Evaluating the Compatibility of a Tool to Support E-Businesses' Security Negotiations

Jason R.C. Nurse[1] and Jane E. Sinclair[2] *

Abstract—**As e-businesses partner to engage in online business scenarios, they face numerous challenges when considering the sharing, comparison, and negotiation on their individual security needs. To aid companies in this task, in previous work we have presented a security negotiations support tool, which acts as a bridge between businesses and streamlines various negotiation tasks. The paper continues the research of that tool by evaluating its compatibility with existing security needs determination methods. Compatibility forms a key requirement as it demonstrates feasibility and gives valuable initial feedback on the ultimate usefulness of the tool.**

Keywords: security actions and requirements, risks, e-business negotiations, IT risk management systems

1 Introduction

In today's business world, the importance of information security cannot be overlooked. Apart from self-preservation and protecting company assets, companies are expected to subscribe to various security best practices (e.g. ISO 27000), and they must now implement security to comply with a range of legal/regulatory requirements (e.g. UK Data Protection Act). When considering security approaches particularly across collaborating e-businesses, the security situation becomes exceedingly complex as partnering entities have a variety of different security needs, maintain differing security postures, may have dissimilar laws/regulations which apply, have different skill sets/experience levels, and so on. Work in [1] supports these difficulties as the author labels the related process, "security mayhem". To assist companies in this collaboration process, especially in terms of security approaches in Web services-based interactions, in previous work we have presented BOF4WSS, a Business-Oriented Framework for enhancing Web Services Security for e-business [2]. The framework's novelty stemmed from its focus on a cross-enterprise development methodology to help collaborating e-businesses in jointly creating secure and trusted interactions.

Having created BOF4WSS, our emphasis has shifted to providing systems and software to support it, and assist its seamless application to business scenarios. In this paper we present the first steps of an evaluation of one of these systems, which was developed to support and ease security negotiation across collaborating e-businesses. In terms of BOF4WSS, this refers specifically to easing the transition from the Requirements Elicitation stage to the Negotiations stage. Problems identified and targeted include: (i) understanding other companies' security documentation—a variety of formats and terminologies are used by companies to express their security needs; (ii) understanding the motivation behind other companies' security needs/decisions—incomplete information provided initially, usually demands that considerable time is spent later on determining core reasons for security needs; and (iii) being able to easily match and compare security decisions from businesses which target the same situation—to identify comparable security decisions involves looking through partners' security documents, and numerous tedious back-and-forth communications. These problems, and the tool (as well as its underlying Solution model) developed to tackle them, are presented in detail in [3]. Evidence to affirm these problems has been provided by relevant industry-based security professionals, and related research in [4].

As mentioned above, and mainly due to space limitations, this paper focuses on the first steps in the evaluation of the tool. Specifically, we assess the compatibility of the tool with existing Risk Management/Assessment (RM/RA) approaches; RM/RA approaches are relevant as companies typically use them to make decisions on security risks and determine their security needs. Compatibility forms a critical requirement because the information (on threats, vulnerabilities, risks, security needs, risk treatment options, motivational factors such as laws, security policies and so on) output by these RM/RA approaches in BOF4WSS' Requirements Elicitation stage, will need to be incorporated into the tool to enable it to fulfill its purpose. If the tool can capture a majority of the security-related information output from popular RM/RA techniques, its compatibility and feasibility as a tool that can work alongside current approaches used in businesses today, will be evidenced.

This paper is structured as follows. Section 2 recaps the Solution model and resulting tool to support security negotiations across e-businesses. In Section 3, we outline

*Manuscript received December 17, 2009.
[1,2] University of Warwick, Coventry, CV4 7AL, UK.
{jnurse[1], jane.sinclair[2]}@dcs.warwick.ac.uk

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the evaluation method followed. Sections 4 and 5 present compatibility tests against two well-known RM/RA approaches. A reflection on the evaluation findings is covered in Section 6. Conclusions are presented in Section 7.

2 Solution Model and Tool

The Solution model is the conceptual base for the software tool developed in our work. It consisted of four component stages: Security Actions Analysis, Ontology Design, Language Definition, and Risk Catalogue Creation. The **Security Actions Analysis** stage focused on reviewing the literature in the security risk management field, and critically examining how security actions and requirements were determined. A *security action* is broadly defined as the way in which a company handles the risk it faces (e.g. 'maintaining availability of data centers is to be outsourced'), and a *security requirement* is a high-to-medium level desire, expressed to mitigate a risk (e.g. 'all connections to the database must be authenticated'). The key outcome of this stage was a thorough understanding of the relevant security domain which could then be used as a foundation for future stages.

The **Ontology Design** stage following, aimed to produce a high-level ontology design, using the findings from the previous stage, to establish a common understanding and semantics structure of the security actions (and generally security risk management) domain. This common or shared understanding was a critical prerequisite when considering the difficulties businesses faced (because of different terminologies used, RM/RA methods applied, and so on) as they tried to understand their partners' security documentation supplied in BOF4WSS' Negotiations phase. The Security Actions Analysis and Ontology Design stages (inclusive of a draft ontology) were discussed in [5].

Next was the **Language Definition** stage and this had two parts. First was the development of a XML-based language called Security Action Definition Markup Language (SADML). This allowed for the establishment of a common format (based on the ontology) by which security actions/requirements information provided by companies could be formally expressed, and also later processed by the resulting tool. Second was the proposal of a user-friendly interface such as a data entry screen or template document by which businesses' security-related data could be entered, and subsequently marked up in SADML. This interface would act as a guide for companies in prompting them to supply complete information as they prepare to come together for negotiations.

The last stage was **Risk Catalogue Creation**, and that addressed the problem of matching and comparing security actions/requirements across enterprises by defining a shared risks catalogue. Given that businesses used risks from this shared catalogue as input to their RM/RA methods, regardless of the security actions that they decided individually, the underlying risks could be used by the tool to automatically match their actions. To increase flexibility, the catalogue would feature an extensive and updatable set of security risks.

Having reviewed the Solution model, Figure 1 shows a process flow of how the implemented model i.e. the tool, works. In this diagram, Comp A and Comp B are companies using BOF4WSS for an online business scenario.

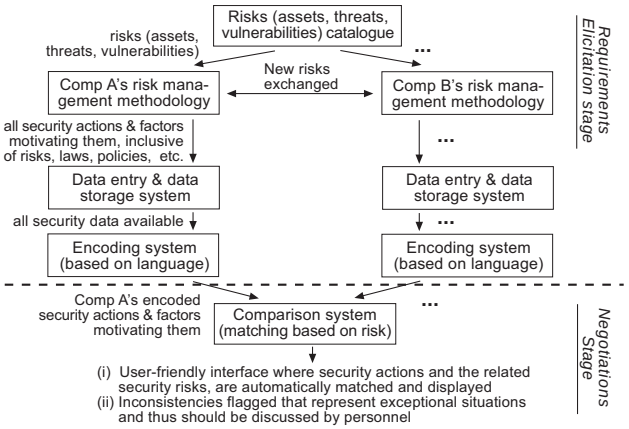

Figure 1: Process flow of implemented Solution model

First, companies would select a set of risks from the catalogue that apply to the business scenario, and use these as input to their different risk management methodologies/processes. Any new risks to be considered which are not available in the catalogue, can be exchanged for this scenario. After companies have used their RM/RA approaches to determine their individual security actions (inclusive of motivational factors), these are then input into the Data entry and storage system. This system uses a user-friendly interface to read in the data (as suggested in the Language Definition stage), and stores it to a back-end database to allow for data retrieval, updating and so on. This interface, and generally the tool, mirror the understanding of concepts defined in the ontology.

As companies are about to come together for Negotiations, the Encoding system is used to read security data from the database and encode it into SADML. In the Negotiations stage of BOF4WSS, companies bring their individual SADML documents and these are passed to the tool's Comparison system. This system matches companies' security actions based on risks which they address, and aims to provide a user-friendly interface in which (i) security actions can be quickly compared and discussed, (ii) any inconsistencies would be flagged for follow-up by personnel, and (iii) a shared understanding of security terms, risks and so on, will be upheld due to the references that can be made to the ontology. Having reviewed the model and tool, Section 3 begins the core contribution of this paper by presenting the evaluation method that will be used to assess tool compatibility.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3 Evaluation Method

In evaluating the compatibility of the tool, the core question was whether information output from typical company RM/RA methodologies could be accommodated by, or mapped to the tool's Data entry and storage system. To guide this compatibility evaluation, the method for mapping security guidelines and standards to an existing ontology (both high-level and formal) proposed in [6], was employed. This method supplied a tested technique in which a detailed assessment could be carried out to determine how well the tool mapped, and thus was compatible with existing RM/RA approaches. The Solution model's ontology was extremely useful here as it embodied all the concepts implemented in the tool. To provide the basis of the compatibility evaluation, two RM/RA methodologies were chosen, namely CORAS [7] and EBIOS [8]. These were selected because (i) they are well-known and used, (ii) there was extensive documentation openly available on each, and (iii) they had supporting softwares which generated machine-readable output (both provide XML-based documents). It is this machine-readable output that is expected to be mapped to, and ideally automatically read into the tool. The next section begins tool evaluation by testing compatibility with EBIOS.

4 Testing Compatibility with EBIOS

EBIOS is a risk management approach created under the French General Secretariat of National Defence. It proposes a methodology and supporting software, for assessing and treating risks in the field of information systems security [8]. To test tool compatibility with EBIOS, our research involved traversing all the steps advocated in [6]. Due to space limitations however, this paper concentrates on the presentation of the high-level mapping completed, and the provision of an example of how security information and knowledge from EBIOS, was mapped to the tool's data entry fields and ultimately, its database.

The high-level mapping of EBIOS concepts to the Solution model's ontology is displayed in Figure 2; ontology concepts are in boxes with unbroken lines, whereas EBIOS concepts have dashed lines. From this mapping one can easily visualize high-level similarities across models and also begin to identify concepts that do not map. Ontology concepts were largely discussed in [5] with the main differences here being the use of the term *security action* as opposed to *risk action*, and the introduction of *security attribute* (a property of an asset that is to be preserved e.g. confidentiality, integrity, availability and accountability), *security requirement* (defined prior), and *treatment* (the known degree to which a security action covers a risk) concepts.

Some of the more interesting concepts covered by EBIOS include a *menace*, which defines a threat to an entity (or asset); a *constraint* described as a limitation faced by the

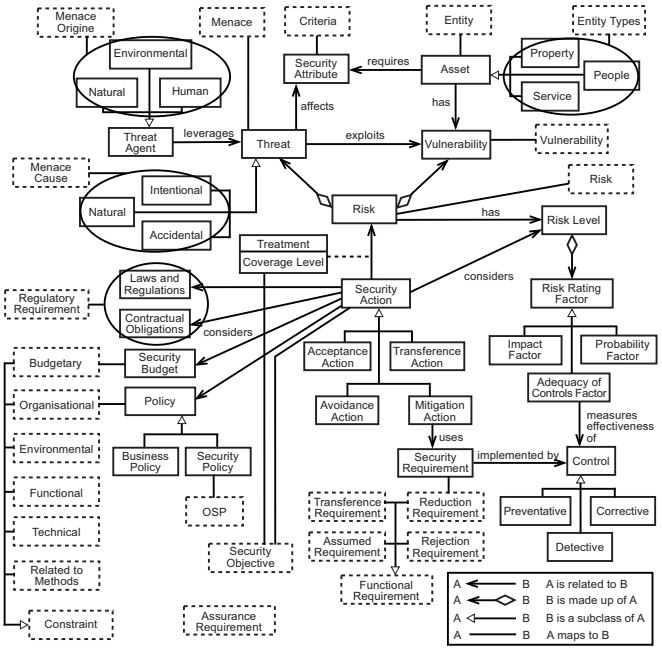

Figure 2: Mapping EBIOS concepts to the ontology

organization; a *security objective* which is the expression of the intention to counter risks or threats and/or comply with the organizational security policies and other assumptions; a *security functional requirement*, a security function to be implemented to contribute to the fulfillment of a security objective; and an *assurance requirement*, defined as the specification of assurance provided by security functions implemented to cover security objectives [8].

To evaluate the tool compatibility at a lower level, we now consider mapping actual output generated from a RA study conducted using EBIOS software, to the tool's database Entity Relationship Diagram (ERD). The XML snippet below describes the *security objective* arising from the study, which was defined to treat a security risk.

```
<SecurityObjective ID="SecurityObjective.1248768933881" label=
   "Eavesdropping protection objective" state="" baseID=""
   type="EBIOS.Text.SO.Type.TOE" content="The organization must
   take measures to ensure there is no eavesdropping on data,
   persons, meetings, etc..." resistance="3"
   resistance_justification="" coverLevel=
   "SecurityRequirementCover.1076860509716" ...>
 <SecurityObjectiveCovers>
   <SecurityObjectiveCover ID="SecurityObjectiveCover.
      1245667560533" reference="RiskScenario.1248601769338"
      type="Risk" />
 </SecurityObjectiveCovers>
</SecurityObjective>
```

To consider the mapping, a *SecurityObjective* in EBIOS corresponds to a *SecurityAction* table record in the ERD. Analyzing the concept's XML attributes, *label* which is the name of a security objective, mapped to *SecurityAction.sa_name*, and *content*, a description of the objective, mapped to the ERD's *SecurityAction.action_remarks*. None of the other attributes allowed for a mapping because no related fields existed in the tool ERD.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The *SecurityObjectiveCovers* sub-element lists aspects (risks, constraints, regulatory requirements, and so on) addressed by the current security objective. The *type* attribute of individual *SecurityObjectiveCover* elements marked the type of aspect addressed, here it is a Risk. In this example, a mapping was made between the risk addressed (identified by unique id 'RiskScenario.1248601769338') and a database record in the *ProjectRiskAction* table (this table holds risks which a security action addresses). Lastly, and more at a general level, because the *SecurityObjective* element does not define a type (i.e. whether it is geared towards risk mitigation, assumption, and so on) some manual intervention was required to complete the mapping to the *SecurityAction* table and thus provide data for the record's *action_type* field. A screenshot of the actual records in their respective tables within the tool database is shown in Figure 3, before general reflections on mappings done thus far.

SecurityAction

sa_id	sa_name	action_type	action_remarks
38	Eavesdropping protection ...	Mitigate	The organization must ...

ProjectRiskAction

pra_id	pr_id	sa_id	coverage_level	coverage_level_detail
22	3	38	*NULL*	*NULL*

Figure 3: Mapped SecurityObjective data

The principal aim of conducting the mapping process was to evaluate the compatibility of the tool and embodied ontology, with existing RM/RA approaches. Having completed the mapping of EBIOS, it can be seen that various of the main concepts and elements could be mapped, both at ontology and ERD levels. This has demonstrated promising evidence to support the case for tool compatibility. Of equal interest however are the concepts and element attributes that proved challenging to map, as these might indicate noteworthy shortcomings of the tool. Below, the primary difficulties incurred are discussed.

No consideration of assurance of security functions: Beyond defining security objectives, and security functional requirements that implement them, EBIOS uses security assurance requirements to provide assurance that functional requirements adequately achieve the objectives they are to implement. Reflecting on the tool and ontology, while both include concepts mappable to security objective and security requirement, neither accommodated the security assurance concept. For EBIOS mapping, this fact acted to highlight a weakness in the tool and ontology (specifically in their ability to capture all security aspects), and hence affected compatibility.

Low-level differences between EBIOS' Security objective and the tool's Security action: At a high level, *SecurityObjective* and *SecurityAction* are semantically similar, and thus allowed for a seamless mapping of concepts. When assessed in detail however, as seen in the lower-level mapping attempted, a few differences emerged (related to attributes and elements) which complicate the process. One such difference deals with the inability to identify an appropriate action type (mitigation, transference, and so on) for the corresponding *SecurityAction* database record without manual intervention. The next difference is centered around the fact that in EBIOS, a security objective can be conceived to address a range of aspects including risks, constraints, regulatory requirements, and security rules/policies. This is a novel fact because it exemplifies a direct relationship between a security objective and aspects that are not risks. This relationship was not represented in the tool or ontology. To take an example, in the tool and ontology, a Security action or risk action is conceived with the prime aim of treating a risk. Aspects such as those mentioned above i.e. constraints, regulatory requirements, and security rules/policies, are mainly viewed as constructs that influence the treatment of the risk. This is as opposed to constructs which independently give rise to security actions or general security needs.

With the compatibility tests with EBIOS complete, the next section considers tool compatibility with CORAS.

5 Testing Compatibility with CORAS

CORAS [7] is the product of an EU research project targeted towards creating a tool-supported methodology for model-based risk analysis of security-critical systems. To report on the evaluation in terms of CORAS, the same process (i.e. high-level mapping, and information mapping example) used for EBIOS above is reused here. Figure 4 therefore presents the high-level mapping accomplished. As seen from the mapping, a majority of CORAS concepts are found in the ontology. Its *unwanted incident* and *threat* concepts proved the most intriguing during mapping, as they covered multiple concepts in the ontology. Formally, an unwanted incident is an event that reduces the value of assets, whereas a threat, defined as a potential cause of an unwanted incident, this encompassed the human, or non-human cause [7]. A full description of all CORAS concepts can be found in [7].

To conduct the mapping of actual CORAS output data next, a case study was prepared in the CORAS software, and then exported to its project XML format. As with EBIOS, default settings were used in the CORAS software and customization was kept at a minimum to maintain an objective mapping. In this low-level mapping here, we test the ability of the tool to map data from the Consequence and Frequency Table in CORAS. This table identifies risks, makes the link to associated unwanted incidents, and values each risk in terms of consequence (impact of an unwanted incident on an asset in terms of loss of asset value) and frequency (the probability for an unwanted incident to occur). The code follows.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

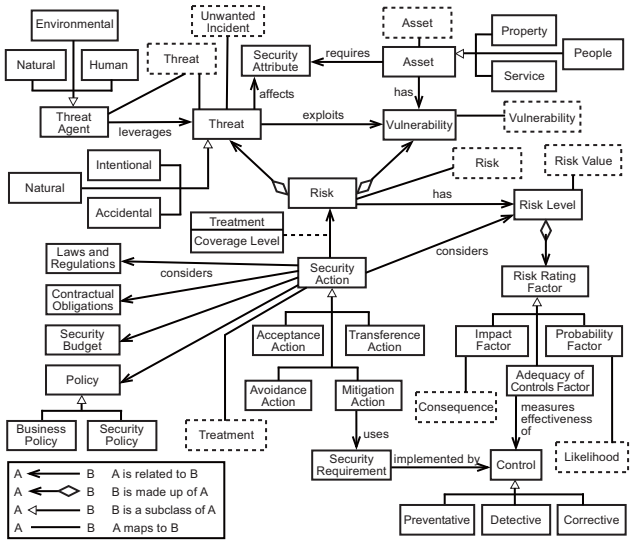

Figure 4: Mapping CORAS concepts to the ontology

```
<row>
 <cell columnId="riskId">RSK-1</cell>
 <cell columnId="assetId">Network1</cell>
 <cell columnId="incident">Unauthorized disclosure of customer
    personal data</cell>
 <cell columnId="consequenceValue">Moderate</cell>
 <cell columnId="frequencyValue">Likely</cell>
 <cell columnId="scenario"/>
</row>
```

To map the risk defined in the <row> element above, the ERD's *Risk* and *ProjectRisk* tables were employed. After creating a new *Risk* database record, the *riskId* element's data was mapped to the *Risk.risk_id* field. For the CORAS row's *assetId*, the respective asset's unique identifier (i.e. *asset_id* in the ERD *Asset* table) for 'Network1' was copied to *Risk.asset_id*. A similar process was adopted for the *incident* element as this would correspond to a record already in the ERD *Threat* table. The unique identifier copied was *threat_id*, and it was copied to the *Risk.threat_id* field. To complete the ERD *Risk* record, the incident's respective *vulnerability* from the CORAS software Scenario Table was used. Once the incident's vulnerability was found (note that in each row in the CORAS Scenario Table is an *incident* and a respective *vulnerability*), the ERD's Vulnerability table was searched for that vulnerability's name (on the *Vulnerability.vulnerability_name* field). When the database record was identified the *vulnerability_id* field was copied/mapped to the respective *Risk* record's *Risk.vulnerability_id* field.

The last task was mapping CORAS consequence and frequency data. Assuming that metrics (i.e. allowed values) for these factors were set to be the same in both CORAS and the tool (note that metrics can be added to the tool using *PrioritizationScheme* ERD table), the 'moderate' consequence in CORAS mapped to 'moderate' value for the *impact* field in the tool's *RiskEstimate* table. Whereas the 'likely' frequency mapped to the 'likely' value for the *probability* field in ERD's *RiskEstimate* table. For the mapping above to be conducted however, a *ProjectRisk* database record was required first. From the ERD, it would be noted that *ProjectRisk* supplies the physical link between a *Risk* and a *RiskEstimate*. Once this record was created and associated with the *Risk* under analysis, the unique *pr_id* key value generated was copied to a new *RiskEstimate* record. The relevant *impact* and *probability* values were then copied to that new *RiskEstimate* record. As before, a screenshot is presented in Figure 5 to show the resulting mappings in the tool database.

Risk

risk_id	asset_id	threat_id	vulnerability_id	general_risk_info
RSK-1	22	43	80	*NULL*

ProjectRisk

pr_id	project_id	risk_id	asset_details	agent_details	thre...
88	9	RSK-1			

Prioritization Scheme

ps_id	priority_name	priority_description	rating_factor_type
7	likely	Possible that the...	probability
33	moderate	(1) May result in...	impact

RiskEstimate

re_id	probability	impact	pr_id	probability_remarks	...
29	7	33	88		

Figure 5: Mapped Consequence and Frequency table

As might be noted from the mapping above, the tool does require companies to first synchronize information on elements such as risk and risk ids to be used (recall that tool comparison is made largely based on common risks), and the metrics for risk valuation i.e. ensuring companies use similar valuation schemes and agree on the meanings of individual metrics. Having completed the mapping of the security information from CORAS software output to the tool's ERD and ontology, the following paragraphs discuss the more salient observations made during the general mapping process.

Reflecting on the general CORAS mapping, there were many high- and low-level concepts that evidenced compatibility of the tool. This was so promising that an automated mapping between the CORAS software and the tool would be almost seamless. The main problems that could prohibit this are highlighted below.

Differences in Threat representation: In the tool and ontology, a *Threat* concept defines an undesired event which has an adverse impact on an asset. Within CORAS, this threat notion is understood in a slightly different way which caused the need for the *unwanted incident* and *threat scenario* (or *threat*) concepts in CORAS, to map to the single *Threat* concept in the ontology. The difficulty at this point therefore is deciding exactly how to map low-level CORAS data, to the tool's database. One option was to map a CORAS *unwanted incident* to a ERD's *Threat* (as these definitions are quite similar) and then discard data in the CORAS *threat scenario* field. The disadvantage of this however was losing data which provided more descriptive information on what actions

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(or causes) constituted a threat to an asset. The second option involved concatenating related data in the CORAS *unwanted incident* and *threat scenario* fields, and then mapping that data to records in the ERD's *Threat* table. This option however would lead to multiple threats (a new ERD Threat record for each *unwanted incident* and *threat scenario* pair) for a single risk. This is not a mapping the tool's ERD at present could accommodate. The first option was therefore preferred in most mappings.

Determining actual risk treatments: CORAS and the tool and ontology, both acknowledge the need for risk treatment concepts. In the CORAS software, they begin by listing all possible treatment options in the Treatment Identification Table. Next, in the Treatment Evaluation Table, they evaluate all the treatments and use priority values to rate them. The difficulty in mapping was because the tool only accommodated actual treatments which were chosen to address a risk. Therefore, the treatment evaluation process documented in CORAS, was taken to be complete from the tool perspective. Another difficulty faced was the identification of the specific treatment which would handle a risk. The CORAS software and its output, maintained no data fields or facility which clearly highlighted a chosen treatment. The *treatmentPriority* element in the Treatment Evaluation Table was considered to aid in mapping, however, because there was no predefined hierarchy of metrics (e.g. high, medium, low) in the CORAS software, the possibilities of values used by companies to rate their treatments was infinite, and thus not mappable. To allow for mapping therefore, a manual process was required where treatments (from the Treatment Identification Table) to be mapped from CORAS to the tool were identified by a user. The use of a manual means for mapping was not ideal but was necessary as it was the only way to definitively identify a treatment to be mapped from CORAS.

6 Reflecting on the Evaluation

From the compatibility tests conducted above, the tool has shown itself to be an adequate system, capable of working alongside common RM/RA approaches and softwares in use today. In the hope of increasing compatibility even more, we are considering three changes to the tool. These are: (i) allowing a *SecurityAction* to directly address aspects other than *Risks*, for example, laws/regulations, technical constraints, and so on—therefore its new meaning is 'any way in which to address a risk, or a constraint to a organization or system'; (ii) introducing a generic *Constraint* concept which encapsulates all constraints (e.g. security budget, contractual obligations, and so on) that affect a risk's treatment (i.e. the Security need), or all constraints that need to be addressed directly by a Security action (see point (i)); and lastly (iii) the facility to map and store risk treatment evaluation data and treatment options. This would allow companies to state multiple security actions, which partners might consider if their first choice action can not be agreed across entities.

7 Conclusions and Future Work

This paper focused on evaluating the compatibility of a tool to support security negotiations across e-businesses. As shown, results thus far have proved favourable, however a few improvements to the tool are envisaged to ensure compatibility with a wider range of RM/RA approaches. Future work will consist of (i) further evaluation of the tool against other RM/RA methodologies, (ii) conducting interviews with security professionals to gather their feedback on the value of the tool from a practical perspective, and finally (iii) assessing how well the tool works when used to support companies in a real world e-business collaboration scenario.

References

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

[2] 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*. IEEE Computer Society, 2009, pp. 286–291.

[3] ——, "A Solution Model and Tool for Supporting the Negotiation of Security Decisions in E-Business Collaborations," in *5th International Conference on Internet and Web Applications and Services*. IEEE Computer Society. (To be published), 2010.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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, Tech. Rep., 2004.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)