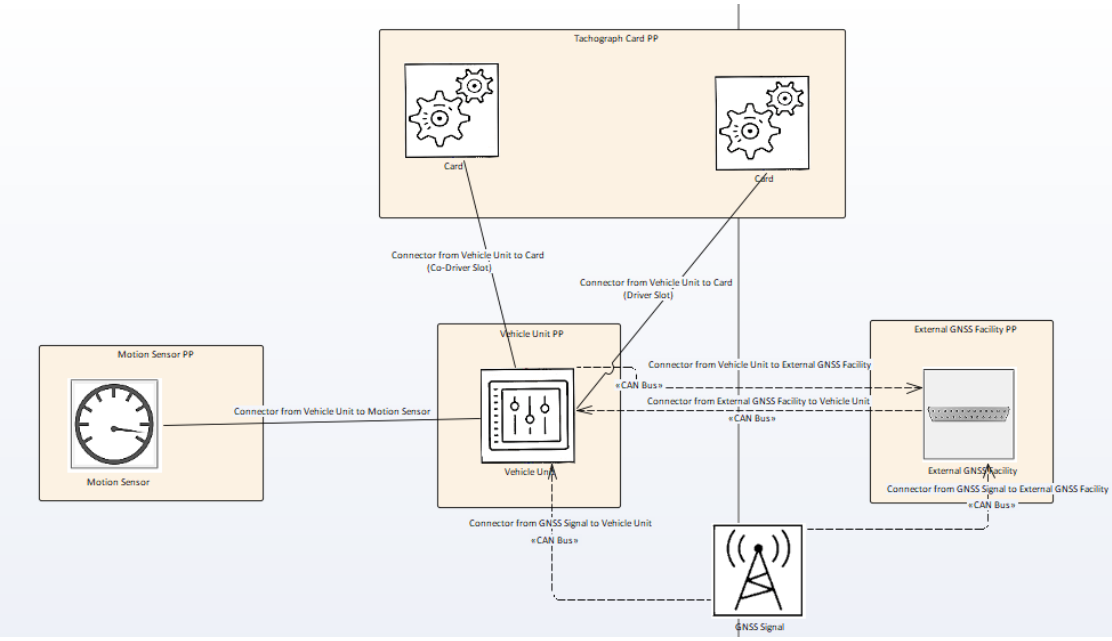

Figure 5.2: Protection Profiles Context in Enterprise Architect with the current ThreatGet Elements

To illustrate how a Protection Profile should work in ThreatGet, part of a rule is modeled in EA and then the ThreatGet operation is executed. This is the rule FDP_ACF.1.2 (5: IS) which will be discussed in the next section.

5.1 FDP_ACF.1.2(5:IS)

"The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:"[2]

- **Rule 1:** "the vehicle unit shall ensure that data related to vehicle motion, the real-time clock, recording equipment calibration parameters, tachograph cards and human user's inputs may only be processed from the right input sources " [2].

In principle, the rule states that communication must be directedto the vehicle unit (VU) and authenticated. Furthermore, the VU must also check that.

Only parts of the rules are examined for the representation by modeling the communication between the real-time clock and the VU. This creates a directed communication from the real-time clock to the VU. If the communication backflies, then another directed

communication must be created. For security reasons, an undirected communication is currently not considered in ThreatGet.

The communication from the real-time clock is illustrated in Figure 5.3, showing the result of the ThreatGet operation in Figure 5.4. As already mentioned, ThreatGet recognizes the type of connection and alerts the user to the two aspects of security. The first is to authenticate the connection and the second to ensure and verify communication from the VU.

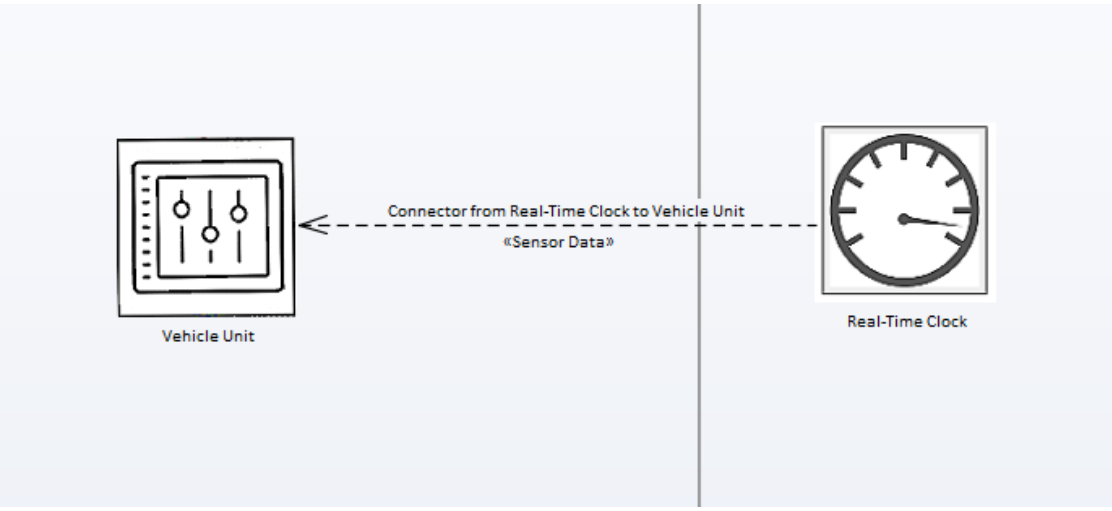

Figure 5.3: Communication between the real-time clock and a VU

5.2 FDP_ETC.2.4

"The TSF shall enforce the following rules when user data is exported from the TOE [2]":

- **Rule 1:** "tachograph cards data update shall be such that, when needed and taking into account card actual storage capacity, most recent data replace oldest data" [2],
- **Rule 2:** "the vehicle unit shall export data to tachograph cards with associated security attributes such that the card will be able to verify its integrity and authenticity" [2],
- **Rule 3:** "the vehicle unit shall download data to external storage media with associated security attributes such that downloaded data integrity and authenticity can be verified" [2].

The section basically describes two rules. The first rule deals with data processing and storage. In the second part and thus the second rule, the communication, integrity and authenticity to the outside area are ensured.

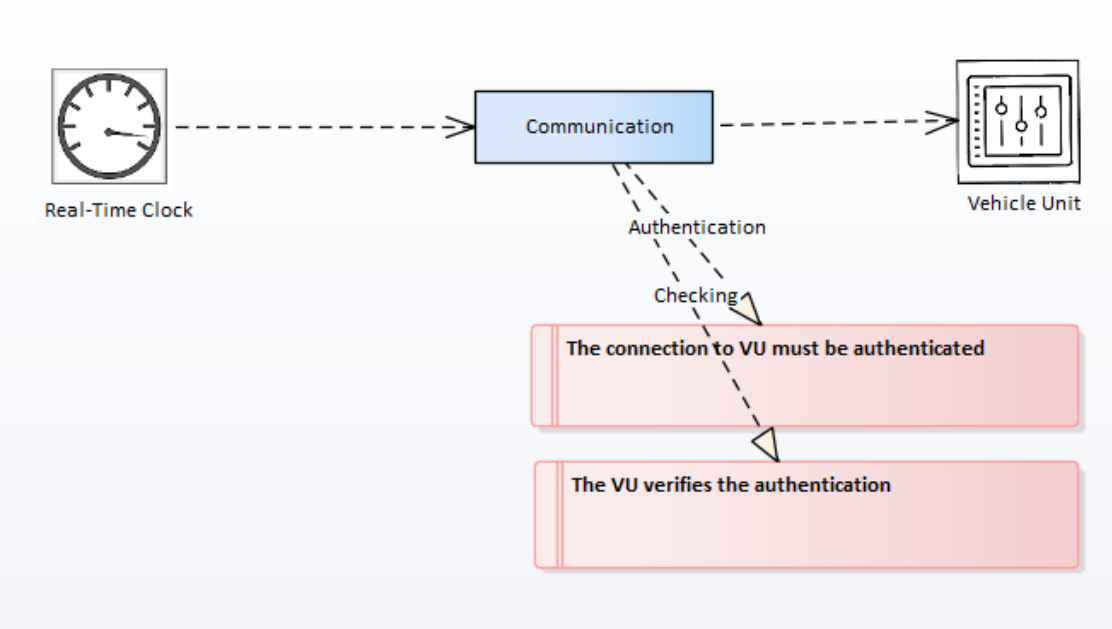

Figure 5.4: Rule 1: Result of ThreatGet operation

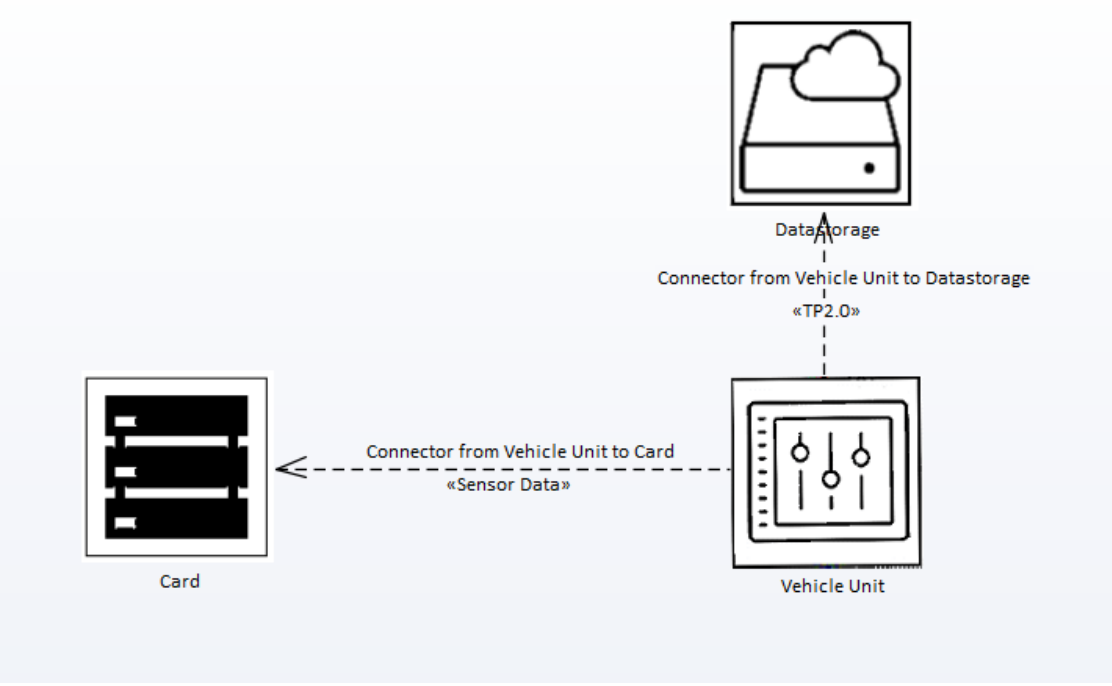

Figure 5.5: The Communication between the VU, Card and the Data Storage

Figures 5.5 shows the modeling of system communication in EA. The VU is connected to a data storage and a card. In executing the operational ThreatGet the compounds are considered separately. The connection between the VU and the data storage shows the Threats "Checking & Replacing". The details are visible in Figure 5.6. Analogously, the second connection of the VU leads to the card. The three discovered threats are illustrated in Figure 5.7.

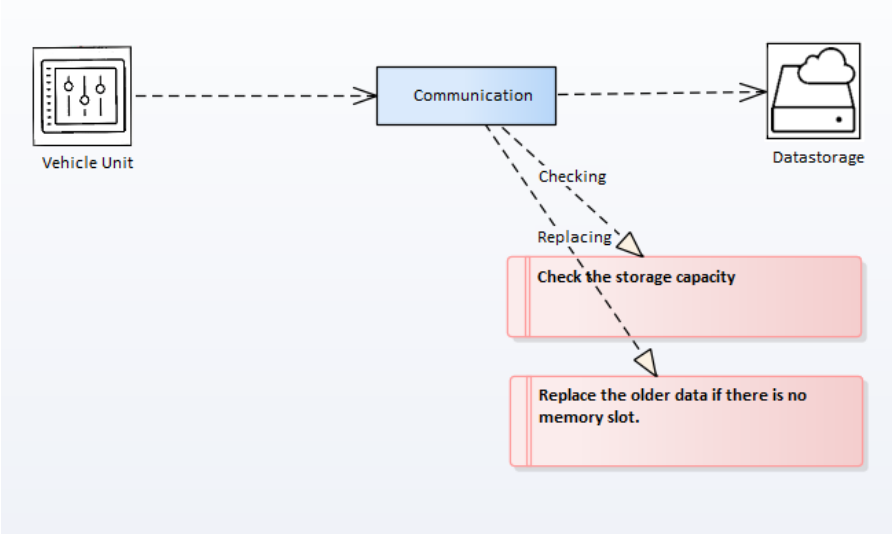

Figure 5.6: Rule 1: The execution and verification in ThreatGet

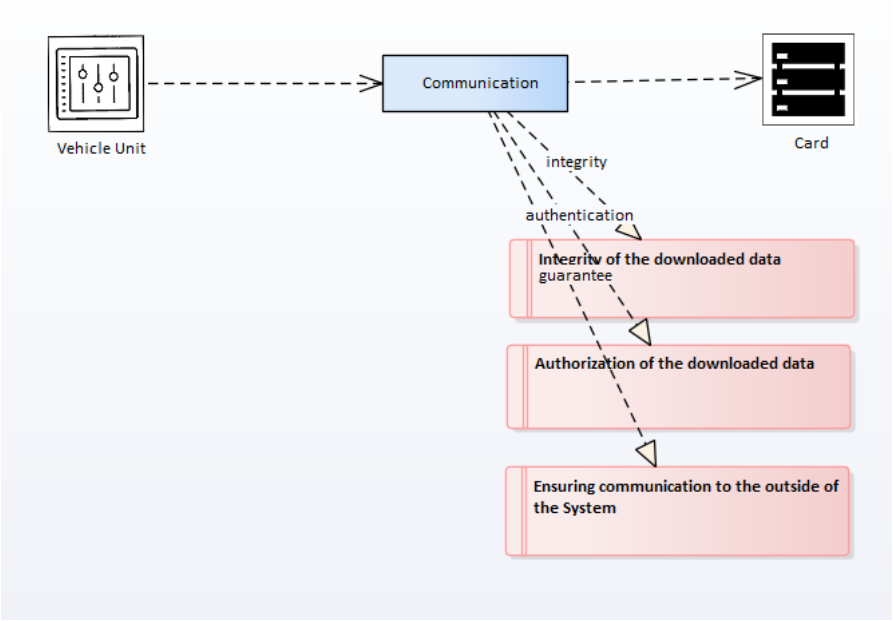

Figure 5.7: Rules 2&3: The execution and verification in ThreatGet

CHAPTER 6

Conclusion

6.1 Summary

Safety and security for critical technical systems must be given in any case. It is not possible to guarantee 100 percent security in all cases, but the goal should be to minimize the threats as much as possible. Checking and adjusting the system to standard security models is a complex topic, especially if the system needs to comply to several standards. ThreatGet provides a database and dynamic development solution because it checks a system for many standards with just a few clicks, and the solutions will be displayed to the user in summary or presented as a report. In the context of this work, the performance has been significantly improved and the application to larger diagrams has become possible. By accessing the data stream of the communication channels, several areas of application opened up in ThreatGet. So it is now possible to use the diagram by using Kruskal or Dijkstra algorithm to connect the shortest possible communication channels. The development of ThreatGet is still on going, but has a lot of potential to check the security for different areas.

6.2 Future Work

Figure 2.2 shows a summary of the results. On the right side of the photo is the impact and likelihood which has not yet been calculated, this is definitely a task for the next version. This is complemented by the use of a dynamic risk analysis. This means that the user can manually change the result by changing the impact or likelihood, which leads to a new calculation and to a new calibration.

Further future work considers the definition of rules. Currently, the data flow is defined by static rules. Another approach for the future is to make the rules dynamic, such that the user has the possibility, to extend the rules to different standard and security criteria.

Thus, it would be theoretically possible to test the same diagram or the modeled system on different ISO and SAE models.

With the expansion of the objects, new applications are opened up in ThreatGet and thus the restriction of the use to autonomous vehicles or a specific ISO or SAE model would be repealed. However, it quickly becomes clear that this will create another problem, the list of objects is infinitely scalable, so the user quickly loses the overview. The problem will be solved in the future by adding up the custom objects. In the future the user will be able to select only one of the corresponding categories, then a new window, or list will open which will contain the objects.

List of Figures

Bibliography

[1] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, et al. Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, volume 4, pages 447–462. San Francisco, 2011.

[2] S. DG JRC – Directorate E – Space, M. Cyber, and D. C. S. U. E3. Bundesamtfür sicherheit in der informationstechnik. .https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0094b_pdf.pdf;jsessionid=CA5887B4393B51E26CEB6EAE4512A60C.1_cid369?__blob=publicationFile&v=3, 2017. [Online; accessed 01-Mai-2019].

[3] C. C. for Information Technology Security Evaluation. Introduction and general model. Version 3.1 Revision 4, CCMB-2012-09-001, 2012.

[4] Information-Technology and Security-Techniques. Iso/iec 27005 - information security risk management. Third edition, 2018.

[5] E. Lisova, I. Sljivo, and A. Causevic. Safety and security co-analyses: A systematic literature review. *IEEE Systems Journal*, 2018.

[6] Z. Ma and C. Schmittner. Threat modeling for automotive security analysis. *Advanced Science and Technology Letters*, 139:333–339, 2016.

[7] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner. Sahara: a security-aware hazard and risk analysis method. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 621–624. EDA Consortium, 2015.

[8] U. T. F. on Cyber security and O. issues (CS/OTA). Cs/ota ad hoc threats 2. Website, 2017. https://wiki.unece.org/pages/viewpage.action?pageId=45383725; [Online; accessed 15-April-2019].

[9] C. Schmittner, M. Latzenhofer, A. Shaaban, and M. Hofer. A proposal for a comprehensive automotive cybersecurity reference architecture. The Seventh International Conference on Advances in Vehicular Systems, Technologies and Applications, 2018.

[10] SparxSystems. Enterprise architect. .https://sparxsystems.com/enterprise_architect_user_guide/14.0/modeling_tools/create_toolbox_profiles_using_.html, 2004. [Online; accessed 15-April-2019].

[11] SparxSystems. Enterprise architect. .http://sparxsystems.com/products/ea/./, 2018. [Online; accessed 30-October-2018].

[12] M. Steger, M. Karner, J. Hillebrand, W. Rom, and K. Römer. A security metric for structured security analysis of cyber-physical systems supporting sae j3061. In *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*, pages 1–6. IEEE, 2016.

[13] F. Swiderski and W. Snyder. *Threat Modeling (Microsoft Professional)*. Vol. 7. Microsoft Press, 2014.