A Requirements Engineering-based Approach for evaluating Security Requirements Engineering Methodologies

Sravani Teja Bulusu, Romain Laborde, Ahmad Samer Wazan, François Barrère, Abdelmalek Benzekri

A Requirements Engineering-based Approach for evaluating Security Requirements Engineering Methodologies

Sravani Teja Bulusu, Romain Laborde, Ahmad Samer Wazan, Francois Barrère, Abdelmalek Benzekri
IRIT / Université Paul Sabatier 118 Route de Narbonne, Toulouse, France

Abstract— **The significance of security requirements in building safety and security critical systems is widely acknowledged. From the multitude of security requirements engineering methodologies available today, selecting the best suitable methodology is a challenging task. In a previous work, we proposed a generic evaluation methodology to elicit and evaluate the anticipated characteristics of a security requirements engineering methodology with regards to the stakeholders' working context. In this article, we provide the empirical evaluation of three security requirements engineering methodologies KAOS, STS and SEPP with respect to the evaluation criteria elicited for network SRE context. The study show that none of them provide good support to derive network security requirements.**

Keywords— Security requirements engineering; evaluation methodology.

I. INTRODUCTION

Security requirements engineering (SRE) deals with the process of eliciting, evaluating and documenting security requirements. Several SRE methodologies have been proposed to improve this process[1]–[3]. However, selecting one best suitable SRE methodology still stands as a challenging task to requirement engineers. Although many comparative and evaluation studies of SRE methodologies were made in the past, their evaluation results were not reusable due to various issues such as: ad-hoc criteria, lack of consideration of all the phases of the RE process; and finally non-consideration of the working context of the security requirement engineers [4]. To address this issue, in our previous work[4] we have proposed a generic evaluation methodology using a requirements engineering based approach. This methodology facilitates to elicit the characteristics of good SRE methodology specific to a known SRE context. These characteristics are considered as evaluation criteria for evaluating the SRE methodologies. In the next following work[5] we have briefed on the instantiation of our evaluation methodology to the context of network security requirements engineering. In this article, we discuss in detail the empirical evaluation of three widely recognized SRE: KAOS[1], STS[2] and SEPP[3] with the help of the evaluation criteria for network SRE context. The study show that none of them provide good support to derive network security requirements.

The rest of the article is structured as follows. Section II introduces our evaluation methodology and the example use case for network SRE context. Section III provide the elicited evaluation criteria specific to the given network SRE context. In Section IV we discuss the performance of the SRE methodologies in network SRE context. Finally, we conclude our work in Section V.

II. PRESENTATION OF OUR WORK

A. Our context of SRE methodologies evaluation

Our work is part of the research project IREHDO2 and concerns the aircraft network security engineering. In this project, the security experts of an aircraft company want to improve their security process in order to increase the assurance on the final security solution enforced on their aircraft networks. More precisely, they are interested in enhancing their security requirement practices. This group of security experts includes the security requirement engineers, risks analysts as well as the security testing experts who are involved at different levels of the security process. Our task in this project consists proposing the best SRE methodology which will help them in writing good security requirements. However, each security expert had a different point of view on what could be a good SRE methodology. As a first attempt, they provided us with a use case scenario summarizing their SRE problem context.

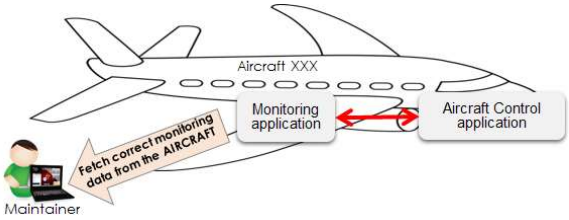

Figure 1: Example scenario context

This scenario in Figure 1 depicts a situation related to the maintenance of the aircraft in order to anticipate its health of the on-board aircraft system by verifying specific parameters. The On-board aircraft system is integrated the aircraft monitoring application and the aircraft control application which are connected to each other via an internal avionic bus network. The maintenance people are allowed to connect their laptops to the monitoring application in order to fetch the monitored parameters. The security goals are expressed in terms of protecting the integrity and availability of the monitored parameters. Security experts needed to derive good network security requirements which can drive them to identify right design solutions i.e., maintenance people can potentially connect to the aircraft using an Ethernet cable or a wireless connection.

Overall, this scenario gathers network security requirements engineering context information in an unstructured format. It provides some insights on what kind of network security requirements can be elicited. However, the question of SRE methodology goodness from the point of view of the security experts is still open. Without this information it will difficult to anticipate what kind of SRE methodology would be interesting to the security experts.

B. Our SRE evaluation methodology strategy

Our proposed SRE evaluation methodology is built on the classical idea of requirements engineering approach by assuming the target system-to-be as the ideal SRE-methodology-to-be that best fits the SRE context. It differs its strategy from previous comparative studies for two reasons. Firstly, it considers the security experts who are the SRE end-users in the whole process. Secondly, it allows the elicitation of SRE evaluation criteria in regards with the anticipated characteristics of a good SRE methodology. Figure 2 depicts an overview of our approach. It subsumes three steps: 1) identifying the problem context and eliciting initial high-level characteristic goals. This is done by coupling the stakeholder's working SRE context as well as the quality criteria of good security requirements; 2) refining the high-level characteristic goals into final requirements of the SRE methodology-to-be (R^M); 3) evaluating the existing SRE methodologies using the elicited requirements (R^M).

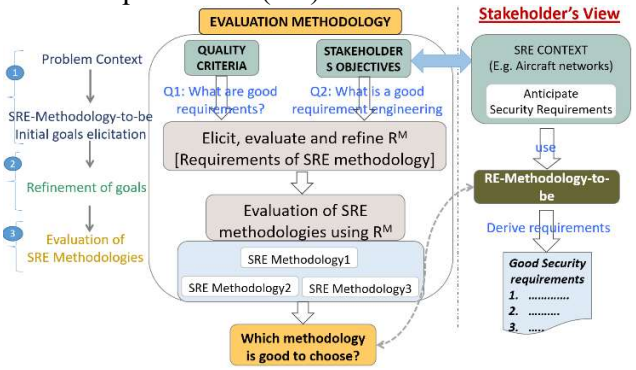

Figure 2: Our evaluation methodology

III. STEP1 AND STEP2: ELICITATION OF EVALUATION CRITERIA

In [5] we have illustrated the elicitation process in the given scenario context which concerns the first two steps of our evaluation methodology. We used the brainstorming technique to encourage the people to exchange ideas on the "best suitable SRE methodology" befitting their needs. Ideally, the ultimate goal of the security experts is to derive good security requirements. The SRE-methodology-to-be is a way to achieve this goal. They are refined into sub-goals that ultimately represent the anticipated characteristics of SRE-methodology-to-be. We represent the elicited goals using KAOS goal modelling notation, see Figure 4. The root goals represent the characteristics of good security requirements. The refinement uses the AND-construct and it is continued until the final refined goals are realized as verifiable. The leaf goal nodes are realized as verifiable eventually become the evaluation criteria R^M. The verification method reflects the suggested way used for evaluating the performance of the SRE methodology against the evaluation criteria. Respectively, the type of verification and expected performance metrics differs with respect to the type of evaluation criteria. For instance, if we consider the evaluation criterion **R^M6.2**. The verification method must facilitate to evaluate the supportability of the SRE-methodology-

to-be in capturing risk attributes related to environmental constraints and interaction dependency constraints, risk priority information. Respectively, the performance metrics to measure the evaluation of this criterion is given in Table 1. The qualitative scale used for performance measure expresses the degree of supportability, i.e., *high* – highly supportable, *medium* – partially supportable, *low* – less likely supportable and *nil* – not supportable.

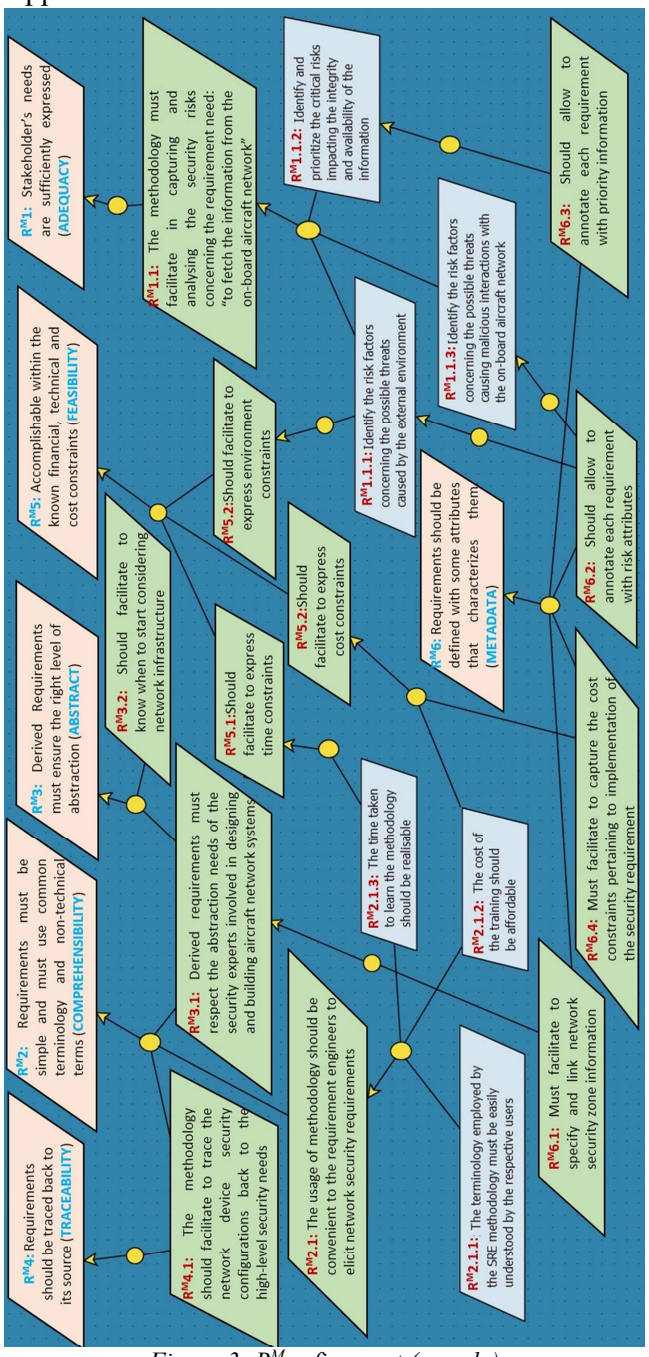

Figure 3: R^M refinement (sample)

Table 1: Verification method for R^M6.2

Verification method	Performance measure
Requirement cannot be annotated with any risk information	nil
Requirements can be annotated with at least one of the attributes	low
Requirements can be annotated with risk priority and threat events	medium
The annotation feature is extensible. Requirements can be annotated with multiple risk attributes.	high

IV. STEP3: EVALUATION OF SRE METHODOLOGIES

The goal of step3 is to test the performance of the SRE methodologies using the elicited evaluation criteria from previous steps. For evaluation, we choose three widely recognized methodologies: Secure KAOS (a goal-oriented methodology – noted KAOS) [1], Secure Socio-Technical System (an agent-oriented methodology – noted STS) [2] and Security Engineering Process using Patterns (a problem-oriented methodology – noted SEPP) [3].

For practical experimentation, a description of the system-to-be as explained in use case scenario (in section II) is presented to three different persons whose initial knowledge fits the aforementioned methodologies the best. Then, each one of them has been asked to elicit security requirements for system-to-be using the methodology that they is familiar with. They weren't allowed to communicate during the requirement analysis phase. Each of them has come up with a different list of security requirements for system-to-be with respect to the example scenario. The results of their works were presented during a meeting that involved the security experts.

In Figure 8, we resumed the evaluation results of the SRE methodologies (in tabular format). From our experimentation we observed that each of these three SRE methodologies exhibit different capabilities with respect to the evaluation criteria. However, when seen from network SRE context, none of the methodologies provides good support. The criteria **RM3.2**, **RM6.1** and **RM6.4** are related to the network security requirements engineering context. In the following we discuss our observations on the performance of the SRE methodologies with respect to the evaluation criteria.

Elicited evaluation criteria list (R^M)	STS	Secure KAOS	SEPP
R^M2.1.1: The terminology employed by the SRE-Methodology-to-be must be easily understood by respective users	high	medium	low
R^M2.1.2: The cost of the training should be affordable	high	medium	low
R^M2.1.3: The time taken to learn the methodology approach should be realisable	high	medium	low
R^M3.2: Should facilitate to know when to start considering network infrastructure	nil	nil	nil
R^M4.1: should facilitate to trace the network device security configurations back to the high-level security needs	mediu m	high	low
R^M6.1: Must facilitate to specify and link network security zone information	nil	nil	nil
R^M6.2: Should allow to annotate each requirement with risk attributes	low	medium	nil
R^M6.3: Should allow to annotate each requirement with priority information	nil	high	nil
R^M6.4: Must facilitate to capture the cost constraints pertaining to implementation of the security requirement	nil	nil	nil

Figure 4: Sample of the evaluation results

A. Secure KAOS methodology

Secure KAOS**,** mainly focuses on eliciting goals and refining them in to sub-goals until they are atomic. Goal refinements are realized via the AND/OR constructs. When a goal cannot be refined further, it is called as the *security requirement* of the system-to-be and is assigned to an agent represented. If a security requirement is assigned to an environment agent (e.g., human), it is called an expectation. The link between a security requirements and a risk is explicitly expressed by the concepts of *obstacles*/*anti-goals*. In addition, KAOS defines some based goal refinement patterns based on a temporal logic in order to introduce formalism. Figure 5 depicts a sample goal model specified in our example scenario context.

We used the KAOS free trial version tool known as *Objectiver* [6]. It took some time and effort to get familiar with the tool and its terminology with the help of available references (***R^M2.1.1***, ***R^M2.1.2*** and ***R^M2.1.3***). Since KAOS drives RE analyst to define *agents* later in the RE process, it does not help in expressing the relation between the agents and their interaction dependencies. While defining the network agents in our scenario context, we had an issue when we needed to add a new device to the network (***R^M3.2***). In the other hand, KAOS notation provides good support to achieve traceability (***R^M4.1***). *Anti-goals* can be refined like 'normal' goals resulting in the specification of attack trees. Obstacles include two risk attributes *likelihood* and *criticality*. However, there is no explicit relationship defined between the priority of a security goal and the risk of an associated obstacle (***R^M6.3***). In addition, it helps in observing the environmental constraints upon the goals through specifying domain properties. (e.g., physical laws), see figure 5.

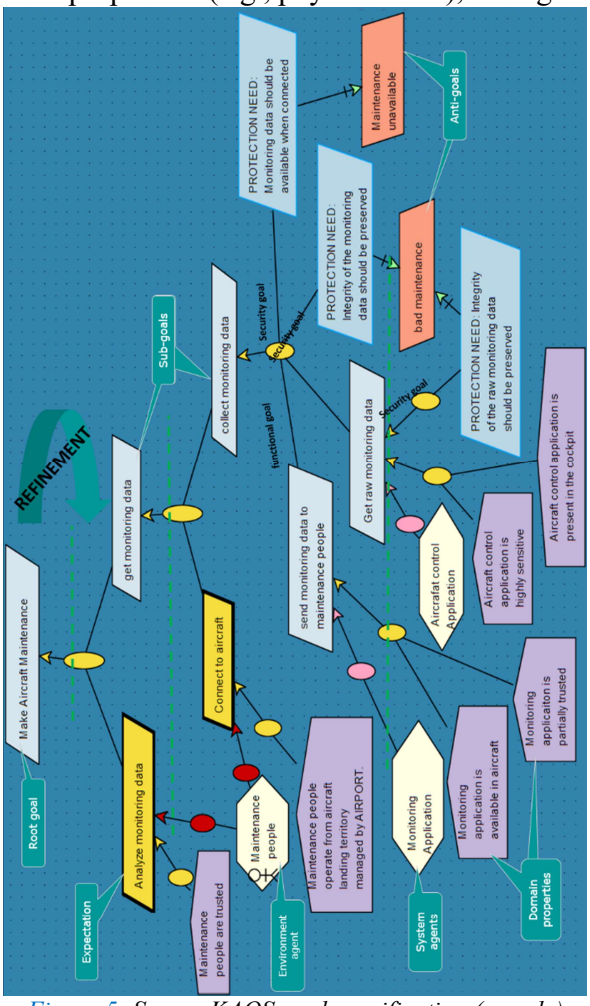

Figure 5: Secure KAOS goal specification (sample)

B. STS methodology

STS mainly focuses on early elicitation of security requirements based on the social

interactions between the agents. Similar to KAOS, STS framework offers the possibility to create composite goals via the AND/OR constructs. However, the respective goals and sub-goals are determined in the scope of each actor. An *actor* can be either a *role* or an *agent.* The social relationships between actors are manifested by the relationships such as *goal delegation* and *resource provision*. This is modelled in the *social view*. *Delegated* goal implies that an actor depends on another actor to achieve a goal. In addition, the social relationships between actors may also include the exchange of *documents* that contain necessary information for the achievement of a goal. Figure 6 depicts a sample social view of the scenario.

We used STS tool [7] that is freely available (***R^M2.1.2***). Simple terminology as well the user friendly tool took less effort to get used to the overall concepts and terminology (***R^M2.1.2*** and ***R^M2.1.3***). However, we had issue in handling the dependencies between the multiple network actors/agents. At some point, the social view became too complex and complicated to express the security interaction needs between the network agents. Furthermore, agent modelling approach defined by STS does not respect abstraction needs of the security experts and it became almost impossible to know when to start considering the network security agents (***R^M3.2***). In addition, STS does not help to achieve traceability in a full-fledged manner like in KAOS (***R^M4.1***). This is because, the *security needs* in STS are expressed implicitly in form of security constraints on the agents' interactions (see figure 6). Furthermore, in STS, the *threat analysis* is used to show the effects over goal trees and goal/resource relationships that may have a threating event, see figure 5. Also, it defines attributes (implicit) to link countermeasures to threat events (***R^M6.2***). However, the threat analysis is limited to threat event propagation.

C. SEPP methodology

SEPP extends the concepts and terminology of problem frames approach to the security context. A *problem frame* is a problem pattern representing the common characteristics of a recognized class of problem. Respectively, SEPP guides the RE analyst to define security problem patterns, called as security *problem frames* (SPF). The security solution patterns, called as *concertized security problem frames* (CSPF) are defined separately. The security problems (security goals) are identified using the what-if analysis technique similar to the Hazard analysis[8]. Figure 6 depicts an example of secure problem frames specification.

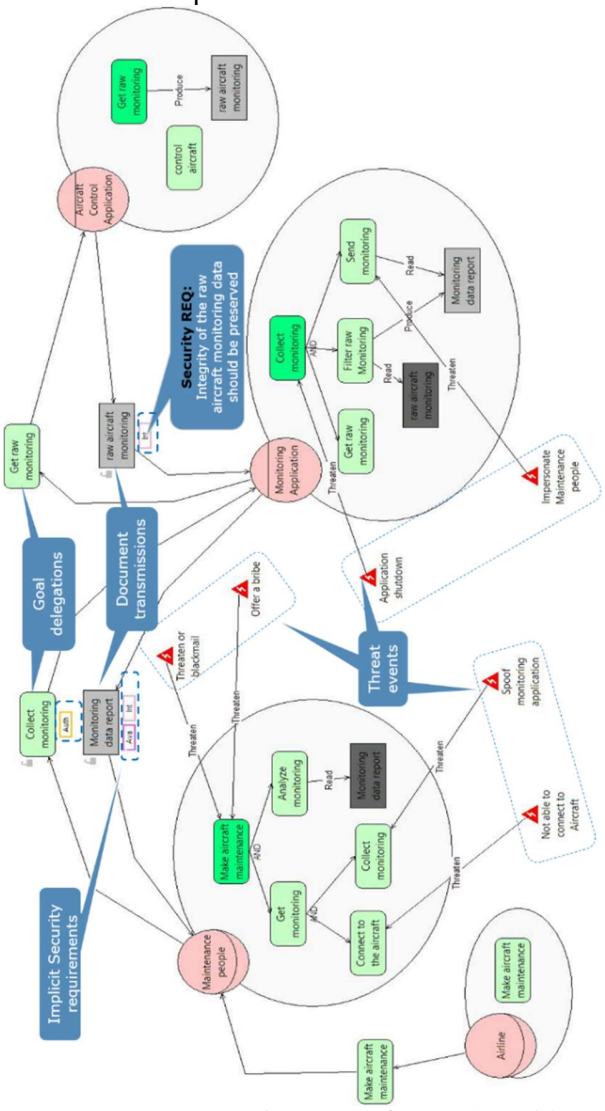

Figure 6: STS social view specification (sample)

The non-availability of a tool for SEPP made our experimentation very hard. As a consequence, in our experimentation we have manually modelled the SPF and CSPF patterns manually. A comprehensive training is must in order to get thorough with its concepts and terminology (***R^M2.1.2***, ***R^M2.1.2*** and ***R^M2.1.3***). SEPP supports reusability these security patterns, by allowing the problem frames separately from security solution frames. However, this feature does not facilitate to specify when to start

considering network infrastructure (***R^M3.2***). During our analysis, we had issues in identifying the all the acting domains in a network environment. This approach seemed to be more suitable if we had known the network design in hand. Furthermore, the constraints on the security requirements are expressed in terms of *pre-conditions* attribute. These are the formalized conditions that must be satisfied by the problem environment on prior, before applying the security problem frame. Similarly, the *post-conditions* attribute correspond to the formal expression of the security requirements.

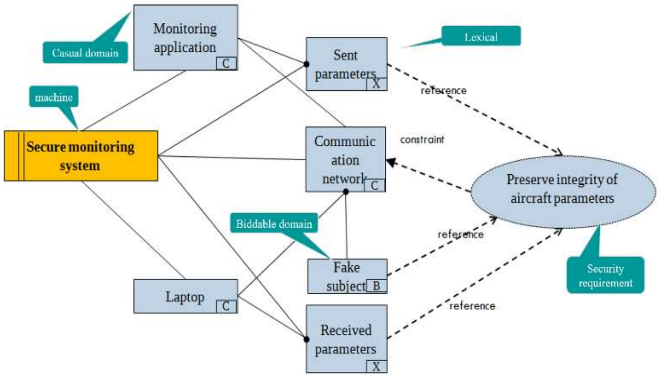

Figure 7: SEPP SPF diagram (sample)

V. Conclusion and Future Work

The principle objective of this article is illustrate the instantiation of our evaluation approach to compare and study three SRE methodologies KAOS, STS and SEPP in network SRE context. Our empirical study show that none of the three SRE methodologies fulfils all the criteria. In particular, they did not satisfy any criteria related to the network security requirement analysis (**R^M3.2, R^M6.1** and **R^M6.4**). It is to note that these evaluation results are purely confined to the context and therefore not to be considered as generalized. That would mean, the evaluation results could change with changing SRE context. Likewise, the performance measure of the criteria verification also differs with regards to the preferences of the security experts. However, there might be some generic characteristics (e.g., *feasibility*, *abstraction, comprehensibility*) that hold common interest of the requirement engineers despite their varying SRE context. Furthermore, our evaluation methodology can be applied to any number of SRE methodologies. This might raise some concerns related to time and costs. In practice, once a SRE methodology is chosen, a lot of time and money is put to train the users and it is very unlikely that one would switch to new methodology soon. Therefore, from industrial usage perspective choosing the best suitable SRE methodology at earlier stages reduces overhead and saves time.

For future works, we would like to apply our evaluation approach to other security engineering contexts. This will help us to determine which evaluation criteria are generic and which are specific to security context. In the end, we intend to build a common repository to maintain the evaluations carried out in each scenario context so that there is no need to re-evaluate a SRE methodology for a similar context already considered in a previous evaluation. Furthermore, this knowledge will constitute a solid foundation to propose future SRE research directions.

Acknowledgment

This work is part of project IREHDO2 funded by DGA/DGAC. The authors thank the security experts at Airbus and the anonymous reviewers for their useful comments.

References

[1] A. Van Lamsweerde, S. Brohez, R. De Landtsheer, and D. Janssens, 'From system goals to intruder anti-goals: attack generation and resolution for security requirements engineering', vol. 3, 2003.

[2] M. Salnitri, E. Paja, and P. Giorgini, 'From socio-technical requirements to technical security design: an sts-based framework', Technical report, DISI-University of Trento.

[3] D. Hatebur, M. Heisel, and H. Schmidt, 'A pattern system for security requirements engineering', in *ARES 2007. the second international conference.*

[4] S. T. Bulusu, R. Laborde, F. Barrère, A. Benzekri, and A. samer Wazan, 'Which Security Requirements Engineering Methodology Should I Choose? Towards a Requirements Engineering-based Evaluation Approach', presented at the ARES'2017.

[5] S. T. Bulusu, R. Laborde, F. Barrère, A. Benzekri, and A. samer Wazan, 'Applying a Requirement Engineering Based Approach to Evaluate the Security Requirements Engineering Methodologies', in *ACM SAC'2018 (To appear)*, Pau, France, 2018.

[6] 'KAOS Tool - Objectiver: HomePage'. [Online]. Available: http://www.objectiver.com/index.php?id=4.

[7] E. Paja, F. Dalpiaz, and P. Giorgini, 'Sts-tool: Security requirements engineering for socio-technical systems', in *Engineering Secure Future Internet Services and Systems*, Springer, 2014, pp. 65–96.

[8] T. A. Kletz, *Hazop and Hazan: Identifying and Assessing Process Industry Hazards*. IChemE, 1999.