Hindawi
Security and Communication Networks
Volume 2020, Article ID 5137625, 12 pages
https://doi.org/10.1155/2020/5137625

Research Article

Integrating Security Requirements Engineering into MBSE: Profile and Guidelines

D. Mažeika and R. Butleris

Centre of Information Systems Design Technology, Kaunas University of Technology, Kaunas, Lithuania

Correspondence should be addressed to D. Mažeika; donatas.mazeika@ktu.lt

Received 25 September 2019; Accepted 10 December 2019; Published 12 March 2020

Academic Editor: Salvatore Sorce

Model-Based System Engineering (MBSE) provides a number of ways on how to create, validate, and verify the complex system design; unfortunately, the inherent security aspects are addressed neither by the SysML language that is the main MBSE enabler nor by popular MBSE methods. Although there are many common points between MBSE and security requirements engineering, the key advantages of MBSE (such as managed complexity, reduced risk and cost, and improved communication across a multidisciplinary team) have not been exploited enough. This paper reviews security requirements engineering processes and modeling methods and standards and provides the MBSE security profile as well, which is formalized with the UML 2.5 profiling capability. The new UML-based security profile conforms to the ISO/IEC 27001 information security standard. In addition to the MBSE security profile, this paper also presents the security profile application use case and the feasibility study of current status for security and systems engineering processes.

1. Introduction

Modern systems among industries such as automotive, medical devices, aerospace, and defence are becoming extremely complex; therefore, traditional engineering methods are not sufficient for their successful realization. The systems have become more complex, due to many factors, to name a few:

(i) Increased spectrum of technologies: complex systems have become cyber-physical systems (CPS) and now depend upon the seamless integration of computational algorithms and various physical components [1]

(ii) Increased customer demands for more sophisticated systems and market or military competition [2]

(iii) Systems consist of a large number of components interacting in a network structure and usually these components are physically and functionally heterogeneous [3]

The discipline of systems engineering (SE) was initiated and developed to manage and unite work results of multidisciplinary engineering teams. The goal of SE is successful realization of systems with the focus on gathering customer needs and defining required functionality early in the development cycle, as well as documenting requirements, then proceeding with design synthesis and system validation [4]. Nowadays, organizations that cannot cope with systems complexity have switched (or are switching) from a document-based approach to a model-based approach in the SE activities. International Council on Systems Engineering (INCOSE) emphasizes MBSE importance, and they envision that MBSE will become a synonym of SE by 2025 [5]. The advantages of using models instead of documents in SE include the following [6–8]:

(i) Increased systems engineering efficiency by

  (a) Reusing existing projects or common components to support design and technology evolution

  (b) Enabling impact analysis of requirements changes

(c) Improving communication across a multidisciplinary team
(d) Enabling autogeneration of documentation

(ii) Reduced risk by early and iterative requirements validation and design verification

(iii) Managed complexity

There are a number of methods that guide users on how to get all of the MBSE benefits when creating a system design model. The detailed review of these MBSE methodologies and frameworks is available in the previous papers [9, 10]; sadly, neither of the analyzed methods deals with the security analysis at an early stage of system design.

Many researchers in their studies [11–14] agree that there is a need to identify and tackle security risks during the systems engineering lifecycle. Nguyen et al. state that security objectives (such as confidentiality, integrity, and availability) should be considered together with the business logic very early, which is crucial in engineering secure systems. Here, MBSE could be a key helper because of the opportunity to manipulate models on higher abstraction level; possibility to tailor generic modeling language (e.g., UML and SysML) with the security-related concepts; and performing reasoning with external analysis tools [12]. Nowadays, the MBSE activity mostly focuses on the design phase which is usually done by the systems engineers. When developing complex systems, the security analysis is conducted in parallel with the design phase. Papke argues that security engineers and systems engineers should work together and a joint design process or framework is needed in order to define security aspects in a common model [13].

In this paper, we present the MBSE security profile that would enable a multidisciplinary team to perform security analysis in parallel to the systems engineering process in one MBSE project. We also introduce a small case study that presents the potential value of using a model-based approach for security analysis.

2. Feasibility Study

Before starting the analysis and security profile development tasks, we conducted a feasibility study to support or deny our initial thesis that the MBSE would be helpful and needed during security analysis. We sent a questionnaire to 10 engineering companies from the following industries: *transportation*, *aerospace and defence*, *maritime*, *healthcare*, and *software*. The survey questions were answered by systems engineers (total: 8), a chief systems engineer, and a security engineer.

The first two questions were dedicated to finding out how many organization members are involved in systems engineering and how many are in security engineering activities. The results are provided in Figure 1.

As shown in Figure 1, the numbers of organization members that are involved in systems engineering activities are much higher than those in security engineering. Since we suggest to include security activities into the MBSE model, the effort of training security engineers the MBSE would be significantly lower than vice versa.

Figure 1: Chart presenting the number of members for systems engineering and security engineering in the surveyed organizations.

The third question was dedicated to finding out the distribution of system engineering activities. The majority of respondents perform system requirements definition and functional design activities; in addition, logical and physical design activities are also widely used. All these activities, except physical design creation, are usually conducted at an early stage of system development. All the results are provided in Figure 2.

The fourth question was "Are the security requirements or other security artefacts represented (or linked) in your systems engineering models/documents?" In total, 6 respondents said that it was linked, 2 said that it was partially linked, and 2 respondents said that the artefacts were not linked. Also, the respondents were asked to elaborate more on this question; here are the opinions:

(i) No security artefacts produced. Security is approached as additional requirements on the system.

(ii) We currently only collaborate internally in our company.

(iii) Some system attributes that are relevant for security are modeled. Some model elements are also specifically created for security analysis purposes (networks, for example).

(iv) Mostly by linked security requirements.

(v) Documentation of assets/system objects and physical and logical connection.

These answers lead to the conclusion that more than half the respondents trace security requirements with the systems engineering elements but not in a consistent way.

The fifth question was, "Does your organization conform to any security standard for system design?" The 43 percent of respondents said that their process conforms to the ISO/IEC 27001 standard; all the answers are provided in Figure 3.

Next, we wanted to find out what techniques organizations practice for security analysis. The majority of respondents (8) rely on security requirements. The *attack/threat scenarios* and *security processes/controls definition* were practiced by 3 respondents. All the results are provided in Figure 4.

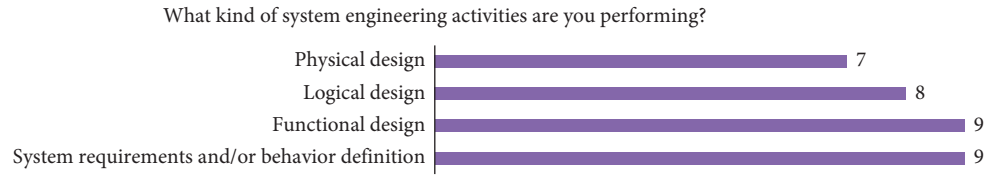

Figure 2: Chart presenting distribution of systems engineering activities.

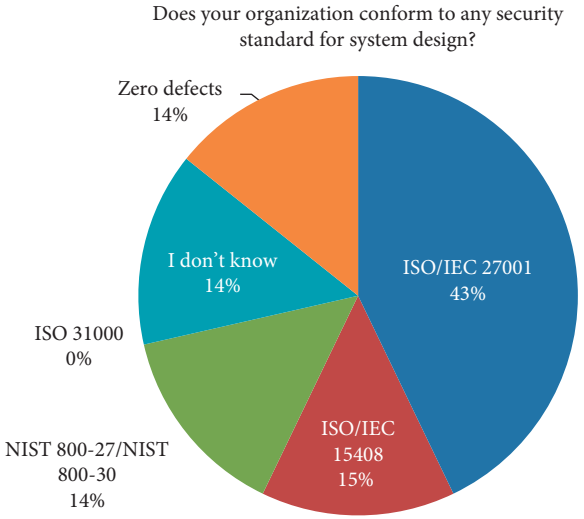

Figure 3: Chart of the question, "Does your organization conform to any security standard for system design?".

Figure 4: Chart of the question, "What techniques does your organization practice for performing security analysis?".

The next question helped us find out whether the security analysis integration into MBSE could bring any benefits. The majority of participants agree or strongly agree that all the listed advantages would be important. All the results are provided in Figure 5.

The last question was dedicated to checking which techniques would be useful for validating/verifying a security model (Figure 6).

Seven of ten respondents answered that the most useful techniques would be model validation (e.g., checking if the current level of risk is acceptable) and change impact analysis (e.g., check what assets will be impacted if the security requirement is changed). Five respondents said that the coverage analysis (e.g., check how many risks are not linked with security controls) and model simulation (e.g., check if attack scenario is executed correctly) would be useful as well.

To summarize, the feasibility survey showed that both systems engineers and security engineers acknowledge the importance and value of integrating systems and security models; however, this has not yet been implemented in practice.

3. Analysis of Related Work

The analysis of the related work includes the Security Requirements Engineering section in which we present the security requirements engineering process definitions. The next chapter, Modeling Approaches for Security Analysis, demonstrates how the traditional security requirements engineering process is incorporated into the different systems modeling methods.

3.1. Security Requirements Engineering. Security Requirements Engineering domain combines methods, techniques, and norms for tackling secure systems creation task during

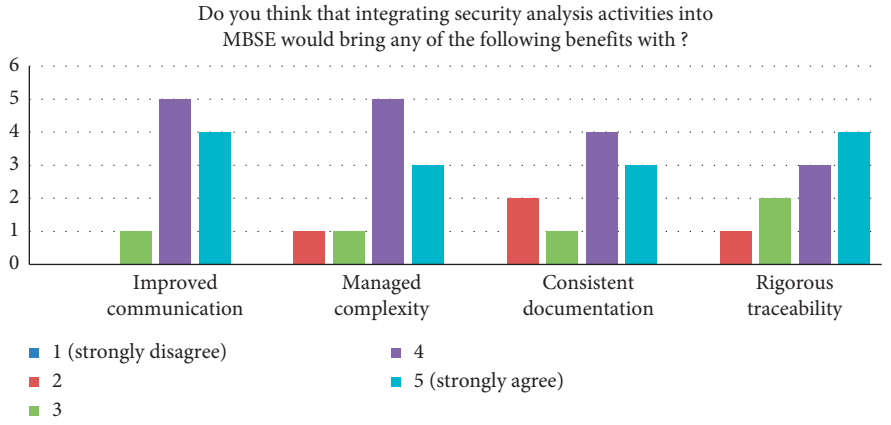

Figure 5: Chart representing benefits of integrating security activities into MBSE model.

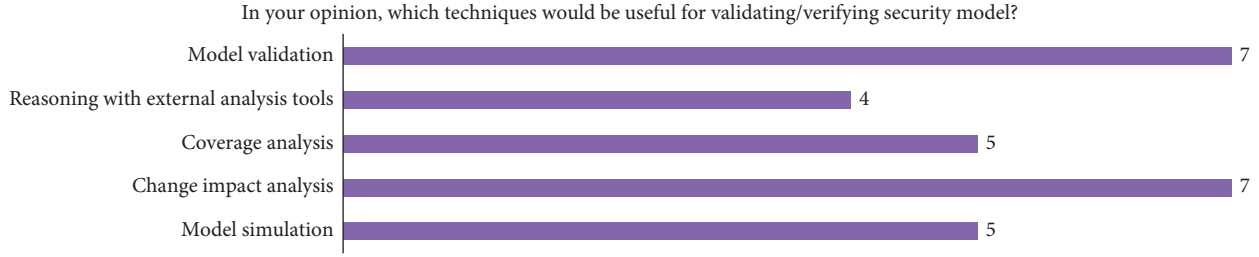

Figure 6: Chart representing MBSE techniques for validating/verifying security model.

the early stages of the system development cycle [15]. Many approaches for performing security requirements engineering have been proposed in the literature. Some of the methods provide guidelines for security-related activities (e.g., SQUARE [16] and CLASP [17]), while some of them operationalize security standards (e.g., SREP is based on ISO/IEC 17799:2005 [18] and CORAS is based on ISO 31000 [19]). The detailed comparison of security requirements engineering methods was provided by Fabian et al. [20], and the comprehensive ontology for the IS risk management domain was defined by Dubois et al. [21].

The security requirements engineering process includes traditional requirement engineering activities such as requirements elicitation, specification, and analysis. The final purpose of security requirements engineering is to prevent harm in the real world by considering security requirements as constraints upon functional requirements [22]. Here, the most recurring word is *security requirement* and it is worthwhile to look at how this term is treated by different authors:

(i) Dubois et al. characterize security requirement as a condition over the phenomena of the environment that system stakeholders wish to make true by designing the system, in order to mitigate risks [21].

(ii) Fabian accents that the security requirement is the detailed refinement of one or more security goals, whereas the security goal refers to a particular part of the CIA (*confidentiality*, *integrity*, and *availability*) model [20].

(iii) Salini and Kanmani agree that security requirements can be treated as a constraint on the functions of the system, and these constraints operationalize one or more security goals [22].

Respectively, security requirements can be considered as a more detailed statement of security goal or objective. In our research, we look further at how this term is interpreted and refined in various modeling approaches.

One remark about security and safety requirements engineering should be noted. Despite the fact that security and safety disciplines have many similarities (e.g., both are protecting assets by creating secure/safe conditions [23]), core differences exist too [24]:

(i) The origin of risk: security focuses on *threats* (e.g., attacker hacks aircraft in-flight entertainment system and overrides the security software), while safety considers *hazards* (e.g., landing gear of the aircraft fails to extend).

(ii) The nature of consequences: unmanaged security risks could cause harm to the system itself or to its environment. The consequences of safety risks are related to the system environment only.

In this research, we do not analyze safety techniques except those that combine both safety and security areas (e.g., CHASSIS).

3.2. Modeling Approaches for Security Analysis. This section provides an analysis of the following modeling frameworks and methods for identifying security risks:

(i) Unified Architecture Framework (*UAF*)

(ii) The combined harm assessment of safety and security for information systems (CHASSIS)

(iii) SysML Sec

(iv) UML Sec

Here, we picked out graphical modeling approaches that could be used at an early stage of system design and integrated into the MBSE process. We excluded formal security methods based on mathematical techniques or semiformal approaches that are based on a different graphical form than UML/SysML (e.g., Petri nets and Bayesian belief network) because the different notation may include additional complexity to the MBSE model and formal methods are usually implemented in a later phase. Also, techniques used in other methods (e.g., misuse cases in CHASSIS) are not separately detailed in this paper.

UAF is an enterprise architecture framework (EAF) created by Object Management Group (OMG) [25]. The *UAF* framework unifies existing military architecture frameworks (such as MoDAF, DoDAF, and NAF) and, unlike the latter, it is applicable to industrial and commercial applications as well [26, 27]. Besides the demilitarization and unification of military frameworks, *UAF* has an additional security domain [28]. The security domain enables users to identify the security constraints and capture information assurance properties that exist during communication between resources and operational performers [25]. These information-assurance properties are aligned to NIST/DOD standards that are the base for the unified information security framework for the entire US federal government [29, 30].

The key security concepts used in *UAF* are *security constraint*, *security property*, *security assets*, *security controls*, *risk*, and *security impact property* [25].

CHASSIS is a mnemonic acronym for the *combined harm assessment of safety* and *security for information systems*. The CHASSIS method allows identifying both security and safety aspects and is based on UML notation [31]. This method comprises three main processes (eliciting functional requirements; eliciting safety/security requirements; and specifying safety/security requirements) and different security management techniques for eliciting and specifying security requirements. The definition of security requirements relies on creating and analyzing UML-based diagrams (misuse case, misuse sequence diagram) and conducting their results with traditional security techniques, e.g., HAZOP table and textual security requirements [31].

Misuse case technique extends the UML use case diagram with the additional elements of misuse case and misuser. These concepts allow defining attackers and their threats to the system of interest. Also, two supplementary relations of threatens and mitigates allow security engineers to specify which use case mitigates misuse case or which misuse case threatens use case. Misuse sequence diagram can be used to represent possible interactions between attacker and system that are arranged in time sequence. Finally, the HAZOP table is used to summarize the relevant information for the safety and security requirements [31, 32].

The key security concepts used in CHASSIS are *attack*, *attacker*, *threat*, *security requirement*, *risk*, and *weakness*.

SysML Sec is a model-driven engineering environment, which presents extended SysML diagrams for security risks as well as the methodology for creating secure real-time embedded systems [33].

The SysML Sec methodology consists of three main phases [33]:

(1) System analysis (based on Y-chart approach for embedded systems)

(2) System design (based on V-model for software development)

(3) System validation (based on model transformation into formal specifications)

The analysis phase covers the definition of security requirements and attack scenarios and serves as an identification of main functions and candidate hardware architecture as well. In the system design stage, security requirements are refined with security properties and security-related functions are defined. The validation phase allows users to formally assess whether security properties are verified. If the model is too large to be verified, model-to-code transformations are used to perform security tests [33, 34].

The key security concepts used in SysML Sec are *assets*, *security requirement*, *security property*, *security-related function*, and *threat*.

The UML Sec approach enables a definition of security requirements for a system under analysis with a lightweight extension of UML. As UML Sec is a lightweight extension, it does not present any new diagrams but provides a set of stereotypes (with tag definitions) and constraints. Security-related stereotypes allow users to specify security requirements and attack/failure scenarios with standard UML diagrams (e.g., use case, activity, and sequence diagrams). The custom constraints written in OCL (Object Constraint Language) help to verify the model with formal semantics [14, 35]. The UML Sec method can be integrated with the Goal-Driven Security Requirements Engineering methodology in order to have a structured framework for secure software systems development [36].

The key security concepts used in UML Sec are *security requirement*, *security property*, *attacker*, and *attack*.

4. Concepts Alignment

This section is dedicated to aligning all the analyzed modeling approaches for security analysis. We present security concepts with definitions, synonyms, and their occurrence in the analyzed modeling approaches in Table 1 (Y indicates that the corresponding concept is used in modeling approach and N means that it is not relevant).

Table 1: Security concepts mapped to modeling approaches.

	UAF	CHASSIS	SysML Sec	UML Sec	Definition	Synonyms
Asset	Y	Y	Y	N	Elements that can be considered as a subject for security analysis [25] Something in the system and/or its environment, to be protected from negative consequences [31]	Software asset, system asset, data asset
Security constraint	Y	Y	Y	Y	A type of rule that captures a formal statement to define security laws, regulations, guidances, and policies [25]	Security requirement, security goal
Security control	Y	N	Y	N	A safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of the asset's information and to meet a set of defined security requirements [25]	Security activity, safeguard, countermeasure, security-related function
Security property	Y	N	Y	Y	Property or constraint on a system asset that characterizes their security needs [25]	Information-assurance property
Risk	Y	Y	Y	N	A statement of the impact of an event on assets [25]	—
Risk impact	Y	Y	Y	N	The potential impact on system due to a specific reasons (availability, integrity, and confidentiality) [25]	Harm, consequence, security impact property
Vulnerability	Y	Y	N	Y	An internal fault that enables an external fault to harm the system [31]	Weakness, security constraint (in *UAF*)
Attacker	N	Y	Y	Y	Someone or something carrying out an attack for altering the system's functionality or performance, or accessing confidential information [31]	Intruder
Threat	Y	Y	Y	Y	Potential attack that targets system assets and that may lead to harm to assets [21] An action carried out to harm system [31]	Attack, security constraint (in *UAF*)

5. Security Domain Model

Once we are finished with literature analysis and concept alignment activities, we can transform analyzed data into the security domain model. The domain model is specified with the MagicDraw modeling tool in the UML class diagram, and it describes security concepts and their relations (see Figure 7).

Three groups in the security domain model were distinguished:

(i) Security assurance concepts (white) describe concepts that allow ensuring system security or mitigating possible risks

(ii) Items to be protected (green) present data and system assets that should be identified and protected

(iii) Risk-related concepts (red) characterize hostile concepts and possible system weaknesses

The security domain model allows classifying various risk terms and establishing logical relationships between them. However, the security domain model is not enough for the model-based security analysis; thereupon, the next chapters present the security profile that would enable such analysis.

6. Security Profile Structure and Content

We use the built-in profiling capability of UML 2.5 that enables us to transform security concepts that were specified in the domain model to the security modeling language. This is a classical modeling language design approach where the key concepts of the domain should be determined at first, and then a new language could be created to support it [37].

The ISO/IEC 27001 information security standard by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) is selected as the basis for the security profile. This standard is designed to help engineering secure systems by providing the best practice recommendations on information security management, risks, and controls [38]. ISO/IEC 271001 enables us to break down the security profile into logical sections, as well as to use industry recognizable terms.

The profile structure is shown in Figure 8. The profile scheme contains the groups (separated with dashed line) where each follows needed steps for establishing an Information Security Management System (ISMS) by ISO/IEC 27001.

The first step suggests that the criteria for accepting risks should be defined before identifying and evaluating the risks. To support this step, we created a stereotype of "Risk Assessment Configuration" with the tag definition of "Criteria for Accepting Risks" that will allow users to specify which risk level will be acceptable in their organization.

The "4.2.1 d" chapter in the ISO/IEC 27001 standard talks about the identification of the following elements:

(1) The *assets* within the scope of the ISMS and the *owners* of these assets

(2) The *threats* to those assets

(3) The *vulnerabilities* that might be exploited by the threats

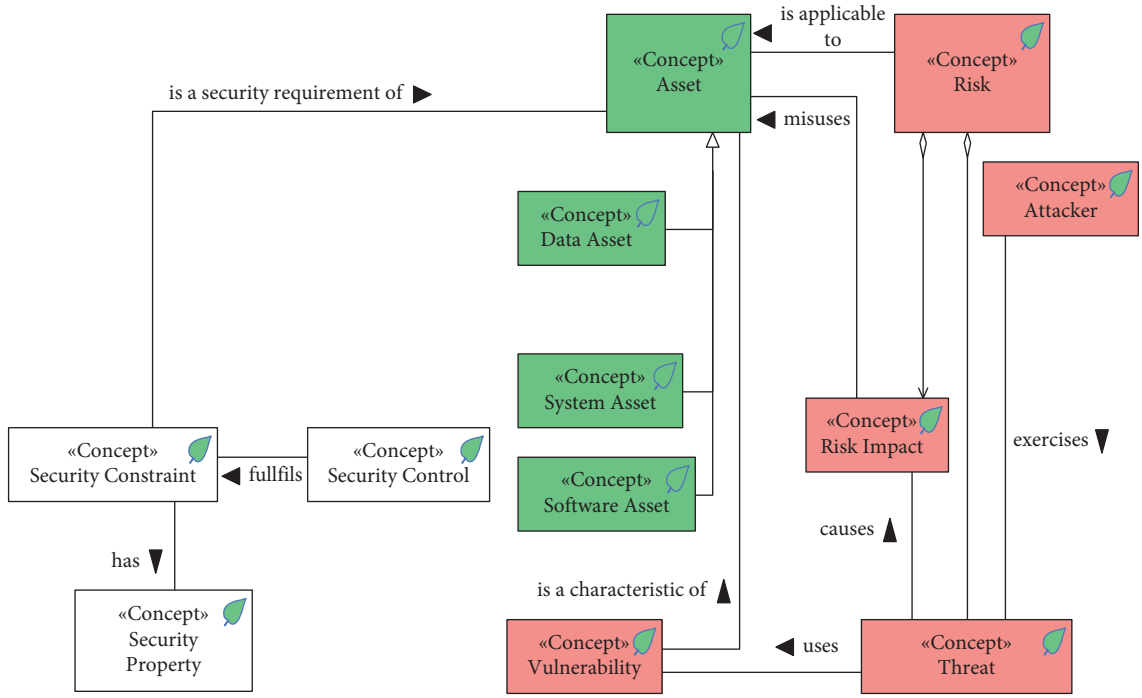

Figure 7: The security domain model.

(4) The *impacts* that losses of confidentiality, integrity, and availability may have on the assets

Respectively, we have created the following stereotypes: *Risk; Asset (Data Asset/System Asset/Software Asset); Vulnerability; Threat; Risk Impact*. Also, the additional stereotypes were created for all possible relations (*Characterize*, *Cause*, *Misuse*, *Use*, and *Applicable To*).

The third step indicates that the business impact and the acceptable risk level scale should be identified at this stage. The "Risk Impact" stereotype will allow users to document risk impact and the "Risk Level" enumeration will provide the scale from 1 to 10.

The fourth step refers to the options for identifying and evaluating the treatment of risks. The *Security Control* concept was identified during domain analysis; however, ISO/IEC 27001 extends this term with the *Risk Treatment* that should have two options: Risk Control and Opportunity to transfer risk to an external party. Accordingly, the following stereotypes are created and added to this group: *Risk Treatment*, *External Party*, *Transfer to External Party,* and *Apply Control*.

The fifth section is dedicated to selecting control objectives and controls for the treatment of risks. The "Control Objective" and "Security Control" stereotypes will help in capturing such information.

The next chapter presents how the security profile can be applied in the real-world SysML model.

7. Security Profile Application Use Case

In order to demonstrate the security profile usage of performing security analysis, we selected Hybrid Sport Utility Vehicle (HSUV) model from the OMG SysML specification [39]. Originally, this model was created to illustrate how the SysML language can support the specification, analysis, and design of a system. We are refining this model by adding the "HSUV Security Analysis" layer which covers the system risk assessment configuration and security artefacts for the power control unit.

Before starting the security analysis for HSUV, the security engineer/manager must develop criteria for accepting risks and identifying the acceptable level of risks. There are many different methodologies for risk assessment that shall ensure comparable and reproducible results. The criteria and methodology should be captured in the *Risk Assessment Configuration* element in an MBSE tool.

For the security analysis, the multidisciplinary (systems and security engineering) team should analyze all the parts of HSUV to find out whether those parts can be violated/attacked, what is the risk and risk impact, and what security prevention controls are possible. The quantitative analysis is enabled by an MBSE tool and it allows calculating such data:

(i) How many system parts are not treated as assets?

(ii) How many risks do not have risk treatment?

(iii) How many risks have a higher level than an acceptable level of risk defined in Risk Assessment Configuration?

For this application use case, we have selected Power Control Unit that is responsible for handling vehicle acceleration and braking pedal. The extract of the Power Control Unit security analysis is presented in Figure 9. The Power Controller assets (system and software) are created and linked with the SysML block of PowerControlUnit.

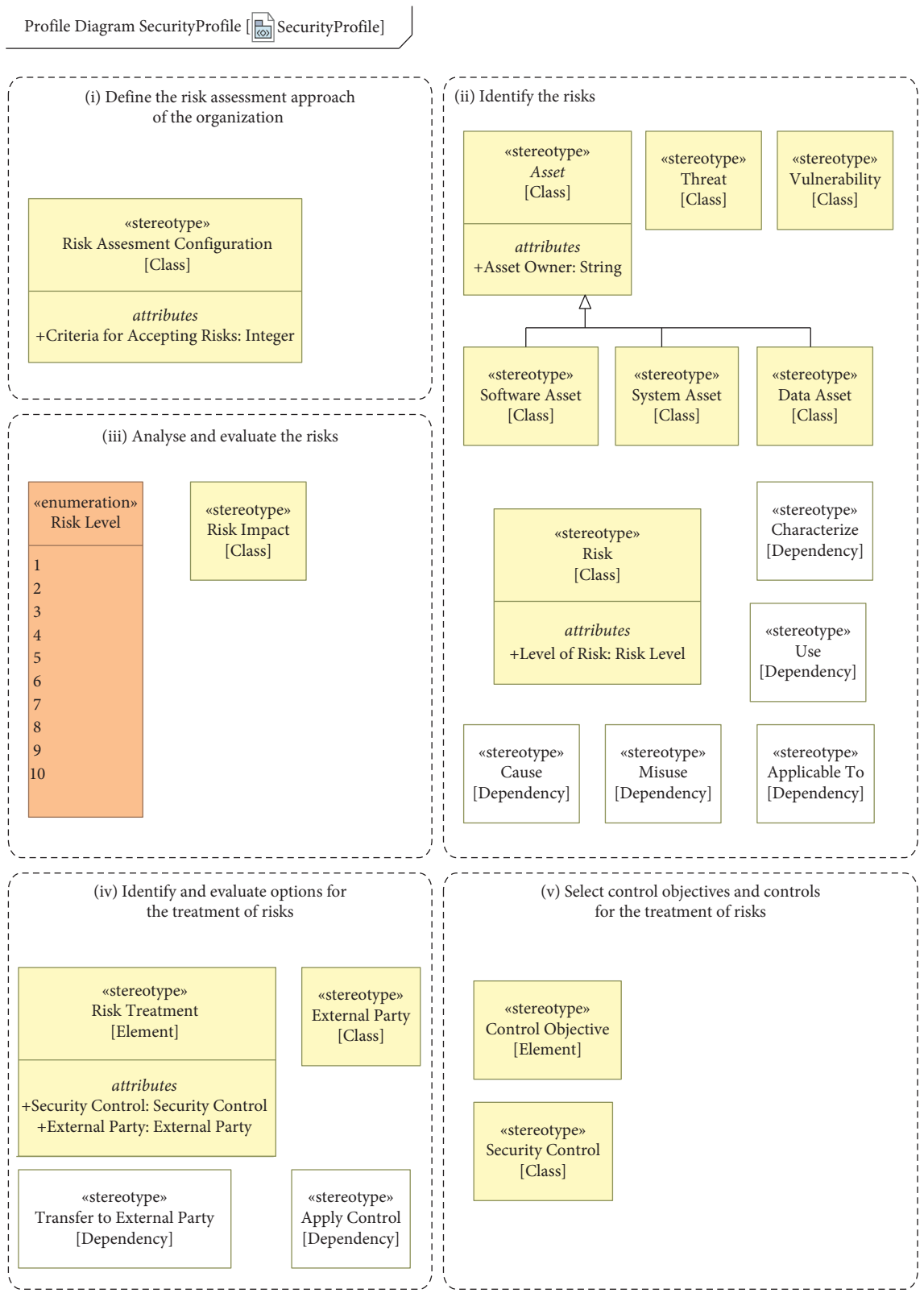

Figure 8: Security profile.

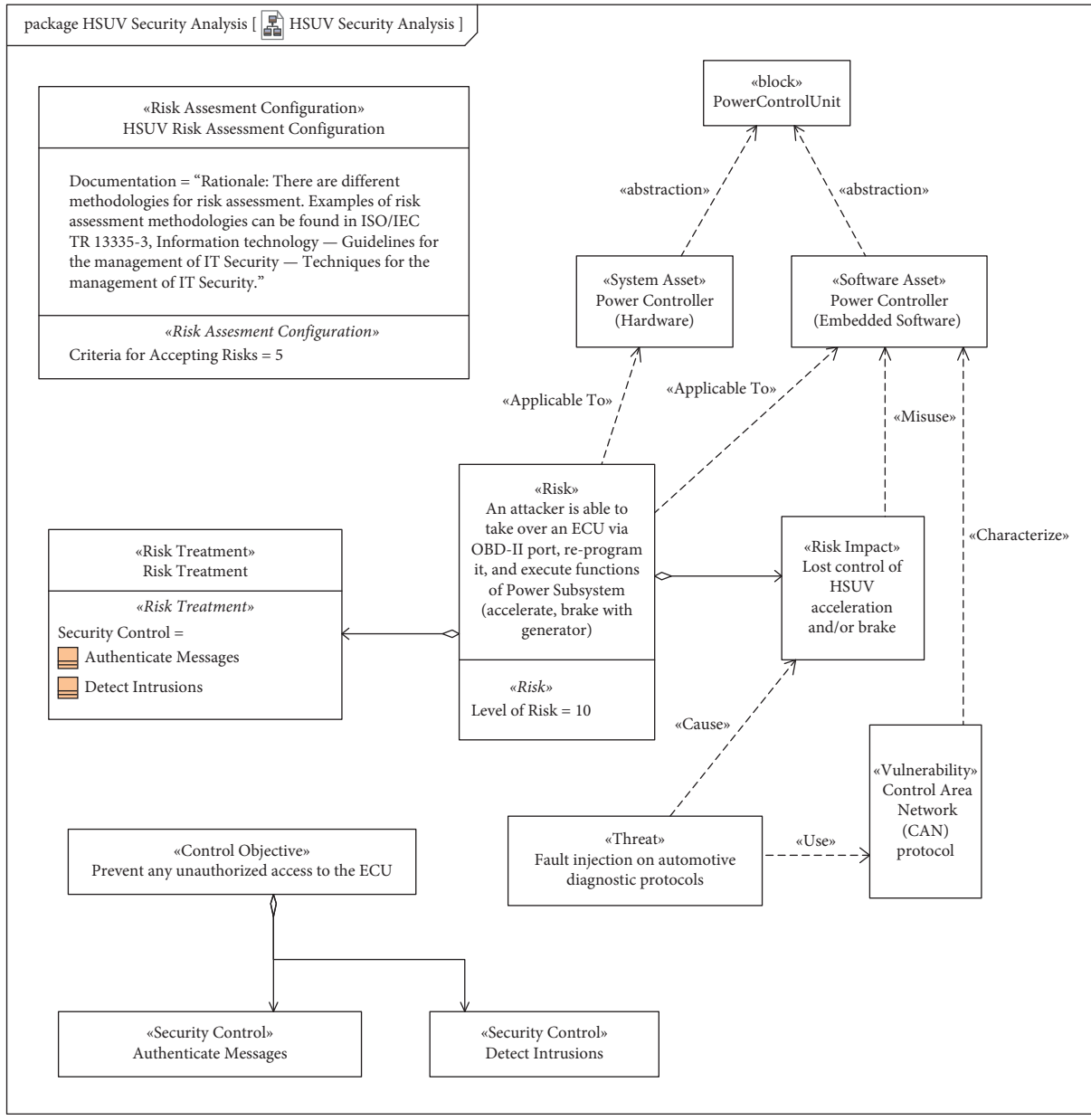

Figure 9: PowerControlUnit security analysis.

```
SysML::Block::allInstances()->select(b|
        not b.supplierDependency->exists(dep|
dep.oclIsKindOf(UML2_Metamodel::Abstraction) and
dep.oclAsType(UML2_Metamodel::Abstraction).client->forAll(c|c.oclIsKindOf(SecurityProfile::Asset))
        )
)
```

Figure 10: OCL query for finding all blocks that are not linked with an asset.

Then, the risk of "An attacker is able to take over an ECU via the OBD-II port, reprogram it, and execute functions of Power Subsystem (accelerate, brake with generator)" is identified and traced to those assets. The level of this risk is set to 10.

In our case study, the risk has the risk impact of "Lost control of HSUV acceleration and/or brake" that can be fatal. The possible threat is "Fault injection on automotive diagnostic protocols" that potentially uses the vulnerability of "Control Area Network (CAN) protocol." On the other side, the risk has the options for the risk treatment with possible control options: *Authenticate Messages* and *Detect Instructions*. If the security controls are already known, users should add the documentation for such control. If the

Criteria

Metric Suite: Blocks coverage by Assets ... Scope (optional): HSUVModel ... Filter:

#	Date	Blocks	Coverage by Assets	Coverage by Assets (percentage)
1	2019.02.03 11.43	35	1	2,86%

Figure 11: Metric table that presents how many blocks are covered by assets.

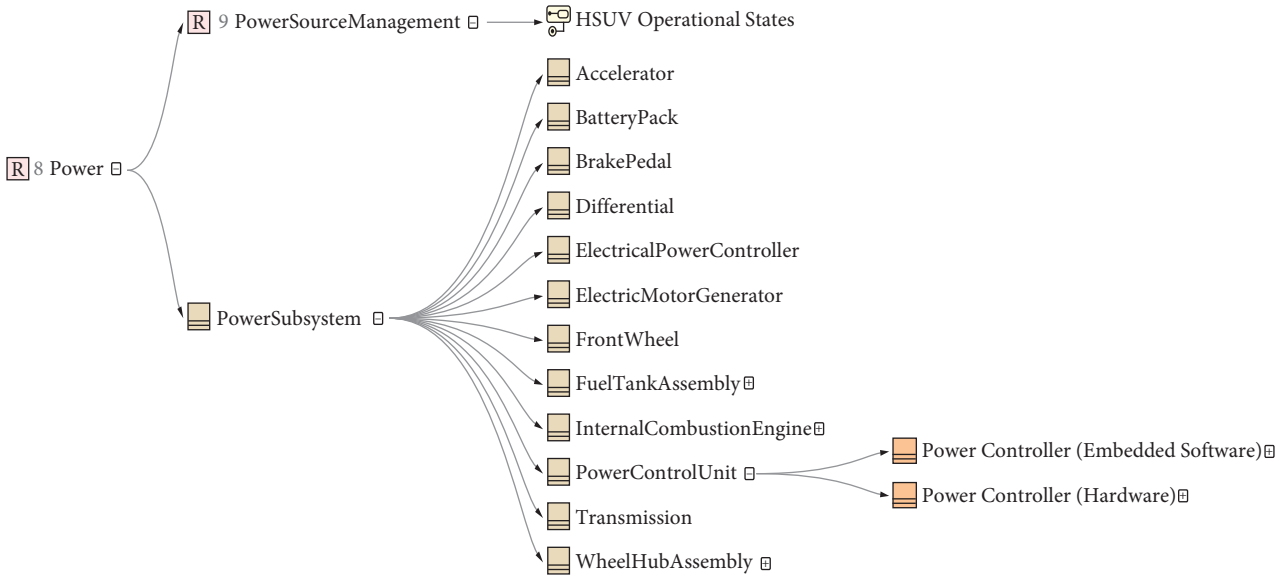

Figure 12: The change impact map for the "Power" requirement.

security control is not known, then the SysML activity diagram could be created under this element and the supposed algorithm should be modeled by the security engineer. The objective of the identified security controls is *Prevent any unauthorized access to the ECU*.

After documenting and linking all the security-related elements, we can create expressions based on Object Constraint Language (OCL) for running quantitative model verification, e.g., finding all system blocks that are not linked with any asset element (Figure 10).

This OCL expression can be used as a base for a metric table (see Figure 11), or it can be a query for collecting corresponding elements, or it can be used to validate the MBSE model in real time.

Furthermore, when both MBSE and security models are integrated, we can perform change impact analysis, e.g., check which system and security elements shall be reviewed, if the initial system requirement is being changed. In Figure 12, we demonstrate the change impact map that shows traceability from the "Power" requirement to the system and software assets.

8. Conclusions and Future Works

There are many common points between MBSE and security requirements engineering; however, these disciplines still have not been connected in terms of the standard method, approach, or framework. This leads to the fact that powerful advantages of MBSE (such as automated document generation, managed complexity, reduced risk and cost, and improved communication across a multidisciplinary team) are still being underexploited in the workflow of security engineers and systems engineers. The literature analysis and feasibility survey showed that systems engineers and security engineers recognize the value of integrating systems and security processes, but this has not been implemented in practice yet.

This paper contributes to linking MBSE discipline with the security analysis approaches in two aspects:

(1) It maps the concepts from the security requirement engineering field and UML/SysML-based modeling approaches for security analysis. The mapping and the security domain model could help users to understand and compare security terms.

(2) It introduces the UML security profile based on the ISO/IEC 27001 information security standard that allows describing and analyzing security aspect together with the system model. The use of model-based techniques ensures that the security and system artefacts are aligned at the early phase of system design and MBSE benefits are extended to security engineer discipline.

The security profile viability was presented by extending Hybrid Sport Utility Vehicle (HSUV) sample from the OMG SysML specification with the power control unit security analysis.

We are planning to continue our work and provide the extended guidelines for the MBSE security profile and check which security techniques are the most effective according to systems engineering and security practitioners.

Data Availability

The data used to support this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

[1] NSF-National Science Foundation, "Cyber-physical systems (CPS)," 2017, https://www.nsf.gov/pubs/2017/nsf17529/nsf17529.pdf.

[2] INCOSE, "The challenge of complex systems," http://www.incose-coa.org/the-challenge-of-complex-syst.

[3] J. Guckenheimer and J. Ottino, *Foundations for Complex Systems Research in the Physical Sciences and Engineering*, NSF Workshop, 2008.

[4] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processing and Activities*, John Wiley & Sons, Hoboken, NJ, USA, 4th edition, 2015.

[5] INCOSE, "SE vision 2025," 2014, https://www.incose.org/AboutSE/sevision.

[6] R. S. Kalawsky, J. O'Brien, S. Chong et al., "Bridging the gaps in a model-based system engineering workflow by encompassing hardware-in-the-loop simulation," *IEEE Systems Journal*, vol. 7, no. 4, pp. 593–605, 2013.

[7] J. Holt, S. Perry, M. Brownsword, D. Cancila, S. Hallerstede, and F. Hansen, "Model-based requirements engineering for system of systems," *Proceedings of the 2012 7th International Conference on System of Systems Engineering (SoSE)*, pp. 561–566, Genova, Italy, July 2012.

[8] INCOSE UK, *What Is Model Based Systems Engineering (V2)*, J. Towers, M. Brownsword, I. Clark, C. Lewis, S. Perry, and A. Stevenson, Eds., http://www.incoseonline.org.uk/Program_Files/Publications/zGuides_9.aspx?CatID=Publications, 2015.

[9] D. Mazeika, A. Morkevicius, and A. Aleksandraviciene, "MBSE driven approach for defining problem domain," *Proceedings of the 11th System of Systems Engineering Conference (SoSE)*, pp. 1–6, Kongsberg, Norway, 2016.

[10] A. Morkevicius, L. Bisikirskiene, and N. Jankevicius, "We choose MBSE: what's next?" *Proceedings of the Sixth International Conference on Complex Systems Design & Management, CSD&M 2015*, p. 313, Paris, France, May 2015.

[11] Research Top Challenges for MBSE in Industry 4.0 and IoT–Workshop Report, October 2015.

[12] P. H. Nguyen, S. Ali, and T. Yue, "Model-based security engineering for cyber-physical systems: a systematic mapping study," *Information and Software Technology*, vol. 83, pp. 116–135, 2017.

[13] B. L. Papke, "Enabling design of agile security in the IOT with MBSE," *Proceedings of the 2017 12th System of Systems Engineering Conference (SoSE)*, pp. 1–6, Waikoloa, HI, USA, 2017.

[14] J. Jurjens and P. Shabalin, "Tools for secure systems development with UML," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 5-6, p. 527, 2007.

[15] D. Mellado, C. Blanco, L. E. Sánchez, and E. Fernández-Medina, "A systematic review of security requirements engineering," *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 153–165, 2010.

[16] US-CERT, "SQUARE process," 2013, https://www.us-cert.gov/bsi/articles/best-practices/requirements-engineering/square-process.

[17] J. Viega, "Building security requirements with CLASP," SESS@ICSE, 2005.

[18] D. Mellado, E. Fernandez-Medina, and M. Piattini, "Applying a security requirements engineering process," *Proceedings of the European Symposium on Research in Computer Security (ESORICS'06)*, Guildford, UK, September 2006.

[19] CORAS, "The CORAS method," 2015, http://coras.sourceforge.net/.

[20] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt, "A comparison of security requirements engineering methods," *Requirements Engineering*, vol. 15, no. 1, pp. 7–40, 2010.

[21] É. Dubois, P. Heymans, N. Mayer, and R. Matulevičius, "A systematic approach to define the domain of information system security risk management," in *Intentional Perspectives on Information Systems Engineering*, S. Nurcan, C. Salinesi, C. Souveyet, and J. Ralyté, Eds., Springer, Berlin, Heidelberg, Germany, 2010.

[22] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1785–1797, 2012.

[23] E. Albrechtsen, *Safety vs Security*, Norwegian University of Science and Technology, Trondheim, Norway, 2003, http://www.iot.ntnu.no/users/albrecht/rapporter/notat%20safety%20v%20security.pdf, 1st edition.

[24] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety*, vol. 139, pp. 156–178, 2015.

[25] Object Management Group, *About the Unified Architecture Framework Specification Version 1.0*, http://www.omg.org/spec/UAF/1.0/Beta2/, 2017.

[26] A. Morkevicius, L. Bisikirskiene, and G. Bleakley, "Using a systems of systems modeling approach for developing Industrial Internet of Things applications," *Proceedings of the 12th System of Systems Engineering Conference (SoSE)*, pp. 1–6, Waikoloa, HI, USA, 2017.

[27] P. Vaughan, "Integrating UPDM with SysML and UML on a DoD acquisition program, OMG *UAF* & MBSE information day," 2015, http://www.omg.org/news/meetings/tc/va-15/special-events/uaf-pdf/5_PV_Integrating_UPDM_SysML_UML_DoD_Program.pdf.

[28] G. J. Bleakley and M. Hause, "The united architecture framework the Internet of Things and power systems," 2016, http://www.omg.org/news/meetings/tc/ca-16/special-events/iot-presentations/Hause-Bleakley.pdf.

[29] SANS Institute, "Using the department of defense architecture framework to develop security requirements," 2014, https://www.sans.org/reading-room/whitepapers/bestprac/department-defense-architecture-framework-develop-security-requirements-34500.

[30] NIST, "NIST, DOD, intelligence agencies join forces to secure U.S. cyber infrastructure," 2009, https://www.nist.gov/news-events/news/2009/06/nist-dod-intelligence-agencies-join-forces-secure-us-cyber-infrastructure.

[31] C. Raspotnig, V. Katta, P. Karpati, and A. L. Opdahl, "Enhancing CHASSIS: a method for combining safety and security," *Proceedings of the Eighth International Conference on Availability Reliability and Security (ARES)*, pp. 766–773, Regensburg, Germany, September 2013.

[32] C Raspotnig, P Karpati, and V. Katta, "A combined process for elicitation and analysis of safety and security

requirements," *Enterprise, Business-Process and Information Systems Modeling*, Springer, Berlin, Heidelberg, Germany, 2012.

[33] Y. Roudier and L. Apvrille, "SysML-Sec: a model driven approach for designing safe and secure systems," *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pp. 655–664, Angers, France, 2015.

[34] Y. Roudier and L. Apvrille, "SysML-sec: a model-driven environment for developing secure embedded systems," *Proceedings of the 8th conference on the security of network architecture and information systems (SARSSI'2013)*, pp. 16–18, Mont de Marsan, France, September 2013.

[35] J. Jürjens, "UMLsec: extending UML for secure systems development," *The Unified Modeling Language*, pp. 1–9, 2002.

[36] H. Mouratidis and J. Jurjens, "From goal-driven security requirements engineering to secure design," *International Journal of Intelligent Systems*, vol. 25, no. 8, pp. 813–840, 2010.

[37] T. W. Olle, J. Hagelstein, I. G. Macdonald et al., "Information systems methodologies," *A Framework for Understanding*, Addison-Wesley, Boston, MA, USA, 2nd edition, 1991.

[38] ISO/IEC 27001, "Information technology-security techniques-information security management systems-requirements," Technical Report, International Standards Organisation, Geneva, Switzerland, 2013.

[39] *OMG Systems Modeling Language Version 1.5*, https://www.omg.org/spec/SysML/1.5/PDF, 2018.