CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Repositorio Universidad de Zaragoza

Modelling Security of Critical Infrastructures: A Survivability Assessment

Ricardo J. Rodríguez[†], José Merseguer[‡], Simona Bernardi[§]

[†] *Research Institute of Applied Sciences in Cybersecurity*
University of León, Spain
[‡] *Dpto. de Informática e Ingeniería de Sistemas*
Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain
[§] *Centro Universitario de la Defensa, Academia General Militar, Zaragoza, Spain*
Email: rj.rodriguez@unileon.es, {jmerse, simonab}@unizar.es

Critical infrastructures, usually designed to handle disruptions caused by human errors or random acts of nature, define assets whose normal operation must be guaranteed to maintain its essential services for human daily living. Malicious intended attacks to these targets need to be considered during system design. To face with these situations, defense plans must be developed in advance. In this paper, we present a UML profile, named SecAM, that enables the modelling and security specification for critical infrastructures during the early phases (requirements, design) of systems development life-cycle. SecAM endows security assessment, through survivability analysis, of different security solutions before system deployment. As a case study, we evaluate the survivability of the Saudi Arabia crude-oil pipeline network under two different attack scenarios. The stochastic analysis, carried out with Generalized Stochastic Petri nets, quantitatively estimates the minimisation of attack damages into the crude-oil network.

Keywords: security, software system engineering, UML, survivability, sensitive analysis

1. INTRODUCTION

A critical infrastructure describes a facility, system, site or network whose loose or compromise represents a major impact on the availability or integrity of essential services for daily life operations, leading to severe financial looses, social consequences or even to put in risk human lives. For instance, the U.S. government defines as critical infrastructures the sectors related to water, energy, food and even information and telecommunications [1].

Infrastructures are usually planned to handle disruptions caused by human errors, or by unexpected acts of nature, having a small or not measurable degradation in performance. However, these critical assets represent a main target for terrorist attack plans, aiming at damaging as much as possible governments and citizens [1]. Suppose that a coordinated attack disrupts the power system paths from an energy station, thus losing power to end customers (home users and industries). As a consequence, power-dependent activities stop, then causing economic distress, among other fatalities. This turns to be even worst when an attacker disrupts safety-critical assets, such as the SCADA (Supervisory Control and Data Acquisition) system of a water treatment plant or an oil & gas distribution network thus provoking a poisoning of the population.

As it is stated in [2], critical infrastructures must not only be safe (i.e., able to face random disrupting events), but also secure systems (i.e., able to face malicious intended attacks). That is, a critical infrastructure must have a security level enabling it to identify unexpected but intended events, must handle them and must recover from them. We thus need to develop in advance defense plans to protect critical infrastructures. Survivability strategies [3] - that include resistance, recognition and recovery phases- aim at providing protection for critical infrastructures in the presence of attacks. Since the attacks of September 11, 2001, to the Twin Towers there has been an even growing interest by the worldwide communities on the physical and cyber-security issues in critical infrastructures [1], so security and survivability should be considered as first-class properties in the system development life-cycle. We propose a specific language named `SecAM` (stands for Security Analysis and Modelling) that allows the security specification and modelling of critical infrastructures. `SecAM` is integrated

with UML (Unified Modelling Language [4]) as a Profile [5], and enables survivability analysis for critical infrastructures, which means to provide capabilities for assessing defense plans.

The benefits of integrating security in the design of critical infrastructures are several. For instance, it enables to specify, both qualitatively and quantitatively, security and survivability requirements as well as to carry out sensitive analysis of different security solutions considering different attack scenarios. In this realm, our approach comes to fill some gaps in the literature regarding the modelling and analysis of critical infrastructures. Firstly, we propose models for the critical infrastructure and for attack patterns, these models can be merged to carry out survivability analysis. In particular, the attack patterns address the three phases of survivability previously mentioned – resistance, recognition and recovery –. Secondly, the survivability analysis is addressed through formal methods, in particular, Generalized Stochastic Petri nets [6]. Our approach obtains well-formed Petri nets that can be verified using standard model-checking techniques. The kind of analyses that can be performed is very rich indeed. Although the paper illustrates qualitative analysis and vulnerability analysis, the Petri net also allows to compute resource optimisation of the critical infrastructure, and to provide feedback to the engineers designing a critical system by reporting system bottlenecks (i.e., the slowest part in the system). Last but not least, the kind of techniques that can be used for Petri net analysis is also very rich. In the paper we apply efficient techniques, linear algebra and linear programming-based techniques [7] and steady state analysis, also simulation of the Petri net can be performed.

`SecAM` promotes an integrated view of security considering all stages of the critical infrastructures lifecycle. We show the potential of `SecAM` in Section 6 by assessing security issues of the Saudi Arabia crude-oil pipeline network. We initially model using `SecAM` this critical infrastructure and propose two different attack scenarios. First, a physical attack to multiple network facilities and later a coordinated cyber-attack to the SCADA system. The attack scenarios develop survivability strategies for the system to be timely recovered.

The outline of the paper is as follows. Section 2 describes technical background. Section 3 introduces `SecAM`, developing its core components. Section 4 details the steps we followed to develop the Saudi Arabia crude oil pipeline network case study. Section 5 models the critical infrastructure and the attack scenarios of the case study. Section 6 assesses the survivability analysis of the infrastructure. Section 7 reviews some works in the literature. Finally, Section 8 concludes the paper.

2. TECHNICAL BACKGROUND

2.1. UML and Profiles

UML is a language for systems and software specification. It addresses structural, behavioural and deployment issues. Profiling [5] is a technique introduced in UML to add new capabilities to the language, in our case security modelling and analysis capabilities. A UML Profile is just an extension of the UML defined in terms of:

- *Stereotypes* or concepts in the target domain that will be added to UML. For example, `SecAM` adds stereotypes to model attacks or intrusions.
- *Tags*, the attributes of the stereotypes. For example, for the attack stereotype we define attributes such as its type, objective or location.

`SecAM` relies on two profiles: the standard MARTE [8] (Modelling and Analysis of Real-Time and Embedded systems) and DAM [9] (Dependability Analysis and Modelling), see relationships among them in Figure 1(a). `SecAM` inherits from MARTE its analysis capabilities, among other features, and from DAM those concepts shared by the dependability and security fields [10].

2.2. Generalized Stochastic Petri Nets

A Generalized Stochastic Petri net (GSPN) [6] is a bipartite graph, in which the vertices can be either transitions or places. The transitions represent events that may occur in the system, they can be immediate or exponentially distributed. The former, depicted by black bars, fire in zero time and the latter, depicted by white bars, fire according to the specified distribution. The places, represented by circles, are used to model conditions. The directed arcs, shown by arrows, describe which places are pre- or post-conditions for which transitions. Places may contain tokens, depicted by black dots.

The GSPN dynamics is governed by the transition enabling and firing rules. A transition is enabled whenever there is at least a token in each of its pre-condition places. When it fires, a token is consumed from each of its pre-condition places and a token is produced in each of its post-condition places.

3. MODELLING SECURITY OF CRITICAL INFRASTRUCTURES

`SecAM` is organized in two main packages, the SecAM_UML_Extensions, that includes a set of stereotypes, and the SecAM_Library, that contains basic and complex types used to define the tags of the stereotypes, see Figure 1(a). `SecAM` stereotypes are divided in four sub-packages, Resilience, Cryptographic, SecurityMechanisms and AccessControl, as depicted in Figure 1(b). This organization addresses different security

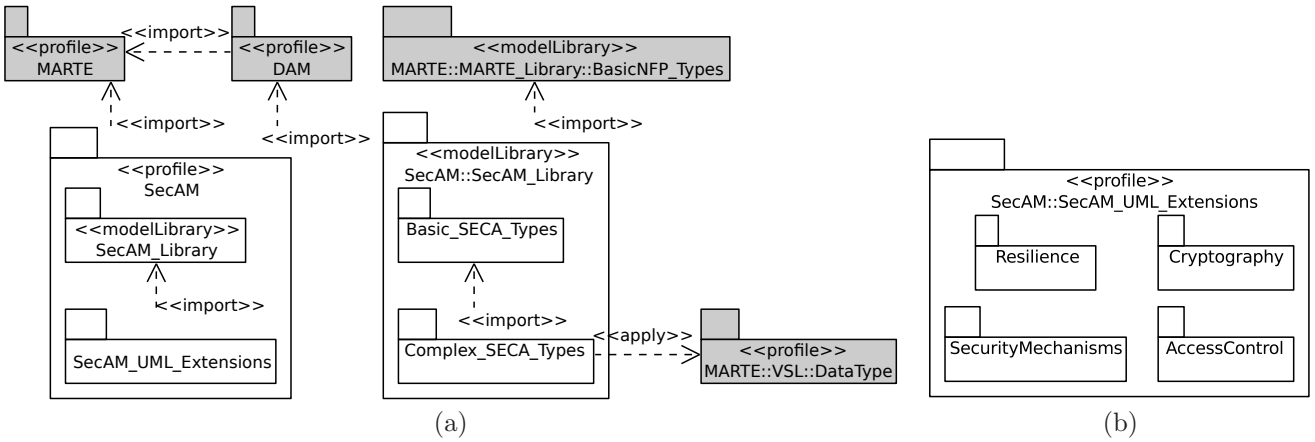


FIGURE 1. (a) `SecAM` profile and `SecAM` Library, (b) `SecAM` UML extensions (subpackages).

Security attributes	SecAM packages (P1)	(P2)	(P3)	(P4)
Integrity	√	√		√
Availability		√	√	
Confidentiality	√	√		√
Authorisation				√
Non-repudiation	√			
Authenticity	√			

(P1): `Cryptographic`; (P2): `SecurityMechanisms`
(P3): `Resilience`; (P4): `AccessControl`

TABLE 1. Security attributes covered by `SecAM`.

issues of relevance for critical infrastructures. These issues are typically dealt by independent research communities, `SecAM` encompasses all of them to produce a common security specification.

We have used a breadth-first approach to provide these common basis for the specification of security. Each sub-package deals with a subset of well-known security attributes [11] (integrity, availability, confidentiality, authorisation, non-repudiation and authenticity) and, as shown in Table 1, the sub-packages overlap with respect to attribute coverage.

3.1. `SecAM::Resilience` package

The Resilience package addresses the (malicious) threat characterization. It enables the specification in UML behavioural diagrams of attacks, vulnerabilities and intrusion concepts, and also their causal relationships (i.e., the attack-vulnerability-intrusion or AVI chain) [12]. The package was initially proposed in [13] to conduct a vulnerability stochastic analysis.

Figure 2 details the package. It mainly contains two stereotypes, `SecaAttackGenerator` and `SecaStep`, that specialise some DAM stereotypes. Hence, by inheritance, they can be applied to all those UML behavioural model elements that can be stereotyped with the DAM ones. For example, `SecaStep` can stereotype action or activity states in activity diagrams. `SecaStep` has three tags, `kind`, `vulnerability` and `intrusion` and also inherits `hostDemand`, which describes its computational requirements, and `prob`, which indicates the probability of the step to be executed (for a conditional execution). Each tag has a type which is completely described in the SecAM_Library, Figure 2 right hand side. For example, the type of `intrusion` is `SecaIntrusion` and it owns three tags, one of them `successProb` to define the probability for the attack to success.

The `SecaAttackGenerator` has the tag `attack` of type `SecaAttack` to completely describe an attack as follows. The type of attack can be active or passive. The different classes of attacks (`ClassOfAttack`) are compliant to the taxonomy defined by Hansman and Hunt in [14], e.g., virus trojan, or worm. The `KindOfAttack` [15] depends on the method adopted by the attacker to succeed in the intent, e.g., injection or resource modification. For the objective of the attack (`AttackObjective`) we can distinguish: denial of service, run arbitrary code, privilege escalation, data modification and information leakage. Considering from where the attack is actuating, three different locations [16] can be identified (`AttackLocation`): single-source (originated at only one host), multi-source (replicated over multiple hosts), and reflector-source (the attacker uses legitimate hosts to attack the victim, hiding so his identity or amplifying his attack [17]). Finally, the `target` of an attack represents execution resources such as classes, instances, components, parts, and deployment nodes stereotyped as `gaExecHost`.

Finally, the concept of coordinated attack is introduced by the complex type `SecaCoordAttack`. A coordinated attack allows attackers to avoid an intrusion detection by splitting a malicious attack pattern in several sub-patterns (`attacks` attribute). It can be classified (`coordType` attribute) as [18]: a *cumulative* attack, when simultaneous attacks are initiated to overcome computer limitations; a *replicated attack*, when several attacks target replicated services occur to bring down the entire service structure; or a *mixed attack*, i.e., a combination of the previous ones.

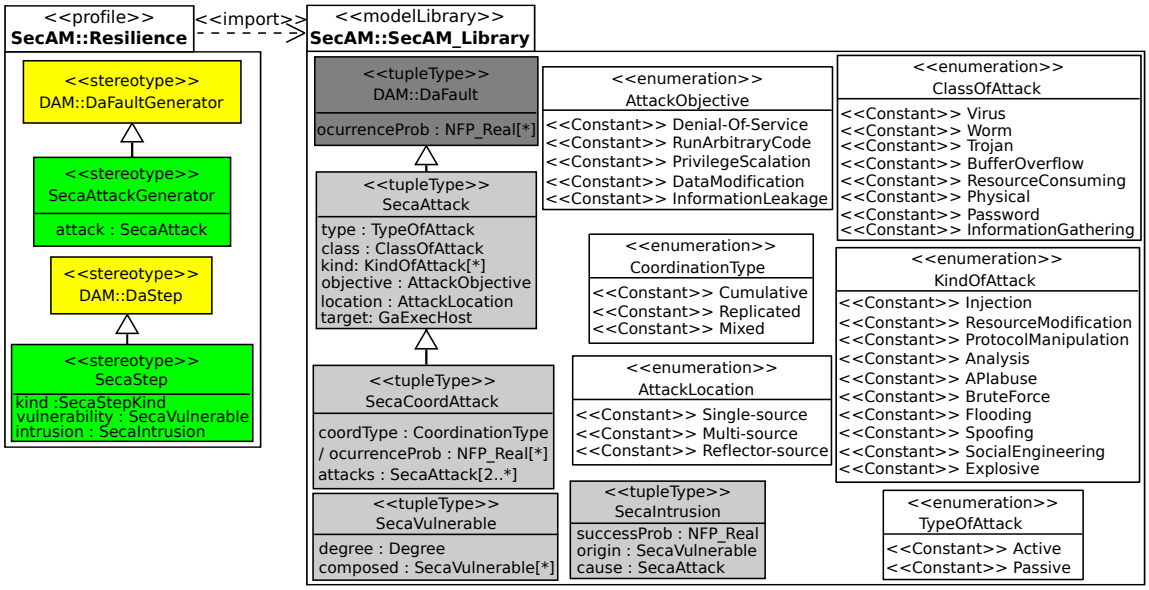


FIGURE 2. The `SecAM::Resilience` package.

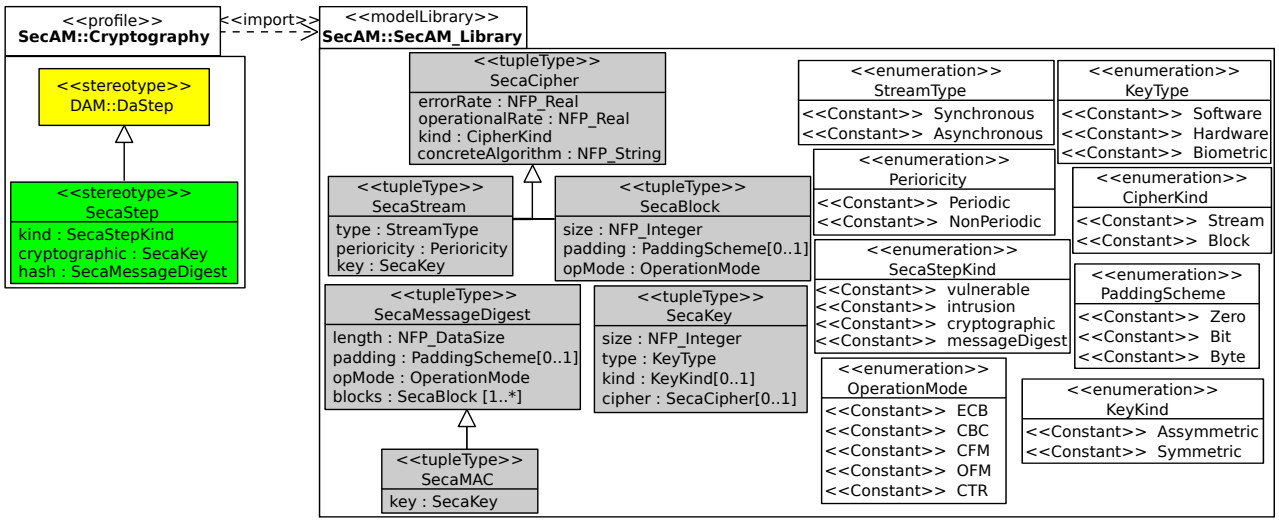


FIGURE 3. The `SecAM::Cryptographic` package.

3.2. `SecAM::Cryptographic` package

Cryptography [19] is primarily used to gain confidentiality over the communications between pairs, however, it also supports data integrity and authentication. The Cryptographic package deals with security properties related to ciphers and algorithms. Hence, the package mainly supports the specification of cryptographic design of the critical infrastructure and its performance characteristics, see Figure 3.

The `SecaStep` stereotype, already considered in the Resilience package, is now used to specify a cryptographic step through the new tags: `kind`, `cryptographic` and `hash`. The `cryptographic` tag is a complex type (`SecaKey`) that enables to characterize the key, either asymmetric or symmetric (`KeyKind`). The latter can be of different types, depending on how/where it is deployed (`KeyType`): software, hardware (i.e., cryptographic devices) or biometric (e.g., fingerprint, facial recognition or retinal scanning). A cipher (`SecaCipher`) can be either a block or stream cipher, depending on the algorithm, and it uses a key. It is characterized by an error rate, i.e., the ratio of errors that the cipher can suffer during the process of encryption/decryption, and an operational rate, i.e., the number of encrypted/decrypted bits per time unit.

A stream cipher (`SecaStream`) uses a key-stream to cipher/decipher plain text, which generates a stream of secret bits given an initial key. It can be either self-synchronous (i.e., *ciphertext-auto-key*, CTAK) or synchronous (i.e., *key-auto-key*, KAK), depending on whether the used key-stream is influenced or not by the ciphered/deciphered text. A block cipher

(SecaBlock) has a block size, that is the number of characters (or bits) of the plain text message which can be ciphered at a time. Normally, the partition of the message into blocks is not exact and, therefore, a padding scheme is needed. Besides, it uses an operation mode (OperationMode) which determines its encryption/decryption scheme, we rely on the ones approved by NIST [20].

A message digest (SecaMessageDigest), also called *hash*, is a value calculated through a cryptographic hash function. Such a value is used, for instance, to determine if a message has been altered. When the cryptographic hash function uses a key (then called keyed hash function), the obtained value is called message authentication code (MAC, SecaMAC). MAC values assure data integrity and data authenticity.

3.3. SecAM::SecurityMechanisms package

The SecurityMechanisms package, depicted in Figure 4, characterises both hardware and software solutions provided by the critical infrastructure to attain security. It comprises basic cryptographic devices [19], more sophisticated ones [21–23], the use of security communication links such as Virtual Private Networks, honeypots [24] or any kind of security software.

A security mechanism can be located either in a host machine or in a network. In the first case, it can be either a software, e.g., the Windows firewall or UNIX packet-filtering rules (SecaHostFirewall). In the second case, it can be a hardware device, e.g., a wormhole [23] or a cryptographic token/accelerator [19]. The type of defense (DefenceType) of the security mechanism can be proactive, reactive or both [23].

Several types of specialised security hardware mechanisms can be distinguished: crypto hardware (SecaCryptoHW), wormholes (SecaWormhole), firewalls (SecaFirewall) and Intrusion Detection and Prevention System (IDPS) (SecaIDPS). Crypto hardware may have a key which is used to perform cryptographic actions (SecaKey complex type in Figure 3).

A wormhole [23] enables both proactive and reactive defense on the system and is characterized by several parameters: the number of tolerated faulty hosts (nFaulty tag), the number of host replicas which can be recovered in parallel (nParallel) and timing parameters (the recovery time, tRecovery, and the period, tPeriod). A firewall [21], usually a network-location device, filters packets from the network at different layers of the OSI architecture (FilterLevel), e.g., data-link or application-based filtering. An IDPS monitors the network (or a single host) to discover security breaches. Several detection methods exist (DetectionMethod), our profile relies on the ones described by NIST [22].

An aggregation of security hardware devices can conform to a demilitarised zone (DMZ) [21]. There exists many configurations for a DMZ, from a single router to a more complex architecture. The hosts deployed on a DMZ zone are called *bastions* (SecaBastion stereotype), they offer secure services, e.g., web, mail, FTP, DNS or honeypot. A honeypot (SecaHoneyPot stereotype) can be a virtual machine, a sandbox (logical) environment or a real system with capability features. Through the operativeSystem tagged-value we can specify the software which is running in such confined environment knowing the security breaches within it. Besides, the monitor tagged-value specifies what operation and what feature are being monitored.

A secure communication link (SecaLink stereotype) is characterised by the number of authentication factor (e.g., one level if only a password is needed, two levels if a correct combination of user/password is needed, and so on), the layer where the secure communication is taking place and the specification of the secure protocol.

3.4. SecAM::AccessControl package

The AccessControl package supports the specification of access control policies to the critical infrastructure. It addresses confidentiality, integrity and authorisation issues. Access control can be classified, depending on the policy, in three basic groups [25]: Mandatory Access Control (MAC)[1], Discretionary Access Control (DAC) and Role-Based Access Control (RBAC) [26].

The small set of stereotypes defined in Figure 5 include the necessary concepts to specify all the aforementioned access control and integrity policies, and also any other access control policies that a user needs to devise for securing the critical infrastructure. Control policies rely on:

1. Who is accessing to the system (SecaSubject stereotype);
2. Which objects should be protected for unauthorised access or modification (SecaObject stereotype);
3. Which operations a user wants to perform on the available objects (SecaOperation).

In SecaSubject the attributes inheritance and exclusion allow to specify delegation of authority and separation of duties, respectively, two major concerns in access control policies. The other attribute, level, allows to specify the level of security of the subject. Such level aims to be defined by the engineer depending on the problem context. For instance, High, Medium, Low levels when designing a military security policy, or Manager, Developer, Administrative when dealing with a company's hierarchy. Namely, Figure 5 shows the NATO clearance levels (TopSecret, Secret, Confidential and Restricted) [27].

The type attribute (SecaOperation stereotype) indicates the type of the operation to be performed.

[1] MAC is also called Lattice-Based Access Control (LBAC).

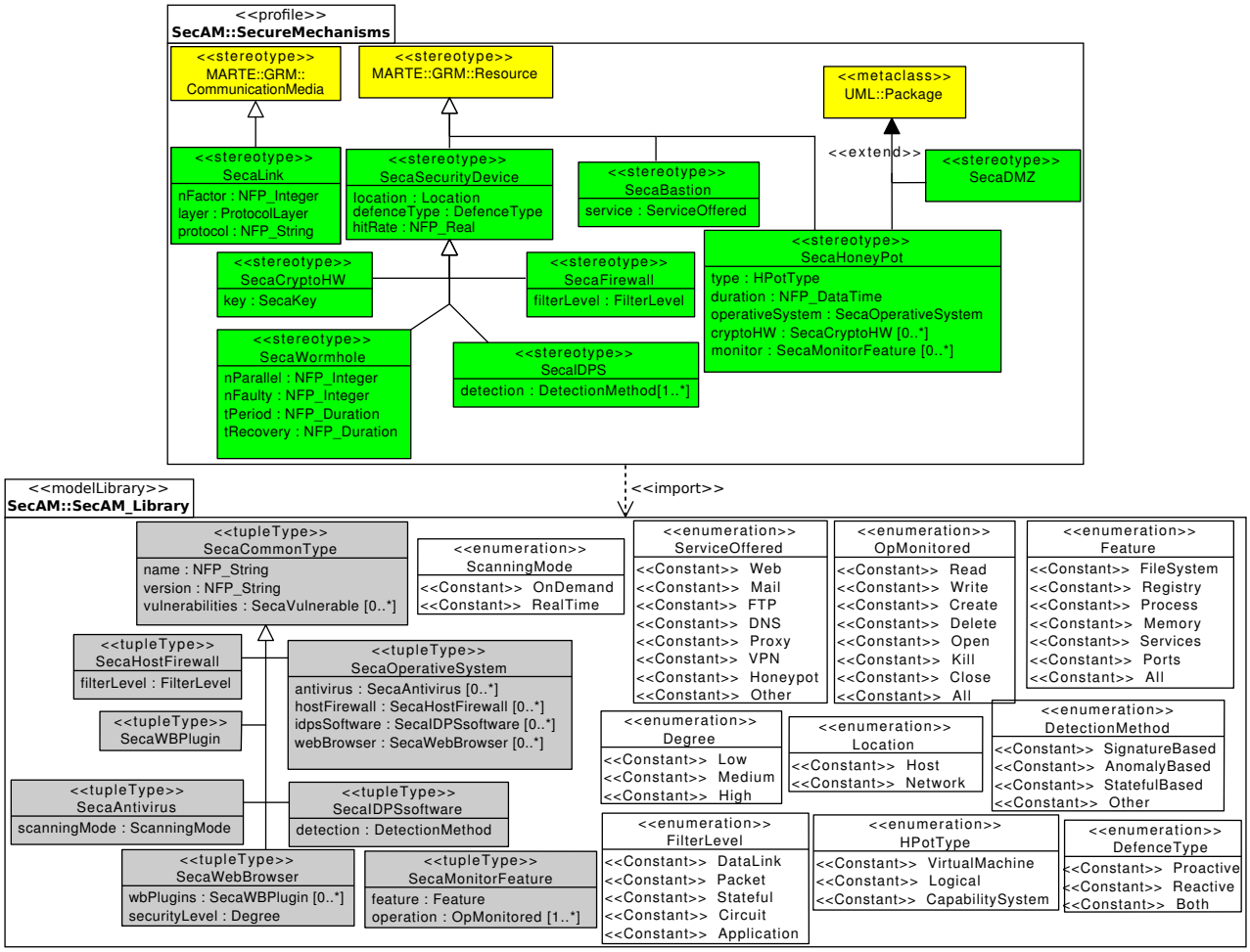


FIGURE 4. The `SecAM::SecurityMechanisms` package.

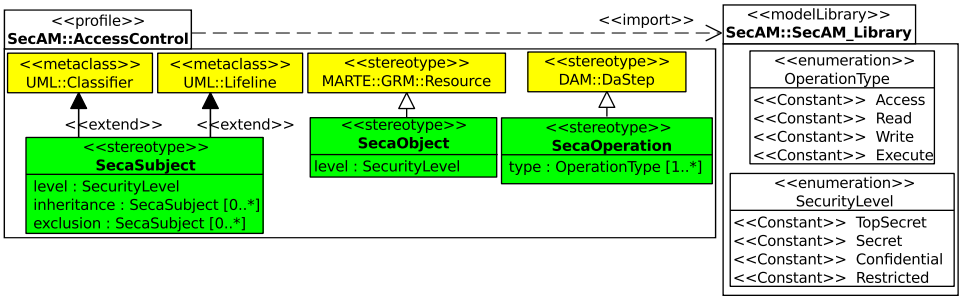


FIGURE 5. The `SecAM::AccessControl` package.

The set of operations (`OperationType` values) can be redefined in order to customize them to the target domain problem. For instance, if the engineer wants to consider a payment as an operation, s/he can define a permission constant such as `payment` which will be valid in his/her domain.

4. TECHNICAL PROCESS

Sections 5 and 6 develop the case study of the Saudi Arabia crude oil pipeline network. These sections follow the main steps outlined in Figure 6 and develop the models proposed in such figure. The model-based process is partitioned in three sides: UML models, Petri net models and analysis and assessment step.

UML models of the critical infrastructure are represented by a flow model and a resource model. The former is developed through a UML activity diagram, while the latter using the deployment diagram. In parallel we develop different attack models using UML activity diagrams also. We consider paradigmatic survivability patterns that may conform to a library of survivability scenarios, for our case study they represent

physical attacks and coordinated cyber-attacks. Both models, the critical infrastructure one and those of the attacks, are annotated using `SecAM`. Finally, each `SecAM` variable in the attack models is parameterized for eventually conducting sensitivity analysis. For example, we can fix probability values for the occurrence of the selected resistance strategies.

Petri net models, in particular Generalized stochastic Petri nets (GSPN [6]), are obtained through UML-2-GSPN model transformation according to the approach in [28]. We obtain a GSPN model representing the critical infrastructure and one GSPN model per attack scenario.

The *analysis and assessment step* starts with a qualitative analysis of the GSPN, which ensures that the net is free of bottlenecks. Then the quantitative assessment proceeds through sensitivity analysis.

5. SURVIVABILITY MODELLING OF A CRITICAL INFRASTRUCTURE

The Saudi Arabia crude-oil pipeline network is a paradigmatic case of critical infrastructure wanted to be attacked by terrorists, in fact it was a terrorist target in 2006 [2]. According to [2] Saudi Arabia was the world's largest producer and exporter of total petroleum liquids in 2010 and the world's second largest crude oil producer behind Russia. Morgan Stanley reports [29] that a reduction in Saudi Arabia crude-oil output to 4 mmbbl/day[2] (only about 5% of world demand), from a current of 8 to 9 mmbbl/day, would cause worldwide economic distress leading to a global recession if the infrastructure could not be repaired in a few months. Unless having survivability strategies leading to quickly recover the infrastructure -in hours or few days-, the report informs that worldwide oil prices shoot skyward.

Saudi Aramco processes the majority of crude oil in a plant at Abqaiq and distributes it through an extensive network of pipelines that connects the plant to three primary oil seaports, Ras Al-Juaymath, Ras Tanura and Yanbu, and two minor terminals, Ras Al-Khafji and Jubail. The deployment diagram in Figure 7 depicts these facilities and the distribution network. This distribution network operates thousands of miles of pipelines and junctions throughout alternative pathways, both domestic and international. Activity diagram in Figure 8 offers an excerpt of the oil flow from the source to the terminals with an example of the time delays for pipes and junctions annotated using `SecAM`. An enormous security force guards the distribution network, but being more than 9000 miles long -only the domestic pathways-, the area cannot be patrolled completely.

Attacks to oil pipeline infrastructures may range from physical ones (e.g., damaging seaport infrastructures, pipeline segments or junctions) to cyber-attacks (e.g., compromising the network control system through a distributed denial of service -DDoS- or infecting it). Survivability strategies should face the reported threats, and may lead to recover the infrastructure -physical and network control system- for guaranteeing oil production timely. We present one standard survivability strategy model for each kind of attack, using `SecAM` to specify both the attacks and the survivability parameters.

The first survivability strategy, Figure 9, identifies the three common stages: if the system does not *resists* the attack then a *recognition* step possibly followed by a *recovery* will try to maintain the infrastructure up. The `secaAttackGenerator` stereotype and its tag `attack` completely describe the physical attack and the resources being attacked (pipes P1, P3 and Qadif junction node). The `secaStep` stereotype is used to model intrusions, as probabilities that will allow a stochastic analysis of the strategy, and the demands for recovering and repairing activities. For the latter we used exponentially distributed random variables, $recovery[3] and $MTTR, which fit the stochastic analysis and also allow to perform sensitivity analysis varying the values assigned to them.

The second survivability strategy, Figure 10, describes a scenario with two coordinated attacks to different nodes of the computing control system of the distribution network. On the one hand, a DoS attack tries to shut down the Qadif node. An IDPS (intrusion detection and prevention system) is the *resistance* strategy that constantly monitorizes the network for malicious activity. For the sake of space, we express in Figure 7 the existence of this IDPS by means of the `SecaIDPS` stereotype in Qadif node (light gray annotation). If the system does not resist *recovering* activities are carried out. On the other hand, an attack to the control node of pipe P1 tries to obtain advantage over a vulnerability in its communication protocol for executing code arbitrarily (thus, gaining access to the control system). The *resistance* strategy is a RSA cryptographic algorithm, which introduces the corresponding delay ($decipher).

6. SURVIVABILITY ANALYSIS OF THE INFRASTRUCTURE

A previous vulnerability analysis of the oil distribution network was carried out in [2, 7]. In particular in [7], a Time Petri Net model was used to evaluate the impact on the network throughput of different attack plans having as objective three network facilities (i.e., two pipes connecting Abqaiq and Yanbu and the junction of Qadif).

Herein, the aim of the analysis is to provide a quantitative stochastic characterization of the survivability requirements, devised for the two attack scenarios specified in the previous section. For this purpose, the survivability steps (resistance, recognition and recovery) have been parameterized in order to

[2] million barrels per day.

[3] Values preceded by $ are variables that will be parameterized during analysis.

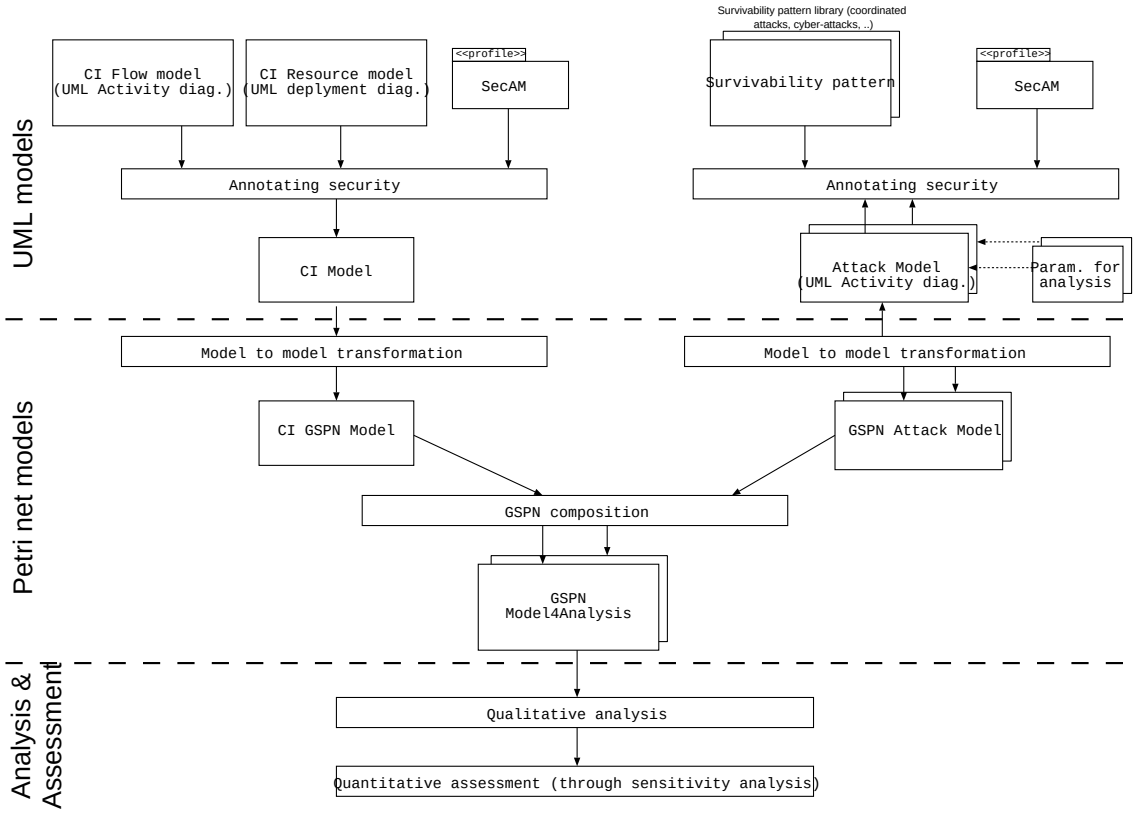


FIGURE 6. Model-based process followed to develop the case study.

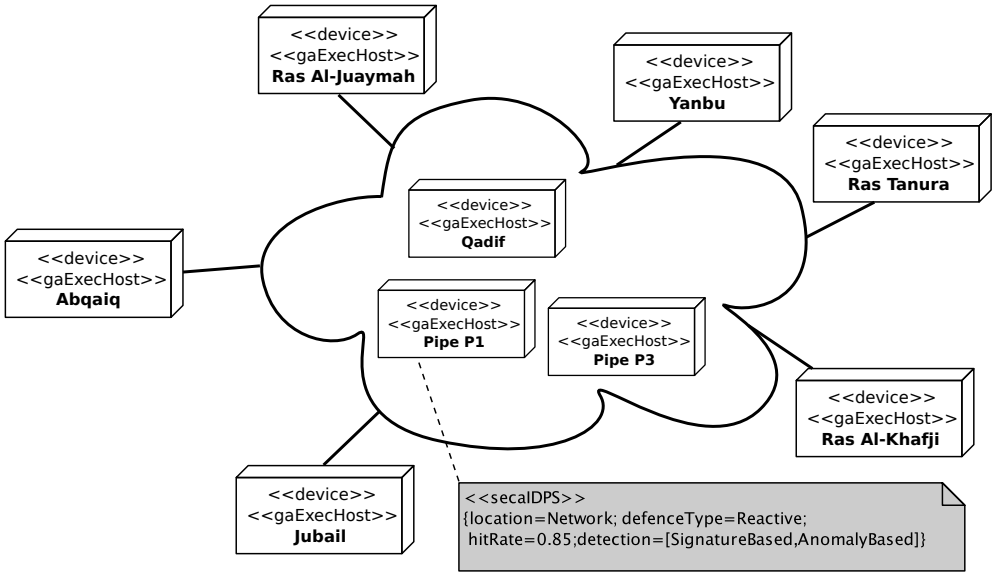


FIGURE 7. An excerpt of the Saudi Arabia crude-oil pipeline distribution network.

evaluate, through sensitivity analysis, the effects of the aforementioned attacks on the network infrastructure.

We reuse the model of the oil distribution network, under stochastic assumptions, and the composition approach in [7] to get the GSPN [6] models of the network under the two types of attacks. A preliminary qualitative analysis of the composed GSPN models has been carry out to check whether they are well-formed and, then suitable for evaluating survivability metrics. In particular, we have applied linear algebra and linear programming-based techniques to verify that:

- the network resources (i.e., pipes and junctions) are used in conservative manner and the oil quantity flowing through the distribution network is preserved, and

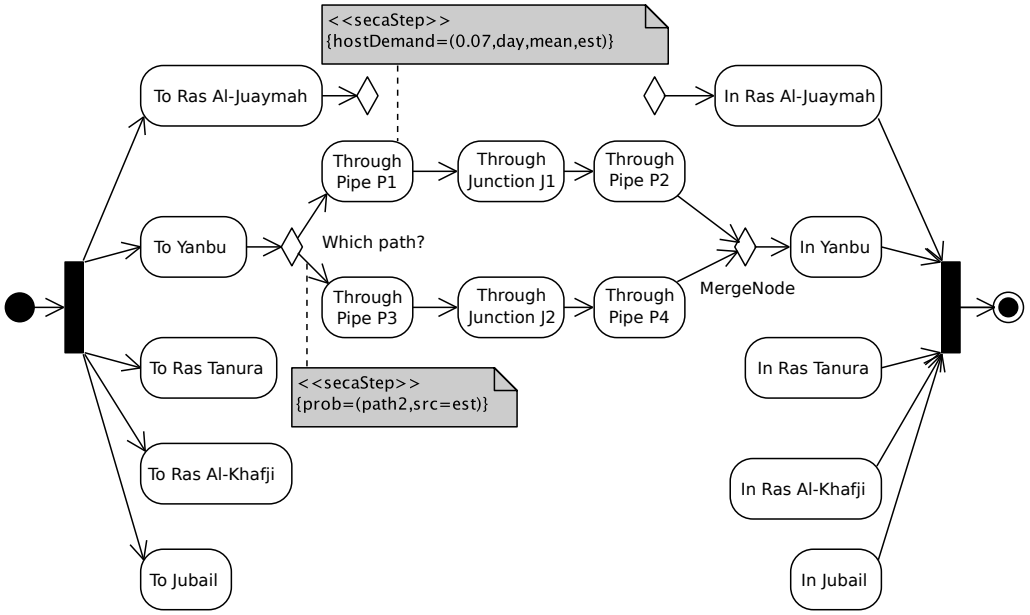


FIGURE 8. An excerpt of the Saudi Arabia crude-oil pipeline system flow.

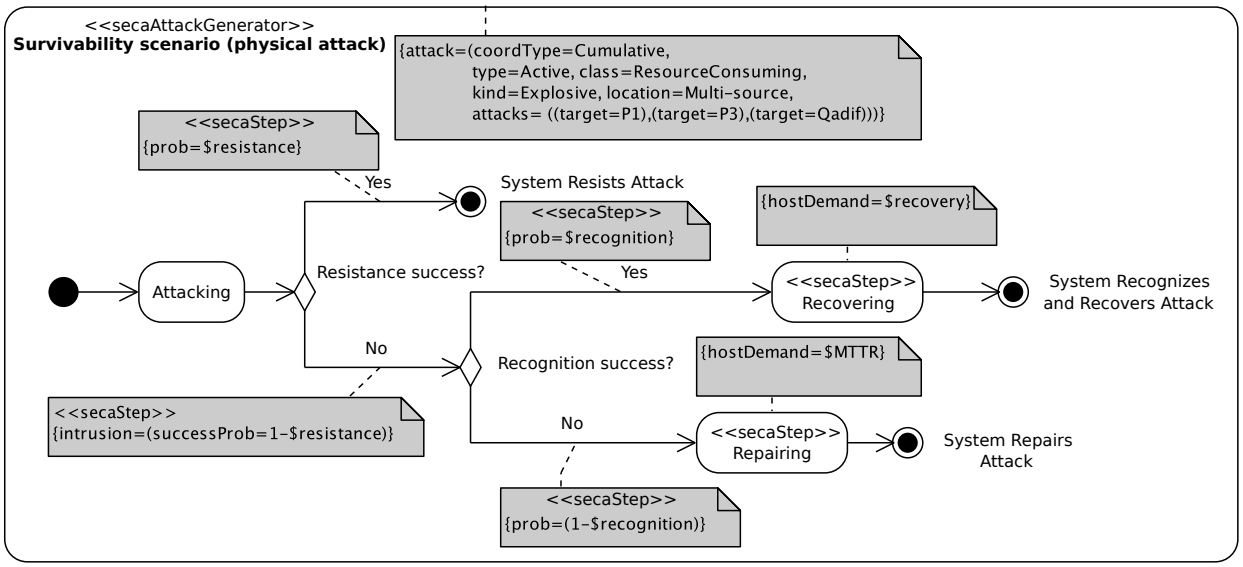


FIGURE 9. Survivability scenario of a physical attack.

- the oil flow injected in the network will reach eventually the port terminals.

It is worth to observe that such type of analysis also provides a feedback to the designer on the correctness of the UML models.

In all the experiments, the GSPN models have been solved with the GreatSPN [30] steady state simulator (confidence level 99% and accuracy 3%).

Figure 11(A1) sketches the GSPN model used in the analysis of the first survivability scenario, where the clouds represent the oil distribution network subnet and the survivability scenario under a physical attack assumption. The latter is detailed in Figure 11(A2) and it is obtained through UML-2-GSPN model transformation according to the approach in [28], where the three concurrent dotted subnets, one per each attacked resource (i.e., the pipes P1, P3 and Qadif junction), are obtained from the AD pattern in Figure 9.

Table 2 summarizes the parameters of the survivability scenario of Figure 9, the (range) values assigned to them in the sensitivity analysis and their mapping to the transition parameters of the GSPN subnet of Figure 11(A2). In particular, in this first study, we have assumed a perfect recognition (probability of recognition equal to one). The distribution network parameters have been set to the values used in the work [7].

The GSPN model sketched in Figure 11(A1) has

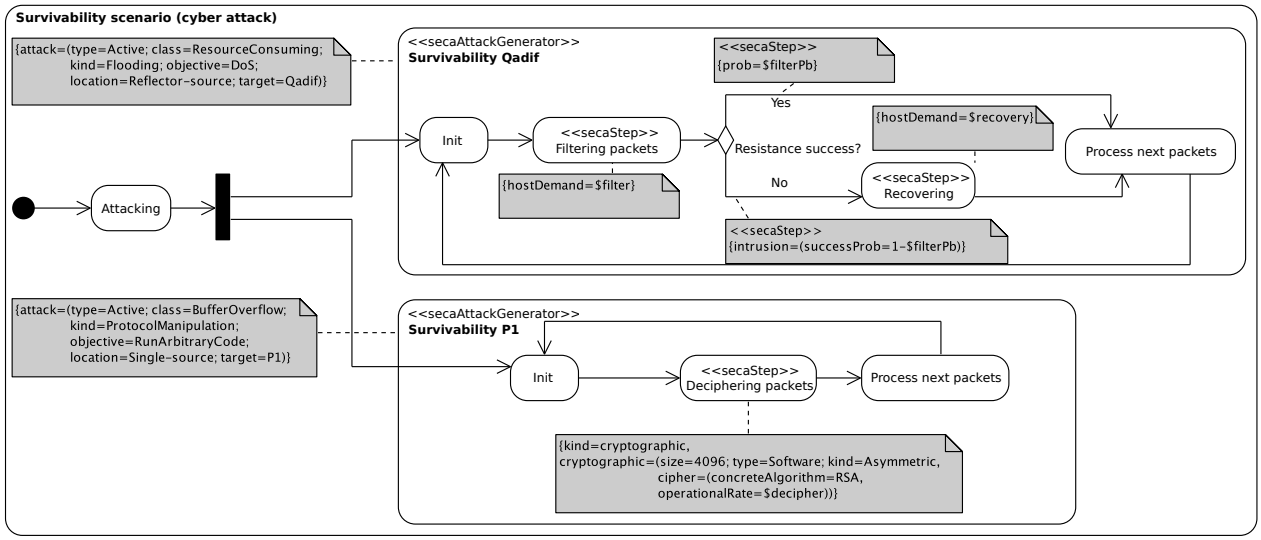


FIGURE 10. Survivability scenario of a cyber-attack.

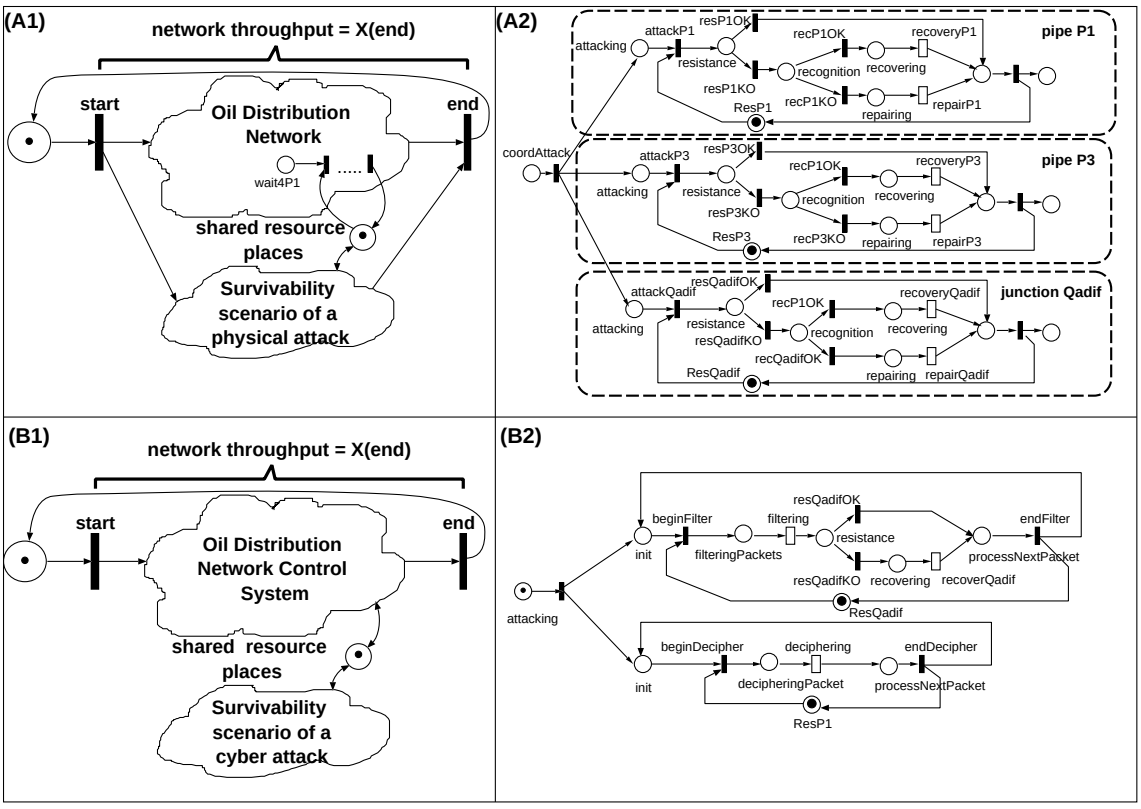


FIGURE 11. GSPN models of the case study.

been used to compute the network throughput (i.e., the throughput of transition `end`) and the mean number of tokens in places modelling the waiting for the resource availability (such as `wait4P1`).

Figure 12(A) shows the percentage of throughput loss due to the physical attack[4] and Figure 12(B) plots the number of oil barrels loss (Mbbl/day)[5]. For low values of the mean time to *recovery* (up to 6 hours) the throughput loss is below 20% (the number of barrels loss is below 240Mbbl/day) and the resistance solution does not affect sensitively the loss. As the (mean) time to recovery increases the effectiveness of the resistance

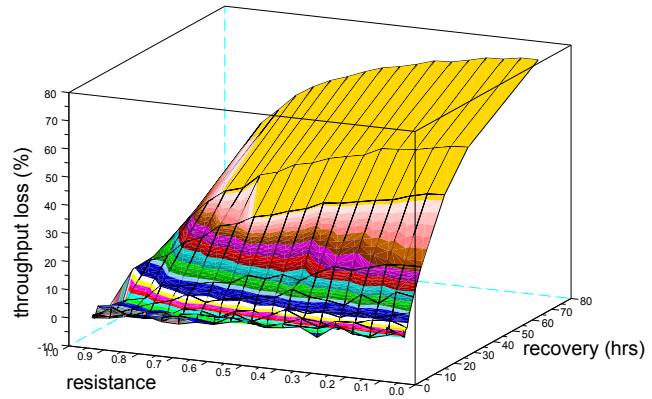
[4]The GSPN model of the distribution network without the attack subnet is characterized by a throughput of $X_{end} = 0.701146$.

[5]The number of oil barrels loss corresponds to the number of oil barrels blocked due to the unavailability of the attacked resources.

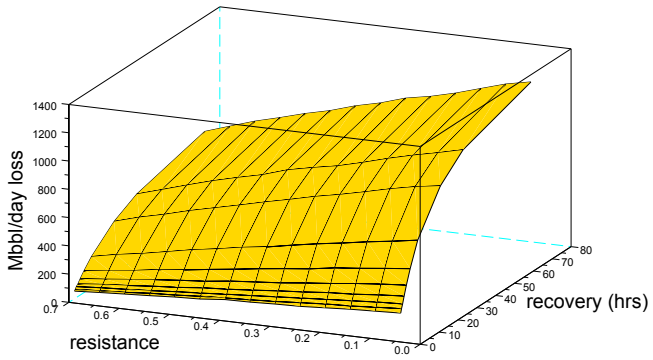
Parameters	Value(s)	GSPN transitions
resistance	[0.05-0.95]	recP1OK, recP3OK, recQadifOK
recognition	1	recP1OK, recP3OK, recQadifOK
recovery	[72-3] hrs	recoveryP1, recoveryP3, recoveryQadif
MTTR	6 months	repairP1, repairP2 repairQadif

TABLE 2. Sensitivity analysis parameters (survivability scenario of a physical attack).

solution becomes more important: if a low quality resistance solution (*resistance* $< 50\%$) is adopted the throughput loss raises up to 40% for a mean time to recovery of one day and up to 77% for a mean time to recovery of three days. The number of barrels loss increases to 900Mbbl/day and 1,212Mbbl/day, respectively. In such a case, to maintain a throughput loss below the 50% a resistance probability of at least 80% should be guaranteed. Such a requirement should be met using hard methods, e.g., surveillance combined with integrated perimeter security solutions (see, for example, [31]).



(A)



(B)

FIGURE 12. (A) Throughput loss (%) and (B) Mbbl/day loss due to the physical attack.

The survivability scenario of a cyber-attack concerns the oil distribution network control system and

Parameters	Value(s)	GSPN transitions
filterPb	[0.50;..;0.95]	resQadifOK
filter	[1.44min;..;14.4min]	filtering
decipher	2.88 min	deciphering
recovery	[11min-12hrs]	recoveryQadif

TABLE 3. Sensitivity analysis parameters (survivability scenario of a cyber attack).

Figure 11(B1) sketches the GSPN model used in the analysis. In this case, the targets of the attack are the software resources (i.e., the SCADA systems at network nodes) that control the flow of the oil through pipe $P1$ and the Qadif junction. The GSPN subnet of the survivability scenario is detailed in Figure 11(B2): similarly to the first study, the GSPN subnet has been obtained by UML-2-GSPN model transformation from the AD pattern in Figure 10 and Table 3 summarizes the scenario parameters. In this second study, we aim at analysing how the filtering & recovery solutions to be developed at the Qadif node may affect the system in terms of throughput overhead. In particular dependent filtering parameters are assumed, i.e., the longer the mean time required to filter the packets (*filter*) the higher is the probability of filtering DoS attacks (*filterPb*).

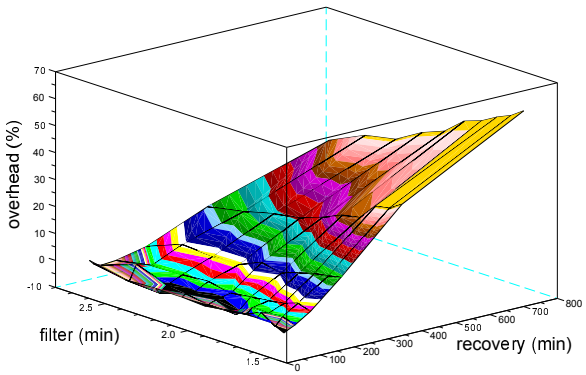


FIGURE 13. Throughput overhead (%) in the cyber-attack survivability scenario.

Figure 13 shows the trend of the throughput overhead versus the filtering and recovery (mean) times. The overhead can be maintained below 16% for recovery solutions that require a mean time of at most 3 hours: in such a case also filtering mechanisms with low quality (e.g., with a filtering probability of 50% and a mean time of about 2 minutes) can be applied. The quality of the filtering mechanism is decisive if the time necessary to recover the node is greater (from 6 to 12 hours). Indeed, the overhead raises up to 61% with low quality filters, while it can be halved by introducing filtering mechanisms which guarantee a filtering probability of $90 - 95\%$.

7. RELATED WORK

We have found very few specific profiles for specification and modelling security of critical infrastructures as SecAM does. Marrone et al. [32] model vulnerabilities using UML profiling and propose an analysis based on Bayesian networks. UML-CI is proposed in [33] as a profile for modelling critical infrastructures not focussed on security analysis but on management.

However, the literature regarding UML profiles for addressing security issues in general is large. In the following we revise a few of them, but a comparison of these approaches with SecAM is difficult since none of them focus on critical infrastructures, survivability analysis nor Petri nets.

Some profiles propose light-weight UML extensions for considering security into UML designs [34–36]. In [34] SecurityAssessmentUML profile for general model-based security assessments is introduced. UMLintr profile for specifying intrusions is described in [35], it aims at specifying intrusion scenarios in UML diagrams, however, this profile does not specify properties of distributed attacks and it is focused only on intrusion domain. A recent work in [36] proposes a UML profile for specification of Role-Based Access Control. Goudalo and Seret claim in [37] a UML extension for security as a real solution, and propose a set of stereotypes to deal with confidentiality into information systems.

Other UML profiles are focused on business processes or grid computing, such as [38–40]. In [38], Rodríguez et al. propose a UML profile that increases the expressive ability of activity diagrams by incorporating security requirements into the business process modelling. In [39], a UML profile is proposed for defining security requirements for Data Warehouses (DW) at the business level. In the mobile Grid context, GridUCSec-Profile is introduced in [40] and used in [41], where a methodology is proposed to analyse, design and construct a Secure Mobile Grid System.

SecureUML [42] allows to build secure distributed systems. In particular, it enables the specification of RBAC-based access control requirements together with several authorisation constraints. Besides, it supports code generation from the secure system model. UMLsec [43] allows to specify security relevant information during development of security-critical systems. It considers RBAC as access control policy, like SecureUML, and also provides tool-support for formal security verification. A recent work [44] combines UMLsec and MARTE profiles, allowing to address both security and timing properties together in a UML model.

Georg et al. propose in [45] an Aspect-Oriented Risk-Driven Development where UML sequence diagrams are transformed to Alloy language. Other works also propose new design framework methodologies to integrate security into the system's design [46–48].

In the security requirements engineering domain a recent survey can be found in [49]. It reviews and classifies works in literature concerning security during the development process, among them we highlight [50–52]. In [50] UML use cases (termed as *abuse cases*) are leveraged to capture security requirements and to perform an analysis which is not based on formal methods, unlike the analysis in our work. UMLsec is used in [51] to express security requirements and to analyse them through model-checking techniques. Finally, it is worth mentioning the work in [52] that introduces a methodology based on a goal-driven requirements engineering process, joining the Secure Tropos methodology and UMLsec and validating the models also through model-checking techniques.

In the survivability analysis context, a review of mathematical tools to achieve this purpose can be found in [53]. These tools include Markov models, trellis graphs and network models, among others. We have found in [54] a work that also uses stochastic Petri nets for survivability analysis, where a three-step survivability framework is proposed, aiming at survivability modelling and analysis for space systems.

8. CONCLUSIONS

The paper has demonstrated the capabilities of SecAM in a real assessment scenario of security. SecAM has proved proficiency to model security parameters and requirements of a real critical infrastructure, the Saudi Arabia oil-crude pipeline network. The attack scenarios modeled using SecAM have led to the obtention of a formal model, in terms of Petri nets, useful for survivability analysis.

The results of the analysis have been highly satisfactory. They have assessed consequences of real threats supporting results in previous reports [2]. At the same time, our assessment, due to the sensitivity analysis and the powerfulness of the formal methods to deeply characterize the infrastructure and the attacks, has offered more insight and more detailed and accurate results. Hence, we have been able to carefully evaluate the survivability strategies with respect to the throughput lost in the network, then assessing the quality of the resistance strategies as well as the impact of the loss.

SecAM has been built on the OMG standard MARTE and DAM profiles, which provide support to performance and timeliness analysis (MARTE) and dependability analysis as well (DAM). By inheritance, it is then possible to specify - through tagged-values - a wide range of non-functional requirements in UML behavioral models, then enabling trade-off analysis of different NFPs (e.g., availability versus performance or confidentiality versus availability). On the other hand, the use of SecAM does not hamper the application of other UML profiles proposed in the literature and their related transformation approaches aimed at evaluating

security and functional requirements, such as UMLSec.

Concerning the functional requirements analysis, the UML-2-GSPN transformation technique applied in the paper enables to get a Petri Net model that can be used for checking the correctness of the UML specification, by using either state-based techniques (e.g., model checking) or linear algebra/linear programming-based techniques (e.g., invariants computation, structural bound/liveness computation). As future work, we aim at combining `SecAM` with other formal methods, e.g., Fault Trees or Bayesian Networks as in [32], to be able to address different kinds of analysis.

Finally, tools are a need in this context. A plug-in for the Eclipse tool, not yet publicly released, has been build to annotate UML models with `SecAM`. The tool we have used for stochastic analysis, GreatSPN [30], has demonstrated skills to carry out the intricacies of the analysis.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish National Institute of Information Technologies (INTECO) accordingly to the rule 19 of the Digital Confidence Plan (Digital Agency of Spain) and the University of León under the contract X43, and by the project TIN2011-24932 of the Spanish Ministry of Economy and Competitivity.

REFERENCES

[1] Department of Homeland Security (2010) *National Security Strategy*. Available at `http://www.whitehouse.gov/sites/default/files/rss_viewer/national_security_strategy.pdf`.

[2] Brown, G. G., Carlyle, W. M., Salmerón, J., and Wood, K. (2005) Analyzing the Vulnerability of Critical Infrastructure to Attack and Planning Defenses. In Smith, J. C. (ed.), *Tutorials in Operations Research*, chapter 4, pp. 102–123. INFORMS.

[3] Ellison, R., Linger, R., Longstaff, T., and Mead, N. (1999) Survivable network system analysis: a case study. *IEEE Software*, **16**, 70–77.

[4] OMG (2011) *Unified Modelling Language: Superstructure*. Object Management Group. Version 2.4, formal/11-08-05.

[5] Selic, B. (2007) A Systematic Approach to Domain-Specific Language Design Using UML. *10th IEEE Int. Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, Santorini Island, Greece, May, pp. 2–9. IEEE Computer Society.

[6] Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1995) *Modelling with Generalized Stochastic Petri Nets* Wiley Series in Parallel Computing. John Wiley and Sons, Bognor Regis, West Sussex.

[7] Bernardi, S. and Campos, J. (2013) A Min-Max Problem for the Computation of the Cycle Time Lower Bound in Interval-Based Time Petri Nets. *IEEE T. Systems, Man, and Cybernetics: Systems*, **43**, 1167–1181.

[8] OMG (2011) *A UML profile for Modeling and Analysis of Real Time Embedded Systems (MARTE)*. Object Management Group. Document formal/11-06-02.

[9] Bernardi, S., Merseguer, J., and Petriu, D. (2013) *Model-driven Dependability Assessment of Software Systems*. Springer, Berlin.

[10] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004) Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, **1**, 11–33.

[11] Pfleeger, C. P. and Pfleeger, S. L. (2006) *Security in Computing*, 4th edition. Prentice Hall.

[12] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004) Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, **1**, 11–33.

[13] Rodríguez, R. J., Merseguer, J., and Bernardi, S. (2010) Modelling and Analysing Resilience as a Security Issue within UML. *Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems (SERENE)*, London, United Kingdom, April, pp. 42–51. ACM.

[14] Hansman, S. and Hunt, R. (2005) A taxonomy of network and computer attacks. *Computers & Security*, **24**, 31–43.

[15] Barnum, S. (2008). Common Attack Pattern Enumeration and Classification (CAPEC) Schema Description. The MITRE Corporation.

[16] Hussain, A., Heidemann, J., and Papadopoulos, C. (2003) A Framework for Classifying Denial of Service Attacks-extended. Technical Report ISI-TR-2003-569b. USC/Information Sciences Institute. (Original TR, February 2003, updated June 2003).

[17] Paxson, V. (2001) An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. *SIGCOMM Computer Communication Review*, **31**, 38–47.

[18] Braynov, S. (2003) On Future Avenues for Distributed Attacks. In Hutchinson, B. (ed.), *2nd European Conference on Information Warfare and Security (ECIW)*, University of Reading, United Kingdom, pp. 51–60.

[19] Menezes, A. J., Vanstone, S. A., and Oorschot, P. C. V. (1996) *Handbook of Applied Cryptography*, 1st edition. CRC Press, Inc., Boca Raton, FL, USA.

[20] Dworkin, M. (2001) Recommendation for Block Cipher Modes of Operation: Methods and Techniques. Technical report. The National Institute of Standards and Technology (NIST). Special Publication 800-38A.

[21] Cheswick, W. R., Bellovin, S. M., and Rubin, A. D. (2003) *Firewalls and Internet Security; Repelling the Wily Hacker*, second edition. Addison-Wesley, Reading, MA.

[22] Scarfone, K. and Mell, P. (2007) Guide to Intrusion Detection and Prevention Systems (IDPS). Technical report. The National Institute of Standards and Technology (NIST). Special Publication 800-94.

[23] Sousa, P., Bessani, A. N., Correia, M., Neves, N. F., and Veríssimo, P. (2010) Highly Available Intrusion-Tolerant Services with Proactive-Reactive Recovery. *IEEE Transactions on Parallel and Distributed Systems*, **21**, 452–465.

[24] The HoneyNet Project (2004) *Know Your Enemy: Learning about Security Threats*, 2nd edition. Addison Wesley Publishing, Boston, USA.

[25] Bertino, E. and Crampton, J. (2007) Security for Distributed Systems: Foundations of Access Control. In Qian, Y., Tipper, D., Krishnamurthy, P., and Joshi, J. (eds.), *Information Assurance: Survivability and Security in Networked Systems*, pp. 39–80. Morgan Kaufman.

[26] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996) Role-Based Access Control Models. *Computer*, **29**, 38–47.

[27] Office of the Deputy to the Under Secretary of Defense (Policy) for Policy Support (1993) *International Programs Security Handbook*. Federation of American Scientists.

[28] López-Grao, J. P., Merseguer, J., and Campos, J. (2004) From UML Activity Diagrams to Stochastic Petri nets: Application to Software Performance Engineering. *Proceedings of the Fourth International Workshop on Software and Performance (WOSP)*, Redwood Shores, California, USA, January, pp. 25–36. ACM.

[29] Chaney, E. and Berner, R. (2004) Global: oil price update: Still higher and more uncertain. `http://www.morganstanley.com/GEFdata/digests/20040507-fri.html`.

[30] GreatSPN (2002) `http://www.di.unito.it/~greatspn`. University of Torino.

[31] FiberSenSys Inc. Fiber SenSys security solutions. `http://fibersensys.com/security-solutions`.

[32] Marrone, S., Nardone, R., Tedesco, A., D'Amore, P., Vittorini, V., Setola, R., Cillis, F. D., and Mazzocca, N. (2013) Vulnerability modeling and analysis for critical infrastructure protection applications. *International Journal of Critical Infrastructure Protection*, **6**, 217 – 227.

[33] Bagheri, E. and Ghorbani, A. (2010) UML-CI: A reference model for profiling critical infrastructure systems. *Information Systems Frontiers*, **12**, 115–139.

[34] Houmb, S. H. and Hansen, K. K. (2003) Towards a UML profile for Security Assessment. *Proceedings of the Workshop on Critical Systems Development with UML (UML)*, San Francisco, CA, USA, October.

[35] Hussein, M. and Zulkernine, M. (2006) UMLintr: A UML Profile for Specifying Intrusions. *Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS)*, Potsdam, Germany, March, pp. 279–288. IEEE Computer Society.

[36] Cirit, c. and Buzluca, F. (2009) A UML Profile for Role-Based Access Control. *Proceedings of the 2nd international conference on Security of Information and Networks (SIN)*, Gazimagusa, North Cyprus, October, pp. 83–92. ACM.

[37] Goudalo, W. and Seret, D. (2008) Toward the Engineering of Security of Information Systems (ESIS): UML and the IS Confidentiality. *Proceedings of the 2d International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*, Cap Esterel, France, August, pp. 248–256.

[38] Rodríguez, A., Fernández-Medina, E., and Piattini, M. (2006) Security Requirement with a UML 2.0 Profile. *Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES)*, Vienna, Austria, April, pp. 670–677.

[39] Trujillo, J., Soler, E., Fernández-Medina, E., and Piattini, M. (2009) A UML 2.0 profile to define security requirements for Data Warehouses. *Computer Standards & Interfaces*, **31**, 969–983.

[40] Rosado, D. G., Fernández-Medina, E., López, J., and Piattini, M. (2010) Developing a Secure Mobile Grid System through a UML Extension. *J. UCS*, **16**, 2333–2352.

[41] Rosado, D. G., Fernández-Medina, E., López, J., and Piattini, M. (2010) Analysis of Secure Mobile Grid Systems: A systematic approach. *Information and Software Technology*, **52**, 517–536.

[42] Lodderstedt, T., Basin, D. A., and Doser, J. (2002) SecureUML: A UML-Based Modeling Language for Model-Driven Security. *Proceedings of the 5th International Conference on The Unified Modeling Language*, Dresden, Germany, September UML'02, pp. 426–441. Springer-Verlag.

[43] Jürjens, J. (2002) UMLsec: Extending UML for Secure Systems Development. *Proceedings of the 5th International Conference on The Unified Modeling Language*, Dresden, Germany, September UML'02, pp. 412–425. Springer-Verlag.

[44] Thapa, V., Song, E., and Kim, H. (2010) An Approach to Verifying Security and Timing Properties in UML Models. *15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, Oxford, March, pp. 193–202.

[45] Georg, G., Anastasakis, K., Bordbar, B., Houmb, S. H., Ray, I., and Toahchoodee, M. (2010) Verification and Trade-Off Analysis of Security Properties in UML System Models. *IEEE Transactions on Software Engineering*, **36**, 338–356.

[46] Mouratidis, H., Giorgini, P., and Manson, G. (2003) Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In Eder, J. and Missikoff, M. (eds.), *Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAiSE)*, Klagenfurt/Velden, Austria, June, Lecture Notes in Computer Science, **2681**, pp. 63–78. Springer Berlin / Heidelberg.

[47] Islam, S., Mouratidis, H., and Jürjens, J. (2011) A Framework to Support Alignment of Secure Software engineering with Legal Regulations. *Software and Systems Modeling (SoSym)*, **10**, 369–394.

[48] Khan, R. (2011) Secure software development: a prescriptive framework. *Computer Fraud & Security*, **2011**, 12–20.

[49] Elahi, G., Yu, E., Li, T., and Liu, L. (2011) Security Requirements Engineering in the Wild: A Survey of Common Practices. *Proceedings of the IEEE 35th Annual Computer Software and Applications Conference (COMPSAC)*, Munich, Germany, July, pp. 314–319.

[50] McDermott, J. and Fox, C. (1999) Using Abuse Case Models for Security Requirements Analysis. *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC)*, Phoenix, Arizona, USA, December, pp. 55–64.

[51] Jürjens, J. (2005) *Secure systems development with UML*. Springer.
[52] Mouratidis, H. and Jürjens, J. (2010) From Goal-Driven Security Requirements Engineering to Secure Design. *Int. J. Intell. Syst.*, **25**, 813–840.
[53] Westmark, V. (2004) A Definition for Information System Survivability. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Big Island, HI, USA, January 10 pp.
[54] Castet, J.-F. and Saleh, J. H. (2012) On the concept of survivability, with application to spacecraft and space-based networks. *Reliability Engineering & System Safety*, **99**, 123–138.