A Reference Model and a Dedicated Method in Support of Cyber-Security by Design: Reality Check

Sybren de Kinderen[1], Monika Kaczmarek-Heß[2] and Simon Hacks[3]

[1]*Eindhoven University of Technology, Eindhoven, The Netherlands*
[2]*University of Duisburg-Essen, Essen, Germany*
[3]*Stockholm University, Stockholm, Sweden*

Abstract

The electricity sector increasingly intertwines IT and the physical grid, increasing the risk of cyber attacks on this critical infrastructure. Hitherto, we have developed a modeling method to support cyber-security by design in the electricity sector by providing (1) a multi-level reference model, (2) a semi-automated security assessment, and (3) a dedicated process model. In this paper, we focus on four challenges identified based on interactions with domain experts, namely: (1) automated model creation; (2) accounting for changing security requirements; (3) multi-level model management; and (4) incentives for modelers. These challenges are relevant to our modeling method and overlap with challenges on the practical uptake of modeling in general.

Keywords

cyber-security by design, multi-level reference model, modeling challenges

1. Background

The electricity sector is increasingly characterized by an intertwining of IT and the physical grid infrastructure [1], expressing itself in ideas such as digital currencies for peer-to-peer electricity exchange [2]. This increased cyber-physical nature of the electricity sector also increases the pertinence of (cyber-)security, especially given that the electricity sector is often considered as critical infrastructure, i.e., being critical to societal functioning [3]. In light of this, in earlier work [4, 5] we have developed a modeling method for supporting cyber-security by design. This modeling method complements existing cyber-security methods by: (1) explicitly capitalizing on the strengths of conceptual modeling and multi-level modeling [6] in particular, and (2) offering end-to-end security by design, meaning, that our modeling method not only treats cyber-security reactively but proactively makes cyber-security a pertinent concern during the design of a cyber-physical artifact for the electricity sector.

Our research project fits squarely into design science research, and we have followed the engineering cycles as proposed by [7]. So far, we have developed a reference model, see [5], using the Flexible Meta Modeling and Execution Language (FMML^{x}), a multi-level modeling

EMISA 2023: 13th International Workshop on Enterprise Modeling and Information Systems Architectures, May 11 and 12, 2023, Stockholm, Sweden

s.d.kinderen@tue.nl (S. d. Kinderen); monika.kaczmarek-hess@uni-due.de (M. Kaczmarek-Heß); simon.hacks@dsv.su.se (S. Hacks)
0000-0002-3756-4788 (S. d. Kinderen); 0000-0002-1621-2775 (M. Kaczmarek-Heß); 0000-0003-0478-9347 (S. Hacks)

approach with an integrated modeling and programming environment called XModeler [8]; and then, see [4], we have provided a first semi-automated assessment in terms of attack path simulations based on concepts resulting out of the reference model. As a next step, we have developed a process model and accompanying guidelines to be used together with the model.

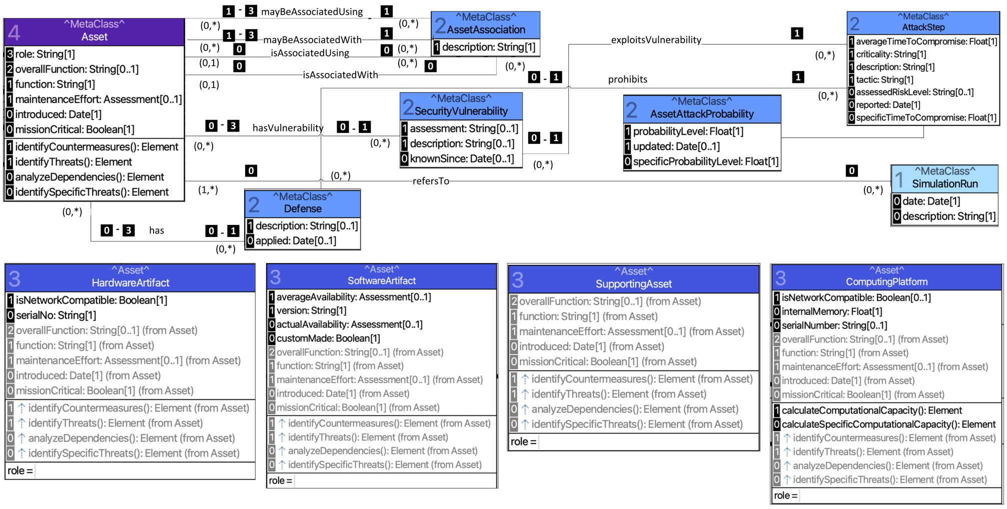
In line with the research approach, we have involved domain experts at different stages of our undertaking to ensure the problem's relevance and perform a reality check. The results of the conducted interviews and interactive workshops are stimulating, not only in terms of feedback on the designed artifact but especially in the sense that they confront our idealistic ideas of what a reference model should be able to accomplish, with down-to-earth insights from practitioners. In this light, *this paper offers a brief reality check of our modeling method, especially the gained feedback on the reference model and its (potential) usage.* The results reported here provide input for our reference model specifically but also touch upon perceived challenges in the domain of conceptual modeling as such, referring to (1) automated creation and maintenance of models, (2) accounting for changes in the body of knowledge, (3) model management and support for different views, as well as (4) incentivizing users.

This paper is structured as follows. First, to make the paper self-consistent, we summarize our main artifact. Then, after discussing the involvement of domain experts, we report on identified challenges and resulting implications. The paper concludes with final remarks.

2. A Reference Model for Cyber-Security by Design and a Supporting Method: a Primer

The targeted method encompasses (1) a reference model, (2) a corresponding process model, as well as (3) supporting software tools. Being a design science artifact, our method is naturally based on a set of design requirements grounded in the state-of-the-art on cyber-security by design, smart grids, reference models, and multi-level modeling. Nevertheless, due to space constraints, we focus on illustrating the artifact and refer to earlier work for further details [5, 4]. Thus, in the following, we (1) discuss the reference model, (2) the main steps of the process model, and (3) present excerpts of the reference model supporting security analysis in the first step of the process model. For illustration purposes, where appropriate, we employ the scenario of a fictitious utility company called ACMe, which wants to roll out a smart metering infrastructure and ensure the coverage of security concerns from the beginning of the project.

Multi-level reference model. The reference model accounts for the terminology used by the community and is based on good practices and existing standards (to ease its adoption and foster trust), among others: the NISTIR 7628 [9] (a reference architecture for defining ideal-type smart grid *scenarios* and associated security *requirements*), as well as the Meta Attack Language (MAL) [10] and icsLang [11]. Fig. 1 presents an excerpt from the reference model. Please note that for readability purposes, we present only selected concepts assigned to different levels of classification. For a detailed description of FMML^{x}, we refer to [8]. Apart from the "traditional" modeling constructs such as (meta)classes (assigned to different classification levels, cf. the number standing on the left side in the class header), attributes, operations, and relationships, it is possible to assign a level of intrinsicness (denoted as a white digit in the black box), which dictates at which level of classification a given property will be instantiated.

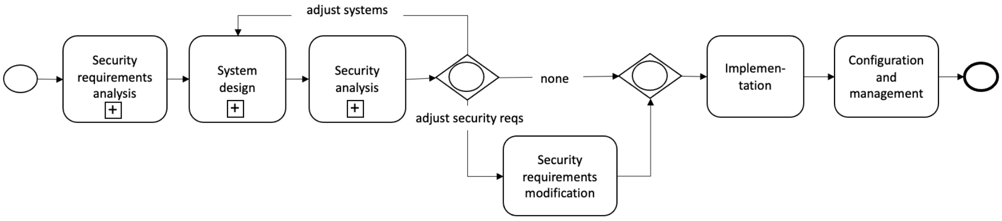


4 ^MetaClass^ Asset
3 role: String[1]
2 overallFunction: String[0..1]
1 function: String[1]
1 maintenanceEffort: Assessment[0..1]
0 introduced: Date[1]
0 missionCritical: Boolean[1]
1 identifyCountermeasures(): Element
1 identifyThreats(): Element
0 analyzeDependencies(): Element
0 identifySpecificThreats(): Element
1 - 3 mayBeAssociatedUsing 1
1 - 3 mayBeAssociatedWith 1
0 isAssociatedUsing 0
0 isAssociatedWith 0
(0,*)
(0,1)
2 ^MetaClass^ AssetAssociation
1 description: String[1]
0 - 3 hasVulnerability 0 - 1
2 ^MetaClass^ SecurityVulnerability
1 assessment: String[0..1]
1 description: String[0..1]
0 knownSince: Date[0..1]
0 refersTo
(1,*)
2 ^MetaClass^ Defense
1 description: String[0..1]
0 applied: Date[0..1]
0 - 3 has 0 - 1
0 - 1
exploitsVulnerability 1
prohibits 1
2 ^MetaClass^ AssetAttackProbability
1 probabilityLevel: Float[1]
1 updated: Date[0..1]
0 specificProbabilityLevel: Float[1]
2 ^MetaClass^ AttackStep
1 averageTimeToCompromise: Float[1]
1 criticality: String[1]
1 description: String[1]
1 tactic: String[1]
0 assessedRiskLevel: String[0..1]
0 reported: Date[1]
0 specificTimeToCompromise: Float[1]
1 ^MetaClass^ SimulationRun
0 date: Date[1]
0 description: String[1]
3 ^Asset^ HardwareArtifact
1 isNetworkCompatible: Boolean[1]
0 serialNo: String[1]
2 overallFunction: String[0..1] (from Asset)
1 function: String[1] (from Asset)
1 maintenanceEffort: Assessment[0..1] (from Asset)
0 introduced: Date[1] (from Asset)
0 missionCritical: Boolean[1] (from Asset)
1 ↑ identifyCountermeasures(): Element (from Asset)
1 ↑ identifyThreats(): Element (from Asset)
0 ↑ analyzeDependencies(): Element (from Asset)
0 ↑ identifySpecificThreats(): Element (from Asset)
role =
3 ^Asset^ SoftwareArtifact
1 averageAvailability: Assessment[0..1]
1 version: String[1]
0 actualAvailability: Assessment[0..1]
0 customMade: Boolean[1]
2 overallFunction: String[0..1] (from Asset)
1 function: String[1] (from Asset)
1 maintenanceEffort: Assessment[0..1] (from Asset)
0 introduced: Date[1] (from Asset)
0 missionCritical: Boolean[1] (from Asset)
1 ↑ identifyCountermeasures(): Element (from Asset)
1 ↑ identifyThreats(): Element (from Asset)
0 ↑ analyzeDependencies(): Element (from Asset)
0 ↑ identifySpecificThreats(): Element (from Asset)
role =
3 ^Asset^ SupportingAsset
2 overallFunction: String[0..1] (from Asset)
1 function: String[1] (from Asset)
1 maintenanceEffort: Assessment[0..1] (from Asset)
0 introduced: Date[1] (from Asset)
0 missionCritical: Boolean[1] (from Asset)
1 ↑ identifyCountermeasures(): Element (from Asset)
1 ↑ identifyThreats(): Element (from Asset)
0 ↑ analyzeDependencies(): Element (from Asset)
0 ↑ identifySpecificThreats(): Element (from Asset)
role =
3 ^Asset^ ComputingPlatform
1 isNetworkCompatible: Boolean[0..1]
0 internalMemory: Float[1]
0 serialNumber: String[0..1]
2 overallFunction: String[0..1] (from Asset)
1 function: String[1] (from Asset)
1 maintenanceEffort: Assessment[0..1] (from Asset)
0 introduced: Date[1] (from Asset)
0 missionCritical: Boolean[1] (from Asset)
1 calculateComputationalCapacity(): Element
0 calculateSpecificComputationalCapacity(): Element
1 ↑ identifyCountermeasures(): Element (from Asset)
1 ↑ identifyThreats(): Element (from Asset)
0 ↑ analyzeDependencies(): Element (from Asset)
0 ↑ identifySpecificThreats(): Element (from Asset)
role =

Figure 1: Excerpt of the Multi-Level Reference Model, for overview of FMML^{x}'s concrete syntax, cf. [8]

Looking at Fig. 1, the model provides an integrated view of the relevant aspects such as assets, their connections, vulnerabilities, or attacks. For instance, in line with the concepts of MAL and icsLang, the model accounts for the characteristics and dependencies among Attack steps (`AttackStep`, Level 2 (L2)), involved Assets (`Asset`, L4), `Countermeasures` (Defenses) (L2) and `Vulnerabilities` (L2). Please note that the designed model uses numerous classification levels to account for generic information (e.g., different categories of assets) and specific information (e.g., specific threats and vulnerabilities of specific IT components). Furthermore, by capitalizing on a relaxed type-instance dichotomy, we incorporate into the model (by assigning a state to classes, populating the model with objects, and defining links) the current knowledge about, e.g., possible attack vectors and their effects. Next, we use the defer instantiation mechanism (in this case, the intrinsicness) to constrain the instantiation of properties to a specific classification level [8]. Finally, to provide semi-automated support for a variety of security analyses, the created reference model also supports a functional perspective.

A macro process for cyber-security by design. Our modeling method, cf. Fig. 2, which is based on [12, 13] and [9, pp. 8-9], consists of six main steps. The first three steps are primarily supported by our reference model. Briefly, in the first step of security requirements analysis,



Security requirements analysis
System design
Security analysis
adjust systems
none
adjust security reqs
Security requirements modification
Implemen- tation
Configuration and management

Figure 2: The cyber-security by design modeling method

we rely on our reference model to elicit security requirements according to priorities from a set of predefined use cases; in the second step of systems design, we select required assets and their interrelation, whereas in the third step, we perform a security analysis on the identified assets, in terms of identifying threats, attacks, and countermeasures.

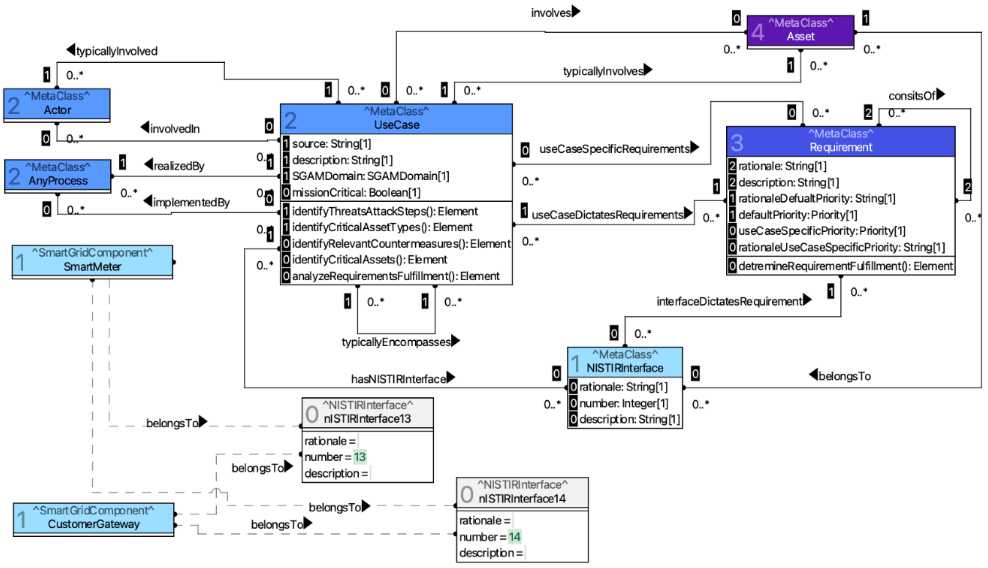
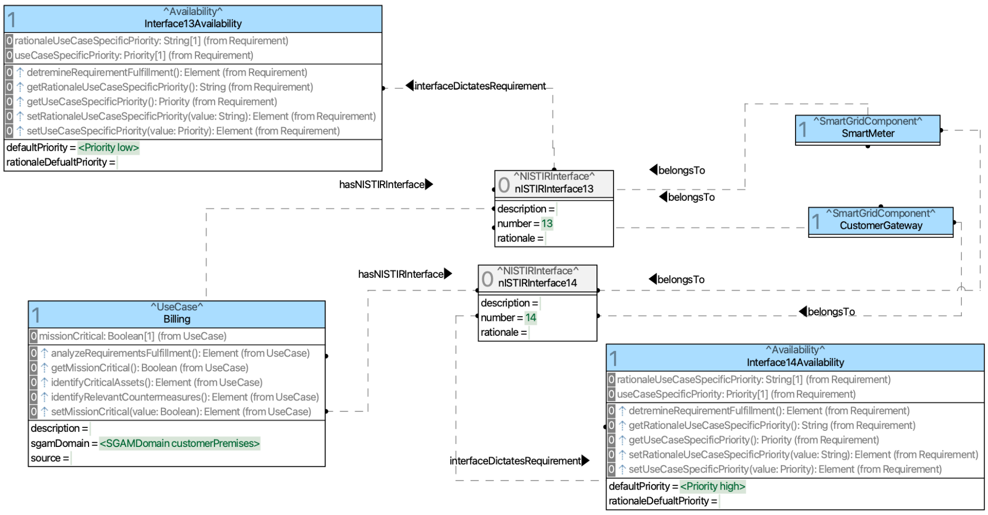


Figure 3: The security requirements view: an excerpt of more invariant concepts



^Availability^
Interface13Availability
rationaleUseCaseSpecificPriority: String[1] (from Requirement)
useCaseSpecificPriority: Priority[1] (from Requirement)
detremineRequirementFulfillment(): Element (from Requirement)
getRationaleUseCaseSpecificPriority(): String (from Requirement)
getUseCaseSpecificPriority(): Priority (from Requirement)
setRationaleUseCaseSpecificPriority(value: String): Element (from Requirement)
setUseCaseSpecificPriority(value: Priority): Element (from Requirement)
defaultPriority = <Priority low>
rationaleDefualtPriority =
interfaceDictatesRequirement
^SmartGridComponent^
SmartMeter
belongsTo
hasNISTIRInterface
^NISTIRInterface^
nISTIRInterface13
description =
number = 13
rationale =
^SmartGridComponent^
CustomerGateway
^NISTIRInterface^
nISTIRInterface14
description =
number = 14
rationale =
^UseCase^
Billing
missionCritical: Boolean[1] (from UseCase)
analyzeRequirementsFulfillment(): Element (from UseCase)
getMissionCritical(): Boolean (from UseCase)
identifyCriticalAssets(): Element (from UseCase)
identifyRelevantCountermeasures(): Element (from UseCase)
setMissionCritical(value: Boolean): Element (from UseCase)
description =
sgamDomain = <SGAMDomain customerPremises>
source =
^Availability^
Interface14Availability
defaultPriority = <Priority high>
rationaleDefualtPriority =

Figure 4: The security requirements view: an excerpt of concepts specific to the scenario

Security requirements analysis supported by multi-level modeling. To provide a more concrete description of our reference model, let us now zoom in on the first step: security requirements analysis. During security requirements analysis, we rely on the reference model excerpts presented in Fig 3 and Fig. 4. Note that these two figures present security-related concepts across multiple abstraction levels: from more invariant concepts, such as Use case and Requirement (in Fig. 3) to more scenario-specific ones, such as Billing being a specific Use case, and according Availability as a security requirement for Billing (in Fig. 4).

Let us now dive into further detail for the step of security requirements analysis, whereby we focus on the ACMe billing scenario. We start with selecting a *Use Case*, consistent with the scenario-driven nature of our modeling method, and specify it in turn of relevant attributes. An example of such an attribute, cf. the security requirements analysis view in Fig. 3, is the SGAM Domain. We subsequently select the relevant *NISTIRInterface(s)* associated with the selected use case. Here, we identify the NISTIR interface by a number, in line with the NISTIR 7628, but for usability purposes, we also offer a short description of the NISTIR interface. The identified NISTIR interface(s) bootstrap the associated security *Requirements*. The requirements, especially the associated priorities, are of relevance. These priorities are set following the necessities of the NISTIR interface in question, and for informed decision-making, the priorities are accompanied by a rationale. Nevertheless, while the priorities are set with a default value (as per the attribute defaultPriority), for a specific use case, a deviating priority can also be selected (as per the attribute useCaseSpecificPriority). The output of this step is a set of prioritized security requirements, a given use case, and NISTIR interfaces.

When exemplifying the above description for our billing scenario, we take as a point of departure *Billing*, being a *Use Case*, which amongst others, is characterized as being relevant for customer premises (a value for the attribute *sgamDomain*, cf. Fig. 4). Subsequently, we notice that Billing is associated with two NISTIR Interfaces: 13 and 14 (both being a *NISITRInterface* whereby the value of the attribute number is initialized accordingly). Subsequently, from the known NISTIR Interfaces, we can derive the relevant security requirements, as shown in Fig. 4: *Interface13Availability*, which (cf. the logic from the NISTIR 7628) has the priority set to low, and *Interface14Availability*, which has the priority set to high. The rationale for these derived priorities is that for typical metering scenarios, availability, in terms of metering data being transmitted in (near) real-time to a utility, is less important relative to the requirements of (1) the metering data received by a utility being an accurate reflection of the amount of electricity that is consumed and produced (cf. the requirement 'Integrity'), and (2) that the metering data can only be accessed by the relevant parties (cf. the requirement 'Confidentiality'). The output for this step is the billing use case, NISTIR interface 13, and security requirements with associated priorities, especially the requirement 'Availability' having the priority low.

3. Challenges and Open Issues

Involvement of domain experts. In line with the design science research approach followed, we have ensured the involvement of domain experts at different stages of our work. In the first phase of the project, we employ semi-structured interviews with two experts from the security and electricity sector respectively, and one expert for security in the electricity sector, to gain feedback on the first version of our reference model, as well as the overall goals and vision of our project, see [5]. We engage with another domain expert, once the supporting process model has been created. We conducted an in-depth two-hour interview during which we presented all constituents of our method and asked the domain expert about the usefulness and completeness of the presented process, the involvement of stakeholders while using the reference model, and any other comments he might have. The domain expert had a master's and a PhD degree in Software Engineering, and several years of experience as an IT supporter for network and server architecture, as well as a senior software architect at the Danish transmission service operator (TSO) since 2016.

Challenges. Based on the conducted interviews, we observed three types of challenges for applying our reference model that can be generalized to modeling challenges. These challenges relate to detailed and up-to-date information (Challenges 1 and 2), management of the model itself (Challenge 3), and a need for incentives (Challenge 4).

Challenge 1: Automated creation of IT infrastructure models, especially concerning Operation Technology (OT). *Rationale:* The domain experts liked the potential of linking the model to code and the flexibility in expressing information at different levels of abstraction. Indeed, in the last interview, in terms of potential, a favorable comparison was made to related modeling languages, especially to the Enterprise Architecture (EA) modeling language ArchiMate, which is currently used internally at the organization of the domain expert. The domain expert recognized the potential flexibility of multi-level modeling compared to ArchiMate by allowing adding new concepts while still providing a formalism allowing automation. At the same time, describing that the recommendations for the displayed scenario were considered a bit high-level and were not pointing towards complex IT infrastructure currently being a part of the organization.

In this light, the domain expert pointed out that he liked the potential of multi-level modeling to keep a model in sync with code and generally keep the model in sync with organizational data. One remark was, for example, especially data on the IT infrastructure, e.g., in line with the idea of infrastructure as code. Nevertheless, our domain expert did point out some challenges in keeping a model in line with IT infrastructure data. One issue is that, especially in the electricity sector, one has to consider the Operation Technology (OT) next to the IT. In particular, organizational stakeholders are careful when it comes to, e.g., scanning the OT infrastructure. Our expert reported that scanning the OT infrastructure by classical IT means letting the different devices crash, leaving the OT infrastructure unusable. Considering it, although newly proposed approaches to automatically create IT landscape models [14], live-IT models, or using artificial intelligence mechanisms to automate the model creation process may be used, additional work is required to meet the specific needs of the electricity sector.

Challenge 2: Accounting for ever-changing threats and countermeasures. *Rationale:* During the demonstration of our reference model, we used a smart metering scenario that was, among others, prone to impersonation attacks, and for which a countermeasure would be to cater for authentication so as to check for user identity. As a reaction, the domain expert stated that while sensible, the attacks and countermeasures encoded in the reference model could use further detail to have a substantial practical impact. Indeed, the domain expert suggested that to perform a security assessment of the system with a more practical impact, it is essential to link the infrastructure assets of a model to known and up-to-date threats and security vulnerabilities so that meaningful recommendations can be deduced from it. For example, such threats and vulnerabilities can be derived from the National Vulnerability Database (NVD[1]).

According to the expert, attacks and countermeasures must be kept in line with up-to-date security information regarding the running system and external sources, such as repositories with vulnerabilities. As such, we observe an overlap with Challenge 1, in the sense that it is crucial that, for future reference, we capitalize on the potentials offered by multi-level modeling and keep the model up-to-date with detailed data on, both the running infrastructure [15] and threats and countermeasures [16].

Challenge 3: Model management for multi-level modeling. *Rationale:* While the reference model was deemed to hold potential, it was also deemed comprehensive. As a result, it was suggested that different views would be required, which present, at different levels of abstraction, different needs/views of both (1) stakeholders using information from the model, as well as (2) stakeholders that keep the information in the model up-to-date.

In light of this, we would like to highlight different points of attention, especially the need for model management mechanisms for multi-level modeling. Such model management mechanisms [17] may take as a point of departure notions from conventional model management, such as views and viewpoints. Still, they should also cater to the particularities of multi-level modeling, like the flexible number of classification levels.

Challenge 4: A need for incentives. *Rationale:* While there is a need for automatically keeping the multi-level model up-to-date with domain information (see Challenges 1 and 2). Indeed, not all concerns related to modeling can be automated. Prominently user input is required on conceptual issues, such as deciding on new scenarios for which security by design would be required. For these concerns, the modeling needs to be performed by domain experts. However, these are classically rarely available, and thus, an incentive is needed so that their input is provided. Moreover, the incentives might need to encourage not only providing information once but to encourage continuous participation. Moreover, making maintenance as easy as possible is vital to keep the effort minimal. This also means that the notion of the multi-level model might be abstracted away as not to confuse the respective experts.

As such, to ensure the relevance of our reference model, we run into classical issues on (1) modeling for the masses [18], which reflects on establishing a bridge between "model like" artifacts like spreadsheets which contain valuable domain knowledge, and a conceptual (enterprise) model, as well (2) the Return on Modeling Effort [19], the highlight of which is that effort put into domain modeling should be met by benefits that one can reap from it.

4. Conclusions

In this work, we have presented the key takeaways resulting from the involvement of domain experts in designing a reference model and a supporting process model. Here, we have highlighted the outcomes of challenges identified in the most intensive interview. These outcomes are not surprising but stress well-known needs in the conceptual modeling domain, such as (1) the automation of modeling tasks and (2) incentives for different stakeholders to provide their knowledge. These insights will shape our future endeavors of evolving our reference model and the linked modeling process to ensure its practical relevance.

[1] https://nvd.nist.gov/

References

[1] M. Z. Gunduz, R. Das, Cyber-security on smart grid: Threats and potential solutions, Computer networks 169 (2020) 107094.

[2] T. Roth, M. Utz, F. Baumgarte, A. Rieger, J. Sedlmeir, J. Strüker, Electricity powered by blockchain: A review with a european perspective, Applied Energy 325 (2022) 119799.

[3] C. Alcaraz, S. Zeadally, Critical infrastructure protection: Requirements and challenges for the 21st century, International journal of critical infrastructure protection 8 (2015) 53–66.

[4] S. Hacks, M. Kaczmarek-Heß, S. de Kinderen, D. Töpel, A multi-level cyber-security reference model in support of vulnerability analysis, in: Enterprise Design, Operations, and Computing, Springer International Publishing, Cham, 2022, pp. 19–35.

[5] S. de Kinderen, M. Kaczmarek-Heß, S. Hacks, Towards cybersecurity by design: A multi-level reference model for requirements-driven smart grid cybersecurity, in: 30th European Conference on Information Systems, ECIS 2022, Timisoara, 2022.

[6] C. Atkinson, T. Kühne, The essence of multilevel metamodeling, in: Proceedings of the 4th UML Conference, Springer, London, UK, 2001, pp. 19–33.

[7] R. J. Wieringa, Design science methodology for information systems and software engineering, Springer, 2014.

[8] U. Frank, Multilevel modeling - toward a new paradigm of conceptual modeling and information systems design, BISE 6 (2014) 319–337.

[9] NIST Smart Grid Cybersecurity Panel, NISTIR 7628-guidelines for smart grid cyber security vol. 1-3, 2010.

[10] P. Johnson, R. Lagerström, M. Ekstedt, A meta language for threat modeling and attack simulations, in: Proceedings of the 13th ARES Conference, 2018, pp. 1–8.

[11] S. Hacks, S. Katsikeas, E. Ling, R. Lagerström, M. Ekstedt, powerlang: a probabilistic attack simulation language for the power domain, Energy Informatics 3 (2020) 1–17.

[12] J. Geismann, E. Bodden, A systematic literature review of model-driven security engineering for cyber–physical systems, Journal of Systems and Software 169 (2020) 110697.

[13] A. C.-F. Chan, J. Zhou, On smart grid cybersecurity standardization: Issues of designing with NISTIR 7628, IEEE Communications Magazine 51 (2013) 58–65.

[14] M. Kleehaus, N. C. Villasana, F. Matthes, D. Huth, Discovery of microservice-based IT landscapes at runtime: Algorithms and visualizations., in: HICSS, 2020, pp. 1–10.

[15] B. Bebensee, S. Hacks, Applying dynamic bayesian networks for automated modeling in archimate: a realization study, in: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), IEEE, 2019, pp. 17–24.

[16] R. Antrobus, S. Frey, B. Green, A. Rashid, Simaticscan: Towards a specialised vulnerability scanner for industrial control systems, in: 4th ICS-CSR Symposium, 2016, pp. 11–18.

[17] P. A. Bernstein, A. Y. Halevy, R. A. Pottinger, A vision for management of complex models, ACM Sigmod Record 29 (2000) 55–63.

[18] K. Sandkuhl, H.-G. Fill, S. Hoppenbrouwers, J. Krogstie, F. Matthes, A. Opdahl, G. Schwabe, Ö. Uludag, R. Winter, From expert discipline to common practice: a vision and research agenda for extending the reach of enterprise modeling, BISE 60 (2018) 69–80.

[19] G. Guizzardi, H. A. Proper, On understanding the value of domain modeling, EMISA (2022).