2010 Fifth International Conference on Internet and Web Applications and Services

A Solution Model and Tool for Supporting the Negotiation of Security Decisions in E-Business Collaborations

Jason R. C. Nurse and Jane E. Sinclair
University of Warwick, Coventry, CV4 7AL, UK
{*jnurse, jane.sinclair*}*@dcs.warwick.ac.uk*

***Abstract*—Sharing, comparing and negotiating security-related actions and requirements across businesses has always been a complicated matter. Issues arise due to semantic gaps, disparity in security documentation and formats, and incomplete security-related information during negotiations, to say the least. As collaborations amongst e-businesses in particular increase, there is a growing, implicit need to address these issues and ease companies' deliberations on security. Our research has investigated this topic in substantial detail, and in this paper we present a novel solution model and tool for supporting businesses through these tasks. Initial evaluation results and feedback from interviewed security professionals affirm the use and suitability of our proposals in supporting the security actions negotiation process.**

***Keywords*-Security negotiations; business-oriented framework; e-business collaborations; security ontology; XML security language; support tool**

I. Introduction

Inter-organizational e-business, endorsed by a wide suite of enabling technologies (e.g., Web services, ebXML, RosettaNet), is now one of the most promising and lucrative business paradigms. To sustain these online interactions, security researchers and professionals have investigated numerous technologies, processes and best practices. In previous work we have also contributed to this area by defining the Business-Oriented Framework for enhancing Web Services Security for e-business (BOF4WSS) [1]. BOF4WSS' uniqueness stems from its emphasis on a detailed cross-enterprise development methodology, to aid collaborating e-businesses in jointly creating secure and trusted interactions. This particularly refers to the creation of a multilayered security solution, which encompasses technologies, processes, policies and strategies, and spans the interacting companies.

Further to the comprehensive guidance supplied by BOF4WSS, our research has explored the provision of a range of useful support systems. These would assist in the framework's application to business scenarios, and seek to streamline various essential, but often arduous or problematic development tasks (e.g., see Section II). The presentation and evaluation of one of these novel support systems and its underlying conceptual solution model, form the primary contributions of this paper.

The specific problem area of interest in this work, concerns the difficulties in negotiating security actions and requirements across companies; a prerequisite activity before the joint systems are developed. Here, a *security action* is defined as any high-level way in which a company handles a risk it faces (e.g., 'the risk of ensuring the security of a server is to be outsourced'), whereas a *security requirement* is a high-to-medium level desire, expressed to mitigate a risk (e.g., 'the integrity of personal data must be maintained'). Security actions thus encompassing security requirements.

The problem area highlighted above, relates to the hardships incurred when transitioning from the individually completed Requirements Elicitation stage, to the subsequent Negotiations stage in BOF4WSS, where companies meet to present, negotiate and reconcile their security actions/requirements. As shown in forthcoming sections, problems faced include (i) understanding other companies' security documentation, (ii) understanding the motivation behind security actions/requirements, (iii) being able to easily match and compare security actions from businesses which target the same situation, and (iv) gathering and assimilating relevant aspects motivating security actions when reconciling security to apply to the foreseen business scenario.

Having assessed the difficulties of stage transition, the next aim is to present the novel Solution Model, which forms the conceptual foundation for the support system tool to address these complications. The implemented software tool is also introduced. With the system presented, we then report on the evaluation to date, including highlighting the benefits and shortcomings of the tool and underlying model.

This paper is organized as follows. Section II examines the difficulties faced as businesses using BOF4WSS move between the stages mentioned above. With that context set, Section III outlines the Solution Model. Next, Section IV gives an overview of a prototype system actually developed. This is followed by a discourse on the evaluation and its findings in Section V, before presenting the related work in Section VI. Conclusions are presented in Section VII.

II. The Stage Transition Problem

Sharing, comparing and negotiating on security actions and requirements across companies, even at a high-level, has always been a complex matter. Tiller's work ([2]) gives insight into this issue as he labels the related process,

978-0-7695-4022-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIW.2010.10

"security mayhem", because of the variety of security aspects (e.g., specific polices, service-level agreements, legal obligations, unique access requirements) to be considered in forming business collaborations. The reality of this problem is underlined by Dynes et al. [3] who set out a research agenda with a core question being: how can a shared vision on risks and security for interacting companies be achieved which appreciates their range of differences?

To investigate the specific issues surrounding stage transition and the negotiation of security actions as they pertain to BOF4WSS, a case scenario was used. This scenario featured companies using the framework during the Requirements Elicitation and Negotiations stages, and especially focused on how security actions were determined, how these actions were documented/expressed, and how parties compared and negotiated on them. To strengthen the practicality of the scenario, security professionals knowledgeable in external company interactions were interviewed and their input used to guide case development. After defining the case scenario, it was analyzed to identify areas which proved difficult, problematic, or overly tedious for companies. Some of the most prominent areas are discussed below.

- **Understanding the security actions documents of the other companies "as is"**: In the Negotiations phase, companies supply their security actions to their business partners for perusal and discussion. A major difficulty even at this early stage was gaining an appreciation of what exactly companies meant (i.e., a semantic issue) when they outlined a security action or requirement in a few brief, informal statements, often with little justification. Included in this, is the reality that companies may use different terminologies for security actions, associated risks, threats, and vulnerabilities. These problems were further compounded by the variety of techniques (e.g., requirement listings, generic checklists, graphical representations) used by businesses to document their security actions. The core issues at this point therefore link to the *semantic gap* likely to be prevalent across companies, and the *disparity in formats* used to document actions. Both of these aspects resulted in the need for companies to spend considerable time and effort understanding actions and requirements before any negotiations could take place.
- **Understanding the motivation behind other companies' security actions and requirements**: From the summary documentation which constituted companies' security actions and requirements, it was often somewhat challenging for other businesses to determine exactly why that security desire existed. Even if the security situation/risk which the security action intended to address was included in the description, there might have been a plethora of other aspects (e.g., laws and regulations, security policies) considered in the preceding risk assessment that were not specified in the action description. These aspects are important because they provide insight into security actions that form the basis for companies negotiations. As a result of this *incomplete information*, companies usually had to enter further discussions to determine these aspects before making decisions on individual security actions.
- **Comparison of companies' security actions and requirements**: This task entailed parsing through other companies' actions and requirements documents to note and question any existing conflicts across businesses. Included in this task was the implicit or explicit matching of security actions from companies which targeted the same situation or risk. Even in the cases where security actions were classified into groups beforehand, the task of *parsing through documents*, and the various *back-and-forth communications* necessary to match and compare actions even at a basic level, resulted in the consumption of a vast amount of man-hours. An additional issue at this point was ensuring that all aspects motivating security actions (e.g., laws, security policies, contractual obligations) were gathered, documented and readily available for consideration, to support actual comparison and negotiations. Any streamlining of the aforementioned processes would save time, money, and effort for parties.

Having presented some of the core problems discovered from the case analysis, Section III outlines the conceptual Solution Model for the system to support stage transition.

III. Solution Model

The Solution Model, shown in Figure 1, contains four components: Security Actions Analysis, Ontology Design, Language Definition and Risk Catalogue Creation. The prime aim of this model is to outline a notional base on which a tool that would actually support the negotiation of security actions across companies, could be implemented. A description of the components is given below.

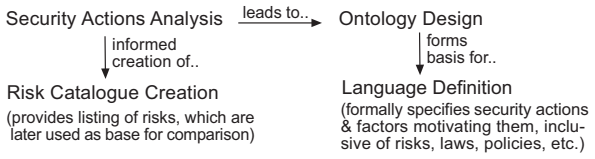

Figure 1. Solution Model

Security Actions Analysis: As a first step to addressing the problems related to the semantic gap and the disparity in formats used to document actions (identified in Section II), an in-depth analysis of the security actions and requirements domain was required. This assessment focused on security literature particularly in the security risk management field (as this area was viewed as key to determining security actions), and critically examined how security actions and requirements were derived. From that analysis, common critical factors, especially those that constituted and motivated their derivation were then identified. This component

stage's findings allowed for a thorough understanding of that domain, and furnished the foundation for following stages.

Ontology Design: Ontologies are widely known for their ability to specify a shared understanding about a particular domain. In this case, an ontology was used to provide a common understanding of the security actions (and generally, security risk management) domain, based on findings from the Security Actions Analysis stage. Establishing this common semantic bridge was a critical prerequisite in creating the overall solution, when considering how different the terminologies, methods, and influential factors internal to each business were likely to be. It was also critical that the ontology was encompassing, and therefore allowed for an easy semantic mapping of concepts onto it from typical security action determination (or simply, security risk management) methods used by companies. Readers should note that the ontology designed here is high-level and mainly diagrammatic (i.e., there is no formal ontology language). As such, it is more of a communications tool, which can also be built on in future components. An ontology draft, and the Analysis component were previously presented in [4].

Language Definition: Two of the core issues identified in Section II center around the numerous formats used for security actions, and the incomplete information initially presented regarding the motivation for those actions. The Language Definition stage addressed these issues by defining a formal language to be used by companies at the end of Requirements Elicitation. The benefit of a formal language as opposed to a shared text-based template, or graphical representation is the automation it would allow; encoded data could now be processed by a machine. This language would enable the formal expression of parties' security actions, and the factors that motivated them (e.g., risks, laws, security policies and so on) in a common format. By having these motivational factors initially included and specified, this negates the need to enter lengthy discussions to determine these aspects later. An XML-based language was preferred to facilitate encoding due to its wide acceptance, XML's platform independence, and the variety of systems support options (numerous APIs for parsing and validation) available. To define the language's syntax, the ontology was an invaluable asset. Aiding in language definition was one of the original purposes of the ontology, as its use ensured that the language was grounded in accepted literature and supported by some common semantics across companies.

Risk Catalogue Creation: To address the problem of matching and comparing security actions across enterprises, emphasis was placed on identifying an aspect which was common to the actions and could be held constant. Therefore, regardless of the divergent security actions for a situation defined by businesses, a common underlying aspect could be used to quickly (or automatically) match these actions. After reviewing the Security Actions Analysis, it was apparent that in a majority of cases, security actions were established to handle or treat some inherent risk. The range of security action determination methods used by companies enforced this reality (see work in [4]). To provide the constant base therefore, a shared risks listing/catalogue was instituted and developed. This catalogue contained an updatable, extensive listing of security risks, and was used by companies as a common input to their risk management processes (i.e. the process that identifies, analyzes, evaluates, and decides treatment for the risks). Although businesses used different processes and derived possibly disparate security actions, they maintained a common base in terms of what risks were addressed by a particular action. Once implemented in a system, this common base would allow for the automated matching of security actions from companies, and thus ease the task of matching and comparing actions.

A general idea of how the implemented Solution Model worked towards significantly easing stage transition, is illustrated in Figure 2. In this diagram `Supplier` and `Buyer` are using BOF4WSS for an online business scenario.

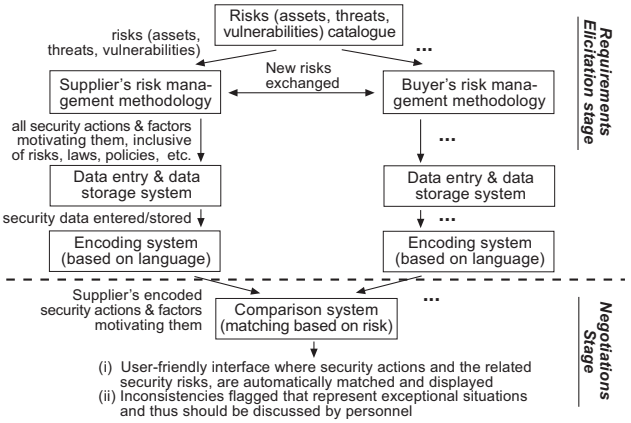

Figure 2. Solution Model in action

A briefly outline is now given on the conceptually implemented model in Figure 2. To begin, risks from the risk catalogue are selected by companies to form input to each entity's risk management methodology (i.e., process to determine security actions and requirements). Once companies determine their individual security actions, these actions and the factors motivating them are transferred into an Encoding system and marked up into the XML-based language defined. When businesses meet in BOF4WSS' Negotiations stage, the encoded documents are then passed to a Comparison system that matches companies' security actions based on the underlying risks they address. Currently, the output of the Comparison system focuses on (i) a user-friendly interface where security actions (supported by related risks, and motivational factors) are automatically matched and displayed, and (ii) flagging of any inconsistencies identified for follow-up by personnel. A noteworthy point is that the Solution Model and resulting tool are

especially geared towards *shared* risks faced by entities. Therefore in some regards, emphasis is placed on the shared risks where companies have to agree on how they will be treated i.e., the type of security action (e.g., mitigation, transference, acceptance, avoidance), and actual action to apply. Section IV formally introduces the tool which embodies the Encoding and Comparison systems above. This is the Security Actions Specification and Comparison System, hereafter SASaCS.

IV. SASACS TOOL

A. Overview

The SASaCS tool represents the culmination of this work, in that, it is the software implementation of the Solution Model. SASaCS consists of all the practical components necessary to support the presentation, sharing, comparison and negotiation of security actions across companies. As a result of its tight coupling with the Solution Model, the general process outlined at the end of Section III applies to the tool as well. In Section IV therefore, we provide more detail on the tool by discussing two of its features, the Data Entry interface and the Encoding system. These aspects were chosen because they allow novel parts of SASaCS to be highlighted, and set the platform for evaluation in Section V.

Once companies have conducted their risk management activities (which are informed initially to some degree, by the shared risk catalogue) and produced their individual security actions, the next task is transferring them into (their locally installed copy of) the SASaCS tool. This is handled by the Data entry and storage system. This system, shown in Figure 2, provides a set of simple, intuitive screens for users to input their security related data (e.g., risks, security actions and factors motivating them) and have it stored to a back-end tool database. To ease usability, the tool also allows the direct referencing and selection of risks from the risk catalogue, that initially factored into the company's risk management activities. Therefore, users can look-up risks from the catalogue, apply them to the current project/collaboration, and then annotate them, or otherwise use them as they see fit (e.g., input' risk priority levels, associate them with a security action, and so on).

As SASaCS is based on the ontology designed, its data entry screens benefit from the unambiguous definition of concepts (such as risk, risk level, and so on) prevalent with the ontology. The ontology diagram itself and its documentation also are useful in assisting users understanding of concepts, and linking data entry fields to output from their risk management methodologies. In addition to having data fields mirroring the basic concepts from the ontology, the Data entry interface defines a number of other fields to allow companies to add more detail on relevant aspects such as company-specific risk descriptions, justifications of risk levels, annotations regarding treatments of risks, treatment coverage levels, and security requirements. Figure 3 shows a screenshot of the security action (or in other terms, risk treatment action) data entry screen in SASaCS.

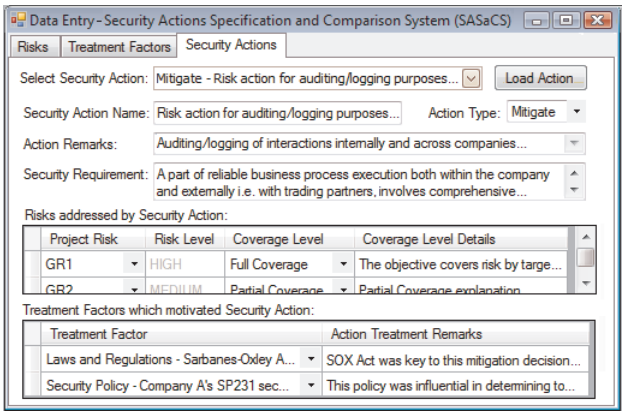

Figure 3. Security action data entry screenshot

After each enterprise has saved their security- and risk-related data to the tool, the following step is encoding that data in preparation for inter-company negotiations. The Encoding system (also installed locally) facilitates this by pulling data from the tool database, marking it up in the XML-based language discussed previously, and outputting a document with the encoded data. When companies meet for negotiations therefore, (i) they use the same format to express security actions/requirements, which is also machine-processable; (ii) there is a shared understanding of the security- and risk-related concepts, promoted by the common ontology and highly supportive tool data entry screens; (iii) information is more complete as factors motivating security actions should initially have been supplied; and (iv) because encoded data (particularly security actions) includes references to risks in the risk catalogue, there are commonalities across companies' documents. The Comparison system uses these commonalities to automatically match security actions/requirements that treat the same shared risks.

As an example of (iv) above, let us assume three companies, A, B, and C. According to information provided in their XML-based documents, we see that A defines a security action to address a risk related to the confidentiality of information (hereafter, $RiskX$). In B's document, they also have a security action stated to handle that risk. C however, has decided not to consider or address $RiskX$, and therefore it is not included in their document. By having all this information supplied in the XML-based documents at the time companies meet for negotiations, SASaCS can be used to assist in a number of important tasks. One such task is automatically matching the disparate security actions of A and B based on their mutual goal towards handling $RiskX$. Another feature is its ability to instantly discover that C is not addressing the shared risk. This could then be flagged for follow-up by personnel. Streamlining these, at times simple tasks, can significantly reduce the time and effort

needed by companies during the initial stages of BOF4WSS negotiations. In the next section, we examine the encoding aspect more by presenting the XML-based language defined. For ease of reference, this language is called SADML, or Security Actions Definition Markup Language.

B. The Language

The structure of SADML was conceived to mirror the knowledge captured in the ontology (largely defined in [4]). As such, various ontology's concepts are represented as XML elements/tags. To comply with XML's hierarchical nature, it was necessary to define a sensible hierarchy of elements. Furthermore, this structure would need to accommodate one-to-many relationships across elements (for example, if a law motivates/supports multiple security actions, this should be appreciated). Considering these and a few other aspects, SADML's syntax was defined. A snippet of the SADML format representing the information in Figure 3 is presented below; the + sign indicates additional data which is not displayed here. The core language is described in the schema, indicated by `urn:risksx-schema` in the snippet.

```
 <needsBase xmlns="urn:risksx-schema" ... >
   <mitigationActions>
    <mitigationAction>
     <name>Risk action for auditing/logging...</name>
     <details>Auditing/logging of interactions...</details>
+    <risks>
     <lawAndRegRefs><lawAndRegRef idref="LR22">
       <relationToRiskAction>SOX Act was key to this miti-
        gation decision based on...</relationToRiskAction>
     </lawAndRegRef></lawAndRegRefs>
+    <securityPolicyRefs>
+    <securityRequirementRefs>
   </mitigationActions>
+  <acceptanceActions>
   <transferenceActions /> <!-- No actions defined -->
   <avoidanceActions /> <!-- No actions defined -->
+  <lawsAndRegs>
+  <securityPolicies>
+  <securityRequirements>
 </needsBase>
```

As can be seen above, `needsBase` is the root element and its sub-elements encompass the four general types of security action, and the main factors identified which motivate them. In practice, SADML groups risks by the *type* of security action (e.g., mitigation, or `<mitigationActions>`) which addresses them, and then the exact written action (e.g., `<mitigationAction>`) defined by a company. Because one security action can address many risks, each action has a `<risks>` element that lists the risks addressed. The elements suffixed with 'Refs' are used to indicate that existing motivational factors, for example laws and regulations (`<lawsAndRegs>`), influenced the treatment of a risk. `<securityRequirementRefs>` is the exception, in that it references security requirements (`<securityRequirements>`) that detail security actions. SADML's structure proposes one way to define security actions, risks and motivational factors, and does not intend to be a panacea in itself.

The novelty of SADML is rooted in the unique business perspective it takes on risks and security actions, which aims to (i) maintain a strong practical foundation (by mirroring the ontology designed) and (ii) place security, at least initially, at a level that understandable to security professionals and business-based decision makers (often the budget holders) alike. Section V which follows, reports on the findings of the evaluation on the Solution Model and SASaCS tool, to date.

V. Evaluation and Findings

The evaluation of this research's proposals was based around assessing the suitability and use of SASaCS (inclusive of the Solution Model's components) in supporting, and even streamlining the overall negotiations process in BOF4WSS. This paper focuses only on one of the initial evaluation stages which investigated the compatibility of SASaCS and the ontology in particular, with existing risk management/assessment approaches. Compatibility formed a critical requirement because information (e.g., security actions, risks) from these approaches, irregardless of how different they are, or what output they generate, should be accommodated in the tool (and by extension the ontology which drives it). If the tool was able to capture a majority of the output (even if this involved a mapping of semantically similar concepts), this would support the completeness of the ontology which underlies the tool, and give evidence to show that SASaCS can adequately fit in, and work alongside current approaches used in companies today.

To evaluate compatibility, two well-known risk management approaches were chosen, i.e., CORAS [5] and EBIOS [6]. The softwares that support these methods made it easier to identify the exact output that companies would produce, which would then need to be accommodated by the tool/ontology. To add structure to this evaluation, the method for mapping security guidelines to an existing model (both high-, and low-level) in [7] was employed. Through the completion of its steps, a detailed assessment was carried out to determine how well the tool/ontology mapped, and thus were compatible with existing risk management/assessment approaches. Having outlined our focus, the following paragraphs present key findings of the completed evaluation.

In general, SASaCS proved itself a compatible solution as it was successfully able to capture a majority of the information output from CORAS and EBIOS. Coverage was especially good for the core concepts such as risks, security actions, risk treatments and security requirements. This mapping was so promising that a semi-automated transference of output from those risk management softwares into SASaCS, is being pursued.

A shortcoming in the ontology discovered from EBIOS output mapping was as a result of the link maintained between security actions (i.e., risk actions in [4]) and risks. In the ontology and therefore in SASaCS, based on the investigation then, it was concluded that security actions primarily originated to handle, or treat risks. This conclusion

however was disproved by EBIOS as a security action could be created to directly address constraints (e.g., operational, financial), regulations, or security rules and policies.

Lastly, in the ontology and tool, laws and regulations, security and business policies, and security budgets were defined as the prime factors which motivated a risk's treatment. Findings during the mapping evaluation however, showed that there were various other aspects which influenced and by themselves lead to the creation of security actions. A good example is the constraints of a operational, technical, budgetary and even territorial nature, faced. To enhance the ontology and tool, the shortcomings identified in this and previous paragraphs will need to be addressed. This is discussed more in future work; next we cover related work.

VI. Related Work

In [8], authors assessed similar disparity problems to the Solution Model, particularly in communicating security requirements. They proposed a framework for formally specifying requirements and detecting conflicts amongst collaborating parties. The core difference between that research and our work is in the layers targeted; the Solution Model supports high-level security negotiations for businesses, whereas [8] considers low-level security requirements (and by extension, only risk mitigation), and formal rules and algorithms for requirements refinement.

Apart from the related literature on the ontology previously presented in [4], the only other area with similar work is the XML-based language defined. In research and industry there have been a plethora of security languages covering from access control (e.g., XACML), to identity management (e.g., SAML). The most relevant to our work is the Enterprise Security Requirement Markup Language (ESRML) [9]. This language is comparable to SADML because it emphasizes the higher layers of security, and the sharing and exchanging the enterprise security information across companies for business purposes. The shortcomings of ESRML in terms of this work however are its lack of emphasis on factors which significantly influence or drive security actions (e.g., regulations, constraints), and its concentration on risk mitigation as opposed to explicitly appreciating other ways to treat risks.

VII. Conclusion and Future Work

In this paper we present a novel solution model and tool to support the negotiation of security actions in e-business collaborations. Although developed to accompany BOF4WSS, these proposals are likely to have further applications in supporting one-off, cross-enterprise business-based interactions, or even internal company negotiations. Having outlined the model and tool, we then briefly reported on one stage of the evaluation process which assessed ontology and tool compatibility with existing risk management processes.

Beyond addressing the shortcomings discovered in the compatibility evaluation, the next steps in our research involve the continued assessment of the Solution Model and tool. Currently, we are concluding an evaluation of the tool, which involved interviews with industry-based security professionals familiar with cross-enterprise security negotiation issues. Next, we will make any necessary changes based on feedback, and then test the tool in a real-world scenario to ultimately evaluate its suitability in supporting the security actions negotiation process.

References

[1] J. R. Nurse and J. E. Sinclair, "BOF4WSS: A Business-Oriented Framework for Enhancing Web Services Security for e-Business," in *4th International Conference on Internet and Web Applications and Services (ICIW)*. IEEE Computer Society, 2009, pp. 286–291.

[2] J. S. Tiller, *The Ethical Hack: A Framework for Business Value Penetration Testing*. Boca Raton, FL: Auerbach Publ., 2005.

[3] S. Dynes, L. M. Kolbe, and R. Schierholz, "Information security in the extended enterprise: A research agenda," in *AMCIS 2007 Proceedings*, 2007.

[4] J. R. Nurse and J. E. Sinclair, "Supporting the comparison of business-level security requirements within cross-enterprise service development," in *Business Information Systems*, ser. Lecture Notes in Business Information Processing, W. Abramowicz, Ed. Heidelberg: Springer, 2009, vol. 21, pp. 61–72.

[5] F. den Braber, G. Brændeland, H. E. I. Dahl, I. Engan, I. Hogganvik, M. S. Lund, B. Solhaug, K. Stølen, and F. Vraalsen, "The CORAS model-based method for security risk analysis," SINTEF, Tech. Rep., 2006.

[6] DCSSI, "Expression des besoins et identification des objectifs de sécurité (EBIOS) – section 1–5," Secrétariat général de la défense nationale, Direction Centrale de la Séccuritéc des Systècmes D'Information (DCSSI), Tech. Rep., 2004.

[7] S. Fenz, T. Pruckner, and A. Manutscheri, "Ontological mapping of information security best-practice guidelines," in *Business Information Systems*, ser. Lecture Notes in Business Information Processing, W. Abramowicz, Ed. Heidelberg: Springer, 2009, vol. 21, pp. 49–60.

[8] S. S. Yau and Z. Chen, "A framework for specifying and managing security requirements in collaborative systems," in *Autonomic and Trusted Computing*, ser. Lecture Notes in Computer Science, L. T. Yang, H. Jin, J. Ma, and T. Ungerer, Eds. Heidelberg: Springer, 2006, vol. 4158, pp. 500–510.

[9] J. Roy, M. Barik, and C. Mazumdar, "ESRML: a markup language for enterprise security requirement specification," in *IEEE INDICON*, Kharagpur, 2004, pp. 509–512.