Using Security and Domain ontologies for Security Requirements Analysis

Amina Souag, Camille Salinesi, Isabelle Wattiau, Haralambos Mouratidis

Using Security and Domain ontologies for Security Requirements Analysis

Amina Souag, Camille Salinesi, CRI, Panthéon Sorbonne University
Paris, France
{Amina.Souag, camille.salinesi}malix.univ-paris1.fr

Isabelle Wattiau, CEDRIC-CNAM & ESSEC Business School, France
isabelle.wattiau@cnam.fr
Haris Mouratidis, School of Architecture, Computing and Engineering, University of East London, UK
H.Mouratidis@uel.ac.uk

***Abstract*— Recent research has argued about the importance of considering security during Requirements Engineering (RE) stage. Literature also emphasizes the importance of using ontologies to facilitate requirements elicitation. Ontologies are known to be rich sources of knowledge, and, being structured and equipped with reasoning features, they form a powerful tool to handle requirements. We believe that security being a multi-faceted problem, a single security ontology is not enough to guide SR Engineering (SRE) efficiently. Indeed, security ontologies only focus on technical and domain independent aspects of security. Therefore, one can hypothesize that domain knowledge is needed too. Our question is "how to combine the use of security ontologies and domain ontologies to guide requirements elicitation efficiently and effectively?" We propose a method that exploits both types of ontologies dynamically through a collection of heuristic production rules. We demonstrate that the combined use of security ontologies with domain ontologies to guide SR elicitation is more effective than just relying on security ontologies. This paper presents our method and reports a preliminary evaluation conducted through critical analysis by experts. The evaluation shows that the method provides a good balance between the genericity with respect to the ontologies (which do not need to be selected in advance), and the specificity of the elicited requirements with respect to the domain at hand.**

Keywords— Security, ontology, domain, requirements elicitation, analysis, method

I. INTRODUCTION

Security is the discipline concerned with protecting systems from a wide range of threats that break the system by exploiting a vulnerability, i.e. a property of the system or its environment that, when faced with particular threats, can lead to failures [1]. Security is a multi-faceted problem; it is as much about understanding the domain in which systems operate as it is about the systems themselves. Recent research has argued about the importance of considering security during RE, SR (SR). Some approaches have considered knowledge-based requirements elicitation, especially with ontologies [3]. Our own experience with RITA [2] a requirements elicitation method that exploits a just one threat ontology, was that "being generic, the threats in the RITA ontology are not specific to the target [bank] industry". Experts involved in the evaluation complained about "the lack of specificity of the types of threats to the industry sector and the problem domain at hand". The problem that remains open is therefore that we need both to exploit security knowledge and domain knowledge to guide the elicitation of domain-specific SR. Our proposal is to use in combination two kinds of ontologies: a security ontology that embeds security specific knowledge, and domain ontology that encompasses domain specific knowledge. The expected outcome is that the SR resulting from the combined use of both ontologies will be more domain specific. The difficulty lies in (a) making sense out of two ontologies that are developed separately for different purposes, and to actually (b) provide some sort of assurance towards the completeness of the requirements elicited with a method that combines security and domain ontologies. Besides, how to be efficient if the proposed method relies on pre-selected ontologies? We need a method that works, as far as possible, with any security ontology and any domain ontology.

This paper presents a method that explores the use of security and domain ontologies for SRE. The approach is generic in the sense that different security ontologies and different domain ontologies can be used with it. However it is domain specific when it is applied in the sense that during its application only one domain ontology is used. The method relies on a collection of heuristic rules that extract relevant security and domain knowledge from ontologies. In addition to the two ontologies, the input are security goals (such as the ones that define a security policy, or the ones that are specified during early requirements elicitation). The output is a specification of SR formalized with Secure I* [4]. The originality of the method lies: (a) in the fact that the combination of security and domain ontologies is not achieved a priori, but at runtime, while the method is applied, and (b) in the genericity of the method, in the sense that it is designed to be used with any pair of security and domain ontologies, as long as they embed some expected knowledge. Our null hypothesis H0 is that "Using domain ontologies in addition to security ontologies to guide SR elicitation does not make a difference with respect to discovering requirements with a security ontology only". To reply to the research question "how to combine the use of security ontologies and domain ontologies to guide requirements elicitation efficiently and effectively?" we undertook a research method based primarily on the general methodology of design science presented in [5].

The rest of the paper is structured as follows: Section 2 introduce preliminary concepts, section 3 presents an overview of our method and its main driving principles. Section 4 reports first evaluations, followed by threats to validity in section 5, Section 6 discuss related works, and section 7 summarizes our current progress and announces future work.

II. PRILIMINARY CONCEPTS

SR pose a set of challenges to RE that together make it a distinctive area of investigation. SRE encompasses the methods, models and frameworks for deriving SR.

A. Modelling security requirements

In this research we focus on goal-oriented approaches to SR engineering. Many publications [4], [7] show that goal modelling is appropriate to handle SR analysis; we decided to use I*, which was specifically adapted to handle security issues. The I* dialect presented in [4] offers a methodological framework to deal with SR, it supports a set of analyses, like dependency vulnerability analysis, countermeasure analysis, access control analysis.

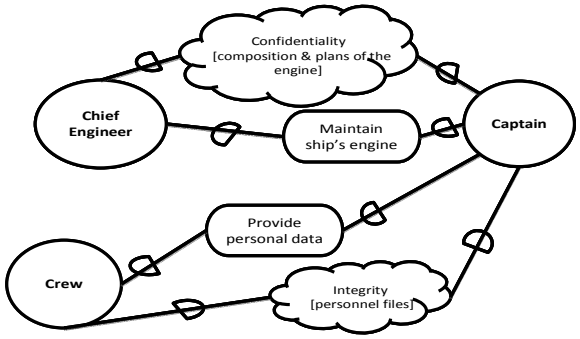

Fig. 1. Part of the initial SR model with I* (maritime domain)

Figure 1 shows an I* initial requirements model that we constructed during the elicitation step with stakeholders (maritime domain). The model presents main actors (company, captain, chief engineer and crew), also goals and security softgoals between these actors, like the confidentiality of plans and composition of the ship's engine. This model was used during evaluation process (See. point 5 of the article)

B. Security and domain ontologies

Several ontologies were proposed in the literature. Our goal for is not to create new ones, but rather to rely on any existing one so as to be able to switch from one ontology to another. In [6] we have reviewed, and classified existing security ontologies. Based on this study, we were able to identify common high-level concepts of all security ontologies, and thereafter to develop a generic view on all security ontologies (at least, those covered by our literature review). Our "upper ontology", (Figure 2), shows the high-level concepts and corresponding relations of the ontology. A threat gives rise to follow-up threats, represents a potential danger to the organization's assets and affects specific security goals as soon as it exploits a vulnerability in the form of a physical, technical, or administrative weakness, and it causes damage to certain assets. For each vulnerability, the asset on which the vulnerability could be exploited is assigned. Controls have to be implemented to mitigate an identified vulnerability and to protect the respective assets. Each control is implemented as an asset concept, or as combinations thereof.

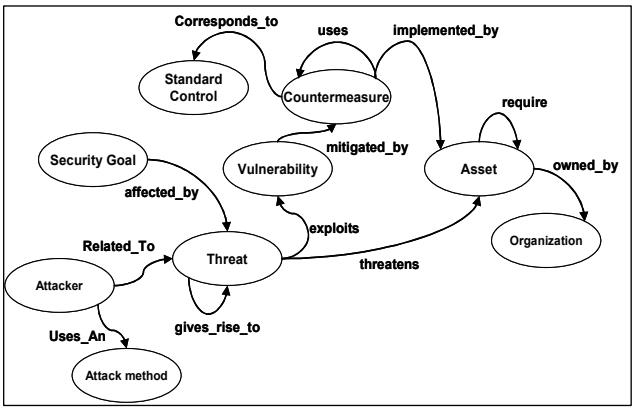

Fig. 2. Upper security ontology adapted from [8].

The notion of domain for software development has been discussed in [9]. Bjoner defends the need for domain expertise for requirements. He defines a domain as an "area" of natural or human activity, or both. In literature, we find a large number of domain ontologies (health, railways industry or container line industry domain [14]). To be independent of any pre-selected domain ontology, we developed a generic view on domain ontologies. The "Multi-level Domain Ontology", where main concepts represented in Figure 3, relies mainly on previous studies presented by [10][9]. The upper domain ontology represents a domain according to two main concepts: (a) Entities to which a domain refers to (such as vessel, container, person, ect.) and (b) their properties (length, location, haveParents, etc.).

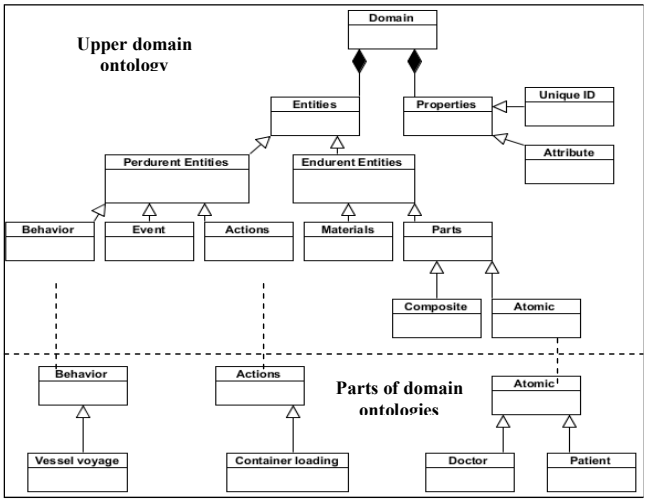

Fig. 3. Multi-Level Domain Ontology

III. PROPOSITION

Our method guides the discovery of SR for a specific domain. This process handled by a series of heuristic production rules that, starting from high level SR, produce a SR specification. Figure 4 shows an overview of our method. There are two sub-sets of rules. The first set of rules handles domain-specific analysis. The second set of rules performs a security specific analysis. Each set of rules exploits different ontologies: respectively domain ontologies and security ontologies. In order to be able to handle different security and domain ontologies, the rules were specified with so-called "upper ontologies", that handle concepts that are (a) common to most ontologies, (b) sufficiently high level to abstract many other concepts in the specific ontologies, and (c) more importantly that represent an important subject of interest for the method. During the elicitation step, an initial I* requirements model is first constructed from the stakeholders' needs and concerns expressed about security at the beginning of the project. At this stage, the analyst defines initial actors, resources, and especially security goals (integrity, confidentiality, traceability...) During the SR analysis stage, the production rules (the red arrows in figure 4) will exploit the security-specific ontology to discover threats, vulnerabilities, countermeasures, and resources, and thus enrich the requirements model by adding new elements (malicious tasks, vulnerability points...).

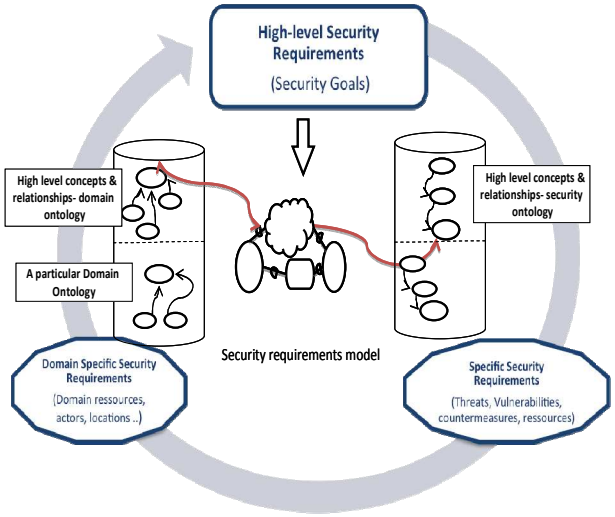

Fig. 4. Method Overview

During the domain specific SR analysis stage, another set of rules explores the domain ontology to improve the requirements model with resources, actors and other concepts that are more specific to the domain at hand; for instance: thieves in the banking domain, hijackers in the aeronautic domain, pirates in the maritime domain, etc.

A . Aligning I with Security and domin ontology*

To build the bridge between the security requirement model constructs, domain, and security ontology, we analysed semantics of concepts (C) and relationships (R). We have resulted in the semantic alignment illustrated in Table 1.

TABLE I. SEMANTIC ALIGNEMENT, REQUIREMENT MODEL, SECURITY ONTOLOGY, DOMAIN ONTOLOGY - (**C** FOR CONCEPT, **R** FOR RELATION)

	I* Requirement Model	Security Ontology	Domain Ontology
C	Softgoal	Security Goal	-
C	Task	Threat	Event
R	Contribution Link (Task,Softgoal)	AffectedBy (SecurityGoal,Threat)	
C	Vulnerable Point	Vulnerability	-
C	Task	Countermeasure	Action
R	ContributionLink (Softgoal, Task)	MitigatedBy(Vulnerability, Countermeasure)	
C	Ressource	Asset	Atomic/ Composite Entity
R	DecompositionLink (Task, Ressource)	ImplementedBy(Count., Asset)	

A Security Goal in the security ontology, is alligned with a softgoal in the requirement model. A threat, is aligned with an event in the domain ontology, and with a task (malicious task) in the requirement model. A vulnerability is aligned to the concept vulnerable point in I*. An asset is aligned either as a atomic or composite entity in a domain. Thanks to the concept resource in I*, the asset from the ontology is characterised in the requirement model.

B. Security requirements elicitation and analysis rules

Let's consider the example of a stakeholder from the maritime domain who expresses a need to ensure the integrity of crew's personal files: "The captain wants to ensure the integrity of it's crews personal files"--> {softgoal: files integrity, actor: Captain, Crew}.This example extracted from our evaluations, is run in the next step.

Each rule is 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 conclusion C can be drawn. The situation holds on an input model and input ontology. Situation is defined using procedural semantics that relies on two kinds of functions:

- EquivalentClass (X, Y) : is true if X in the input model has the same semantics (meaning) as the concept Y in the input ontologies.

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

Conclusions indicate elements that should be added to the output model. They are specified with three kinds of functions:

- CreateClass(ClassC, X): indicate that a concept X that instantiates the <ClassC> class can be created in the model.

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

- ReplaceClass(ClassC, X): Class C belongs to model, X to an ontology, indicates that the ClassC will be replaced by the concept X, this function is used to get more precision from the domain ontology.

The next section represent some of the security-specific and the domain specific analysis rules: (security goal identification, threat identification/ domain threat identification, vulnerability identification, countermeasure identification, resource identification/ domain resource identification). The input of this rules is the requirements model (R) constructed in the initial phase of the process, the security ontology (SO), and the domain ontology (DO).

– *Security goal identification (rule 1):* Each soft goal (softgoal) in the input requirements model is mapped to security goals in the ontology (secgoal), using the EquivalentClass function. The equivalence is based on a lexical and synonymy mapping using a thesaurus such as Wordnet. If an equivalence is found, then the softgoal is considered as security goal and therefore, it can be specified as such in the SR specification.

Rule 1: *EquivalentClass(softgoal, secgoal)*

Example: EquivalentClass (files integrity, integrity)

- *Threat identification (rule 2):* The security ontology indicates that each goal can be related to a threat by the relation affectedBy that indicates that the threat may be a risk causing the SecGoal failure (ie. at least during a given period, the goal is not maintained). This relation is explored throughout the input requirements model and the security ontology as follows. A threat can be identified when a task (malicious task) is performed by an agent that is an attacker. If the situation is met, then the threatening task is created in the security model, and linked by a contribution link (break) to the softgoal. The reasoning behind is that the threat in the ontology becomes a task performed by an attacker (any one of the actors can be a potential attacker [4]), which breaks the intended security goal.

Rule 2:

$$\forall \text{Threat} \in \text{SO}, \exists \text{SecGoal} \in \text{SO}, \exists \text{SoftGoal} \in \text{R}$$
$$\text{OntologyLink}(\text{IsAffectedBy}, \text{SecGoal}, \text{Thrat}) \wedge$$
$$\text{EquivalentClass}(\text{SecGoal}, \text{SoftGoal})$$
$$\rightarrow \text{Threat} \in \text{R},$$
$$\text{CreateClass}(\text{Task}, \text{Threat})$$
$$\text{CreateLink}(\text{ContributionLink}, \text{Threat}, \text{Softgoal})$$

Example:

OntologyLink (IsAffectedBy, Intigrity, Trigger Fire)

& EquivalentClass (Files Integrity, Integrity)

--> CreateClass (Task, Trigger Fire)

CreateLink(ContributionLink, Trigger Fire, files Integrity)

- *Domain Threat identification (rule 3):* Using the domaine ontology we can get more precision concerning a given threat. Because threats might differ from a domain to another. A domain threat can be identified throgh domain events.

Rule 3:

$Task \in$ R, event $\in$ DO

If EquivalentClass(Task, event)

ReplaceClass (Task, event)

Example:

ReplaceClass (Trigger Fire, overheating bearing Fire)

- *Vulnerabilitiy identification (rule 4):* In this rule, each threat that is related by the relation *exploits* to a vulnerability (Vul) is looked for throughout the security ontology. A vulnerability is a weakness that allows an attacker to reduce a system security. The rule explores this information and adds it to the requirement model (R) when the security ontology reveals that there is a potential vulnerability in the system under the form of a threat that typically impacts a vulnerability to which one of the softgoals in the input model can be assimilated.

Rule 4:

$$\forall Contermeasure \in SO,$$
$$\exists (Vul, Threat, SecGoal) \in SO, \exists SoftGoal \in R$$
$$Onto\log yLink(Exploits, Threat, Vul) \wedge$$
$$Onto\log yLink(IsAffectedBy, SecGoal, Threat)$$
$$EquivalentClass(SecGoal, SoftGoal)$$
$$\rightarrow Vul \in R$$
$$CreateLink(Vu\ln erablePo\text{int}, Vul, SoftGoal)$$

Example:

OntologyLink(Exploits, Trigger Fire, improper storage of combustible material)
& OntologyLink(IsAffectedBy, Integrity, Trigger Fire)
& EquivalentClass (Integrity, files integrity)

--> VulnerablePoint (improper storage of combustible material, files integrity)

- *Countermeasure identification (rule 5):* This rule searches for the controls (countermeasures) that can be used to handle these threats. Countermeasures are looked for in the input security ontology throughout the MitigatedBy relationships that are associated with a countermeasure corresponding to standard control. Controls such as “training of maintenance and administrative staff”, or standard controls such as “ISO 27001 control” are added as tasks to the requirements model, and linked by means of an end link to the softgoal.

Rule 5:

$\forall Countermeasure \in SO, \exists (Vul, Threat, SecGoal) \in SO,$

$\exists SoftGoal \in R$

$OntologyLink(IsAffectedBy, SecGoal, Threat)$

$\wedge\ OntologyLink(Exploits, Threat, Vul)$

$\wedge\ OntologyLink(MitigatedBy, Vul, Countermeasure)$

$\wedge\ EquivalentClass(SecGoal, SoftGoal)$

$\rightarrow Countermeasure \in R,$

$CreateTask(Countermeasure)$

$CreateLink(ContributionLink, Countermeasure, SoftGoal).$

Example:

OntologyLink (MitigatedBy, improper storage of combustible material, Perform fire protection control*)*

OntologyLink (Exploits, Trigger Fire, improper storage of combustible material)

OntologyLink (IsAffectedBy ,Integrity, Trigger Fire)

*EquivalentClass (*files integrity, integrity*)*

--> Task (Perform fire Protection Control)

ContributionLink(Perform fire protection, files integrity).

- *Resource identification (rule 6):* According to the security ontology a control needs an asset to be implemented. This is the case when a relation implementedBy appears in the ontology between the control and the asset. The rule scans in the ontology in order to find assets that are used for implementation of a control such as access control systems, safety doors, or virus protections. When they are found, they are introduced in the SR specification under the form of resources, then linked by a decomposition link to the previously added task.

Rule 6:

$\forall Asset \in SO, \exists (Countermeasure, Vul, Threat, SecGoal) \in SO,$

$\exists SoftGoal \in R$

$OntologyLink(AffectedBy, SecGoal, Threat)$

$\wedge\ OntologyLink(Exploits, Threat, Vul)$

$\wedge\ OntologyLink(MitigagtedBy, Vul, Countermeasure)$

$\wedge\ OntologyLink(implementedBy, Countermeasure, Asset) \wedge$

$EquivalentClass(SecGoal, SoftGoal)$

$\rightarrow Asset \in R,$

$CreateClass(resource, Asset)$

$CreateLink(DecompositionLink, Task, Asset).$

Example :

OntologyLink (ImplementedBy, Perform Fire protection control, Fire extinguisher)

*OntologyLink (MitigatedBy (*Improper storage of combustible material, Perform Fire protection control*)*

OntologyLink (Exploits ,Trigger Fire, Improper storage of combustible material)

OntologyLink (IsAffectedBy, Integrity, Trigger Fire)

*EquivalentClass (*files integrity , integrity*)*

--> Resource (Fire Extinguisher)

DecompositionLink (Perform fire protection control, Fire extinguisher)

- Domain ressource identification (Rule7):

The input models contain many information about resources, some of which are security-independent, others of which are more relevant in the specification of the SR. On the other hand, the domain ontology indicates which asset is typically used in the domain to handle security issues, for example “encryption filter”, “antivirus”, “fire extinguisher”, etc. depending on the domain. The domain ontology (DO) is thus explored to look for entities that can be assimilated to resources identified in the input model. Each resource in the input requirement model is mapped to the corresponding asset in the domain ontology using the EquivalentClass function.

Rule 7:

EquivalentClass(resource, Entity), resource $\in$ R , entity$\in$ DO

for each Entity'$\in$ D*O*

if is - a (Entity', Entity)

$\rightarrow$ *Entity*'$\in$ *R*,

CreateClass(Resource , Entity'),

CreateLink(DecompositionLink , resource, Entity')

endif

endfor

Example:

EquivalentClass(fire extinguisher, ship's use fire extinguishers)

is-a (FM-200 gaseous based fire extinguisher system, ship's use fire extinguishers)

--> Ressource (FM-200 gaseous based fire extinguisher system)

--> DecompositionLink (Fire extinguisher, FM-200 gaseous based fire extinguisher system)

is-a (FlexiFOG - Low pressure water mist fire generator, ship's use fire extinguishers)

--> Ressource (FlexiFOG - Low pressure water mist fire generator)

--> DecompositionLink (Fire extinguisher, FlexiFOG - Low pressure water mist fire generator)

is-a (FlexiFOAM- New generation of high expansion foal generator, ship's use fire extinguishers)

--> Ressource (FlexiFOAM- New generation of high expansion foal generator)

--> DecompositionLink (Fire extinguisher, FlexiFOAM- New generation of high expansion foal generator)

The security specific rules and the domain specific rules can be applied iteratively until one believes all the possible artifacts have been added into the requirements specification. The requirements identification process can be considered as complete when no new security goal can be added in the specification. The SR specification can be considered complete when no new security threat or asset can be added.

IV. EVALUATION

The intended goal of the method and the whole project is to improve the definition of SR (efficiency and effectiveness). However at this stage of progress and in this

paper, we describe the validation of the feasibility and advantages (perceived usefulness, intention to use, etc) of the method with experts (qualitative validation), based on Moody's method evaluation model [11]. Further validation stages (quantitative validation with a tool and practitioners) is part of future research.

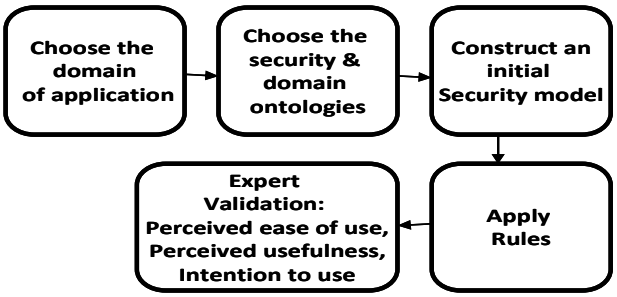

Fig. 5. Evaluation process.

The evaluation process that we undertook is shown in Figure 5. We chose a domain of application, namely the maritime domain, for which we selected a domain ontology. Based on a first interview of the maritime domain expert, we constructed an initial SR model. This initial security model mostly specified the main agents and goal dependencies between the involved in a merchandise ship company. The validation process was then carried by an interview of four experts: the domain expert was a ship's captain in the civil navy. The requirements engineering expert was a scientist well known in the requirements engineering community. Two security standardization experts served as security experts. Each interview was performed face to face with the expert, and took the form of a review of the I* model produced by application of our method. The experts also challenged the method and the rules with regards to their knowledge and experience in their respective areas of expertise (requirements, security and maritime). First, the context of the project and the method were presented to them. Second, the SR maritime domain model before and after applying the method. Based on Moody's evaluation framework [16], we prepared a template (measurement variables and their corresponding questions) to evaluate the method, and the requirements model according to different aspects that were embodied into variables. Note that some experts were more relevant to certain aspects than others. For example the maritime domain expert was more relevant to validating the domain concepts used than the security experts. Three main variables were used to construction interview. With each variable a couple of questions were asked to experts: **Perceived ease of use**: the degree to which the expert believes that using this method would be free of effort. (Q. I found the procedure for applying the method complex and difficult to follow) / P**erceived Usefulness**: the degree to which the expert believes that this method will be effective in achieving its objectives (Q.I believe that this method would reduce the effort required to build SR models for specific domains) / **Intention to Use:** the extent to which the expert intends to use our method. (Q. I would definitely not use this method to define SR; Q. I intend to use this method in preference to the existing SR definition methods for a specific domain.)

All the comments, answers of experts were analyzed and discussed. The time spent for each interview was different from one person to the other, ranging from half an hour to several hours of presentation and discussion, stretching over several days by mail. Four interviews are unfortunately not enough to conduct a statistical analysis of the interview results. We can however summarize the salient results of the interviews. (a) First, the main recurrent comments made by the experts were that a) the method and its idea of automation is "very" (the adjective was employed several times) "useful", "interesting", and "important". (b) Using the method based on domain in addition to security ontologies to discover new SR for a specific domain makes a real difference with respect to discovering requirements using only security ontologies.(c) Although the method is interesting, using I* as a SR specification language was found limited, because there are no attack scenarios. This issue could be handled either by extending the notation, or by completing the I* model with another more concrete kind of model such as misuse cases. (d) The concept of softgoal used in the input model and in the I* models specifying the SR was not found adequate for precisely specifying the SR.

V. THREATS TO VALIDITY

This first step of evaluation responded to our needs at this phase of the project. However, there are some threats to validity that need to be addressed. This validation was carried with four experts, a very small group to draw a conclusion about the method. Recall that the same group was along the experience, they saw the initial model and the new one (after the use of ontologies), and this is another threat to conclusion. . The evaluation was carried with the maritime domain and a chosen domain ontology, we still need to see how does this works with another domain ontology. The qualitative validation we carried, even if its considerable, remains very subjective.

VI. RELATED WORK

Due to the increasing interest about security at initial steps of development (SRE) on the one hand and the interest in ontology-based approaches on the other hand, many publications have been proposed. [12] discussed ontology-based RE. However, their proposition focuses only on the construction of the ontology and not on its use. [13], [14], [15] and [16] proposed some reasoning methods to use ontologies for eliciting and analyzing requirements. However their interest was in functional requirements, far from non-functional, SR. We (in [2]) proposed the RITA method, using a threat ontology to define SR. The main limitation noticed is that the resulting requirements were generic, not oriented to a specific domain, which was due to the generic nature of the used ontology. Same as [20], whose frame work doesn't introduce domain in the SR analysis,. Some recent approaches like [17] who used a threat and defence ontologies to specify requirements, the main difference remains in that their proposition was related to textual SR whereas ours deals with security requirement models, we

also exploited here more the domain knowledge then them. Dritsas et al. [18] and [19] worked on SR for e-health applications and parsive healthcare respectively, although their contributions were interesting they were restricted to one domain, we've tried through our upper domain ontology to propose a method which is not restricted to one domain but can be applied with different domains.

VII. CONCLUSION AND FUTURE WORKS

In this paper we join our effort to the early SRE community, by proposing a requirements definition method, based on ontologies. The method does not rely on a specific security or domain ontology since we relied on a so-called "upper" ontology for both security and domain. It is based on reasoning rules to discover and adds new elements for the SR model. The method makes it possible to define SR related to a specific domain. This work is an ongoing project; our next step is to extend its application to other languages such as [7]. We also would like to extract more domain knowledge (like responsibilities, roles, locations…), and to adapt risk based and attacks analysis. Experts enabled a preliminary validation but a green light to continue the project, the evaluation process still needs to be more refined with quantitative validation with a focus group through a prototype which is under construction.

REFERENCES

[1] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, 2010.

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

[3] M. Saeki, « Semantic Requirements Engineering », in *Intentional Perspectives on Information Systems Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 67‑82.

[4] L. Liu, E. Yu, et J. Mylopoulos, « Security and privacy requirements analysis within a social setting », in *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International*, 2003.

[5] A. Hevner et S. Chatterjee, *Design Research in Information Systems: Theory and Practice*, 1re éd. Springer, 2010.

[6] A. Souag, C. Salinesi, et I. Comyn-Wattiau, « Ontologies for Security Requirements: A Literature Survey and Classification », in *Advanced Information Systems Engineering Workshops*, vol. 112, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[7] H. Mouratidis, (2006) 'Analysing Security Requirements of Information Systems using Tropos', Proceedings 1st Annual Conference on Advances in Computing and Technology (AC&T). London - UK

[8] S. Fenz et A. Ekelhart, « Formalizing information security knowledge », in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, New York, NY, USA, 2009.

[9] D. Bjørner, « Rôle of Domain Engineering in Software Development—Why Current Requirements Engineering Is Flawed□! », in *Perspectives of Systems Informatics*, A. Pnueli, I. Virbitskaite, et A. Voronkov, Éd. Springer Berlin Heidelberg, 2010, p. 2‑34.

[10] D. Bjørner, « Domain Engineering », in *In BCS FACS Seminars, Lecture Notes in Computer Science, the BCS FAC Series (eds. Paul Boca and Jonathan Bowen*, 2008.

[11] D. Moody, « The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods », *ECIS 2003 Proceedings*, janv. 2003.

[12] G. Dobson et P. Sawyer, *Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web. In: Dependable Requirements Engineering of Computerised Systems at NPPs*. 2006.

[13] M. Shibaoka, H. Kaiya, et M. Saeki, « GOORE□: Goal-Oriented and Ontology Driven Requirements Elicitation Method », in *Advances in Conceptual Modeling – Foundations and Applications*, vol. 4802, Éd. Springer Berlin / Heidelberg, 2007, p. 225‑234.

[14] H. Kaiya et M. Saeki, « Using Domain Ontology as Domain Knowledge for Requirements Elicitation », in *RE, 14th IEEE International Conference*, 2006.

[15] D. V. Dzung et A. Ohnishi, « Ontology-Based Reasoning in Requirements Elicitation », in *Software Engineering and Formal Methods, 2009 Seventh IEEE International Conference on*, 2009, p. 263 ‑272.

[16] L. Wei, H. Ke-Qing, W. Jiang, et P. Rong, « Heavyweight Semantic Inducement for Requirement Elicitation and Analysis », in *Third International Conference on Semantics, Knowledge and Grid*, 2007

[17] O. Daramola, G. Sindre, et T. Moser, « Ontology-Based Support for Security Requirements Specification Process », in *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, P. Herrero, H. Panetto, R. Meersman, et T. Dillon, Éd. Springer Berlin Heidelberg, 2012, p. 194-206.

[18] S. Dritsas, , L. Gymnopoulos, M. Karyda, T. Balopoulos, S. Kokolakis, C. Lambrinoudakis, and S. Katsikas. "A knowledge-based approach to security requirements for e-health applications."

[19] Koay, Nigel, Pavandeep Kataria, Radmila Juric, Patricia Oberndorf, and Gabor Terstyanszky. "Ontological support for managing non-functional requirements in pervasive healthcare." In System Sciences, 2009. HICSS'09.

[20] Lasheras, Joaquín, Rafael Valencia-García, Jesualdo Tomás Fernández-Breis, and Ambrosio Toval. "Modelling reusable security requirements based on an ontology framework." Journal of Research and Practice in Information Technology 41, no. 2 (2009): 119.