Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 319–329
DOI: 10.15439/2024F5017
ISSN 2300-5963 ACSIS, Vol. 39

Automotive Cybersecurity Engineering with Modeling Support

Alexander Fischer
0009-0001-1737-7395
Nuremberg Institute of Technology,
Nuremberg, Bavaria, Germany
Email: a.fischer@th-nuernberg.de

Juha-Pekka Tolvanen
0000-0002-6409-5972
MetaCase,
Jyväskylä, Finland
Email: jpt@metacase.com

Ramin Tavakoli Kolagari
0000-0002-7470-3767
Nuremberg Institute of Technology,
Nuremberg, Bavaria, Germany
Email: ramin.tavakolikolagari@th-nuernberg.de

***Abstract*—Rapid advances of connected and autonomous vehicle technology have led to an increase in cyber-attacks. This in turn has driven the development of the ISO 21434 standard aimed at supporting the management of cybersecurity risks in the automotive industry. There is, however, a disconnect between the standard and the currently applied model-based development approaches that are increasingly applied for systems and software development. In this paper, we present tool support created for model-based automotive cybersecurity engineering. This tool is built upon the existing automotive systems development language, EAST-ADL, with extensions to address security in accordance with the ISO 21434 standard covering modeling support, calculation of security-related metrics such as impact, risk, and attack feasibility, and generation of ISO 21434 compliant security threat reports. Meeting the requirements of cybersecurity engineeering according to ISO 21434 are demonstrated with two examples.**

I. Introduction

THE DIGITIZATION and networking capabilities of modern vehicles require appropriate cybersecurity measures. As vehicles become more advanced, the risk of cyber-attacks increases, making it essential to identify and assess vulnerabilities in order to implement effective countermeasures. The ISO/SAE 21434:2021 standard "Road Vehicles–Cybersecurity Engineering" [1] provides appropriate means for identifying and assessing risk in the automotive industry, providing guidelines for identifying and assessing potential threats.

The importance of cybersecurity in the automotive sector is underscored by real-world incidents where attackers have exploited vulnerabilities to gain unauthorised access to vehicles (see Section II). These cases highlight the ease with which vehicles can be compromised due to insecure encryption systems or societal underestimation of the risk of an attack. In addition, the introduction of new connectivity features, such as infotainment systems, introduces additional attack vectors and presents new challenges in securing vehicle systems.

Model-based engineering is emphasised as the state-of-the-art approach in automotive software development. This methodology uses models to represent different aspects of the system, enabling the design, analysis and validation of complex systems. We describe the role of model-based approaches in more detail in Section III. Today, models of system and software development are typically kept separate from security models. Yet, integrating cybersecurity into the overall system design is critical, especially with the increasing reliance on software components and the development of autonomous vehicle systems discussed in more detail in Section IV. Collaboration between system and security engineers is necessary to implement security-by-design principles. Models facilitate this collaboration by ensuring traceability of system functions and requirements, defining security objectives and analysing vulnerabilities.

The Security Abstraction Model (SAM, see Section VII) was developed with a focus on modelling cybersecurity threats and measures for automotive systems engineering. Recently, its scope has been extended to align with the ISO/SAE 21434 standard (see Section VI), which addresses cybersecurity in road vehicles. Although SAM offers a robust foundation for ISO/SAE 21434 by providing a metamodel that supports metric calculation and security threat reporting, there has been no tool support available for it until now.

The goal of this research is to show that tool support for SAM and model-based development is possible and can meet the requirements of ISO/SAE 21434 standard. We describe how tool support was developed as well as how it is applied. Our tool support is implemented in MetaEdit+ that enables collaborative development between systems and security engineering, see Section V. The developed tool features for cybersecurity engineering include in addition to modeling, calculation of security-relevant metrics such as impact, risk and attack feasibility, and the creation of ISO 21434-compliant reports. We demonstrate and show that the created modeling tool is viable and align with ISO 21434 with examples in Section VIII. In addition, many of the features described in ISO 21434 are also specified in other cybersecurity standards, so the security attack modeling components from SAM, including any that deal with social engineering attacks, are applicable to many relevant standards.

II. Security Relevance

The increase in cyber-attacks on vehicles reveals a need for action in the automotive sector to protect system components from external attacks. In particular, cases in which attackers were able to gain access to vehicles and start the engine by transmitting the radio key signal have been reported on by public media [2]. Vehicle owners must be made aware that

unlocking doors is easier than generally assumed and that even major manufacturers have been using insecure encryption systems for years [3]. In addition, the secrecy of data sheets does not ensure greater security, but makes thorough security verification more difficult [4]. The increasing connectivity of vehicles and the introduction of convenience features such as e.g. infotainment systems provide further attack vectors that lead to new challenges in identifying and addressing vulnerabilities [5].

Moreover, the trend towards vehicle-to-everything (V2X) communication, which includes vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) interactions, introduces additional risks. These systems rely on the exchange of information between vehicles and infrastructure to improve traffic flow and enhance safety. However, they also create new opportunities for attackers to intercept and manipulate data, potentially leading to collisions or traffic disruptions.

The proliferation of electric vehicles (EVs) also brings unique cybersecurity challenges. EVs often come with connected charging stations, which can be targeted to disrupt charging infrastructure or gain access to the vehicle's internal network. This not only affects the availability of charging but also raises concerns about the potential for large-scale attacks on the power grid.

In response to these threats, it is crucial for the automotive industry to adopt a multi-layered security approach. This includes implementing robust encryption methods, regular security updates, comprehensive testing of all systems, consideration of social engineering attacks, and an integration into the state-of-the-art automotive software systems development approach, i.e., model-based engineering, see the following section.

III. Model-Based Development in the Automotive Domain

A key advantages of model-based development is its ability to enhance communication and collaboration among development teams. By using models, engineers from different disciplines (such as software, hardware, security, and systems engineering) can work together more effectively, sharing a common understanding of the system under development. This collaborative development approach helps to reduce errors and misunderstandings, leading to higher quality software and faster development cycles.

Model-based engineering currently represents the state of the art in the field of automotive software engineering. The primary reasons for model-based approaches are managing complex engineering tasks in better ways and effective communication [6]. In addition to support for collaboration it makes possible to design, analyze and validate complex systems by using models that represent different aspects of the system. It has proven to be extremely effective in supporting the development of automotive software as it enables the systematic design and analysis of functions [7]. The models are typically created with some general-purpose modeling language like UML or SysML [8], [9], or with domain-specific languages targeting automotive systems like:

- Architecture Analysis & Design Language (AADL) [10] is a domain-specific language used for modeling the architecture of embedded systems, including automotive systems. It allows for the representation of both software and hardware components and their interactions. AADL is beneficial for performing performance analysis, such as timing and resource utilization critical in automotive applications.
- AUTOSAR (AUTomotive Open System ARchitecture) [11] is a standardized automotive software architecture framework that allows for the design and development of vehicle software with interoperability and scalability. It defines a set of specifications for software architecture, enabling the integration of components from multiple suppliers. AUTOSAR models are used to specify software components, their interfaces, and communication patterns, ensuring consistency and compatibility across different ECUs (Electronic Control Units).
- EAST-ADL (Electronics Architecture and Software Technology — Architecture Description Language) [12] is a domain-specific language tailored for automotive electrical and electronic systems. It provides a framework for modeling the architecture of vehicles, focusing on requirements engineering, functional analysis, dependability, and system design. It covers a more abstract design level compared to AUTOSAR. EAST-ADL supports the development process by linking requirements to design models and analysis tools, facilitating traceability and verification.

It is worth noticing that these well-known modeling languages applied in automotive do not recognize security, cybersecurity or support for ISO/SAE 21434:2021 standard (see languages used in the automotive industry [10], [11], [12]). We see that model-based development provides a basis for also supporting cybersecurity engineering and it can be done with an integrated manner. We introduce The Security Abstraction Model (SAM) in more detail in Section VII, as an extension to the EAST-ADL providing an integration and traceability between models of system development and cybersecurity.

IV. Need for Integration of Security Design into Systems Modeling

The increasing use of software components instead of mechanical components in vehicles and the development of autonomous vehicle systems require robust cybersecurity measures. Models allow system engineers and security engineers to collaborate and thus put the principle of "security-by-design" into practice. This collaborative modeling approach ensures that security considerations are embedded from the very beginning of the design process, rather than being retrofitted after the fact.

[13] gives an overview of the following advantages of the integrated approach:

- Models provide a structured way to document and trace system functions and requirements throughout the development lifecycle. By incorporating security requirements alongside functional requirements, engineers can ensure that security is treated as a core aspect of the system. This traceability allows better management of dependencies and the identification of potential security impacts arising from changes in system functionality.
- Security objectives need to be clearly defined to protect critical system assets and ensure the overall safety and privacy of vehicle occupants. Models can help in articulating these objectives in a precise manner, providing a clear roadmap for implementing necessary security measures. This includes specifying access control policies, data protection mechanisms, and secure communication protocols.
- To defend against potential cyber-attacks, specific security measures must be integrated into the system design. Models facilitate the systematic design and evaluation of these measures. For instance, threat modeling techniques can be used to identify potential attack vectors, and countermeasures can be designed and validated within the model. This proactive approach helps in mitigating risks before they materialize in the physical system.
- Continuous vulnerability analysis is crucial for maintaining the security of automotive systems. Models enable the simulation and analysis of various attack scenarios, helping engineers to understand the potential impact of different vulnerabilities. By analyzing these scenarios within the model, engineers can prioritize vulnerabilities based on their severity and likelihood, and implement appropriate mitigation strategies.
- The automotive industry is subject to stringent regulatory requirements regarding safety, e.g., ISO 26262, and security, e.g., ISO 21343. Integrating security within the design models ensures that the development process aligns with these regulations and industry standards. This alignment is essential for achieving certification and ensuring that vehicles meet legal and market requirements.

V. Need for Tool Support

Collaborative development work creating specifications, analyzing, checking and versioning them as well as transforming models to code, reports etc. requires tool support. In this paper we apply MetaEdit+ tool [14] to create and use modeling support for cybersecurity. MetaEdit+ is applied because it already supports existing automotive system development languages such as EAST-ADL and AUTOSAR. Second reason for using MetaEdit+ is that it can generate code directly from the models as well as allows creating generators for various purposes other then producing code, like checking, reporting, as well as producing input to other tools like simulators and analysis tools. This function not only provides considerable time and cost savings in development effort, but also improves the overall quality of the system developed.

Thirdly, and crucial for our work on security modeling, MetaEdit+ can extend and combine languages via metamodels, as well as create new domain-specific modeling languages. This flexibility allows for the customization of modeling languages to suit specific domain requirements. Once a metamodel is defined, developers can use it as their domain-specific language for modeling [15].

In Section VIII we describe how modeling support was created by defining security-related language concepts, rules and notation. We also present the generators that calculate security scores and produce relevant security documents as in ISO 21434. We demonstrate resulting tool support with examples.

VI. ISO Standard 21434

ISO/SAE 21434 contains objectives, requirements and guidelines related to cybersecurity engineering and can be used to implement a cybersecurity management system that also involves cybersecurity risk management [1]. The standard specifies the technical requirements for managing the cybersecurity risk of electrical and electronic systems (E/E-Systems) in road vehicles, including their components and interfaces. No specific technologies or solutions for cybersecurity are prescribed. ISO/SAE 21434 mandates risk treatment for all identified risks using classical options: risk avoidance, reduction, sharing, or retention and permits risk acceptance up to a defined threshold, as long as the decision is documented along with the retained risks [16]. According to ISO 21434, road vehicle cybersecurity is achieved when assets are adequately protected against threat scenarios. Assets worthy of protection include the various tangible and intangible components of systems such as software and hardware components, sensitive information and communication links. Threat scenarios are the potential cause for the compromised protection objectives of one or more assets [1]. ISO 21434 defines item as one or more components that implement a function at vehicle level, whereby a component is defined as a logically and technically separable part [1]. The item definition defines the target development system, which is subject to a cybersecurity-oriented development process, as precisely as possible and specifies the physical limits of the system under consideration as well as the areas to be protected. Based on the item definition, a threat analysis and risk assessment (TARA) is carried out from the perspective of affected road users. It serves to systematically identify threats and analyze the attack and defense mechanisms in the examined system and essentially consists of the following elements:

1) Item Definition [1, section 9.3]
2) Asset Identification [1, section 15.3]
3) Identification of Threat Scenarios [1, section 15.4]
4) Impact Rating [1, section 15.5]
5) Attack Path Analysis [1, section 15.6]
6) Attack Feasibility Rating [1, section 15.7]
7) Risk Value Determination [1, section 15.8]
8) Risk Treatment Decision [1, section 15.9]
9) Cyber Security Goals [1, section 9.4]

10) Cyber Security Claims [1, [RQ-09-06]]
11) Cyber Security Concept [1, section 9.5]

Cybersecurity engineering analysis identifies and explores potential actions that an abstract attacker could perform maliciously and the damage that could result from compromising the cybersecurity of a vehicle's E/E systems. Cybersecurity monitoring, remediation and incident response depend on changing environmental conditions, i.e. there is a constant need to identify vulnerabilities in road vehicle E/E systems and counteract new attack techniques.

The abbreviation CAL stands for Cybersecurity Assurance Level and, similar to the ASIL (Automotive Safety Integrity Level) in the ISO 26262 standard, is used to appropriately adjust the effort and care required for subsequent activities in the area of cybersecurity. The ISO/SAE 21434 standard specifies that an appropriate CAL should be defined for each threat scenario based on the associated impact and attack vectors. This is similar to setting risk values. While the risk value is dynamic and can change during the development process, the CAL is intended to remain stable during development as it is an integral part of a development requirement.

VII. Security Abstraction Model with Extensions

Security Abstraction Model (SAM) provides concepts for modeling security aspects of automotive systems. Figure 1 describes the metamodel of SAM illustrating which kind of security aspects are specified when modeling automotive systems with security considerations. In this figure, we present the complete metamodel so that the relationships between the entities become visible, as this is relevant for the reporting described later.

Originally SAM [17] did not recognize the later published ISO 21434 standard but this is now integrated into SAM and its metamodel [18]. This creates a link between the security requirements of the ISO 21434 standard and the models created based on SAM. Similarly, SAM was not originally developed explicitly for modeling social engineering but an extension has been developed that enables the modeling of social engineering attacks and maps the relationship of these attacks to the actors and the rest of the model [13]. These extensions enable a more comprehensive specification of cybersecurity aspects, their reporting as in ISO 21434 and calculating related metrics and scored. We describe these extensions in the next subsections, and their implemention to the modeling tool in Section VIII.

A. Integration with System Design

EAST-ADL [12] is a language for describing the system architectures of software-intensive automotive systems using an information model that represents technical information in a standardized way. The descriptions cover vehicle functions and features as well as functional and hardware architecture. The EAST-ADL model is structured according to abstraction levels, with each sub-model representing the relevant details of the complete embedded system of the respective abstraction level.

Security Abstraction Model and EAST-ADL are linked by the common concept of item. In EAST-ADL, item represents a functional or non-functional requirement of the system that is being described and modeled. SAM extends the concept of item by incorporating security properties. This enables SAM to specify security requirements that are necessary to fulfill the overall system requirements. These security requirements are integrated into the model to enable a comprehensive security analysis and to identify potential vulnerabilities and threats in the system.

Although SAM is developed as part of the EAST-ADL, it is not necessarily bound to EAST-ADL, offering flexibility in its application. SAM can be used independently of the rest of the system model to provide an overview of security-critical system parts before or at the beginning of the system engineering process. This independent utility allows engineers to identify and address potential security vulnerabilities early in the development cycle.

B. Scores

The latest version of SAM used at the time of writing includes a number of entities from ISO 21434 to enable a detailed risk assessment. These entities include Asset, Damage Scenario, Threat Scenario, ImpactRatingScore, RiskScore, AttackFeasibilityRating and AttackFeasibilityScore (see Figure 1). By integrating ISO 21434, not only can vulnerabilities now be assessed, but so can potential attacks and their impact on the system. For this purpose, the AttackFeasibilityScore, ImpactRatingScore and RiskScore are included in SAM. The AttackFeasibilityScore is calculated on the basis of the CVSS formula and makes it possible to estimate the feasibility of an attack [1].

In addition to the previously mentioned scores, ISO 21434 contains further scores that are available in the SAM metamodel. The ImpactRatingScore evaluates the impact of an attack scenario based on various factors such as the severity of the damage caused, the extent of the impact on the system and the potential duration of the impact. The RiskScore assesses the overall risk associated with a particular threat scenario. It takes into account the probability of a successful attack and the possible consequences of the attack. The RiskScore makes it possible to prioritize potential threats and take appropriate security measures to reduce or control the risk. The combination of these scores in SAM enables a more comprehensive security analysis and risk assessment. By taking into account vulnerabilities, attack scenarios, ImpactRatingScore and RiskScore, emerges a holistic picture of the security of a system in the automotive context. This makes it easier to identify potential risks and threats and take appropriate measures to increase the security and reliability of the system.

C. Social Engineering

SAM provides a basis for the assessment of social engineering attacks by including various scores and entities. A qualitative scoring system has been developed to specifically focused on social engineering. Integrating a scoring system

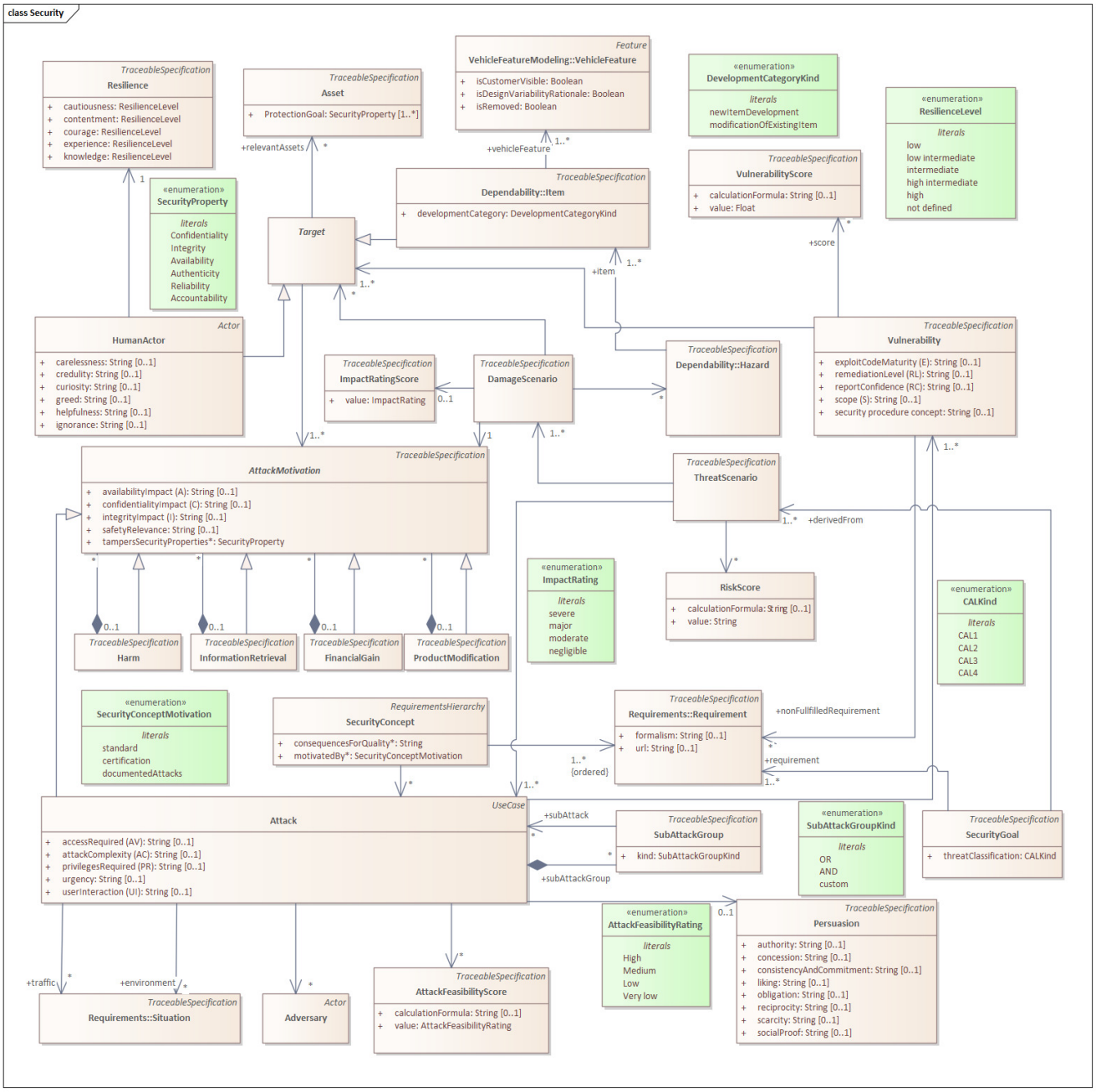

Fig. 1. SAM metamodel. View Online at https://bitbucket.org/east-adl/sam/

further improves the assessment and understanding of risks associated with social engineering, helping to develop appropriate security measures to minimize the impact of these attacks. Extensions to SAM were implemented at different levels, including new meta-entities, extensions to existing meta-entities, and supplementary documentation, enabling greater consideration of social engineering and standards.

Due to the existing integration of CVSS scores and other assessments in the Security Abstraction Model, it was necessary to investigate whether the extension and harmonization would create redundancy. Redundancy is beneficial if there are reasons for mapping an issue in different ways. The social engineering entities were integrated into the metamodel to ensure a clear capture of security aspects without creating unnecessary duplication or repetition of metrics.

D. Reporting

Reporting is essential in the context of ISO 21434, which explicitly requires it. Specifically, a cybersecurity assessment report (RQ-06-31) serves as appraisal of the level of cy-

bersecurity. Although the standard does not provide explicit guidelines on the format or structure of such reports, our implementation of the report generation adheres closely to the principles described in ISO 21434. This ensures that the cybersecurity assessment report effectively communicates the findings and recommendations derived from the assessment process.

In accordance with ISO 21434, the report is primarily focused on assets, reflecting the standard's emphasis on asset-oriented cybersecurity management. Its structure is closely based on the example in Annex H of the standard. However, by integrating the social engineering aspect of SAM, we have also introduced reporting that focus on human actors and recognize the importance of the human element in cybersecurity. In addition, we have included a section dealing with miscellaneous items and how they relate to vulnerabilities and the associated vulnerability scores. While the report sections relating to social engineering and miscellaneous are not explicitly included in the standard, their inclusion broadens the scope of the report and provides stakeholders with a holistic reporting of cybersecurity risk and mitigation.

There are several advantages to automatic report generation:

- It enables the hierarchical organization of multiple models, facilitating the creation of comprehensive reports that cover different aspects. This hierarchical structuring enables a systematic and coherent presentation of information across different levels of abstraction.
- Automated report generation can incorporate item definitions by linking to EAST-ADL architecture models, providing insight into potentially at-risk vehicle features and their interrelationships. This integration increases the depth and specificity of the report.
- By selecting relevant properties, the calculation and reporting of scores is automatically generated, which ensures efficiency and accuracy. In cases where multiple values are applicable, these are aggregated, with the maximum value being reported.

VIII. Tool Support: Language Definition and Usage

This section presents the tool support for security modeling. We first describe the implementation of support for SAM, including the modeling language, score calculators and the reporting of security threats in accordance with ISO standard 21434. Subsequently, we provide two examples demonstrating the use of the developed modeling tool, alongside score calculation, reporting and tracing to other system design models.

Our implementation of tool support began by extending the existing language definition of EAST-ADL and its associated security language. Although EAST-ADL is supported by various tools, we applied in MetaEdit+ the latest version of EAST-ADL (v2.2)[1]. Since MetaEdit+ enables the co-evolution of metamodels and models [19], the changes made to the modeling support were automatically updated to already existing models.

The language definition covered all parts needed for obtaining tool support: Not only the metamodel and related constraints, but also the notation, guidance for creating and editing models, as well as updating older versions or notifying modelers to make changes when automatic update were not considered feasibly, such as when there was a risk of losing relevant data. Finally, generators for various score calculations and threat security reporting were defined, in addition to those available in MetaEdit+ for EAST-ADL, such as Simulink, HipHops and ReqIF, or defined by users targeting external tools like SPIN, UPPAAL, Stateflow and Reliability Workbench[2].

A. Metamodel Extensions

For modeling support, the metamodel of SAM was defined by two person with MetaEdit+ Workbench, and then tested by other modelers by using the same language with the modeling editors, browsers, and collaboration tools of MetaEdit+. We created several security models as test cases including the Brake-By-Wire example presented here later[3]. Figure 2 shows the elements of the security modeling language in MetaEdit+. The list of Objects shows the key modeling objects, the list of Relationships shows the connections between these elements, and the list of Roles shows how an object participates in the relationships, such as being directed or undirected, having constraints, or detailed properties.

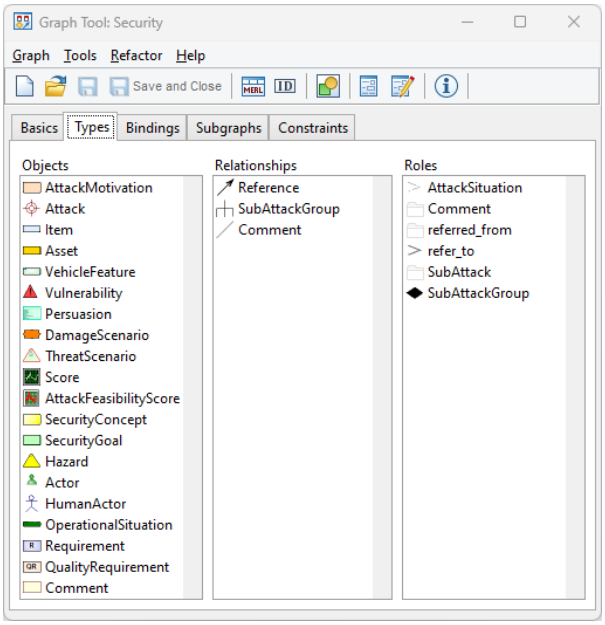

Fig. 2. Extended SAM definition

Figure 3 details the definition of HumanActor, which has 11 properties. The first three are inherited from EAST-ADL

[1]https://east-adl.info/

[2]https://metacase.com/solution/east-adl.html

[3]SAM implementation can be accessed at https://bitbucket.org/east-adl/sam/src/master/MetaEdit-Extension/Reporting_Examples

and AUTOSAR metamodels. These three properties have rules and constraints, such as 'Short name' being mandatory and starting with an alphabetical character followed by possible characters, numbers, or underscores and constraint with maximum length (defined as a regular expression: `[a-zA-Z][a-zA-Z0-9_]{0,127}`). These are followed by the characteristics of HumanActor in SAM (see Figure 1): Curiosity, Helpfulness, Credulity, Greed, Ignorance, and Carelessness – all of which are fixed value enumerations.

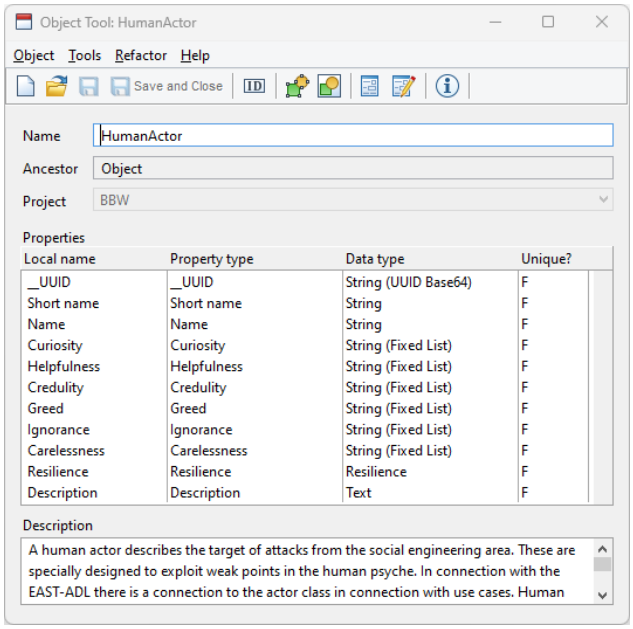

Fig. 3. Definition of HumanActor in metamodel of MetaEdit+

While the metamodel in Figure 1 identifies many language concepts as individual objects, such as resilience or metrics elements, we aimed to minimize the modeling effort not only in terms of creating model elements but also in terms of updating, deleting and checking specifications. As a result, the implementation as a modeling language exhibits some differences from the metamodel illustrated in Figure 1. This is mainly to minimize the modeling effort and improve usability. The main differences are:

- Since Resilience is a mandatory item for a human actor, it is a property of HumanActor. This way language user is expected to add – and later edit – just one element in a model rather than two and a connection between.
- The same approach is also applied for the 4 metrics elements: while they can be added to and visualized in the model, they are not mandatory. Modeling editor can calculate the metrics even if those metric elements are not explicitly added to the model. Figure 4 illustrates this in the user interface at the bottom of the screen by showing individual metric values for vulnerabilities and attacks yet showing CVSS basic and temporal scores for vulnerability as well as ISO 21434 feasibility score for the attack directly in the diagram, which is what the user wanted in this case.
- Attack motivation is an element of the modeling language, and its subtype is selected from the property with mandatory value. Thus, the type of AttackMotivation (Harm, Financial Gain etc.) can be changed without deleting the old one and creating and re-connecting a new one.
- While the metamodel of SAM defines directed associations among security concepts, the modeling language does not expect models to be created in that order: the created editor shows the correct direction regardless of how the user opts to link model items. In other words, the model is created correctly independently of the order in which the modeler decides to create relationships.
- Default values for enumerations are provided.
- Properties of model elements are listed in the order that would be the order that would be most natural for considering the security properties.

We did not enforce all rules as mandatory, such as requiring each HumanActor to have a defined Resilience. Instead, we allowed for more flexibility in modeling, but we also provided guidance to language users to complete the security model. We defined 17 checks derived from the metamodel to provide warnings, which were shown to the language user during modeling. As an example, at the bottom of Figure 4 is shown a warning that SecurityConcept is not related to any Requirement. Additionally, we defined recommendations for creating security models that deal with optional links: linking Attacks to Actors, SecurityConcepts with Attacks, and DamageScenarios to Hazards – the last been shown as a recommendation by the tool for the security model in Figure 4.

B. Notation and Guidance

The security model example also illustrates the notation: How models are presented for humans to read, edit and use for communication. Our tool implementation therefore covered creating notation for the respective language elements. Figure 5 shows the definition of notational symbol for DamageScenario: It shows the name that user enters and impact rating score produced by the score calculator generators. The notation also shows the type of model element as a part of the notational symbol. Such guide is useful when creating or reading the models in the first place but for experienced modelers it becomes redundant text that consumes extra space and thus can be hidden by the language user from the diagrams if desired.

In addition to providing guidance during modeling, the defined metamodel thoroughly describes individual language elements. These descriptions are accessible directly in MetaEdit+ through the help system, which is available from the editor's toolbar or individually for each language element when in use.

C. Co-evolution

Given that SAM itself also evolved, we implemented guidance to update the existing security models to be compliant

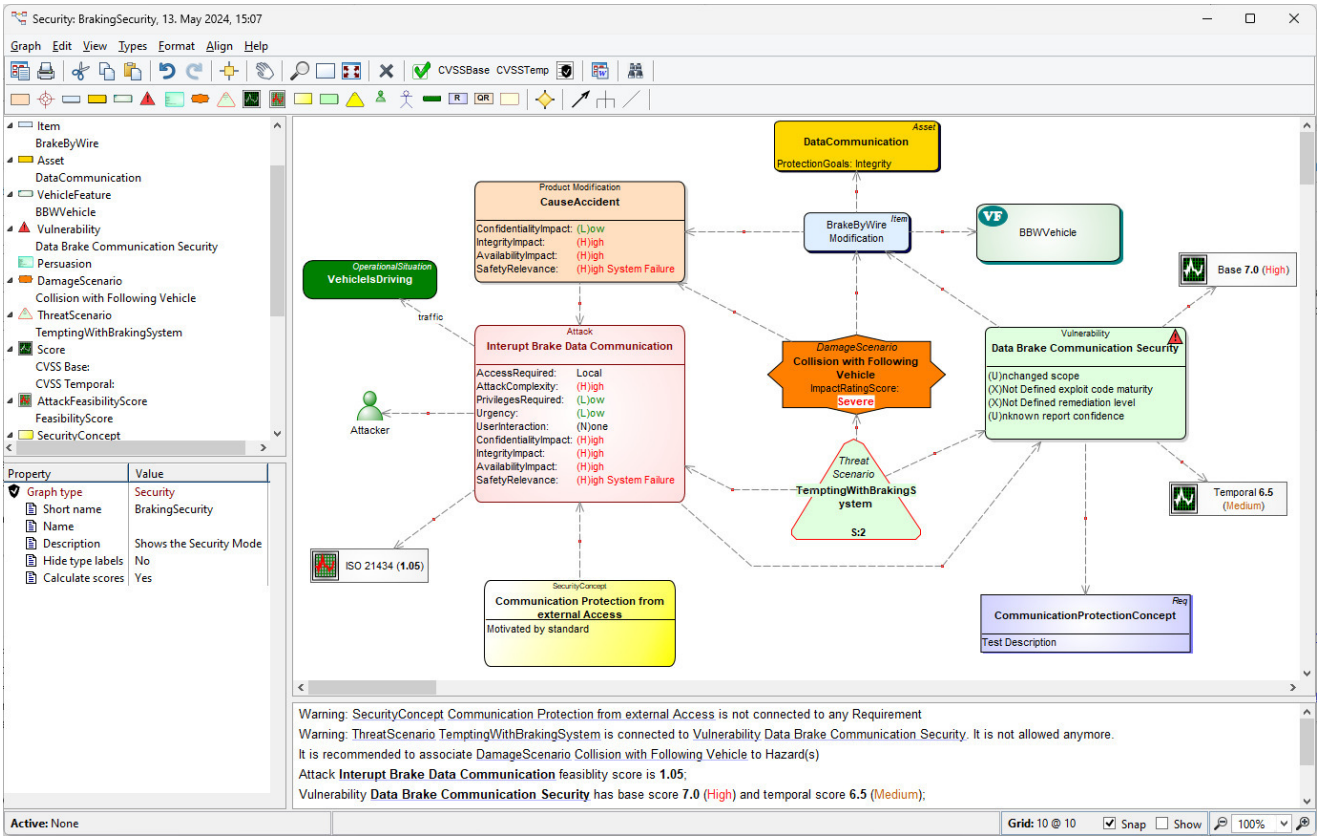

Fig. 4. Security model in modeling editor illustrating checks and recommendations

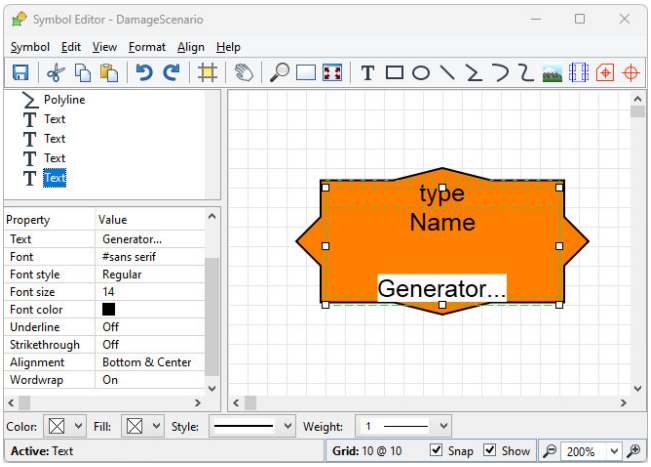

Fig. 5. Defining the notation for DamageScenario

with the latest language version. MetaEdit+ manages with in-built functionality automatically model changes that are caused by adding and renaming items in the metamodel, constraints and notation (for details see [19]). However, some changes to models may require human intervention when automatic updates are not feasible. To prevent the loss of critical model information, we followed a deprecation strategy: existing models can still be used, but all new models follow the latest language version. Additionally, we implemented guidance within the modeling editor to assist users in updating their models. This feature is illustrated at the bottom of the editor in Figure 4: "Warning: ThreatScenario Tempting-WithBrakingSystem is connected to Vulnerability Data Brake Communication Security. It is not allowed anymore." Similar co-evolution support could be applied in the future when support for cybersecurity modeling evolves or new versions of the ISO standard or SAM are developed.

D. Metrics

As models provide rich details on security aspects, they can be used for various assessment purposes. We implemented support for SAM-based security models with the CVSS. Once a security model is created with the required data, the modeling tool calculates various scores automatically, like in Figure 4 for vulnerability of Data Brake Communication Security the score of Base CVSS is 7.0 (High) and Temporal CVSS is 6.5 (Medium) and for the specified attack ISO 21434 score is 1.05.

Since attacks can consist of subattacks, calculating vulnerability metrics must consider the whole attack subtree. In our implementation of CVSS, we considered the most severe case by recognizing the most severe attack within attack tree as a basis for calculation. The same principle is applied when

different types of individual attacks are related to the same vulnerability.

Scores on vulnerability and attack feasibility are calculated similarly at the time of modeling and illustrated either in the diagram or in a separate report pane below the diagram. Figure 4 illustrates scores at the bottom of screen and AttackFeasibilityScore next to the Attack element. The impact rating for Damage scenario (Severe) and risk score for ThreatScenarios (S:2) are also illustrated in Figure 4.

E. Documenting and Reporting

Existing documentation generators were available in MetaEdit+ for the purposes of reporting. These generators, however, did not recognise the needs of ISO 21434. Given that the SAM was made to recognize explicitly cybersecurity, we defined a threat reporting generator based on the reporting requirements (as in Section VII-D).

Figure 6 shows the result of this generator produced from Figure 4 and from the related system design specifying the vehicle features (Figure 7) and the system functions (Figure 8). Figure 7 shows a small part of the model specifying features related to the braking system. These features are realized by some design functions and hardware functions of EAST-ADL. Figure 8 illustrates a part of the logical design functions of the braking system that are also recognized in the generated security report. Both security report and metric calculators were implemented with generator system of MetaEdit+ [14].

Standard ISO/SAE 21434
There are currently 1 Items with 1 Features and 0 Human actors affected by overall 0 Hazards and 1 Attacks.
1 security models are not handled yet (BrakingSecurity).
2 of the Development Phases are affected. Item BrakeByWire is affected by the highest threat.

Item definition
The item BrakeByWire:
is linked to DesignFunctionType BBW_FDA (DesignFunctionArchitecture for Braking from Pedal to ABS) via
is linked to HardwareComponentType HDA (Hardware architecture for braking system.) via feature BBWVehi

Asset identification
Asset DataCommunication is related to DamageScenario Collision with Following Vehicle with the following se

Impact rating
DamageScenario Collision with Following Vehicle has impact rating: Severe

Threat scenario identification
DamageScenario Collision with Following Vehicle is related to ThreatScenario TemptingWithBrakingSystem (

Attack analysis
ThreatScenario TemptingWithBrakingSystem is related to: Attack Interupt Brake Data Communication (Test D

Attack feasibility rating
Interupt Brake Data Communication has attack feasibility rating **1.05**

Risk value determination
TemptingWithBrakingSystem has aggregated attack feasibility rating **1.05** and Impact rating Severe with risk

Targets
Target BrakeByWire is related to Data Brake Communication Security

Vulnerability analysis
Vulnerability Data Brake Communication Security has vulnerability base score **7.0** (High), and
temporal score **6.5** (Medium)

Fig. 6. Sample of Security Analysis Report

Traceability from security models to system design is visible in the security analysis report. For example, in Figure 6 the item definition at the beginning of the report is linked to the design functions and hardware functions of the braking system. Also the summary at the beginning of the generated report shows that security models are related to two different development phases of EAST-ADL, namely to the vehicle level in which features related to items are defined as well as to the design level functions realizing those features.

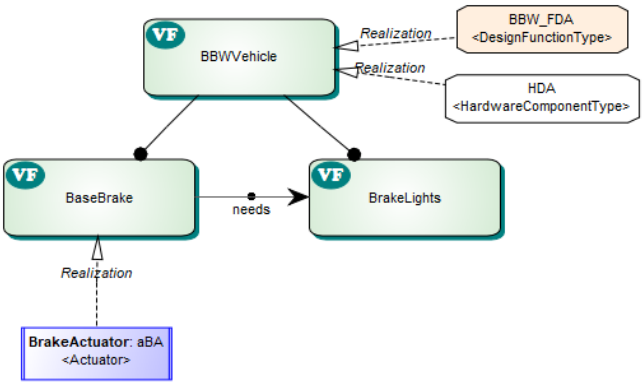

Fig. 7. Vehicle feature model: braking (fraction)

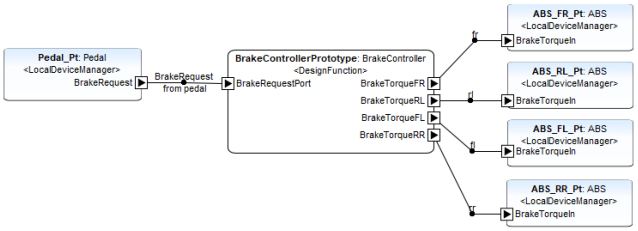

Fig. 8. Design level functions of braking system (fraction)

Figure 9 shows another report targeting analysis of social engineering threats in automotive systems. This report is generated from a security model shown in Figure 10 representing a social engineering attack that affects the braking system. It shows a baiting attack in which the braking system is compromised through deception maneuvers. The report identifies the human actors involved, their vulnerabilities and their resilience to such attacks. Additionally, it provides insights into the persuasion methods used in the social engineering attack, improving understanding of the potential dangers posed by human manipulation tactics. This holistic approach to reporting provides valuable insight into the intricacies of cybersecurity risks associated with social engineering and helps develop robust countermeasures to protect automotive systems from such vulnerabilities.

The reports illustrated in Figures 6 and 9 show that they provide links from reported items back to the security models and other system development models. This clear traceability shows that security aspects do not need to be addressed in isolation, but can be linked to the rest of the system development. These reports can be produced directly to external files like used for word processors or web browsers.

In addition to reporting on individual security model – as illustrated in the previous examples – security threat reporting is also available for all EAST-ADL models: It can be generated

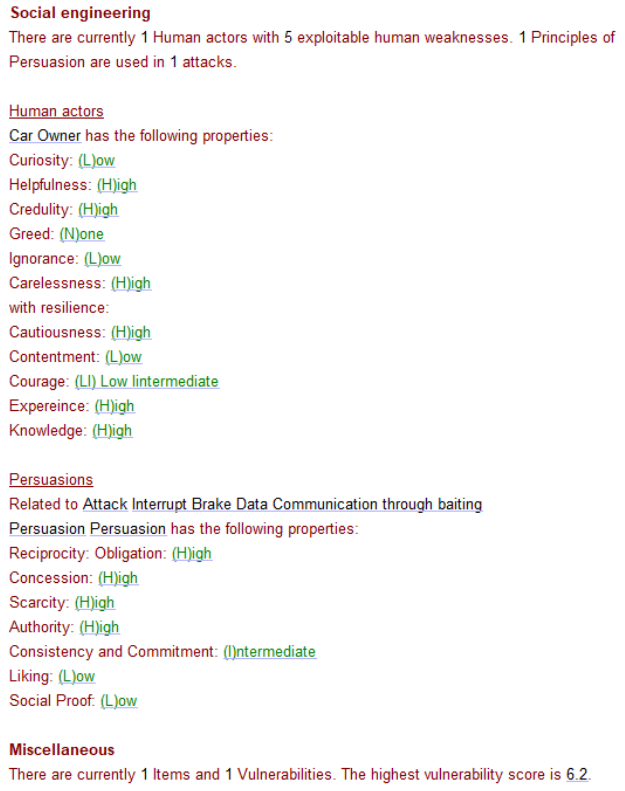

Fig. 9. Sample of social engineering report

for any selected hierarchy of EAST-ADL models combining multiple security models into a single security threat report. This capability enhances collaboration by allowing traces from system designs to be followed to all vulnerabilities and attacks across the entire developed system.

IX. Conclusion

This paper presents a tool support for model-based cybersecurity engineering in the automotive domain. It shows how tool support can meet the requirements of ISO/SAE 21434 standard in model building, calculating metrics and security threat reporting. Our tool, built on the EAST-ADL language with security extensions, provides a solution to support these model-based approaches. By integrating system and security modeling, along with capabilities for calculating security metrics and generating ISO-compliant reports, the tool enables engineers to navigate the complexities of automotive cybersecurity with confidence. Furthermore, the tool's ability to guide engineers in defining and integrating security models with system models underscores its user-centered design and practical utility.

The significance of this work extends beyond its immediate application in automotive cybersecurity. As the latest enhancements to the metamodel enable a complete representation of the ISO 21434 standard, it lays the groundwork for broader adoption across industries where cybersecurity standards are of highest importance. Moreover, the versatility of the extensions, particularly those related to social engineering attacks, positions it as a valuable resource for compliance with various cybersecurity standards beyond ISO 21434.

While the modeling support is readily available our plan is to apply it to model various security cases to evaluate it and identify possible areas for extensions. Another direction for future research is to extend tool support, and possibly the metamodel of SAM, to support the latest versions of metric calculators like version 4.0 of CVSS.

Future research could investigate the use of Large Language Models (LLMs) to automatically generate models based on attack data. This approach has the potential to rationalize the modeling process and enable not only security engineers but also automotive engineers to contribute to the creation of security models. By automating the generation of parts of the models that currently require manual modeling, such as specific attack scenarios and vulnerabilities, significant time savings can be achieved.

Other extensions to the metamodel could relate to the implementation of specific mechanisms, such as cryptography. Although the metamodel already allows the modeling of requirements and security concepts, these additions could allow a more detailed and accurate modeling of the internal relationships of these mechanisms.

The Cybersecurity Assurance Levels (CALs) from the ISO 21434 standard can be specified in the tool for a security goal. However, these security goals and other entities from the concept phase, such as requirements, are not currently included in the reporting, as the current reports focus primarily on risk assessment. For CALs, it is important to note that no consensus has yet been reached on how to determine and treat such a parameter, so this aspect has been relegated to the Annex only [20]. This could be a potential future extension, allowing for the creation of reports that encompass requirements, security goals, and concepts, even though this is not explicitly required by the standard.

References

[1] "ISO/SAE 21434:2021, Road vehicles – Cybersecurity engineering," Aug. 2021, https://www.iso.org/standard/70918.html
[2] J. Li, Y. Dong, S. Fang, H. Zhang, and D. Xu. 2020. "User Context Detection for Relay Attack Resistance in Passive Keyless Entry and Start System," Sensors, vol. 20, no. 16, p. 4446, Aug. 2020, doi: https://doi.org/10.3390/s20164446.
[3] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès. 2016. "Lock it and still lose it – on the (In)Security of automotive remote keyless entry systems," in 25th USENIX Security Symposium (USENIX Security 16), ser. SEC'16. Austin, TX, USA, Aug. 2016, pp. 929-944.
[4] L. Wouters, E. Marin, T. Ashur, B. Gierlichs, and B. Preneel. 2019. "Fast, furious and insecure: Passive keyless entry and start systems in modern supercars," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2019, no. 3, pp. 66-85, https://doi.org/10.13154/tches.v2019.i3.66-85
[5] Costantino, A. La Marra, F. Martinelli, and I. Matteucci. 2018. "Candy: A social engineering attack to leak information from infotainment system," in 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), pp. 1-5, https://doi.org/10.1109/VTCSpring.2018.8417879
[6] H. Gustavsson, E. P. Enoiu and J. Carlson. 2022. "Model-Based System Engineering Adoption in the Vehicular Systems Domain," 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, pp. 907-911, https://doi.org/10.15439/2022F47.

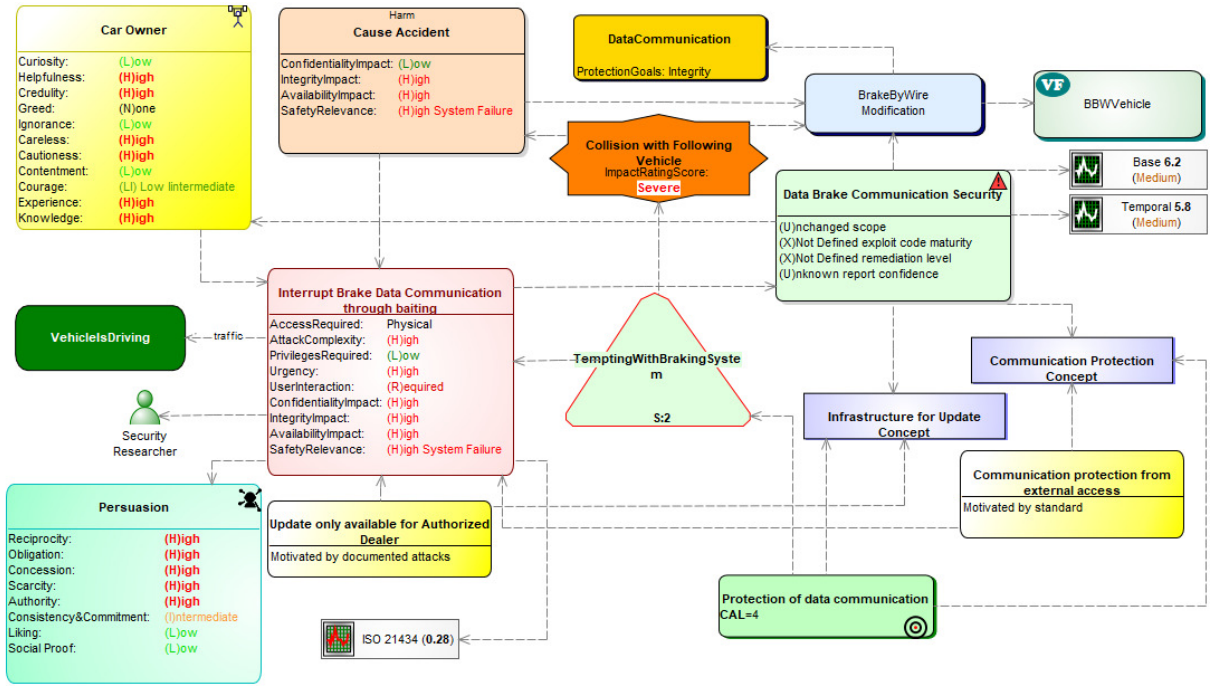

Fig. 10. Security model in modeling editor illustrating a social engineering attack

[7] M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda and D. Ratiu. 2010. "Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments," in Proceedings of the IEEE, vol. 98, no. 4, pp. 526-545, https://doi.org/10.1109/JPROC.2009.2037771.

[8] OMG. Unified modeling language specification version 2.5.1. 2017. https://www.omg.org/spec/UML/2.5.1/

[9] OMG. Systems Modeling Language Specification version 1.6. 2019. https://www.omg.org/spec/SysML/1.6/

[10] P. H. Feiler, D. P. Gluch and J. Hudak. 2006. "The Architecture Analysis & Design Language (AADL): An Introduction," https://doi.org/10.1184/r1/6584909.v1

[11] AUTOSAR: Enabling Continuous Innovations. 2024. https://www.autosar.org/

[12] H. Blom, H. Lönn, F. Hagl, Y. Papadopoulos, M.-O. Reiser, C.-J. Sjöstedt, D.-J. Chen, F. Tagliabò, S. Torchiaro, S. Tucci et al. 2013. "EAST-ADL: An architecture description language for automotive software-intensive systems," in Embedded Computing Systems: Applications, Optimization, and Advanced Design. IGI Global, pp. 456-470.

[13] M. Bergler, J.-P. Tolvanen, M. Zoppelt, and R. Tavakoli Kolagari. 2021. "Social Engineering Exploits in Automotive Software Security: Modeling Human targeted Attacks with SAM," 31st European Safety and Reliability Conference, ESREL 2021, Sep. 2021, pp. 2502-2509, https://dx.doi.org/10.3850/978-981-18-2016-8_720-cd

[14] MetaCase. 2023. MetaEdit+ 5.5 User's Guides, https://metacase.com/support/55/manuals/ (accessed May 2024)

[15] J.-P. Tolvanen and S. Kelly. 2023. "Effort used to create domain-specific modeling languages," 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, Oct. 2023, https:///doi.org/10.1145/3239372.3239410.

[16] C. Jakobs, M. Werner, K. Schmidt and G. Hansch. 2022. "Heuristic Risk Treatment for ISO/SAE 21434 Development Projects," 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, pp. 653-662, https:///doi.org/10.15439/2022F136.

[17] M. Zoppelt and R. Tavakoli Kolagari. 2019. "SAM: A security abstraction model for automotive software systems," in Security and Safety Interplay of Intelligent Software Systems, B. Hamid, B. Gallina, A. Shabtai, Y. Elovici, and J. Garcia-Alfaro, Eds. Cham: Springer International Publishing, pp. 59-74, https://doi.org/10.1007/978-3-030-16874-2

[18] M. Bergler and R. Tavakoli-Kolagari. 2023. "Automotive Software Security Engineering based on the ISO 21434", in Proceedings of the 2023 5th World Symposium on Software Engineering. Association for Computing Machinery, New York, NY, USA, 17–26, https://doi.org/10.1145/3631991.3631994

[19] J.-P. Tolvanen and S. Kelly. 2023. "Evaluating Tool Support for Co-Evolution of Modeling Languages, Tools and Models", 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), https://doi.org/10.1109/models-c59198.2023.00144

[20] Macher, C. Schmittner, O. Veledar, and E. Brenner. 2020. "ISO/SAE DIS 21434 Automotive Cybersecurity Standard - In a Nutshell," Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops, pp. 123–135, https://doi.org/10.1007/978-3-030-55583-2