A Thorough Evaluation of the Compatibility of an E-Business Security Negotiations Support Tool

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

Abstract—**For the benefits of e-business to be fully realized, there are numerous challenges to be overcome particularly with respect to security. Some of the most significant of these difficulties is incurred even before businesses fully enter the joint e-business interactions. A key example is the challenge faced as partnering e-businesses come together initially to share, compare and negotiate on their individual security needs. In previous work, we have proposed a support tool to assist in this activity and streamline several of the difficult security negotiation tasks which arise. This paper aims to advance the research of that tool by engaging in a very detailed evaluation of its compatibility with existing security needs determination methods (commonly, risk management and assessment techniques). Compatibility forms a crucial requirement as it evidences feasibility and yields worthwhile initial feedback on the ultimate usefulness and practicality of the tool.**

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

1 Introduction

The importance of information security in today's fast-paced business world cannot be neglected. In addition to self-preservation and defending company assets, businesses are expected to subscribe to a variety of security best practices (e.g. ISO 27000 series), and they must also implement security measures to comply with a range of legal/regulatory requirements (e.g. EU Data Protection Act and Health Insurance Privacy and Portability 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, 3]. 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. This would complement existing approaches and best pratices such as [4, 5, 6].

Having created BOF4WSS, our emphasis has shifted to providing systems and software to support it and assist in its seamless application to business scenarios. In this paper we extend work in [7] to look in greater detail at 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 [8]. Evidence to affirm these problems has been provided by relevant industry-based security professionals and related research in [9].

As mentioned above, this paper focuses in detail 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,

*Manuscript received October 4, 2010.

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

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 fulfil 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 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 consists of four component stages: Security Actions Analysis, Ontology Design, Language Definition, and Risk Catalogue Creation. The **Security Actions Analysis** stage focuses 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 is a thorough understanding of the relevant security domain which can then be used as a foundation for future stages.

The **Ontology Design** stage following, aims 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 is a critical prerequisite when considering the difficulties businesses faced (because of different terminologies used, RM/RA methods applied, and so on) as they try 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 [10].

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

The last stage is **Risk Catalogue Creation**, and that addresses the problem of matching and comparing security actions/requirements across enterprises by defining a shared risks catalogue. Given that businesses use risks from this shared catalogue as input to their RM/RA methods, regardless of the security actions that they decided individually, the underlying risks can 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.

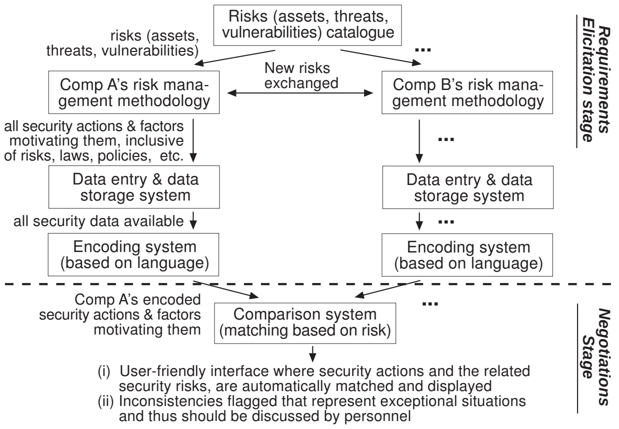

Risks (assets, threats, vulnerabilities) catalogue
risks (assets, threats, vulnerabilities)
...
Requirements Elicitation stage
New risks exchanged
Comp A's risk management methodology
Comp B's risk management methodology
all security actions & factors motivating them, inclusive of risks, laws, policies, etc.
...
Data entry & data storage system
Data entry & data storage system
all security data available
...
Encoding system (based on language)
Encoding system (based on language)
Comp A's encoded security actions & factors motivating them
Comparison system (matching based on risk)
...
Negotiations Stage
(i) User-friendly interface where security actions and the related security risks, are automatically matched and displayed
(ii) Inconsistencies flagged that represent exceptional situations and thus should be discussed by personnel

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 (database) 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 extended paper by presenting the evaluation method that will be used to assess tool compatibility.

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 (database) 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 [11] 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. For the lower level mapping of data, the tool's database Entity Relationship Diagram (ERD) was referenced. This is the database that was the back-end (or storage facility) for the Data entry system to the tool. A table-level view of the ERD is presented in Figure 2.

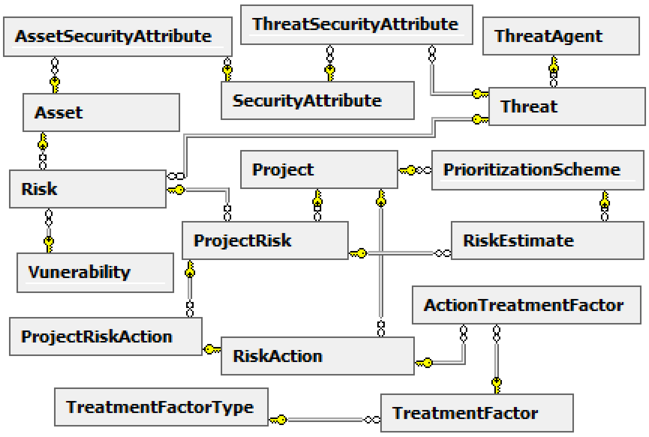


Figure 2: Tool's supporting ERD

A high-level description of the tool's ERD tables are as follows:

Asset: Defines anything of value to a business

AssetSecurityAttribute: Allows the security attributes of an asset to be defined
ActionTreatmentFactor: Defines treatment factors (e.g., laws, business policies, budgets and so on) that influence a security action
Project: Projects are defined for each RM/RA undertaking or scenario
ProjectRisk: Defines risks that has been selected for use in a particular project
ProjectRiskAction: Defines the risks which a security action addresses. It also defines the level of coverage provided by the action
PrioritizationScheme: Allows the creation of metrics (such as, High, Medium or Low) for each project that would be used to value risks and rate probability, impact and adequacy of controls
Risk: A risk is the potential that a threat will be leveraged by a threat agent to exploit a vulnerability in a given asset resulting in harm to an organization
SecurityAction: Specifies the way in which a company handles the risk(s) it faces
RiskEstimate: Defines the value of a risk, the probability and impact of it occurring, and the effectiveness of current controls in preventing that risk
SecurityAttribute: This is a property of an asset that is to be preserved and is another term for an information security goal. Examples of attributes are confidentiality, integrity, availability and accountability
Threat: A threat is an undesired event with an adverse impact on an asset
ThreatAgent: Specifies the cause of a threat
ThreatSecurityAttribute: Allows the security attributes which a threat affects to be outlined
TreatmentFactor: Defines the elements that affect the treatment of a risk
TreatmentFactorType: Lists the generic types of treatment factors
Vulnerability: A weakness in an asset or an existing security element intended to protect an asset

These descriptions and definitions are in line with those in our ontology.

To provide the basis of the compatibility evaluation, two RM/RA methodologies were chosen, namely CORAS [12] and EBIOS [13]. 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 software 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. Automation would suggest a promising mapping. The next section begins tool evaluation by testing compatibility with EBIOS.

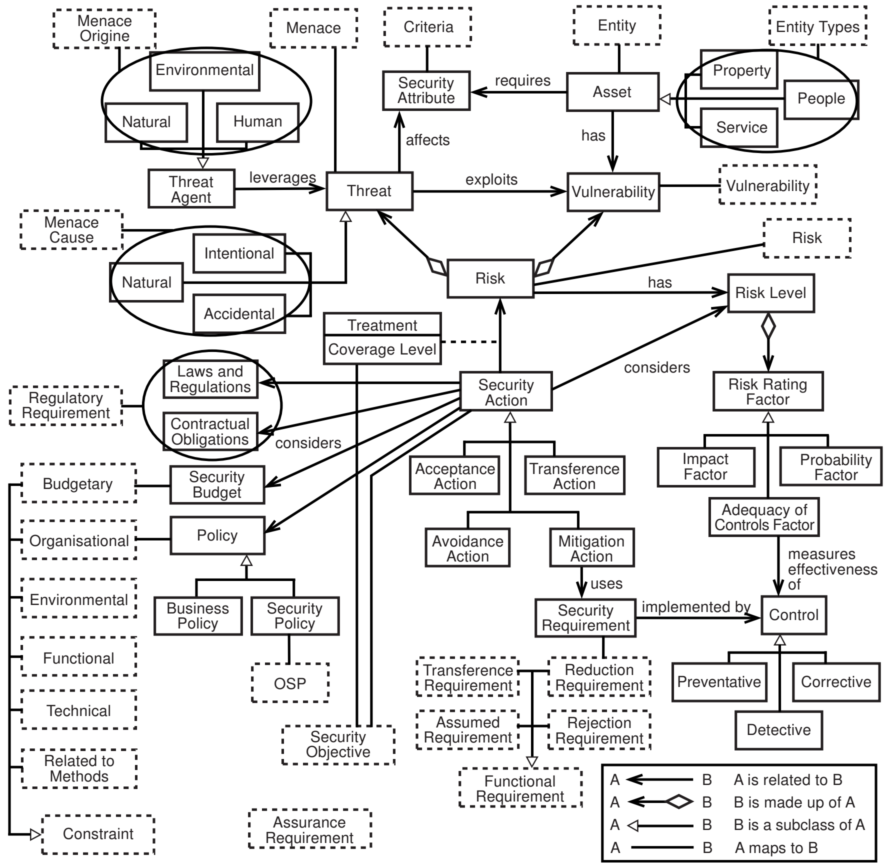


Figure 3: Mapping EBIOS concepts to the ontology

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 [13]. To test tool compatibility with EBIOS, our research involved traversing all the steps advocated in [11]. Focusing on the core of the work, this paper concentrates on the presentation of the high-level mapping completed and the provision of key examples of how security information and knowledge (low-level data) 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 3; 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 [10] with the main updates 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 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 [13].

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). For the tests, two output examples are used in this research. Both highlight concepts central to an EBIOS study. The first is presented below; this code covers EBIOS' XML representation of a Menace.

```
<Menace ID="Menace.1050437920519" label="19 - EAVESDROPPING"
   selected="true" description="Type: Human.  Deliberate
   cause: Someone connected to communication equipment/media
   or located inside the transmission coverage boundaries
```

```
    of a communication can use equipment, which may be very
    expensive, to listen to, save and analyze the information
    transmitted (voice or data). ..." justification=""
    descriptionMenaceElement="" potentiel=
    "AttackPotential.1070307963407">
  <MenaceThemeList ID="MenaceThemeList.1244991940438">
   <Theme id="Theme.1014431415703" comments="" />
  </MenaceThemeList>
  <SeverityScale ID="SeverityScale.1050985081072">
   <MenaceSeverity ID="MenaceSeverity.1244928987097"
      criteria="Criteria.1014877221686" severity="1"
      violation="true" />
  </SeverityScale>
  <MenaceCauseList ID="MenaceCauseList.1244991940438">
   <MenaceCause id="MenaceCause.1011656568285" comments="" />
  </MenaceCauseList>
  <MenaceOrigineList ID="MenaceOrigineList.1244991940438">
   <MenaceOrigine id="MenaceOrigine.1052902060343" comments="" />
  </MenaceOrigineList>
</Menace>
```

The XML excerpt describes EBIOS knowledge of an eavesdropping menace faced by a system in an RA study. To start, a *Menace* in EBIOS was mapped to a *Threat* in the tool. Considering the lower level, the *selected* attribute of the *Menace* XML element was assessed first. This attribute defined whether or not a menace was selected from the dataset of menaces and thus whether it applied to the current RA study. A ‘true’ value indicated that it was selected and thus the *Menace* should be mapped (or specifically, transferred) to the tool’s database and ERD.

The *label* and *description* attributes of the menace present descriptive information about the ERD’s *Threat* concept and so were mapped to the fields *Threat.threat_name* and *ProjectRisk.threat_details* respectively. To accommodate the latter of these mappings, it was noted that there would need to be an EBIOS’ *Risk* related to the *Menace* and it would need to have been mapped previously (to a tool/ERD’s *Risk* and *ProjectRisk* record). This would allow the menace description data to be added to the ERD *ProjectRisk* record. Neither the *justification*, *descriptionMenaceElement*, or *potentiel* attributes had mappable fields in the ERD.

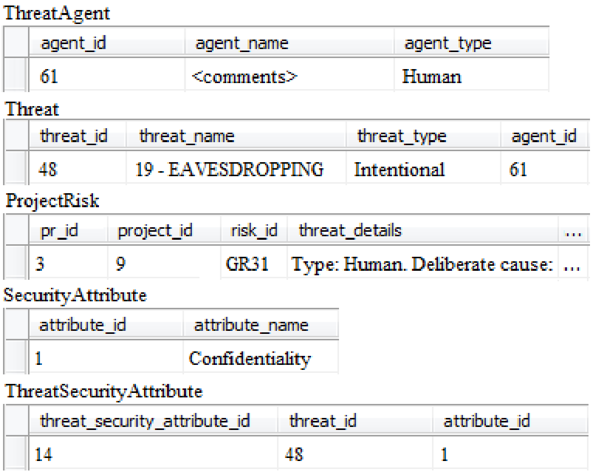
The *MenaceThemeList* sub-element listed the EBIOS attack method themes from which the *Menace* was deduced. The related *Theme* element (theme id ‘Theme.1014431415703’) in this case represented the ‘Compromise of information’ attack method. As the ERD did not maintain the overarching concept of attack themes however, mapping was not achieved.

The *SeverityScale* element allowed for an exact mapping (apart from the *severity* and *violation* attributes) because its focus, that is, *Criteria*, corresponds to the *SecurityAttribute* table data in the tool/ERD. The *SeverityScale* element allows for Security *Criteria* (such as availability, integrity and so on) that the menace affects to be specified. In this example, the criteria affected is ‘Confidentiality’, which is represented by the unique id ‘Criteria.1014877221686’. To map this *Menace* knowledge to the tool, the *SecurityAttribute* and *ThreatSecurityAttribute* tables were used. *SecurityAttribute* and specifically *SecurityAttribute.attribute_name* stored the types of security criteria (or attributes in the tool), whereas the *ThreatSecurityAttribute* table and the database record created provided the link between a *Menace/Threat* and the affected *Criteria/SecurityAttribute*.

The final two sub-elements, *MenaceCauseList* and *MenaceOrigineList*, list the causes (MenaceCause) and origins (MenaceOrigine) of the threat respectively. In the ERD, *MenaceCause* data mapped to the *Threat.threat_type* field of the current eavesdropping menace database record. Generally this mapping was ideal as the specific *MenaceCause*, ‘MenaceCause.1011656568285’, refers to a ‘Deliberate’ menace cause and therefore corresponds to the ERD’s ‘Intentional’ *threat_type* option (which is shown in the ontology).

For the *MenaceOrigine* element within the *MenaceOrigineList*, a relation was found in the *Threat.agent_id* field. To enable this mapping the *ThreatAgent* table was also required. This is because the menace’s origin, that is, ‘Human’ (indicated by unique id ‘MenaceOrigine.1052902060343’), would first map to the respective *ThreatAgent* database record (identified by *ThreatAgent.agent_type* and in this case ‘Human’ as well). Then the *agent_id* would be copied to the eavesdropping record in the *Threat* table.

Figure 4 pulls together the mapping example and displays a screenshot of the database records (in their respective tables) that would be created in our tool as a result of the mapping. In later stages this type of data would then be exported to SADML when companies are ready to compare and reconcile their security actions.



ThreatAgent

agent_id	agent_name	agent_type
61	<comments>	Human

Threat

threat_id	threat_name	threat_type	agent_id
48	19 - EAVESDROPPING	Intentional	61

ProjectRisk

pr_id	project_id	risk_id	threat_details	...
3	9	GR31	Type: Human. Deliberate cause:	...

SecurityAttribute

attribute_id	attribute_name
1	Confidentiality

ThreatSecurityAttribute

threat_security_attribute_id	threat_id	attribute_id
14	48	1

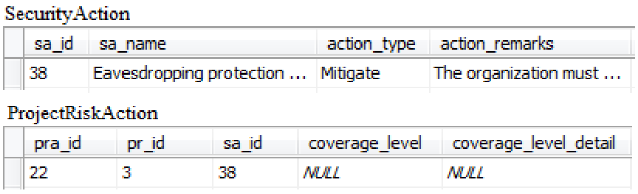
Figure 4: Mapped *Menace* data

The next knowledge mapping example was based on the EBIOS SecurityObjective element. The XML snippet below describes the *security objective* defined in the RA study. This objective was to treat the risk associated with the menace identified in the previous example.

```
<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.

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 5, 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 5: 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.

From a general perspective however, because the assurance concept was not prevalent in the range of popular RM/RA methodologies examined in [10], it may not be a standard concept in this context. Rather, a peculiarity of the EBIOS technique. Nonetheless, assurance is a generally well-accepted security facet therefore might need to be accommodated in the tool and ontology model.

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.

Only one security requirement or detailed treatment method for a security action: EBIOS allows for the use of multiple security functional requirements to implement a single security objective; in essence a one-to-many relationship. Conversely, the tool only accommodates one security requirement or detailed treatment

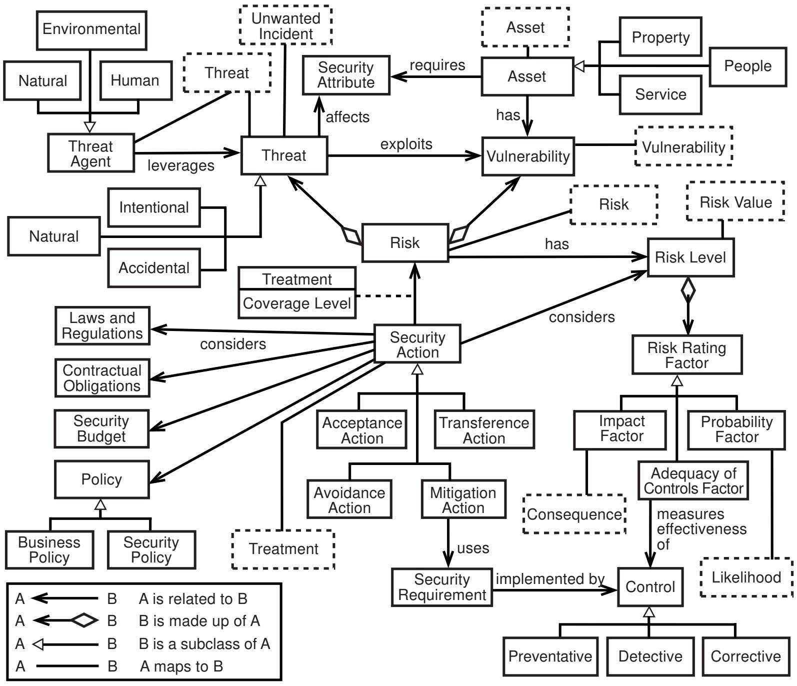


Figure 6: Mapping CORAS concepts to the ontology

method for each database *SecurityAction* record; that is, one-to-one. This difference in cardinality causes an obvious disparity during the mapping process as there is an inability to link multiple detailed treatment methods (in EBIOS, security functional requirements) to a single Security action (in EBIOS, security objective). From a practical perspective, the one-to-many relationship present in EBIOS should also be represented in the tool because such scenarios are foreseeable in reality. Its exclusion highlights an oversight in design and implementation.

With the compatibility tests with EBIOS complete, in the next section we consider tool compatibility with CORAS.

5 Testing Compatibility with CORAS

CORAS [12] 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 6 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 [12]. A full description of all CORAS concepts can be found in [12].

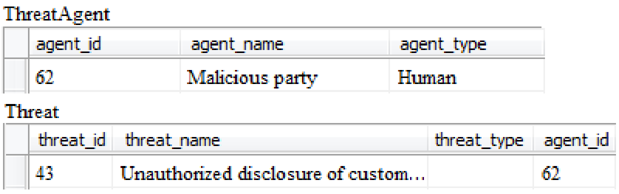
To conduct the mapping of actual CORAS output data next, a case study was prepared in the CORAS software, and then exported to its project XML format. The software's Help section and a case study by Fu et al. [14] which used CORAS, guided scenario preparation. As with EBIOS, default settings were used in the CORAS software and customization was kept at a minimum to maintain an objective mapping. In this low-level mapping here, we report on two examples to test the ability of the tool to map CORAS data. These cover threat scenarios, and risk identification and estimation. The mapping commences with the threat scenario XML below.

```
<row>
 <cell columnId="scenarioId">SNR-1</cell>
 <cell columnId="assetId">StaffNetwork1</cell>
 <cell columnId="reference">Sequence diagram 1 documentation
    </cell>
 <cell columnId="threat">Malicious party</cell>
 <cell columnId="vulnerability">Circulating information in
    clear text</cell>
 <cell columnId="incident">Unauthorized disclosure of
    customer personal data</cell>
 <cell columnId="scenario">Accessing and stealing of
    customers personal data</cell>
</row>
```

The code snippet above was taken from the CORAS Scenario Table and describes various aspects pertaining to a single threat scenario. A threat scenario or simply scenario, is how a threat leads to an unwanted incident. At the high level, an association has previously been made from data in this table (for example, threat scenario and incident) to the ontology and ERD's *Threat* concept. To conduct the lower-level mapping, the first task was to ensure that a “malicious party” threat cause (that is, the *threat* columnId in the code snippet) already existed in the respective ERD table, which is the *ThreatAgent* table. If there was no record, it needed to be created. The respective ERD *agent_id* field data (for that threat cause) was then used, in addition to the CORAS *incident* (short for unwanted incident) and *scenario* data to create a new *Threat* database record in our tool.

To consider the mapping in greater detail, the threat cause's *agent_id* was copied from the respective *ThreatAgent* record to the new *Threat* record's *Threat.agent_id* field. This sets up the database foreign key relationship. Also, the *incident* data formed the main input to the same *Threat* record's *Threat.threat_name* field. Data from the CORAS *scenario* element was appended to the record's *threat_name* data to provide an additional description of the new ERD *Threat* record. Appending this data however was not a panacea as it proved appropriate only if an *incident* had one associated *scenario*. CORAS allows multiple threat scenarios to culminate in one or more incidents; which could mean multiple rows (<row>) in the Scenario Table with the same *incident* data but different *scenario* data.

Regarding the Scenario Table's *assetId*, *reference* and *vulnerability* elements, there was no mapping to the ERD's *Threat* database record. The *vulnerability* element is pivotal in later stages however as it defines the related existing vulnerability which, along with a threat, constitute data for an ERD *Risk* record. This aspect would therefore be revisited when mapping risks. Figure 7 gives a visual presentation of the mapped data as it is captured in our tool's database.



ThreatAgent

agent_id	agent_name	agent_type
62	Malicious party	Human

Threat

threat_id	threat_name	threat_type	agent_id
43	Unauthorized disclosure of custom...		62

Figure 7: Mapped *Scenario* table

The second example of knowledge mapping uses the Consequence and Frequency Table in CORAS. This table defines 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.

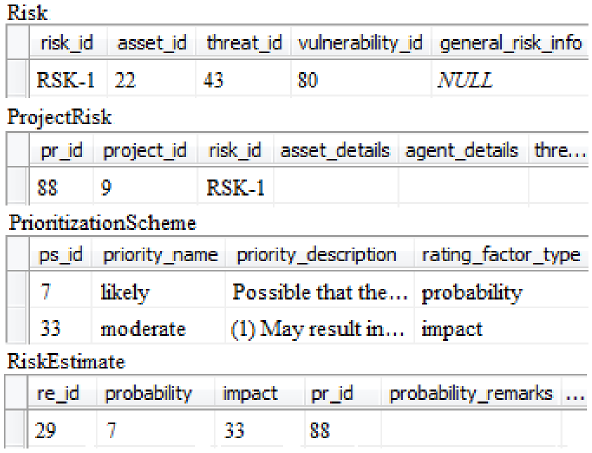
```
<row>
 <cell columnId="riskId">RSK-1</cell>
 <cell columnId="assetId">Network1</cell>
 <cell columnId="incident">Unauthorized disclosure of customer
    personal data</cell>
 <cell columnId="consequenceValue">Moderate</cell>
 <cell columnId="frequencyValue">Likely</cell>
 <cell columnId="scenario"/>
</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 8 to show the resulting mappings in the tool database.

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



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			

PrioritizationScheme

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 8: Mapped *Consequence and Frequency* table

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 (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.

Grouping of risks: Within CORAS, a shortcut mechanism is supplied that allows risks to be grouped into categories. These categories are used to allow a group of risks to be estimated using a single risk value and then treated using a single risk treatment. Such a grouping of risks functionality however does not exist in our tool's ERD. To enable for mapping therefore, risk categories needed to be broken down such that each risk was viewed individually. One interesting fact noted in the groupings was the facility to value/estimate a group of risks. This functionality was not accommodated in the tool or ontology but its use could be seen especially in cases where a set of related and minimal risks were grouped for joint valuation.

In the proposed mapping process itself, a noteworthy caveat was identified when extracting risks from within CORAS risk categories. This problem was linked to the suggested mapping's assumption that a risk category's value would be applied to each individual risk when the category was broken down. For example, assuming a group of risks have level 'High', if they were to be considered separately, each of their individual levels would also be 'High'. This however may not be the case as a system user (security professional or analyst) might have chosen to modify a risk category's valuation level depending on the combined consequence and frequency values of all the risks in the category. As a result of this fact, some manual intervention may be necessary during mapping to confirm each risk's individual risk value when risk categories are being broken down.

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

Both EBIOS and CORAS proved useful and insightful methods which aided in this research's evaluation. They are widely used risk management and assessment techniques which collect and generate a variety of risk and security data and information. Out of the two methods, EBIOS had the more comprehensive and detailed methodology and software, and it also had the most unmapped concepts. Even though having a number of unmapped concepts was not ideal (as it does not affirm compatibility), in retrospect, the reason for the difficulties incurred might be linked to the nature of EBIOS. Recall that EBIOS was developed under government direction (National Defense) and geared towards government industries [13]. The method therefore might be geared to very high security environments. This reality would account for the critical value and detail placed on security and security assurance (for example, the various fields for security data, coverage levels for risks and security objectives, and so on).

The CORAS technique offered a less detailed and more standard methodology and software. The mapping difficulties present with CORAS were not serious and hardly any main concepts differed across models (that is, from CORAS to the tool/ontology). CORAS did not introduce any new profound concepts either. Generally therefore, the differences were regarded as trivial and stylistic, and ones which could be easily accommodated during mapping by occasional manual intervention.

To briefly consider the methodology by Fenz et al. [11] used for the mapping, this supplied a tested technique from the literature to guide and add structure to the ontology mapping process. This technique was useful and easy to follow, albeit partially targeted at formal ontologies as opposed to database schemas or diagrams.

In terms of identifying specific scenarios, industries or RM/RA methodology types in which the proposed tool and ontology would prove more favourable to be applied, none were evident from the mapping of these two methodologies. It was clear however that basic risk concepts could be handled and therefore any methodology utilizing standard RM/RA concepts should allow for an adequate mapping.

Having discussed the high-level mappings and applicable usage scenarios, reflection turned to the lower-level aspects. To begin, an assessment was done to find any common weaknesses of the tool or ontology, that were incurred during mappings to both EBIOS and CORAS. If present, these might highlight areas where further work or modifications in the tool/ontology were warranted. From this assessment however, no common weaknesses were discovered.

Looking to the future, the next step was to consider the capabilities of each methodology and identify any aspects worth adopting to boost the compatibility of the tool/ontology to these or any other RM/RA methods. Other popular methods such as the NIST SP 800-30: Risk Management Guide [15] and OCTAVE [16] were also briefly studied to determine if there was any support for adopting novel EBIOS or CORAS aspects. From a research perspective, these considerations look to learn from and react to the general evaluation findings. Four aspects were chosen; three from EBIOS and one from CORAS. The choices made were primarily because the aspects were general enough to apply to any RM/RA method and also because they addressed what were regarded as key shortcomings in the tool/ontology.

These aspects were: (i) allowing a *Security action* to directly address aspects other than *Risks*, for example, laws and 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) adding the capability to have multiple Security requirements address or cover one Security action—this is a necessity in real-world situations; (iii) introducing a generic *Constraint* concept which encapsulates all constraints (such as security budget and contractual obligations) that affect a risk’s treatment, or all constraints that need to be addressed directly by a Security action (see point (i)); and lastly (iv) the facility to map and store risk treatment evaluation data.

Aspect (iv) above does not address a shortcoming as such but is suggested because it could be beneficial when businesses using the tool are trying to reconcile Security actions. If this evaluation data could be mapped from a RM/RA methodology, the tool could use it to display alternate risk treatments (along with risk reduction levels, treatment priorities and so on, if they exist) which companies might wish to consider in making reconciliation decisions. All four of the extensions suggested above would add greater flexibility to the tool and ontology, and increase chances of compatibility with more RM/RA methodologies and their software.

Factoring in some of these extensions, a first draft of an updated ontology is displayed in Figure 9. A new respective ERD has also been developed and will be seen next. As these are drafts, further tests will be needed to verify their rigour, identify any accompanying problems and make any other necessary updates.

The new ERD draft is displayed in Figure 10. As done with the previous entity diagram in Figure 2, a high-level description of the tool’s ERD tables is below (only new or updated tables are described):

Constraint: Defines limitations faced by the organization. These limitations can influence the treatment of a risk or give rise to a security action themselves (This replaces and extends TreatmentFactor to some extent)
ConstraintSecurityAction: Defines the constraints addressed by a security action. It also defines the level of coverage provided by the security action
ConstraintType: Lists the generic types of constraints
ProjectRiskSecurityAction: Defines the risks which a security action addresses. It also defines the level of coverage provided by the security action.

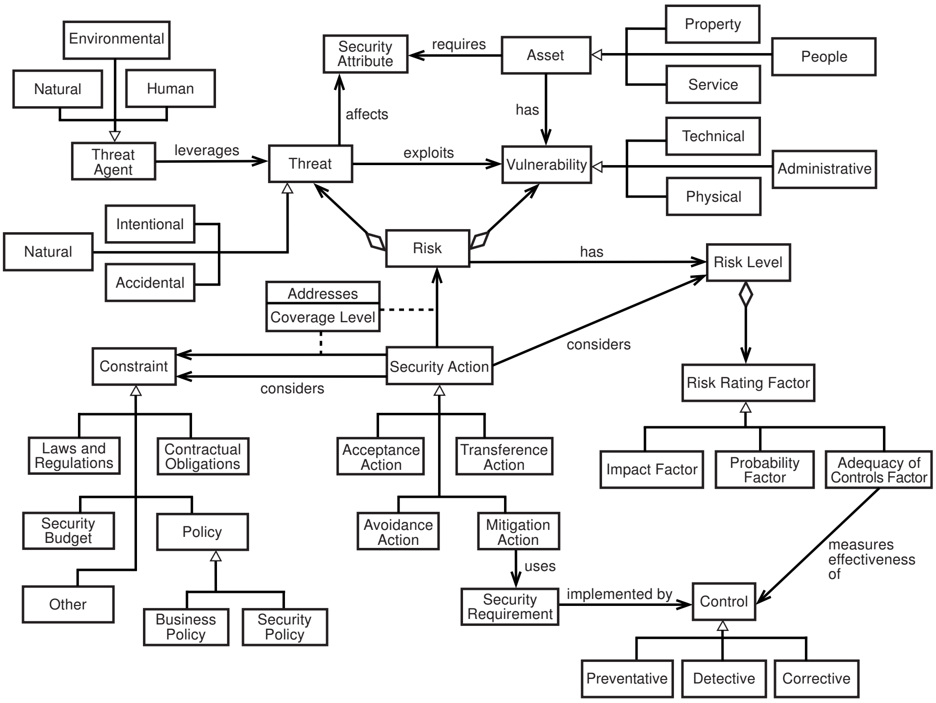


Figure 9: Draft of the new ontology

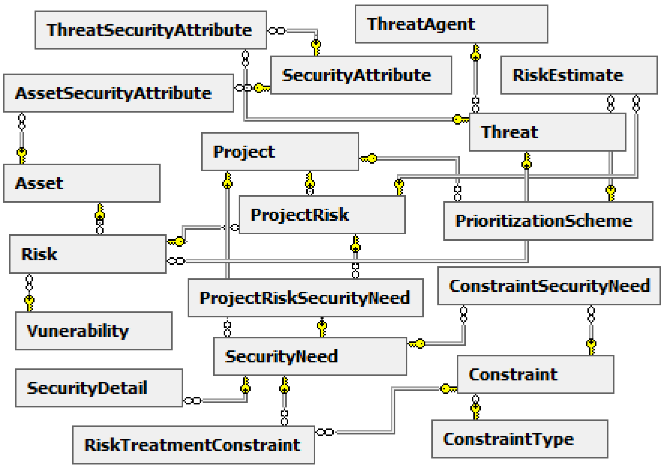


Figure 10: Draft of the new ERD to support the tool

RiskTreatmentConstraint: This defines constraints (e.g., laws, business policies, budgets and so on) that influence a risk's treatment, i.e. a security action
SecurityDetail: Defines measures used to implement security actions. Mitigation security actions are reflected in security requirement fields whereas transference, avoidance and acceptance security actions are reflected in the generic security detail field
SecurityAction: Defines any way in which to address a risk, or a constraint to a organization or system. This concept is an extension of the previous ERD's SecurityAction

One general foreseen weakness of the updates suggested above is that as risks no longer form the sole basis for security actions, the tool's current matching of risks to compare actions will no longer be fully adequate. Possible options to be investigated for comparing security actions based on non-risk components (primarily these will be Constraints) are centered around the matching of constraint groups or some other common constraint denominator. For example, comparing Security actions based on the types of Constraint that they address. These Constraint types could be laws or budget-related as seen in Figure 9, or more detailed and thus focus on data privacy, specific system types, or even organizational limitations.

The general incompatibilities and difficulties in mapping in previous sections do raise very interesting questions concerning how the tool and ontology would handle other RM/RA methodologies. For example, there is the reality that other methodologies will have varying concepts, approach risks in different ways and so on. Any future mapping techniques to be developed therefore need to more aptly appreciate this and either: provide some generic way for differences to be included; allow for an extensible framework where support for new approaches can be plugged in; or enable these concepts to be treated somehow externally to the system. The reflections in this section supply a good start for further work.

7 Conclusions and Future Work

The focus of this paper was assessing the compatibility of our proposed tool and the ontology which it embodied, with RM/RA methodologies used by businesses today. In particular, this paper aimed to extend the work in [7] by providing more detail on the thorough evaluation conducted. As stated previously, a good level of tool compatibility was imperative given the necessary interactions between the tool and these RM/RA approaches expected within BOF4WSS. From the comparison and mapping evaluation completed, it was seen that a majority of the findings were in support of the tool/ontology compatibility. In some situations however, noteworthy shortcomings of the tool/ontology were discovered and in these cases, updates were drafted to address them. Broadly considered nonetheless, the tool and the Solution Model it implemented were seen to aid in resolving a number of the transitional issues businesses face in coming together for negotiations in the framework.

Future work will consist of first updating the tool based on findings and reflections in this paper. Secondly, we aim to conduct interviews with security professionals to gather their feedback on the value of the tool from a practical perspective. Then, once final prototypes are developed, our research will assess 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] J. R. Nurse and J. E. Sinclair, "Securing e-Businesses that use Web Services – A Guided Tour through BOF4WSS," *International Journal On Advances in Internet Technology*, vol. 2, no. 4, pp. 253–276, 2009.

[4] C. Gutiérrez, E. Fernández-Medina, and M. Piattini, "PWSSec: Process for web services security," in *IEEE International Conference on Web Services*, Chicago, IL, September 2006, pp. 213–222.

[5] A. Singhal, T. Winograd, and K. Scarfone, "Guide to secure web services (NIST SP 800-95)," National Institute of Standards and Technology (NIST), Tech. Rep., 2007.

[6] C. Steel, R. Nagappan, and R. Lai, *Core Security Patterns: Best Practices and Strategies for J2EE^{TM}, Web Services, and Identity Management.* Prentice Hall PTR, 2005.

[7] J. R. Nurse and J. E. Sinclair, "Evaluating the Compatibility of a Tool to Support E-Businesses' Security Negotiations," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2010, WCE 2010*, London, UK, 2010, pp. 438–443.

[8] J. R. Nurse and J. E. Sinclair, "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, 2010, pp. 13–18.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Y.-P. Fu, K.-J. Farn, and C.-H. Yang, "CORAS for the research of ISAC," in *International Conference on Convergence and Hybrid Information Technology*, 2008, pp. 250–256.

[15] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems (NIST special publication 800-30)," National Institute of Standards and Technology (NIST), Tech. Rep., July 2002.

[16] CERT Coordination Center (CERT/CC), "OCTAVE® information security risk evaluation," n.d., http://www.cert.org/octave/ (Accessed 13 September 2010).