A Concept for Engineering Smart Grid Security Requirements based on SGAM Models

Christian Neureiter, Günther Eibl and Dominik Engel
Josef Ressel Center for User-Centric
Smart Grid Privacy, Security and Control,
Salzburg University of Applied Sciences, Austria
Email: {firstname.surname}@en-trust.at

Stefanie Schlegel and Mathias Uslar
OFFIS – Institute for Information Technology
Oldenburg, Germany
Email: {surname}@offis.de

***Abstract*—The Smart Grid Architecture Model (SGAM) is widely used for modelling, requirements engineering and gap analysis. In this paper, a formal method for engineering security requirements with SGAM is proposed. Asset security classes, risks and vulnerabilities are modelled formally and a method for deducing security requirements from these entities in the context of an SGAM model is developed. A reference implementation of this method is presented, which allows the automated extraction of security requirements from SGAM models. This set of requirements can serve as an initial starting point for a thorough security analysis. Experience from practical application demonstrates the usefulness of the proposed approach.**

***Index Terms*—SGAM, Security, Requirements Engineering, Patterns, Risk Assessment**

I. Introduction

When developing systems in the context of critical infrastructures, two dimensions and aspects are of particular interest from the perspective of the architecture development process. In general, different components from different vendors have to act as one system. An important aspect to achieve this is interoperability driven architecture management. The aspect of interoperability is one key element from the so-called Smart Grid Architecture Model (SGAM), a core result from the M/490 mandate of the European Commission [1]. The SGAM has proven to be a meaningful way to properly document a static view onto the individual Smart Grid use cases. However, requirements engineering not only has to focus on the functional aspects [3], but also on non-functional aspects.

Security by design can be seen as one important paradigm for the development of the Smart Grid as a critical infrastructure. Requirements elicitation has proven to be very challenging for all stakeholders in the scope of the Smart Grid. For the aspect of security, various collections for security requirements exist, but have not yet been harmonized. The state of the art provides good blueprints as a starting point for the requirements elicitation process at design time. However, established concepts from software engineering like, e.g., patterns or misuse cases are not used in this very context. Within this paper, we will show a possible contribution in terms of tool support and theoretical models to model security mitigation strategies and threats. SGAM provides a way to properly formalize these. The results have been implemented in the *SGAM-Toolbox* (http://www.en-trust.at/sgam-toolbox), which is a publicly available plugin that extends the Enterprise Architect modeling software with the SGAM.

The paper is structured as follows: After this introduction, section two outlines the related work. Section three introduces the approach taken, focusing on the conceptual framework and risk estimation for security assets. In section four, a reference implementation of the presented approach is described, focusing on applicability of ideas discussed. Section five provides the context of the evaluation from the INTEGRA project. Finally, section six concludes and outlines the need for future work in the context of security requirements engineering based on SGAM.

II. Related work

For the approach presented in this paper, various research fields have to be taken into consideration. First, general methods for requirements engineering, especially in context of security are in focus. This issue is addressed by numerous publications. The elicitation of the requirements used in this paper, descend from the best practise methods presented in [5]. An extension for the presented framework is proposed in [10]. This extension is enhanced by privacy approaches and quality requirements which needs to be fulfilled in the Smart Grid, as a critical infrastructure. Besides these methods, there are different power system specific standards like the IEC 62351 [7] series and the series “critical infrastructure protection” from the NERC [9] available. But even in the specific field of Smart Grids notable work exists. One is the IR 7628 from the National Institute of Standards and Technology in the “NISTIR 7628” series [12]. It is divided into three parts whereas the first Volume is of particular interest for the consideration of Smart Grid security requirements. It describes the overall approach and presents a so-called high-level architecture followed by a sample logical interface reference model. This model is used to identify and define 22 logical interface categories within and across seven conceptual Smart Grid domains. For these, high-level security requirements are described.

The Smart Grid Information Security Report [11] aims to answer technical and organizational needs for sustainable state of the art Smart Grid information security, data protection and privacy. Based on the *Smart Grid Architecture Model*, *Security Levels*, *Data Protection Classes* and the *Security View*

per SGAM layers are introduced and used to provide security requirements and recommendations on their implementations. A standards landscape illustrates the role of standards in requirements implementation and establishes a current picture and a target for this landscape.

Finally, as the proposed approach deals with formal concepts, existing work addressing formalized representations of architectures for Smart Grid systems is of special interest. The European Commission's Standardization Mandate M/490 [4], [13] developed a holistic viewpoint of a comprehensive architecture, namely the aforementioned Smart Grid Architecture Model (SGAM). The work is based on existing approaches and subsumes the different perspectives and methodologies of the needed Smart Grid concepts. An in-depth description can be found in [3].

Even if the original intention of the SGAM was the identification of standardization gaps, its simple and clear structure has turned out to be of great applicability for architecting Smart Grid systems. The utilization of the SGAM in architecting Smart Grid systems has been discussed in detail in [2]. Moreover, a concept for an SGAM based Model Driven Development approach (MDA) is introduced. For this purpose, a Domain Specific Language (DSL) was developed that reflects the structure and elements of the SGAM. The implementation of this DSL was done as UML profile and is integrated in a publicly available toolbox. The metamodel of the implemented DSL will serve as basis for the presented concept.

III. Approach

The goal of the presented approach is to support the engineering of security requirements for Smart Grid systems. Therefore the formal representation of system architectures, based on the metamodel as introduced in [2] should be exploited to obtain a basic set of security requirements for a certain architecture. These generated security requirements can serve as a starting point for the requirements engineering process.

Considering the above mentioned metamodel, the architectural aspects of a Smart Grid system (Component, Communication and Information Layer) can be interpreted as a graph consisting of nodes and edges. Hereby the nodes are represented by the individual *SGAM Components* and the edges are realized as relations of type *ICT Connection, Communication* and *Information Object Flow*.

The basic idea of the presented approach is to map the architectural description, represented by the discussed graph, to a set of *security assets* a_i. Different security needs of e.g. system and network assets can be modeled by associating each security asset a_i with *security asset classes* A_j. Moreover, for every security asset class A_j a specific set of high level security requirements $r_{j,k}$ can be identified. This set of security requirements is denoted as *requirement pattern* $R_j = \{r_{j,1}, ..., r_{j,n}\}$ and can be instantiated for every individual security asset a_i that is associated with A_j.
Thus, in a first step a classification of security assets and an initial requirements engineering for theses security asset classes needs to be done.

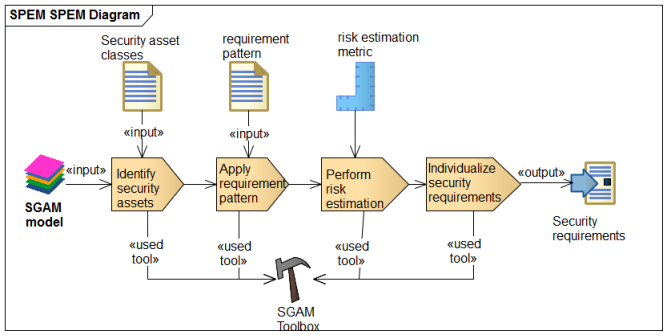

Fig. 1. Requirements engineering process

As these tasks require a formal integration with the existing DSL, the metamodel is extended by an appropriate conceptual framework. The conceptual framework, the classification concept for security assets and the initial requirements engineering are discussed in more detail in Section III-A. In addition, a reference implementation is described in Section IV.

After having obtained a set of high level security requirements (e.g., *authentication*) they need to be refined into more specific requirements (e.g., *two-factor authentication*). Typically the specific requirements for a certain asset are determined on basis of the *risk* the asset is threatened by. Thus, in a first step the risk for every security asset needs to be determined. This task has proven to be very challenging, as the underlying parameters (*potential harm*, *probability*) typically are hard to get, especially the factor "probability".

As the proposed work aims to support the development of real world projects, it focuses on practicable methods. Hence, instead of determining the absolute value for risk, our approach tries to evaluate the risk in a qualitative way that allows to identify the most critical risks. As in real world projects the implementation of security countermeasures always is a trade-off with cost efficiency, the identification of the most critical risks is of higher interest than the absolute value of a security risk. According to this, the approach of qualitative risk assessment promises to be suitable.

However, the presented approach introduces a qualitative method for risk estimation with the *potential harm* being derived from a security assets position within the SGAM plane and the estimation of the *probability* on the basis of preliminary defined *Attack Probability Indicators (API)*.

The concept of risk estimation is discussed in more detail in Section III-B and a possible implementation is described in Section IV. However, the described process for security requirements engineering comprises the four basic tasks *Identify security assets*, *Apply requirement pattern*, *Perform risk estimation* and *Individualize security requirements*. Figure 1 depicts the complete process, together with the used artefacts and tools in more detail.

A. Conceptual Framework

As already discussed, the architectural representation of a certain Smart Grid system should be mapped to a set of

individual security assets a_i that can be assigned to security asset classes A_j. Moreover, for every A_j a set of high level security requirements, referred to as *requirement pattern* R_j, should be provided that can be instantiated for every a_i.

In order to proceed as discussed, it is necessary to extend the above mentioned metamodel from [2] in an appropriate way. Thus, a Conceptual Framework (CF) , following the philosophical ideas of ontology design [6], is created. Basically the CF extends the element *SGAM Component* from the meta-model with security related terms. The structure of the CF is based on the ideas from [5] and depicted in Figure 2.

The basic assumption of the presented CF is the linkability between individual components of a certain Smart Grid architecture to some specific security assets. Of course, one could argue that for example a communication relation also represents a security asset and thus investigating the components will not be sufficient. However, as a communication relation is part of the model and directly linked to the concerning components it will yield a *communication security asset* for every involved component and thus the assumption appears to be suitable.

Following the CF from Figure 2, a certain component *constitutes* one or more security assets. Formally described, every component from a specific Smart Grid system architecture constitutes one ore more security assets a_i, that is associated with a security asset class $A_j \in A = \{A_1, \ldots, A_J\}$. Moreover, every Security Asset class A_j can be related to a set of vulnerabilities $V_j = \{v_1, \ldots, v_K\}$ and $V_j \subseteq V$ with V representing the vulnerabilities for the whole architecture.

In context of *Threats* and *Attacks* a Threat is a *potential* exploitation and an Attack is an *actual* exploitation of of a certain vulnerability v_k, $k \in \{1, \ldots, K\}$, executed by a certain attacker.

In order to *mitigate* the individual vulnerabilities v_k, some specific *countermeasures* need to be implemented. It is the goal of the presented work to support the elicitation of appropriate *security requirements* that are to be realized by the countermeasures. The identified security requirements can be related to the component by a *has* relation.

Considering these relations, for every individual vulnerability $v_k \in V_j = \{v_1, \ldots, v_K\}$, a set of security requirements $R_{j,k} = \{r_{j,k,1}, \ldots, r_{j,k,n}\}$ can be derived. The collection of security requirements $R_j = R_{j,1} \cup \ldots \cup R_{j,K}$ that adresses all vulnerabilities v_k is further referred to as *Security Requirement Pattern* R_j for a specific security asset class A_j.

This leads to the following mapping:

$$A_j \rightarrow V_j = \{v_1, \ldots, v_K\} \tag{1}$$

$$\text{for } k \in \{1, \ldots, K\}: \; v_k \rightarrow R_{j,k} = \{r_{j,k,1}, \ldots, r_{j,k,n}\} \tag{2}$$

$$\Rightarrow V_j = \{v_1, \ldots, v_K\} \rightarrow R_j = R_{j,1} \cup \ldots \cup R_{j,K} \tag{3}$$

which leads to

$$\forall j : A_j \rightarrow R_j = \bigcup_{k \in \{1,\ldots,K\}} R_{j,k} \tag{4}$$

In terms of security it is a good practice to distinguish between different kinds of security strategies such as e.g.

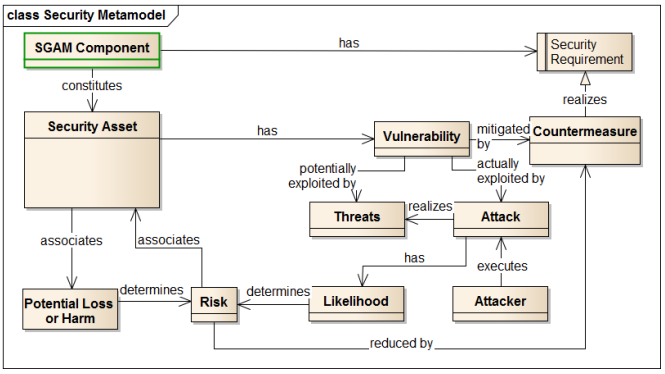

Fig. 2. Conceptual framework

policies and technical measures. Thus, a multilayer security strategy is applied reflecting the structure of the security requirements. Since each single requirement belongs to exactly one Security Layer $s \in \{1, \ldots, S\}$, the set of security requirements $R_{j,k}$ can be decomposed into security requirements for every Security Layer through:

$$\forall k \in \{1, \ldots, K\} : R_{j,k} = R^1_{j,k} \cup \ldots \cup R^S_{j,k} \tag{5}$$

Combining (4) and (5) the final mapping looks as follows (see also Figure 3):

$$A_j \rightarrow R_j = \bigcup_{k \in \{1,\ldots,K\}} \; \bigcup_{s \in \{1,\ldots,S\}} R^s_{j,k} \tag{6}$$

Hereby, the mandatory need for Security Requirements for every individual Vulnerability and for every Security Layer is described by the additional constraint

$$\forall j, k, s : |R^s_{j,k}| \geq 1. \tag{7}$$

As the chosen structure of the security requirement patterns takes reference to individual assets, vulnerabilities and security layers it provides capabilities for an easy and structured requirements assessment. To make the concept more clear, Figure 4 depicts the structure of the security requirement patterns in detail.

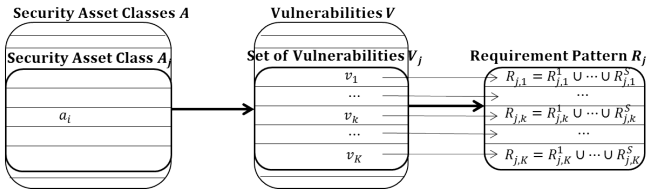

Fig. 3. Mapping $A_j \rightarrow V_j \rightarrow R_j$

The security requirement pattern R_j can be elaborated for every identified security asset class A_j in a general and high level manner. During the requirements engineering process these requirements can be applied to every security asset a_i that is an instance of A_j and hence serve as starting point for further individualization.

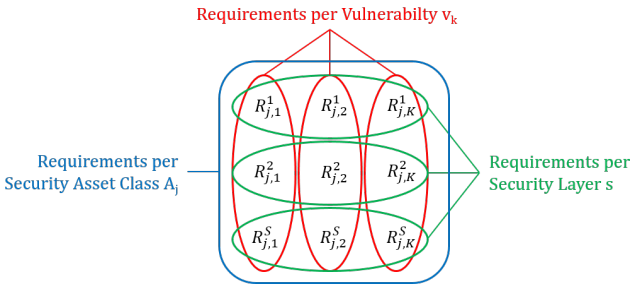

Fig. 4. Structure of a security requirement pattern

B. Risk Estimation

As described in the previous section, for every security asset a_i a set of high level security requirements, based on the instantiated security asset class A_j can be obtained. These high level security requirements (e.g., *Authentication*) need to be refined to more specific requirements, e.g., *authentication with Single Sign On (SSO)* or *two-factor authentication*.

To be provided with a basis for this task, an individual risk assessment for every a_i is necessary. As already mentioned, the quantification of the risk a system is threatened by has proven to be a challenging task. In addition, even if a comprehensive requirements engineering for security requirements is done, often only a sub set of the requirements with high priority is realized.

Following these considerations, it is rather of interest to draw a qualitative picture of the overall risk for individual assets than to exactly determine their individual value. However, the risk for a certain security asset a_i basically can be calculated as product of the *potential harm* and the *probability* for a successful attack. Both of theses values typically are very hard to get, especially the probability for a successful attack.

In [11] a concept is introduced that determines the severity of a potential attack by the potential electric loss. As the SGAM hierarchically decomposes the electric system into domains and zones, the position of a certain component within the SGAM plane can be used to indicate the potential harm. Hereby, this concept maps the position of a component within the SGAM plane to a *Security Level (SL)* between 1 and 5. As this is a very practicable method, our approach utilizes the SL as indicator for the potential harm during the risk estimation process.

One of the aspects of SGAM models is the representation of all systems involved in a certain architecture. This comprises both, systems that will have direct operational effects to the electric systems and systems (e.g., CRM) that don't. However, as we suppose the first group to be more critical, we complement the SL by a factor 2^{DOE} with DOE standing for *Direct Operational Effects* and $DOE = \{0, 1\}$.

The determination of the probability for a successful attack, as mentioned before, is a more complex task. Instead of trying to calculate the exact probability, which is an extensive piece of work, the presented work only tries to draw a qualitative picture of the attack probability. Therefore, numerous *Attack Probability Indicator (API)* should be defined and analyzed and in order to deliver an indication for the probability of a successful attack. Examples for such API's could be *Hacker's motivation*, *Asset reachability* or *Propagation of secret*. Formally, the calculation of the estimated risk can be described by Equation 8;

$$risk = SL \cdot 2^{DOE} \cdot \sum_{i=1}^{n} API_i \tag{8}$$

Hereby, the resulting risk is a relative value without absolute reference or dimension.

IV. Reference Implementation

As described in the beginning, the presented work focuses on the applicability of the presented approach. Thus, the reference implementation aims at evaluating the overall concept rather than the detailed elaboration of the specific parameters. According to this, the introduced elements like *security asset classes*, *security requirement patterns* and *Attack Probability Indicator (API)* only serve as a basis for evaluation. Of course the authors are aware that for real world implementation a more granular refinement is necessary but for the evaluation of the proposed concept, the defined parameters are sufficient.

However, the reference implementation was done as extension to the publicly available *SGAM-Toolbox* in order to provide a seamless integration with existing architecture models of Smart Grid systems. To be more precise, the metamodel of the SGAM-Toolbox has been extended according to the Conceptual Framework as described in Section III-A. Moreover, the elaborated security requirement patterns for the identified security asset classes were integrated as *design patterns* which allows an easy instantiation.

A. Security Asset Classes

In a first step, a set of security asset classes $A = \{A_1 ... A_J\}$ is developed. To identify typical security asset classes, the nature of an attack is analyzed. In general, a successful attack can be said to be the *manipulation of* or the *theft of data from* a specific system. However, in both scenarios a system first must be *reachable* for an attacker, whether by physical access or by breaking into a certain network segment. Having access to a certain system, an attacker can try to manipulate this system by exploiting various interaction channels. One possibility, of course, is to gain direct access to a system by it's dedicated user interfaces. Besides this possibility, one could try to manipulate the communication of a system in order to compromise the system's internal state.

Analyzing the considerations above, three basic security asset classes can be identified:

- Asset Class A_1 : System Security
- Asset Class A_2 : Communication Security
- Asset Class A_3 : Network Security

		A_1 : System Security Asset	A_2 : Communication Security Asset	A_3 : Network Security Asset
L1:	Policies (14)	9	3	2
L2:	Technical measure (22)	8	9	5
L3:	Detection and forensics (16)	7	5	4
L4:	Containment (14)	7	5	2
	Total (66)	31	22	13

TABLE I
DISTRIBUTION OF THE SECURITY REQUIREMENTS

B. Requirements Pattern

After the individual security asset classes A_j have been identified, initial high level security requirements for every A_j can be elaborated. As described in Section III-A, the security asset classes are analyzed in respect to their vulnerabilities V_j and subsequent, these vulnerabilities can serve as a basis for engineering the security requirement patterns R_j for every security asset class A_j. However, as the security requirements should be elaborated in respect to the underlying security strategy, the individual security layers need to be defined first.

The described reference implementation assumes a security strategy consisting of four individual layer:

- Layer L1: Policies
- Layer L2: Technical measures
- Layer L3: Detection and forensics
- Layer L4: Containment

Taking these security layers into account, the requirements engineering process can be executed and requirements for every vulnerability can be derived. Moreover, these requirements can be used to build the security asset class specific requirements pattern:

- Requirement Pattern R_1: System Security
- Requirement Pattern R_2: Communication Security
- Requirement Pattern R_3: Network Security

For this reference implementation the elicited security requirements are closely related to requirements from literature as discussed in Section II.
A total amount of 66 requirements could be identified and integrated to the concerning patterns. The distribution of these requirements in reference to the individual security asset classes and security layer is depicted in Table I.
To make the example more clear, the requirements pattern R_1 for the security asset class A_1 is depicted in detail in Figure 5. This requirements pattern assumes three vulnerabilities that are assigned to the security asset class A_1:

- Vulnerability v_1: Outsider incursion
- Vulnerability v_2: Compromised client attack
- Vulnerability v_3 Evil insider manipulation

As proposed in III-A, the mapping of the v_x leads to the Requirement Pattern R_1

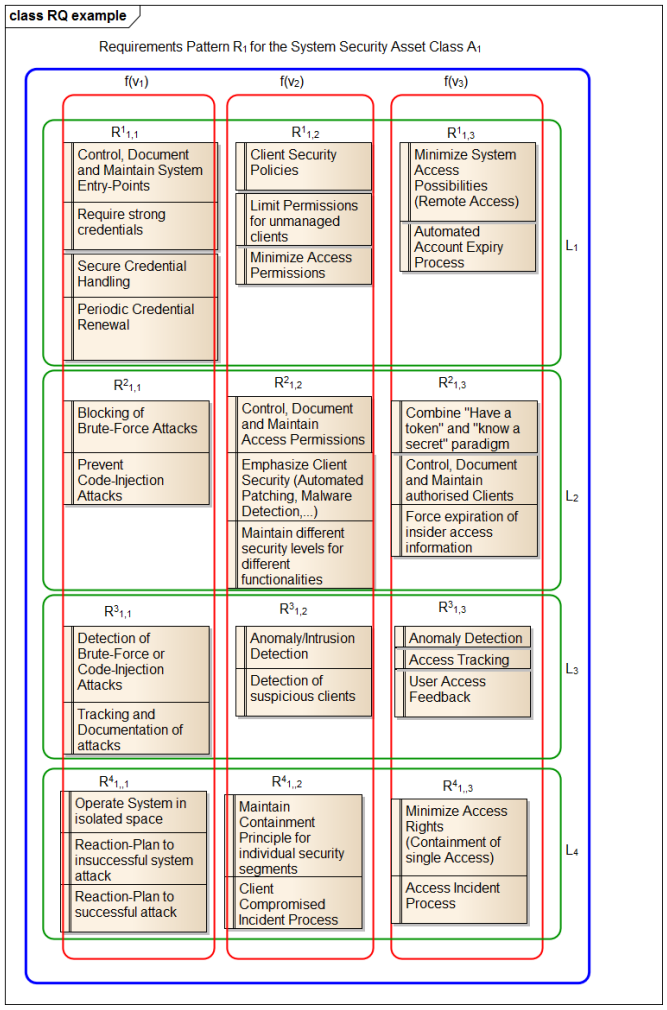

Fig. 5. Requirements pattern R_1 for security asset class A_1

C. Risk Estimation

The basic concept of risk estimation, based on the potential harm and the attack probability was discussed in Section III-B. As the qualitative value for the potential harm can be derived from the position within the SGAM plane, the reference implementation only has to define the *Attack Probability Indicators* (API) that are used to describe the attack probability in a qualitative manner.

Again, the proposed work mainly focuses on the overall process and thus only three representative API's were defined. Of course the authors are aware that in real life projects more indicators need to be identified. The defined API's for the reference implementation are:

- Hacker's motivation
- Asset reachability
- Propagation of secret

Hereby *Hacker's Motivation* describes the attraction of a certain asset for hackers. *Asset reachability* reflects if an asset for example is reachable via the internet, only from the corporate network or by physical presence in a restricted area like for example the control room. The last API, *Propagation*

of secret, considers the number of legitime users that know a secret in order to access a certain asset.

This API assumes that a hacker would try to steal the secret from legitime users for example by means of social engineering. Thus, a higher number of users (potential victims) represent a higher risk. Beside defining the individual API's it is necessary to define the possible values that can be assigned. As these values will have direct impact to the estimated risk, some specific considerations are necessary. First, the values assigned to the API's should yield to similar results if assigned from different people. Hence, it was decided to only allow three different values. Second, the estimated risk as calculated by Formula 8 should reflect a good balance between potential harm and indicated risk and thus, the range of both, potential harm and indicated risk, should be of a similar dimension. As the maximum qualitative value for the potential harm in this reference implementation is 10, the possible values for the API's were defined with $1, 2, 3$ which features a maximum attack probability of 9. Again, even if this reference implementation only focuses on the evaluation of the overall process, it comes clear that some individual considerations are necessary for tayloring the presented approach to the individual needs.

V. Conclusion and Future Application

The presented approach introduces a way on how to obtain security requirements for smart grid systems that are modeled in context of the SGAM. Nevertheless, this approach and the described reference implementation are of a very general type and in order to be applicable for real world implementations, some refinements are necessary.

An application of the proposed concept and the reference implementation get applied in context of the INTEGRA research project. This project aims at deriving a smart grid reference architecture out of numerous realized Smart Grid systems from the Smart Grid Modelregion Salzburg (http://www.smartgridssalzburg.at). The individual systems [8] already were modeled by utilizing the aforementioned toolbox and subsequently provided a valuable basis.

According to the nature of an abstract reference architecture, the modeled architectural solutions are of a very general type. For engineering security requirements this only yields to abstract requirements, as it is not possible to determine for example how many people will have access to a certain asset in a specific implementation.

However, as the proposed work focuses on the introduced process, the architectural models from INTEGRA deliver a suitable basis for evaluation of the presented concepts. The application of the process for requirements engineering in INTEGRA has turned out to be very convenient. Especially the application of specific requirements patterns to automatically obtained security assets is of great value. Nevertheless, it has to be mentioned that these requirements only can serve as starting point and subsequent tasks for refinement (e.g., workshops,...) cannot be skipped.

Even if the described concept delivers a suitable approach, it is important to note that the proposed concept can only serve as a conceptual framework and adoptions to the individual need have to be done. Our future work in this field mainly focuses on a more detailed elaboration of the individual parameters for this approach, like *security asset classes*, *requirement patterns*, *Attack Probability Indicators* and the parameterization of the risk estimation formula.

Acknowledgements

The financial support by the Austrian Federal Ministry of Economy, Family and Youth and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Funding by the Austrian Federal Ministry for Transport, Innovation and Technology and the Austrian Research Promotion Agency (FFG) under Project 838793, "INTEGRA", is gratefully acknowledged.

References

[1] J. Bruinenberg, L. Colton, E. Darmois, J. Dorn, J. Doyle, O. Elloumi, H. Englert, R. Forbes, J. Heiles, P. Hermans, J. Kuhnert, F. J. Rumph, M. Uslar, and P. Wetterwald. CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture. Technical report, CEN, CENELEC, ETSI, 2012.

[2] C. Dänekas, C. Neureiter, S. Rohjans, M. Uslar, and D. Engel. Towards a model-driven-architecture process for smart grid projects. In P. J. Benghozi, D. Krob, A. Lonjon, and H. Panetto, editors, *Digital Enterprise Design & Management*, volume 261 of *Advances in Intelligent Systems and Computing*, pages 47–58. Springer International Publishing, 2014.

[3] H. Englert and M. Uslar. Europäisches Architekturmodell für Smart Grids - Methodik und Anwendung der Ergebnisse der Arbeitsgruppe Referenzarchitektur des EU Normungsmandats M/490. In *Tagungsband VDE-Kongress 2012, Stuttgart*, 2012.

[4] European Commission. M/490 Standardization Mandate to European Standardisation Organisations (ESOs) to support European Smart Grid deployment, 2011.

[5] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt. A comparison of security requirements engineering methods. *Requirements Engineering - Special Issue on Security Requirements Engineering*, 15(1):7–40, 2010.

[6] W. Hesse. Ontologie und Weltbezug – vom philosophischen Weltverstaendnis zum Konstrukt der Informatik. In *Informatik-Spektrum*, pages 298–307. Springer Berlin Heidelberg, 2014.

[7] IEC. 62351-1 TS Ed.1: Data and communication security - Part 1: Introduction and overview, Jan. 2007.

[8] P. Mattle, C. Neureiter, and F. Kupzog. Projekt SGMS – INTEGRA Übergang zu netz- und marktgeführtem Betrieb im Smart Grid. In *Proceedings of the Fourth Workshop on Communications for Energy Systems*, pages 44–52, Vienna, Austria, Sept. 2013. Austrian Electrotechnical Association.

[9] NERC. NERC CIP-002-5.1 to CIP-011-1 Cyber Security, 20012.

[10] C. Neureiter, G. Eibl, A. Veichtlbauer, and D. Engel. Towards a framework for engineering smart-grid-specific privacy requirements. In *Proc. IEEE IECON 2013, Special Session on Energy Informatics*, Vienna, Austria, Nov. 2013. IEEE. to appear.

[11] Smart Grid Coordination Group. Smart Grid Information Security. Technical report, CEN-CENELEC-ETSI, 2012.

[12] The Smart Grid Interoperability Panel Cyber Security Working Group. NISTIR 7628 - Guidelines for Smart Grid Cyber Security vol. 1-3, 2010.

[13] M. Uslar, S. Rohjans, M. Specht, J. Trefke, C. Dänekas, J. M. G. Vazquez, C. Rosinger, and R. Bleiker. *Standardization in Smart Grids: Introduction to IT-related Methodologies, Architectures and Standards (Power Systems)*. Springer, 2012.