Using the AMAN-DA method to generate security requirements: a case study in the maritime domain

Amina Souag, Raúl Mazo, Camille Salinesi, Isabelle Comyn-Wattiau

Using the AMAN-DA method to generate security requirements: a case study in the maritime domain

Amina Souag[1], Raúl Mazo[1,2], Camille Salinesi[1], Isabelle Comyn-Wattiau[3]
[1] CRI, Université Panthéon Sorbonne, Paris, France
{amina.souag, raul.mazo, camille.salinesi}@univ-paris1.fr
[2] GiDITIC, Universidad Eafit, Medellin, Colombia
raulmazop@eafit.edu.co
[3] CEDRIC-CNAM & ESSEC Business School, Paris, France
isabelle.wattiau@cnam.fr

Abstract. [Context and motivation] Security requirements are known to be "the most difficult of requirements types", and potentially the ones causing the greatest risk if they are not correct. One approach to requirements elicitation is based on the reuse of explicit knowledge. AMAN-DA is a requirement elicitation method that reuses encapsulated knowledge in security and domain ontologies to produce security requirements specifications. **[Question/Problem]** The main research question addressed in this paper is to what extent is AMAN-DA able to generate domain specific security requirements? **[Principal idea/results]** Following a well-documented process, a case study related to the maritime domain was undertaken with the goal to demonstrate the utility and effectiveness of AMAN-DA for the elicitation and analysis of domain specific security requirements. The usefulness of the method was also evaluated with a group of twelve experts. **[Contribution]** The paper demonstrates the elicitation of domain specific security requirements by presenting the AMAN-DA method and its application. It describes the evaluation and reports some significant results and their implications for practice, and future research, especially for the field of knowledge reuse in requirements engineering.

Keywords: Security, requirements engineering, domain, ontologies, case study.

1. Introduction

At the heart of Information Systems (IS), security aspects play a vital role and are becoming a central issue in IS effective usage. With ever-growing digitization of activities in various sectors (communication, health, banking, insurance, etc.), IS are getting more and more complex. They must comply with emerging usages and varied needs, and are permanently exposed to new vulnerabilities. Not a single week goes by without an announcement indicating that the IS of some private or public organization was attacked. IS attacks may target strategic data such as information exchanged by CEOs, financial data, R&D documents, customers and human resources information, etc. The consequences for organizations are manifold: deterioration of the image and brand, disturbance of activity, financial losses, and even threats to the socio-economic, political and military ecosystems.

During the last decade, the research community started calling for early consideration of security, throughout the Requirements Engineering (RE) phase. Considering security during early stages of IS development allows IS developers to envisage threats and their consequences and countermeasures before a system is in place rather than following the destruction of a possibly disastrous attack [1].

At the same time, recent studies have shown that the lack of information security knowledge at the management level is one reason for inadequate or nonexistent information security management strategies and that raising management information security awareness and knowledge level leads to more effective strategies. Haley states that, among the challenges for security projects, there is the difficulty of expressing security requirements and producing exhaustive specifications [3]. In fact, most information systems and software developers are not primarily interested in (or knowledgeable about) security [4]. For decades, the focus has been on implementing as much functionality as possible before the deadline, and patching the inevitable bugs when it's time for the next release or hot fix [5]. However, the information systems and software engineering communities now realize that information security is also important for systems whose primary function is not related to security. Firesmith claims that most requirements engineers are poorly trained to elicit, analyze, and specify security requirements [6]. Consequently, they often confuse security requirements with architectural security mechanisms that are traditionally used to fulfill requirements, and end up making architecture and design decisions. Zuccato et al. [7]

report that security requirements engineering is in practice frequently performed by security non-experts and that security expertise is "scarce"; often security requirements and their dependencies are not directly known by requirements engineers. In addition to the lack of explicit security knowledge, there is the lack of domain knowledge, which was found to be another important element to consider during requirements engineering practices [8]. Authors affirm that RE must elicit and understand the requirements from the relevant stakeholders and perform the requirements specification together with them. Thus, in order to maximize environment comprehension, a common understanding of the involved concepts must be achieved. This means that requirements analysts should endeavor to work on understanding the language used in the universe of discourse, to then initiate the modelling of this universe. A model of the domain represents the reality and considerably improves its comprehension. Thus, a crucial part of RE is the establishment of a common terminology by diverse stakeholders.

Regarding the issues mentioned above (i.e. the lack of information security knowledge, deficiency of training for eliciting security requirements, lack of domain knowledge, absence of a domain model), some methods were proposed in the literature [8] [19]. However, they remain too general, in the sense that they are for general problem domains where problem-specific domain knowledge is not used and that they do not support explicitly the utilization of domain knowledge [9].

In order to address all these issues, AMAN-DA[1] was developed. AMAN-DA can be defined as a method that allows requirements engineers to **elicit and analyze domain specific security requirements of an information system by reusing knowledge modeled in domain and security ontologies**. The basic idea of AMAN-DA is to take as an input security goals, and security and domain ontologies in order to produce, as output, security requirements specifications. The goal is to guide and help the requirements engineer in the elicitation of security requirements for various projects in different domains. AMAN-DA is not necessarily meant to be used by developers, their role comes later in the information system process by coding into functions some of the elicited requirements. AMAN-DA is also not meant to be used by domain experts, although they have been often consulted during the project. AMAN-DA can be used by security experts, if they are interested by dealing with security at business level. Security experts can join the requirements engineers while using the method to validate some requirements, although lot of their expertise is formalized into the security ontology.

This paper presents the AMAN-DA method (though some primary versions of it are reported in prior early works on the project in [9] and [27]). The paper reports the evaluation of the method's utility and effectiveness using a case study related to the maritime domain followed by an experiment with experts to assess the method's usefulness. Following a well-documented process [13], the paper reports the case study design, the collection of data – i.e. the interviews performed with the maritime stakeholder in order to capture the security goals (using AMAN-DA's security goal model). The security goals being not enough to generate security requirements, two ontologies (security and domain ones) were used as a source of knowledge to discover threats, vulnerabilities, security requirements and their actors, organizational goals, and other domain specific concepts. Acknowledging the small number of publications tackling this issue and providing an evaluation of proposals on real cases [11], we felt the motivation and necessity to undertake this empirical research.

Throughout the paper, the term security covers the term safety as well. In some approaches, security focuses on preventing harm from malicious attacks as well as safety deals with preventing harm from accidents. The differences between security and safety are not remarkable; both are conditions where one is well protected. As far as AMAN-DA is concerned, the term security deals with accidental as well as deliberate harm. The basic idea is protecting assets from hazards/threats creating safe/secure conditions.

The paper is structured as follows: Section 2 presents the research methodology, Section 3 presents the method including its different parts and implementation. Section 4 presents the case study as well as the results of using AMAN-DA to elicit and analyze security requirements for systems in the maritime domain, which show the efficacy and potential of the method. It reports some threats to validity. Section 5 reports the evaluation and discussions with experts. Section 6, threats to validity. Section 7 presents the lessons learned, Section 8 the related

[1] *AMAN* (أمان) is the Arabic word for security. DA is for Domain of Application. The name was chosen to refer to security requirements engineering for domains of application.

work, and Section 9 close this paper with some conclusions obtained from the study and some directions for future research in the topic of security requirements generation.

2. Research methodology

The current paper is the summary of a research project that aimed to investigate the question of elicitation of security requirements that are domain-specific with a reuse strategy. To address the main research question and to test the research hypotheses, a research strategy based on the design science process model proposed by Peffers et al. [45] was implemented. This process contains six main steps: (identify problem and motivate, define objectives of a solution, design and development, demonstration, evaluation, communication). Figure 1 presents the design science process model for information system research, and the application of this process to the research carried out.

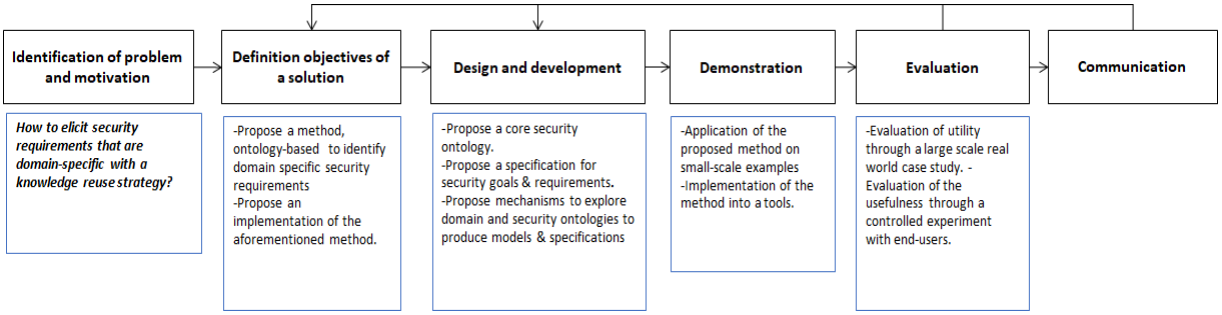


Figure 1 Application of the design science process model for information system research (Peffers et al. 2007) to the research carried out in the project.

The step of the identification of the problem and its motivation was carried out through a detailed and systematic state of the art of previous research and current practices. The results were published by the authors in [11] and [46].

The steps of definition of a solution, design and development, and demonstration are reported in section 3 of this paper, where the AMAN-DA method, its different parts, and its tool are presented as well as its application with a small example. The core security ontology (a part of AMAN-DA) was published by the authors in [12].

The evaluation step is reported in section 4, where the case study is detailed to evaluate the method's utility and effectiveness. This case study was carried out following a well-referenced process proposed by Runeson et al. [13].

As for the communication step, this paper is the first publication of the work in an international journal with the goal of communicating the results to the academic and professional communities.

The reader may note that the chosen research methodology proposed by Peffers et al. [45] and the chosen case study process by Runeson et al. [13] are coherent with previous work on building theories from case studies, in particular the one by Eisenhardt [47] from which most steps (selecting cases, instrument and protocols, analysis within case data, enfolding literature, etc.) were taken into consideration.

3. The AMAN-DA method

AMAN-DA is a security requirements elicitation and analysis method. It was mainly developed to guide requirements engineers (who have vague and tacit knowledge in security) during early requirements elicitation phase. AMAN-DA takes as an input: (i) security goals expressed by stakeholders, such as the ones that are captured during interviews (middle-top Figure 2), (ii) and two kinds of ontologies: a security ontology that embeds security specific knowledge (left Figure 2, Figure 4, Figure 17), and a domain ontology that encompasses domain specific knowledge (right Figure 2). The output of AMAN-DA is a specification of security requirements formalized (i) with Secure Tropos [10] [29] models (middle Figure 2) and (ii) textually into a specification

document (middle bottom Figure 2). The resulting Secure Tropos model covers the three views (organizational, security requirements, and attack views) with their respective concepts and relations. Whereas, the textual specification contains the assets to protect, the potential threats, vulnerabilities, and the security requirements to consider (this can be visualized later in Figure 13).

To achieve that, AMAN-DA propose a special syntactic pattern for security goals and another one for security requirements (section 3.2). In addition, the method relies on a collection of heuristic rules (section 3.3) that extract relevant security and domain knowledge from ontologies (the arrows in Figure 2 from the domain and security ontologies to the Secure Tropos model [10] [29] and security goals).

The expected outcome from using security and domain ontologies is that the security requirements resulting from the combined use of both ontologies will be more specific to the domain at hand.

Secure Tropos [10] [29] is a security-aware software system development methodology, which combines requirements engineering concepts, such as actors, goals, security constraints together with security engineering concepts such as threats, security constraints and security mechanism, under a unified process to support the analysis and development of secure and trustworthy software systems. Most of the Secure Tropos' concepts used in AMAN-DA are shown in Figure 2 (goal, (s) goal, security constraint, security mechanism, security objective, threat). The reader may refer to the tutorial in [42] to know about all the concepts and relations used in Secure Tropos.

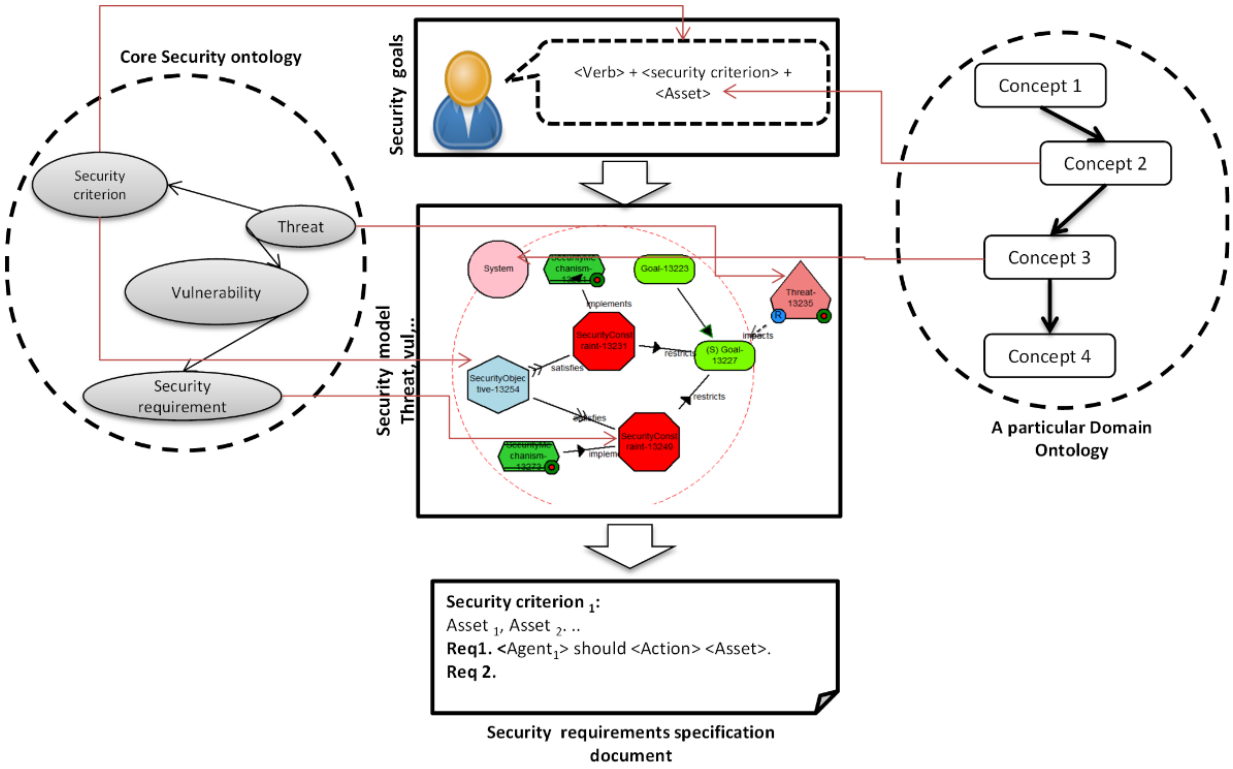


Figure 2. Overview of AMAN-DA

The choice of Secure Tropos was motivated by the fact that it is one of the richer modeling frameworks in terms of concepts that are used to model security requirements according to a recent systematic mapping study [11] on the subject. However, one needs to be "*knowledgeable*" about security and security requirements when using Secure Tropos, which is not always the case as reported in the introduction. This is where the role of AMAN-DA appears. AMAN-DA relies on explicit security and domain *knowledge* modeled in ontologies to produce Secure Tropos models.

The originality of the method lies on the fact that (a) the combination of the security ontology and domain ontologies is not achieved a priori, but at runtime, while the method is applied. (b) The method is generic in the sense that it is designed to be used with a generic security ontology [12] and any domain ontologies; as long as they embed, some expected knowledge. (c) The defined rules allow the method to automatically exhibit an appropriate ontological semantics (security and domain specific) to the engineer in charge of requirements elicitation (agents, objects, threats, security requirements, etc.). (d) The Secure Tropos models (organizational, attacks and requirements views) and textual security requirements are generated automatically by the tool that implements the method (Section 3.5).

AMAN-DA does not propose new process for security requirements elicitation, since many processes exist in the literature. We resumed the process presented by Mayer [1], where first the assets and goals are identified. Then, the threats and vulnerabilities are analyzed. Finally, security requirements are elicited. Nevertheless, section 3.4 presents guidelines to use the AMAN-DA method.

The next sub-sections explain with more details each brick of the method, and guidelines to use those bricks together in AMAN-DA.

3.1. Ontologies in AMAN-DA

For the domain part, the first goal was to allow AMAN-DA to be independent of any pre-selected domain ontology – i.e. to allow the method to be applied to any domain and to be restricted to one single domain, the second goal was to provide an extensible ontology well structured, easy to understand, and to extend. To achieve those goals, a multi-level domain ontology was developed. The "**Multi-level Domain Ontology**", whose main concepts are represented in Figure 3, relies on previous studies on domain ontologies [40] [41]. The ontology was designed to be easily extensible by sub-ontologies, depending on the domain at hand.

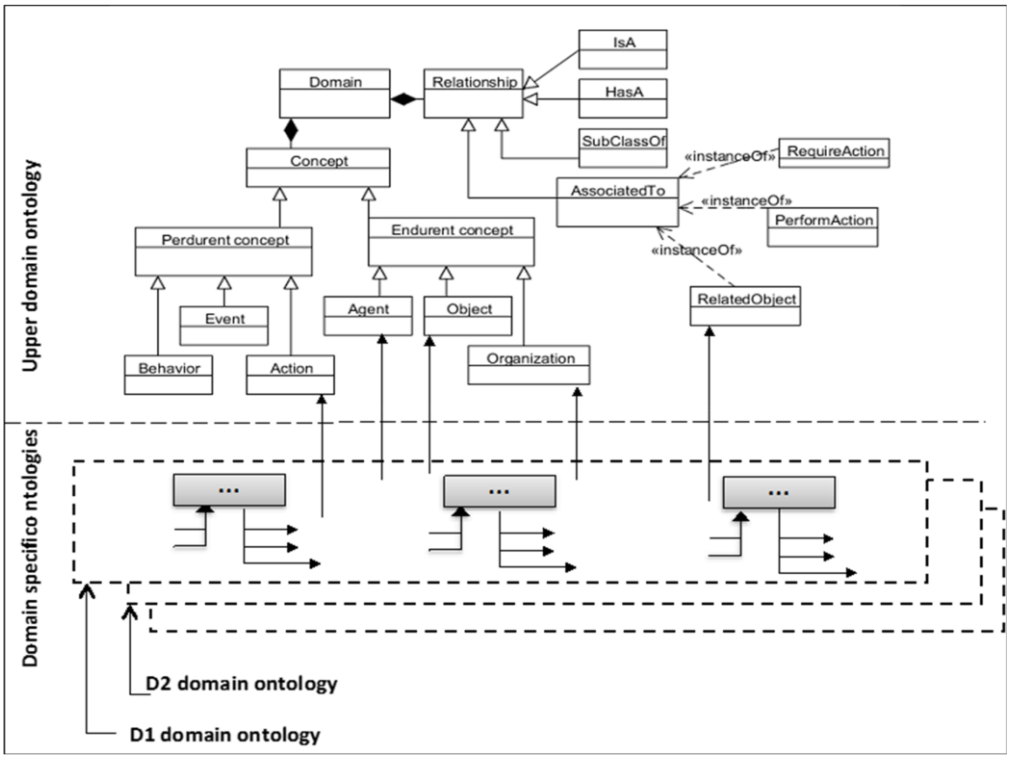


Figure 3. Multi-level domain ontology

The Multi-level domain ontology is represented as follows:

Upper-domain ontology (upper level): The upper domain ontology represents a domain according to two main components: (a) concepts to which a domain refers (such as vessel, car, person, patient, etc.) and (b) relationships between those concepts (HasLength, HasLocation, etc.). All domains share common concepts represented by this upper view. Concepts can be perdurants or endurants. Endurant concepts are those entities that can be observed – perceived as a full, at no matter which given snapshot of time. Examples include agents and material objects (such as an apple). Perdurants are entities that unfold themselves over time in successive temporal parts or phases. Perdurants include behaviors (vessel voyage, visiting ports, etc.), events (e.g. new employee arrival) and actions (e.g. remove container from vessel, develop web page).

Relationship between those concepts include: Is-A relationships (generalization/specialization, HasA (aggregation) (e.g. haveParents), SubClassOf (e.g. a registered customer is a SubClass of web customer), AssociatedTo (e.g. the class 'Web developer' is associated with the relationship PerformAction to the class 'Develop web application').

Specific domain ontology (lower level): contains the specific information to the application domain. The details of basic domain information (concepts and relations) represented in the upper ontology are clearly defined at this lower level which is specific to a particular domain and can vary from a domain to another. For example, the concept 'agent' which is an endurant concept at the upper level can be specialized into 'customer' in a business domain ontology, the same concept can be 'patient' in a health domain ontology or a 'teacher' in an education-school domain ontology.

Based on this multi-level domain ontology, four domain ontologies were developed (instantiated): maritime (Figure 18), online shopping, web publishing, sales and experienced with AMAN-DA. It is possible to use AMAN-DA with another domain ontology as soon as this ontology is developed instantiating the upper-domain ontology. This article report application and case study of the maritime domain ontology (section 4).

For the security part, AMAN-DA uses a **core security ontology** that considers the descriptions of the most important concepts related to security requirements and the relationships among them. "Core" refers to the union of knowledge (high-level concepts, relationships, attributes) present in other security ontologies proposed in the literature. It contains number of potential threats, vulnerabilities, and security requirements. The concepts were grouped into three main dimensions (organization, risk, treatment). This security ontology was developed within the AMAN-DA project, it has been evaluated by checking its validity and completeness compared to other security ontologies. A controlled experiment with end-users was performed to evaluate its usability. The paper [12] presents in detail the construction of the ontology, its concepts and relations and reports its evaluation.

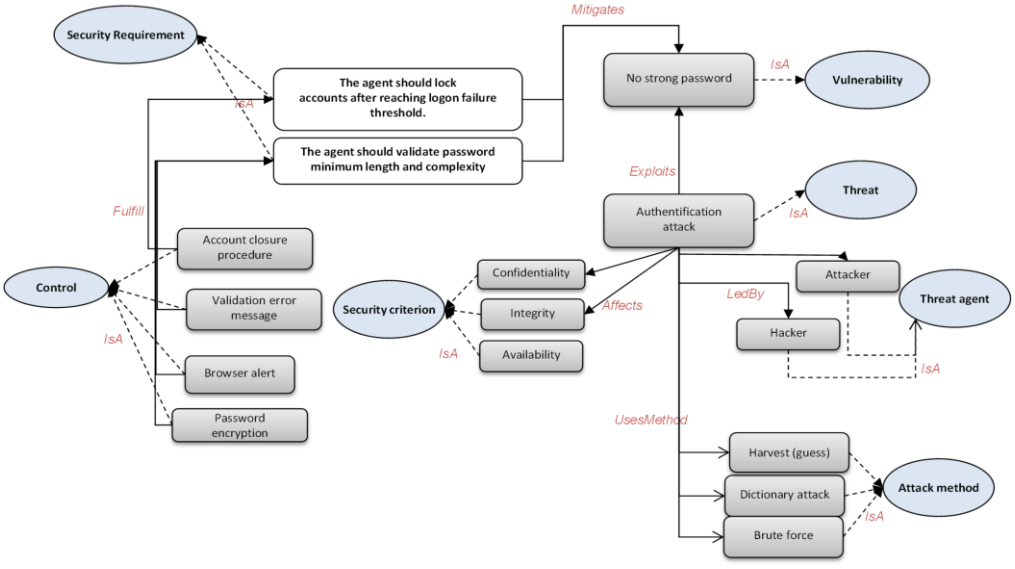


Figure 4. A deep view on the security ontology

Figure 17 gives an overview of the core security ontology, while Figure 4 presents a deep view of the ontology with some instances.

3.2. Specifying security goals and requirements in AMAN-DA

Stakeholders can express security concerns at different levels of detail. AMAN-DA distinguishes between security goals (abstract) and security requirements (more detailed). Security goals are inputs of the AMAN-DA method; security requirements are part of the security ontology (Figure 4) and are outputs of the method (Figure 2). Two syntactic patterns were proposed to specify security goals and security requirements.

AMAN-DA suggests that security goals can be expressed as a clause with (i) a main verb, (ii) one or many security criteria (e.g., confidentiality, integrity, availability) and (iii) one or many target assets that need to be protected. Figure 5 presents an UML class diagram corresponding to the structure of a security goal required as an input by AMAN-DA.

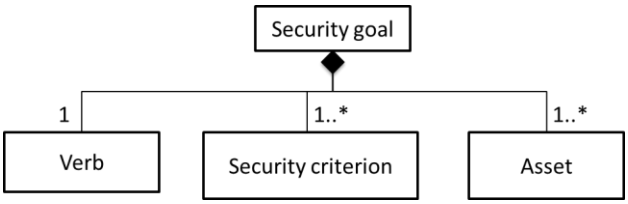


Figure 5. Security goal model in AMAN-DA.

For example, the customers of a bank may have the goal that their financial situation remains confidential. This can be formalized as "(Maintain) $_{\text{Verb}}$ (the confidentiality) $_{\text{Security criterion}}$ of (financial situation) $_{\text{Asset}}$"

The authors of scientific articles submitted to a given journal, may ask to maintain the integrity of the submitted articles, which can be formalized as (Keep)$_{\text{Verb}}$ (the integrity) $_{\text{Security criterion}}$ of (articles) $_{\text{Asset}}$

Security requirements materialize security goals. Based on the work of Rupp et al. [21], AMAN-DA proposes the syntactic pattern for documenting security requirements presented in Figure 6.

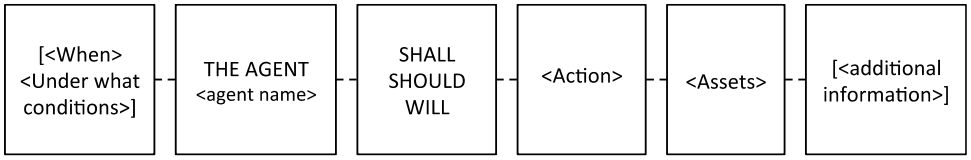


Figure 6. Syntactic pattern for documenting security requirements in AMAN-DA.

In the context of AMAN-DA this pattern should be used to specify security requirements and is composed of the following elements:

- <When> <*Under what condition*>: This element defines the temporal or the logical conditions under which the function documented in the requirement shall be performed.

- <*Agent name*>: This element defines the name of the agent, which shall provide the documented function. As security requirements are defined depending on what we want to protect and on the target security level, they can be related to databases, applications, systems, organizations, and external environments. That is why the agent who performs the requirement can be a system (e.g. "The system should lock accounts"). It can also be any agent of the domain. (e.g. "The chief engineer should provide fire extinguishers"). This agent is the grammatical subject of the sentence.
- "*Shall/Should/Will"*: Same as in the pattern of Rupp [21], these elements indicate the importance of the requirement. They are consistent with the overarching objective of clear and effective communication.

- <*Action*>: Actions constitute the activity or set of activities that processes the requirement (e.g. provide, lock, etc.).
- <*Assets*>: Assets are the objects used for, or are part of the actions (e.g. fire extinguishers, accounts).

Hence, the security goal expressed by authors as:

(Keep)$_{\text{Verb}}$ (the integrity) $_{\text{Security criterion}}$ of (articles) $_{\text{Asset}}$

Can be materialized by the security requirement:

(The web publishing system) $_{\text{agent}}$ (should) (lock) $_{\text{action}}$ (accounts) $_{\text{assets}}$ (after reaching logon failure threshold) $_{\text{under what condition}}$

The next section explains the passage from the security goal to the security requirement and the generation of the specification.

3.3. Security requirements elicitation rules

To build the bridge between the security ontology, the security goals, domain ontologies, and produce Secure Tropos models, a set of mapping and production rules are described under the form Concept $_{\text{source}}$ → Concept $_{\text{target}}$. A given concept in the source is mapped to another concept in the target, where source and target are either the security ontology, the domain ontology, the security goal, or a Secure Tropos concept.

For example, the rule: "Agent $_{\text{Domain Ontology}}$ → Actor $_{\text{SecTro}}$" means that the concept *Agent* in the domain ontology is mapped to the concept *Actor* in the Secure Tropos model.

The rule "Security criterion $_{\text{security goal}}$ → Security Objective $_{\text{SecTro}}$" means that the concept *Security* criterion in the security goal is mapped to the concept *Security objective* in the Secure Tropos model.

Production rules are described under the form <S→C>, where S is a situation and C a conclusion. <S→C> means that if the situation S is meant, then the conclusion C can be drawn. The situation holds on the input security goals, the input security ontology, and the input domain ontology.

The situation is defined using a first order logic predicate that relies on two kinds of functions `EquivalentDomainConcept` and `OntologyLink`:

- `EquivalentDomainConcept`(Concept $C_{generic}$, Type-Concept, Ontology DomainOnt): where Concept $C_{generic}$ is a generic concept, Type-Concept denotes its type, DomainOnt is the domain ontology at hand. The function will return the concept(s) in the domain ontology that has(ve) the closest semantics to the concept $C_{generic}$ in the security ontology or the security goal.

- `OntologyLink(Type, X, Y)`: is true if, in the input ontology, there is a link from concept X to concept Y that has the type <Type>. For instance, `OntologyLink (IsAffectedBy, X, Y)` is true if, in the input ontology, X and Y are related by an "affects" link from Y to X.

Conclusions indicate elements that should be added to the output Secure Tropos model. There are two conclusions-functions that are used for creation of new concepts and relations in the Secure Tropos model …:

- `CreateConcept (ClassC, X)`: indicates that a concept X that instantiates the <ClassC> class should be created in the model.

- `CreateLink(LinkTypeL, X, Y)`: indicates that a link from X to Y, of type <LinkTypeL>, should be created in the model.

Continuing with the articles submission example, the function `EquivalentDomainConcept` can be used to make the security goal more domain specific using a given domain ontology related to the domain at hand:

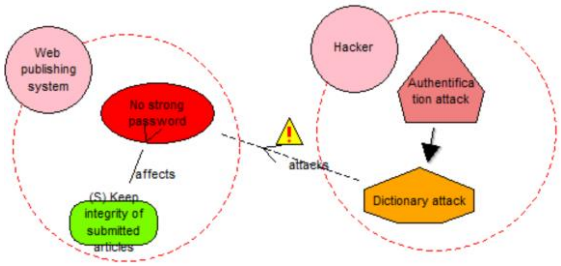
(Keep)$_{\text{Verb}}$ (the integrity) $_{\text{Security criterion}}$ of (articles) $_{\text{Asset}}$

Becomes:

(Keep)$_{\text{Verb}}$ (the integrity) $_{\text{Security criterion}}$ of (submitted articles) $_{\text{Asset}}$

Table 1 illustrates the use of the defined rules to produce a possible part of Secure Tropos model (the attack view in this case) using the security ontology.

Table 1. Threat analysis using the rules

Mapping	Production	Model (a possible part)
Security criterion $_{\text{security goal}}$ → Security criterion $_{\text{Security ontology}}$ Threat $_{\text{Security ontology}}$ → Threat $_{\text{SecTro}}$ Threat agent $_{\text{Security ontology}}$ → Malicious Actor $_{\text{SecTro}}$ Agent $_{\text{domain ontology}}$ → Malicious Actor $_{\text{SecTro}}$ Attack method $_{\text{security ontology}}$ → Attack method $_{\text{SecTro}}$ Vulnerability $_{\text{security ontology}}$ → Vulnerability $_{\text{SecTro}}$	OntologyLink (*Affect*, authentication Attack, Integrity) CreateConcept (Threat, authentication attack) CreateLink (*Impacts*, authentication attack, Keep integrity of submitted articles) OntologyLink (*LedBy*, authentication attack, Hacker). CreateConcept (Malicious Actor, Hacker) OntologyLink (*UseMethod*, authentification attack, dictionary attack). CreateConcept (Attack method, dictionary attack) CreateLink (*Embedded*, authentication attack, dictionary attack) OntologyLink (*Exploits*, authentication attack, no strong password) CreateConcept (Vulnerability, no strong password) CreateLink(*Attacks*, dictionary attack, no strong password) CreateLink (*Affects*, no strong password, (S) Keep integrity submitted articles)	

The vulnerability "no strong password" can be mitigated by the security requirement "The agent should lock accounts after reaching logon failure threshold".

Here too, the function `EquivalentDomainConcept` can be used to make the security requirement more domain specific, which gives:

(The web publishing system) $_{\text{agent}}$ (should) (lock) $_{\text{action}}$ (editor, reviewer and author, accounts) $_{\text{assets}}$ (reaching logon failure threshold) $_{\text{under what condition}}$

Then the functions: `OntologyLink`, `CreateConcept` and `CreateLink` can be used to create the rest of the model elements.

Using an input maritime domain ontology (Figure 18), the core security ontology (presented in [12], Figures 4 and 17) and the security goal and requirements specification models in addition to the rules presented above, a case study related to a maritime domain was undertaken in order to assess the efficacy of AMAN-DA for generating security requirements.

3.4. Security requirements elicitation guidelines with AMAN-DA

We propose the following twelve steps for eliciting security requirements with AMAN-DA.

A. Goal and asset analysis

1. Look for the organization in the domain ontology.
   E.g. The organization in the publishing domain ontology is 'publishing company'.
2. Create the organization concept 'publishing company' in Secure Tropos model's organizational view. Use the function (`CreateConcept`) .
3. Find out from the domain ontology who are the agents (*has_agent*) of the organization, and their actions (*perform_action*). Use the function (`OntologyLink`)
   E.g. 'Web publishing system' *perfom _action* 'manage articles'.
4. Find out from the domain ontology who are the agents who (*require_action*). Use the function (`OntologyLink`)
   E.g. 'Editor' *require_action* 'manage_articles'
5. Create the concepts Actor and Goals in the Secure Tropos model's organizational view. An agent is mapped to an actor, an action is mapped to an organizational goal accordingly. The actor (A1) depends to another actor (A2) to realize the organizational goal (G1) if, in the domain ontology, the agent (A2) performs an action that the agent (A1) requires. Use the functions (`CreateConcept`).
6. Create the relative dependency relations between the actors and the organizational goals in the Secure Tropos model's organizational view. Use the function (`CreateLink`) .
   E.g. In Secure Tropos model, the new created actor 'Editor' depends on the created actor 'The web publishing system' to fulfil the organizational goal 'manage_articles'
7. While interviewing the stakeholder, identify the security goals that correspond to a given organizational goal and express them respecting the security goal model (Figure 5) as a clause with a verb, a security criterion, and an asset.
   E.g. 'Keep integrity of article' is a security goal that corresponds to the organizational goal 'manage articles"
8. Use the domain ontology to improve the security goal identified, i.e. map the asset of the security goal to the objects of the domain ontology and find which concept is the closest semantically. Use the function (`EquivalentDomainConcept`). Replace it.
   E.g. 'Keep integrity of articles' becomes 'keep integrity of submitted article".
9. Create in the Secure Tropos model the new security goal and create the link between it and the organizational goal. Use the functions (`CreateConcept,CreateLink`).
   E.g. Create the security goal '(S) Keep integrity of submitted article'. Create a link between the organizational goal 'manage_articles'

B. Threat analysis

10. Use the security ontology to find the threats that *affect* the security criterion expressed in the security goal. Then, create this threat in Secure Tropos model. Create the link *affect* between this threat and the security goal. Use the functions (`OntologyLink,CreateConcept,CreateLink`).
    E.g. The threat 'authentication attack' is a potential threat to the criterion 'integrity'. Create in the Secure Tropos model the threat 'identification attack'. Create the link *affect* between the threat 'authentication attack' and the security goal '(S) keep integrity of submitted article'.
11. By the same reasoning, use the security ontology to find the potential threat agents, attack methods, and vulnerabilities. For each new concept found in the security ontology, create a new corresponding concept in the Secure Tropos model and link it to the existing concepts with the right relation. Use the functions (`OntologyLink,CreateConcept,CreateLink`).

C. Security requirements analysis and elicitation

12. By the same reasoning as well, find out in the core security ontology which security requirement mitigates a given vulnerability. Use the function (`OntologyLink`)
    E.g. The vulnerability 'no strong password' can be *mitigated* by the security requirement 'the agent should lock accounts after reaching logon failure threshold'.

13. Use the domain ontology now to improve the semantic of the security requirement that mitigates the given vulnerability. The asset clause of the security requirement can be replaced by the semantically closest domain objects. The agent clause is replaced by the agent initially responsible for the organizational goal under analysis. It can be replaced by another agent from the domain ontology if the stakeholder so chooses. Use the function (`EquivalentDomainConcept`)

    E.g. The security requirement (The web publishing system) $_{agent}$ (should) (lock) $_{action}$ (accounts) $_{assets}$ (after reaching logon failure threshold) becomes (The web publishing system) $_{agent}$ (should) (lock) $_{action}$ (editor, reviewer and author, accounts) $_{assets}$ (reaching logon failure threshold) $_{under\ what\ condition}$
14. The security requirement taken from the security ontology, and improved semantically with the domain ontology, is modeled by a security constraint in Secure Tropos model. Use the functions (`CreateConcept, CreateLink`)
15. By the same reasoning, the security ontology can be used to find control that fulfil a security requirement. A control can be modeled to a security mechanism in Secure Tropos model. Use the functions (`OntologyLink`, `CreateConcept, CreateLink`)
16. Finally, generate the textual specification from the produced Secure Tropos model.
17. Move on to another organizational goal and repeat steps from 7 to 16.

The following gives an implementation of AMAN-DA and a detailed evaluation of the method through the case study.

3.5. Implementation of the AMAN-DA method

AMAN-DA was automated via a tool that the requirements engineer or the security analyst can use during the security requirements analysis and elicitation process. The tool was implemented on Java Eclipse. The technical architecture of the tool is organized around five main levels (user, presentation, application, API, knowledge) (Figure 7).

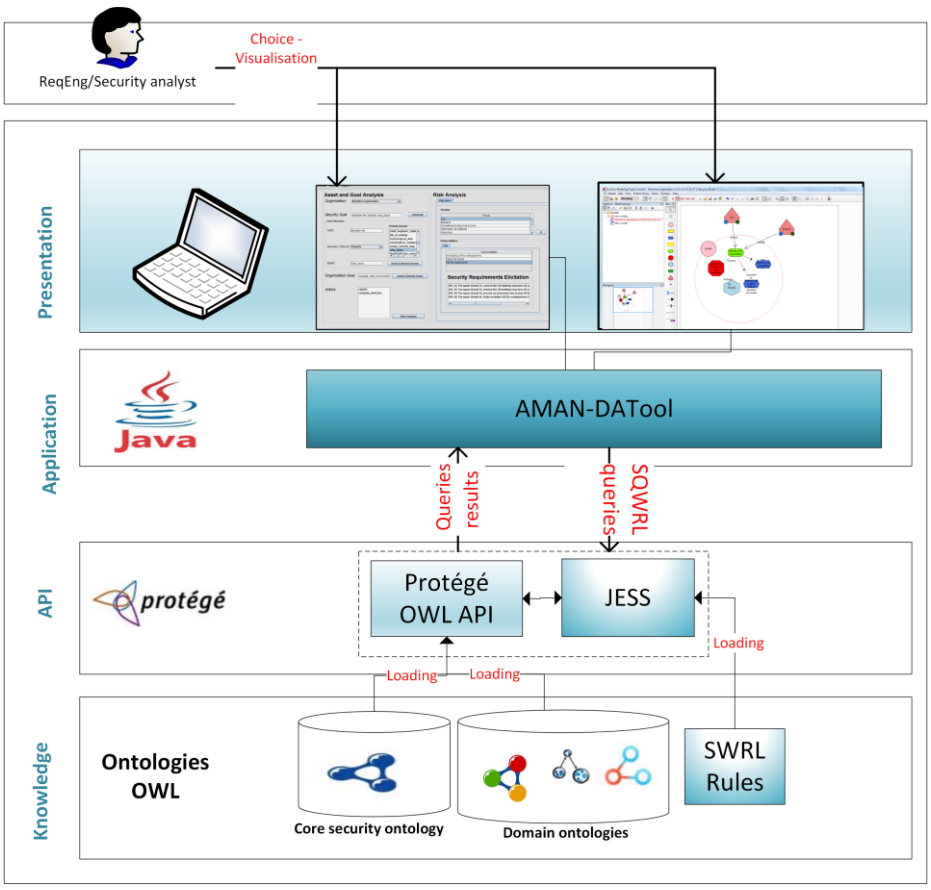


Figure 7. Technical architecture of the tool

The user (requirements engineer or security analyst) use the tool AMAN-DATool through its presentation windows. He/she performs his/her security requirements analysis starting by introducing the security goals (choosing the assets to protect and the security criteria), analyzing the potential risk (choosing the potential threats, attackers, attack methods, and vulnerabilities); finally, generating the adequate domain specific security requirements. The different choices of the user are translated into SQWRL (Semantic Query-Enhanced Web Rule Language) queries that the tool generates dynamically and automatically. These queries are intended to the ontologies stored in the knowledge layer. The interaction between the presentation and application layers and the knowledge layer (ontologies) is ensured thanks to the APIs and the Jess engine[2]. At the end of the analysis, AMAN-DATool offers the generation of the specification document in addition to the Secure Tropos model that can be visualized with SecTro tool (Figure 8).

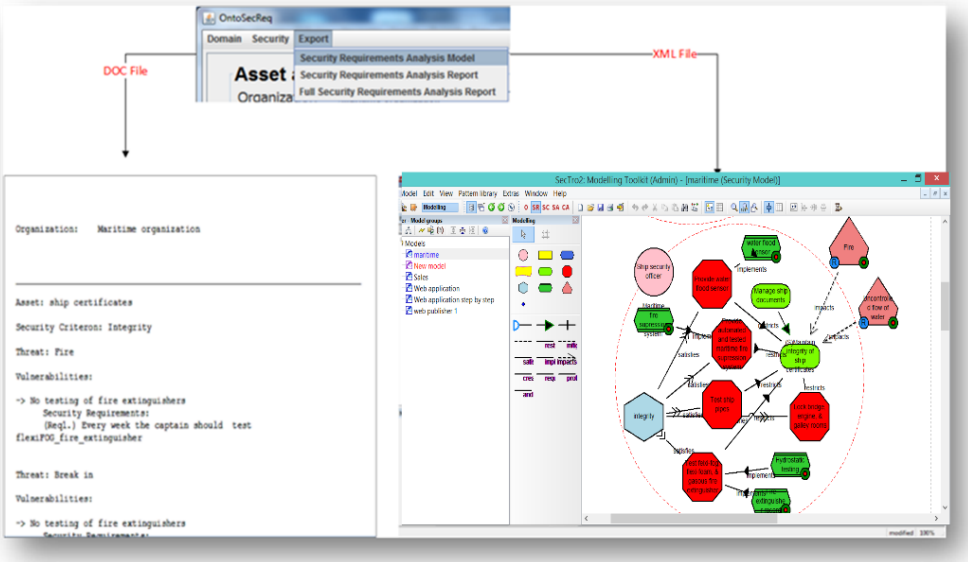


Figure 8. Generation of security models or textual specifications

A detailed demonstration of the AMAN-DAtool can be visualized in the link:
https://www.youtube.com/watch?v=czVGj6ct1i8

4. Case study

A case study to assess the utility and effectiveness of AMAN-DA was performed by following the guidelines presented by Runeson et al. [13] since we found it well documented. Conducting research on real-world phenomena implies a constant trade-off between level of control and degree of realism. The realistic situation is often complex and nondeterministic, which hinders the understanding of what is happening [13]. Relying on a well-defined research process can help applying a practical research strategy for a specific research situation.

Prat et al. [25] propose a taxonomy of evaluation methods for information system. Since we wanted to know whether AMAN-DA is able to produce domain specific security requirements, the focus was on two criteria from

[2] http://www.jessrules.com/

the taxonomy: effectiveness and utility. Effectiveness is defined as the degree to which the artifact achieves its goal in a real situation [34] [35]. Utility measures the value of achieving the artifact's goal [36].

The next section reports the application and evaluation of AMAN-DA. In our case, the maritime domain was chosen. The stages of the case study development are: design, preparation, collection of data, and analysis of data.

1. Case study design: objectives are defined and the case study is planned.
2. Preparation for the data collection: procedures and protocols for the data collection are defined.
3. Analysis of collected data: data analysis procedures are applied to the data.
4. Collecting evidence: data collection procedures are executed on the studied case.
5. Reporting: the study and its conclusions are packaged in feasible formats for reporting.

We additionally include a subsection for dealing with the threats to validity. Also, Runeson et al. report that researchers should take into account a range of elements when designing a case study. Among these elements, we cite the rationale (why is the case study being done?), the purpose (what is expected to be achieved with the study?), the case (overall, what is being studied?), and the methods for data collection (how will data be collected and analyzed?).

4.1. Case study design

A) Rationale: *Why was the study done?*

The study was undertaken when arriving at a certain stage of the research process, after the AMAN-DA method was proposed. Our team was interested to know more about, to evaluate, and to test the efficacy of AMAN-DA in a real context and with a particular domain of application.

B) Purpose: *What is expected to be achieved with the study?*

After applying AMAN-DA on small-scale examples, we were interested in using AMAN-DA for a large-scale case study. Through one member's contact network, the team got in touch with a captain in the maritime navy working for a well-known maritime company. The case study was then undertaken to analyze the method AMAN-DA for the purpose of evaluation with respect to *efficacy* that measures to what extent AMAN-DA is capable in eliciting security requirements specific to the maritime domain.

As reported in more than one reference [30] [31] [32], security plays a crucial role in the maritime domain nowadays, "*the maritime industry is one of the most heavily targeted industries in the world and also suffers cyber-attacks regularly*." [30]. We decided to explore the use of AMAN-DA to capture stakeholder's security goals in this context and elicit security requirements specific to the maritime domain as part of the modelling of the information system of a bulk carrier independently of the other information systems of maritime transport.

C) Case and Units of Analysis: *What is being studied?*

The study is part of the elaboration of the ship's information system, and focuses on the (security) requirements elicitation phase. The unit of analysis considered is a ship (a bulk carrier) that belongs to a maritime company, which manages various other ships. The ship's information system may have interactions with some other systems such as the port facility information system or the company information system.

As main information, a ship is composed of two departments:

- The deck department.

- The ship's engine department.

The deck department is managed by the captain, the second officer and their subordinates. Its main functions involve administrative tasks such as scheduling work, quality control, coordinating with other departments, and conflict resolution.

The ship's engine department is managed by the chief engineer, the second engineer and their subordinates. The engine department is responsible for all maintenance and operation of the electrical and mechanical equipment. Besides the engines in the engine room, the engine department crew is responsible for all of the sewage, air conditioning, lighting, and water on the ship. In addition to those main departments, crew's cabins, offices, galley, and some stores are handled by the steward, the cook, the baker, the waiter, and their subordinates.

D) Research Questions: *What knowledge will be sought or is expected to be discovered?*

In the beginning of the research project, many research questions motivated the development of AMAN-DA, i) whether relying on domain ontologies or relying on domain ontologies in addition to security ontologies will make a difference in the elicitation of domain specific security requirements (goal)? ii) Weather relying on well-defined rules is a good option to achieve the same goal? iii) What would be the adequate artifacts to structure security and domain knowledge?

AMAN-DA was developed based on domain and security ontologies and well-defined rules to extract the relevant knowledge and generates the requirements specification. The goal of the case study presented in this paper is to evaluate the utility and effectiveness of AMAN-DA, the main research question we investigated during this case study was:

"*To what extent is AMAN-DA able to generate maritime specific domain security requirements?*"

4.2. Preparation for the data collection

A) Methods of Data Collection

The main sources of information for this case study were gathered after long hours of interviews with three members of a cargo ship (the captain, the chief engineer and a duty officer). The interviews lasted 3 consecutive days (~ 15 hours in total) the first time to capture as much information as possible. Then there was a meeting by using a voice call software once every two weeks during two months to review some points and clarify some ambiguities. The interviews were performed as informal interviews, more in the form of a discussion, using the interview instrument (a set of pre-prepared questions)[3] as a guide of areas available to discuss. Some of the interviews were direct, while others were performed via a voice call softare when the captain, the chief engineer, the duty officer or our team was unable to meet.

The interview instrument was constructed by one member (PhD student) and validated by the other members (supervisors). It was adapted slightly as the interviews progressed. Adaptations were primarily made with the purpose of gaining further information about security issues as they are seen on the ship. The interviews concentrated mainly on understanding the structure of a ship, its departments, its employees (crews), and the different interactions between those parts. They identified some useful documents recommended by the maritime experts that deal with security in the maritime domain, as well as the main security goals of a ship as an organization during its travels as well as during its docking. The captain suggested a couple of documents that were also consulted [15] [16] [17]. Some notes were taken during the interviews, but the main form of documentation was the sound recording intended for transcription as a part of the analysis. The interviews were stored into a memory card and analyzed later on the computer.

B) Case and data selection strategy

From the beginning, the team had a strategic goal of looking for a domain where security is a critical issue. It seemed that the same cases were repeated over and over in the literature. Most of those dealing with security were related to the banking or health insurance sectors. We needed a case that was slightly different to increase our

[3] The interview instrument can be consulted on this link: https://www.dropbox.com/s/34nb8d4xo4hwuc8/Interview%20instrument.pdf?dl=0

evidences, strengthen our findings, and avoid bias by tackling a domain that was a little different. Targeting the IS security of a cargo ship is a critical issue since it may impact the safety of persons. The choice of the captain, the chief engineer, and a duty officer as the persons to interview was also a deliberate choice in the selection of the data sources, the three of them being the most "knowledgeable" persons about the organization of the ship and play strategic roles in the ship.

C) Confidentiality and ethical considerations

As the reader may notice, during the whole of the reported case study, the name of the company, the name of the stakeholder, the various agents and any sensitive information are omitted for confidentiality's sake.

4.3. Analysis of collected data

Figure 9 presents an overview of the analysis process. After the interviews with the stakeholders were recorded and notes were taken, the data that had been gathered were transcribed and categorized. We then applied AMAN-DA on the identified security goals. The results of the application (the different models and specifications generated) were discussed with experts. The next sections discuss these steps in detail.

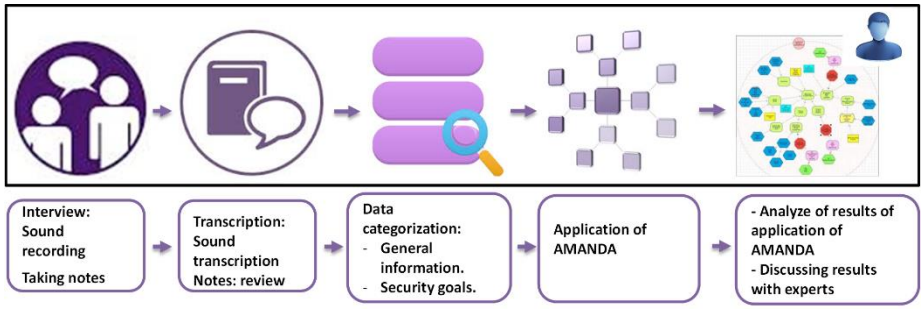


Figure 9. Overview of the analysis process

A) Recording, transcribing and categorizing data

The interviews with the maritime stakeholders were recorded as an audio file; some notes were taken in parallel. The files were then transcribed and the taken notes were reviewed one by one. The information gathered from the interview was mixed and pertains to different aspects of the maritime organization on the ship, it was also related to different levels of abstraction. Despite being semi-structured and based on pre-prepared questions, the interview with the maritime stakeholders tended to go out of the scope of the study from time to time. The transcribed data were dispatched into three categories: Information not useful for the case study; general information useful for the case study; and security goals.

The last category contained clear security goals expressed by the ship's members. The latest were formalized using AMAN-DA security goal model and were part of the input of the method. AMAN-DA was then applied on the captured security goals. The outputs of the application were the different potential threats, vulnerabilities, security requirements that are likely to be present in the domain. The different Secure Tropos models were generated as well as the final specifications.

Table 2 presents part of the data categorization step. The transcribed data were stored in a table with the three categories identified to ensure full traceability.

Table 2. Examples corresponding to the interview with the ship's members.

Category	Some parts of the interview
Information not useful to the case study	"I worked for passenger ships then moved to bulk carriers, it is much easier to work with products than people!"
General useful information	"A ship is composed of the deck department and the engine department. The deck department is managed by the captain, the second officer and their subordinates. The ship's engine department is managed by the chief engineer, the second engineer and their subordinates"
Security goals	"When communicating with other ships, the captain needs to be sure of the identity of his interlocutors, especially in areas of sensitive navigation", "During navigation, the captain would like to maintain the meteorological data, the nautical charts, the ocean current maps, and the logbooks available"

The next two sections present the application of AMAN-DA on some captured security goals.

B) Case study execution and obtained results

Inputs of AMAN-DA for the case study

The following are some security goals (SG_i) captured during the interview. These goals are formalized according to the security goal template model of AMAN-DA (cf. Figure 5).

- (SG_1) The chief engineer is asked to (maintain) $_{verb}$ the (confidentiality) $_{security\ criterion}$ and (integrity) $_{security\ criterion}$ of the (ship's engine plans) $_{asset}$.
- (SG_2) The captain would like to maintain the integrity and confidentiality of the documents that he manages (ship's certificates, ship's plans, and communication certificates). These documents can be consulted by the SSO (Ship Security Officer).
- (SG_3) Due to previous incidents, the captain would like to maintain the availability of the communication equipment on board such as the VHF (Very High Frequency) radio, the walkie-talkies.
- (SG_4) During navigation, the captain would like to maintain the availability of the meteorological data, the nautical charts, the ocean current maps, and the logbooks.
- (SG_5) The company would like to keep the evaluations of the ship's crews confidential; these evaluations allow for recruitment of a crew for other missions.
- (SG_6) During loading/unloading of goods, the captain would like to preserve the integrity of the bill of lading document, the stability booklet and the stowage plan.
- (SG_7) He would also like to make them available for the department members.
- (SG_8) The system should ensure the non-repudiation of communication between the ship and the other ships navigating in the same area.

In addition to the captured security goals, AMAN-DA takes as inputs:

- **A maritime domain ontology**: the ontology, presented in Figure 18, contains 85 concepts and 115 relationships related to the maritime domain. As recommended by AMAN-DA, the ontology contains the main organization (maritime company), agents of the organization (crew, chief engineer, company director, ship's captain, etc.), objects (ship pipes, engine plans, etc.) Different actions between agents (Maintain ship engine, manage ship documents, deliver the goods, etc.) This maritime domain ontology was developed in beforehand from various maritime sources; its concepts are instantiations of an upper domain ontology that covers domain knowledge proposed in the context of AMAN-DA. The reader may refer to section 3.1 for more details.

- **Core security ontology:** the core security ontology, presented in eearlier in the article is the third input. It has the necessary security related knowledge for the security requirements process; it contains number of potential threats, vulnerabilities, and security requirements.

In the following will be presented a potential analysis scenario of two goals and the generated specifications (models and texts)

1. Potential scenario with SG_1 and SG_4

Tables 3 and 4 present a potential analysis scenario of security goals (SG_1 and SG_4). With the four security goals as an input, the analysis through the use of AMAN-DA discovers 8 potential threats, 17 potential vulnerabilities, and 28 potential security requirements. The same analysis can be applied to the other security goals. Each table is organized as follows: The name of the organization is taken from the domain ontology, the security goal is formalized according to AMAN-DA, the organizational goals and actors are discovered from the domain ontology. The potential threats and their corresponding vulnerabilities are discovered from the security ontology. Finally, the corresponding security requirements are elicited using both the security and the domain ontology. Table 3 presents the analysis of SG_1 and Table 4 presents the analysis of SG_4.

Table 3. Analysis of SG_1

Goal and asset analysis	
Organization	Maritime organization.
Security Goal (SG_1)	(Maintain) $_{\text{verb}}$ the (confidentiality) $_{\text{Security criterion}}$ of the (engine plans) $_{\text{asset}}$
Organizational goal	Maintain ship engine
Actors	Captain Chief engineer
Threat analysis	
Potential threats (Part of)	(engine plans)
	T1. Unauthorized physical access.
Potential vulnerabilities	**V1.** No key management. **V2.** No surveillance cameras. **V3.** No alarm system. **V4.** Open windows.
Security requirements elicitation	
Potential security requirements (Part of)	**V1**.
	Req1. The chief engineer should set proper termination procedures. **Req2.** The chief engineer should sign and account engine room keys, bridge keys, captain cabin keys and crew keys.
	V2.
	Req3. The chief engineer should provide an adequate maritime surveillance system.
	V3.
	Req4. The chief engineer should equip the structures with ship security alarm system, machinery space alarm.
	V4.
	Req5. After working hours, the chief engineer should lock engine room windows.

Table 4. Analysis for SG_4

Goal and asset analysis				
Organization	Maritime organization.			
Security Goal (SG_4)	(Maintain)$_{verb}$ the (availability)$_{Security\ criterion}$ of the (meteorological data, ocean current maps) $_{asset}$			
Organizational goal	Manage weather data and equipment.			
Actors	Captain Weather officer			
Threat analysis				
Potential threats **(Part of)**	(meteorological data)			
	T1. Configuration error. **T2.** Untrained personal **T3.** Failure of systems. **T4.** Network attack.			
Potential vulnerabilities	**T1**	**T2**	**T3**	**T4**
	V1. No change of preset password. **V2.** Insecure installation and configuration of password.	**V3.** Lack of training.	**V4.** No backups. **V5.** Insufficient maintenance.	**V6.** No secure internet connection.
Security requirements elicitation				
Potential security requirements (Part of)	**V1**.			
	Req1. The duty officer should use adequately fixed marine padlocks.			
	V2.			
	Req2. Regularly, the duty officer should make and update the bridge inventory, bridge documentation inventory, the navigation equipment inventory, ship radio station equipment inventory, replacement part inventory, paint inventory, cleaning equipment inventory.			

2. Generated Secure Tropos models

Figures 10 to 12 represent the generated Secure Tropos models after applying AMAN-DA to the previous security goals (SG_1 and SG_4). These models correspond to the analysis made in tables 3 and 4. For instance, Figure 10 represents part of the organizational view generated. It contains the main actors (captain, Company director, weather engineer, duty officer, chief engineer) and the goal dependencies between them as well as the name of the organization (maritime organization).

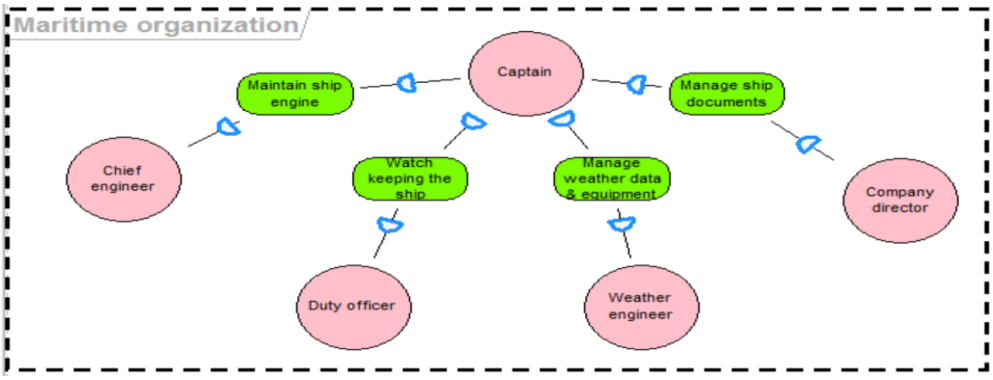


Figure 10. Part of the organizational view.

Figure 11 represents the Secure Tropos attack view; it displays the threat (Unauthorized physical access), the attack method (social engineering) and vulnerabilities exploited (V1 to V4).

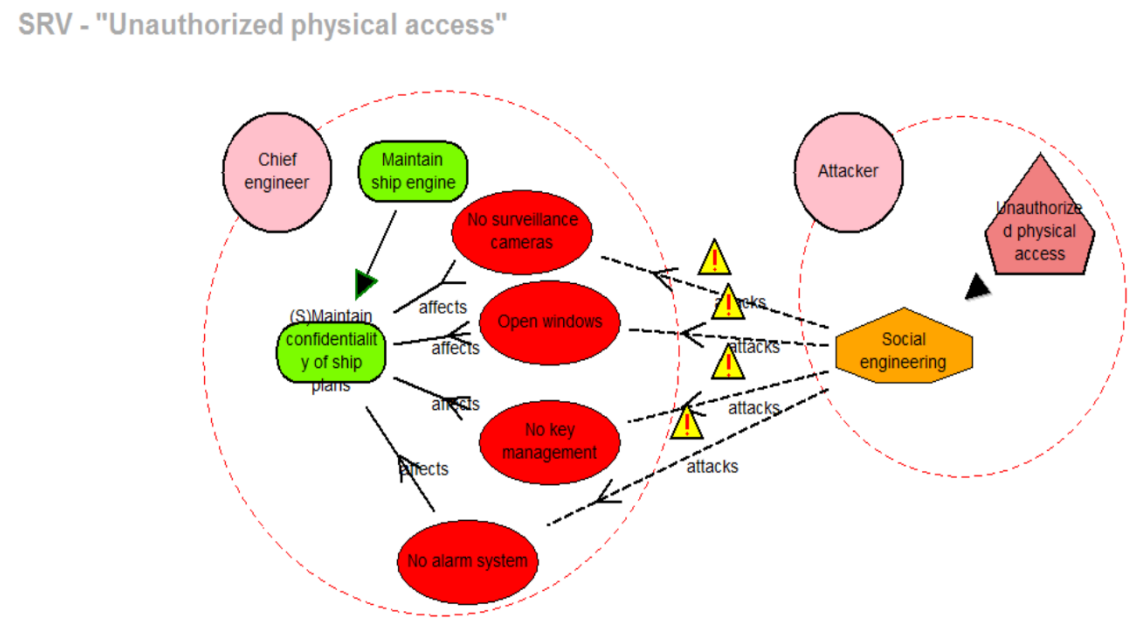


Figure 11. Part of the attack view (SG_1)

Figure 12 represents part of the security requirements view that corresponds to the security goal (SG_1). This view includes the main security requirements to be considered (security constraints in Secure Tropos).

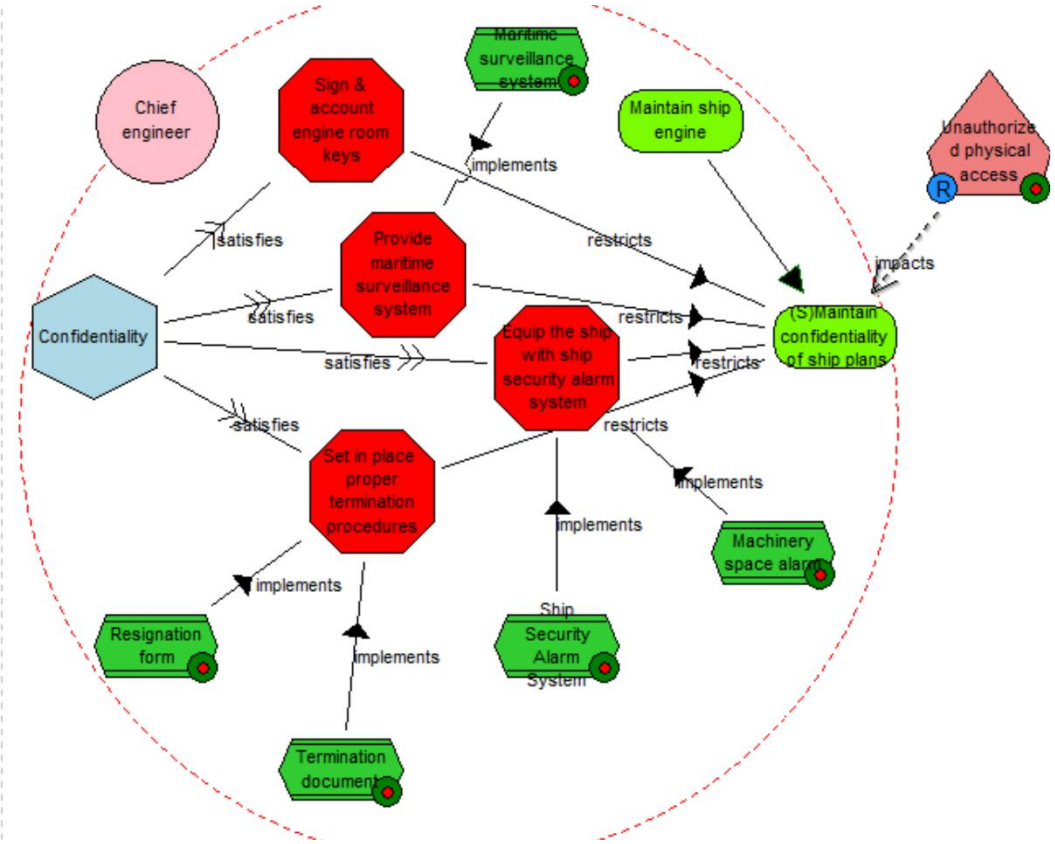


Figure 12. Part of the security requirements view (SG_1)

3. Generated textual specification

The other output of AMAN-DA is the textual specification related to the maritime case study. Figure 13 represents part of the specification generated automatically; it contains the name of the organization, the different assets to protect, the potential threats, the potential vulnerabilities, and the security requirements to consider.

```
Organization:    Maritime organization
_____________________________________________________________________________________
Asset: engine plans
Security Criterion: Confidentiality
Threat: Unauthorized physical access.
Vulnerabilities:
-> No key management
  Security Requirements:
  (Req1.) The chief engineer should set proper termination procedures.
(Req2.) The chief engineer should sign and account engine room keys, bridge keys,
captain cabin keys and crew keys.

-> No surveillance cameras
Security Requirements:
(Req3.) The chief engineer should provide an adequate maritime surveillance system.

-> No alarm system
Security Requirements:
(Req4.) The Chief engineer should equip the structures with ship security alarm system,
machinery space alarm.

*****************************************************************************************
Asset: ship certificates
Security Criterion: integrity
Threat: Fire
Vulnerabilities:

-> No testing of fire extinguisher
 Security Requirements:
 (Req1.) The ship security officer should test FlexiFog fire extinguishers, FlexiFoam
fire extinguishers, and gaseous based fire extinguishers.

-> Open fire doors
 Security Requirements:
 (Req2.)After working hours, the ship security officer should lock bridge, engine and
galley rooms' doors.

->No fire suppression
 Security Requirements:
 (Req3.) The ship security officer should provide an automated and tested maritime fire
suppression system.
 (Req4.) The ship security officer should make available the fire extinguishers
(FlexiFog, FlexiFoam, and gaseous fire extinguishers)
-> No regular testing of pipes
 Security Requirements:
 (Req5.) The ship security officer should test the ship pipes.
-> No water detector
 Security Requirements:
 (Req6.) The ship security officer should provide an automated and tested water flood
sensor to the relevant areas.
```

Figure 13. Textual specification of security requirements related to the maritime case study.

Table 5 in the following gives the number of the produced artifacts in the case study. Hence, with one security goal as an input, and one threat chosen (from the ontology), four vulnerabilities are potential and five security requirements can be derived. With four security goals as an input, eight potential threats can be chosen, 17 potential vulnerabilities and 28 security requirements. With 8 security goals, 21 potential threats can be chosen from the ontology and 44 security requirements derived.

Table 5. Number of produced artifacts.

Security goals	Threats	Vulnerabilities	Security requirements
1	1	4	5
4	8	17	28
8	21	38	44

5. Evaluating AMAN-DA with experts

As a continuity to the case study, the usefulness of AMAN-DA was evaluated with a group of experts. The usefulness is defined the degree to which the artifact positively impacts the task performance of individuals [37]. The group contained twelve experts, their average age was 31 years old (min. 25, max. 38). 7 of them were women, 5 men. Among the participants there were maritime domain experts, requirements engineering scientists, security standardization experts, PhD students chosen based on their subjects of research that were related to our research. First, the context of the project and the method were presented to them. Second, participants manipulated AMAN-DA though its tool, and reviewed the Secure Tropos model and security requirements produced by the application of AMAN-DA using its tool. At the end of the presentation, the experts were invited to fill in a questionnaire. The questionnaire contained 24 questions and aimed to ask the participants of their point of view on the usefulness of the method and its tool. Each question proposed four possible choices (strongly agree, agree, neither agree nor disagree, disagree, strongly disagree). The full list of questions can be consulted in the link: (https://www.dropbox.com/s/wpk2vq0r9peukb5/Evaluation%20form.pdf?dl=0). The evaluation was followed by a discussion summarized a bit further in this article. Table 6 presents some of the questions addressed during the evaluation.

Table 6. Questions about the usage of the method

N°	Question
Q14.	Do you think the method makes an improvement (compared to your usual practice) in the elicitation of security requirements for specific domains?
Q15.	Do you think that the method will be effective in discovering new security requirements for the specific domain compared to other methods?
Q16.	Do you think the method is useful overall?

Figure 14 and Figure 15 report the results of participants' answers to Q14 and Q15. The results express "agreement" and "strong agreement" of the participants regarding the advantages of the method for the elicitation of security requirements for specific domains compared to their previous practices using other methods. One participant view was: "***This is definitely better, provided that the domain specific ontology is adequate***". Another participant mentioned: ***"Yes, this is better because many specific threats/vulnerabilities, and requirements are***

listed for each asset." All participants agreed on the usefulness of the proposed method (Figure 16, Q16), "***In particular with regard to traditional risk assessment methods like EBIOS*** [26] "according to a participant.

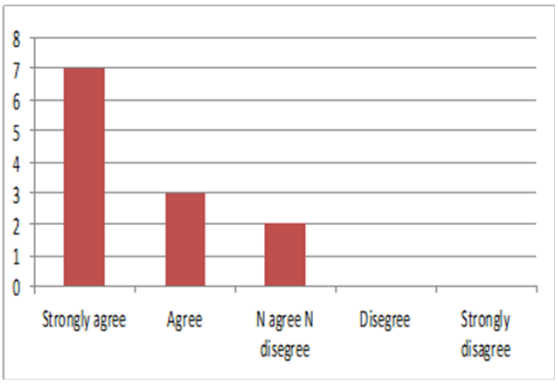


Figure 14. Results of Q14: Do you think the method makes an improvement in the elicitation of security requirements for specific domains?

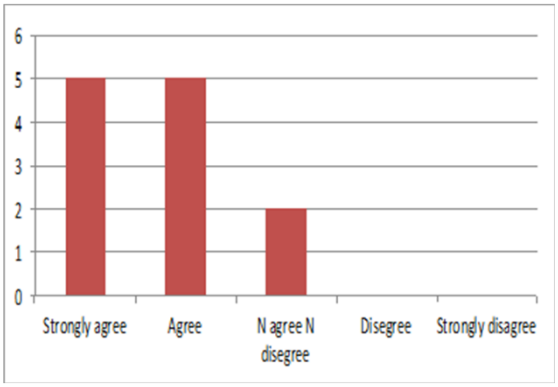


Figure 15. Results of Q15: Do you think that the method will be effective in discovering new security requirements for the specific domain comparing to other methods?

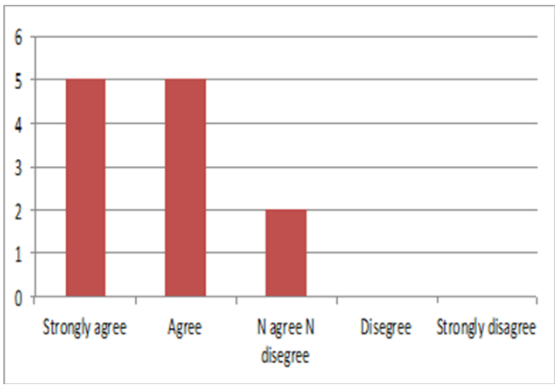


Figure 16. Results of Q16: Do you think the method is useful overall?

Overall, we can summarize the results of the discussions with experts in the following:

- First, the main recurrent comments made by the experts were that the method and its idea of automation are "interesting", and "important".

- From the results of application of AMAN-DA in the maritime domain, experts consider that the method is usefull to produce domain specific security requirements but asked to see more applications to other domains for a better generalization of this statement. The criticism of experts was that one application to a single domain may lead to consider the method dependent on this domain and not generic.

- One expert was interested in the richness of the security requirements modeling language chosen (Secure Tropos in our case) and asked whether it was enough to model all the domain concepts. Also, he asked why it is the only language that the method handles.

The method and its tool were developed to assist and guide users in the elicitation of security requirements during early stages of systems developments – a task known to be difficult due to the tacit, informal knowledge of these users about security and the domain at hand. This step is often left to ad hoc practices such as copy pasting security requirements from other projects. It is even harder when it comes to building security requirements models. Users left to the concepts of the language (Secure Tropos for example) cannot do much.

Our method and its tool meet users' needs (as the results of the experiment demonstrate). The developed core security ontology, the reasoning rules that we defined to elicit and build Secure Tropos models, and the mechanisms we use to make these requirements more domain specific via the use of different domain ontologies, all make it possible to obtain a structured and structuring outcome.

Many ideas were proposed by the participants of the experiment, to improve the method and to make the tool more attractive.

6. Threats to validity

There are few threats to validity during the application of AMAN-DA to the case study; they are discussed in the following points.

Construct validity: the main threat to construct validity in the case of AMAN-DA is the number of security requirements generated and their quality. Although the method has been defined on well-formalized ontologies and rules, its main goal is the semi-automatic elicitation of requirements; there remain room for improving the quality of generated security requirements (models and texts). The case study has shown the ability of AMAN-DA to produce domain specific security requirements, but it can be improved by comparing the number of generated security requirements with the ones generated by a manual human expert.

External validity: threats to external validity are concerned about whether a study's finding can be generalized. The case study was selected based on an existing contact network. Due to the timing of the research, we could not study another context. Application to only one domain and with one company might not be enough to generalize the results.

Conclusion validity: The experts involved in discussing and analyzing the results included one of the paper authors, which may have left room for subjectivity.

7. Lessons learned and discussion

The application of AMAN-DA to the case study might change established thought, in the sense that security knowledge by itself is not enough if it is not completed with domain knowledge. Vice versa, domain expertise alone remains weak if it is not reinforced by security knowledge. This result is a step forward for the research community. Recall that, before AMAN-DA, some propositions were limited to relying on generic security ontologies only (RITA [8], Velasco et al. [19]) and their shortcoming was that the resulting security requirements were too generic and not specific to the domain at hand. Now the new open question is whether using both domain and security ontologies is sufficient, or using domain specific security ontologies (i.e., constructing security ontologies for each domain and using them in AMAN-DA) would be a better solution.

This study can be considered as another piece of evidence that should persuade the research community of the importance of knowledge reuse, especially with ontologies, in future research on methods engineering for information systems. Requirements engineering methods should not be limited to the conceptual level (metamodeling) but need to consider knowledge reasoning and reuse as part of the elicitation and analysis process.

The other point learnt or noticed is that, even with the security goal model proposed by AMAN-DA, the step of capturing security goals from stakeholders remains difficult; careful recording and transcribing of stakeholder interviews was needed before applying the method. This step could be improved in further research on how to conduct an interview with stakeholders successfully in order to capture security goals more easily. For the researchers, another open question would be: what are the necessary parameters to take into consideration in the formulation of the security goals and security requirements? For instance, the level of security (i.e., low, medium, high) is an example to explore. Future versions of AMAN-DA should consider other assumptions on the security goals; for instance, the verbs can be taken from a closed list or taxonomy of possible verbs.

As noted in threats to validity, the quality of the produced security requirements has not been the primary concern in the method. Further research should tackle this aspect by systematically and formally verifying whether the elicited requirements are consistent with the INCOSE guide for writing requirements [43].

It is known from previous surveys presented at REFSQ [20] that practitioners (requirements engineers, security officers, security engineers, etc.) do some kinds of reuse; the commonest technique is copy-pasting from other projects (more than 50%). This is an ad-hoc way of working. The case study has shown that AMAN-DA provides a methodological and organized way to produce specifications (c.f. Figures 10-13). Participants in the evaluation agreed on this. The remaining open question here is what the result of application for AMAN-DA with other domains would be.

Overall, the AMAN-DA method can be useful to practitioners (mainly requirements engineers, but also security officers, security engineers, etc.). This method will help beginners in requirement engineering by facilitating the process of identifying, analyzing, specifying, and managing security requirements. Requirements teams often do not include subject matter experts in security [44]. Such a body of knowledge can be made available to this intended audience.

The method's outputs (graphical and textual specifications) should help security, software and information systems architects, because they reuse knowledge at the corporate level, and their responsibilities include leveraging knowledge reuse. For any given set of requirements, an architect can and should typically identify and evaluate multiple different architectures and architectural mechanisms before selecting what he or she thinks will be the optimum way of fulfilling the requirements. Unfortunately, poor requirements specification weighs on architects who often find themselves obliged to do a job from scratch that is beyond their scale of time or responsibilities, which badly influences on the final product.

8. Related work

The idea of exploring ontologies for security requirements engineering is quite recent and it is gaining more and more attention among researchers. Velasco et al. [19] propose an ontological representation for reusable requirements, which allows incompleteness and inconsistency in requirements to be detected and semantic processing in requirements analysis to be achieved. However, the framework seems to be at an early stage, in the sense that it does not permit security requirements elicitation and analysis. To date, the contribution is limited to the proposed ontologies. Kaiya and Saeki [22] explored a domain ontology for requirements elicitation. Farfeleder et al. [23] did some work on ontology-guided requirements elicitation as part of an EU/ARTEMIS project called CESAR. Their contribution relies on a domain ontology to produce textual requirements. The down side here is that their contribution focused more on safety requirements rather than security, and did not deal with security requirements models. The work by Ruhroth et al [24] is interesting; authors propose an approach for ontology-aided reuse of knowledge, they define a set of operators to add, change or remove knowledge for reuse purposes. To the best of our knowledge, the related work cited above was limited to textual requirements. AMAN-DA explores a security ontology in addition to a domain ontology and produces security models and not just textual security requirements.

Besides the use of ontologies, some interesting contributions exist for requirements conceptual modelling, the ISSRM model [1] (sISSRM in its latest version [33]) is a conceptual meta-model that represents the concepts used for risk based security requirements engineering, it has been designed from a systemic security literature: risk management standards, security-related standards, security risk management standards and methods, and security requirements engineering frameworks. This meta-model is a step forward into unifying the language that (security) requirements engineers use, it has been a good reference when building the core security ontology in AMAN-DA. However, to the best of our knowledge, the ISSRM approach do not rely on automated knowledge repositories. The choice in AMAN-DA to rely on ontologies was motivated by the fact that these ones support run-time knowledge exploitation and automated reasoning.

As far as we know, AMAN-DA is the first contribution that attempts to produce security requirements models for different domains relying on well-formed ontologies.

9. Conclusion

This paper presented the AMAN-DA method and its application on a real-world case study. The originality of the method lies: (a) in the fact that the combination of the security ontology and domain ontologies is not achieved a

priori, they are completely separated, but their respective knowledge is used alternatively, while the method is applied, and (b) in the genericity of the method, in the sense that it is designed to be used with a generic security ontology and any domain ontologies, as long as they embed some expected knowledge. (c) The defined rules and presented process allows the method to automatically exhibit an appropriate ontological semantics (security and domain specific) to the requirements engineer (agents, objects, threats, security requirements…).

The case study carried out and reported in this paper is a first evaluation of AMAN-DA for security requirements elicitation. The obtained results make us appreciate AMAN-DA as a promising approach. The case study (the security goals) was elaborated with experts from maritime domain. The generated models and specification demonstrate the ability of AMAN-DA to produce a domain specific security requirements engineering analysis. This promising contribution needs to be reinforced by experimenting AMAN-DA in other domains to evaluate its genericity.

Acknowledgment

Authors would like to thank Dr. Zeinab Hmedeh for her valuable help during the development of the AMAN-DA tool and Prof. Bénédicte le Grand for her fruitful discussions and feedbacks all over the AMAN-DA project.

References

[1] Mayer N. Model-Based Management of Information System Security Risk. Presses universitaires de Namur. 2012

[2] Fenz S., Ekelhart A. « Formalizing information security knowledge ». In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 183-94. ASIACCS '09. New York, NY, USA: ACM. 2009.

[3] Haley, C.B., Laney R., Moffett J.D., and Nuseibeh B.. « Security Requirements Engineering: A Framework for Representation and Analysis ». IEEE Transactions on Software Engineering 34 (1): 133-53. 2008.

[4] Tondel I. A., Jaatun M. G., and Meland P. H.. « Security requirements for the rest of us: A survey ». Software, IEEE 25 (1): 20-27. 2008.

[5] Meier, J. D. « Web application security engineering ». Security & Privacy, IEEE 4 (4): 16-24. 2006.

[6] Firesmith D. G. «Specifying reusable security requirements ». Journal of Object Technology 3 (1): 61-75. 2004.

[7] Zuccato, A., Daniels N., Jampathom C.. « Service Security Requirement Profiles for Telecom: How Software Engineers May Tackle Security ». In the Sixth International Conference on Availability, Reliability and Security (ARES'11), 521-26. doi:10.1109/ARES.2011.81. 2011

[8] Salinesi C., Ivankina E., Angole W. « Using the RITA Threats Ontology to Guide Requirements Elicitation: an Empirical Experiment in the Banking Sector ». In the First International Workshop on Managing Requirements Knowledge, 2008. MARK '08., 11-15. 2008.

[9] Souag A., Salinesi C., Comyn-Wattiau I., Mouratidis H., « Using Security and Domain Ontologies for Security Requirements Analysis », in Computer Software and Applications Conference Workshops (COMPSACW), 2013, p. 101-107.

[10] Mouratidis H., Giorgini P. « Secure Tropos: A Security-Oriented Extension of the Tropos Methodology.». International Journal of Software Engineering and Knowledge Engineering 17 (02): 285-309. 2007.

[11] Souag A., Mazo R., Salinesi C., Comyn-Wattiau I., « Reusable knowledge in security requirements engineering: a systematic mapping study », Requirements Engineering Journal., p. 1-33. 2015.

[12] Souag A., Salinesi C., Mazo R., Comyn-Wattiau I., « A Security Ontology for Security Requirements Elicitation », in Engineering Secure Software and Systems, F. Piessens, J. Caballero, N. Bielova, Ed. Springer International Publishing, 2015, p. 157-177.

[13] Runeson P., Host M., Rainer A., Regnell B. « Case Study Research in Software Engineering: Guidelines and Examples ». 1 edition. Hoboken, N.J: Wiley. 2012.

[14] Yin R. K. « Case study research: Design and methods ». Sage publications. 2014.

[15] Inetrnational Maritime Organization, « International Convention for the Safety of Life at Sea (SOLAS), 1974».

[16] International Maritim Organization, « ISPS Code ». 2011.

[17] Chebli A. S. « La piraterie maritime au début du XXième siecle: panorama, modes opératoires et solutions». Mémoire pour le DU Analyse des menaces contemporaines. 2009.

[18] Norton. 2013. « 2013 Norton Report ».

[19] Velasco J. L., Valencia-Garcia R., Fernandez-Breis J. T., Toval A. 2009. « Modelling Reusable Security Requirements Based on an Ontology Framework ». Journal of Research and Practice in Information Technology 41 (2): 119.

[20] Palomares C., Franch X., and Quer C. "Requirements reuse and patterns: a survey." Requirements Engineering: Foundation for Software Quality. Springer International Publishing, 2014. 301-308.

[21] Rupp C., Simon M., Hocker F. 2009. « Requirements engineering und management ». HMD Praxis der Wirtschaftsinformatik 46 (3): 94-103.

[22] Kaiya, H., Saeki M. 2006. « Using Domain Ontology as Domain Knowledge for Requirements Elicitation ». In the 14th IEEE International Conference on Requirements Engineering, ,189-98.

[23] Farfeleder, S., Moser, T., Krall, A., Stålhane, T., Zojer, H. and Panis, C., 2011. "DODT: Increasing requirements formalism using domain ontologies for improved embedded systems development". In Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2011 IEEE 14th International Symposium on (pp. 271-274). IEEE.

[24] Ruhroth, T., Gärtner, S., Bürger, J., Jürjens, J. and Schneider, K., 2014. Towards adaptation and evolution of domain-specific knowledge for maintaining secure systems. In Product-Focused Software Process Improvement (pp. 239-253). Springer International Publishing.

[25] Prat, N., Comyn-Wattiau, I. and Akoka, J., 2015. A Taxonomy of Evaluation Methods for Information Systems Artifacts. Journal of Management Information Systems, 32(3), pp.229-267.

[26] EBIOS., Secrétariat Général De la Défense Nationale, 2004. « EBIOS-Expression des Besoins et Identification des Objectifs de Sécurité ». http://www.ssi.gouv.fr/guide/ebios-2010-expression-des-besoins-et-identification-des-objectifs-de-securite/

[27] Souag, Amina. Towards a new generation of security requirements definition methodology using ontologies. CAiSE, Jun 2012, Gdansk, Poland. pp.1-8.

[28] Souag, Amina. AMAN-DA: A knowledge reuse based approach for domain specific security requirements engineering. 2015. Dissertation. Université Paris 1 Panthéon Sorbonne.

[29] Mouratidis, H. Secure software systems engineering: The Secure Tropos approach. JSW. 2011 Mar; 6(3):331-9.

[30] Belmont Kate B., Maritime Cyber Attacks: Changing Tides. Last modified Nov 2015. http://maritime-executive.com/blog/maritime-cyber-attacks-changing-tides.

[31] Paganini Pierluigi, Hacking Ships: Maritime Shipping Industry at Risk. Last modified March 31, 2015. http://securityaffairs.co/wordpress/35504/hacking/hacking-maritime-shipping-industry.html

[32] Fitton, O., Prince, D., Germond, B., & Lacy, M. (2015). The future of maritime cyber security.

[33] Naudet Y., Mayer N., Feltus C. Towards a Systemic Approach for Information Security Risk Management. In: Availability, Reliability and Security (ARES), 2016 11th International Conference on. IEEE, 2016. p. 177-186.

[34] Checkland, P., and Scholes, J. Soft Systems Methodology in Action. Chichester:Wiley, 1990.

[35] Venable, J.; Pries-Heje, J.; and Baskerville, R. A comprehensive framework for evaluation in design science research. In K. Peffers, M. Rothenberger, and B. Kuechler (eds.), Proceedings of the Seventh International Conference on Design Science Research in Information Systems and Technology (DESRIST 2012). Las Vegas: Springer Verlag, 2012, pp. 423–438.

[36] Gregor, S., and Hevner, A.R. Positioning and presenting design science research for maximum impact. MIS Quarterly, 37, 2 (2013), 337–355.

[37] Davis, F.D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13, 3 (1989), 319–340.

[40] Bjørner D. 2010. « Rôle of Domain Engineering in Software Development—Why Current Requirements Engineering Is Flawed! ». In Perspectives of Systems Informatics, edited by Amir Pnueli, Irina Virbitskaite, Andrei Voronkov, 2-34. Lecture Notes in Computer Science 5947. Springer Berlin Heidelberg.

[41] Kaiya, H., Saeki M. 2006. « Using Domain Ontology as Domain Knowledge for Requirements Elicitation ». In the 14th IEEE International Conference on Requirements Engineering, ,189-98.

[42] Secure and Dependable Software Systems. University of Brighton. 'Modelling Method Conceptualisation within OMiLab: The Secure Tropos approach'. May 2017. http://vienna.omilab.org/repo/files/T-SecTr/2017-05-12%20RCIS%202017%20SecureTroposTutorial.pdf

[43] Requirements Working Group. "International Council on Systems Engineering (INCOSE), "Guide for Writing Requirements", INCOSE (2012).

[44] D. Firesmith, « Specifying reusable security requirements », Journal of Object Technology, vol. 3, no 1, p. 61-75, 2004.

[45] Peffers K., Tuunanen T., Rothenberger M. A., Chatterjee S. 2007. « A design science research methodology for information systems research ». Journal of management information systems 24 (3): 45-77.

[46] Souag, A., Salinesi, C., & Comyn-Wattiau, I. 2012. Ontologies for security requirements: A literature survey and classification. In Advanced Information Systems Engineering Workshops (pp. 61-69). Springer Berlin Heidelberg.

[47] Eisenhardt, K. M. (1989). Building theories from case study research. Academy of management review, 14(4), 532-550.

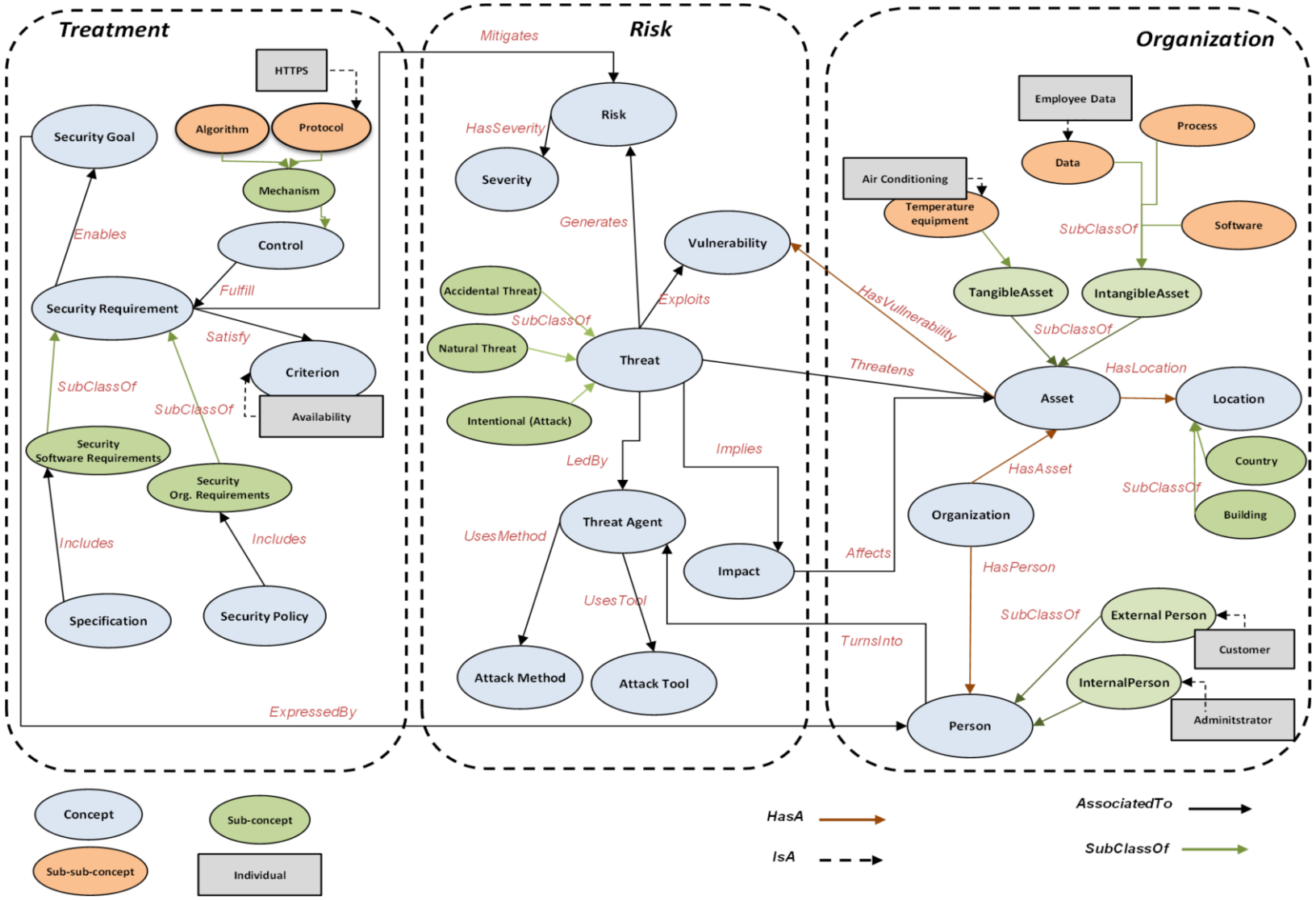


Figure 17. The core security ontology

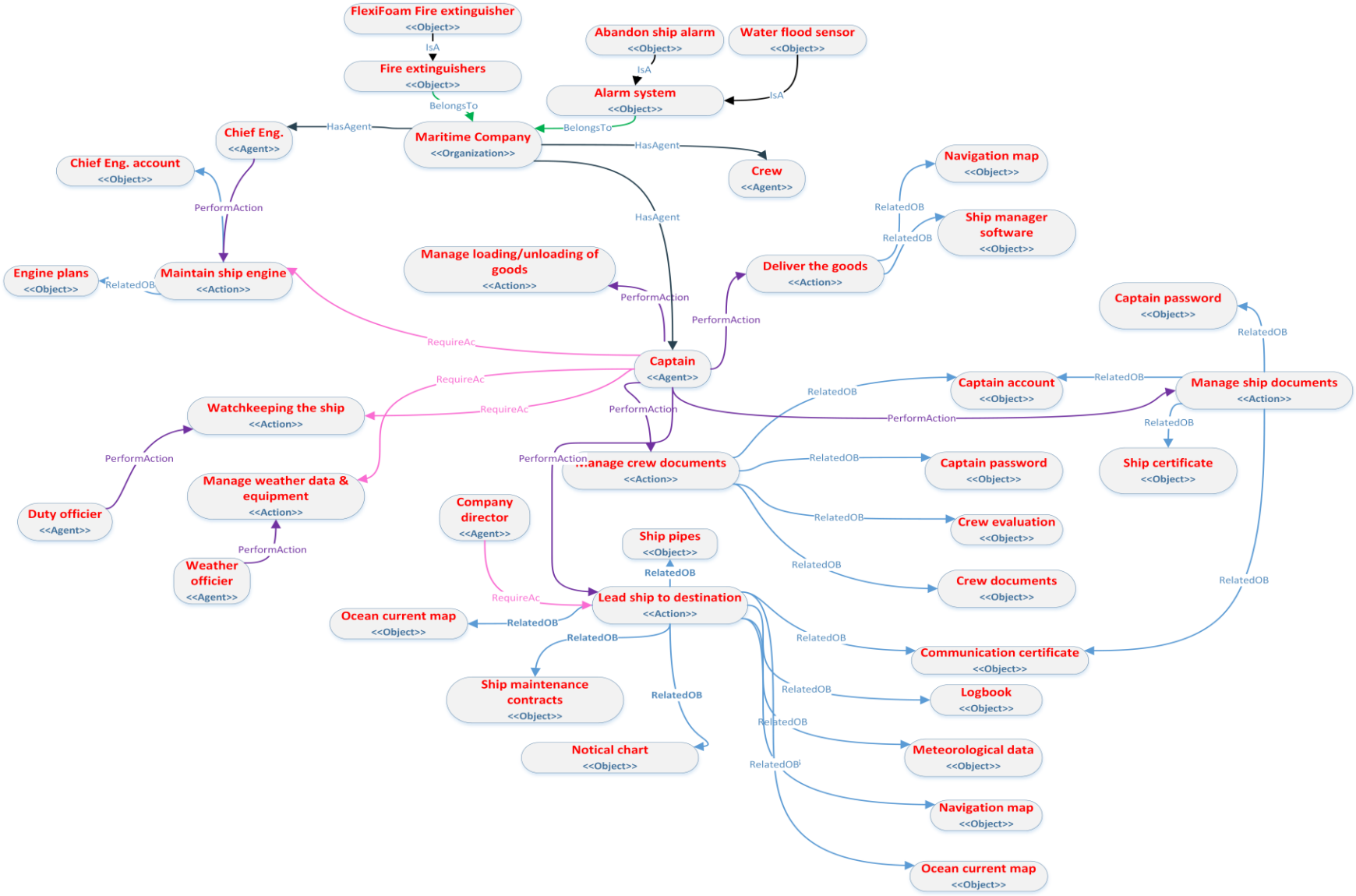


Figure 18. Maritime domain ontology