THE UNIVERSITY OF
WARWICK

AUTHOR: **Jason Ricardo Corey Nurse** DEGREE: **Ph.D.**

TITLE: **A Business-Oriented Framework for Enhancing Web Services Security for e-Business**

DATE OF DEPOSIT: .....

I agree that this thesis shall be available in accordance with the regulations governing the University of Warwick theses.

I agree that the summary of this thesis may be submitted for publication.

I **agree** that the thesis may be photocopied (single copies for study purposes only).

Theses with no restriction on photocopying will also be made available to the British Library for microfilming. The British Library may supply copies to individuals or libraries subject to a statement from them that the copy is supplied for non-publishing purposes. All copies supplied by the British Library will carry the following statement:

> "Attention is drawn to the fact that the copyright of this thesis rests with its author. This copy of the thesis has been supplied on the condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's written consent."

AUTHOR'S SIGNATURE: .....

USER'S DECLARATION

1. I undertake not to quote or make use of any information from this thesis without making acknowledgement to the author.
2. I further undertake to allow no-one else to use this thesis while it is in my care.

DATE SIGNATURE ADDRESS

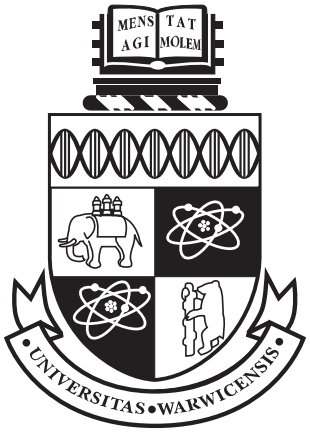
.....

.....

.....

.....

.....



A Business-Oriented Framework for Enhancing Web Services Security for e-Business

by

Jason Ricardo Corey Nurse

A thesis submitted in partial fulfilment of
the requirements for the degree of

Doctor of Philosophy
in Computer Science

Department of Computer Science

October 2010

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

PhD:- Doctor of Philosophy (formal), Piled Higher & Deeper (phdcomics.com), Permanent Head Damage (grad school slang)

Regardless of the meaning preferred, completion of my PhD degree could not have been achieved without the help of the persons I acknowledge below.

First of all, I would like to thank my supervisor Dr. Jane Sinclair for her exemplary guidance, patience and support. Her direction and encouragement played a crucial role in the successful and timely completion of this thesis. I would also like to express my thanks to Dr. Mike Joy, my second supervisor. Mike has always been more than willing to offer advice and constructive feedback on my research.

My special gratitude goes to Dr. Darren Mundy for accepting the request to be the external examiner for my thesis.

Other persons that deserve my appreciation include, my CS333 office mates (for 2007-2010), fellow patrons of DCS Coffee Club, doctoral colleagues throughout Warwick, and of course my friends in the UK and back home in Barbados. Each of these persons has played an integral part in my experience and as a result, the conducting of my research and final composition of this thesis.

Second to last, I must thank my family; Dad, Mom and Sis especially. What would I do without your direction, encouraging words, prayers, and calming spirits. No one could ask to be tied to a better lot. :-)

Finally and most importantly, I thank **God** for everything.

Declarations

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated. Parts of this thesis have been published previously as follows.

The framework in Chapter 3 was published in a conference paper in [138], and a much more detailed version was published in a journal article in [139]. The proposed model in Chapter 5 has been published in [144] along with some sections of the prototype system from Chapter 7. Chapter 6's work has also been published in conference proceedings [140]. An overview of the findings from the evaluation in Chapter 8 have been published in [142]. Two journal articles ([141] and [145]) have been submitted which go into detail on separate parts of Chapter 8 and extend [142]. Finally, Chapter 9's evaluation and its results were submitted as an invited journal paper in [143].

Abstract

Security within the Web services technology field is a complex and very topical issue. When considering using this technology suite to support interacting e-businesses, literature has shown that the challenge of achieving security becomes even more elusive. This is particularly true with regard to attaining a level of security beyond just applying technologies, that is trusted, endorsed and practiced by all parties involved. Attempting to address these problems, this research proposes BOF4WSS, a Business-Oriented Framework for enhancing Web Services Security in e-business. The novelty and importance of BOF4WSS is its emphasis on a tool-supported development methodology, in which collaborating e-businesses could achieve an enhanced and more comprehensive security and trust solution for their services interactions.

This investigation began with an in-depth assessment of the literature in Web services, e-business, and their security. The outstanding issues identified paved the way for the creation of BOF4WSS. With appreciation of research limitations and the added value of framework tool-support, emphasis was then shifted to the provision of a novel solution model and tool to aid companies in the use and application of BOF4WSS. This support was targeted at significantly easing the difficulties incurred by businesses in transitioning between two crucial framework phases.

To evaluate BOF4WSS and its supporting model and tool, a two-step approach was adopted. First, the solution model and tool were tested for compatibility with existing security approaches which they would need to work with in real-world scenarios. Second, the framework and tool were evaluated using interviews with industry-based security professionals who are experts in this field. The results of both these evaluations indicated a noteworthy degree of evidence to affirm the suitability and strength of the framework, model and tool. Additionally, these results also act to cement this thesis' proposals as innovative and significant contributions to the research field.

Abbreviations

B2B	Business-to-business e-commerce
BOF4WSS	Business-Oriented Framework for enhancing Web Services Security for e-business
CORAS	Consultative Objective Risk Analysis System
EBIOS	Expression des Besoins et Identification des Objectifs de Séurité, or Expression of Needs and Identification of Security Objectives
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ISS	Interaction Security Strategy
IT	Information Technology
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OMG	Object Management Group
OWL	Web Ontology Language
QoS	Quality-of-Service
RA	Risk Assessment
SADML	Security Action Definition Markup Language
SASaCS	Security Action Specification and Comparison System
SLA	Service-Level Agreement
SOA	Service-Oriented Architecture
RM	Risk Management
WM	Waterfall Model
WS	Web services
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Chapter 1

Introduction

If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology. — Bruce Schneier

1.1 Background and Motivation

Electronic business (hereafter, e-business) has matured into one of the most cost-efficient and streamlined ways of conducting business today. Within this business paradigm, one area which has been doing particularly well is the electronic collaboration between e-businesses [108], with service offerings such as online order processing and electronic payments. To facilitate this collaboration, Web services (WS) technology is playing a progressively significant role [231, 217, 74].

The novel benefit available with WS is rooted in its ability to allow for seamless integration of business processes across disparate enterprises, due to the use of open technologies backed by standardized protocols [29, 28]. Papazoglou [157] even states that the facilitation and automation of these business processes is the ultimate goal of WS technology. As the use of WS thrives however, ensuring adequate levels of security for these service offerings emerges as a critical goal. Even in this arguably early stage of worldwide WS adoption, the literature has identified numerous security challenges and issues [189, 92, 197] which threaten the use of WS for business. Security throughout this thesis is

defined as the degree to which harm to some object is prevented, reduced and properly responded to [60].

In an attempt to address new security challenges accompanying WS, standards setting bodies (OASIS [153] and W3C [225]) have proposed numerous pioneering standards. These standards aim to both solve problems caused by common threats and also to further the WS paradigm by enabling substantially more dynamic security interactions between services. As WS matures however, the move from lower level security details such as standards and technologies, to higher level considerations is imminent [189]. Security, irrespective of the context, is a multilayered phenomenon constituting other aspects such as processes, methodologies and procedures. This factor is especially true in the case of WS which as Hartman et al. [77] note, adds significant complexity to the e-business security landscape by making security a much broader and comprehensive concern.

Considering the points made above, this research focuses on identifying a novel, business-oriented approach to guide interacting companies in achieving and maintaining agreed levels of security across their enterprises. The approach envisioned is such that it could be used by companies in a collaborative manner, to tackle and manage the comprehensive concern that security in the WS business environment has become.

1.2 Statement of the Problem, Research Aims and Objectives

The security of Web services technology is a complex and very topical issue [189, 218, 36, 51]. When considering the use of this technology to support interacting e-businesses, the challenge of achieving security becomes even more elusive [158, 17, 77, 157, 197]. This is especially true with regard to reaching a level of security beyond the technological layer, that is trusted and acceptable to all collaborating entities for the duration of the e-business interactions.

To address this problem, this research aims to:

- Consider security in a business context and, with regard to its various components (policies, processes, technologies and so on), to develop a joint approach in which collaborating e-businesses could achieve an enhanced and a more comprehensive security solution for their Web services interactions; and
- Evaluate the suitability and strength of this approach's proposals in aiding businesses to reach the requisite levels of enhanced inter-organizational security and trust.

Specifically, the objectives are to:

- Examine the field of e-business security and also to assess approaches for achieving Web services security within the e-business context.
- Develop a business-oriented framework which companies can use to guide them in working with business partners to achieve an enhanced and a more comprehensive security solution for their Web services interactions. To some extent, this solution would also aim at fostering trust across these interacting companies.
- Identify and concentrate on a specific stage/problem within the framework for further in-depth analysis.
- Propose a detailed approach to support businesses' use of the framework at the identified stage.
- Develop a prototype system which implements the aforementioned proposal.
- Evaluate the detailed approach embodied in the prototype using a mixture of methods to determine its suitability in supporting the relevant stage.
- Conduct an evaluation of the business-oriented framework proposed using qualitative methods to gather feedback on whether it might aid businesses to jointly reach the desired levels of security and trust.

– Analyse the respective evaluation's findings from both objectives above and use these to make conclusions as to (i) the adequacy of the detailed approach proposed in supporting the relevant stage, and (ii) the viability of the types of activities the framework generally purports in aiding businesses to achieve the desired levels of enhanced inter-organizational security and trust.

1.3 Scope and Limitations

The business-oriented framework initially proposed intends to be very broad in research focus. It aims to provide a comprehensive guide which businesses can use from the early stages of considering WS use, to the design and development of a trusted cross-enterprise security solution. To properly evaluate this proposal however, ideally all components would have to be validated. This would necessitate having partnering businesses willing to use the framework, a reasonable length trial period to gather data pertaining to its functional use and lastly, a case study analysis to determine its viability. These aspects, noting the magnitude of such a task and the lack of industry contacts possessed by the principal researcher, placed serious practical restrictions on any attempt to evaluate this proposal in its entirety during this doctoral course.

As a result of these considerations, the core of this research narrows in its scope to concentrate on a particular, more manageable sub-problem within the framework. By analysing that sub-problem in depth, assessing specific research-based issues and providing critically evaluated solutions to those issues (appreciating restrictions above and thus independent of any case study analysis), this would act to do two things. First is to add a more conclusive level of research to this thesis and second, by extension, is to support the suitability and strength of the overarching framework's proposals. To supplement this work and reinforce the framework as a research contribution, a small evaluation was also sought through the use of interviews with industry-based professionals with the aim of gaining expert feedback on the framework and what it proposes. These interviews

however, focused only on gathering insightful feedback, as opposed to a thorough and detailed assessment, as this would only be possible with a full case study.

1.4 Research Methodology

To achieve the aims and objectives above in a valid research manner, defining a clear and appropriate research methodology was crucial. In simple terms, a research methodology is a systematic and well-structured way in which a research problem is solved [105]. This includes the underlying research methods chosen, questions asked, data collected and techniques used for data analysis.

The methodology adopted by this research constituted of a combination of methods and processes, each of which was considered to be best suited for the particular research stage and task. Generally however, the definition of problem statements and research questions throughout this research are all based on critical analyses of the relevant literature. For the first research contribution which deals with the creation of a business-oriented framework therefore, the necessary literature was examined and the framework designed to address the key problems highlighted. The creation of the framework also was guided by a standardized development model to support its structure and process.

With appreciation of the scope and limitations (as outlined in Section 1.3), the research focus then shifted to a more manageable research problem within the framework's context. This problem was drawn from the analysis of an informed scenario and then backed up by existing literature. From this, a detailed approach to address that problem and its related research questions was proposed. This approach was then implemented as a prototype and evaluated in terms of its compatibility with existing approaches and tools used in business today. This was important as one of the key envisioned benefits of the framework was its ability to allow companies to plug in and use their own techniques during various stages in the overall approach. Strong compatibility would support the completeness of key components of the proposed approach and give evidence to show that it

could adequately fit in and work alongside currently used techniques.

Considering that both the business-oriented framework and detailed approach were intended for direct use in industry, it was important that the overall evaluation processes considered this as much as possible. To allow for an industry-based evaluation of the framework and complement the completed compatibility evaluation of the detailed approach, interviews were conducted with a number of industry-based security professionals to gain practical feedback on both proposals. This strengthened the previous evaluation of the detailed approach (and therefore considered a wider scope than just compatibility) and provided an evaluation of the framework which appreciated the research limitations.

The last aspect of the research methodology was the examination of data gathered. This involved the application of data analysis techniques such as counting and measurement (in terms of testing prototype compatibility) and content analysis and coding [12] (for processing interviewees' feedback). These established techniques enabled data to be systematically assessed and representative observations made. The evaluation results that emerged were then used to complete the project aims and objectives, and make appropriate conclusions on this research.

1.5 Thesis Outline

To achieve the research objectives, the thesis structure is as follows.

Chapter 2 begins the core of the research with a literature survey into the fields of e-business, Web services, and their security. This chapter identifies gaps within existing research, providing motivations for the current work.

Chapter 3 targets some of the outstanding research issues identified in the previous chapter and proposes a business-oriented solution approach. Formally, the name given to that approach is the Business-Oriented Framework for enhancing

Web Services Security for e-business, hereafter BOF4WSS. Throughout the chapter, detailed coverage is given of the framework to exhibit its uses, application and scope. This framework constitutes one of the main thesis contributions.

The purpose of **Chapter 4** is to supplement the previous chapter's largely abstract discussion by providing a comprehensive example of how BOF4WSS would work in practice. Specifically, this consists of applying BOF4WSS to a developed scenario. This application is expected to add to the practicality of the framework's proposals and highlight some of its uses and benefits. This scenario is also reused in numerous discussions throughout the thesis.

With appreciation of the project limitations, **Chapter 5** narrows the focus to a more manageable research problem still within the confines of the framework's investigation. To aid in this task, the scenario from the previous chapter is analysed for a more suitable research area. Having determined that smaller, specific problem, a detailed approach—called the Solution Model—is proposed to address it and therefore support companies' use of BOF4WSS.

Chapter 6 continues the presentation of the Solution Model by reporting on the implementation of some of its more conceptual components. These form the foundation for a number of later aspects and Model implementations.

Chapter 7 covers the remaining Solution Model components and presents a system implementation of the proposed Model. For ease of reference, this implementation is called the Security Action Specification and Comparison System, or SASaCS. The prototype of SASaCS to be developed would serve two purposes. First is to act as a proof-of-concept to the Model's theories and second is to allow for the Model's practical evaluation.

Chapter 8 is the first of two evaluation chapters. The aim in this chapter is

to critically assess whether the system implementation (SASaCS) and the underlying Solution Model to support the framework are compatible with existing security approaches used by companies today. Testing the interplay and compatibility of approaches is critical considering that BOF4WSS allows companies the flexibility to plug in their own approaches and techniques at various framework stages. All evaluation findings are presented and discussed.

Chapter 9 reports on the final stage of this research's evaluation process. This chapter provides an evaluation of BOF4WSS and the Solution Model (by way of SASaCS) through the use of interviews with industry-based security professionals. The feedback from these persons is key to this research as it supplies an objective third-party assessment from professionals in areas where the framework and Model are ultimately to be applied (that is, industry). Throughout the chapter, findings and conclusions regarding the proposals are also discussed.

Chapter 10 builds on the evaluated research proposals and briefly investigates a novel approach to further develop the Solution Model. The emphasis in this chapter is not on defining a critically evaluated approach but rather exploring its use and looking at some of the interesting research issues that emerge. These will flow into future work.

Chapter 11 concludes the thesis and summarizes the research project. Also included in the chapter are the main contributions of this project to the research field. Amongst these are BOF4WSS and the Solution Model. The final section outlines future research directions that could both improve and further the ideas embodied in this work.

1.6 Summary

This chapter introduced the research which culminated in this thesis, inclusive of the project aims and objectives, scope and limitations, and research methodology employed. To recap the general theme, the core focus of this research was towards two aspects. These are, the development of, and investigation into an approach that would enable interacting e-businesses to achieve a more comprehensive and trusted security solution for their Web services interactions.

Chapter 2

E-Business, Web Services, and their Security: State of the Art

While Web Services make it possible for applications to interoperate, they complicate the security landscape. A new dimension is added to the security problem.
— Bret Hartman, Donald Flinn, Konstantin Beznosov and Shirley Kawamoto

2.1 Introduction

Having introduced the goals of this project in Chapter 1, Chapter 2 provides the theoretical foundation for this research by conducting a thorough literature survey. This survey critically assesses the state of the art in the fields of e-business, Web services (WS), and their security, with the aim of identifying gaps in research and practice which would motivate this research.

Considering the vast amount of literature in the fields of interest, this chapter begins by presenting a review of e-business and WS, independent of security. This enables these topics and particularly WS use for e-business to be clearly discussed. Key aspects of emphasis include the significance of e-business and the novel benefits of using WS to support cross-enterprise e-business interactions.

Building on this foundation, the security situation with regards to these two fields is then explored. This consists of a detailed analysis of the current approaches, frameworks, procedures and technologies proposed to address e-business

and WS security. Apart from being a contribution in itself, this comprehensive literature review also leads to the identification of outstanding research issues. It is some of these issues, specially around layered and collaborative security, which form the motivation for this thesis.

2.2 Review of E-Business and Web Services

2.2.1 E-Business and Business-to-Business E-Commerce

In today's globally connected world, e-business is one of the most active topics in the business and Information and Communication Technology (ICT) fields. With this in mind, it should come as no surprise that the term itself is not unequivocally defined. For example, some authors [108] associate it with internal operations, others [132, 232] with both internal and external business processes, and yet others using it interchangeably with electronic commerce as mentioned in [25]. The penultimate perspective is preferred in this thesis as it imparts a more precise and complete description, and one which is supported by a wider literature base (such as [132, 232, 95, 38]).

The definition assumed in this article for e-business therefore is, "the carrying out of business activities that lead to an exchange of *value*, where the parties interact electronically, using network or telecommunications technologies" [95] (p.83). This definition is preferred because it encompasses all possible parties—both internal and external, and because it specifies the exchange of *value* as opposed to only goods and services. This is a key indicator often used to describe the less generic term of electronic-commerce or 'e-commerce'. With the meaning of e-business outlined, this thesis considers its significance and the value of its most dominant category; business-to-business e-commerce (see [107, 120, 41] for detail on this category).

The importance of e-business has arisen due to its phenomenal growth spanning both industry [108] and research within academia [132]. This growth, especially within business-to-business e-commerce (hereafter B2B), has been fu-

eled by a myriad of potential benefits for companies that adopt this business model. These range from accelerated interactions between suppliers and vendors, to fully automated processes leading to vastly improved operational efficiencies [232, 39]. The use of electronic technologies however is not only being adopted due to their benefits but in some cases with the sole objective of remaining competitive. Porter [165] even extends this view and highlights that the economy is approaching a state where Internet applications will be mandatory assets for a company's basic survival.

The significance of B2B is best expressed by its size and remarkable sustained growth. Laudon and Traver [108], commenting on market projections, note that money spent in B2B is estimated to soar to US$6.3 trillion by 2012. This shows a steady growth when compared to the US$3.8 trillion expected in 2008. Two disciplines in particular where researchers and industry professionals have stressed the use and popularity of B2B-type applications are Supply Chain Management (SCM) and e-procurement [232, 186]. In simple terms, both of these processes focus on the optimization of activities involved in the purchasing and selling of products. The prime lure to these two areas is the promise of boosted internal and external process integration and automation. This would result in greater transaction accuracy, less administrative overheads and enhanced operational efficiencies [30, 35, 79].

As Laudon and Traver aptly observe, "If even just a portion of inter-firm trade could be automated, and parts of the procurement process assisted by the Internet, then literally trillions of dollars might be released ... This is the promise of B2B e-commerce" [107] (p.682). Visions like these have perpetuated various highly interconnected hybrid business approaches. One of the most remarkable being the *extended enterprise*, which even though not distinctly defined [71], maintains the common characteristic of a set of interacting businesses (usually within the supply chain) that pursue repeated transactions and have relatively strong ties at organizational and ICT levels [71, 40, 49].

To support these grand visions for business online, there is an implicit

need for a sturdy technological infrastructure. To date, numerous B2B enabling technologies have been proposed. Some of the most popular and relevant to this research are Electronic Data Interchange (EDI), e-Business eXtensible Markup Language (ebXML) [50], RosettaNet [175] and WS. In this thesis the focus is on WS and therefore this is discussed next.

2.2.2 Web Services as a B2B Enabling Technology

Similar to e-business, there have been various definitions put forward for WS, some [158, 223] very detailed and others [197] straight to the point. Instead of engaging too much in this debate, this thesis adopts a simple definition and focuses on a number of the most important tenets outlined in the literature. WS is therefore defined as a distributed systems technology suite which emphasizes qualities such as flexibility and ease of deployment [15]. Other key and widely accepted tenets include being loosely coupled, self-contained, platform-independent, network-accessible and based on open standards [30, 82, 167, 234]. All of these are contributors to why WS is increasingly regarded as the de facto implementation technology for integration projects and distributed computing paradigms such as Service-Oriented Architecture (SOA). SOA is generally defined as a logical way of describing systems in terms of publishable, discoverable and encapsulated services (details in [156]).

One point which merits clarification is the slight difference between a *service* in an SOA and a *Web service (WS)*. A *service* (in an SOA) is assumed to take on more of an abstract meaning and thus can be regarded simply as a distinct or packaged unit of logic or functionality (similar to work in [51]). A *Web service (WS)* however is the technology- or standards-level implementation of a *service* or part thereof. A service is therefore viewed as the conceptual prerequisite to the actual standards-based Web service implementation. Other than this difference, a number of the tenets identified for a Web service also apply to a service. It was important to identify this disparity for the clarity of this research, but additionally to stress that the SOA and its inclusive services, can be implemented

using technologies other than the WS suite. Examples, as mentioned by [155] are Jini [93] and Open Grid Services Architecture [70].

At the basic level, WS is constituted of three core technologies, all based on the eXtensible Markup Language (XML). They are: SOAP—a messaging protocol for exchanging information such as documents and instructions between Web services [22]; Web Services Description Language (WSDL)—a specification schema for describing the publicly accessible interface of a Web service [157]; and Universal Description, Discovery, and Integration (UDDI)—an initiative to create (i) a registry standard for Web services description and discovery and (ii) a respective registry facility that supports the publishing and discovery processes [157].

As WS has matured, numerous other related technology specifications have been proposed. An area in particular which has been heavily targeted is WS use for online business and providing the technologies to support secure and reliable business interactions and process execution between companies. This in many ways validates the view held by some authors (such as Papazoglou [157]) that a central aim of WS is in the facilitating and automating of internal and external business process collaborations.

To consider WS use for business in more detail, there are two standards specially designed to facilitate the more complex, long-running service interactions that would make up business processes. These are, the Web Services Business Process Execution Language (WS-BPEL) and Web Services Choreography Description Language (WS-CDL). WS-BPEL (BPEL hereafter) is a language that allows for the specification of business process behaviour based on Web services [152]. Amongst other things, it provides an executable language that can be run by software engines to orchestrate internal message, control and data flows. WS-CDL addresses yet a higher WS layer, and supplies a standard mechanism for defining WS collaborations and the choreographies of message exchanges from a global viewpoint [224].

In simple terms, BPEL describes WS interactions from the perspective of *one endpoint* and thus is ideally used internally to direct service and message

flows. WS-CDL however focuses on the global view (not from any endpoint's perspective) and defining the expected behaviour of *all WS participants* involved in a business collaboration. Papazoglou [157] can be referred to for a more in-depth discussion on WS-CDL and BPEL with examples. To put the standards discussed into context, Figure 2.1 is included.

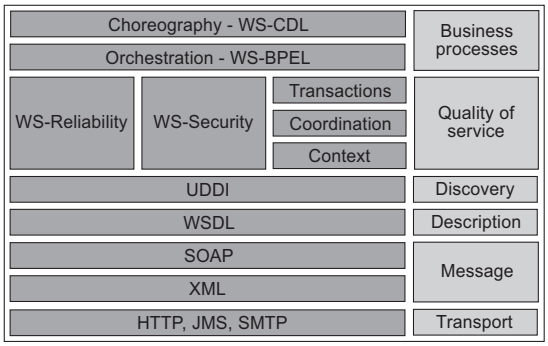


Figure 2.1: The Web services technology stack (adapted from [157])

Returning to the usage of WS, its impact has been such that Zhang [231] strongly argues that WS is "changing the face of the Internet", and transforming it from the traditional image as just a data repository. As the author stresses, WS promotes this transformation in three ways: they facilitate easier B2B; provide a uniform framework for distributed computing; and lastly, present a cost-effective approach to quickly develop and deploy dynamic Web applications. This listing helps to highlight why the three main usage scenarios for WS as outlined by Zimmermann [234] are enterprise application integration (EAI), B2B and common services; with B2B being identified in popular texts (Alonso et al. [4]) as the ultimate goal of WS. One real-world example of WS use in B2B is W3C's travel agency scenario (discussed in detail in [221]) illustrated in Figure 2.2.

This scenario features: a travel agent which offers customers the ability to book complete vacation packages; an airline and a hotel company that provide the flights and hotel rooms respectively; and a credit card company that provides the guarantee for customer payments. The use and core benefit of WS in such a B2B-type communications scenario is the vast automation it can easily enable.

Assuming a situation where each business exposes its internal systems

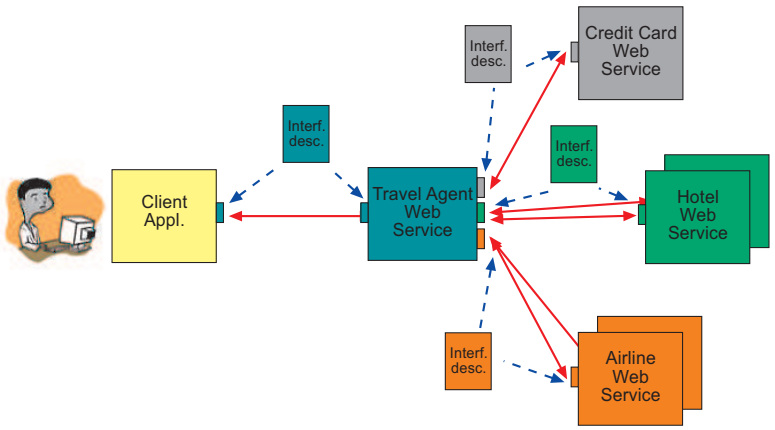


Figure 2.2: Travel agency Web services scenario (adapted from [222])

as Web services, the travel agency Web service could query these services in real-time, using preferences provided by the end consumer to access and book complete vacation packages. There is no longer the need for faxes, phone calls and other manual mechanisms. This is the basic concept being illustrated in Figure 2.2. To apply the standards outlined; WSDL provides the service interface description, SOAP is used for messaging (red arrows), WS-CDL could specify the expected behaviour of all companies' Web services (from a global viewpoint) for this booking process, while BPEL would be used by companies to orchestrate their respective internal process, service and data flows.

A noteworthy advantage of applying WS to the travel agency scenario is the implicit ability to wrap or encapsulate existing legacy systems and provide a standard application programming interface to allow these systems to be much more easily accessed. This ability to 'service-enable' existing systems is a prime application scenario of WS in business. Other scenarios include using a Web service as a (i) self-contained business task, (ii) full-fledged business process, or (iii) a simple application (or program) [157]. All of these scenarios, plus the reality of seamless interaction at the systems level between disparate companies, accounts for why WS use for B2B has become so prominent.

This ideal link between WS and B2B however, even though largely accepted, is not always shared by researchers. Possibly the most fascinating source of deliberation—to be discussed in subsequent paragraphs—pertains to the suit-

ability of WS usage for B2B [226]. The main issues here attributed to innate qualities of WS and its perceived lack of maturity (standards-wise) in supporting the complex business interactions and processes common to B2B.

Regardless of these opposing perspectives, WS use for intra- and inter-enterprise integration and B2B are applications that continue to flourish. This point is supported by industry surveys as Gutiérrez et al. [74] report, by academic research in [231, 29] and also by other sources such as work in [168] by Pulier and Taylor. The driving forces behind this growth are plentiful and range from strong industry endorsement and marginal investment costs, to the ability to integrate and reuse legacy systems while enabling faster, more flexible integration between trading partners with disparate systems [29, 168]. These are the types of novel benefits offered by WS to B2B in general, and specifically to elements such as SCM, the procurement process and a plethora of other integration projects [29, 82, 119]. It is these benefits that set WS apart from any other integration or distributed computing technology.

Albeit a promising enabler and integrator, WS does not come without its caveats. In the broad context, drawbacks include: performance issues—XML, a core WS specification, is by nature a sizable data format [14]; interoperability and standards confusion—WS is still maturing and thus standards are still in development, this is compounded by the numerous bodies (OASIS, W3C, Liberty Alliance) developing what are at times overlapping or non-interoperable standards [189, 61]; and lack of product support—there have been indications of a lack of WS products which support key application features (for example, non-functional requirements) [233].

In the context of B2B, the literature suggests three core types of limitations. The first as discussed by Goethals et al. in [71] is generic and is primarily associated with the existence of communication gaps (technical and nontechnical) between businesses. This typically arises from the classic problems in business-ICT alignment and critical issues such as agreeing on semantics. This issue as it relates to WS is further supported by Alonso et al. [4] and validated to some

extent by an empirical study by Prokein et al. [167]. In that latter study the authors conclude that technical and economic integration problems constituted the main challenge to WS use for intercompany cooperation.

The second limitation is specific to WS and focuses on its innate characteristics and its perceived lack of maturity in supporting the complex, highly coordinated business processes in B2B. Yan and Klein [226] analyse this maturity problem in depth as they compare and evaluate WS and ebXML, and each technology's suitability for e-business. From that evaluation, the authors assert that real-life business is much more complicated than the collection of request/response pairs approach purported by WS. Moreover, they stress for WS to be more than a middleware technology, business standards for process orchestration, choreography and reliable coordination will have to be established. This limitation, even though not unanimously shared (see [234, 159] in terms of support for complex interactions), is a worthwhile consideration when adopting WS for complex B2B.

The final limitation deals with security in WS and will be discussed in subsequent sections. Below therefore, a brief look is taken at two other related and popular B2B enabling technologies; ebXML and RosettaNet. Their general aims as compared to that of Web services are shown in Table 2.1.

2.2.3 Other Related B2B Enabling Technologies

Technology	**Aim**	**Suitability**
RosettaNet	E-marketplace business processes	Purpose built
ebXML	Universal business processes	Purpose built
Web services	Support business processes (Internal and external)	Additional standards being developed to support processes

Table 2.1: B2B enabling technologies standards stack (partially based on [157])

E-Business XML (ebXML) aims to provide a set of specifications to enable all types of enterprises to conduct business on the Internet [50]. As such, this technology was *designed specifically for e-business and B2B*, unlike WS which arguably has its roots in distributed messaging and integration technology [28,

226, 187]. The core value of ebXML is in its ability to offer a complete suite of standards for conducting business online, hence the 'universal business processes' aim in Table 2.1.

Whereas ebXML caters to all business types, RosettaNet is a vertical business standards set and is thus aimed at a specific industry, namely the supply chain. RosettaNet's goal is in universally standardizing common e-business processes in a particular marketplace. This would therefore enable vastly reduced setup times for new business partners and the great ease with which companies can switch between suppliers [175, 115].

Possibly the most intriguing fact about RosettaNet is that, because of its vision at the high level of industry processes, it has made use of existing, established standards for the lower level implementation-based aspects. Examples of this include use of the Business Process Specification Schema from ebXML—used to describe sequencing and choreography of processes, and the option of ebXML Message Service or WS—used for the required messaging capabilities [158, 115].

Next, this chapter explores the security developments in e-business and WS with the end goal of highlighting outstanding security issues.

2.3 Exploring the Security Situation

2.3.1 E-Business and B2B Security

Security has always been a serious consideration for businesses but with the widespread adoption of the Internet, its significance—whether voluntarily or not—has exponentially increased. By simply connecting internal networks to the Internet, businesses are susceptible to a variety of attacks [135, 31] and as recent surveys [166, 170, 171, 210, 33] exhibit, they are being exploited. Malware, viruses and even directed attacks (such as Denial of Services (DoS)) identified as some of the prime threats.

Within e-businesses where the aim is conducting business electronically, the risks faced are again drastically increased [85]. Businesses have to be cognizant

(in their planning and strategies) of the fact that the ubiquitous nature of the Internet means that attacks can occur at any time, from anywhere. Nachtigal and Mitchell [132] aptly stress that the openness to the environment which is the uniqueness of e-business also acts to be its real danger. A problem compounded, they argue, by the fact that traditional security approaches, even though no longer adequate in the era of e-business, are still in use and being developed.

In addition to the problems faced by e-businesses individually, in the connected B2B domain, the significance of security is also prevalent. One of the core drivers behind this is purely monetary and is linked with the enormous value (trillion-dollar market) of B2B and its continued high growth rate [108]. These factors plus the unique challenges already facing e-business make B2B a naturally alluring target for malicious parties. The second noteworthy driver is that as businesses attempt to work together thus forming extended networks, negotiating, accommodating and managing the security desires of each partner represents a formidable challenge [205]. This reality is highlighted by Tiller [205] when he states, "different partners have unique access requirements, want specific security policies in place, and have varying SLAs [Service-Level Agreements] and legal obligations, all leading to security mayhem" (p.68).

A third driver for B2B security is rooted in one of its core benefits, that is, the ability to facilitate very closely knit and highly automated and accelerated processes across enterprise boundaries. This is a point argued in the specific context of the e-supply chain in Baker et al. [6] and more generally in the extended enterprise context by authors in [49, 48].

The problem in these cross-enterprise situations as researchers stress is that as organizations increasingly rely on the Internet to closely join companies and support internal and external business processes, each firm's individual security decisions impact the overall security infrastructure for all the businesses it interacts with [48]. This perspective is validated by an independent survey in Baker et al. [6]. The most significant finding of that survey however is the establishment of a positive correlation between the degree of collaboration (between supply chain

partners) and the prevalence of IT security incidents and risk. Common examples of these incidents being unauthorized network access, data theft and malicious code infections, all linked to business partners as the source.

The dilemma faced by businesses therefore is that in fostering the close relationships to enable streamlined interactions, they have also made themselves susceptible to peculiarities in their partners' security posture. This consideration is novel in that it emphasizes what is in essence a complex security problem as businesses try to (i) protect themselves individually and (ii) devise a strategy to protect collections of legally autonomous businesses, when aiming to form closely knit hybrid organizations like the extended enterprise. The anatomy of the extended enterprise provides just one example of how B2B itself can increase the complexity of the security problem.

With the significance of the security problem outlined, it is worth assessing how security has been achieved in these online systems thus far. In examining this topic, Padmanabhuni and Adarkar [155] note that to achieve security in effect means the satisfaction of a collection of implicit security objectives/requirements. These requirements for online systems are stated broadly to be confidentiality, data integrity, authentication, authorization, non-repudiation, privacy, trust, availability and intrusion detection.

To consider how well the requirements listed have been met in the e-business arena, Padmanabhuni and Adarkar [155] state that the current set of available security technologies is proven to be adequately able to handle these requirements. Examples of these technologies include passwords, encryption, digital signatures, Public Key Infrastructure (PKI), traditional network-level firewalls and Secure Sockets Layer (SSL). Boncella [17] agrees with this perspective as he argues that SSL, PKI and firewalls are able to meet the technical-level security requirements for conventional Web traffic over Hyper-Text Transfer Protocol (HTTP). Work in [100] is yet another research reference that supports this view.

A salient point made by Katsikas et al. [100] which is of great relevance to this research is that if technology exists to solve e-commerce security problems,

why do breaches persist at such alarming rates? Their answer to this is founded in the reality that security is not only a technical or physical concept. Specifically, they state, "... while everyone recognizes the need for securing e-commerce, what they do not know is that security is more than erecting physical and electronic barriers. The strongest encryption and most robust firewall are practically worthless without a set of organizational security measures, built around a security policy that articulates how these tools are to be used, managed and maintained" [100] (p.556).

The remarks above help to identify that security, even in the highly dispersed e-business world, is much more than just technical solutions. This reality can also been seen in Dynes et al. [49] in terms of extended enterprise business collaborations. This is as companies look towards creating an appropriate security approach (for interacting partners) consisting of strategies, processes, systems, culture and incentives. Building on this more comprehensive view of security (thus, not only technologies), the work of Laudon and Traver [107], displayed in Figure 2.3 is cited. This model perfectly exemplifies the layered nature of security both in e-commerce and also, broadly to all business security.

Figure 2.3: The e-commerce security environment [107]

In Figure 2.3's layered model, laws and industry standards guide companies in security but also put regulations in place that enable security violators to be investigated and prosecuted. Secondly, organization policies and procedures mainly attempt to have rules and processes (or generally higher-level approaches) internally that enable the fulfilment of a company's security objectives/requirements. Finally, technology solutions are the specific, lower-level mechanisms that implement the security objectives/requirements for the data and systems. In practical

terms related to the previous paragraphs, organizational security measures such as a security policy would fit in the 'policies and procedures' layer, whereas encryption, digital signatures, PKI, traditional network-level firewalls and SSL would be part of the 'technology solutions' layer.

In the next section, the security aspects of WS as an enabling technology for e-business are discussed. Emphasis is placed on the significance of security, current approaches towards achieving it and its outstanding issues.

2.3.2 Securing e-Businesses that use Web Services

The substantial advantages to e-business that WS promises regrettably come at a high cost in the area of security. In [217, 194], authors stress that WS by its very nature creates a multitude of new security challenges. Apart from these views, companies in industry have also identified the importance of the security problem in the area of B2B as is seen in the study in [167]. This section identifies and briefly discusses three of the most significant challenges.

The first challenge faced in using WS for business-to-business interactions is that conventional mechanisms used to satisfy security requirements of normal e-business interactions fall short when applied to WS [17, 62, 122, 10]. Typical examples are SSL's inability to provide end-to-end security—SSL is only point-to-point; and the inadequacy of traditional firewalls to protect against XML-based threats—traditional firewalls cannot scan documents for included XML-based threats. Padmanabhuni and Adarkar [155] sum up the disparity between mechanisms in e-business and those in WS as they emphasize that the loosely coupled, dynamic nature of SOA (which can be also taken to apply to WS) necessitates additional security features and mechanisms.

Secondly, with the new technologies constituting WS, an abundance of new and adapted technical threats has surfaced, a reality worsened by the fact that WS was conceived primarily for interoperability, speed and convenience and not with security natively in mind. These threats endanger all aspects of the WS paradigm and target the range of security objectives.

Another important factor is that threats are not only targeted at the surface level (that is, the application directly interacting with the Internet), but at internal business applications as well. As organizations look to WS-enable their legacy systems to facilitate streamlined integration, they also provide a direct line and new avenue of attack into these systems [189, 197]. Publications in [189, 92, 218, 62, 228] together provide an extensive list of now common attacks and threats against WS. Further to this, some authors [233] even contend that there is a lack of products to aid in providing sufficient security against these threats.

The last challenge considers WS at somewhat of an overarching business level. In their work on WS, Hartman et al.[77] stress that despite its numerous benefits, WS adds significant complexity to the e-business security landscape. Security is now a much broader and comprehensive concern which cuts across business lines much easier and quicker than before. As such, an inadequate security posture in one company can become a real-time increased security risk for its partners—immediate and extended.

Due to the complexity with using WS, trust between businesses has also been identified as a related concern. Prokein et al. [167], for example, conclude that WS technology was primarily being used for connecting well-known transaction partners due to a lack of trust in using them with unknown businesses. Trust here and generally in this thesis is defined as the belief that a party's promise or word is reliable, and that a party will fulfil his/her obligations in an exchange relationship [185]. With just three challenges outlined, it is understandable that some industry professionals (such as Curphey [36]) have deemed WS 'a developer's dream and hacker's heaven'.

As was done in discussing e-business and B2B security, this section now examines how security is achieved when WS is used in business. Generally however, WS security techniques are the same regardless of where they are applied, business or elsewhere. This examination starts by considering the security requirements that lead towards the fulfilment of security.

At a basic level, WS is simply another technology that enables business online (e-business). Deductively therefore, all the security requirements for online systems (named in Section 2.3.1) still do apply when approaching security with WS. This point is seen in [194, 228] as they identify WS security requirements that are similar to e-business requirements, for example, confidentiality, integrity, authentication, authorization, auditing, and intrusion detection and prevention.

For WS specially, work by Steel et al. [194] extends the basic requirements above with three aspects. The first is Single Sign-On (SSO) and delegation. This is the ability to transparently handle authentication to multiple interacting services and also decentralized access controls. The second aspect is identity and policy management. This enables the sharing of identities and policies that spread across disparate systems and trust boundaries. Finally, there is security interoperability which in simple terms, ensures that the standards/protocols used are interoperable. All of these additions specifically target the unique security challenges accompanying the distributed, loosely coupled and highly dynamic WS technology suite.

Apart from analysing WS security requirements only, due to WS's close association to SOA, SOA-related security requirements also have proved to be applicable. In their work on SOA requirements for example, Padmanabhuni and Adarkar [155], like Steel et al. [194], show appreciation for SSO and delegation requirements. Additionally, Padmanabhuni and Adarkar [155] identify two more requirements: malicious invocations—having appropriate code inspection technologies to assess for malicious data in service invocations, and repeated invocations—ensuring mechanisms are in place to protect against repeated WS-specific attacks leading to denial of services. From the set of requirements covered in this and the previous paragraph, one can begin to grasp the complexities of providing even technical-level security to WS interactions. With these requirements outlined, the next step is to present how they are currently being handled (or the proposals publicized to handle them) in the literature.

To address the new security challenges and security requirements as men-

tioned above, consortiums such as OASIS and W3C have developed and ratified numerous standards. These standards aim to solve problems caused by common threats and also to further the WS paradigm by enabling substantially more dynamic security interactions between services. Due to the large number of these standards and their inherent complexities, this section does not aim to discuss them in detail. The intention instead is to provide a contextual overview.

Arguably the best and most intuitive approach to this review is to present standards according to the challenges and requirements they address. The National Institute of Standards and Technology (NIST) article in [189] (based on the work in [136]) is one work that provides a detailed categorization of security standards. The security dimensions NIST identifies are *secure messaging*, *resource protection*, *negotiation of contracts*, *trust management* and *security properties*. That article was chosen as the primary resource for the following overview mainly due to its extensive coverage and well established literature base. For clarity in presentation and to put some of the standards to be identified in context, Figure 2.4 has been included.

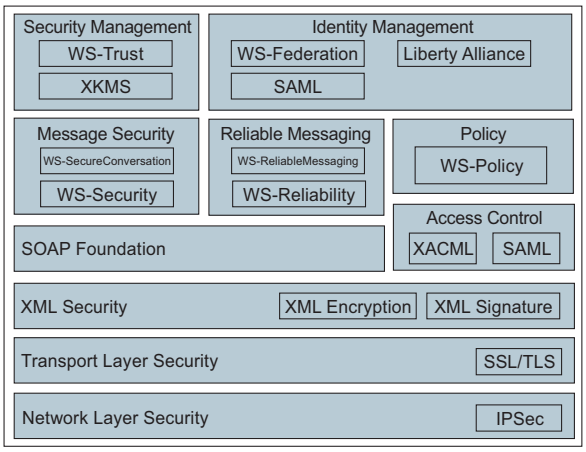


Figure 2.4: Web Services Security Standards: Notional Reference Model [189]

One of the primary goals of WS security is *secure messaging*. In this dimension, security specifications include SSL and Transport Layer Security (TLS) to secure the message at the transport layer and WS-Security (which leverages XML security techniques such as XML Encryption and XML Signature) to secure the

message at the SOAP level. These specifications satisfy the security requirements of authentication, confidentiality, non-repudiation and integrity. WS-Security specification in simple terms can also be thought of as a mechanism to support the security credential interoperability requirement [155].

At the *resource protection* dimension, thus considering the service as a resource itself, requirements for privacy are met by OASIS's eXtensible Access Control Markup Language (XACML), whereas ensuring only authorized use is addressed by eXtensible rights Markup Language (XrML) and again, XACML. The basis behind both standards is in providing a universal syntax for managing rights and authorization decisions.

To facilitate the technical level trust and *trust management* capabilities required between disparate services, various standards are proposed. These include: WS-Trust and XML Key Management Specification (XKMS) for establishing trust, Security Assertion Markup Language (SAML) and WS-Trust for trust proxying, and finally WS-Federation and Liberty Alliance ID-FF for federation, or loosely, 'sharing' of trust. Another way to view these standards as shown in Figure 2.4 is to associate WS-Trust and XKMS with security management, and SAML, WS-Federation and Liberty Alliance with federated identity management (or simply, identity management across trust domains). Research in [155] can supplement or at least simplify some of these descriptions as they specifically identify that SAML addresses the security requirements of SSO, and authentication and authorization interoperability, and XKMS addresses the need for XML-based PKI.

Beyond the topic of trust, *contract negotiation* is also regarded as a key dimension of WS security. In its requirement for registries and semantic discovery, the two highlighted standards are: Universal Description, Discovery, and Integration (UDDI), which from its inception has been considered a place to hold description information (including contracts) on services, and the Ontology Web Language for Services (OWL-S), a newer standard which focuses on semantic markup for WS to enhance discovery capabilities.

The last dimension of security is related specifically to *security properties* of services and the requirements pertaining to their usage policy, security policy and availability. The first two requirements are supported by WS-Policy and WS-SecurityPolicy respectively, and focus on how to express capabilities, preferences and needs of a service in a standardized way. The availability requirement is simply concerned with ensuring a level of reliability in message transmission and this is addressed by standards WS-ReliableMessaging and WS-Reliability as shown in Figure 2.4.

This concludes this overview of WS security standards landscape. For detailed information on the standards mentioned above, readers are directed to: [153] for WS-Security, WS-Trust, WS-SecurityPolicy, WS-ReliableMessaging, WS-Reliability, XACML, SAML and UDDI, [225] for XKMS, XML-Signature, XML Encryption, WS-Policy and OWL-S, [9] for WS-Federation, [110] for Liberty Alliance ID-FF, [34] for XrML, and in general see [157, 189, 36, 155, 14, 68, 214].

From this overview of standards, one can appreciate that there is a large amount of resources dedicated to the security of WS. Regardless of this progress however, and contrary to the perspective that there are already too many standards (see Alonso et al. [4]), there still remains the view that existing proposals are not an adequate solution to the technology level security problem. In Zhang [231] for example, the argument is made for a new layer called WS-Trustworthy. Zhang hypothesizes that WS-Security and related technologies at their core only address the security issue of Web services-centred computing. Thus, overlooking the issue of the overall service trustworthiness; trustworthiness in this regard deals with the level of confidence that services will act as intended, and encompasses attributes such as reliability, availability, interoperability and fault tolerance [231].

Additionally, Sidharth and Liu [187] highlight the need for their new framework based on the notion that the applicability of protocols such as WS-Security, WS-Trust and WS-Federation are in fact limited, as they only protect communications between trusted parties who share an established security context. The following statement aptly sums up their contentions, “The pervasiveness of web

services and SOAP API [Application Programming Interface] that can be invoked by anonymous consumers introduces security vulnerabilities [that] are not addressed by the existing standards" [187] (p.23).

Aside from WS standards and the systems which implement them, applications and software play a crucial role in the technical security solution. There are, for example, no standards available that can detect or protect against XML-based attacks such as XML-DoS. This means that the requirements of availability, malicious invocations and repeated invocations are not met. According to some authors, this is where XML-aware security appliances become useful. Work by Steel et al. [194] and Bebawy et al. [10] support this use and highlight the need for technologies such as the XML firewall. This is a tool that is capable of inspecting the XML content native to WS interactions (for example, SOAP payloads or attachments) for potentially harmful data or other threats [10].

Before moving on, the final significant concern regarding standards in both WS security and WS in general, is that as new standards are developed, they too may introduce new vulnerabilities to the WS technology architecture. New vulnerabilities then translate into new threats, risks and challenges for companies pertaining to their security. Yau and Rao [228] mention this reality as they discuss future trends for WS security in e-business. To remedy this problem, they stress that the security aspects of all new WS technologies should be carefully examined. This thesis observes however that even after that is done, it is unlikely that all vulnerabilities will be identified before technologies are implemented and deployed in businesses. Standards, technologies and software therefore, even though important components of the WS security solution, can at times exacerbate the security problem. This is a reality that businesses face which emphasizes the need to view security, even with implemented technologies, as an ongoing goal.

Reflecting on the layered security model in Figure 2.3 to put the above work into context, all of the literature reviewed thus far in this section fits within the generic 'technology solutions' layer. This ranges from WS standards such as WS-Security, SAML and XACML to the XML-aware security appliances. Next

therefore, a look is taken at the higher-level 'policies and procedures' layer to examine how security for WS is addressed there. For this research's purposes, this layer is simply viewed as processes for security. It therefore spans non-technical aspects such as policies, procedures, methodologies and best practices. Figure 2.5 is a slight modification of the conceptual model in Figure 2.3 which reflects this perspective.

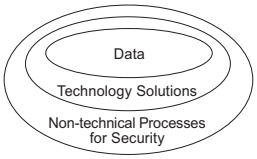


Figure 2.5: Business security environment (based on [107])

Beyond studying and addressing the perceived inadequacies of the current standards base, academics and practitioners are now looking towards the higher layers of security and approaches for achieving it (that is, the 'Non-technical processes for security' layer in Figure 2.5). These actions bring to life a prediction made by NIST in [189], which expressed that as WS technology matured, methodologies and recommended practices for security would become the next step in the goal of developing secure systems.

Some of the most noteworthy proposals in the higher layer have been in the following articles. Gutiérrez et al. [74, 75] aim to provide a methodical development approach for constructing security architectures for WS-based systems. Wang et al. [217] develop a method that uses fuzzy logic to measure the risk associated with WS, with full appreciation of the fact that due to WS' volatility, information on threats is usually incomplete or imprecise. Charfi and Mezini [26] build on existing standards and the theory of Aspect-Oriented Programming to provide a framework for securing WS compositions (necessary in B2B) using WS-Security and WS-Policy. Steel et al. [194] provide a concise, but essential outline of technology-agnostic WS security best practices. Lastly, there is the Event-driven Framework for Service Oriented Computing in van den Heuvel et al. [211]. This is a standard-agnostic, multilayered framework that aims to address the

problem of defining and enforcing access control rules for securing services use at the level of business processes. In their work, authors particularly focus on dynamic authorization, independent of specific standards [211].

These proposals all address areas crucial to services, but additionally act to stress the truism that security goes beyond just technical standards. This point is developed more in the next section which discusses the outstanding security issues plaguing WS and B2B.

2.3.3 Outstanding Security Issues

Security approaches geared towards WS within e-business should aim to be thorough in their considerations when planning, developing, implementing and maintaining an adequate solution. Figure 2.3 gave a general view of the layers that are thought to lead to well-rounded security. Standard security components encompass technologies, but as recent literature [183] in the study of security in general has emphasized, it also includes policies, processes, procedures, methodologies and best practices. Lichtenstein [111] supports this view and underlines the importance of these non-technical aspects in achieving holistic information security. Holism here is used to refer to the all-encompassing picture rather than a focus on the individual components alone [111]. Security is a multifaceted, multilayered phenomenon irrespective of where it is applied and therefore, neglecting any component, or focusing too much on another component, may lead to an inadequate security solution.

To the detriment of WS however, the importance of holism does not appear to be unanimously appreciated as any attention on the other aspects is being drowned out by the proliferation of various new technology standards. It may be very tempting therefore to regard such mechanisms as the 'solutions' to the WS security problem. Undoubtedly this could also be linked to the fact that WS is a technology itself, thus, providing technology solutions for technology-based problems. Whilst the work of technologists is valuable in building security and trust, alone they cannot form the entire solution. This is particularly when

considering WS' use in a business context. In fact, all these mechanisms address is the technology layer of security and the threats which emanate at that level. Thus, only providing a stepping-stone towards the goal of reliable, comprehensive, multilayered security for e-businesses.

Singhal et al. [189] strongly support this view as they stress that standards alone do not provide all that is necessary to develop robust, secure systems. Processes and best practices such as effective risk management, secure software development and defence-in-depth through security engineering, are also pivotal. Some of these higher-level approaches and methods (such as [74, 217, 75, 194, 26, 77]) were mentioned in the section above. This research, in addition to [74, 189], attributes the current focus on security standards to the newness of WS. This is an understandable reality as standards and technology arguably form an initial and basic part of a security solution. Moreover, technology solutions at this time still have unresolved challenges [189] and therefore more work is required before an established technical base is in place.

Companies who use WS however, must appreciate and remember that security is not a technology, standard or product. The other components of security previously outlined, along with methodologies and best practices are crucial considerations [189]. This fact is especially pertinent noting the high degrees of interconnection between businesses that WS readily and easily facilitate. To place the research in security for e-businesses using WS into context, the model in Figure 2.5 can be used. To date therefore, a large amount of research and emphasis has been placed on the 'technology solutions' layer, with only recently work emerging in the 'Non-technical processes for security' layer (some of which were mentioned above).

An intriguing research area which has received little emphasis is at the level of *cross-enterprise interaction* (that is, interactions spanning, and including collaborating businesses and their internal systems). Specifically this refers to providing some comprehensive approach to aid businesses in collectively handling security as the broad, inter-organizational concern it has become. This approach

would not solely be at the technology level but would also encompass a number of other fundamental aspects (including security directives, policies, business risk considerations, negotiations necessary and expectations towards reliable security) that businesses should jointly consider when developing and engaging in B2B interactions employing WS. This is particularly with the knowledge that in WS, lack of security in one business can very easily mean elevated security risk for a partnering entity, its systems and its data [77].

To briefly analyse the aforementioned research in [217, 194, 75, 26] towards fulfilling the needs of the 'Non-technical processes for security' layer (Figure 2.5), these can all be seen to successfully complement available technologies and provide useful security approaches. Assessed critically however, they do have shortcomings in terms of business security at the broad level of cross-enterprise interactions. Their main caveat is that they consider security predominantly from one company's internal viewpoint, that is, what a company should do internally to secure itself. This highly isolated perspective is inadequate due to the very nature of WS and the high degrees of interconnection between businesses—spanning exposure of legacy systems to purpose-built Web applications—that WS enables.

In van den Heuvel's article [211], even though that work allows for a layered and more comprehensive model for WS security during business process execution, its predominant focus is towards access control and particularly for highly dynamic environments. Both these aspects act to make it too specific a framework. The work in [77] possesses similar shortcomings, but mainly because as it progresses it focuses almost solely on technical implementations.

The basic notion behind security at the cross-enterprise interaction level can been seen in the small and largely exploratory research study carried out in [137, 91]. This work acknowledges the comprehensive security dilemma e-businesses face and proposes a generic model to enhance security. Figure 2.6 presents that conceptual model.

This model aims to highlight how businesses secure themselves internally (using policies, procedures and technologies) and how a similar approach (us-

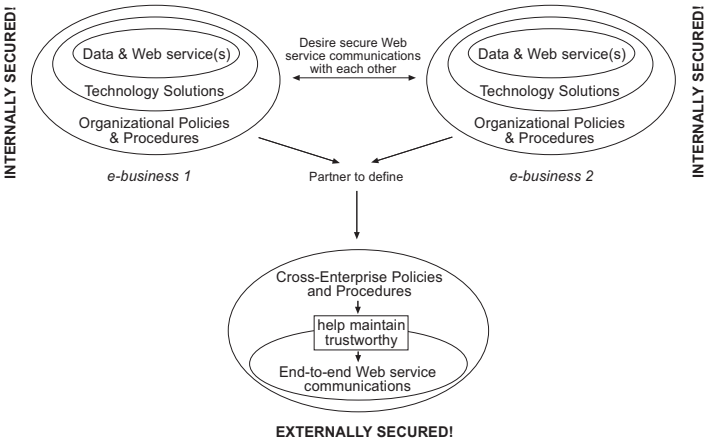


Figure 2.6: Model for enhancing Web services security ([137])

ing cross-enterprise policies) might be employed through businesses partnering (working closely together), to secure the entire WS communications [137]. These 'entire WS communications' refer to the same aspects as the aforementioned cross-enterprise interactions, therefore looking at both *internal* and *external* security. In many respects, this thesis's general proposals form an extension of that exploratory work, to delve into the intricacies of what would constitute such a comprehensive security approach.

An additional aim of this thesis is a more sturdy validation of the proposed approach, a glaring weakness of that initial study. Comparing Figure 2.6 to previous models such as the model within Figure 2.5, Figure 2.5 can be thought of as the security components necessary for an e-business' internal security (for example, e-business 1 in Figure 2.6). The similarities between the models are clearly apparent. Comparing these two models in general also helps to explicitly clarify the broad, inter-organizational level which motivates the research within this thesis. A main contention of this thesis is that when WS is used for e-business, internal security alone is insufficient because WS exposes companies like never before. A more comprehensive approach is necessary to reach good levels of security in these types of interactions.

Further to the previously mentioned goals, this new cross-enterprise approach would also aim to facilitate increasing the *trust* between business part-

ners, their systems and the overall service interactions, as an intrinsic objective. Trust in the context of this research was previously defined in Section 2.3.2. The importance of trust in e-business (with or without WS) is stressed by several authors [169, 101, 73, 208] and at the risk of oversimplifying its elusive nature, some of its most salient attributes in this context are transparency, accountability, predictability, reliability and benevolence [40, 212, 181]. This approach would aim to foster trust between partners, their systems (which will no longer be 'black boxes' to business partners) and the overall service interactions, by stressing these and related factors.

Generally, the approach could be seen to facilitate a level of security and confidence in services and partners not obtainable if businesses integrate security merely at the technology level. Technology-level integration, even though essential, is only part of the security solution. In discussing the general topic of WS' usage for B2B, Alonso et al. [4] note that WS enables "a company to open its IT infrastructure to external partners" however it does "not help with the many legal and contractual issues involved in B2B interactions" (p.299). Similarly, technology-level security integration can be done, but to allow for a more holistic security solution in B2B—and particularly in businesses which have cross-enterprise security as a critical goal—other higher-level aspects must be considered. These aspects go beyond the flashiness of dynamic security and trust negotiation possible with WS standards, and deal with a business-level security approach to risks and each organization's needs and goals as they pertain to security.

The close-knit, comprehensive concept suggested here is somewhat analogous to that recently called for in work by Dynes et al. [49] in the domain of the extended enterprise. The extended enterprise provides an excellent example of the close relations now prevalent in B2B, but also the substantially enlarged scale of the security problem in such relations. Commenting on this topic, Dynes et al. [49] compellingly argue this point and stress that many important risk management and security challenges are exhibited in extended enterprise-type asso-

ciations. These are challenges which unfortunately are not being addressed [49], hence the research agenda outlined in their article.

The similarities between the envisioned solution for WS security in e-business and the call for security approach research in the extended enterprise are rooted one main factor. That is, as businesses view cross-enterprise security in WS interactions as critical, this will justify them working closely together in a joint approach analogous to how research [40, 49] suggests it could be done in the extended enterprise. Some of the reasons for security being paramount to businesses could be: (i) large and long-term investments, (ii) strict government regulations, and (iii) importance of accurate, trustworthy business process execution and/or the nature of processes. Regarding point (iii) for example, this could be processes that are mission-critical, have strict privacy requirements (such as the health sector), or are susceptible to attack (such as the financial and banking sectors) and so on.

Before concluding this literature survey, a brief review of security-focused approaches, frameworks and models which are not specifically targeted towards WS, is undertaken. This review assesses some widely used methods and draws its relevance from the fact that at the core, the approach envisioned adds to a number of existing ones to provide another option for companies' security. There are four approaches of note from the literature. These are TOGAF [202], SABSA [186], SDL [124] and TSP-Secure [192, 191].

TOGAF [202] is a detailed method accompanied by a set of supporting tools for developing an enterprise information systems architecture. Although not originally directed at security, there have been initiatives put forward which incorporate security (including risk analysis, security policies, technologies and so on) in TOGAF's core architecture development method. SABSA [186] on the other hand is a framework for supplying cohesive information security solutions to businesses. Its six-layered model ensures that security is an integral part of a company's IT lifecycle and management infrastructure. The SDL [124] method is a software security assurance development process. This admittedly has its

concentration on a lower and more technical layer than TOGAF and SABSA. SDL specially aims at creating a trustworthy, secure computing base for applications. TSP-Secure [192, 191] is the final approach and it focuses on a strong team-based outlook to secure software development. As with other methods above, typical tasks include identifying risks, security requirements engineering, defining security designs, and security testing of the code and systems.

Broadly assessing the methods discussed above, these are all very adequate de facto security methods. There are two key factors which differentiate them and the envisioned approach however. First, they aim at a much more generic systems development level—this links to the previously identified fact that they are not aligned with WS, its challenges, tools or technologies. This has its benefits nonetheless as their approaches are not tied to particular technologies. Second, they do not pay special attention to cross-enterprise development and management of those collaborative interactions. For example, key aspects of interest such as creating an atmosphere of trust and security across parties is not a consideration. Other than these two factors, there are likely to be various similarities between these security models and the envisioned approach.

2.4 Summary

This chapter has taken an in-depth look at the field of e-business, WS, and their security, with the aim of identifying gaps in current research and therefore paving the way for this work. From this analysis, it was apparent that there are various unaddressed issues as e-businesses look towards creating and maintaining a comprehensive, trustworthy WS security solution. Primarily these stem from two aspects. First is an overly reliant emphasis on technology, alluding to standards and systems as the complete solution to WS security in e-business. The second aspect notes an overly isolated or individualistic security stance. This focuses on the process *one* company should follow to secure itself internally, therefore ignoring the comprehensive security issue across collaborating entities introduced

by WS use.

To tackle these issues an approach to security was suggested in the previous section which would focus on a number of key aspects. These centre around: (i) the consideration of the full nature of WS and its security implications particularly when used for e-business, (ii) recognizing and appreciating the 'live' inter-organizational security issue now faced by interacting e-businesses, and finally (iii) promoting the use of a collaborative or extended enterprise-type approach to provide enhanced levels of security and trust across interacting parties.

Chapter 3

The BOF4WSS Approach

Security must be omnipresent throughout your infrastructure in order for you to begin to feel your application or service is secure. In order to accomplish this, it is imperative that you follow a structured methodology. — Christopher Steel, Ramesh Nagappan and Ray Lai

3.1 Introduction

Having critically examined the literature on e-business, Web services (WS) and their security in Chapter 2, this chapter presents a possible solution approach to address some of the outstanding issues. This solution follows naturally from the type of approach described in Sections 2.3.3 and 2.4. Specifically therefore, it targets issues which stem from an overly reliant emphasis on technology for security and security approaches that are too isolated and individualistic.

The core of this chapter begins with an overview of the proposed approach, namely the Business-Oriented Framework for enhancing Web Services Security for e-business (BOF4WSS). This includes highlighting its novelty, and indicating and justifying design aims and goals. The chapter then moves on to a very detailed discussion of the framework, its objectives, steps and processes. Finally, BOF4WSS' scope is considered to identify the intended target group of businesses. During these discussions, the terms 'framework' and 'development methodology' are considered to be very similar and therefore are used interchangeably. Both are

taken to portray a collection of related stages towards achieving a goal, including inputs, processes and their outputs.

3.2 Overview

BOF4WSS, displayed in Figure 3.1, was conceived to address the outstanding security issues identified in Chapter 2 and strengthen available security/trust solutions. The framework consists of nine stages which, in general, semantically resemble those found in typical systems development methodologies. Formally these stages are Requirements Elicitation, Negotiations, Agreements, Analysis/Architectural, Agreements, Systems Design, Agreements (for Quality-of-Services), Development and Testing, and Maintenance. Compared to typical methodologies, the Negotiations and Agreements stages are novel. Their inclusion was found to be crucial in BOF4WSS noting the cross-enterprise nature of the development and imperative need to discuss, negotiate and agree on clear paths forward.

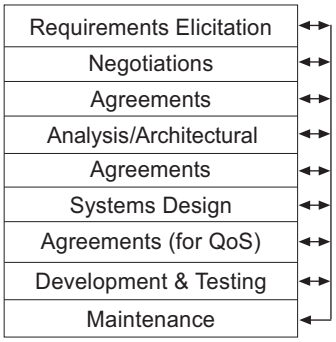


Figure 3.1: BOF4WSS Overview

The Waterfall Model (WM) methodology [103] in particular was the main influence for the framework's design. This can be seen when comparing the framework's phases to those of the WM, such as system feasibility study, requirement analysis and project planning, system design, detailed design, coding, testing and integration, installation and maintenance [103]. Depending on the article sourced, the stages of the WM may be named differently (for example, [177, 16, 193, 213]); [103] is referenced because of its detailed view of typical WM tasks/stages.

The WM was preferred to other methodologies such as prototyping [133], spiral [16] and object-oriented [81] approaches due to the transparent, well-organized, highly documented, and strongly disciplined process it can bring to a large inter-organizational development project [103, 18]. Some practitioners even view the structure possible with the WM as an ideal fit for the corporate (somewhat bureaucratic) world and a key reason why the WM is here to stay [27].

With appreciation of the flexibility and quick turnaround benefits of agile and more lightweight methods, these were also considered at length. These techniques were not chosen for the framework's foundation however, because literature [193, 213] does not advise them in situations: (i) of large development projects, (ii) where development teams might be in different places and dealing with complicated interactions with other hardware and software, or (iii) in critical systems development. These are all likely situations where BOF4WSS might be used, as mentioned in previous (Section 2.3.3) and also, later sections.

Despite the benefits listed, it is accepted that the WM does have shortcomings and criticisms. For example, researchers have identified that it lacks flexibility in the original model when traversing stages and freezes requirements too early [103, 16]. To compensate for these concerns, BOF4WSS allows for flexibility through bottom-up progression and feedback (shown on the right in Figure 3.1). Additionally, even though requirements are determined early in the framework, these are only high-level requirements (as opposed to the traditional WM that defines all requirements) which may change (subject to agreement) at subsequent stages closer to design.

Underlying both of the points above is the emphasis BOF4WSS places on the involvement of key stakeholders throughout the entire process to ensure gathering and circulation of necessary information, making of changes and so on.

The prime novelty in BOF4WSS is found in its emphasis on providing a collaborative development methodology which focuses on WS security. This methodology would accommodate multiple autonomous businesses working together. To address the outstanding issues from Section 2.3.3, BOF4WSS aims at:

considering the full nature of WS and its security implications within e-business; appreciating the real-time inter-organizational security issue now faced by interacting e-businesses; and finally promoting the use of a collaborative approach to provide enhanced levels of security and trust. These are its high-level design goals.

As will be seen below, the framework and its phases give detailed guidance on what should occur and how, and its relevance in attaining desired levels of holistic security for these *cross-enterprise interactions*. To recap, cross-enterprise interaction security refers to ensuring businesses are secured *internally*, but also that the *external* interactions encompassing collaborating businesses are secure to some level. External interactions with a company simply mean interactions that occur in transit (that is, while they are being passed between companies), and to some extent what occurs regarding the security of these interactions while being processed by business partners. This internal and external focus is revisited at various points in BOF4WSS's presentation.

Returning to the point regarding detailed guidance given by the framework, this will involve defining the expected inputs to stages along with their required outputs/outcomes, but especially the recommended low-level goals, activities and steps within those stages that can help achieve the outcomes. Where suitable, this guidance aims to reference and reuse existing methods and practices—both from industry and academia—thus concentrating on the compilation of these into a coherent, well-defined process.

Another main design goal of the framework is to utilize Web services specifications and tools wherever and whenever useful. This includes validated proposals from the research community as well. These choices were made to provide companies that adopt BOF4WSS with a practical methodology that pulls together key WS specifications and tools from the plethora of technologies available. Furthermore this shows exactly where and how they can fit into the development of a WS solution. To date, the author is not aware of such a broad methodology as BOF4WSS, which aims to fit together a majority of the critical pieces of the WS

security puzzle in the context of cross-enterprise, highly structured, extensible (allowing approaches/tools to be plugged in), business-oriented framework.

To support the largely textual description of the framework's activities next, a number of diagrams are included illustrating each stage and its respective workflow. Since security issues are a central concern to BOF4WSS, the discussion concentrates primarily on these aspects rather than an isolated discourse on functional and quality related aspects. Quality aspects or requirements in this regard refer to non-functional requirements excluding security, such as performance, scalability and so on. At some stages however, in the interest of completeness, this chapter does give some guidance on these areas. This is particularly when they relate to useful WS standards and technologies.

Lastly, BOF4WSS assumes that businesses have previously agreed (through feasibility studies, initial dialogue and so on) to use WS to support a generally defined business scenario. In other words, the broad scenario is known. BOF4WSS's task therefore is to provide a methodology for its planning, development and implementation. Below, the framework's stages are presented.

3.3 Requirements Elicitation Phase

The **Requirements Elicitation phase** (shown in Figure 3.2) is the first stage and within it each company works largely by itself. As with typical system development approaches, the phase involves analysing internal business objectives, constraints, security policies, relevant laws and regulations and so on, to determine their high-level needs for the expected WS business scenario.

The first implicit but crucial step therefore is to organize teams within businesses to work on the project. They need not be dedicated exclusively to this project but should be committed and have a clear idea of the goals of the envisioned scenario. Ideally, these teams will consist of some top executives, domain experts (the domain being related to the business processes involved), project

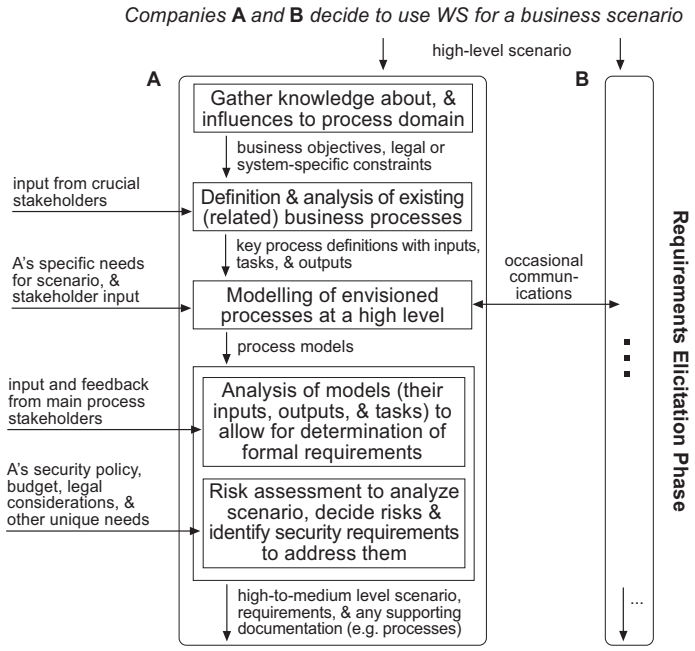


Figure 3.2: Workflow model of the Requirements Elicitation phase

managers, business and systems analysts, system designers and developers, end users, IT security specialists and legal counsel (preferably with experience of technology law, data protection acts and such). These teams would report to the top executives, who will have participated in the embryonic stages of taking on the project/scenario. Additionally, executives will provide the financial and higher-level backing for the project throughout the company. During the application of the framework and inclusive methods such as Demirörs et al. [45] (which is presented next), the involvement of all the types of stakeholders mentioned above is heavily stressed. As validated by studies such as that of Hartman and Ashrafi [78], this is a critical success factor in managing and developing information systems.

Once teams have been established, this framework's phase utilizes the methods proposed by Demirörs et al. [45], which focus on the definition and analysis of business process models to elicit requirements (functional, quality and security-specific). This approach is preferred due to its innate emphasis on business processes—the culmination of WS interactions.

As illustrated in Figure 3.2, the approach in [45] consists of firstly gathering relevant knowledge about the process domain and what influences it. This

information could include business objectives, legal or system-specific constraints, existing process models, system architectures and so on. The second task is the analysis and modelling of current processes (particularly if existing models are not accurate) to enable for a full appreciation of critical process flows, and their inputs and outputs. This will primarily focus on internal and external processes directly involved in envisioned WS interactions and those that are candidates for redesign. Crucial stakeholders for this information will be members of the teams already formed, and other relevant top executives, domain experts, analysts and end users. For the modelling activity in this task, the Unified Modeling Language (UML) [149] is suggested for use as it is a standard technique likely to be known by both enterprises.

If companies are entering a process they have not done before (therefore, there are no 'current processes' specifically related to envisioned interactions), the task above will not be as relevant. The aim instead is considering how their other internal processes will integrate with newly envisioned interactions.

The third task is the modelling of new processes. At this point, the needs of new business interactions (driven by the companies and at the core, the stakeholders) result in new processes, but often also include enhanced and updated existing processes. Legacy systems deserve special emphasis because if they are to be included in new processes, they can be either re-engineered (reimplemented), repurposed (changing interface and encapsulating some business logic), or partitioned and packaged into deployable functional components [158]. The choice between these methods will largely be dependent on benefit versus cost and whether legacy systems can adequately fulfil new business goals.

Generally, these new processes are expected to be high-level, and mainly cover internal (known) as opposed to external (envisioned) operations. This however may not always be the case. For example, if the external processes with the other company are known due to prior transactions, businesses may be able to develop initial medium-level process flows which encompass the external interactions. In either case, occasional communications with business partners are

required to enable useful processes to be defined. Again UML is suggested for (i) the reason previously given and particularly because these high-level models can be used to aid in discussions in the Negotiations Phase, and (ii) the fact that it adequately enables high- or medium-level processes to be defined. True to the envisioned flexibility of BOF4WSS however, companies are free to use their modelling language of preference.

The last task in Demirörs et al.'s approach [45] is the actual requirements determination. This is accomplished through analysis of the newly defined process models, and business analyst and domain experts can direct this task. By assessing the inputs, outputs and tasks involved, general requirements (functional and quality-based) for each stage of the process can be defined from a high level.

For quality requirements in particular it is understood that these may be hard to state this early and at this rather high level. Businesses however should make an effort to give some idea of their desires for system quality (such as in terms of extensibility, scalability, performance). To elicit security-specific requirements, authors mainly suggest the analysis of the access restrictions of the actors (users and applications) on processes and process inputs and outputs. This is where the information security specialists on the team will be involved. In these last two stages, BOF4WSS heavily involves the previously highlighted stakeholders.

In addition to the security requirements identified, the framework strongly suggests a scenario risk assessment to provide more extensive security documentation. This assessment, as opposed to the one above which focuses primarily on access restrictions in processes, enables a comprehensive, security-driven scenario analysis. The assessment is strongly suggested primarily to combat the unfortunate reality that if left alone, a significant number of businesses would not carry out formal security risk assessments to identify key risks faced [210].

To aid in the assessment process, there are a range of methods that the security specialists of companies might use. BOF4WSS suggests well-documented and internationally validated techniques such as NIST SP 800-30: Risk Manage-

ment Guide [195], OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [24, 3], CRAMM [188, 204], CORAS [47], ISRAM (Information Security Risk Analysis Method) [99] and the IS risk analysis based on a business model [196].

Generally, some of the crucial factors considered in a chosen technique should include assets, threats, vulnerabilities, risks and their priority levels (that is, severity and impact if risks materialize), organizational security policies (policies directly convey a company's security posture), pertinent laws and regulations (those governing internal operations and those with respect to working with external parties), security budgets (balancing cost and security is paramount), and security goals expected to be met by new business partners.

All of the factors listed above significantly aid in the determination of the security actions and security requirements that should be factored in during these envisioned WS communications. Throughout this report, a *security action* is defined as any way (that is, setting up protective measures or not) in which a company treats or handles a risk it faces, whereas a *security requirement* is a high-to-medium level desire, expressed to protect against risk. Security actions therefore encompass security requirements. An example of an action is, 'the risk of ensuring the security of a server to be outsourced'. A requirement however could be, 'the integrity of personal data must be maintained'.

Requirements to be carried forward should particularly address areas that (i) need additional security internally (and relate to the overall scenario), and (ii) relate to the interactions with the business partner. After these requirements have been gathered, they are added to the previously identified requirements and documented to provide the stage's output: *a high-to-medium level scenario process (inclusive of the models defined), high-level requirements (functional, quality and security), and any other the supporting information.*

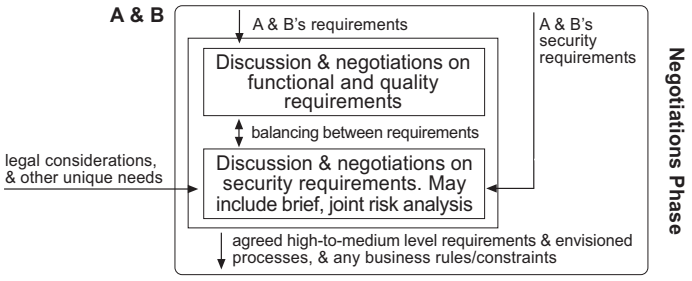


Figure 3.3: Workflow model of the Negotiations phase

3.4 Negotiations Phase

In the **Negotiations phase** next, teams consisting of project managers, business and systems analysts, domain experts and IT security professionals from the companies meet, bringing together their requirements from the previous phase for discussions. Figure 3.3 displays the workflow. The purpose is to use the stage inputs as a basis to chart an **agreed** path forward in terms of scenario requirements and high-to-medium level process definitions. As compared to typical development methods such as the WM, BOF4WSS explicitly includes negotiations as a phase to stress its importance in the inter-organizational scenario. This is especially noting the varying expectations each company is likely to have towards security. Expectations (and requirements) could vary with regards to whether a process (or set of service interactions) needs to be secured, to what level it is to be secured, how security will be applied and so on.

The two main tasks in this phase therefore are, Discussion and negotiation on (i) functional and quality requirements, and then (ii) security requirements and actions that arise. Depending on the preferences of businesses using the framework, the latter of these tasks may include a joint risk analysis aimed at identifying any risks (and subsequently, requirements) not conceived previously. Deliberations on statutory and regulatory requirements are especially important when discussing security, as businesses may not be in the same industry, or even country. Where necessary, as is seen in the workflow, backward progression from the security requirements definitions to functional/quality requirement definitions is allowed. This is mainly to support balancing between functional/quality and

security actions and requirements.

The Negotiations phase facilitates its purpose by accepting that each business constitutes a different security domain (and is likely to have different desires and obligations). It therefore explicitly stresses the need to negotiate on security actions, rather than adopting one company's needs, or assuming integration of desires at this level will be seamless. Work by Tiller [205] clearly highlights that in forming these extended networks or partnerships of companies, this integration task is formidable. It is however a necessary and pivotal precursor to engaging in interactions. After the identified tasks have been completed, the expected output of this stage will be *the agreed high-to-medium level requirements, high-to-medium level envisioned processes, and any business rules/logic and constraints important for future stages.*

3.5 Agreements Phase

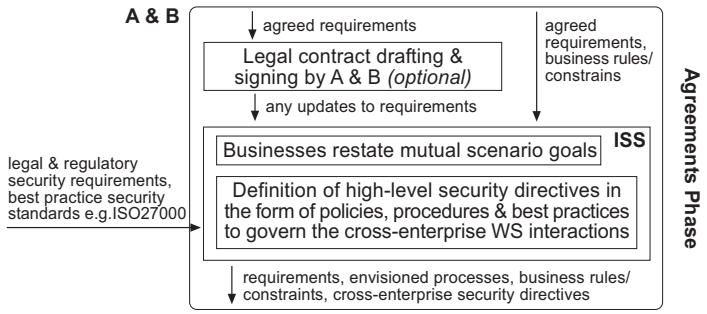


Figure 3.4: Workflow model of the Agreements phase

The **Agreements phase** depicted in Figure 3.4 builds on the concluded negotiations and initially advocates a legal contract to solidify the understanding of the requirements between companies thus far. A legal agreement at this point is not compulsory however, as it is appreciated that businesses may choose to include the contract at another stage. The reason the contract is suggested here is to create a safety net for both companies during these early stages of planning and negotiations. The contract would focus on two main aspects, binding the parties to negotiations for possibly future business interactions in good faith

(non-disclosure agreements may be used for example) and secondly, defining the groundwork for a more comprehensive contract to follow in later stages. The initial agreement and definition of needs in the Negotiations phase makes the latter of these tasks (that is, defining the groundwork) less arduous.

The contract is followed by the Interaction Security Strategy (ISS) which in itself is a novel contribution. As opposed to the legal document above, the ISS is a less rigid management structure that defines high-level, cross-enterprise security directives to guide the interactions and relevant security decisions internal to companies. These directives are typically in the form of security strategies, policies, procedures, best practices and objectives. The novelty in the ISS is its provision of a more pragmatic security governance structure for companies. This appreciates a variety of important factors and is not stated in rigid, hard to understand and follow, contractual terms. Formally, the ISS can be seen to build on and considerably extend the idea of cross-enterprise policies introduced in [137, 91].

Figure 3.4 shows that the central activities in the creation of the ISS are: (i) restating businesses' mutual goals for the scenario—this will provide a clear vision for the strategy, and (ii) actual definition of the security strategy's directives. In addition to the use of requirements and business constraints, when defining these directives the framework suggests consideration of two aspects. These are, the legal and regulatory mandates which may influence companies and interactions, and secondly the best practice security standards available from industry. These are discussed below.

In business today, legal and regulatory requirements pertaining to security are becoming increasingly important, especially within the arena of online business. These mandatory requirements cover topics such as data protection, data privacy, computer misuse, incident disclosure and notification, third-party auditing and even security within business relationships. The aim of the ISS with regards to these requirements is mainly to stress that businesses make themselves aware of the content of these laws and regulations. This is not only to fulfil the

statutory need, but also because a number of these laws stress principles of good, reliable security that should be practiced by companies.

Some of the most relevant laws to be considered include the following. The Sarbanes-Oxley Act (SOX) of 2002 (U.S.) emphasizes the maintenance of adequate internal controls to ensure the accuracy of financial information [194, 128]. The Health Insurance Privacy and Portability Act (HIPAA) of 1996 (U.S.) focuses on confidentiality, integrity and availability of personal data (medical or personal records) ensuring it is protected whilst in storage and during transmission, both within and external to the company [128]. The Data Protection Act (DPA) of 1998 (U.K.) is targeted towards personal data, ensuring that it is adequate, accurate and processed fairly and lawfully, amongst other things [216]. The Gramm-Leach-Bliley Act (GLB) of 1999 (U.S.) is mainly aimed at financial institutions, and stresses activities such as the evaluation of IT environments to understand their security risks, establishment of security policies to assess and control risks, and the scrutiny of business relationships to ensure partners have adequate security in place [128]. Knowledge of, and adherence to these regulations is critical as companies look to conduct business in an increasingly regulated marketplace.

In addition to promoting the compliance with legal and regulatory requirements, the ISS emphasizes the incorporation of best practice standards in the approaches by companies towards inter-organizational security. Whilst it may be tempting to assume that businesses already accommodate such standards, recent surveys [210] have shown that companies are largely not aware of key security guidelines.

The ISO/IEC 27000 series is a perfect example of important standards and as Figure 3.4 shows, they form a key input into this framework stage. This standards set, in particular, is targeted at the provision of an internationally recognized, organization independent framework for effective, extensive information security management [19]. Internal security management system for an organization is a fundamental objective of this standards set.

Looking directly at the ISS' emphasis on standards during this Agreements phase, there are many benefits. As mentioned above, the term 'cross-enterprise interactions' denotes interactions spanning and including collaborating businesses and their internal systems. Therefore, securing the *internals* of businesses which participate in these interactions is also a crucial goal—this is especially where the ISO/IEC 27000 standards set is useful. Two specific benefits of applying these standards are that they provide organizations with a systematic way of fulfilling legal and regulatory responsibilities (for example, some standards can help meet SOX requirements) and secondly, through accreditation schemes, businesses that can demonstrate adherence to guidelines can be issued with a certificate. This would show customers and business partners that their systems and practices are secure to an international standard [19].

At the *external* level security standards also prove useful as certain clauses (for example, ISO/IEC 27001, Control A.6.2) deal specially with external parties. These would typically focus on maintaining the security of an organization's information assets as they are accessed, processed, communicated with, or managed by external parties [19]. The two main tasks involved in this attempt are the identification and addressing of risks directly related to external parties; these are two activities that were completed to some extent during the risk assessment in the preceding Requirements Elicitation phase. Reflecting on Control A.6.2 therefore, as opposed to resulting in an exhaustive legal contract (as is suggested by the Control), any new risks and their respective controls which were identified would feed into the cross-enterprise security directives for the ISS.

Having discussed the ISS, its goals and its main influences, a look is taken at examples of what the ISS could cover. The first example is the definition of best practices each company should abide by internally. One best practice might be related to ensuring parties maintain sufficient logs of system events, as this information would be very useful in cases of a security breach. Another example of an aspect the ISS would address would be the definition of scenario incident response activities. This would consider what procedures companies

should jointly follow if a security incident is suspected or has occurred. The third, and somewhat general example, relates to the responsibilities and expectations of companies towards security. The ISS would enable companies to almost always have some clear vision of what their partners should be doing (likely stated in terms of policies and procedures) relating to aspects of security.

The final example is the creation of a small, cross-enterprise team specifically to handle security matters and update the ISS and other security measures as and when appropriate. Here, the ISS recognizes and appreciates that security is an ongoing concern. Therefore, it calls for a team to be formed constituted of persons from all enterprises to manage this concern. In essence, the ISS forces businesses engaging in joint interactions to consider and address security issues, both internally and externally, that previously may have been overlooked due to overly simplistic or isolated approaches towards security.

By jointly creating an ISS companies can have some degree of certainty that partners are committed to maintaining an acceptable security posture. This leads to another central goal of this strategy, that is, to foster trust amongst business partners. The ISS aims to foster trust through predictability and transparency in security approaches, by outlining a security strategy and subsequent framework that all businesses agreed to adopt and follow. Trust within e-business was outlined before (in Section 2.3.3) and its importance should not be neglected.

This research does appreciate other, more direct methods to assess a business partner's commitment to security, such as audits, on-site visits and questionnaires (suggested by Misrahi [128]), but leaves this choice to individual organizations that adopt BOF4WSS. Within very closely-knit and highly collaborative relationships (such as the e-supply chains) however, audits amongst other precautionary mechanisms are strongly recommended, and this opinion is supported by Baker et al. [6]. The closer businesses are, the more likely they are to be affected by each other's security risks. Businesses should be mindful of this factor as they seek to work with other enterprises. To complete the Agreements phase the following documents and information should be produced and carried forward to the

next stage. These are *the high-to-medium level requirements, high-to-medium level envisioned processes, any business rules and constraints, and cross-enterprise security directives in the form of strategies, policies, procedures, best practices and security objectives (or more formally the ISS).*

3.6 Analysis/Architectural Phase

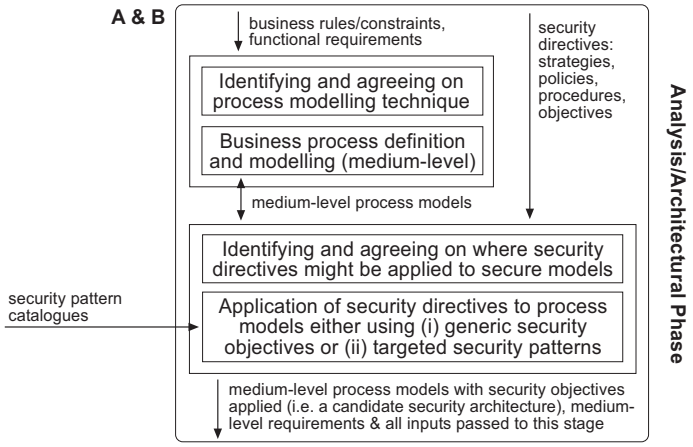


Figure 3.5: Workflow model of the Analysis/Architectural phase

Following on from agreements, next is the **Analysis/Architectural phase**. The workflow is given in Figure 3.5. This phase's purpose is to enable companies to take the agreed requirements and define conceptual (medium-level), secured business process models for the foreseen interactions. These models are expected to encompass not only the high-level company-to-company process flow, but each company's internal process flows that constitute part of the general business scenario. Internal process definition and sharing is encouraged to cultivate an atmosphere of openness between the companies, but especially to make companies properly analyse the expected internal flows and how they fit into the general scenario. At this point, it is still relatively easy for companies to make any necessary updates.

Since it is almost certain that businesses would have engaged in process modelling at some point before, the teams' business and systems analysts are likely to have preferred techniques. As a result of this, the first task in this phase

is agreeing on the technique that they will use. The framework does not stipulate a particular method for use but does advise businesses to carefully deliberate the benefits and shortcomings of the options available.

To define the medium-level business process models needed, various standard modelling techniques are available ([158, 69, 2]) for project teams to use, and analysts and domain experts are key personnel at this stage. Some of the most popular of these are UML (inclusive of its many specialized profiles—see [146]), Data Flow Diagrams (DFD), Integration Definition for Function Modeling (IDEF) techniques and the Business Process Modeling Notation (BPMN). The UML 2.0 extension for SOA, UML4SOA [117], is a recent proposal from research which also provides an interesting technique. This profile however appears to be targeted at service orchestration only (thus, internal as opposed to cross-enterprise systems). Yet another option is the UML profile in [190] for WS composition. This could be very useful because a main design goal is the inclusion of transformation rules that allow designed UML models to be transformed to Web service compositions that are executable (for example, BPEL, albeit an older version)—a necessary task in future stages.

Companies' discussions on modelling techniques should bear in mind: (i) the goal of this phase, that is, the definition of **secured** medium-level process models, (ii) the fact that these models will have to be further decomposed and used to express varying aspects at lower level, and therefore having standard ways to state these aspects may be beneficial, and (iii) the impending need to translate these models into more WS-specific formats, for external and internal usage.

Regarding the last two points, businesses for example might find it useful to know that there have been proposed extensions to UML to account for security. Furthermore, with highly esteemed options like UML and BPMN, there are mechanisms publicized that can translate these medium-level models to WS-specific languages as will be seen in subsequent sections. Figure 3.6 is a reference guide that BOF4WSS would provide to companies for a summary view of UML and BPMN with respect to modelling security. Information on a UML profile for

QoS and FT, Security requirement with a UML 2.0 profile, and Extension for the Modeling of Security Requirements, can be found in [147, 172, 173] respectively.

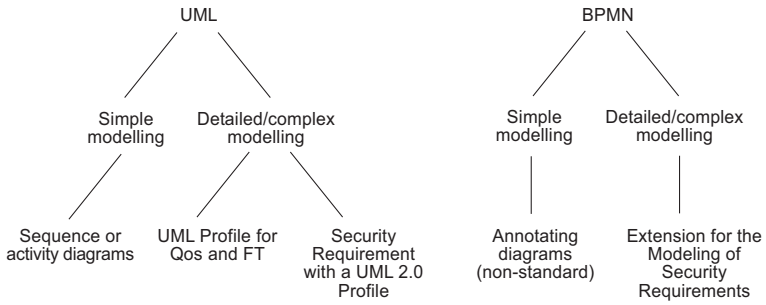


Figure 3.6: Options for modelling security with UML and BPMN

Research work by Aguilar-Savén [2] and Giaglis [69] has investigated the nuances of a number of popular process modelling techniques and their findings would be a first point of reference (suggested by BOF4WSS) to guide companies in choosing a method. The first article provides a taxonomy of modelling techniques to assist decision makers in evaluating and selecting a suitable option based on the project and/or the specific purpose for modelling [69]. Purposes could range from functional (task-focused) to informational (data flow-based), or from process development to simply enabling understanding and communication. The second article is a more recent review of the techniques for modelling and culminates in a detailed summary of these approaches (covering their attributes, characteristics, strengths and weaknesses), and a framework classifying them according to their purposes [2].

Once the modelling technique has been agreed, Figure 3.5 shows that businesses then proceed to use the phase's inputs to define and model the cross-enterprise processes. During this task, companies should be wary of the temptation to prematurely define the processes in great detail. Even though it is understood that this is the next step (the Design Phase) and that for some security objectives low-level analysis is ideal, agreeing on and defining a conceptual model is a critical base step to the following stages. This degree of modelling enables visualization and description of processes at an abstract but holistic level. It is also comprehensible by all members of the companies' teams, as opposed to

only systems designers or software developers. Conceptual process definition can allow companies to analyse processes, weigh alternatives and assess process interrelations. Most importantly however, it enables the achievement of agreement on the vision for the medium-level architecture and process flow, in and across enterprises prior to low-level design.

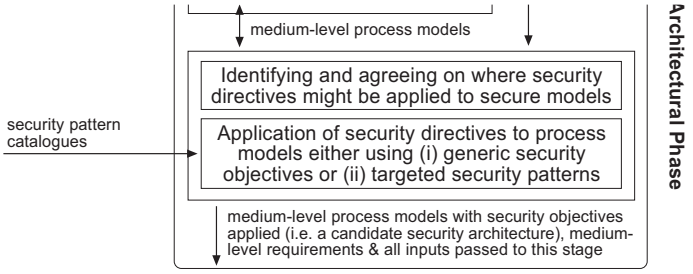


Figure 3.7: Identification and application of directives

After defining the cross-enterprise process models, the next general task presented again in Figure 3.7 is to conceptually apply the security directives to provide options for the security architecture. This is therefore not the final architecture nor does it concentrate on a low-level application of directives. Due to the range of directives and the variety of possibilities in which they could be applied to even these medium-level models, businesses are faced with a complex undertaking. Initially therefore, the framework suggests that companies (especially system analysts and security professionals) focus on identifying and agreeing on where security directives might and should be applied to secure the models. A detailed table is one simple way that companies could match relevant security directives to the processes they will affect.

When the matching has been completed, there are two well-accepted methods in which directives can be initially applied to the process models. These are, (i) through the use of generic *security objectives* (as done by Röhrig and Knorr [174]) or (ii) by employing targeted *security patterns* (see Steel et al. [194]). These two methods are especially suitable for BOF4WSS because they provide good security procedures which are generic enough to be applied, even if only by way of annotations, to a number of the aforementioned modelling techniques.

Figure 3.8 diagrammatically presents the general process.

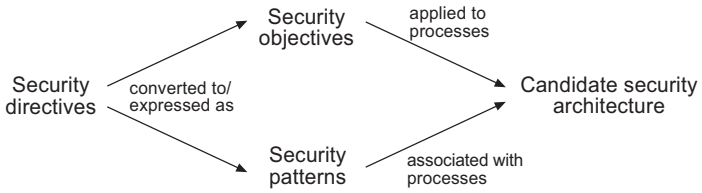


Figure 3.8: Process from security directives to security architecture

To use the first approach ([174]) in its original form, companies' security professionals will have to ensure that process-related security directives are stated with regard to the *security objectives* of confidentiality, integrity, availability and accountability. This list however is not definitive as the framework does appreciate the desire of businesses to add other relevant objectives that reflect the directives. These additions might include objectives on nonrepudiation and authorization for example.

After this is complete, individual process components (inputs, outputs, activities and actors—users of process activities) are assigned rating values (for example High, Medium, Low) in terms of these objectives. These values indicate level of security desired for the component and should be based on previous risk analysis findings and the security directives as opposed to being just randomly chosen. The following gives an example of assignment; if a data value α (representing a bank account number) is output from an activity and the risk analysis or security directives dictate that α is very sensitive data and its confidentiality is likely to be threatened, companies might assign process component α with a confidentiality rating of 'High'. This type of assignment activity is done for all process components in the previously defined models.

The second approach is the application of *security patterns* to secure the process models [194]. Formally, "a security pattern describes a particular recurring security problem that arises in specific contexts and presents a well-proven, generic scheme for its solution" [184] (p.5). Simply, it can be thought of as a well-proven, generic solution to a recurring security problem. An immediate benefit of employing this approach therefore is that it would utilize catalogues of proven

and tested security patterns to address the requirements in the security directives. This accounts for the input of the security pattern catalogues to this stage as shown in Figure 3.7. Steel et al. [194] have investigated this topic in detail and have provided an extensive listing of existing and new patterns spanning the Web, Business, WS, and Infrastructure and Quality of Services tiers of a typical company's systems.

Using the example of data value α from the *objectives* approach above, personnel would check through the security catalogues for an appropriate pattern to protect α. Having identified suitable alternatives, these would then be noted for formal analysis and application during the subsequent framework Design Phase. The goal at this Architectural stage therefore (as illustrated in Figure 3.8 and also done in [194]) is mainly the identification of relevant security patterns.

To briefly compare the security objectives ([174]) and security patterns ([194]) methods, the first approach is likely to be more time consuming, as applying priorities for the security objectives to each process component is a substantial task. Conversely, two benefits accompanying this method are, the simplicity of use and application and secondly, that it naturally enables the security priorities (of High, Medium, Low) to be associated with the specific components. The latter of these tasks is not inherently accommodated in the security pattern concept, albeit easy to add in some cases.

If parties' security professionals and analysts choose to use patterns, the advantages include having their security problems addressed in a structured way, and also the ability of non-security experts to reference and apply proven security solutions (through the use of pattern catalogues) to solve otherwise complex problems [184]. An additional benefit of using the pattern catalogue in 'Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management' ([194]) specifically, is that it is largely geared towards WS interactions and is thus equipped with standards and technologies that can be used to implement the pattern in later stages. Regardless of the method chosen, the Architectural stage's output should be *medium-level process models with*

security directives applied (formally, this constitutes the candidate security architecture), the medium-level requirements (functional and security-specific) accompanying these models, and the inputs passed into this phase.

3.7 Agreements Phase

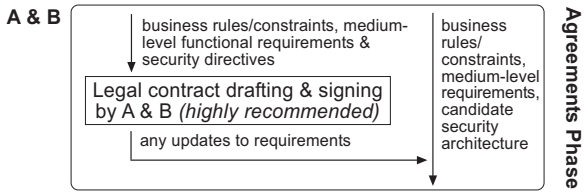


Figure 3.9: Workflow model of the Agreements phase

Following the formal conceptual process definition, the framework suggests the use of another **Agreements phase**. The respective workflow can be viewed in Figure 3.9. This agreement is in the form of a more thorough legal contract reflecting detailed expectations of parties included in the envisaged scenario. The business rules and constraints, functional requirements and security requirements all factor into this contract. The medium-level requirements are especially important as they provide further detail on the agreed interactions. As with the previous contract, the business and system analysts have the role of ensuring that business requirements and needs are transferred adequately into the legally binding agreement/contract. Security professionals act on the security requirements.

During contract drafting, requirements may change and therefore any updates made can be fed back into the known requirements and process models. Again, this legal document is used primarily as a safety net (in the event that companies have an irreconcilable disagreement and need formal arbitration) and therefore still relinquishes the role of governing day-to-day security interactions to the ISS. Many authors [40, 186] support this and similar views, and define a number of drawbacks to using contracts as the sole basis for conducting business. The outputs of this phase are *the medium-level process models with security directives applied (formally, this constitutes the candidate security architecture),*

the updated medium-level requirements (functional and security-specific) accompanying these models, the business rules and constraints and any other the inputs passed into this phase.

3.8 Systems Design Phase

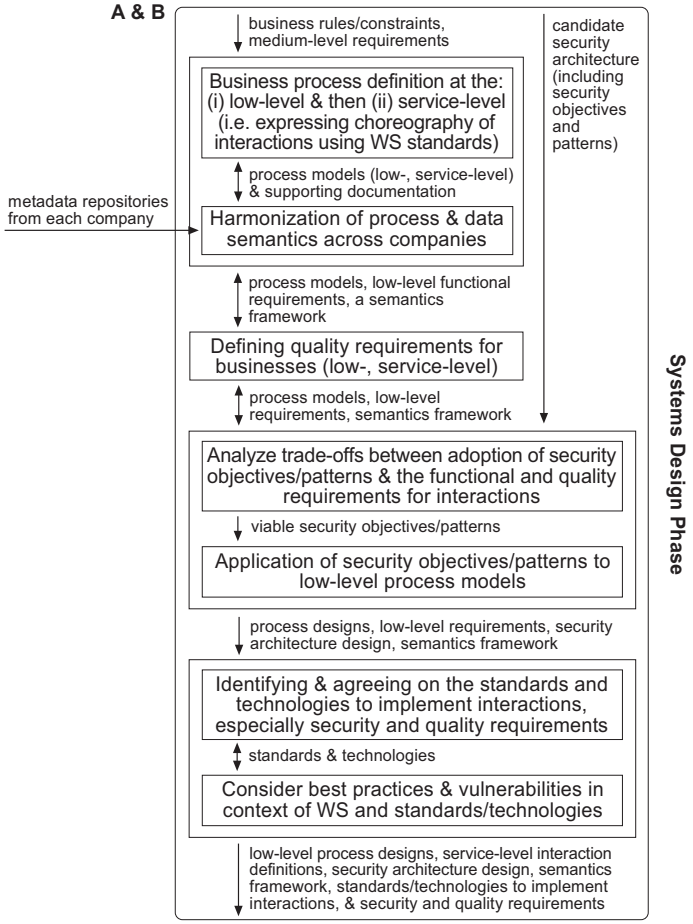


Figure 3.10: Workflow model of the Systems Design phase

The **Design phase** is analogous to a company's internal systems design process (such as that present in [194]) and therefore targets the definition of a low-level (or logical) systems-related view of exactly how the conceptual model from the Architectural phase will be put in place. Figure 3.10 defines the tasks.

The first activity is for the teams from each business to jointly define the low-level process models. The systems analysts within the teams should be involved at this point as they will have a more practical and low-level orientation

towards models. The framework advises businesses to reuse the modelling technique chosen before (in the Architectural phase) but on this iteration, to break down the medium-level models to the lowest level of detail. The goal is to decompose models such that the individual message flows between companies and the specific tasks which constitute each process activity can be seen.

In defining these low-level interactions, it is critical for company teams to identify the actual *services* and define the interactions in terms of these services. Erl [51] is one commendable reference for companies suggested by the framework, that examines moving from business processes to service models and designs, and also provides thorough guidance. Generally however, businesses should be attempting to identify aspects of functionality within processes that could form distinct logic units. To exemplify this task, Figure 3.11 is used.

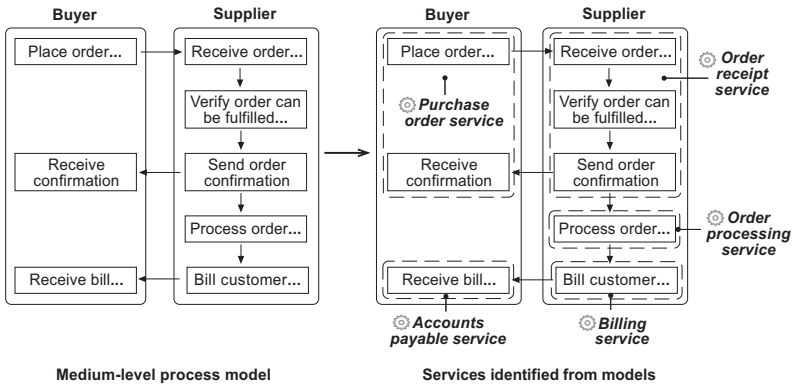


Figure 3.11: An example of moving from processes to services

This diagram shows a simplified medium-level process flow of a typical order processing scenario (on the left) and next to it (on the right) the services that were deduced from it. In identifying services, special attention was paid to subprocesses that could be somewhat independent and could be grouped and encapsulated with related tasks. The purchase order service is a good example of this as it encapsulates the 'place order' and 'receive confirmation' subprocesses into one unit of functionality that can be referenced.

Depending on how open companies have chosen to be with how their processes (or systems) will work internally, the low-level process definition task might

be primarily of the interactions *between* companies, or the interactions *between and also within* the businesses. To use Figure 3.11 to explain this point, the former of these tasks refers mainly to the arrows connecting the Buyer and Supplier, whereas the latter refers to those arrows plus the arrows and flows within companies. Even though the ultimate degree of openness maintained by parties throughout the framework's activities is largely left to the individual teams, BOF4WSS stresses that openness and transparency could foster trust. This trust is a key ingredient to successful future business interactions.

Building on the low-level process definitions, Figure 3.10 shows that the following task for system analysts is the application of WS process specification technologies, to state these low-level definitions in terms of WS-level interactions (expressing them in terms of WS wherever appropriate). This transformation task is made much easier once the low-level processes have been stated to resemble *services*. WS should be viewed as the Internet-based implementation technology that will implement designed services. At this point, expressing the interactions from a global perspective (that is, showing interactions *between* companies rather than *internal* process flows) is desired as it allows for the creation of a contract that defines a jointly agreed set of orderings and constraint rules whereby service message exchanges can take place [224].

To facilitate the expression of this global services contract, the framework suggests one of two options, either (i) the use of W3C's Web Services Choreography Description Language (WS-CDL)—WS-CDL provides a standard mechanism for defining WS collaborations and choreographies of message exchanges from a global viewpoint [224], or (ii) BPEL4Chor—a recent proposal from the research community built on Business Process Execution Language (BPEL), that aims to address a number of perceived shortcomings of WS-CDL [44, 43]. These approaches were chosen specially because of their suitability for WS and ability to produce formal, Web service-level process specifications that could feed into future framework phases.

In deciding whether to use WS-CDL or BPEL4Chor, the framework notes

the following factors for consideration by businesses. In terms of politics in the standards world, WS-CDL is likely to have more support from industry because it is under the charter of the W3C. A second advantage of WS-CDL is that from the process specification defined, companies have the flexibility to then use their preferred internal technologies to implement the definitions. Furthermore, there is scope for automatically generating workflow templates for these internal technologies that mirror the global perspective [157, 121]. A high-level example is given in [224], where one company may use BPEL engines to drive workflow whilst another uses a more traditional J2EE^{TM}solution.

Another factor regarding WS-CDL is that if companies had chosen to use the UML Profile for Schedulability, Performance, and Time Specification [148] to model processes in the Architectural phase, research work by Cambronero [21] has investigated a method for translating those models into WS-CDL documents. This might be plugged in and used by companies to automate document creation.

Lastly, there is some free (albeit very limited) tool support targeted at providing users with the ability to produce, view, simulate and validate WS-CDL choreographies—namely WS-CDL Eclipse [5], Pi4SOA [163] and LTSA WS-Engineer [63]. Alternatively, software could be purchased, including Oracle SOA Suite 11g [151] (a complete suite of SOA products for development and execution) and IBM Rational Software Architect for WebSphere Software [88] (platform focused on development and deployment of SOA solutions).

At its core, the second approach, BPEL4Chor, defines extensions to BPEL to enable the definition of choreographies [44]. In light of this close association, BPEL4Chor can be seen to be specially suited for situations where businesses will desire subsequent BPEL workflow specifications for their internal process flows. The ability to allow for a seamless transition between choreographies (in BPEL4Chor) and orchestrations (in BPEL) is actually one of the main advantages this approach has over WS-CDL (when considering moving from WS-CDL to BPEL workflows) according to its proponents [44].

A second noteworthy factor is that if analysts have used BPMN to model

processes in previous stages, research by Decker et al. [43] describes how these BPMN models can be reused and largely transformed to BPEL4Chor. A plug-in for an available graphical modelling environment is also proposed to aid in this transformation. Decker et al. [44] could be referenced by companies for more nuances of this approach as compared to WS-CDL. In summary, WS-CDL and BPEL4Chor are both viable solutions for Web service-level process specification. With the information provided above, businesses can choose their technologies of preference.

Along with the low-level process definition shown in Figure 3.10, harmonization of process and data semantics across companies is critical. In this research however, this activity is not covered as it would necessitate an extensive discussion that digresses considerably from the focus on security. For information, some of the main aims during this stage would be tackling the semantic interoperability problem at both the data and business process levels. This problem as it relates to the B2B context is discussed in detail by Papazoglou [157]. Addressing these issues would likely include the use of tools such as ontologies, shared vocabularies, metadata repositories and depending on companies', also technologies such as Semantic Web Services.

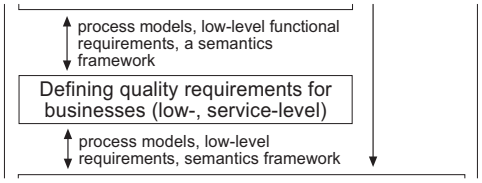


Figure 3.12: Definition of quality requirements task in Systems Design

Next is the determination of the quality requirements at these lower levels. For ease of reference Figure 3.12 illustrates the task. In earlier stages, quality requirements were produced at a high level and these form the base for the actions here. For this task analysts and systems designers play central roles.

Business teams will need to identify details such as availability of systems/services, acceptable latency levels, performance expectations by parties and

more general aspects including scalability, and even maintainability of envisioned systems. Work by Garcia and Felgar de Toledo [66] has compiled an appropriate listing of WS QoS attributes that can be used as a starting point by teams. These requirements and their relation to processes should be well conceived because they constitute prime factors against which the security design will have to be balanced. Businesses can either mainly discuss and agree on these quality requirements, or if a more hands-on approach is preferred, use available techniques to specify requirements. UML for example has a profile for modelling quality characteristics (see OMG [147]) that can be used.

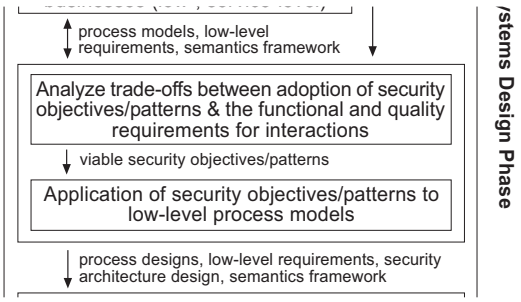


Figure 3.13: Security analysis and application tasks in Systems Design

The next step in BOF4WSS (shown in Figure 3.13) returns the focus to security and aims to finalize the security architecture and build the security design. The first task in fulfilling this aim is analysing the trade-offs between the adoption of security objectives/patterns and the low-level functional and quality requirements from prior tasks. Cost, where possible, should be generally factored in by business teams as it pertains to adopting the security directives, remembering that these will translate into security mechanisms and technologies later. Systems designers and security professionals with knowledge of this area can aid significantly in this task.

Sherwood et al. [186] yield a perfect example of the hard task faced by businesses in attempting to balance these often conflicting objectives. Abstracting to the three basic, conflicting aspects, namely security (for example, security requirements), cost (that is, a general limitation) and usability (normally a quality

requirement), the authors state, "To obtain higher security ... will cost more. To increase security often impacts upon usability and visa versa" [186] (p.27).

Once the analysis is complete, the viable security objectives/patterns are then applied to the low-level process models to fashion the business process designs. Figure 3.13 covers this task. In the Architectural phase, security objectives have already been applied therefore if businesses have utilized this method, the task now is to break down the secured medium-level processes and associate the objectives with lower-level process components (from the low-level models above). For example, as opposed to specifying a confidentiality objective of 'High' on all outputs from one activity (or task or system), businesses should consider the individual messages output and whether they all need the 'High' confidentiality rating. The messages should be visible from low-level process models, therefore the ideal situation would be to take the low-level models and modify them to show the new, specific levels of security required for all process components.

For the application of viable security patterns, depending on the modelling technique chosen, patterns can be easily woven into the low-level process models. Companies will first need to gather the associations made between the medium-level processes and security patterns from the Architectural phase. Then, using the associations, teams can begin to link low-level processes (from which the medium-level processes were defined) to the relevant security patterns. This is followed by the actual application of patterns to models either conceptually (by way of detailed annotations), or logically (within the formal models). Even though some techniques may prove more efficient at this application task, the conceptual solution that security patterns provide should enable a relatively manageable task for the security professionals on the teams.

Due to its versatility and extensibility, UML again forms one of the better techniques for the modelling task. In Figure 3.6 it was shown that, for simple modelling, sequence or activity diagrams are useful. To facilitate detailed modelling, one suggested option is the UML profile for security, quality and fault

tolerance requirements. This profile is defined in OMG [147] and provides a standard mechanism for expressing security.

Another noteworthy option still within the structured confines of UML can be found in Rodríguez et al. [172]. This research work supplies a UML profile specifically for secured business process modelling using activity diagrams. Security aspects accommodated include auditing, and security requirements such as integrity, attack detection, non-repudiation, access control and privacy. UMLsec [96] and SecureUML [112] are two additional, more detailed security-related extensions to UML that might also be of interest to businesses.

Figure 3.14: WS standards agreement and assessment tasks

The penultimate task in the Design phase depicted by Figure 3.14 is identifying and agreeing on the standards that will be used to implement the services, and especially the security and QoS requirements. In general, even though WS is one of the leading interoperability technologies today, basic tasks such as agreeing on standards (within WS) is still crucial to a successful deployment. Fischer and Werner [61] allude to this fact as they discuss the "What's missing" in WS. The main interoperability problems they identified stem from the existence of too many standards (innoQ [86] shows over sixty), the tweaking of standards by individual companies and the numerous versions of even the basic WS standards. Fischer and Werner [61] accept that WS-Interoperability (WS-I) [219] profiles can address some of these problems, however they note that this is only possible if companies make their WS compatible with the WS-I profiles. Their work provides just one example of the importance of the agreement on the standards to

be used by businesses.

In this task, systems analysts and designers knowledgeable in the intricacies of WS should take the lead. As analyst help to provide the bridge between the previous works (requirements, low-level processes and so on), designers look at service and technology details. Due to the extensive number of technologies available and the frequent updates made, instead of covering the standards within the framework, BOF4WSS documentation directs companies to key information sources which they can reference. Sources range from published texts [157, 194, 4, 28] for introductory- and intermediate-level material, to the standards Web sites such as W3C [225], OASIS [153], Liberty Alliance Project [110] and WS-I [219], for up-to-date, definitive information. InnoQ [86] is a good reference for a diagrammatic overview of standards placed within their context, including transaction specifications, reliability specifications and so on.

To identify security standards, the work of Steel et al. [194] is particularly relevant if companies have used their security pattern catalogue in previous framework phases. The reason for this is that within their catalogue, also supplied is a list of standards and technologies that can implement the respective patterns.

Briefly touching the topic of standards and technologies for QoS requirements, this area is less developed. Companies however can find some information in articles such as [230]. This covers a number of WS QoS aspects, mentions standards which are used to implement them and discusses techniques to improve service quality.

A final point companies should be mindful of during the identification and selection of standards is the tool support available to actually use the standards in a production environment. This consideration should help guide the choice of technologies. If there is an absence of tools, regardless of the benefits of standards proposed, these standards cannot be applied.

Having agreed on the standards and technologies to be employed, BOF4WSS

(see Figure 3.14) advises companies to consider (i) the common vulnerabilities and pitfalls in WS and the mechanisms chosen, and (ii) the best practices in using them and implementing them as securely as possible. Both of these factors may have been analysed in some respects before, but because of their significance and the complexities regarding technologies themselves, it is reiterated here.

As done above with standards and technologies in the previous task, because of the large number of vulnerabilities and range of best practices, BOF4WSS references more complete and detailed sources rather than listing them. For the first factor that is, for common vulnerabilities and pitfalls in WS, two prime sources are documents from organizations such as NIST ([189]) and WS-I ([218]). These give information on common attacks, risks and typical security challenges. The research community is another useful source for up-to-date information, for example, [92]. Examples of vulnerabilities in Jensen et al. [92] range from those linked to XML or networking generally, to more recent ones which target WS cryptography techniques, BPEL processes and internal workflow engines.

For the second task in Figure 3.14, namely the consideration of best practices in using standards and dealing with the various security challenges, the following articles provide designers and developers with some useful techniques. Industry-based work by Singhal et al. [189] provides general guidance in addressing threats and on secure implementation tools and technologies. Jensen et al. [92] outline a few of the general technology-related mechanisms to defend against attacks in the WS arena. Authors in [194] give various best practices and design strategies. WS-I [218] identifies typical countermeasures (technologies and protocols) to mitigate common WS threats.

Finally, [187] lists techniques to protect against more threats to WS. In light of the vulnerabilities and best practices discussed, BOF4WSS gives companies the option of revisiting the preceding task to reassess the standards and technologies chosen. This progression can be seen in Figure 3.14 and is highlighted because depending on vulnerabilities or best practices, teams may often opt to use different, more robust standards or technologies with extensive guid-

ance (practices) on their use.

This completes the Design phase and the expected outputs are *low-level process designs, service-level interaction definitions, security architecture design, a semantics framework, the standards and technologies of choice to implement the WS interactions and the low-level requirements (functional, security and quality).*

3.9 Agreements (for QoS) Phase

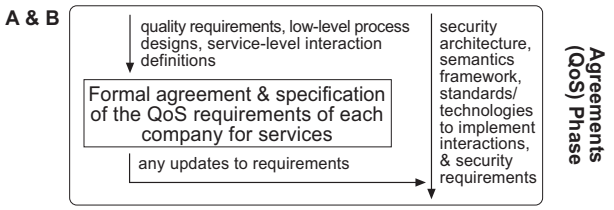


Figure 3.15: Workflow model of the Agreements (for QoS) phase

With the low-level process designs and service-level interactions defined, the **Agreements phase** now concentrates on the agreements necessary at the QoS level. During the task shown in Figure 3.15, the goal is to actually specify the mutual understanding of the priorities, responsibilities and guarantees expected by each business with respect to the other entity, regarding the actual WS. This phase directly extends the preparatory work on quality requirements in the Design phase and results in a set of formal and contractual agreements.

As done before, QoS requirements typically assessed include service availability needs (such as, a service uptime of 99.98%), performance requirements (for example, average response time of 30 milliseconds) and so on. Besides quality requirements, process designs and service interactions are necessary for input because they too need to be considered in defining appropriate QoS levels for services and systems.

To specify the QoS requirements agreed, businesses' executives, analysts and lawyers (these are the people that would be directly involved) have a few alternatives. The first and most common option is a contractual, natural language agreement referred to as a Service-Level Agreement or SLA. SLAs date back to

many years before WS and since their inception have proved very useful mechanisms to define levels of service in a measurable way (to allow for monitoring), and also the penalties where agreed levels are not fulfilled. For WS, SLAs will have the same usage and general mode of application. The only difference may occur in how services are monitored, as more WS-specific tools and techniques are likely to be employed which enable increased granularity and efficiency in monitoring. For more details on SLAs and what can be included in a WS context, BOF4WSS directs companies to reference [157, 131].

Another option is to make use of accepted policy standards such as WS-Policy to specify a service's quality requirements [65]. This method however is ideally suited for dynamic interactions where quality requirements greatly influence the services, or service providers chosen for use. The last noteworthy approach is the Web Services Level Agreement (WSLA) framework described by Keller and Ludwig [102]. Broadly, this framework allows for the technical specification and monitoring of SLAs for WS. It enables service users and providers (or companies in BOF4WSS' context) to define a variety of SLAs, specify the SLA parameters (including availability and response time) and the method for their measurement, and finally relate them to implementation systems. Implementations of the WSLA framework have been built and are available for use in some IBM products [89, 87].

Once the specification of the QoS requirements of each company for services is complete, the outputs of the phase to be made ready are *QoS agreements, low-level process designs, service-level interaction definitions, security architecture design, a semantics framework, the standards and technologies of choice to implement the WS interactions and the updated low-level requirements (functional, security and quality).*

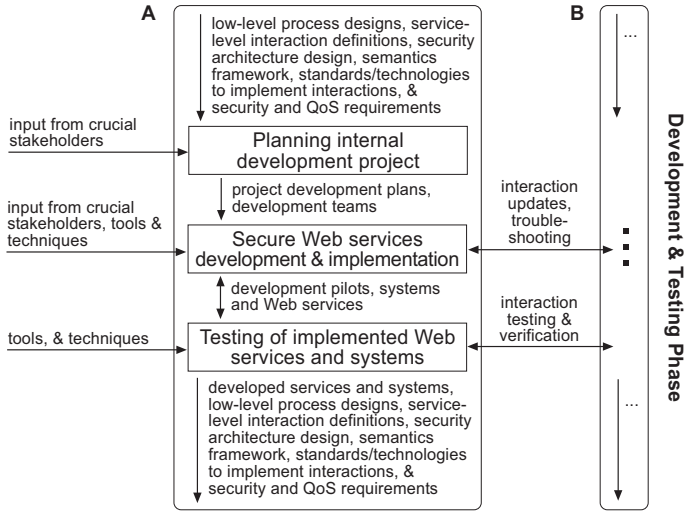


Figure 3.16: Workflow model of the Development and Testing phase

3.10 Development and Testing Phase

As with most methodologies, the penultimate stage in BOF4WSS is the **Development and Testing phase**. Having discussed how services and systems would interact in a cross-enterprise context, this phase (shown in Figure 3.16) is centered on the actual development, implementation, deployment and testing of services and systems in the companies. Because of this factor, it is mainly carried out by companies individually. This involves all members of the project teams from each company working on their own systems development, and this development would be guided by previous companies' agreements. Occasional, or even prolonged joint interactions are however greatly appreciated especially for services testing, updates, troubleshooting and systems verification to the requirements established in previous framework phases.

All the inputs to this phase are to be used by companies and their development teams to steer the internal systems implementation. It is stressed that even though Testing is presented last (after discussing Development), companies may choose to do some testing as services and systems are developed.

Unlike some of the previous tasks covered by BOF4WSS, activities for the development stage appear to be somewhat well-established in literature and practice. This is consistent with this research's argument regarding the significant

focus on technology-based and -oriented solutions (which are dominant during this phase). The benefit of this to the framework is that there are a variety of tested development processes, techniques and tools that can be plugged in during this framework phase.

Practically therefore, this phase is much less strictly prescribed, with Figure 3.16 mentioning only three very generic tasks (Planning, Development and Implementation, and Testing) which are not structured in detail like prior tasks. BOF4WSS's aim at this point therefore becomes the identification of relevant, mature and largely complete development processes, techniques and tools that can be employed, and allowing companies the freedom to combine them to best suit their respective situations. Two such processes which might be of great interest in aiding in this internal process are described in [157, 75].

In the first process mentioned above, Papazoglou [157] presents a WS lifecycle methodology that concentrates on critical internal aspects. These include application integration, packaging legacy applications into reusable components, migration from old to new WS-based processes, and the 'best-fit' ways of implementation which appreciate company constraints, risks, costs and returns on investment. This methodology is cyclic (as opposed to linear) and consists of nine stages, namely Planning, Analysis, Design, Construction, Testing, Provisioning, Deployment, Execution and Monitoring. This process is one of the most appropriate and comprehensive within the literature. It covers from initial analysis of internal systems to the construction and final installation or deployment of services.

A caveat to the lifecycle methodology however is its lack of emphasis on security concerns—a prime target and goal within BOF4WSS. To compensate for this shortcoming, another suggestion businesses might consider is the integration of PWSSec [75]—a detailed development process for creating secure WS. This would be integrated such that it could run in parallel.

The novelty behind PWSSec is (i) its appreciation of the complex task faced by businesses as they attempt to make use of WS, (ii) the highly structured,

methodical approach to constructing a security architecture for WS systems, and (iii) the emphasis on traceability and reusability which translates into the establishment and use of a number of repositories and record stores. The three phases in PWSSec are Web Services Security Requirements, Web Services Security Architecture and Web Services Security Technologies. These work together to enable the development of secure WS systems. In brief, another general point of reference to supplement the two already mentioned can be found in [234]. This text provides some useful guidelines that can be applied within the planning task, related to planning and staffing a WS development project.

Probably the biggest benefit of using the processes listed above is that almost all of the information gathered and produced earlier in the framework can be reused to quickly complete their initial stages. Such information includes functional, security and QoS requirements, risk assessment data and business process models. To consider the Analysis phase in Papazoglou [157] for example, in BOF4WSS's Requirements Elicitation and Architectural phases, companies have already worked on the current and envisioned (or "to-be" processes).

Regarding the Design phase (in [157]) and the specification of business processes (looking towards WS-CDL and BPEL), BOF4WSS's Architectural and Systems Design phases have previously defined business processes to even these lower levels. Even though the framework's focus was towards WS-CDL (and BPEL4Chor), these process definitions can be converted to the BPEL advocated in [157]. For the more security-specific PWSSec [75], the medium- and low-level security requirements and security patterns identified from BOF4WSS can be reused in PWSSec's Requirements and Architectural stages. That approach ([75]) also uses UML and the profile for security ([147]) for some of its modelling—this is a method supported in BOF4WSS. These are just a few concise examples of how the outputs from BOF4WSS's previous stages can be reused in these processes.

In addition to the identified processes, as mentioned above, literature has supplied a number of techniques and tools to help in this internal development task. An area in particular which has received great focus is the automated

creation of BPEL processes from theoretical modelling techniques (such as UML and BPMN). To recap, BPEL allows for the specification of business process behaviour based on Web services [152]. Amongst other things, it is an execution language which can be run by software engines (such as [123, 150]) to orchestrate message, control and data flows. If companies have modelled processes in UML or BPMN therefore, techniques such as [190, 154, 20, 114] that offer some aid in translating these models to executable processes (in BPEL) are ideal. Specifically, [154] works with the translation of BPMN models to BPEL definitions, whereas [190, 20, 114] aim at transforming their various UML variants and extensions to their respective BPEL representations.

A critical activity in the Development phase is the implementation of the security standards and technologies that have been agreed. Implementation includes the actual application of standards and security levels to the services and systems, but also the correct configuration of the security mechanisms employed. Even though output from the previous phases gives a clear outline of security and where, and to some extent how, it is to be applied, noting the peculiarities of WS (such as service policies, federated security), this task is still far from trivial.

Researching security configurations for WS, Tatsubori et al. [198] highlighted the difficulty in this task and the usability problem faced by developers regarding choosing cryptographic algorithms, encryption keys and so on. To aid in this activity therefore, they propose a tool to fill the gap between business-level security requirements and the lower-level, concrete, technology-specific policies implementing them. This GUI tool, called the WS-Policy Organizer (WSPO), enables users to partially create a platform-specific WS-SecurityPolicy document from a somewhat high-level process definition, through the use of a number of preset security patterns. The integration of this tool within the framework should be reasonably simple because the process scenarios necessary are available from previous BOF4WSS stages. Secondly, the preset security patterns used can easily be matched to the security objectives and patterns from the Architectural and Design phases.

Before moving on, it is worth explicitly stating the importance of including tools for monitoring both the QoS levels defined in the SLAs and the security implementations for their reliability and robustness. QoS monitoring constitutes the main focus of the Monitoring stage in the WS lifecycle methodology from [157]. Companies that use that methodology therefore can receive more information on it there. Regarding security monitoring, the key is to install software to maintain adequate logs, audit trails and records that can be referred to as required. Steel et al. [194] highlight that having these audit trails has even become a requirement of some laws, for example, SOX. Intrusion detection or prevention software may also be of interest to businesses.

The final task within this phase is the testing of the developed WS and systems. This is done to verify that the developed applications meet the intended requirements. It can and should be done at a cross-enterprise level (that is, internally and externally across companies). Testing can occur from three main perspectives, functional (do WS do what they should), quality (are the set performance, usability, scalability, etc. requirements met) and security (is there adequate protection in place for WS and systems).

Guidance on testing the functional and quality requirements is given in the lifecycle methodology [157] mentioned before. A much more complex operation is testing the security of the applications developed. Whereas one can pass input data into a system or process and (based on the output) quickly determine whether a functional requirement has been met, security is not that absolute nor can it be so easily measured [186]. This does not however mean that testing is impossible nor should it be viewed as a task to be avoided by businesses.

Like approaches for the other testing perspectives, the initial activities are the same, therefore, identify requirements (these may be in terms of actions, goals, threats that should be handled), and carry out controlled tests to see if, or how well requirements have been addressed. For testing the security of the implemented WS, Barbir et al. [8] offer a number of strategies and guidelines.

These are both generic (that is, just highlight the use of test suites, patterns and so on) and targeted (focus on specifics such as testing application data).

Vulnerability analysis is another aspect that needs to be addressed in detail during testing. For this task, companies can refer to [229] regarding various guidelines on software vulnerability analysis for WS. These include checking for cross-site scripting, services traversal, DoS attacks and access validation attacks. Businesses can also reuse the original listing of threats that were factored into security requirements determination, and conduct penetration tests against services to evaluate how well the implemented security addresses these threats. Particularly keen companies, or businesses that lack the expertise internally, may consider employing security companies to conduct these tests. This decision however, should not be taken lightly as exposing systems to external parties demands great amounts of trust.

Processes, guidelines and techniques are all essential in testing, but to enhance or at least ease this task, tool support would be ideal. Unfortunately, there has not been much notable work in this area as yet, perhaps this is likely because WS testing is a discipline still in its infancy [8]. One tool that has surfaced however is wsChess [134]. wsChess is described by its makers as a freely available toolkit for WS assessments and defense [134]. Sidharth and Liu [187] give a brief example of how wsChess can be used to probe for vulnerabilities and formulate attacks against services.

To assess Web applications that may constitute part of the developed WS systems, a number of tools are available. Curphey and Arawo [37] provide a comprehensive, albeit slightly dated source of information on these tools and an objective discussion of their aims. These industry-based researchers also outline a taxonomy of tools which encompass prime testing areas such as source-code analysers, Web application scanners, runtime analysis tools and configuration management tools, to assist companies with their tool selection [37]. The tools and techniques supplied here and those available from other sources should be used wherever possible to enable thorough, adequate testing of the developed WS

systems. This testing activity completes the Development and Testing phase, and phase outputs are the *developed services and systems, low-level process designs, service-level interaction definitions, security architecture design, a semantics framework, the standards and technologies of choice to implement the WS interactions and the low-level requirements (functional, security and QoS agreements).*

3.11 Maintenance Phase

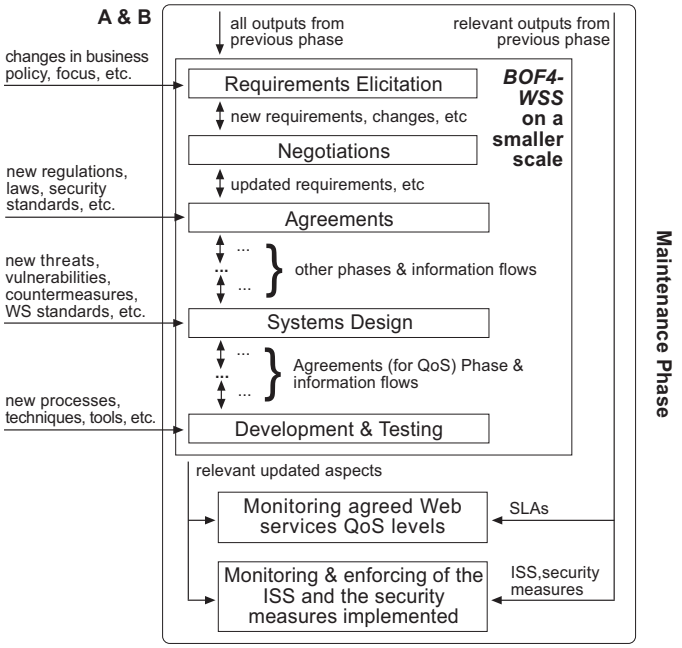


Figure 3.17: Workflow model of the Maintenance phase

Having developed this comprehensive, multilayered security solution, its upkeep becomes the next crucial undertaking. BOF4WSS addresses this and other typical monitoring and preservation tasks in the **Maintenance phase** shown by Figure 3.17. It is important to recognize that this phase is a continuous one (unlike the others which have clearly defined endpoints) and will last for the lifetime of the implemented systems.

Specifically, this stage will involve continuous functional and quality-based system enhancements, but additionally will stress the continued updating and

enforcement of security measures, both in developed systems and the overarching ISS. To facilitate the required maintenance activities, the framework strongly suggests that businesses form cross-enterprise maintenance and monitoring teams. Ideally, the majority of the persons chosen should be members of the teams that participated in the BOF4WSS process. The advantage of this is the experience they bring. One team already mentioned is the security team from the first BOF4WSS agreements stage. These personnel are responsible for monitoring the internal and external environments and considering new threats, laws and security requirements, and how these will be included in system updates.

When considering the updating activities of the Maintenance phase, companies must be extremely careful in how they make changes and updates to cross-enterprise agreements, directives and systems. This is even when these updates are agreed by both companies involved. Changes should not be made in isolation without first analysing what effects they might have on other system aspects and whether respective updates to these other aspects would be necessary. It is for this reason that a smaller scale BOF4WSS process is suggested in this phase (see Figure 3.17). Reiteration of this process for new needs in the form of updates and changes allows modifications to be made in a structured and controlled context. Repetition of previous phases is not uncommon during software maintenance as noted by Sommerville [193].

Because the BOF4WSS process has been discussed in detail previously, it is not covered here or in Figure 3.17. Instead, Figure 3.17 is used to display some of the key **new** inputs (in addition to the ones outlined in previous phases) which are very likely to surface. Examples are, changes in the business policies (reflecting possibly new goals and aims), new regulations (therefore new, mandatory security actions for interactions and systems), new threats and vulnerabilities (these need to be assessed and addressed), and new techniques and tools (these may facilitate easier development or even system testing).

The other tasks depicted by Figure 3.17 focus on monitoring in general, but specifically as it relates to (i) QoS levels and (ii) the monitoring and enforcement

of the ISS and implemented security measures. In the first task, the goal is to take the actual service levels (recorded by management, auditing or tracking software added in the Development phase) and compare them with the SLAs and QoS agreements made earlier, to determine if quality requirements are being met by parties. Particularly of interest to companies will be aspects that affect general WS performance levels such as service response times, and system downtime and latency. SLAs are also the point of reference that dictates the penalties and options for recourse if the agreed levels are not fulfilled.

The second task deals with the monitoring and enforcement of the ISS and the security measures implemented. Security, in every regard, is a constant process. In their description of information security, Calder and Watkins [19] liken it to a journey, not a destination. Within this journey, monitoring of the implemented security mechanisms and rules is critical. The reason for this is that new threats may surface, new attacks might be launched and consequently, there needs to be constant monitoring to detect (and initiate a reaction to) these advances. Again, the output (audit trails, logs) from monitoring and detection software is used in this activity.

Beyond tracking new threats and attacks, it is imperative that companies use this information to identify areas where directives and measures may need to be enforced, both *internal* and *external* to a company. Therefore, in addition to monitoring and following up on internal security concerns, business partners should be periodically assessed to ensure that they are maintaining the agreed levels of security. These levels can be found in the ISS and systems design documentation, amongst other documents. Some of the common options to assess the security posture of partners has been covered before (in the first agreements phase) and includes audits (done by a third party possibly) and on-site visits.

A final noteworthy aspect shown in Figure 3.17 is that as smaller scale BOF4WSS processes are conducted to accommodate for updates, the final updates are then re-input into the respective monitoring tasks. This is done to keep the information used for monitoring as up-to-date and relevant as possible. This

last task concludes the BOF4WSS process. Next, a brief summary and justification of BOF4WSS is presented. That section also identifies the main target group of businesses for which the framework is intended.

3.12 BOF4WSS' Scope

Reflecting on BOF4WSS in its entirety, especially with regard to its use by companies, this is not a process to be taken lightly. In the design of this framework, not only were security practices within WS and business processes in general assessed, but also literature on joint business ventures such as the extended enterprise (for example, in work by Davis and Spekman [40]) and how security and trust—beyond the technical layer—is reached and maintained across enterprises there. With these factors in mind, the framework is thus aimed particularly towards businesses that *emphasize trust and medium-to-high levels of security and expect long-term interactions as opposed to the short-term, highly dynamic, e-marketplace-type interactions also possible with WS. Ideally, a set of business partners in the early planning stages for a WS project will adopt BOF4WSS to create an agreed, communications security infrastructure. Collaboration, which encompasses business and technical layers is the real focus of this approach.*

Another prime application case for the framework is in *extended enterprises* that emphasize security, which utilize or are considering adopting WS to support a business scenario. The suitability of the framework to this context should be no surprise as the security approaches and research problems in the extended enterprise domain were considered in the development of the framework.

Due to the long-term nature envisioned, it is not expected that companies will frequently enter or leave the business scenario, therefore scalability is not a critical issue. Should companies be added however, it is crucial that they go through some of BOF4WSS's phases. It will be up to existing businesses whether the new partners adopt the active security charters and infrastructure, or if they all recomplete key security-related framework phases.

In general, the framework tasks to be executed when new partners join will be very context dependent. For example, depending on the new company and its purpose, additional services may need to be created by all companies, or only a small subset of companies. The extent of the services necessary, or the companies that are required to make modifications to their systems, will then determine the level of systems development that is required using BOF4WSS. There may even be cases where new partners already have their systems exposed as services and therefore technical integration is not a problem (therefore no need for in-depth emphasis on later framework phases). In situations like these, existing companies may choose to focus more on initial phases of BOF4WSS, such as identifying risks and negotiating on security actions. Then, they would look towards ensuring that all companies share the same goals with regards to cross-enterprise security. The ISS would be very relevant in this regard.

E-businesses adopting this framework will have to be committed to collaborating and devoting resources—financial and nonfinancial (including time, skills and experience)—to this business venture. If these resources are not available internally, employing consultancy firms, particularly in terms of process reengineering and security, should be an option. Generally therefore, the framework will require changes in how the businesses worked before WS adoption, but potential benefits to WS use might merit this.

3.13 Summary

This chapter presented a solution approach targeting the core problems identified in Chapter 2. To reiterate, these problems centre around an overly reliant emphasis on technology for security and security approaches that are often too isolated and individualistic. The BOF4WSS approach attempts to tackle these issues by concentrating on all components of the security environment and stressing a much more collaborative approach to businesses' security. These steps also have the secondary design goal of fostering trust across the interacting entities.

Within the chapter, each of the nine phases of BOF4WSS was discussed in detail. This discourse included considering the inputs and outputs of phases, key activities to be undertaken and critical stakeholders that should be involved in each phase. In addition to providing details of all the individual parts of the framework, this chapter presented information on how the aspects would enhance collaborative e-business security. The last section rounded up the presentation by highlighting likely areas and scenarios where the framework might be best suited, stressing the point that BOF4WSS might not be ideal for all scenarios.

Chapter 4 builds on the detailed framework presentation and aims to outline a thorough example of how the framework would work in practice. This consists of applying BOF4WSS to a designed scenario. This seeks to add a more realistic perspective to the framework's discussion and provides a clear example of its use and application.

Chapter 4

Applying BOF4WSS to a Scenario

Example is always more efficacious than precept. — Samuel Johnson

4.1 Introduction

In Chapter 3, BOF4WSS was presented as an approach for enhancing the security of companies that conduct business online using Web services (WS) technology. This chapter builds on that presentation and aims to provide a comprehensive example of how BOF4WSS would work in practice. Specifically, this consists of applying it to a developed scenario. Given that the scenario would be largely fictitious (considering research limitations), a number of published articles such as [127, 158, 157, 30, 194, 203, 137] (a majority of these containing cases themselves) were referenced for guidance and real-world requirements. To further strengthen scenario practicality, particularly in terms of security, an IT security professional with experience in cross-enterprise interactions was informally interviewed and his input used to inform case development.

The purpose of this chapter therefore is to supplement Chapter 3's largely abstract discussion and supply realistic examples of BOF4WSS' use. This is expected to add to the practicality of the framework's proposals, to highlight

some of its usages/benefits and also to identify any possible problem areas (as this would feed into future chapters). Taking steps to achieve this goal, the next section outlines the scenario to be used. This is followed by a phase-by-phase application of BOF4WSS to that scenario. Finally, the last section discusses the framework application in terms of the uses in and advantages to the scenario and in general.

4.2 Scenario Background

This section provides a very brief overview of the scenario **BEFORE** the framework has been applied.

Focus: Raw materials procurement process—that is, all activities which are involved in a company acquiring items (raw materials for production) from its suppliers. These activities generally include (i) searching for items required, (ii) purchasing the right quality/quantity items and at the right time, (iii) making the final payment for items, and lastly (iv) receiving the delivery of those items.

Participants: `Supplier` is a medium-size seller of high-performance electrical parts (raw materials). `Buyer` is a high-tech equipment manufacturer which uses a variety of high-spec electronic parts to manufacture a set of final products.

Background: In the past, `Buyer` has used various suppliers for raw materials under many short-term contracts. The company's procurement process in that regard is primarily conducted using Electronic Data Interchange (EDI) for automated data exchange, and manually using phone, fax and email with spreadsheets.

To gain a better competitive advantage and minimize the administrative overhead, `Buyer` is deciding to work closely and more exclusively with its prime supplier, `Supplier`. This is expected to be a strategic alliance and entered into for the foreseeable future (5-10 years). To enable their processes to be more

integrated and streamlined, the companies are choosing to use the Internet and WS technology suite for business-to-business communications.

The security of this Web-based scenario and its cross-enterprise communications is also of great importance to both parties. This is attributed to the large volume of expected transactions, the proprietary nature of some of the data transferred, the costs of the items (hi-tech) involved, and the large sums of money constantly exchanged. Furthermore, in the past there have been targeted denial-of-services and virus-based attacks on `Buyer`'s Web sites from unidentified external sources.

As a result of these factors, the companies are choosing to adopt BOF4WSS to aid in the creation of a secure WS-based business scenario. The companies are expecting that the framework would be able to supply a comprehensive yet flexible cross-enterprise methodology, that also encompasses WS-specific guidance to support secure scenario implementation. Flexibility is particularly important as businesses wanted to be able to choose and plug in their own techniques, tools, software packages, standards and such.

In addition to creating fully functional interactions, `Buyer` and `Supplier` are especially keen on attaining: (i) adequate and **agreed** security across partners, (ii) a security governance structure beyond rigid contracts and technology products and systems, (iii) traceability in original security needs through to final technical and policy implementations, and (iv) generally, a inter-organizational business scenario that all partners can trust.

Having presented the general scenario, the following sections show how BOF4WSS is applied to it. The aim is not to be exhaustive in the discussion, but rather to give practical details of the flow of processes and tasks which are involved. As such, each section is partitioned into **Task** and **Case** segments. A **Task** represents guidance by BOF4WSS (typically in the form of some activity companies should/could do), whereas a **Case** is the application of that guidance to this particular scenario. Throughout the presentation, emphasis is placed on

security components rather than detailed coverage of functional or QoS aspects. This is in line with the framework's core security focus and allows those issues to be adequately explored.

In the discussion of the first phase, the process is considered from one company's perspective, that is, `Buyer`. This was done to maintain clarity, and is similar to how the framework advocates a largely individually completed Requirements Elicitation phase. `Supplier` would however go through the same general process internally. Phase outputs are shown for `Supplier` as necessary.

4.3 Phase-by-Phase Application

4.3.1 Phase 1: Requirements Elicitation

Task 1: Forming teams and then gathering knowledge about processes related to the envisaged scenario, identifying influential stakeholders, process or business constraints, and so on.

Case: As suggested in the framework, `Supplier` and `Buyer` first begin by creating teams responsible for the cross-enterprise project. These consist of project managers, process domain experts, business and systems analysts, system developers, end users, legal counsel, and IT security specialists/professionals. After forming teams, they set about speaking to their business managers and personnel involved in related procurement and order processing services (domain experts, users and so on). Some of these persons will already be on the teams formed. Having identified these sets of persons, the teams collect documentation on how the existing processes function. This includes models, produced reports, functionality charts and so on.

Task 2: Use the data collected to define and analyse existing processes to thoroughly understand the current flows. This activity will enable the identification of the core processes to carry forward.

Case: Following the guidance in the framework, `Buyer`'s domain experts and an-

alysts define the current procurement process (largely done manually) at a high level. That process is shown below.

Step 1. `Buyer` needs materials. `Buyer`'s employee queries a number of suppliers to determine which of them could fulfil the company's need. Once found, `Buyer` selects materials, quantity and sets delivery schedule. The Internal department then approves order. The order is placed (using EDI, an online system, email or fax) with selected supplier.

Step 2. The chosen supplier receives order, and `Buyer` is billed (using EDI, an online system, email or fax)

Step 3. `Buyer` receives goods shipment and checks that the specific goods ordered have been received. Accounting Department checks goods received against the original order placed and then issues payment to the supplier.

Figure 4.1 gives more detail of this process flow, the inputs, outputs, and actions involved using a UML diagram defined by `Buyer`. UML is one of the prime techniques suggested within BOF4WSS.

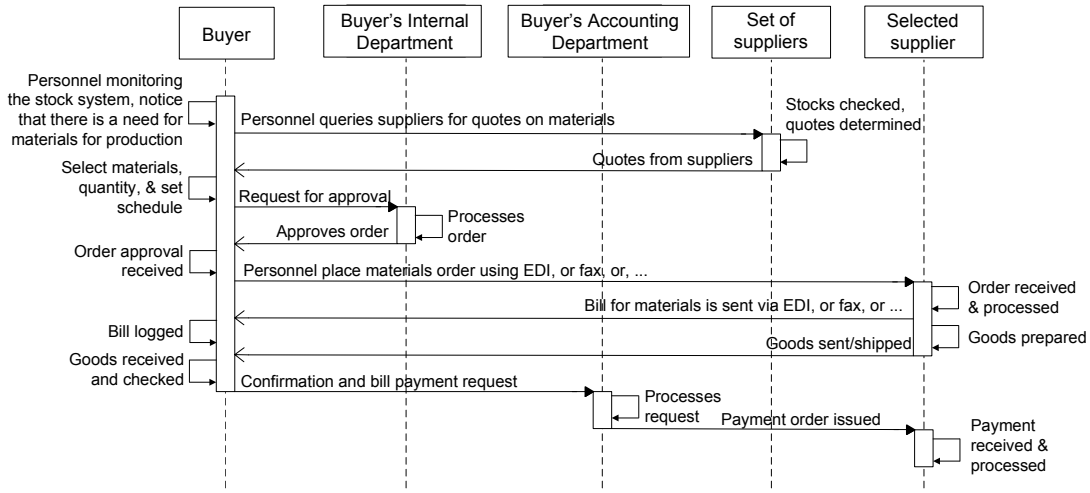


Figure 4.1: Current process displayed in a UML sequence diagram

Task 3: Model the envisioned business processes. This may involve process redefinition/redesign depending on how different the new processes are to be.

Case: Based on the information and current needs, `Buyer`'s domain experts and analysts define the envisioned procurement process:

Step 1. `Buyer` needs materials. `Buyer`'s system recognizes this need (because minimum stock levels have been reached) and identifies materials, quantities, and sets delivery schedule. The order is sent to the Internal department for approval. The Internal department approves order (manual). `Buyer`'s system places the order with `Supplier`'s system over the Internet using WS.

Step 2: `Supplier` system receives the order from `Buyer`, and `Buyer` is billed using Web services

Step 3. `Buyer` receives goods and checks that the specific goods ordered have been received. Accounting Department checks goods received against the original order placed, and then directs the system to issue payment to `Supplier`.

Figure 4.2 is created by `Buyer` to provide more detail into the new process.

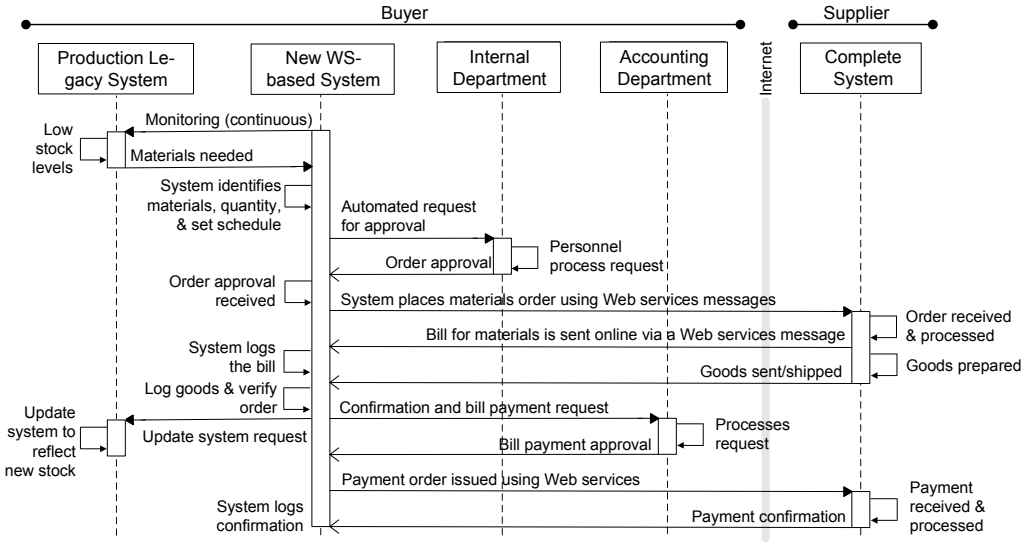


Figure 4.2: Envisioned process displayed in a UML sequence diagram

Task 4a: Analyse the process flows at a relatively high level to identify what participants should be able to do, and what inputs, outputs and actions are required for a process to be successfully completed. If UML sequence diagrams

are being used, it should be relatively straightforward to identify these. This produces the *functional requirements*.

Case: With consideration of the sequence diagram in Figure 4.2, business analysts at `Buyer` note that there should be a Procurement system capable of the following functionalities (inputs and outputs are clearly shown in the figure and are therefore not specifically identified again here).

i Continuously monitoring the stock levels for all production materials. The Procurement system will therefore need to interface with the existing legacy stock system—this is a system which has up-to-date information on all the materials needed in production.

ii Automatically generating an electronic purchase order depending on materials needed, including quantity and schedule of dates.

iii Passing the order for manual approval to personnel in the Internal Department (possibly using email).

iv Subsequently enabling the electronic purchase order to be sent (using WS messages) over the Internet to an order processing service system at `Supplier`'s enterprise.

v Logging bills sent by `Supplier` using WS.

vi Interfacing with personnel in the Accounting Department and allowing for their manual approval of bill payments.

vii Issuing payment orders to `Supplier` for goods supplied.

viii Interacting with the existing legacy stock system to enable new stocks to be recorded and stock levels updated.

In terms of BOF4WSS' suggestion that companies should identify quality requirements, `Buyer` and `Supplier` personnel opt to define these aspects at a later date in the Systems Design phase and last Agreements phase.

Task 4b: Identify the general security actions and requirements for the processes and the scenario.

The first step is to analyse access restrictions on processes, actors etc., and assess the threats, vulnerabilities and risks that can affect processes. Next would be a scenario risk assessment as this enables a comprehensive security-driven analysis of the overall business scenario to be undertaken. The general process is to identify risks, estimate and evaluate them, decide on possible treatments, and finally, elicit and generalize to the high-level *security actions and requirements*.

Case: To identify security necessary for the scenario, the IT security specialists on the project team at `Buyer` choose to apply the NIST Risk Management approach [195]. This is the preferred approach in their company. Instead of presenting the complete process, a few of the key aspects of the approach are shown below. First, in Table 4.1 is part of the list of assets, vulnerabilities and threats within the scenario that are identified by the security specialists.

Item	**Asset**	**Vulnerability**	**Threat source**	**Threat action**
6	WS messages	Sending information over the Internet in inappropriately secured formats	Malicious party	Eavesdropping and tampering with data in a WS' message (in transit)
7	Procurement system	No facilities for the logging of transactions between `Buyer` and `Supplier`	Third party, Malicious party, Insider	Repudiation of services; unauthorized changes made to orders or system
8	Sensitive WS messages, for example, payment orders	No allowance for sensitive messages to have added security	Malicious party with sophisticated tools	Unauthorized access gained to sensitive data. For example in payment orders, they are bank codes and account numbers

Table 4.1: Assets, vulnerabilities and threats faced by `Buyer`

From here, the next step is the identification and estimation of related security risks in Table 4.2. Table 4.3 supports this by outlining descriptions of risk levels from the NIST Guide.

The next aim in the Guide is to assess the treatment options for risks in terms of related organizational policies, laws and regulations, and effectiveness of recommended options/controls. For the first two risks in Table 4.2, the following information is key in making the treatment decision.

Risk no.	**Related item**	**Likelihood**	**Impact**	**Risk Level**
RSK-12	Item 6	Medium	High	Medium
RSK-13	Item 7	Medium	Medium	Medium
RSK-14	Item 8	Medium	High	Medium
RSK-15	Item 9	Low	Medium	Low

Table 4.2: `Buyer`'s risks faced and their estimated values

Risk Level	**Risk Description and Necessary Actions**
High	If an observation or finding is evaluated as a high risk, there is a strong need for corrective measures. An existing system may continue to operate, but a corrective action plan must be put in place as soon as possible.
Medium	If an observation is rated as medium risk, corrective actions are needed and a plan must be developed to incorporate these actions within a reasonable period of time.
Low	If an observation is described as low risk, the system's approving authority must determine whether corrective actions are still required or decide to accept the risk.

Table 4.3: Risk scale and necessary actions ([195])

For Risks RSK-12 and RSK-13, the Sarbanes-Oxley Act of 2002 (a United States federal law) is a critical consideration. This act requires that companies are able to confirm that only authorized persons have access to private information and sensitive systems. Additionally, audit trails are necessary to ensure this is done and enable other types of checking and verification. Regarding RSK-12 specifically, `Buyer` also has a security policy that strongly advocates the protection of integrity and confidentiality of all non-public communications.

With the aspects above in mind and a risk level of Medium, security specialists at `Buyer` decide to mitigate the two risks and define respective security requirements to achieve mitigation.

Applying the general process above to find all risks, threats and so on, `Buyer` derives the following high-level security requirements (recall that a security requirement represents the application of a mitigation action to treat a risk). Only some are shown noting space considerations.

1. The Web services messages to be used between companies to facilitate business transactions cannot be adequately secured by normal transport-level security (that is, Secure Sockets Layer or Transport Layer Security). These

mechanisms provide only point-to-point security and not the end-to-end security needed by Web services (readers can view Boncella [17] for details). As a result, measures should be taken to support security at the Web services message level to ensure data integrity and confidentiality. This relates to risk RSK-12 above.

2. A part of reliable business process execution both within the company and externally (thus, with trading partners) involves comprehensive logging and subsequent enforcement. This especially relates to the maintenance of an audit trail for transaction monitoring, which would also facilitate tracing and accountability checks when required. This relates to risk RSK-13.
3. The interactions within and external to `Buyer` (that is, with business partners such as `Supplier`) involve the transference of a wide variety of information. This information has varying levels of sensitivity. Appreciating this fact, measures should be taken to secure this information (in terms of confidentiality and privacy) to mirror these sensitivity levels. Classification levels of High, Medium and Low are proposed to do this, with different degrees of security on each level. To enable this, all information should first be classified. This relates to risk RSK-14 above.
4. No matter how secure and sophisticated the application infrastructure is, a denial-of-services attack can cripple companies or business partners by making their applications/services offline and unavailable. Thus, companies and business partners should adopt appropriate measures that can help defend and respond against a security breach and ensure further service continuity without disrupting legitimate user/system requests. (Adapted from Steel et al. [194])

The security requirements to be assumed as output from `Supplier`'s Requirements Elicitation process are next.

Contrary to the bullet-point requirements listing `Buyer` produces, security professionals at `Supplier` employ two other formats/techniques to express their

requirements. The first of these formats is a predefined, standard security requirements checklist commonly used by the company when dealing with external parties. An excerpt of the checklist is displayed in Table 4.4.

Item Description	**Yes/No**
Requirement 1. There should be an infrastructure plan that ensures the capability of having systems, applications and services available in the event of a security breach or accidental (human or natural) incident. If such an event occurs, `Supplier`, and previously defined business partners should have mechanisms in place to recover from the event. Such mechanisms may even stop the event from occurring in the first place. (Adapted from [194])	
Requirement 2. Organizations participating in a B2B transactions should have security directives in place that address the topic of viruses, malware and all other malicious types of applications. Typically this should be handled by a well-known antivirus software, which is updated regularly, installed on the local networks, and used to scan all networks periodically. ([203] aided definition)	
Requirement 3. Authentication is critical and should be used.	

Table 4.4: `Supplier`'s security requirements checklist

To complement the standard requirements checklist, professionals from `Supplier` apply the CORAS [47] risk assessment technique to identify other necessary security requirements and risk treatments. As security team members from `Supplier` are of the opinion that diagrams would be more useful for companies' discussions in the Negotiations phase, they supply the diagrammatic output from the CORAS technique. An example of the requirements diagrams which `Supplier` would bring into the Negotiations phase of BOF4WSS are shown in Figures 4.3 and 4.4. These figures display `Supplier`'s requirements #4 and #5 respectively; specifically, the 'Treatment' components of the diagram depict the actual requirements.

To summarize therefore, for their requirements `Supplier` is taking the checklist and the set of diagrammatic models into the Negotiations phase.

In addition to the references mentioned above, [13, 32] also aid as a general resource in defining the security actions and requirements.

Case Output: Each company's functional requirements and security actions – the requirements above are examples of what is output and what they carry

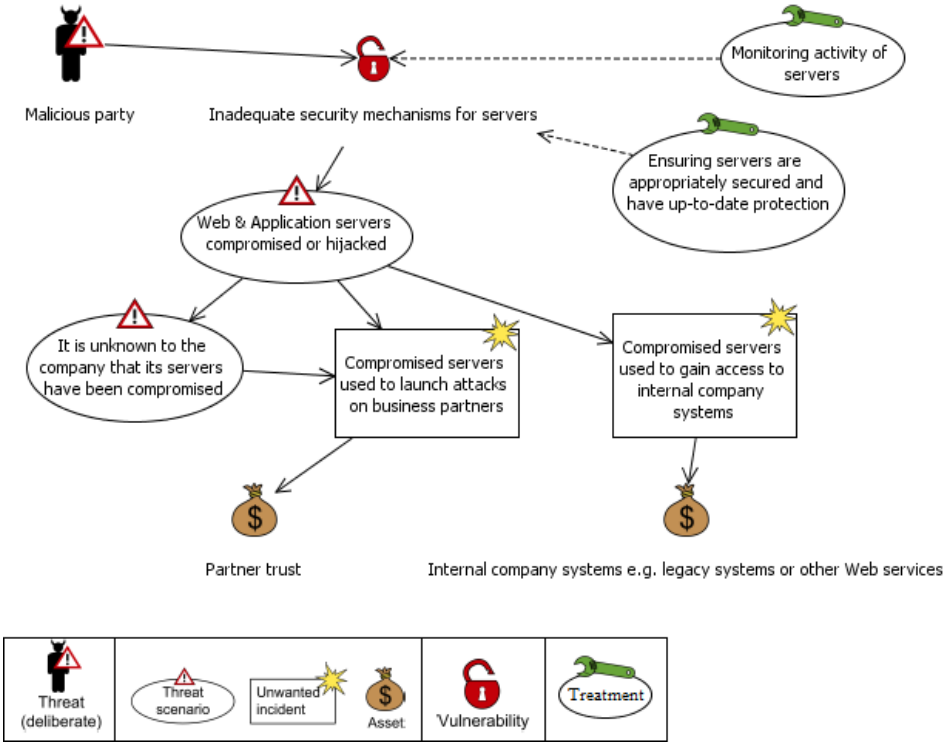


Figure 4.3: Requirement #4: `Supplier`'s risk treatment for servers

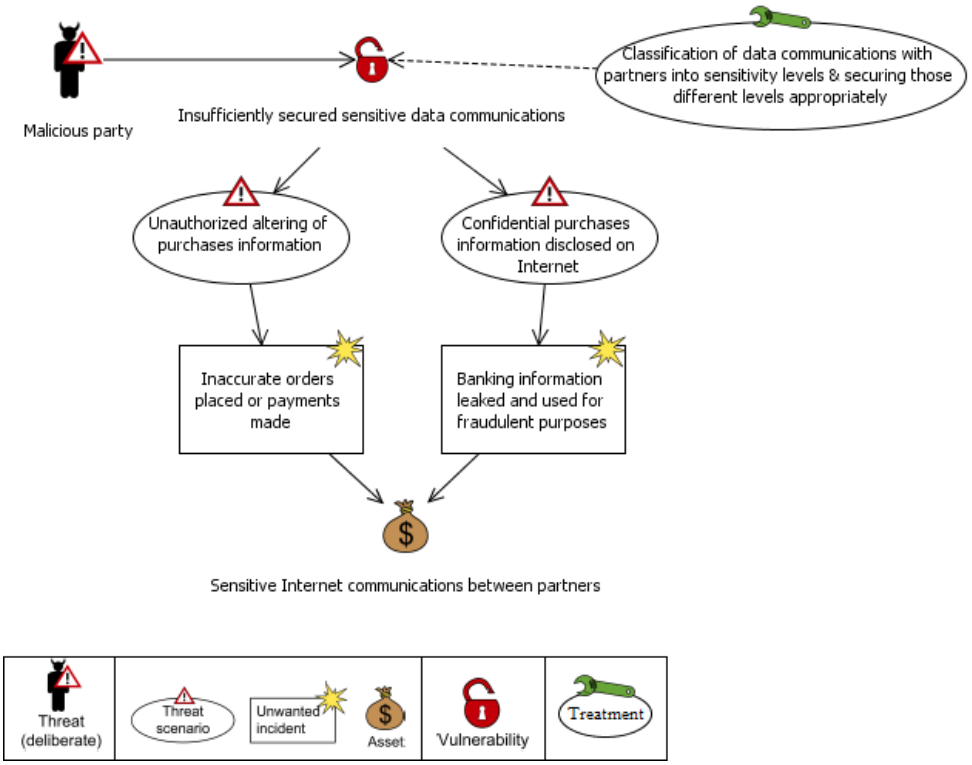


Figure 4.4: Requirement #5: `Supplier`'s risk treatment for sensitive communications

forward to subsequent phases. Also, the supporting documentation, for example process models.

4.3.2 Phase 2: Negotiations

Task 1: Discuss and negotiate on functional and quality requirements for systems. This starts with the requirements and process models brought by each company.

Case: This task is not shown because of the space that would be required, and especially considering that the focus is on security in this scenario. An assumption is made however that the general (UML) process flow and resulting functional requirements which are outlined by `Buyer` have been agreed for adoption by companies.

Task 2: Discuss and negotiate (or compare and reconcile) security actions and requirements for scenario interactions.

Case: Teams from `Buyer` and `Supplier` consisting of business and systems analysts and security professionals, meet to negotiate on their documented security actions and chart an agreed path forward.

The first task is the exchange of security documents/reports to allow each company to gain an understanding of the high-level security requirements of the other company. As requirements documents and formats across companies are so different however, this entails many queries to the other company's personnel as they attempt to understand: (i) what exactly the other company means by particular requirements, (ii) explanations of the proprietary checklists, requirements listings, and the graphical formats used to express requirements (these are unknown to business partners), and (iii) the motivation behind the need, that is, why a company has a particular security action/requirement.

Specific questions asked in line with these factors therefore include the following. Is there an underlying reason behind the security requirement? Is there a risk the company is trying to protect against in checklist items? If so,

why is it considered important to mitigate this risk as opposed to some other option? This process is conducted for each requirement.

With some understanding of the other company's actions and requirements, the second task was for both companies to look at comparing their individual actions/requirements. This process is conducted by security professionals and analysts from each company. The aim being to determine if `Supplier`'s and `Buyer`'s security actions for the scenario were compatible, and therefore could be carried forward to be applied to the pending business scenario. To ease comparison, the companies first choose to categorize each of their requirements by topic (for example, requirements regarding communications security were grouped). Table 4.5 displays the resulting groupings for nine requirements.

Category	**Buyer Sec. Req.**	**Supplier Sec. Req.**
Communications security	#1, 3	#5
Auditing/Logging	#2	–
Availability of services/data	#4	#1
Systems security	–	#4
Entity verification	–	#3
Malicious software security	–	#2

Table 4.5: Grouping of companies' security requirements

Next, each company's personnel use the other company's requirements document/reports to determine if that company has similar security requirements to their own. This is done on a requirement-by-requirement basis. If that company does not, then discussions take place questioning that fact. Typical questions include the following. Why has this type of requirement not been considered? Does the company not see the potential risk associated here? Are there other reasons/factors supporting why the underlying risk was chosen not to be handled?

Taking one requirement as an example of the narrative above, the following shows the supporting information given by `Buyer` for its security requirement #1 (listed previously). `Supplier` queries this requirement because it had no similar requirement. The supporting information by `Buyer` includes the following.

- The reality that there are unique risks associated with the use of WS-specific technology and therefore these require additional security considerations.
- The fact that the possible impact of threats on interactions would be very costly. Also, there is a medium risk likelihood as `Buyer` has been the victim of targeted attacks in the past.
- The Sarbanes-Oxley Act of 2002 (a United States federal law) requires that companies should be able to confirm that only authorized users have access to sensitive information and systems.
- `Buyer`'s security policy (labelled SPX2) strongly advocates the protection of the integrity and confidentiality of all potentially sensitive communications.

The final task is to identify a way forward and determine what requirements to keep and apply to the business scenario. Security actions (and their supporting information) from `Buyer` and `Supplier` are therefore discussed and compared at this point. In deciding security actions/requirements to keep, numerous factors play a part. Prime factors that aid how analysts and security professionals at companies decide on and reconcile security actions are: the actual factors that support requirements, for example, laws and regulations (legal compliance), limited IT security budgets, and calculated likelihoods and impacts of underlying threats and so on.

Expanding on the requirement example from above, the following snippet shows how these factors are used by companies' personnel to negotiate on and reconcile conflicting security actions of `Buyer` and `Supplier`:

- `Buyer` security requirement #1: The Web services messages which are used between companies to facilitate business transactions cannot be adequately secured by normal transport-level security. As a result, measures are to be taken to support security at the Web services message level, to ensure data integrity and confidentiality. Factors supporting `Buyer`'s requirement were given before therefore are not restated.

- `Supplier` did actually consider the underlying risk associated with message-level security during its Requirements Elicitation phase. Its personnel however choose to *accept* (and not *mitigate*, and thus have a security requirement for) this risk. This is primarily because of limited security funding and the unlikelihood that this risk would materialize as message-level attacks require above-average skill.
- **Solution**: After discussions on the main factors supporting requirements, the companies agree to adopt the security requirement. The decisive factors were: (i) the mandatory need to satisfy a law, that is, the Sarbanes-Oxley Act (previously `Supplier` did not consider that this law applied to interactions), and (ii) the increased threat likelihood and possible impact, both in terms of loss in production time and costs if measures were not put in place to protect communications at this WS level. Combined, these factors make stronger arguments for risk mitigation than risk acceptance. This therefore leads to the reconciliation choice above. To address `Supplier`'s limited budget constraint, options of shifting resources from other requirements or using free open source tools is put forward.

This provides one example of a reconciliation where one company's security action (in this case, a security requirement) was chosen. Other situations could result in: (i) combining similar requirements, for example, `Buyer` requirement #4 and `Supplier` requirement #1 are quite similar therefore either could easily be adopted or rephrased to the satisfaction of both companies, or (ii) dropping a company's security requirement if the reasons supporting it were not adequate, deemed surplus to security actions, or the other company had a stronger case for their treatment of the requirement's underlying risks.

Case Output: High-to-medium level scenario functional requirements and security actions and requirements. The agreed security actions and requirements identified above can be used as an example of what is output and to be carried forward to subsequent phases.

4.3.3 Phase 3: Agreements

Task 1: Legal contract drafting and signing by `Buyer` and `Supplier`.

Case: To formally specify the agreements made thus far, companies opt to define a legal contract at this point. The contract covers requirements and also include a non-disclosure agreement. The business analysts and legal teams from entities are useful here. Analysts work with lawyers to ensure that business agreements are adequately converted to legal documents. In Table 4.6 an excerpt of the contract is displayed. (Examples in [199] aided in the contract structure and definition.)

Joint Development and License Agreement

`Buyer` ("Buyer") and `Supplier` ("Supplier") enter into this Joint Development and License Agreement (the "Agreement") as of 29 March 2010 ("Effective Date").

RECITALS

WHEREAS, `Supplier` is a seller of high-performance electrical parts and `Buyer` is a high-tech equipments manufacturer. To enable their business processes to be more integrated, an IT system (the "System") spanning both enterprises is to be jointly developed.

NOW, THEREFORE, `Buyer` and `Supplier` agree as follows:

1. Scope of Services
Companies will perform the services and fulfil the agreed requirements described in Annex A (the "Work"), in order to develop the System according to time and specifications set forth therein. This Agreement is expected to mainly form a basis for interactions that will be built upon in a more detailed contract at a later date.

2. Terms and Termination
Unless terminated as provided herein, this Agreement will extend to and terminate upon completion of the Work as defined. Parties may terminate the Agreement without cause upon 60 days written notice. Additionally, either entity may terminate the Agreement for breach in terms provided that 20 days notice is given.

3. Non-disclosure
A. All information (especially about business processes, security risks and security needs) relating to `Buyer` and `Supplier` known to be confidential, proprietary, or which is labelled as such, will be held in confidence and not disclosed to others without the express written consent of the owning party.
B. All information relating to `Buyer` and `Supplier` known to be confidential, proprietary, or which is labelled as such, will only be used for the purposes of entering a foreseen business relationship with the other entity.

4. Cost Terms ...

5. System Ownership ...

Buyer	**Supplier**
Print name: _____	Print name: _____
Title: _____	Title: _____
Company seal:	Company seal:

Annex A

Specifications Document 1: General
Requirements Overview

- The developed System should enable streamlined interactions over the Internet between `Buyer` and `Supplier`. These interactions would support the `Buyer`'s procurement of high-performance electrical parts from `Supplier`.
- The necessary security features and capabilities should be in place to ensure accurate and trustworthy transactions, and fulfil regulatory, contractual and standards requirements.

...

Table 4.6: Joint Development and License Agreement

Task 2a: Restate mutual business scenario goals to provide vision for Interaction Security Strategy (ISS) definition.

Case: Companies state: `Supplier` is a seller of high-performance electrical parts and `Buyer` is a high-tech equipments manufacturer. The proposed agreement and business scenario would enable communications to be significantly streamlined across businesses. This alliance would lead to operational and production efficiencies and an increased competitive advantage for involved entities.

Task 2b: Define the cross-enterprise directives (strategies, policies and such) that form the core of the ISS. For this task, companies are to start with the agreed security actions/requirements from previous framework phases and any business rules or constraints facing companies (for example, costs or skills limitations). Entities should also consult the legal frameworks (company lawyers or specialists) and security standard best practices (for example, ISO 27000, NIST guides) for further guidance.

Case: To enable trustworthy interactions, `Buyer` and `Supplier` are applying a number of their agreed security requirements to produce the following security directives. Laws, regulatory requirements and security best practices also guide these directives. They are the following.

1. All information which is used and processed as a part of interactions between `Buyer` and `Supplier` is to be classified as either High, Medium or Low priority. These priority levels are to indicate the degree of importance of the information and therefore the grade of protection needed. Once all information has been classified, appropriate security controls appreciating these levels are to be implemented and maintained by all parties.

   The companies' decision to include this requirement is based on agreed security requirements and legal responsibilities focused around financial-related data protection, particularly the SOX Act.
2. Comprehensive logging is a crucial part of reliable service communications. This is especially related to the maintenance of an audit trail for transaction

monitoring, which would also facilitate/enable tracing and accountability checks as necessary.

In addition to requirements and auditing SOX Act requirements, this directive is strongly motivated by security best practices listed in [194], titled "Logging and Recording of Audit Trails".

3. The Web services messages to be used between companies to facilitate business transactions cannot be adequately secured by normal transport-level security. As a result, measures should be taken to support security at the Web services message level to ensure data integrity and confidentiality.
4. An initial mandatory step to all communications should be the identification of both parties.
5. A security team should be assembled to handle ongoing evaluation and updates to ISS directives, create programmes to instill the requisite level of training and awareness in personnel, and also handle the enforcement of directives. The team is to consist of members from both companies and will include business analysts, security professionals, and Web services architects and developers.

Case Output: Agreed functional requirements, envisioned processes (which have been modelled) and the cross-enterprise security directives exemplified above.

4.3.4 Phase 4: Analysis/Architectural

Task 1a: Identifying and agreeing on a process modelling technique.

Case: Business analysts at `Buyer` typically use UML to model business processes, whereas at `Supplier`, analysts prefer Data Flow Diagrams (DFDs) and a proprietary modelling format. At first, both companies are keen to use their own techniques but after deliberations and concerns about standardization they agree to use UML. This was primarily because, as the framework notes, UML is a standard technique and there are a number of simple and comprehensive extensions to account for security and QoS.

Task 1b: Business process definition and modelling.

Case: After discussing the processes, Figure 4.5 is jointly defined by analysts from `Buyer` and `Supplier` to model their cross-enterprise processes.

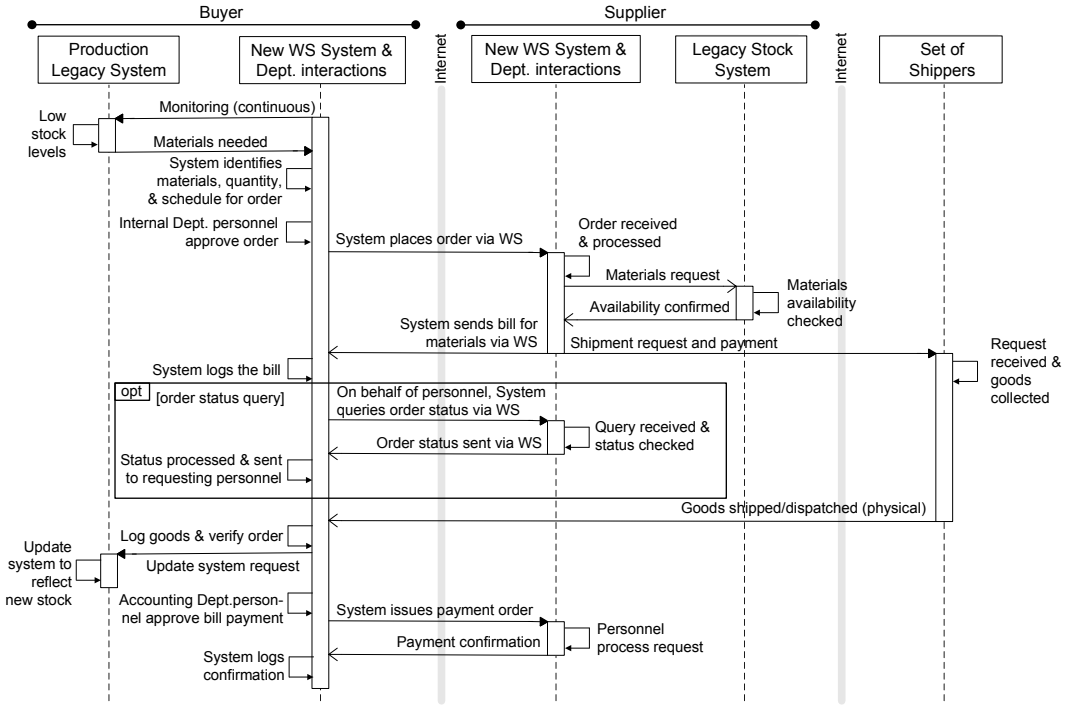


Figure 4.5: Full envisioned process flow

This figure shows key inputs, outputs, processes and requests as a part of the companies' interactions.

Task 2a: Identifying and agreeing on where to apply security directives to secure the process models.

Case: Smaller teams from `Buyer` and `Supplier` consisting of analysts and security professionals first map out the directives and consider how they relate to the models. From this exercise, they choose to identify three general processes and process aspects. These are shown in the left column of Table 4.7. The respective security directives agreed by companies' personnel is presented in the right column of the table.

Task 2b: Initial application of security directives to process models either using

Processes and their aspects	**ISS Security directives**
All process information, data, and messages transferred between and within companies. For example, purchase orders, material bills, orders status queries, payment orders	Classification of information (Directive #1); Web services message level security (Directive #3)
Sending and receipt of order-related messages and requests by systems and companies' personnel. For example, newly placed purchase orders, and bills sent to `Buyer` via WS. This also relates to internal company systems and their processing.	Identification and authentication of communicating parties (Directive #4); Maintenance of system-wide audit trails (Directive #2)
System-based or manual processing of order-related data and information. This includes data input, processing conducted, and information output. For example, receipt and processing of orders by `Supplier`, and logging of the materials bill by `Buyer`.	Maintenance of system-wide audit trails (Directive #2)

Table 4.7: Processes and respective security directives

generic security objectives or targeted security patterns. Recall that objectives are mainly related to stating the process aspects in terms of what needs confidentiality, authentication, integrity and so on. Whereas, patterns provide a well-proven, generic solution to a common problem.

Case: To enable application of directives, analysts and security professionals first deliberate on the advantages and drawbacks of the two possible application methods. Out of this discourse, `Buyer` and `Supplier` choose to use security patterns rather than security objectives. This decision is attributed to the standard, proven solutions to common problems they provide, and the availability of comprehensive pattern catalogues such as that highlighted by BOF4WSS in 'Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management' [194].

According to BOF4WSS, the next step is the identification of possible patterns to achieve security directives. These would then be passed into the subsequent Systems Design phase for formal analysis and actual application. As an example, the following paragraphs present information regarding the first two ISS directives.

To address **Directive #1 (Classification of information)**, the security patterns from the catalogue in 'Core Security Patterns: Best Practices and

Strategies for J2EE, Web Services, and Identity Management' ([194]) being considered are the 'Secure Pipe' pattern, 'Message Interceptor Gateway' pattern and 'Message Inspector' pattern. These patterns and how they address the problems and directives, are now presented in detail.

The 'Secure Pipe' pattern focuses on securing a basic connection between trading parties or systems (typically point-to-point, and over the Internet but can also be applied internally). It adds further value by giving the option of requiring mutual authentication and establishing confidentiality and/or non-repudiation between entities. The capabilities possible with application of this pattern would aid in ensuring that the appropriate security mirroring data classification levels are maintained as information is passed between companies and systems.
The 'Message Interceptor Gateway' pattern purports a centralized location or gateway to manage security enforcement tasks. For example, tasks include the application of transport and message level security mechanisms/standards necessary for securely communicating using WS. This pattern is responsible for adding the security that keeps Purchase order-related messages and data appropriately protected.
The 'Message Inspector' pattern works closely with the 'Message Interceptor Gateway' and handles the verification and validation of the security elements in the data or message delivered. This ensures that Purchase order-related messages received, which are headed for order processing systems, have not been tampered with.

To address **Directive #2 (Maintenance of system-wide audit trails)**, the security patterns from the catalogue in [194] being considered are the 'Secure Logger' pattern and 'Audit Interceptor' pattern. Descriptions of these patterns are given below.

The 'Secure Logger' pattern provides logging services for all security-related activities. It records all events and messages including the identities of senders,

requested operations and results/output returned. This pattern would support the need for system-wide audit trails of security-related activities specifically.
The 'Audit Interceptor' pattern supplies a centralized means for capturing and recording system audit events. It allows developers of `Buyer` and `Supplier` to define events of interest that are to be captured in an audit log for recording purposes or audit trails as required.

Case Output: Medium-level process models with security directives applied (or in terms of the patterns, initially chosen), the medium-level requirements (functional and security-specific) accompanying these models, and the inputs passed into this phase.

4.3.5 Phase 5: Agreements

Task 1: Legal contract drafting and signing by authorized representatives from `Buyer` and `Supplier`.

Case: Lawyers from `Buyer` and `Supplier` under the guidance of their analysts and security professionals, draft a new contract including the lower level requirements derived from the process definition in Figure 4.5. As the prime difference between this contract and the previous one (see Table 4.6) is the level of detail of the requirements in the Annex, only the new contract's Annex is displayed. This is shown in Table 4.8. Examples in [199] aid in the contract structure and definition.

Joint Development and License Agreement

`Buyer` ("Buyer") and `Supplier` ("Supplier") enter into this Joint Development and License Agreement (the "Agreement") as of 17 June 2010 ("Effective Date").

...

Annex A

Specifications Document 1: Detailed Functional Requirements

There should be a procurement system at `Buyer` capable of the following functionalities (inputs and outputs are clearly shown in Figure 4.5 therefore are not specifically identified again here):

- Continuously monitoring the stock levels for all production materials. The procurement system will therefore need to interface with the existing legacy stock system, this is a system which has up-to-date information on all the materials needed in production.
- Automatically generating an electronic purchase order depending on materials needed, including quantity and schedule of dates
- Passing the order for manual approval to personnel in the Internal Department (possibly using email)
- Subsequently enabling the electronic purchase order to be sent using Web services messages over the Internet to an order processing system at `Supplier`'s enterprise
- Logging bills sent by `Supplier` using Web services
- Contacting the system at `Supplier` to check the status of a specified order
- ...

There should be a order processing system at `Supplier` capable of the following functionalities (inputs and outputs are clearly shown in Figure 4.5 therefore are not specifically identified again here):

- Accept purchase orders via Web services messages at any time (availability is discussed later), and process them immediately
- Order processing at `Supplier` should involve checking the availability of items/materials in legacy stock systems
- Billing `Buyer` via Web services for specified materials orders
- Interfacing (requesting shipments and providing payments to and) with shipping companies to allow for items/materials to be delivered to `Buyer` as per time defined in the original schedule
- Responding to queries from external entities (for example, `Buyer`) regarding status of particular orders
- Logging and processing payments from `Buyer`
- ...

Information and Process Security

In addition to the functional requirements above, security features and capabilities should be in place to ensure accurate and trustworthy transactions, and fulfil regulatory, contractual and standards requirements. These security features are outlined within the directives and requirements in the Interaction Security Strategy (ISS). Parties should refer to that document for further details.

...

Table 4.8: Joint Development and License Agreement (Detailed)

Case Output: A detailed legal contract and the inputs passed into this phase.

4.3.6 Phase 6: Systems Design

Task 1a: Business process definition at the low-level and then service-level (that is, expressing choreography of interactions using WS standards).

Case: Using medium-level process models input into this stage, systems analysts at `Buyer` and `Supplier` define low-level interactions (and models) in terms of *services* (where a *service* is a distinct or packaged unit of logic or functionality). As an example of the process definitions resulting, Figure 4.6 is displayed. This covers the `Buyer`'s internal systems.

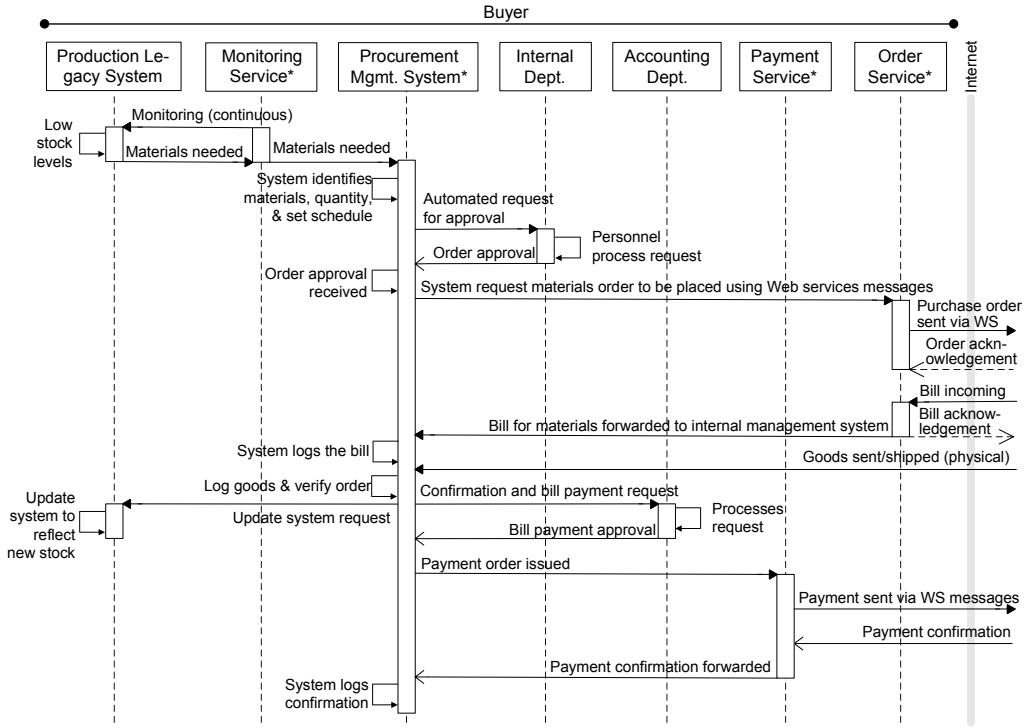


Figure 4.6: Full envisioned service-based process flow

The next step is business process definition at the lower service-level. Based on the options provided in the framework, teams start by considering the Web Services Choreography Description Language (WS-CDL) and BPEL4Chor. These technologies would be used to enable the message choreography to be defined and thus provide a global view of message interactions between companies. From

those options, company discussions lead to the preference of WS-CDL. This is due to its industry support and the existence of tested modelling tools. Parties select Pi4SOA [163] software (identified in the framework) to facilitate modelling and code generation. Apart from a usable GUI, business personnel viewed Pi4SOA a good choice as it allows for automated generation of Web Services Business Process Execution Language (WS-BPEL) code for internal system orchestration. In this scenario, Pi4SOA is used particularly because it is freely available. However, companies might opt for more polished software suites such as those from Oracle or IBM mentioned in BOF4WSS.

Applying Pi4SOA to the UML processes thus far, Figure 4.7 gives a screenshot of the tool's GUI and the graphical WS-CDL-based models which the companies create. This figure specifically focuses on the actual message interactions between entities, therefore the activities of placing an order, sending a bill and requesting shipment.

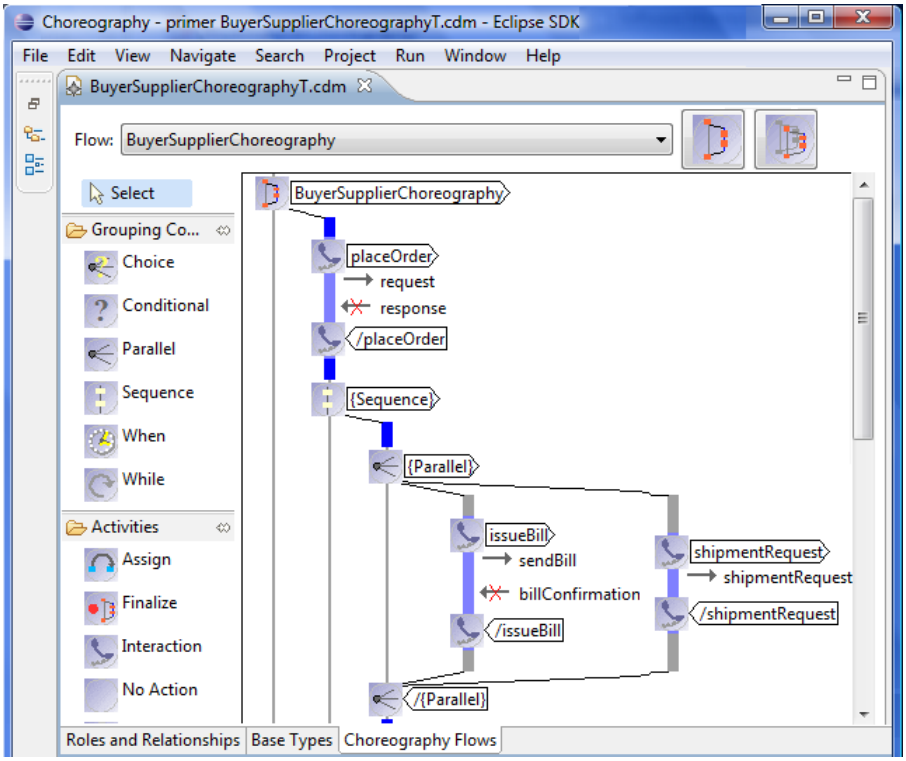


Figure 4.7: A screenshot of the processes defined in Pi4SOA

A sample of the WS-CDL code itself which is behind the graphical model

in Figure 4.7 is displayed in Code Snippet 1. This therefore accompanies the models above and serves as the service-level business process definition output by this framework task.

```
<?xml version="1.0" encoding="Cp1252"?>
<org.pi4soa.cdl:Package xmi:version="2.0" name="BuyerSupplierChoreographyT"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:org.pi4soa.cdl="http:///org/pi4soa/cdl.ecore" ... >
  <typeDefinitions>
    <informationTypes name="purchaseOrderType" typeName="PurchaseOrderMsg" />
    <informationTypes name="billType" typeName="BillMsg"/> ...
    <tokens name="purchaseOrderID" informationType="intType"/>
    <tokens name="billNum" informationType="intType"/> ...
    <roleTypes name="Buyer">
      <behaviors name="buyerForSupplier" interface="BuyerSupplierPT"/>
    </roleTypes> ...
    <relationshipTypes name="BuyerToSupplierRel" firstRoleType="Buyer"
          secondRoleType="Supplier" ... /> ...
    <channelTypes name="BuyerChannelType" ... >
      <identities tokens="purchaseOrderID"/>
    </channelTypes> ...
  </typeDefinitions>
  <choreographies name="BuyerSupplierChoreography" root="true">
    <variableDefinitions name="purchaseOrder" type="purchaseOrderType" />
    <variableDefinitions name="billNum" type="billType" /> ...
    <activities xsi:type="org.pi4soa.cdl:Interaction" name="placeOrder"
         operation="handlePurchaseOrder" ... >
      <exchangeDetails name="request" ... /> ...
    </activities>
    <activities xsi:type="org.pi4soa.cdl:Sequence">
      <activities xsi:type="org.pi4soa.cdl:Parallel">
        <activities xsi:type="org.pi4soa.cdl:Interaction" name="issueBill"
             operation="handleBill" ... >
          <exchangeDetails name="sendBill" sendVariable="billNum" ... /> ...
        </activities>
        <activities xsi:type="org.pi4soa.cdl:Interaction" name="shipmentRequest"
             operation="handleShipmentRequest" ... >
          ...
        </activities>
      </activities>
    </activities>
  </choreographies>
</org.pi4soa.cdl:Package>
```

Code Snippet 1: WS-CDL underlying the graphical model

Task 1b: Harmonization of process and data semantics across companies. In agreeing and outlining the service-level interactions, companies will also have to agree on the capabilities, description and syntax of the various data and services. Generally, this covers the semantics that will govern their interactions.

Case: To address this task and set up a semantics structure, project managers and analysts opt for the use of a shared vocabulary. This would act as a bridge between each company's metadata repository and allow for a direct mapping of data across companies' systems and software. This is not covered in more detail

noting the chapter's focus on security.

Task 2: Define quality requirements for businesses (low- and service-level).
Case: To identify the necessary quality requirements, personnel on the project teams use the companies' previous EDI quality requirements as a basis. This, along with the guidance article (that is, Garcia and Felgar de Toledo [66]) which is referenced in the framework aid in the creation of a list of quality requirements. Some of these are listed below.

- *Performance* (this concerns how fast a Web service is processed and the time it needs to complete the transaction): 30 milliseconds both ways (`Buyer` to `Supplier` and `Supplier` to `Buyer`) during normal loads, 40 milliseconds during high loads
- *Availability* (the probability that the Web service is up and in an immediate usable state): 99.99% uptime requested by `Buyer` of `Supplier`. `Supplier`'s availability request of Buyer is 99.97% uptime

Task 3a: Analyse trade-offs between adoption of security patterns and the functional and quality requirements for interactions.
Case: From an analysis of the security patterns and how their fulfilment would affect functional/quality requirements, company personnel identify a few concerning areas. For example, to apply all of the security patterns which are outlined (particularly the 'Secure Pipe', 'Secure Logger', 'Audit Interceptor'), achieving Performance QoS requirements above may not be possible. These patterns, although providing defence-in-depth, slow down interactions and increase processing time. Additionally, `Supplier` is mindful of the cost of the software and hardware to provide all these security functions. As a result of these factors, project teams decide to abandon mutual authentication at the transport level that is offered by the 'Secure Pipe' pattern, and to only apply the 'Secure Logger' pattern to the bill payment (`Buyer` to `Supplier`) transaction.

Task 3b: Application of viable security patterns to low-level process models.

Case: With viable security patterns which reflect ISS directives, system designers from businesses apply these to the low-level process models. The application of the 'Secure Pipe', 'Message Interceptor Gateway' and 'Message Inspector' patterns to the `Buyer` side of the 'bill issue' activity of the low-level systems models (displayed in Figure 4.6) is shown in Figure 4.8. This yields the security systems design. For continuity from Figure 4.6 and ease in presentation, only the systems used at `Buyer` are shown.

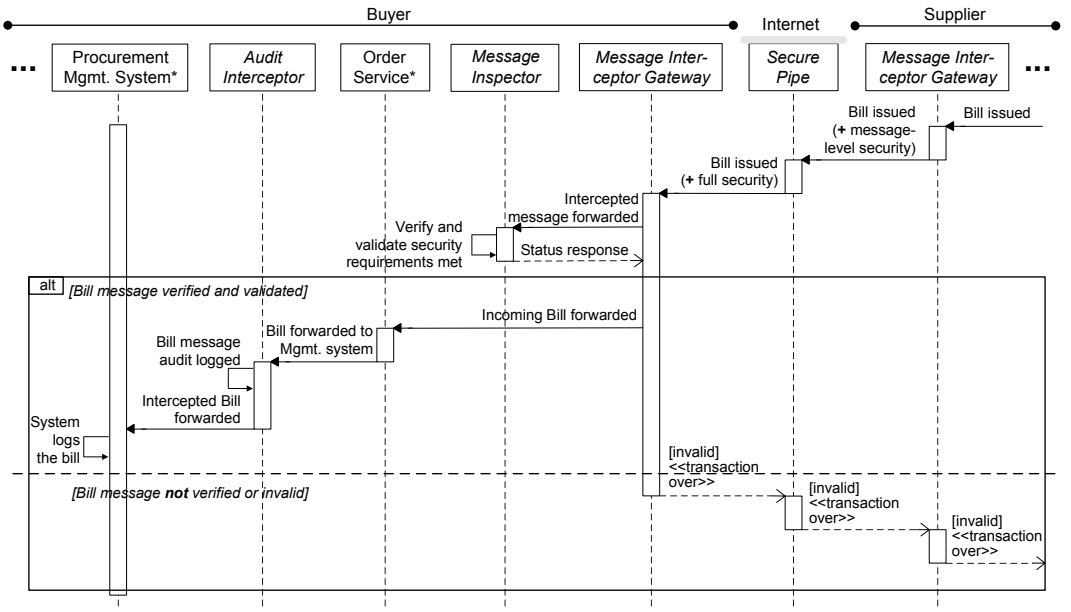


Figure 4.8: Applying security patterns to process model

Task 4: Identify and agree on the standards and technologies to implement interactions, especially security and quality requirements. Also, consider the best practices and vulnerabilities in the context of WS.

Case: According to BOF4WSS, analysts and designers begin by considering the options available which are listed on WS standards websites such as W3C and OASIS. The overview in innoQ's article [86] (also in the framework) is particularly useful in enlightening designers not familiar with detailed WS technologies. In addition to the general research, because both entities would need to purchase additional WS-specific business process software (application servers, orchestration engines and so on) to facilitate internal processing, standards implemented

by off-the-shelf products from IBM, Oracle and Microsoft are considered.

From these discussions and analyses, designers from companies agree to utilize the following standards to support functionality of services: SOAP (for messaging), WSDL (for service description and binding), WS-BPEL (for internal process orchestration), WS-CDL (for external process definitions) and WS-Addressing (aiding in the addressing of services). A majority of these are employed by a number of the commercial (such as Oracle SOA Suite $11g$) and open source tools (including ActiveBPEL Engine [1]) currently available.

To implement the security patterns and QoS requirements, analysts follow the framework's guidance and made use of the extensive pattern catalogue in 'Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management' [194]. The advantage of this catalogue is that it also supplies technologies that could be used for pattern execution. Furthermore, with an appreciation that the next framework step includes considering best practices and vulnerabilities, company analysts and designers factor in publications and practices from NIST ('Guide to Secure Web Services' in [189]) and WS-I [218] in choosing technologies. Table 4.9 provides an excerpt of the data in this catalogue which is adopted by businesses.

To implement QoS requirements for service interactions between entities, Table 4.10 displays the agreements made.

Pattern	**Standards and Technologies**	**Details**
Secure Pipe	HTTPS (SSL/TLS) (256, 512 and 1024 bit encryption for Low, Medium and High priority information classifications respectively)	These generally accepted standards are used to authenticate the recipient and establish point-to-point confidentiality and non-repudiation at the transport level.
Message Interceptor Gateway	WS-Security, XML Signature, XML Encryption, SAML tokens, X.509 certificates, WS-SecurityPolicy (Generally these standards are purported for use in services. WS-SecurityPolicy is key as it will be used to specify security requirements for individual services at this lower level, that will also implement and enforce higher-level directives such as varying levels of priority - and thus security - for messages. High priority messages for example will require services to digitally sign and encrypt the messages, include message expiry times, and nonce values (to protect against message replay). This can be specified in an individual service's policy.)	WS-Security specification provides a framework that integrates and unifies multiple security models and technologies (XML Signature, XML Encryption). Its prime goal is to define how to attach security information (such as SAML tokens or X.509 certificates) to provide end-to-end (or message-level) security for SOAP messages. The Gateway provides a central location to apply this specification on outgoing messages as is dictated in the service's security policy.
Message Inspector	WS-Security, XML Signature, XML Encryption, SAML, XKMS	The pattern checks for, verifies and validates the quality of the XML message-level security mechanisms such as XML Signature and XML Encryption in conjunction with a security token.

Table 4.9: Security technologies to implement patterns (adapted from [194])

QoS Requirement	**Methods and Technologies**
Performance (Response time as main metric)	Generally: The use of simple SOAP data types; compression of data from the sizable XML format to a binary format; and load balancing with servers hosting services. These are all suggested in an academic article [230]. Monitoring could also be done by management and reporting software that would automatically check services for compliance.
Reliability and Availability	WS-Reliability specification can be used as it ensures guarantees delivery of messages, elimination and/or detection of duplicate messages, and right order delivery of messages
Availability of service (Focus on protection and recovery from attacks that disrupt service from being in a usable state)	Generally: Web service clustering, that is, making the same service available over multiple servers. Specifically: An XML (message-level) Firewall could be deployed at a company's perimeter to inspect and filter messages for malicious content before any harmful messages reach the service.

Table 4.10: QoS methods and technologies

Case Output: Low-level process designs, service-level interaction definitions, security design, a semantics framework supporting processes, the list of standards/technologies to implement WS interactions, and low-level requirements resulting from processes (functional, security and quality-based).

4.3.7 Phase 7: Agreements (for QoS)

Task 1: Formal agreement and specification of the QoS requirements of each company for services.

Case: Using the quality requirements from the previous task, companies involve their analysts, lawyers and executives to formally specify the agreements. After assessing the options available (including WSLA and WS-Policy additions) which are mentioned in the framework, a natural language SLA is defined. Designers prefer this choice as they are not confident in the reliability, use, or maturity of these WS standards. An excerpt of the SLA which the entities create is presented in Table 4.11.

Service-Level Agreement
`Buyer` ("Buyer") and `Supplier` ("Supplier") enter into this Joint Development and License Agreement (the "Agreement") as of 3 August 2010 ("Effective Date"). **RECITALS** WHEREAS, `Supplier` is a seller of high-performance electrical parts and `Buyer` is a high-tech equipments manufacturer. To enable their business processes to be more integrated, an IT system (the "System") spanning both enterprises is to be jointly developed. NOW, THEREFORE, `Buyer` and `Supplier` as it pertains to quality requirements agree as follows: 1. Performance Performance is defined as how fast a Web service can be processed and the time needed to complete the transaction. The requirement is: 30 milliseconds both ways (`Buyer` to `Supplier`, and `Supplier` to `Buyer`) during normal loads, and 40 milliseconds during high loads 2. Availability Availability is defined as the probability that the Web service is up and in an immediate usable state. The requirement is 99.88% uptime requested by `Buyer` of `Supplier`. `Supplier`'s availability requests of `Buyer` are at 99.7% uptime – Exceptions: Objectives may not be met due to the following: scheduled maintenance (for example, hardware or software upgrades); network issues not in direct control of service provider; ... – Obligations: In cases of expected downtime, businesses must notify and have the support of other partners at least 48 hours prior. In cases of unexpected downtime, businesses must notify other partners immediately. – Other: Evaluation of objectives should be carried out on a monthly basis to ensure quality requirements are being met and maintained. **...**

Table 4.11: Service-Level Agreement for `Buyer` and `Supplier`

Case Output: The SLA QoS agreements, low-level process designs, service-level interaction definitions, security design, a semantics framework, the standards and technologies of choice to implement the WS interactions, and the updated low-level requirements (functional, security and quality-based).

4.3.8 Phase 8 and 9: Development and Testing, and Maintenance

Thus far in this chapter, BOF4WSS's phases have been applied to the outlined scenario to give a clear example of the framework's uses. These phases were somewhat theoretically driven (in terms of ability to document) and highly structured,

and therefore lent themselves to good presentation and discussion. Phases 8 and 9 however are not similar to these for two main reasons. These are, firstly the lack of a clear structure—phase 8 particularly provides only very general phases and encourages companies to plug in methods they deem as appropriate, thus being extremely context dependent. Secondly, with the actual systems already defined, it is primarily acting on these definitions that is required. The second point therefore involves hands-on programming, testing, system configurations, monitoring services, making updates and such. These are not viewed as activities that would be appropriate or able to be adequately covered in this chapter's discussions.

As a result of the two reasons mentioned above and the reality that approaches/methods suggested for application by BOF4WSS (WS lifecycle methodology and the Process for Web Services Security(PWSSec)) are well documented (and even have their own case study analyses available), it was decided not to cover these in detail in this thesis. Instead, this work seeks to provide an idea of how and where the outputs from previous phases would fit in and be consumed by methods. First therefore, the WS lifecycle methodology is presented in Figure 4.9. Secondly, in Figure 4.10, the integration of documentation from BOF4WSS into the parallel security process PWSSec.

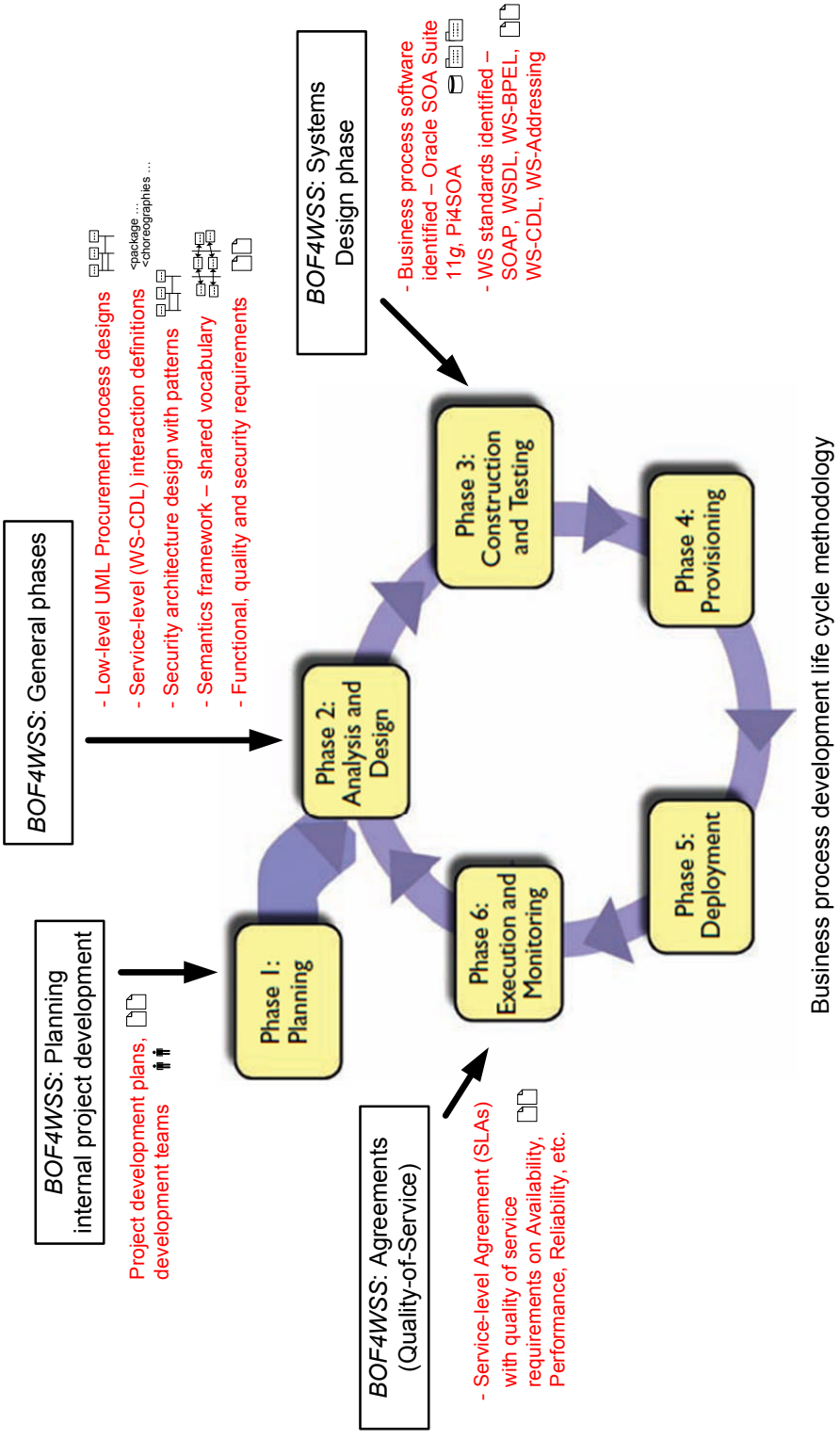


Figure 4.9: The use of framework output in the WS lifecycle methodology

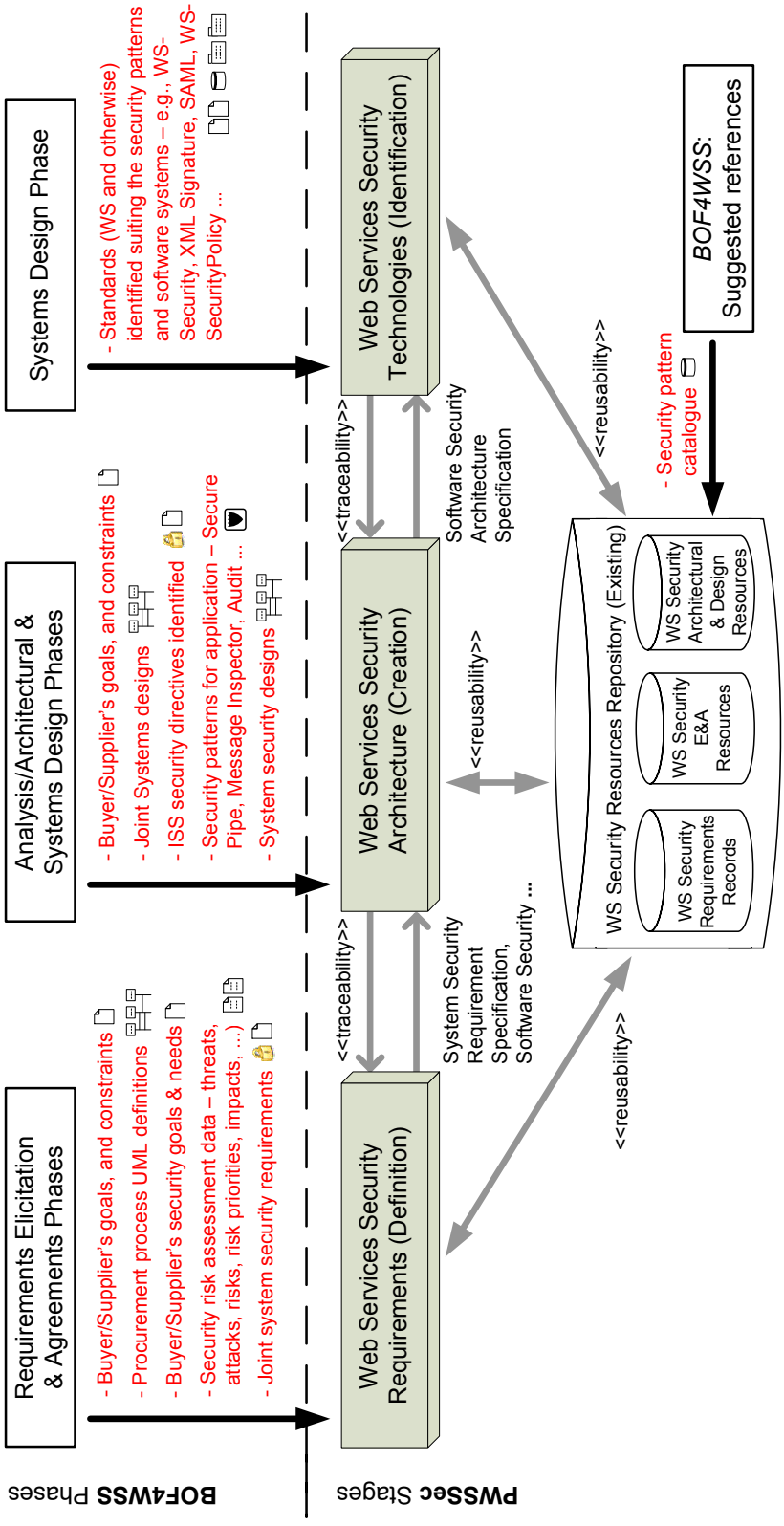


Figure 4.10: The use of framework output in PWSSec

The Maintenance phase within BOF4WSS is somewhat similar to the Execution and Monitoring phase in the WS lifecycle methodology. `Buyer` and `Supplier` would therefore be monitoring their interactions and making updates to services and functionalities as required. From a security perspective, the aim becomes maintaining the security of the developed systems and ISS directives by constantly being mindful of new threats and attacks. 'Audit Interceptor' and 'Secure Logger' security patterns would also help in checking for and identifying any breaches in systems.

According to the framework, `Buyer` and `Supplier` should also monitor new computer and data protection legislation as these would affect the ISS directives and security requirements defined above. Lastly, in terms of ongoing support personnel for functional modifications and security issues, persons from the project teams would be identified by companies to fulfil these tasks.

4.4 Discussing the Framework's Use

Concentrating on the scenario presented in the sections above, the uses and advantages of BOF4WSS can primarily be found in its adequate fulfilment of the companies' expectations. In line with company needs, the framework is comprehensive, in that it covers the full spectrum of activities from planning to development, while also remaining relatively flexible. This was exemplified in the fact that even though Risk Management (RM) approaches were suggested to determine security requirements, companies were allowed to use their own methods, found in the CORAS and NIST RM Guides. Furthermore, the framework satisfied `Buyer` and `Supplier`'s desire for a WS-specific and security-focused methodology.

From a security specific perspective, the uses of the framework included attaining the agreed and adequate security infrastructure—BOF4WSS and its constant focus on negotiations and agreements at every level helped this. Secondly, in terms of a security governance structure, the framework's ISS acts as a less rigid but clearly defined charter for companies, which guides their joint

security posture. This strategy naturally appreciates security as more than a technical issue and is supported on an ongoing basis by a security team from companies.

Thirdly, the framework aids in achieving the traceability requirement desired by `Buyer` and `Supplier`. This is apparent as BOF4WSS encourages suitable model documentation early on and continuously builds on initially agreed security requirements to achieve final designs and security technologies. A good example is the security requirement regarding confidentiality of WS messages. This was first conceived as a result of a RM process. Then, appropriate security patterns ('Message Interceptor Gateway' and 'Message Inspector') to apply to the models were defined. Lastly, WS security technologies such as WS-Security and XML Signature were identified for the technical implementation.

The final expectation of BOF4WSS placed by businesses was trust across the newly formed scenario. Even though the framework is not a panacea in this regard, it would be expected that from the constant interactions process, `Buyer` and `Supplier` would have established some trust. This could be trust in each other (business level), trust between project teams that worked together, or trust in the scenario interactions that they jointly planned, designed and developed.

The points highlighted in the paragraphs above provide a few examples of how BOF4WSS can be useful and advantageous to `Buyer` and `Supplier` and their particular scenario. Considering the nature of those companies' expectations however, it would also be fair to state that many of those expectations would be relevant in a wide variety of situations. This is particularly in terms of flexibility in process, WS-specific security methodology, security guidance/governance structure and cross-enterprise trust. This concludes this section.

4.5 Summary

The purpose of this chapter was to supplement Chapter 3's discussion of the framework and supply real-world examples of BOF4WSS' use. A key underlying

aim was to support the practicality of the framework, and to highlight some of its usages and benefits. To do this, a scenario was presented and BOF4WSS applied to it phase-by-phase. Options were discussed as companies would face them and choices in terms of the framework were made. At the end of the scenario's presentation, Section 4.4 provided a brief discussion on the framework's use and advantages in terms of the case, and in general.

The next chapter focuses this project on a more manageable research problem within the framework's investigation. As will be seen, the specific problem of interest relates to the likely transitional issues arising when companies move between two framework phases. Chapter 5 forms a bridge between the higher level framework approach in this and the previous chapters, and the lower level, more detailed discourse in future chapters.

Chapter 5

Supporting BOF4WSS and the Transition Between its Phases

Extended networks, for many organizations, represent an enormous challenge with regard to security controls. Different partners have unique access requirements, want specific security policies in place, and have varying SLAs [Service Level Agreements] and legal obligations, all leading to security mayhem. — James S. Tiller

5.1 Introduction

Having presented the framework and discussed its practical application to a scenario, this chapter narrows the scope to the identification of a specific and more manageable research problem for in-depth analysis. This problem would still be in the confines of the framework. Both these aims are in line with the original research objectives (specifically the third objective) in Section 1.2.

To aid in the problem identification process, the scenario from Chapter 4 was used and analysed in detail. The aim was to identify any processes or tasks that could prove difficult or overly labourious for companies as they attempted to use BOF4WSS to support their interactions. From this analysis, one area that proved intriguingly problematic and also had the potential to be a significant barrier for companies, was the inherent difficulty that surfaced when companies tried

to share, compare and negotiate on the high-level security actions/requirements. This related to when these security needs were passed from the *individually* undertaken **Requirements Elicitation** phase into the subsequent *joint* **Negotiations** phase (where companies reconciled their actions on security) in BOF4WSS.

The core problem during the phase transition process mentioned above was the disparate nature of the varying security actions and requirements supplied by each business. This problem was perpetuated by the need for companies to understand each other's actions and the motivations behind them, be able to discuss actions thoroughly, and then arrive at appropriate agreements and judgements. Focusing on the disparity specifically, dissimilarities in approaches were apparent at the semantic and practical levels, and even in the format of supplied security actions. For example, Chapter 4 shows that companies might use graphical means, checklists, or bullet points to express requirements. This makes even simple comparison of these actions across companies much more painstaking.

With appreciation of the importance of a smooth transition through the framework's phases (particularly in terms of BOF4WSS' adoption by companies) and the intricacies of this problem at a research level (as will be detailed further), it was chosen for this project's focus.

The next section of this chapter examines the transition problem in detail, drawing upon literature and a critical study of the scenario in Chapter 4 to identify the core issues. Once the problem area has been presented, specific investigative questions are defined. A solution model is then proposed in Section 5.3 which sets the platform for all subsequent work. The solution model and its components (conceptual and physical) complement BOF4WSS in forming the main contributions of this thesis. Before proceeding, readers are reminded that a *security action* is a generic treatment method for a risk, whereas a *security requirement* is a treatment method geared towards risk mitigation. Security actions therefore encompass security requirements.

5.2 The Transition Problem

5.2.1 Related Literature

Sharing, comparing and negotiating on security actions and requirements across companies, even for security at a high-level, has always been a complex issue. In Section 2.3.1, Tiller's work ([205]) gave insight into this issue as he labeled the related process "security mayhem" because of the variety of aspects of security to be considered in forming business collaborations. This area has also been defined as a core research question by Dynes et al. [49] in terms of how interacting companies can achieve a shared vision on risks and security which appreciates their range of differences.

In the somewhat similar field of outsourcing, a few approaches have been seen which attempt to tackle the 'coming together' problem (this problem is akin to the transition problem identified previously in BOF4WSS). Kajava et al. [98] for example offer generic guidance in the form of preconditions to any outsourcing agreement. These include, "Both partners' information security capabilities must be at a high level prior to cooperation" and "The agreement negotiations are an essential process and, if need be, the parties must utilize the services of external specialists to ensure success".

Work by Todd et al. [206] documents a real-world case involving two large companies, BT and HP, and exemplifies another approach. To help decide some of the security actions necessary, first each company's respective security standards, policies and procedures were compared and contrasted (it is not mentioned, but it is assumed this was done in a manual fashion). Next, a principle-based approach was utilized where the highest security level from whichever company was adopted. Joint security risk assessments were also used.

The two approaches identified above are valid methods, however they supply only very high-level guidance on negotiating security actions and requirements across companies. Furthermore, some of this guidance is not likely to be widely applicable. In situations where there are tight security budgets or limited exper-

tise in companies, for example, approaches such as the "adopt the highest security level"-based approach in [206], are infeasible. Reflecting on the general problem defined in the first paragraph therefore, this research has found no appropriate approach in literature or practice which might be applied to aid in the transition from the Requirements Elicitation to Negotiations phases.

5.2.2 Problem Area

To solve the transition problem particularly in the context of BOF4WSS, it was first necessary to identify the core difficulties faced by companies in this process. For support in this task, the scenario from Chapter 4 was analysed critically from this perspective. The aim of this section is to present the main problem areas discovered from that assessment. These areas are also somewhat generic, in that they would apply to all scenarios in which BOF4WSS is used.

Predominant emphasis on security requirements

As companies meet up for negotiations, it is common to bring together the security requirements for the envisaged scenario. This very standard task however has a shortcoming. By supplying only their security requirements, companies actually only show information on the risks that they would like to mitigate (recall that security requirements link to risk mitigation, as highlighted in Chapter 3). Risks that companies choose to accept, avoid or transfer are usually given no mention as exemplified in the case study by Steel et al. [194] and in Chapter 4's scenario. This incomplete information leads to a problem during comparison because in situations where there are conflicts in how risks are to be treated, companies have to re-enter discussions. This is as opposed to merely referencing a treatments document (which would list complete information on suggested treatments for all risks) as currently done for security requirements. The prolonged, repetitive discussions required are therefore the core issue here.

For companies that are going to be working closely together and need to agree on ways to treat (that is, choosing either mitigation, acceptance, avoidance

or transference) certain *shared risks*, this research strongly asserts that entities must initially acknowledge more than just one risk treatment option (risk mitigation) by way of security requirements. This acknowledgment does not require any new tasks or techniques, but simply involves having companies also supply non-mitigation-based treatments (and information on all pertinent shared risks they have assessed) into the Negotiations stage. This would at the very least start companies with a more complete information base and reduce the repetition in negotiations and discussions.

Understanding the security actions documents/reports of the other companies "as is"

There are three sub-problems in this area, two relate to semantics and one to security actions format. At the beginning of the Negotiations phase, companies supply their high-level security actions (in Chapter 4, all were security requirements) to their business partners for review. A major difficulty even at this early stage is gaining an appreciation of what exactly companies mean when they outline a security action or requirement in a few brief, informal statements, often with little justification. This is the first semantics problem.

In the scenario in Chapter 4 for example, `Supplier` notes "Authentication is critical and should be used". This statement is so abstract that it has little meaning to `Buyer`, which would cause them to ask: "Authentication of, or between whom?", "Is this authentication at the level of business parties or otherwise?", "Is authentication required for use all the time, or only in specific scenarios?". At times these requirements may be relatively obvious, but at other times, numerous queries are likely to result which can lengthen this discussion process considerably.

The second semantics issue relates to the ambiguities in the security field and the plethora of terms, concepts and meanings which exist. As companies meet therefore, they are likely to be using a range of disparate terminologies which then adds to the possible misunderstandings and makes negotiations even

more difficult. The lack of any common terminology or common understanding of terms and concepts is a barrier to interactions.

The last sub-problem is centred around the variety of unique formats that could be used by businesses to document their security actions/requirements. This point is exemplified in Chapter 4 as `Buyer` uses a requirements listing whereas `Supplier` uses their company's standardized security checklists along with graphical models and tools. From this simple example with only two entities, three different formats have been applied, all of which need to be explained and discussed. This is particularly pertinent if the formats are proprietary or other partners are simply unfamiliar with them. The core issue here therefore is linked to the disparity in formats and subsequent need for all companies to understand these formats prior to any negotiations taking place.

Understanding the motivation behind other companies' security actions and requirements

From the few brief statements which constitute companies' security actions and requirements, it is often somewhat challenging for other businesses to determine why that security desire exists. Even if the security threat/risk which the requirement is intended to address is included in the requirement description, there may be various other considerations in the preceding risk assessment that are not specified in the requirement statement provided. These factors are important because they provide insight into security actions that form the basis for companies' negotiations and decisions.

The scenario in the previous chapter depicts the point mentioned above clearly as it looks at the reasons supporting `Buyer`'s security requirement #1. The way this supporting information was elicited was to begin discussions once again, and for each company to query the other to gain a better appreciation of the requirements and their supporting factors. This process however is quite tedious and may require assessing a number of the requirements again. Additionally, depending on the number of the companies involved (the scenario has only two,

however more are possible) and the amount of requirements to be discussed, this can be very time consuming. In summary, the main problem thus results from a lack of information on motivational and supporting factors provided initially.

Categorization of security requirements

As companies come together there is likely to be a wide range of security requirements spanning a variety of topics. There is little guarantee that the security requirements (and the threats/risks they intend to address) conceived by one company have not been overlooked or forgotten by another company. Categorization tries to group requirements into similar topic areas with the hope of easing the comparison task faced by personnel.

The challenges with this step however are that (i) once more, it is very time consuming (companies have to identify and agree on suitable categories, assign requirements, etc.) and (ii) it may have little added value since it is highly dependent on requirements listings and their innate similarities, and also the granularity of the categories chosen. For example, in the scenario, only two of the chosen categories ('Communications security' and 'Availability of services/data') have similar requirements across companies. Therefore the majority of categories have requirements from one company only; this represents four out of nine requirements in total. Both of the aspects mentioned act to somewhat defeat the purpose of categorizing requirements.

Comparison of companies' security actions and requirements

This task entails studying other companies' requirements to identify (and question) similarities and differences. Initial requirements categorization would have made this task easier, however it is far from seamless. In the 'Communications security' category for example, there is no comparable requirement from `Supplier` to suit `Buyer`'s security requirement #1. Companies would therefore need to discuss this requirement further. Once it is clarified that this requirement represents a unique security desire, `Supplier` queries `Buyer`'s motivation and factors that

support this security action.

Fortunately, some of the motivational factors may have been identified before (in one of the first tasks). However, as companies look to support and promote their individual security actions, more concrete and well-justified aspects will need to surface (relevant laws for example, or further details on pertinent organizational policies). In summary, the main difficulties therefore emerge from the huge tasks of studying requirements, finding similar requirements, and querying other entities once again for adequate justifications. This is especially when requirements represent conflicting needs across companies.

Deciding on security actions to apply to the scenario

Having discussed security actions, defined the aspects supporting each company's desires and compared the actions and requirements across companies, the next task is to reconcile the security actions to carry forward. ('Security actions' are held to be carried forward as opposed to 'security requirements' because there is no guarantee that companies will always agree to mitigate a shared risk.)

In this last task and particularly in situations of *conflicting* security actions (therefore, where there are different action types, for example one company wants to *mitigate* a risk whereas another wants to *accept* it), the factors supporting each entity's security actions played a crucial role. The scenario exemplifies this point as two factors that supported `Buyer`'s security requirement #1 (that is, the Sarbanes-Oxley Act of 2002 and the increased threat likelihood and possible impact to businesses if the underlying risk/threat materialized) were crucial to making the final decision. The challenges at this decision stage therefore are again reviewing factors supporting *each* action and then arriving at a conclusion on the way forward with each security decision.

5.2.3 Research Question

In an attempt to address most of the problems identified above and support the transition between phases, the following research question was posed. This

question is directly linked to the fourth research objective in Section 1.2.

– As companies bring their various security actions together for negotiations, a tedious, repetitive and long-winded task often ensues. Can this process be streamlined or significantly automated to allow for an easier and quicker transition between the framework's Requirements Elicitation and Negotiations phases?

To aid in answering this research question four guidance steps have been defined.

Step 1: A first step towards supporting the negotiation of security actions across companies is the establishment of some common and shared security-related semantics. This would act as a bridge between the internals of companies (for example, terminologies and structures) and the internals of their business partners. Section 5.2.2 previously identified a semantics gap in terms of security actions and terminologies. However, there is also the possibility of semantic misunderstandings when considering the factors motivating/supporting security actions and exactly where and how they fit into businesses' decisions.

In light of these aspects, it would be beneficial to have a common semantics structure shared by companies which encompasses security actions, their motivational factors and other relevant aspects in that domain. To avoid introducing any additional semantic issues, it is imperative that this new structure is easy to understand and reference. Furthermore, it should allow a range of heterogeneous company terminologies and structures to be mapped to and captured by it.

Can a common semantics structure (such as a conceptual model, ontology or meta-model) be defined, therefore, to fulfil the aims mentioned above? Furthermore, how seamless would it be to map some of the internals of companies (for example, the outputs of typical security action determination methods) to the shared semantics structure? At the very least, this could act as a test for the adequacy and use of the model.

Step 2: A semantics structure shared between companies in the context of their interactions is an important initial step to easing the discussion and comparison of security actions at the high level. Beyond using this structure to tackle pure semantic issues however, are there any additional ways in which it could be used to address other problem areas, or in generally expediting parts of the transition between stages? For example, could this structure be used as a basis for the creation of a common format or template for companies to express their security actions and related factors as they enter negotiations? If this were possible, it would (i) help in resolving the disparity in the formats problem area and (ii) provide a guide for companies on what information they should supply as they prepare to come together. Point (ii) thus seeks to address the problems associated with incomplete information in Section 5.2.2.

Lastly, assuming that a common template or format could be created, could it also have a machine-processable representation? This would open the approach to opportunities for automation. Automation for example might be seen in expressing companies' security actions or in comparing them across businesses.

Step 3: The security actions and requirements brought by companies are likely to span a wide range of topics, and address a variety of risks and threats to parties and the overall business scenario. As exemplified in Section 5.2.2, this reality typically leads to a number of difficulties in action categorization, matching and comparison. These difficulties mainly centre around the tedious and repetitive tasks which need to be undertaken by companies' personnel.

With appreciation of the difficulties mentioned, are there any ways to streamline these arduous processes? More specifically, how could this matching and comparison problem be addressed such that it lends itself to some noteworthy degree of automation? For example, is there a way that some risk or security action listing could be used as a *common*, shared base for companies? If this could be achieved, a system might be investigated that would automatically categorize, match and allow for the personnel-led comparison of security actions from

businesses.

Step 4: Within the goal of easing phase transition, there must be an appreciation that any information generated, especially that from Step 3, should be very usable by humans. This is primarily to aid in making decisions on security actions to apply to the scenario. Amongst other things therefore, personnel from companies should be able to (i) see detailed information on risks and security actions being compared, (ii) possibly tweak comparisons, for example, requesting that only differences in security action types across companies be highlighted, and (iii) have inconsistencies that represent exceptional situations flagged for follow-up. All of this information should be supplied in a user-friendly interface which significantly supports users in their decision making activities. Step 4's guidance questions therefore are, how can this be done and how best can information be presented to users?

5.3 A Solution Model

To answer the research question and generally support the progression between phases and subsequent negotiation of security actions, the Solution Model in Figure 5.1 was defined. The model contains four components, namely, Security Actions Analysis, Ontology Design, Risk Catalogue Creation and Language Definition. There is also a system that will be implemented to consolidate these components. In the following paragraphs an overview of the components is given. This also forms an introduction that puts work in future chapters into context. Chapter 6 presents a detailed discussion of the first two components and Chapter 7 covers the remaining two and the overall system implementation.

Security Actions Analysis

Step 1 from the previous section explores the notion of a common semantics structure to define aspects in the security action domain. A security action (or

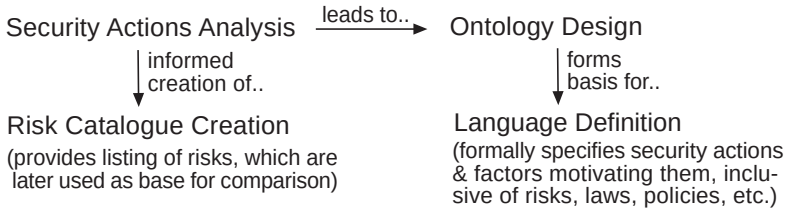


Figure 5.1: Solution Model

risk treatment choice) is just the end result of a very detailed and often non-trivial risk-based process. The aim of this component therefore would be to critically examine security in general, but especially how security actions come to be, and identify common, critical factors that motivate/influence their derivation. This will include aspects such as threats and vulnerabilities, but also higher level factors including security policies and company budgets. This component relies heavily on security literature to allow for a firmly grounded analysis. The Ontology Design stage complements this analysis and presents an ontology as the semantics structure to address Step 1.

Ontology Design

Ontologies are widely known for their ability to specify a shared understanding about a domain across persons and systems [55]. In this case, an ontology is employed to provide a conceptual relational model and supporting semantics documentation for the security actions domain (specifically resulting from findings in the Security Actions Analysis stage). These characteristics fulfil the requirements for an easy to understand and reference structure for companies, (as the ontology is at a high level and diagrammatically presented), showing relationships between concepts, and including documentation. Companies are therefore free to refer to the ontology at any time for a clear understanding of the common terminology to be used throughout interactions. Readers should note that at this point the ontology is not formally expressed and therefore there is no corresponding formal ontology language representation. This will be covered in a subsequent component.

To avoid limiting the usability of the ontology, general concepts in the security field are also modelled. Establishing this single understanding and semantics structure is a critical prerequisite in creating the overall solution when considering how different the terminologies, methods and influencing factors internal to each business are likely to be. It is also important that the ontology is accommodating and allows for an easy mapping of concepts onto it from typical security action determination methods used by companies. Providing this, in addition to a structure well-grounded in the security actions domain (that considers relevant standards, RM/RA approaches and so on), will strongly support the use and applicability of the ontology and any derived models/tools.

Risk Catalogue Creation

To address research Step 3, a shared risks base has been chosen. This choice was made after reviewing the Security Actions Analysis and noticing that in a majority of cases, security actions were established to handle or treat some inherent risk. The creation of a risks catalogue which contains an up-to-date, extensive listing of threats and vulnerabilities (or generally the security risks) serves one main purpose therefore — it forms a *common* input to each company's RM/RA processes (that is, the process that identifies, analyses, evaluates and decides treatment for the risks). Thus, even if businesses use different processes, they will maintain a *common* input in terms of what risks are considered.

Furthermore, the risk catalogue would be updated to accommodate new risks found by companies in their actual RM/RA methods. Using a shared risks base means that regardless of the derived security actions, the underlying security risk can always be used to automatically match security actions across companies. The difficulty with categorizing, matching and comparing actions therefore becomes much less arduous.

Language Definition

This component stage targets research Step 2 and therefore seeks to use the shared semantics structure (represented in the ontology model) to define a common template/format for expressing companies' security actions. This format should also act as a guide for companies on the information they should furnish as they prepare to come together. Considering these facts and the request for a machine-processable template/format, one way to address this step is to specify a common formal format (for example, based on XML or Web Ontology Language (OWL)) to express actions, and then to encapsulate it in a user-friendly interface such as a data entry screen or template document. This interface would be the information guide for companies. The underlying formal format which actually specifies the security actions and related factors would be the common basis across parties.

To create this common format which represents a more formal implementation of the ontology, an XML-based language and schema were preferred. XML is a useful format as it is a very mature, widely recognized, platform independent markup language which has numerous applications and systems support options (APIs for parsing, verifying and validating XML are readily available). The increasing popularity of OWL in the field of ontologies also led to it being considered for this task. After some investigation however, it was concluded that OWL offered much more functionality (layers of logic and reasoning) and a much higher degree of expressivity than was required at this point. Additionally, these advantages came at the price of added complexity. Euzenat [53] provides a good example of ontology ordering in terms of degrees of expressivity and formality. In that work, XML schemas are one category away from very formal ontologies (such as those represented in OWL).

The aim of the language and schema in this report therefore, is to allow data on security actions and factors influencing them (such as risks, laws and policies) to be defined in a highly structured way that could be exchanged and compared across companies.

5.4 Solution Model in Action

5.4.1 Process Overview

A general idea of how the *implemented* Solution Model would work towards easing and possibly automating stage transition is illustrated in Figure 5.2 (readers are asked to assume that `Supplier` and `Buyer` are businesses using BOF4WSS to facilitate an envisioned online business scenario). As the diagram is somewhat self-explanatory and the general implementation discussed in detail in future sections (such as Chapter 7), it is not examined at this point.

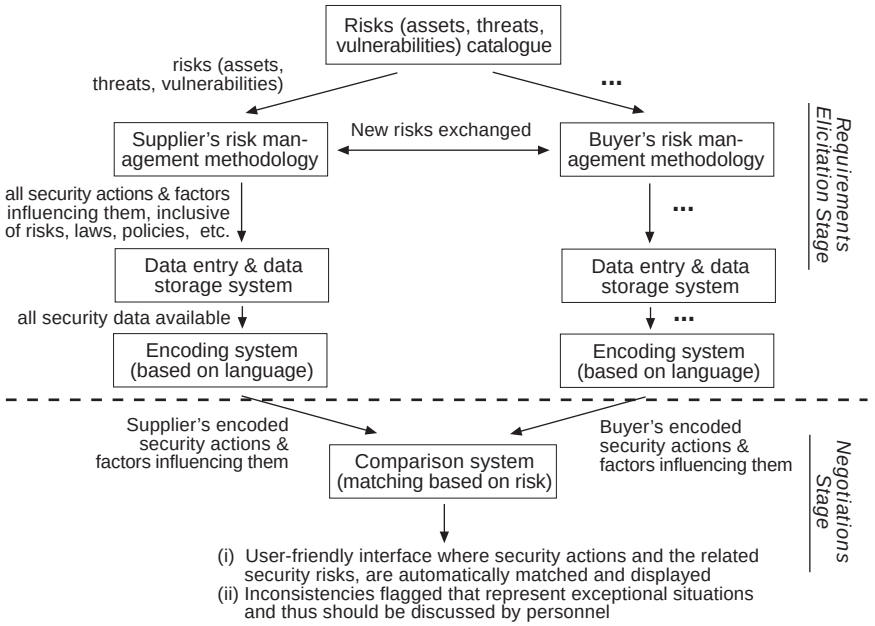


Figure 5.2: Solution Model in action

Another prime objective of this implementation is addressing the remaining research step, that is, Step 4 from Section 5.2.3. Referencing Figure 5.2, the output of human-readable information in points (i) and (ii) from the Comparison system are seen to address the guidance questions posed in Step 4. In detail, matching of different companies' security actions is done based on the risks which they address, whereas tweaking comparisons can be done by changing system query settings. Specifics of the user-friendly interface encapsulating the output are forthcoming in Chapter 7.

It is worth noting that to accommodate the Solution Model and the resulting implemented system, a slight change is needed in the BOF4WSS' process flow. This change includes the addition of a formal task in the Requirements Elicitation phase which focuses on inputting (automatically or manually) security actions, risks and motivational factors into the system created. All of these aspects are then transferred to the Negotiations phase. This is unlike the framework previously where only security requirements were typically carried forward. The Negotiations phase works similarly to before. The only change is that companies use the Model and implemented system to streamline a variety of initial tasks during the negotiation and discussions on security. Companies can for example centre in on problem areas and quickly get to the core discussions necessary for stage progression.

5.4.2 Related Models

Before concluding this chapter, related work by Yau and Chen [227] deserves mention. In that article the authors assessed similar disparity problems in communicating security requirements and proposed a framework for formally specifying requirements and detecting conflicts. Their framework also utilized an ontology for providing a common base to facilitate the unambiguous specification of requirements amongst collaborating parties. The differences between that research and the Solution Model are apparent in the Model's focus on high-level security needs, and the fact that [227] only considers security requirements as opposed to other ways to address risks (such as acceptance or avoidance).

5.5 Summary

This chapter engaged in a detailed assessment of the framework and its application to a real-world scenario. From this assessment, the transition problem from the Requirements Elicitation to Negotiations phase of BOF4WSS was identified as a key area for research emphasis.

Once the problems were clearly defined, a research question and a number of guidance steps were derived to chart a research path forward. A solution model was then proposed to address these problems and significantly support phase transition. This model consisted of four components: Security Actions Analysis, Ontology Design, Risk Catalogue Creation and Language Definition. Beyond discussing the conceptual Solution Model, insight was also given into its operation and implementation. The chapter concluded with a brief look at related work.

In the next chapter, discussion on the Solution Model components is expanded. Emphasis is placed particularly on the Security Actions Analysis and Ontology Design, and reporting on those stages, their outputs and contributions.

Chapter 6

An Ontology for Defining Factors Influencing Security Actions

The growing popularity of ontologies is in a large part due to what they promise: a shared understanding of some domain that can be communicated between people and application systems. — Dieter Fensel

6.1 Introduction

In Chapter 5, the Solution Model was defined to address the phase transition problems identified within practical use of BOF4WSS. The aim of Chapter 6 is to expand on that Model and report on the implementation of some of its core steps, namely Security Actions Analysis and Ontology Design. This chapter will conduct a critical analysis of what security actions are and the methods companies use to determine them. This analysis allows for the identification of several important factors that influence companies' individual security actions, and therefore adds to the disparity between businesses outlined in the previous chapter. These factors and their relationships are then modelled in a high-level ontology which, along with the preceding analysis, constitute the main contributions of this chapter.

It is worth noting that the main steps in the general process consist of considering ontology goals, defining core terms/factors in the domain (for this

work, this is the security actions domain) and identifying relationships amongst other things. This process is presented by King and Reinold [104] as a core methodology for the creation of an ontology.

6.2 Security Action Determination Methods

6.2.1 Defining Security Actions and Requirements

Security actions and security requirements were first introduced in Chapter 3 and since then have formed important terms and topics in this research. This section seeks to complement previous discussions by recapping and expanding on them. In addition to presenting these terms from more of a literature-based perspective, this discourse sets the context for the following sections in the chapter.

Security actions and requirements have occupied discussion for many years and in countless fields spanning both practice in industry and research focus in academia [58, 67, 76]. Whereas a security requirement is a well-known term—albeit not always clearly defined (see Haley et al. [76]), a security action, as a term, is a new proposal in this work and is aimed at covering a wider topic range than a requirement (as shown in Section 3.3).

To avoid confusion in the context of this research, a *security requirement* is defined as a high-to-medium level desire, largely in terms of the information security goals of confidentiality, integrity, availability and accountability. Requirements are therefore usually expressed to protect against risk associated with a general process or action, as opposed to specific systems or technical-level constraints. In this context, this approach to requirements is similar to that alluded to in [174, 201].

A *security action* refers to any way (that is, setting up protective measures or not) in which a company treats or handles a risk it faces—the use of the analogous term of risk treatment choice/action in prior chapters highlights this. The main difference between the action and requirement concepts is that a security action does not always represent a positive action to eliminate a risk or reduce its

severity level, whereas a security requirement does. The types of security actions are covered in detail in forthcoming sections.

6.2.2 Risk-related Methodologies

Regardless of the intended purpose, a security action or requirement represents the culmination of a detailed and often non-trivial process, which considers a number of aspects (including risk assessments, statutory and contractual requirements, and business principles and objectives) as highlighted by Jones and Ashenden [94]. To examine this process from a general perspective, this section considers the work of Firesmith [59].

In [59], as Firesmith researches the topic of engineering security requirements, he outlines some of the most basic and crucial precursors to defining requirements, and thus generally, security actions. These factors, as inferred in Figure 6.1, include the identification of assets and services to be protected, analysis of their innate vulnerabilities and also a review of the security threats which endanger the assets. Only after this process has been carried out can appropriate and ultimately useful requirements be ascertained.

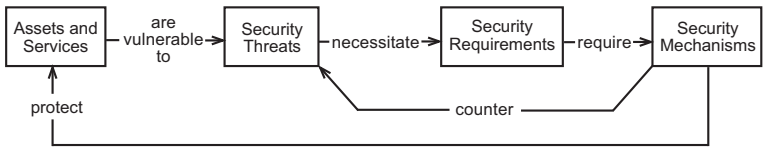


Figure 6.1: Relation of requirements to assets, threats and mechanisms [59]

To consider the process in Figure 6.1 formally and in a comprehensive organizational context, this series of tasks constitute a process analogous to that of Risk Management (RM). RM here, and as defined in its most basic form by ISO [90], represents the full complement of coordinated activities to direct and control an organization with regard to the risks it faces.

Reflecting on the comparison made above, the activities to determine risks (in RM) are similar to the 'identification of assets and services, and the threats to which they are vulnerable to' (the first two boxes in Figure 6.1). Whereas

the activities to control an organization's risks (in RM) are held to match to the 'definition of security requirements, or generally security actions, and respective mechanisms' (the last two boxes in Figure 6.1). The tight association of risks with assets, threats and vulnerabilities is not surprising, and can be seen in various articles such as [67, 52, 106]. For a more detailed definition of RM which is within an IT-specific context, readers are referred to [195].

To aid in the complex process of managing organizational security risks, an abundance of methodologies has been offered and researched. These proposals target not only RM in its entirety, but also inclusive tasks such as risk analysis which is defined as, "systematic use of information to identify sources and to estimate the risk" [90] (p.5). Following on from this definition, key intentions of the analysis therefore include the identification and assessment of risks, plus their subsequent valuation [67, 196, 215].

Returning to the discussion on RM, RM methodologies include the NIST SP 800-30: Risk Management Guide [195] and OCTAVE [24, 3]. Focusing on component stages within RM, CORAS [47], ISRAM [99] and the IS risk analysis based on a business model [196] (hereafter, ISBBM) provide good examples for risk assessment and analysis processes. Combined, these form the main reference methodologies for this chapter's work. Their effectiveness, popularity and extensive documentation are the main driving factors behind selection.

The wide variety of methodologies available to manage, assess and treat risks is undoubtedly of great use and benefit to businesses. By having a number of choices, organizations can adopt one or a combination of methods that best addresses their needs and also fits their respective company structure and culture. This is with full appreciation that different methods will have varying uses, strengths and weaknesses, and may stress a dissimilar set of aspects. To consider OCTAVE for example, its creators clearly identify it as being targeted at organizational risk and focused on strategic, practice-related issues; this is as opposed to a technology-focused assessment mechanism targeted at technological risk [3]. Conversely, in the Risk Management Guide provided by NIST, the ultimate goal

is stated as aiding organizations to better manage IT-related mission risks [195]. This is one example of two methodologies targeting different levels of risk.

A similar scenario is apparent in the comparison of risk analysis methodologies when assessing the way risks are valued and prioritized. Vorster and Labuschange [215] identify this as they examine and compare a number of risk analysis methodologies. Possibly the best example of the difference in focus can be seen in the formulae of risk valuation used by the CORAS and ISRAM processes. Vorster and Labuschange concluded that CORAS prefers simplicity and thus provides a simple 'impact and probability' approach to determine loss, whereas ISRAM employs a complicated, all-inclusive formula to value risk, thereby stressing accuracy of valuation over simplicity.

The focal point of the previous two paragraphs is that with different methodologies, different factors will be stressed, different tools used, and ultimately a wide range of security actions and requirements will be determined. These actions, even at this high level, are likely to be very dissimilar across companies.

6.3 Security Actions Analysis

6.3.1 Justification of Approach Chosen

Having provided an overview of the process in which security actions are determined, this section aims to narrow that scope and investigate the primary factors which influence action derivation. To conduct this investigation, two of the most intuitive approaches are methodology-by-methodology or RM stage-by-RM stage. The first option involves individually examining a range of methodologies and then identifying the pertinent factors. The second approach differs as it decomposes RM into smaller stages, and then identifies the factors applied by each methodology at that respective stage.

For the investigation, the latter of these options was preferred. The reason for this was twofold. Firstly, using general RM stages allows for methodologies

to be concisely compared and contrasted. Secondly, this method provides an excellent opportunity to define some major processes that constitute RM such as risk assessment, risk analysis and risk evaluation. These processes, albeit very mature in practice, are at times still equivocally defined; for example see [90, 106, 195] in terms of risk analysis versus risk assessment, and [106, 196, 215] as compared to [67, 161] on the core objectives of a risk analysis.

As previously defined in ISO [90], RM is the umbrella term for activities to direct and control risk. Within this process ISO has identified a number of activities. A subset of the most critical of these is presented in Figure 6.2.

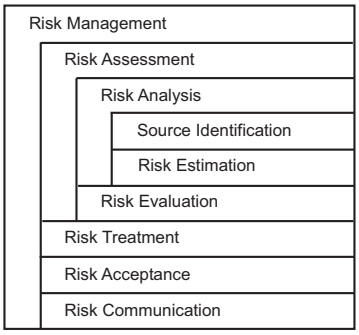


Figure 6.2: Relationship between RM processes (adapted from [90])

From the illustration in Figure 6.2, one can easily grasp the relationships between these processes and put each into context with regards to the overarching RM task. The focus of the following section's analysis is on the lower level, well-defined, core processes of source identification (which we consider analogous to risk identification, a view seemingly shared by Munteanu [129]), risk estimation, risk evaluation and risk treatment. For ease of reference, the factors that play a pivotal role are italicized.

6.3.2 Factors Influencing Security Action Determination

Risk Identification is defined as the "process to find, list and characterize elements of risk" [90] (p.5). In simple terms this process identifies the risk to the organization, system and so on. Within the methodologies assessed, it was found that the general consensus (as shared by CORAS, OCTAVE, ISBBM and

Landoll [106]) as to the factors which help ascertain risks is based strongly on *assets*, their *vulnerabilities* and the *threats* to them; this finding confirms the prior analysis in Section 6.2.2.

In other methods such as the NIST Guide, the concept of risk identification is not as definitive. In assessing their definition of risk and their Risk Determination stage however, it is clear that they heavily associate the notion of risk not only with assets, vulnerabilities and threats, but also with the *likelihood of threat occurrence* and *potential of impact* (if the threat occurs). This definition suggests another way (in terms of valuation) that risks can be initially viewed. Hogganvik and Stølen [83] give an example of other work which holds a similar perspective of risk.

Risk Estimation follows, and is the "process used to assign values to the probability and consequences of a risk" [90] (p.6). This stage, also known as risk valuation, is often regarded [196, 161, 205] as one of the most critical in a risk analysis. As implied from the ISO definition, the *probability* or *likelihood* of a risk occurring and the *consequence* or *impact* if it materializes, are two critical factors to estimating the value or level of a risk. This perspective is supported by the fact that ISRAM, OCTAVE, ISBBM, CORAS and published texts [106, 161] all employ these factors, even if only as the basis for more complicated estimations.

Beyond probability and consequence, Kairab [97] introduces a third factor which focuses on the importance of assessing the *effectiveness of existing controls* (a control is a risk-reducing measure) in reducing a risk to an acceptable level. This factor acknowledges that the adequacy of a control directly affects the possibility that a risk will materialize, and thus should have a value which is considered in the final risk value estimated.

Thus far, all the methodologies assessed aim to be high-level and therefore either focusing on general organizational or IT risks. As the lower-level of WS is important to this work however, assessing Web application risk approaches is necessary for a comprehensive analysis. To facilitate this evaluation and identify more influential factors within risk estimation, the DREAD model [118] of risk

estimation was examined. Apart from its suitability at the Web application layer, this model was selected because it was supported by a market-leading company, that is, the Microsoft Corporation.

The DREAD model uses the *Damage potential* (how great is the damage if the vulnerability is exploited), *Reproducibility* (how easy is it to reproduce the attack), *Exploitability* (how easy is it to launch an attack), *Affected users* (roughly, how many users are affected) and *Discoverability* (how easy is it to find the vulnerability) factors to assess and ascertain a qualitative rating for risks. As compared to the previous methodologies, DREAD, albeit targeted at a lower level, is not drastically different. The 'Damage potential' and 'affected users' aspects for example are simply specific categorizations of an impact, whereas the three other factors can influence the probability of a risk occurring. For example, if it is easy for an attacker to discover a vulnerability (the Discoverability factor) then the probability of it occurring will be higher.

Risk Evaluation's purpose is to compare the estimated risks against a given risk criterion to determine the significance of the risk [90]. As stated by ISO [90], risk criteria will normally include *associated costs and benefits*, *legal and statutory requirements*, *socio-economic aspects*, *concerns of stakeholders* and the *determined risk priorities*. With regards to the chosen methodologies, only CORAS treats risk evaluation as a separate stage. However, even in CORAS' case, it only encourages finalizing the risk values and therefore does not state any specific influencing factors/criteria. In OCTAVE, evaluation is a part of risk analysis, whereas ISRAM and ISBBM exclude risk evaluation—probably because they are strictly risk analysis methods. Lastly, the NIST Guide moves from Risk estimation directly to assessing control recommendations to reduce risk levels.

Risk Treatment is the final stage to be examined and entails a "process of selection and implementation of measures to modify risk ... treatment measures can include avoiding, optimizing, transferring or retaining risk" [90] (p.6). Having estimated and evaluated the risks, the next task is deciding how they are to be treated/addressed (a salient fact being that not all risk might be mitigated).

In making this decision, the NIST Guide stresses the consideration of various factors. Some of the most distinct are *legislation and regulations* (do any laws affect the decision), *organizational policy* (are any business policies to be assessed) and *cost-benefit analysis of recommended controls* (what is the cost—maybe financial, operational, feasibility—of treatment compared to benefit or effectiveness of control implementation).

To look more closely at the cost-benefit analysis and what is involved, Houmb and Georg [84] provide an excellent example of how some of the aspects mentioned above influence the determination of the treatments. Key factors identified are *policies*, *regulations*, *risk priorities* and *treatment effects and costs*. Apart from OCTAVE which emphasizes the *influence of stakeholders*, none of the other methodologies specially focus on new risk treatment factors.

The next section uses the factors highlighted above to create a representative ontology design.

6.4 Ontology Design

6.4.1 Ontology Use

An ontology can be defined as a set of assertions meant to model some particular domain [53]. These assertions can be very formal (see Fensel [55]) and thus utilize special languages such as Web Ontology Language (OWL). However, they can also be considerably high-level and therefore use conceptualizations such as diagrams and object-oriented models [53, 164]. Regardless of the level conceived, a prime use of an ontology is in providing a shared understanding of a domain that can be communicated between people and application systems [55]. This fact is the main motivation for applying an ontology in this research.

Specifically, the ontology is used to produce a high-level relational model of the most pertinent factors from the RM process which influence security actions and requirements derivation. This model would act as a common reference point to communicate the relationships of factors between companies and their

personnel. Furthermore, it would be used as the basis for a tool to help in security actions comparison and negotiation. In the following section, the related work is reviewed before presenting the ontology design.

6.4.2 Related Ontologies and Models

There have been numerous articles presented which research into and use ontologies to convey ideas and knowledge in their respective domains. The most noteworthy and relevant of these articles are reviewed below with the aim of drawing attention to their individual use and identifying some key concepts within their models. As before, concepts identified are italicized. It is worth noting that in all of these articles, a diagrammatic ontology design is included, even though not always presented (due to space considerations) in this review.

As identified previously, an ontology is a prime candidate for use in the field of knowledge representation, sharing and management. Fenz and Ekelhart [56] exemplify this fact as they develop and present a security ontology to formally model information security domain knowledge. Additionally, their model is intended to include concepts and relations used by common information security risk management methodologies, thereby extending its scope. From the high-level ontology model developed and shown in Figure 6.3, some of the main constituent concepts are *organization*, *asset*, *threat*, *security attribute*, *vulnerability*, *control* and *standard control*.

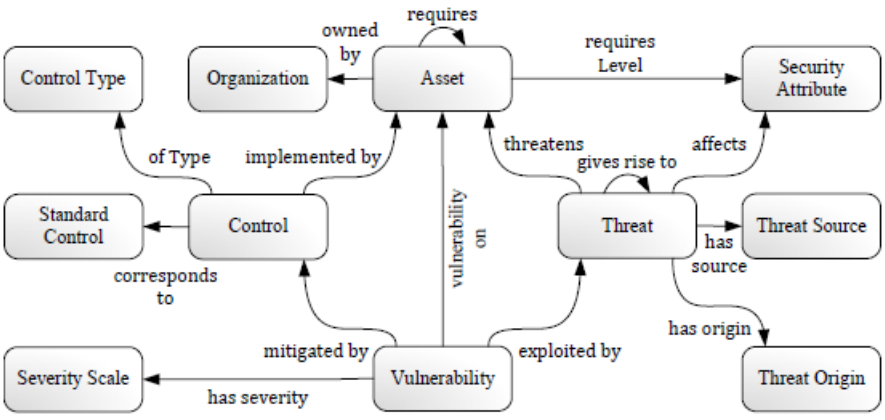


Figure 6.3: Security relationships concepts in [56]

Tsoumas and Gritzalis [209] also employ an ontology as a knowledge resource for the foundation of their work. They investigate the topic of security management and provide a framework for reusable security knowledge interoperability, aggregation and reasoning. Their goal is to exploit security knowledge from a range of useful and diverse resource sets. The ultimate aim is to provide a structured approach to help security experts to consider these various information sources and better transition from high level statements from risk analysis documents to deployable technical security controls. Their framework uses a Risk Analysis Security Ontology as a basis for its development and some of its most pertinent concepts include *assets*, *vulnerabilities*, *impact*, *security policy*, *controls* and *countermeasures*.

Within the CORAS method previously assessed, several supporting models have been defined which give special emphasis to aspects such as risks and security requirements. Two of these models can be found in [72] and [46]. Unlike the other articles above which are geared towards formal ontologies, the aim of these models is to use a visual depiction to convey meaning and to illustrate relations between important risk assessment concepts. The aspects prevalent in these models include *risks*, *risk value*, *assets*, *asset values*, *security requirements*, *security policy*, *unwanted incidents*, *likelihood* and *treatment*.

The next model for review is seen in Firesmith's article [60]. In that research, Firesmith hypothesizes about utilizing reusable, parameterized templates for specifying security requirements. One of his first tasks in that process is the clear definition of a conceptual model including factors which influence security requirements and the relationships between them. The resulting model is one of the most comprehensive and complements its design by describing terms and concepts used. Some of these terms are *security*, *security risk*, *security goal*, *security requirement*, *security policy*, *security mechanism*, *harm*, *attack*, *people* and *property*.

Mayer et al. [116] define another intriguing ontology targeted specifically towards the information system security risk management domain. This is de-

picted in Figure 6.4. The purpose behind this ontology is its further use as a meta-model for the security risk management modelling language which the authors intend to define. The main relevance with this meta-model is its strong base in RM literature to define and relate the concepts. Concepts in their domain model include *risk*, *event*, *impact*, *vulnerability*, *attack method*, *asset*, *security criterion*, *risk treatment*, *security requirement* and *control*.

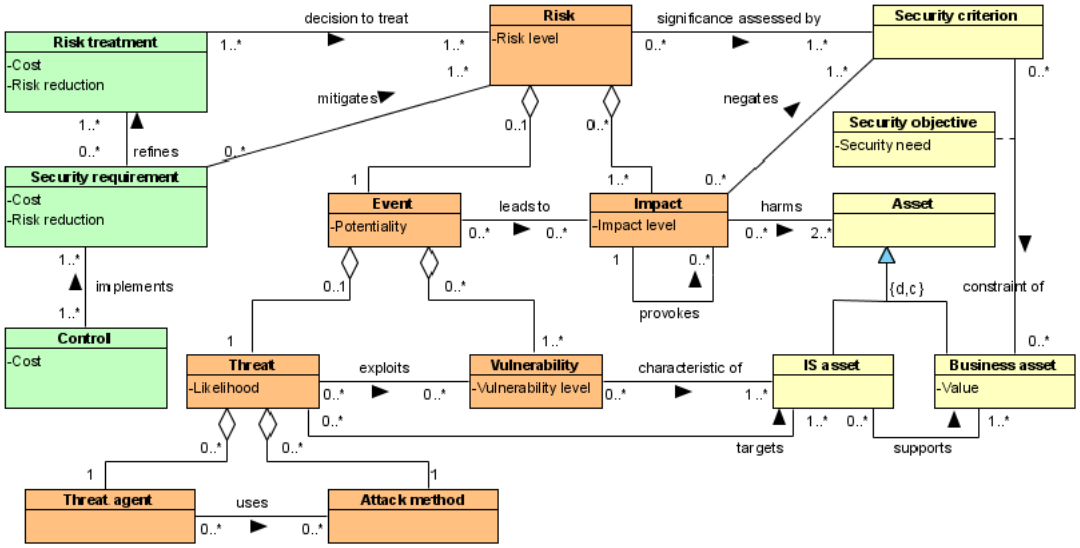


Figure 6.4: IS Security Risk Management meta-model ([116])

The last ontology of interest is that introduced by Houmb and Georg [84]. The primary aim of that model is to illustrate the key concepts involved in their risk assessment framework. Once identified, these concepts are then used to specify and compute the Return of Security Investment for the various treatment strategies available. This ontology and its concepts are similar to those previously discussed, but its most salient factor is the explicit inclusion of *risk level* and constituent elements such as *frequency* and *impact*.

The works discussed above provide excellent examples of how ontologies and models can be employed for various purposes within security and risk-related processes. Considering the suitability of these conceptualizations to the identification of factors influencing security actions determination (one of the core purposes of this chapter) however, none of them alone offers an adequate foundation. In [209, 60] for example, the ontologies seem to be reasonably comprehensive (in

that they span from risk identification to risk treatment), however neither appreciates the importance of a risk level/priority concept. A similar case is apparent with Fenz and Ekelhart [56], as they also do not include the concept of a risk or risk level/priority. Only at a lower ontological level do they begin to consider ratings for the effectiveness of control implementations and probabilities of threats occurring.

In addition to the CORAS models, the proposals in [84, 116] were the most relevant models. This is likely due to the grounding of the models in RM methodologies. Mayer et al. [116] especially adopt a comparable approach to that used in this chapter to define their domain model. One core problem with some of these conceptualizations however, like their counterparts above, is that they do not show—either explicitly or implicitly—an appreciation for varying approaches to treat or handle the evaluated risks. In [46, 84] for example, based on their models and concept definitions, the authors apparently consider risk treatment only as a method to reduce the risk and its severity (usually in terms of impact or probability). Unless there is some implicit meaning associated with these concepts which gives them a wider range, these therefore exclude the different types of security actions discussed in Section 6.2.1 which are output from RM methodologies.

In the context of this research, another shortcoming suffered by all the models was their predominant focus on risks (including constituent elements such as vulnerabilities, threats and probabilities) and risk mitigation aspects (for example, security requirements and controls). This was as opposed to appreciating other important factors that aid companies in determining the actual security action or risk treatment. Examples of such factors from the NIST Guide include company policies or country laws and regulations.

Comparing the reviewed ontologies and models with the various stages in the RM process from Section 6.3.2, for risk identification, factors previously outlined such as asset, vulnerability, threat, unwanted incident are generally included in the models. Within risk estimation, only the factors of likelihood (held to be

synonymous with frequency and probability), consequence (or impact) and end priority (or risk value) can be constantly seen. Whereas for risk evaluation and treatment, none of the models explicitly consider laws, organizational policies or costs, as influential factors in deciding risk treatment specifically. Therefore, even though these models may be suited for their individual purposes, for the reasons outlined above, they are inadequate candidates for use "as is" in this research. The next section of this chapter will build on and reuse parts of these ontologies and models, and factors from Section 6.3.2, to design an appropriate ontology.

6.4.3 Risk-based Ontology

Under the guidance of the ontology creation methodology in King and Reinold [104], and a thorough examination of factors which influence security actions and requirements, the high-level ontology depicted in Figure 6.5 was developed. This ontology will be critical in supporting the progression between phases, and subsequent comparison of security actions across companies during BOF4WSS.

In defining this ontology, special emphasis was placed on including factors that were heavily supported in the literature reviewed in previous sections. A UML-type notation was preferred to specify this design as it provides a standard, widely accepted modelling tool to describe concepts and their relations. No standard ontology design notation was identified at the time of writing. The application of UML to ontology modelling is discussed by Wongthongtham [220], and examples of its use can be seen in Falbo et al. [54] and less explicitly in Houmb and Georg [84].

As a *risk* is the first significant point of contact, this discussion commences there. The identification of a risk typically involves an analysis of the *vulnerabilities* in *assets* and the *threats* leveraged by *threat agents* to exploit these vulnerabilities. A vulnerability is thus regarded as a weakness in an asset or an existing security element intended to protect an asset, an asset is anything of value to a business, a threat is an undesired event with an adverse impact on an asset, and lastly a threat agent is the cause of the threat [106]. Each of these aspects can

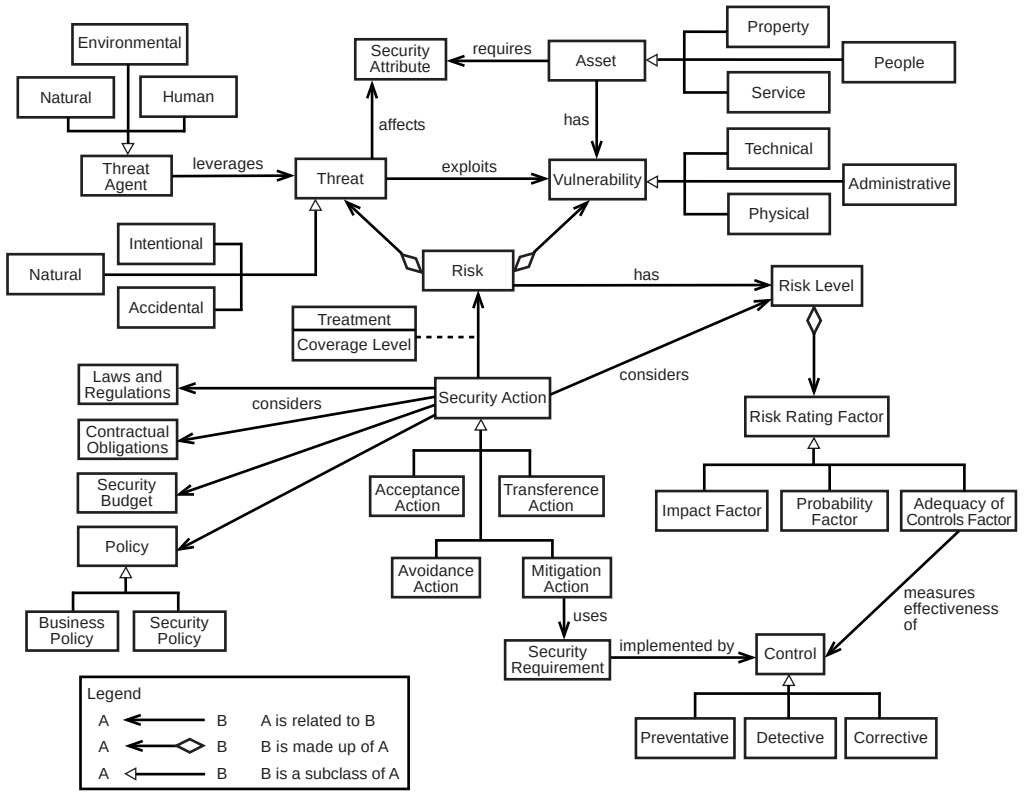


Figure 6.5: Risk-based ontology

be further decomposed to display specific types as shown in the diagram. These types are self-explanatory and are therefore not explained in detail.

A vulnerability can be technical, administrative or physical [106]. As shown by Firesmith [60], a company's assets can be generally regarded as its people, its property (hardware, software, data or facility) or the service (that is, an activity) provided. If one were assessing WS therefore, most assets would be of type property or service. A threat is typically an intentional, accidental or natural (floods or hurricanes) event [160]. Lastly, a threat agent can be natural, human or environmental (long-term power failure or chemicals) [195]. Readers are directed to Jones and Ashenden [94] for more information on threat agents including the elements needed for them to be effective, for example, motivation, capability, opportunity and catalyst.

Under the heading of risk identification, the *security attribute* concept which is identified by Fenz and Ekelhart [56] was included in the ontology. A security attribute 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. The benefit of the inclusion of a security attribute in the ontology is its ability to allow for a standard relationship between an asset and the threat which it may be affected by. In reality, an asset requires a certain security property, whereas a threat when executed will affect that security property. For example, sensitive data is required to be kept confidential, however if a hacker breaks into the database this will invariably compromise the confidentiality of that data.

Regarding risk estimation, the *risk level* concept is explicitly employed to represent the goal of this RM stage. Therefore, a risk after estimation has (or is assigned) a risk level. This level in essence is the value or priority grade indicating the severity of the risk. In determining the level, the following risk rating factors were found to be the most pertinent based on their acceptance and coverage: *impact*—the consequence (financial, reputation or client-related) if the risk materializes, *probability*—the likelihood the risk will occur, based on frequency of past occurrences and subjective estimation factors (such as intuition, educated guesses and indirect information) [3] (probability is estimated without consideration of any controls that might be in place, similar to work by Kairab [97]), and *adequacy of controls*—the measures adopted and their effectiveness to mitigate risk associated with vulnerabilities [97].

The factors within DREAD were considered but subsequently discarded due to their close links (discussed above) with the already included impact and probability concepts. The theory behind rating factors however is very open to interpretation and expression at varying levels of detail. Because of this, the intended use of the model will be the critical factor in deciding whether or not to include these aspects. The relationship between the factors chosen and risk level is depicted by the *risk rating factor* concept.

Security action is primarily linked to the risk treatment stage and refers to any way in which a company treats or handles a risk it faces. As stated in previous sections, to decide on a method (namely security action or risk treatment action)

to address a risk, a number of factors should be considered. The most critical of these identified in this work are the *risk level*, *relevant laws and regulations*, *contractual obligations* (that a company is previously legally binded to), the *security budget* (balancing security actions and limited resources is paramount), and *policies*, decomposed into *business*-related (organizational principles, business requirements) and *security*-specific (unique, tailored security directives). This accounts for the 'considers' relationship from security action to these concepts.

Emerging from the security action concept itself, specific types of actions (supported by [94, 106, 205]) are apparent. These are *acceptance* actions (if risk level is negligible or cost to mitigate exceed value of the asset), *mitigation* actions (given a risk is to be mitigated, that is, its level reduced to an acceptable degree), *transference* actions (if risk is to be assigned to another party through insurance), and *avoidance* actions (if the risk is handled by eliminating the risk cause, for example decommissioning the vulnerable asset). As with any subtype-to-type relationship, aspects considered by the security action are also noted for sub-actions. A similar example of the security action and sub-action concepts is given by Falbo et al. [54]. Even though the four types of action listed are not the only types seen in the literature (see Stoneburner et al. [195]), they are the most common and accepted generalizations.

The *security requirement* concept is used by the mitigation action to define high-to-medium level desires with respect to the information security goals. These security desires (when implemented) act to reduce the identified risk to an acceptable level or simply mitigate the risk. To implement the security requirement to a more detailed extent a *control* is employed. A control is broadly defined as any risk-reducing measure, technical or non-technical [195]. Control types prevalent in the literature and included in this design are *preventative*—measures to deter undesirable events from occurring, *detective*—measures that indicate an undesirable event has happened, and *corrective*—measures to correct, or recover from the damage caused by undesirable events [106]. The relations adopted in this ontology between mitigation action, security requirement and control are supported

by Mayer et al. [116].

The final concept presented is the *treatment* association concept. This attempts to further describe the relationship between the risk and security action. In the handling or treating of a risk, reality dictates that a security action might not always completely address a particular risk. Practitioners often accept that some of the risk is left behind. This 'left over' risk is commonly referred to as the residual risk [195]. The treatment association concept in the ontology therefore seeks to capture the known degree to which an action covers a risk. This degree is represented in the *coverage level* attribute of the treatment concept. For example, a company may note that a particular security action only offers 'partial treatment coverage' for a stated security risk.

6.5 Summary

In this chapter an analysis was conducted into the various risk-related methodologies in practice that are used to determine security actions and requirements. This assessment enabled the identification of several critical factors within the literature which influence how security actions are derived. With these factors identified, an ontology was used to model them and construct a conceptual representation of their relationships to each other. These tasks fulfilled the aims of this chapter, which were to expand on the Solution Model from Chapter 5 and implement the Security Action Analysis and Ontology Design components.

The real novelty of the ontology designed in this chapter is its emphasis on the core factors that influence the definition of a security action. This focus, coupled with the model's appreciation of multiple ways to treat risks, form the main differentiating characteristics when compared to existing security ontologies.

The next research steps include using the ontology to define a formal XML-based language and creating a Risk Catalogue to aid in comparing security actions across companies. These steps form part of the following chapter which documents the development of the system that implements the Solution Model.

Chapter 7

Security Action Specification & Comparison System (SASaCS) Prototype

At times, especially when the new system is an attempt to push the state of the art, it may be necessary to build a preliminary prototype as a proof of the concept. New solutions, particularly those based on new technology, may not be well accepted or well understood. In that situation, the project team can build a proof of concept prototype to illustrate that a solution is possible and feasible. — John W. Satzinger, Robert B. Jackson and Stephen D. Burd

7.1 Introduction

The work in the previous chapter began the implementation of the Solution Model from Chapter 5 with the Security Actions Analysis and Ontology Design steps. This chapter continues the Model's implementation by concentrating on the remaining steps and the system which they jointly result in. The three aims of Chapter 7 therefore are as follows.

The first aim is to introduce the aforementioned system, which is named the Security Action Specification and Comparison System (SASaCS). This is the system which the Solution Model culminates in and is responsible for matching risks and comparing their resulting security actions across companies. Secondly,

the chapter aims to report on how the prototype of SASaCS was created, including functionality, key design decisions and so on. Finally, there is a brief presentation of the developed prototype itself. This is intended to complement the chapter by adding visuals to the largely theoretical discourse.

In keeping with the notion of a prototype, the developed system is primarily intended as a proof-of-concept of the research ideas which underpin this doctoral research. This prototype also enables these research ideas to be practically explored, but more importantly, critically evaluated in subsequent sections.

To achieve the aims above, this chapter starts by providing an overview of SASaCS's goals. This is especially with respect to the Solution Model it implements. Next, a general description of the development methodology applied when creating the system is presented. Following this, design documentation is outlined for the complete version of SASaCS. This provides further insight for readers into the system's overarching goals.

Finally, the scope of the implemented prototype is discussed. At that point, the main goal is to describe the functionality of the prototype in the context of the full system, and to present the justification of key design and implementation decisions made. Exhaustive reporting on the prototype's development was avoided because the prime emphasis of this research is the research ideas it embodies.

7.2 SASaCS Overview

SASaCS has its roots in the Solution Model and 'Solution Model in action' constructs (from Sections 5.3 and 5.4 respectively) and consequently many of the system's aims and components mirror those models. To recap, the general goal of those constructs was to facilitate the streamlining or significant automation of the security actions negotiation process across interacting companies. By fulfilling this goal, the transition from the Requirements Elicitation to Negotiations phases in BOF4WSS should be much quicker and considerably less arduous on companies and their personnel. The next two paragraphs look in more detail at

the goals and functionality expected of SASaCS.

As discussed in Section 5.4, the first step would be businesses choosing risks from the Risk Catalogue (updating it as necessary) which are to be considered in the scenario. Next, companies would conduct their preferred RM methodologies to assess and define treatments for existing and any newly discovered risks. Findings and output from each company's RM methodologies would be entered into SASaCS and then encoded into an XML-based language. This language is the one discussed in the Solution Model which is based on the ontology from Chapter 6. For ease of reference, the language is called SADML (Security Action Definition Markup Language) and its actual creation is discussed later.

Documents from companies in the SADML format would then be processed by the system according to any preferences set by businesses' personnel. Security actions would next be compared by the system as much as is automatically possible. Detailed information on risks and security actions being compared would then be output to users. Where matching was not achievable or desired commonalities across documents were not identified, inconsistencies would be flagged for follow-up by business analysts and security professionals from companies. With this overview complete, the next section presents the methodology which guided the development of SASaCS.

7.3 Development Methodology

Sommerville [193] identifies four fundamental activities common to all development processes: Specification or Requirements Engineering, which defines the functionality of the system and constraints on its operation; Design and Implementation, that is, producing systems that meet specification; Validation, or ensuring systems meet their intended purpose; and Evolution which is having systems evolve to meet their changing needs. These activities enable systems to be developed in a structured manner to suit their varying aims.

For the development of SASaCS, these four activities were conducted (or in

the case of Evolution, are expected to be conducted) in a largely sequential fashion analogous to the Waterfall Model (WM) methodology of systems development (see van Vliet [213] for details). This methodology allowed for a well-organized and highly structured approach to development.

Albeit a useful technique, the WM does have a few shortcomings in terms of being too mechanical or rigid, especially in lower-level tasks such as requirements elicitation, discovery and testing. To compensate for this, the Prototyping development technique was also utilized. Prototyping refers to the "iterative process of developing an experimental system" [193] (p.395), and it is regarded as an extremely useful technique to employ throughout systems development (for example, to gather requirements, test systems and so on ([23, 193]). The use of Prototyping within the general software development process is discussed by Sommerville [193] and specifically in the WM by Cerpa and Verner [23].

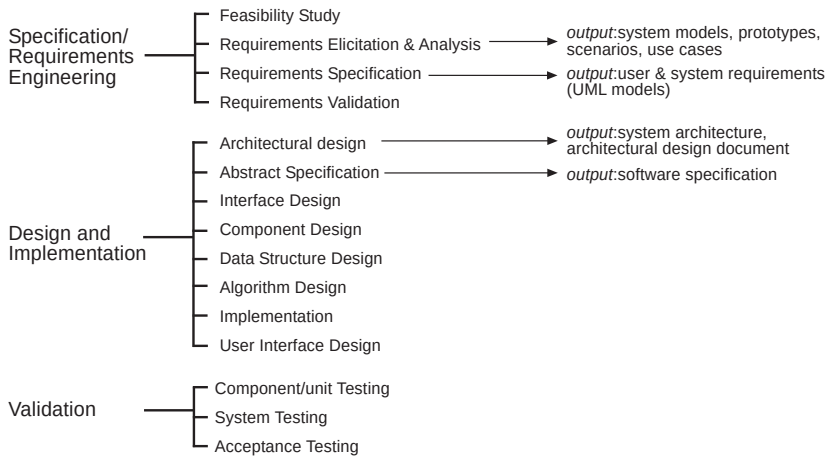


Figure 7.1: Fundamental activities in systems development ([193])

Figure 7.1 (based on Sommerville [193]) shows the main activities which guided SASaCS development in greater detail. In the diagram, the Evolution stage is excluded because it is not critical to this discussion. Additionally, for the benefit of the reader, Figure 7.1 provides an example of some of the output produced by activities. The next section focuses on the Specification and mainly Architectural Design phases displayed in the graphic.

7.4 SASaCS Development

7.4.1 Requirements Specification

As shown in Figure 7.1, the Requirements Elicitation and Analysis tasks are key activities within Specification/Requirements Engineering. To facilitate the gathering of requirements for SASaCS, the two techniques used were analysis of scenarios (that is, real-life examples or narrations of how the system would work) and small-scale prototypes. These enabled varying levels and perspectives of requirements to be viewed. The main scenario used is documented below.

> **Typical Scenario**: Company A and Company B first select a set of pertinent risks from an independently owned and moderated central risks catalogue/listing. These risks will factor into their RM methodologies. This is done to ensure that both companies start from the same point concerning the security risks which face the envisioned business scenario. The central risks listing is to be extensive and updated regularly. Public submissions/suggestions of risks are to be allowed, but moderated by third party security professionals.
>
> Having selected the security risks, companies individually apply their methodologies. If new risks are identified, these are shared with the business partners to ensure all entities consider the same risks. After each company's general RM process, the findings and documentation produced are entered into SASaCS. The system enables users to select the risks agreed previously and add respective information on (i) their security actions, that is, how the company has decided to handle the risks, and (ii) the various aspects (such as laws, risks' severity levels and so on) that have influenced how the company treats the risk. Once complete, the system encodes all the information entered into a standardized document format.
>
> Personnel from Company A and Company B would then meet up, bringing together their documents with the encoded information. These docu-

ments are fed into the comparison feature of the SASaCS tool. The system is expected to match the risks and compare the treatment of risks across companies. This therefore enables an easier negotiation, discussion and reconciliation of conflicting security actions. Various settings should also be available in the system to allow detailed security action and risk data comparison. Output of the system should be (i) a user-friendly interface where security actions and the related security risks are matched to some degree and displayed, and (ii) a list of flagged inconsistencies that represent exceptional situations and thus should be discussed by companies' analysts and security professionals.

This scenario of how the developed system would be used offers a general view of the major tasks that a fully implemented SASaCS should accommodate. With this documented, the next objective was a scenario analysis to determine and formally state the requirements of the system. The output of this analysis is included in a UML diagram in Figure 7.2. The UML use case diagram was preferred as it is a largely accepted graphical modelling technique to express system behaviours and functional requirements [193].

To briefly explain the use case diagram, the large box on the right represents the actual system, which in this case is SASaCS. The two entities on the left are external actors (or persons) that interact with the system. Within the system box, there are a number of use case ovals which provide a top-level description of the behaviour that the system is to exhibit. Figure 7.2 employs a $\ll uses \gg$ notation to show where system behaviour is part of a larger task. For example, the 'Maintain accessible central risk listing' behaviour is comprised of the 'Update global risk listing' and 'Accept new risk submissions' tasks. Conversely, the $\ll extends \gg$ notation is employed to show where sub-tasks are optional, for example, 'Synchronize to global risk listing' may or may not occur during the execution of the 'Maintain local risk listing' task. Finally, actor-to-use case lines are used to connect actors and the specific use cases within the system with which

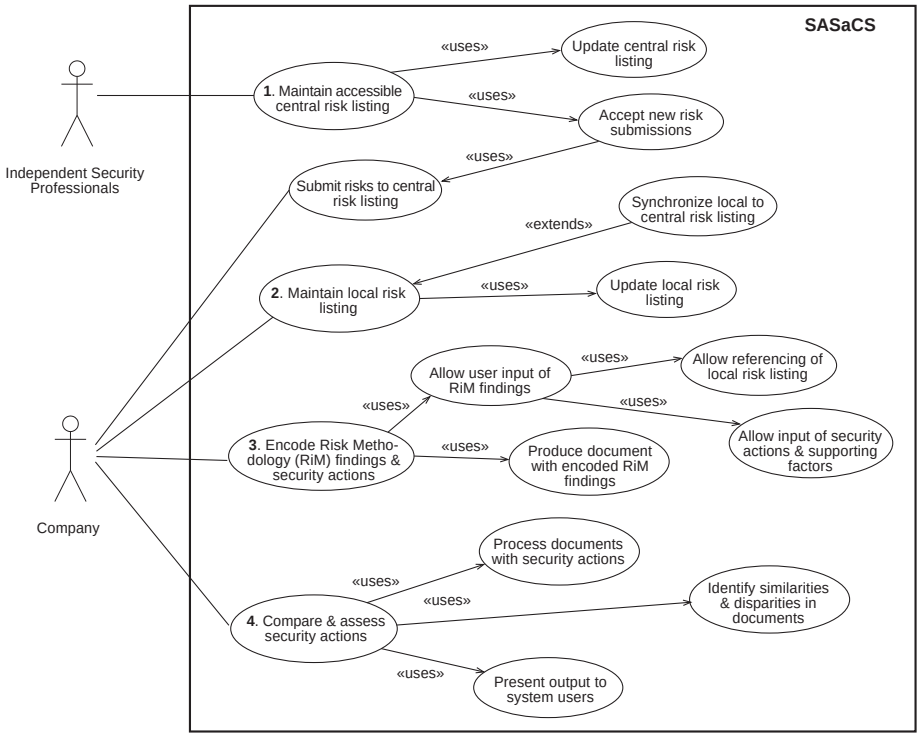


Figure 7.2: Use case model of the complete SASaCS version

they interact.

Progressing from the diagram's notation itself, there are four main use cases: 'Maintain accessible central risk listing', 'Maintain local risk listing', 'Encode Risk Methodology (RiM) findings & security actions' and 'Compare and assess security actions'. The tasks involved in these use cases largely mirror the activities discussed in the typical case scenario above. Furthermore, in some situations they add detail on how the activities may be executed (for example, the 'Synchronize local to central risk listing' case expands on how a local risk listing is maintained). As a result of these two factors, this section does not provide further descriptions of these use cases.

To accompany a use case diagram, Lunn [113] advocates the application of use case descriptions and documentation templates. These templates tend to be very useful as they enable more detail to be provided for the cases outlined in the diagram. Sommerville [193] adds to the discussion and notes that use cases can be further supplemented with UML sequence diagrams. The benefit of these is their ability to expand on a use case and provide a graphical, low-level model of

the sequence of interactions which constitute it. Both these techniques (templates and sequence diagrams) were employed to aid in the Requirements Engineering activity during SASaCS development.

Having provided an outline of the functionality expected by SASaCS, the next section takes a brief look at the design architecture. This provides an insight into the general design ideas supporting the complete system and also sets the context for the prototype implementation section which it precedes.

7.4.2 Architectural Design

The main goal with the architectural design stage is to lay the groundwork for a system which will satisfy the previously specified requirements. Sommerville [193] expands on this notion and describes the architectural design as being concerned with "establishing a basic structural framework that identifies the major components of a system and the communications between these components" (p.242). Because of these aims, the design task is a critical undertaking where various fundamental but important system decisions are made.

For SASaCS, the architecture design task was guided by [193] and therefore one of the main emphases was on identifying system components and their interactions (readers can refer again to Figure 7.1 for the overview). The architecture produced which is shown in Figure 7.3 is also heavily based on the 'Solution Model in action' construct. That model supplies a justification for the architecture conceived. From Figure 7.3, one can also see the implementation of the use case behaviours outlined in Figure 7.2. To supplement the diagram, an example of a typical process flow is given. This example provides practical insight and further detail into the workings of the model that were excluded from the diagram to avoid clutter. Here are the steps.

1. Businesses reference the central Web site and agree on a risks catalogue version to use for their interactions. Each entity then synchronizes their local risk catalogue to the agreed Central Risk Catalogue.

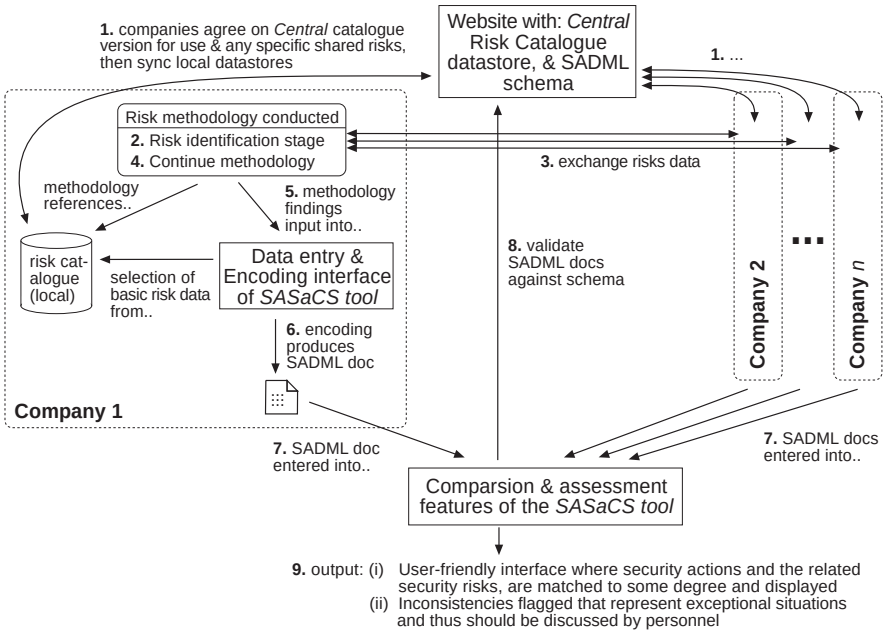


Figure 7.3: SASaCS architecture overview

2. Companies then carry out their individual risk identification processes for the scenario, in which they also consider risks from their now-synchronized local risks catalogue.
3. If risks are found which are not included in a business' synchronized risks catalogue, these are then communicated to the other parties so that all of them can assess these risks in their subsequent risk assessment stages. In more specific terms, companies will add these newly suggested risks to their local risk catalogues and refer to them later during risk assessment and ultimately data entry and encoding.
4. Businesses then branch off to continue conducting their RM methodologies. Key aspects of interest include risk severity/priority levels and other factors that aided in how entities decided to treat the identified risks. Full documentation should be made and kept.
5. Each entity would then input data from their RM methodology documentation into the Data Entry interface of SASaCS. The aim is to get pertinent information on risks and factors influencing risk treatment choices (that is, security actions) into the system. A typical process for a single risk would

be:

(a) Within the Data Entry interface, select a risk (the system would query from the local risk catalogue to allow risks to be chosen)
(b) Find the selected risk in the risk methodology documentation
(c) Enter pertinent information related to the risk into the Data Entry interface. This encompasses data on the risk's severity level, factors (such as policies, budgetary constraints and so on) that influenced the decision to choose a security action to address that risk and, finally, the security action itself.

6. After companies are finished, the Encoding interface of SASaCS allows them to encode all the information entered and have it output in a SADML document.
7. At the comparison and assessment stage, businesses then bring their documents together and allow them to be assessed by the Comparison features of SASaCS.
8. The system validates the documents to ensure they are well-formed (a common XML check) and conform to purpose-built SADML schema rules.
9. The system conducts the comparison of security actions. The predefined, shared risks are a critical component during comparison as they form the common base on which security actions can be matched and compared. Finally, the system outputs (i) a user-friendly report where security actions and the related security risks for all companies are matched and displayed on screen, and (ii) inconsistencies that represent exceptional situations which need to be followed up and further discussed by companies' personnel.

Having reported on the design of the full SASaCS version, the next section narrows the scope to focus on the prototype system that was implemented.

7.5 Prototype Implementation Scope

7.5.1 Functionality

In deciding on the implementation scope of the SASaCS prototype, emphasis was placed on two aspects. First, identifying areas which would demonstrate some of the core research ideas embodied in the system, and second, using areas that provided a platform for a good evaluation which appreciated the limitations of this project. The functionalities chosen which met this criteria were the 'Encoding of Risk Methodology (RiM) findings & security actions' (Use case 3) and the 'Comparison and assessing of security actions' (Use case 4). These cases (viewable in Figure 7.2) adequately characterize core underlying research themes and also encompass the overarching goal of easing the progression between Requirements Elicitation and Negotiations phases in BOF4WSS.

Use cases 1 and 2 were accepted as important to this research but their implementation was not seen to add significant value to the prototype or any subsequent evaluation of it. They were therefore excluded.

To expand briefly on the functionality of Use case 3, two enabling components deserve mention. These are, the Risk Catalogue from which risks are selected (this relates to both the central and local catalogues), and the SADML format which is the formal language used to encode risk methodology findings and security actions data. The former of these aspects encompasses the creation of a simple database system and its population with risks data. As this task is relatively trivial, further detail on the construction process is not given; the database structure for the risks system can however be seen in the more general entity diagrams in Appendix A.

The next enabling component promotes the SADML format as a novel document format to define RM/RA data which may also have uses independently of the SASaCS tool. Example usages of SADML include using it as a document exchange format or for storage of RM data from various disparate techniques. With appreciation of the novelties of this new XML-based format, the next section

gives an overview.

7.5.2 Security Action Definition Markup Language (SADML)

The structure of SADML was conceived to mirror the knowledge captured in the ontology (from Chapter 6) and as such, a number of the ontology's concepts have been 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 security policy relates to multiple security actions, this should be appreciated). An excerpt of such a structure is displayed in Figure 7.4.

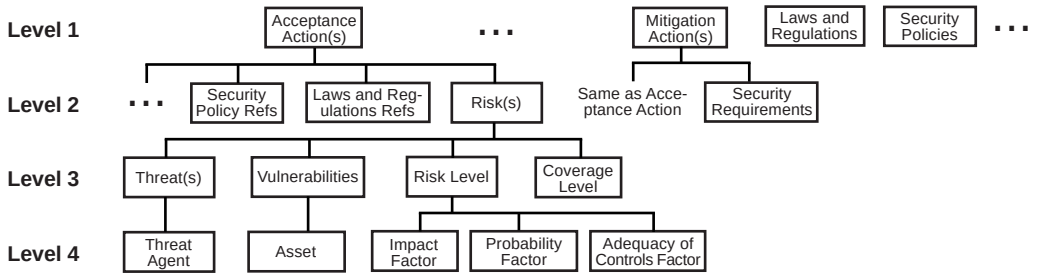


Figure 7.4: A hierarchical structure of the elements

As shown in Figure 7.4, the top-level elements are based around the security actions and treatment factors (such as laws and policies). This was because the security actions were found to be encompassing elements in which various other concepts (for example, risks and threats) could be logically composed. Similarly, the treatment factors were independent elements which were only referenced (using XML elements/tags ending in 'Refs') in security actions. In practice therefore, there might be an acceptance action that addresses one or more risks, and one or more security policies may have been used/referenced to determine that action. Part of the SADML format is presented in Code Snippet 2. The + sign indicates that there is additional data which is not displayed here.

```
 <needsBase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:risksx-schema">
   <mitigationActions>
     <mitigationAction>
       <name>Security action for auditing/logging purposes</name>
       <details>Mitigation action for auditing/logging purposes</details>
       <risks>
         <risk id="GR1">
+          <threats>
+          <vulnerabilities>
+          <riskLevel value="high">
           <riskComment>Risk associated with general logging, auditing, and ...</riskComment>
           <riskActionCoverageOfRisk>
             <coverageLevel>full coverage</coverageLevel>
             <coverageDetail />
           </riskActionCoverageOfRisk>
         </risk>
+        <risk id="GR2">
       </risks>
       <lawAndRegulationRefs>
         <lawAndRegulationRef idref="LR215"><relationToRiskAction /></lawAndRegulationRef>
       </lawAndRegulationRefs>
+      <contractualObligationRefs>
       <businessPolicyRefs />
+      <securityPolicyRefs>
       <securityBudgetRefs />
       <securityRequirementRefs>
         <securityRequirementRef idref="SR230"> ... </securityRequirementRef>
       </securityRequirementRefs>
     </mitigationAction>
+    <mitigationAction>
   </mitigationActions>
+  <acceptanceActions>
   <transferenceActions />
   <avoidanceActions />
   <lawsAndRegulations>
     <lawAndRegulation id="LR215">Sarbanes-Oxley Act of 2002 (SOX) requires that companies
        maintain a ready available, verifiable audit trail and ... </lawAndRegulation>
   </lawsAndRegulations>
+  <contractualObligations>
   <businessPolicies />
+  <securityPolicies>
   <securityBudgets />
   <securityRequirements>
     <securityRequirement id="SR230">A part of reliable business process execution both within
        the company and externally, involves comprehensive logging ... </securityRequirement>
     ...
   </securityRequirements>
 </needsBase>
```

Code Snippet 2: SADML example

The snippet gives a snapshot of the implemented XML hierarchy mentioned prior. The core of the SADML format is described in the XML schema designed (indicated by *urn:risksx-schema* in the document).

The novelty of SADML is rooted in the unique perspective it gives on security actions and the emphasis it places on aspects which have been overlooked in other languages. From a review of the literature, only one related generic (that is, not system-specific) XML-based language was identified. This was the Enterprise Security Requirement Markup Language (ESRML) [176].

ESRML, as Roy et al. [176] note, is for specifying enterprise information security requirements which are in compliance with the ISO 17799 standard. The knowledge gap filled by this language is similar to the area targeted by SADML. This is particularly as ESRML looks at (i) the higher layers of security and not specifics such as access control (such as the eXtensible Access Control Markup Language (XACML)) or identity management (such as the Security Assertion Markup Language(SAML)), and (ii) sharing and exchanging the enterprise security information across companies for business purposes. The shortcomings of ESRML in terms of this research stem from its lack of emphasis on factors which significantly influence security actions, and its concentration on risk mitigation as opposed to other ways to treat risks. These are areas addressed in SADML. With the prototype's scope now presented, the next section discusses some important implementation decisions made.

7.5.3 Justification of Implementation Decisions

To implement the functionalities outlined for the prototype, three fundamental questions were identified. These pertained to what software development tools to employ, exactly what type of application—desktop versus Web-based—would be most appropriate to build and lastly, for security actions comparison, should the system be developed to accommodate any number of companies or would a more manageable subset be adequate.

The answer to the first of these questions was found in the use of Mi-

crosoft Visual Studio 2008 [126] (VB.NET language) and SQL Server 2005 [125] to support SASaCS prototype development. These tools were preferred (to, for example, a Java-based tool-set) because of the researcher's previous experience in working with them and the vast array of predefined controls available. Both of these factors therefore acted to speed-up and enhance the development process.

The second decision related to the type of application created and this was important because of the varying nuances linked to desktop and Web-based applications (see Liberty and Hurwitz [109] for detailed benefits and differences). After careful analysis, a desktop application was chosen as it allowed for the development of a much more flexible solution in terms of user interface, data entry and format of system output.

The final decision focused on the general question of whether the prototype's comparison features should accommodate any number of companies, similar to the full SASaCS version, or if catering to a smaller subset of companies (for example, two or three) would suffice. Following deliberations on this topic, it was decided that allowing for these operations between two companies would suffice. Although increasing the number of companies might be better as it would more closely resemble the full system version, for the purposes of the prototype, the added value of doing this did not merit the increased implementation complexity. This was mainly because the general conceptual approach which would be applied to two companies is also the way in which multiple companies would be handled.

7.6 Prototype

The purpose of this section is to complement the development discussion thus far and display a few screenshots of the created SASaCS prototype tool. These particular visuals were chosen as they exemplify implemented Use cases identified previously. Four screenshots are shown in this section and an additional two in Appendix A.

Figure 7.5 displays the data entry screen in which data on risks to the

companies/scenario are added or edited. These data include information on the relevant asset, vulnerability and threat. The retractable bottom half of the screen is where information on the risk's (priority) level is entered. The second screenshot in Figure 7.6 shows the Security Action data entry screen. The top holds general information on the specific security action, in the middle there is a list of risks which the action addresses and lastly, factors which influenced the risk treatment choice are entered at the bottom of the screen.

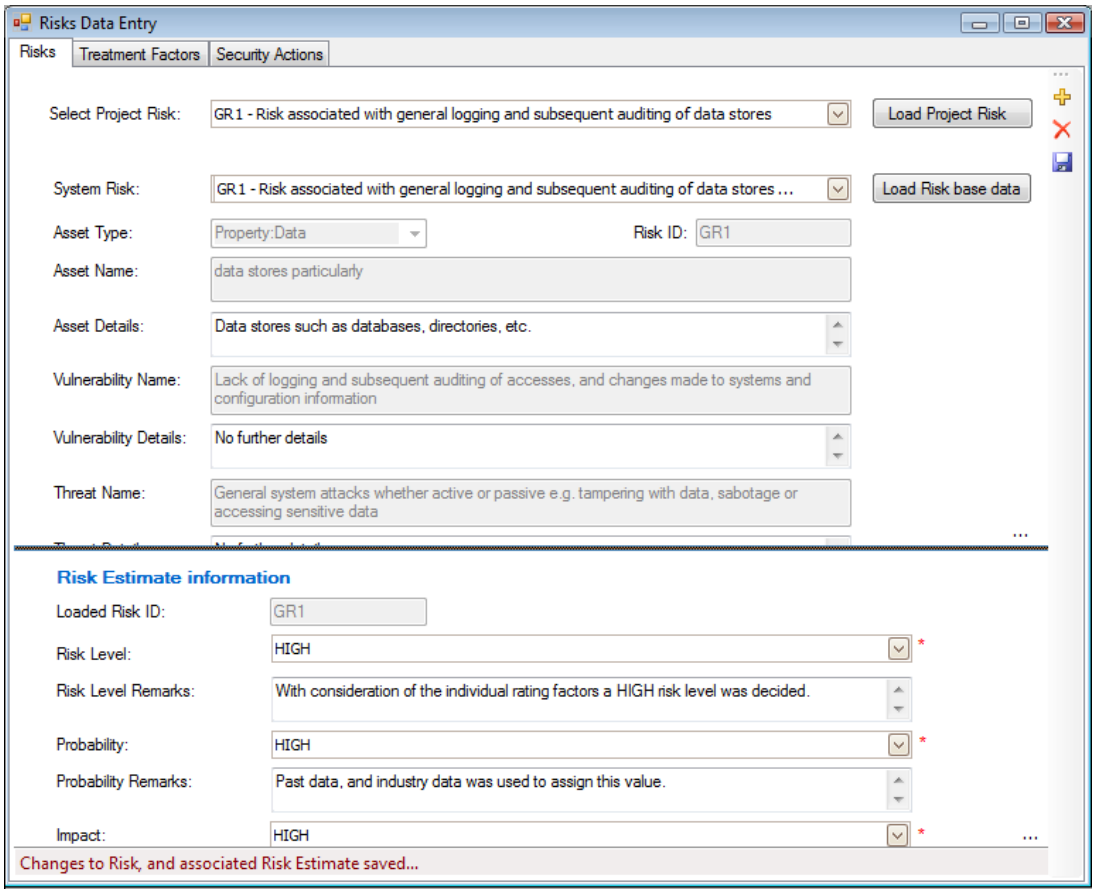


Figure 7.5: SASaCS Risk Data Entry screenshot

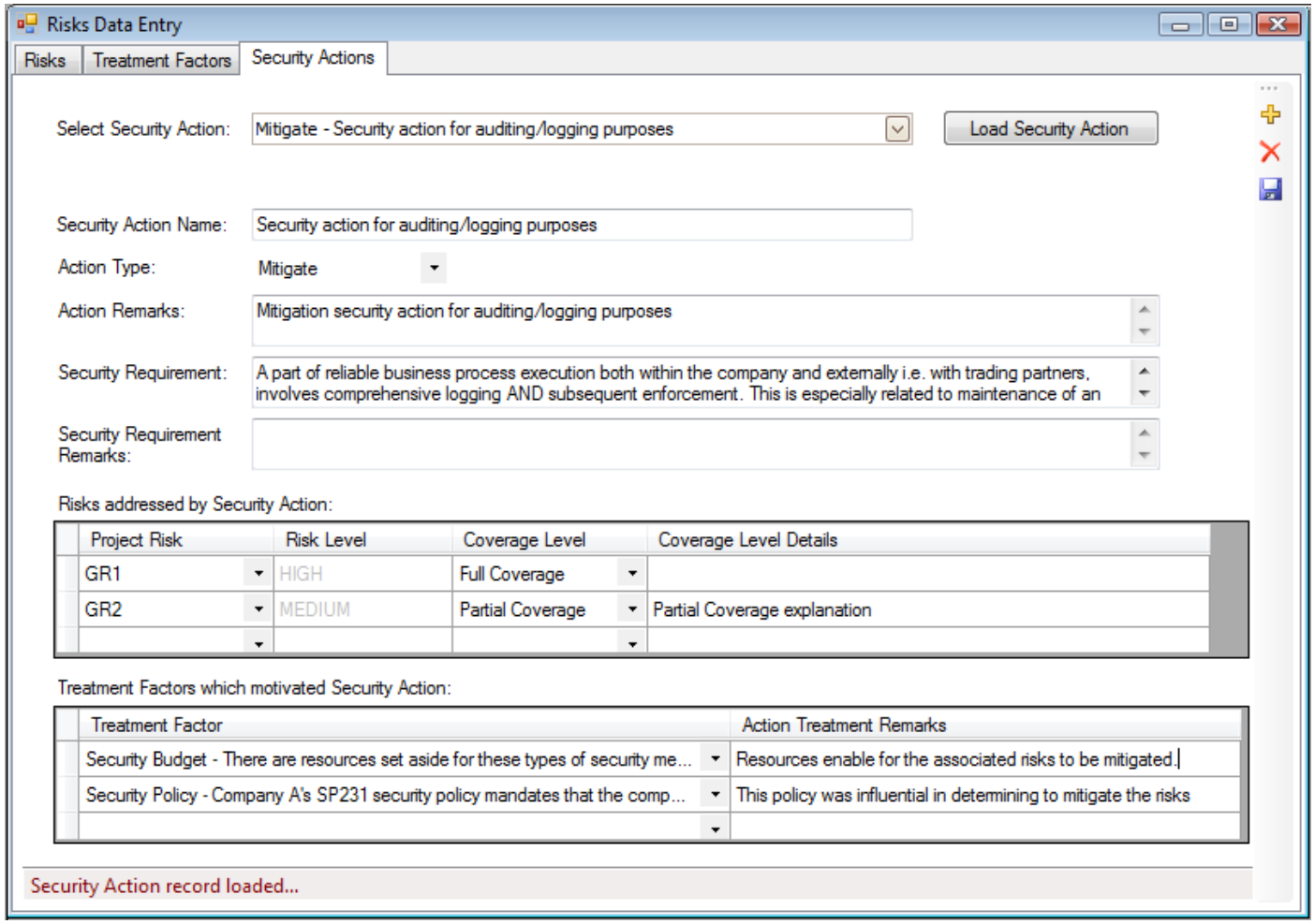


Figure 7.6: SASaCS Security Action Data Entry screenshot

Progressing from the figures listed above which illustrate the data entry features of Use case 3, Figure 7.7 focuses on Use case 4 and capturing the various settings and options available in SASaCS to enhance security action comparisons. Four individual tabs (on the left) are shown, each with numerous settings that companies can modify to customize comparison tasks.

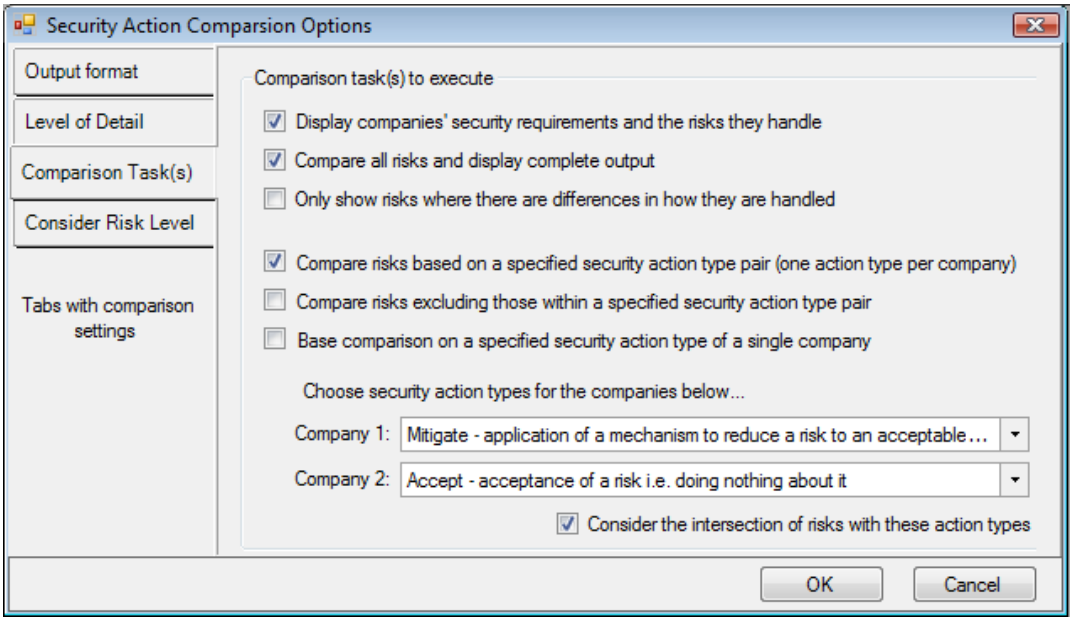


Figure 7.7: SASaCS Security Action Comparison Options screenshot

Generally, comparison settings include (i) outputting report data in HTML or plain text (and thus more transferable and malleable) format, (ii) only displaying risks where there are conflicts in how companies handle them (this would be based on security action types), (iii) allowing businesses to request to view shared risks that are handled by specific security action types (for example, personnel requesting to see risks that Company A wants to mitigate but Company B wants to accept), (iv) outputting the report data in varying levels of detail, therefore enabling companies to view bare minimum information (as done in Figure 7.8) or all risk and treatment factor data (as shown in Appendix A), and lastly (v) allowing comparisons to be conducted that factor in a risk's priority level—this parameter, for example, could enable risks with conflicting treatments to only be output/shown if they are different risk levels/priorities. The setting in (v)

is useful particularly if personnel suspect that the reason for conflicting security actions is due to different perspectives on the respective risk's priority.

The result of the security actions comparison is shown in Figure 7.8. This diagram depicts the 'user-friendly output interface (HTML is used) where security actions and the related security risks are matched to some degree and displayed', which was mentioned in Figure 7.3. This output, presented in the form of a colour-coded report (red is used to highlight conflicting actions), is expected to be one of the key aids to businesses' decision makers. With relevant information on a conflicting security action viewable on screen, analysts and security professionals could apply their judgment, immediately discuss the issue and agree on a course of action. This removes various of the initial semantic issues, security requirements format problems and difficulties emanating from incomplete security information. Furthermore, if the output on screen shows the existence of few conflicting actions, this might indicate that detailed negotiations are not required at this point as companies have similar security postures.

Figure 7.8 is a screenshot of the output produced when the system was set to execute a high-level comparison task of security actions across companies. All of the features covered in this and the previous two paragraphs address research Step 4 in Section 5.2.3. More detailed prototype diagrams can be seen in Appendix A, which also includes different perspectives the SASaCS tool's output.

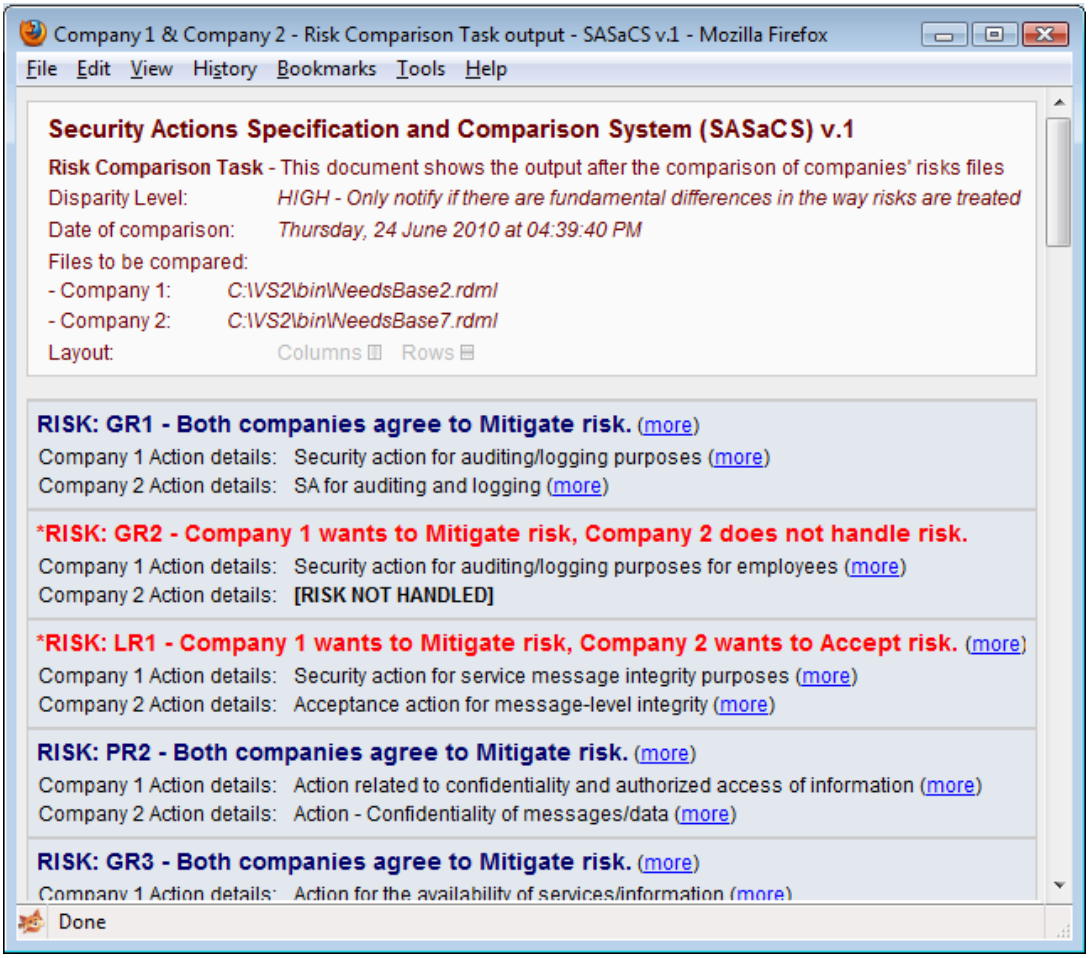


Figure 7.8: SASaCS Security Action Comparison output screenshot in Firefox

7.7 Summary

This chapter introduced the implementation of the Solution Model (namely SASaCS) and reported on the system prototype developed. The functionality selected for the prototype was seen to adequately cover the core research areas and also to give insight into the prime use of the Model and general SASaCS tool.

Having completed the main proposals of this research project, Chapter 8 commences the evaluation process. That chapter critically assesses whether the SASaCS tool, and the ontology (from Chapter 6) which it is based on, are compatible with existing Risk Management/Assessment approaches used by real-world companies today. Compatibility is crucial considering the interaction required between these approaches and the tool as exhibited in Figure 5.2 in Chapter 5.

Chapter 8

Evaluating the Compatibility of the Tool and Ontology

Compatible: able to exist together harmoniously — Collins Dictionary

8.1 Introduction

The evaluation process of this research contained two main stages, each with a different method of analysis. The first stage is covered in this chapter and the second in Chapter 9. Chapter 8's evaluation task involved the assessment of the Security Action Specification and Comparison System (SASaCS) tool and the ontology (from Chapter 6) which it embodies. The overarching research point which guided this evaluation stage was drawn from the unanswered query on seamlessness of mapping (under Step 1) identified in Section 5.2.3.

In general therefore, this chapter aims to consider whether the tool and ontology are compatible with (and thus allowed a good mapping to and from) existing security Risk Management (RM) and Risk Assessment (RA) approaches. Compatibility forms a vital requirement because, as the 'Solution Model in action' in Chapter 5 depicts, there is a noteworthy degree of interaction between RM/RA approaches (and their respective software outputs) and the SASaCS tool. This involves risk data (from the Risk catalogue) being input into the RM

process/software and output from that RM process/software (such as security actions and influential factors) being input back into the SASaCS tool.

To achieve the aim of this chapter, first the evaluation method used is presented and justified. A detailed assessment of how well RM/RA approaches can be mapped to the tool and the underlying ontology is then presented. Special attention is paid to particularly good mappings as this might highlight fields in which the tool and ontology are better suited. Difficulties incurred are also of great interest because these may suggest weaknesses in this work's proposals or areas where they will not apply.

The last section in the chapter pulls together the compatibility evaluation of tool and ontology (and generally Solution Model) through the use of a full scenario analysis. This enables a more complete evaluation of the proposals because unlike the compatibility assessment preceding it, it progresses from the initial Risk Catalogue to the final SASaCS output produced. Topics covered include: how Risk Catalogue data is transferred to the RM/RA approaches/software (as expected in Section 5.4), how typical RM/RA approach information is represented in SADML, and finally, how close, if at all, SASaCS can bring together the different RM/RA approaches used by companies to ease transition in BOF4WSS.

8.2 Evaluation Method

To guide this evaluation and add structure to its process, the method for mapping security guidelines and standards to an existing ontology (both high-level and formal) proposed by Fenz et al. [57] was employed. This provided a tested technique in which data could be sourced from a security guideline, standard or methodology and then mapped on to an existing ontology. Formally, the methodology's steps are **ontological analysis**, **knowledge base analysis**, **mapping concepts and relations**, **mapping knowledge** and lastly, **evaluation**; these are described in future sections. Through the completion of most of these steps, a detailed assessment was carried out to determine how well the tool and ontology

mapped, and thus to ascertain how compatible these are with existing RM/RA approaches.

To provide the basis of the compatibility evaluation, two RM/RA methodologies were chosen, CORAS [47] (which was introduced in previous chapters) and EBIOS [42]. These were selected because (i) they are well-known, used and established techniques [47, 42], (ii) there was extensive documentation openly available on each, and (iii) they both have supporting software which generate 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 SASaCS tool.

Honing in on the question of compatibility, the specific objective during this stage of evaluation is to investigate whether there was an adequate mapping possible from the output (particularly the software's machine-readable output) of existing RM/RA approaches, such as CORAS and EBIOS, to the SASaCS tool and ontology proposed in this research. Here, an 'adequate mapping' is defined as one where a majority of concepts could be easily mapped. General counting and measurement of successful mappings were therefore the data analysis techniques used. In the interest of time, the mapping was done at a conceptual and not practical level (that is, there was no coding involved). It was felt that the conceptual level mapping would supply an acceptable proof-of-concept test of compatibility to RM/RA methods.

The next section applies the evaluation method in Fenz et al. [57] to each of the chosen RM/RA methodologies. The **ontology analysis** step was skipped as this activity is similar to the ontology discussion covered in Chapter 6. The **evaluation** step was also excluded as the method's authors devised that step for cases where very formal ontologies (defined in OWL, etc.) were used. The main evaluation in terms of this research therefore will be considering the good aspects, the difficulties or thought-provoking points during mapping, assessing output and/or concepts which could or could not be mapped from RM/RA methods to the tool proposed, comparison of mappings and so on.

8.3 Mapping EBIOS to the Tool and Ontology

EBIOS is a risk management approach created by Central Information Systems Security Division, under the French General Secretariat of National Defence. It provides a methodology and supporting software for assessing and treating risks in the field of information systems security [42].

Following Fenz et al. [57], the first step in the mapping process was the **knowledge base analysis**. This step identifies the main entities/concepts and relationships in the RM approach being studied. For this and various of the following tasks, the EBIOS software's XML output (from sample and purpose-built case studies) and the general documentation on EBIOS in [42] were utilized. Throughout this section, keywords which identify concepts, entities, attributes and database fields in EBIOS and the tool and ontology are shown in italics for ease of reference.

From the knowledge base analysis process, some of the concepts identified in EBIOS included: an *entity*, defined as an asset; a *menace*, which defines a threat to an entity; a *vulnerability*, weakness/flaw in terms of information systems security; a *security objective* which is the expression of the intention to counter risks or threats and/or comply with the organizational security policies and assumptions; a *security functional requirement*, which is a security function to be implemented to contribute to the fulfilment of a security objective; and an *assurance requirement*, defined as the specification of assurance provided by security functions implemented to cover security objectives. A full list of concepts is displayed in subsequent sections.

The next step consisted of **mapping the concepts and relations** identified in EBIOS (particularly those from its software XML output) to the SASaCS tool and embodied ontology. This mapping was approached from two levels. The first level involved mapping the EBIOS concepts and relations identified to the ontology. The mapping is displayed in Figure 8.1, where the EBIOS concepts are in boxes with dashed lines and ontology concepts in solid lines. A description of this

mapping was not given here because it was considered relatively self-explanatory. Furthermore, mapping details and justifications are available in the subsequent paragraphs (and the set of tables presented). The ideal use and benefit of this pictorial mapping was the ease with which one can visualize high-level similarities across models and techniques. It should be noted that as done by Fenz et al. [57], and to avoid clutter, mainly the mappable concepts from EBIOS were shown in Figure 8.1.

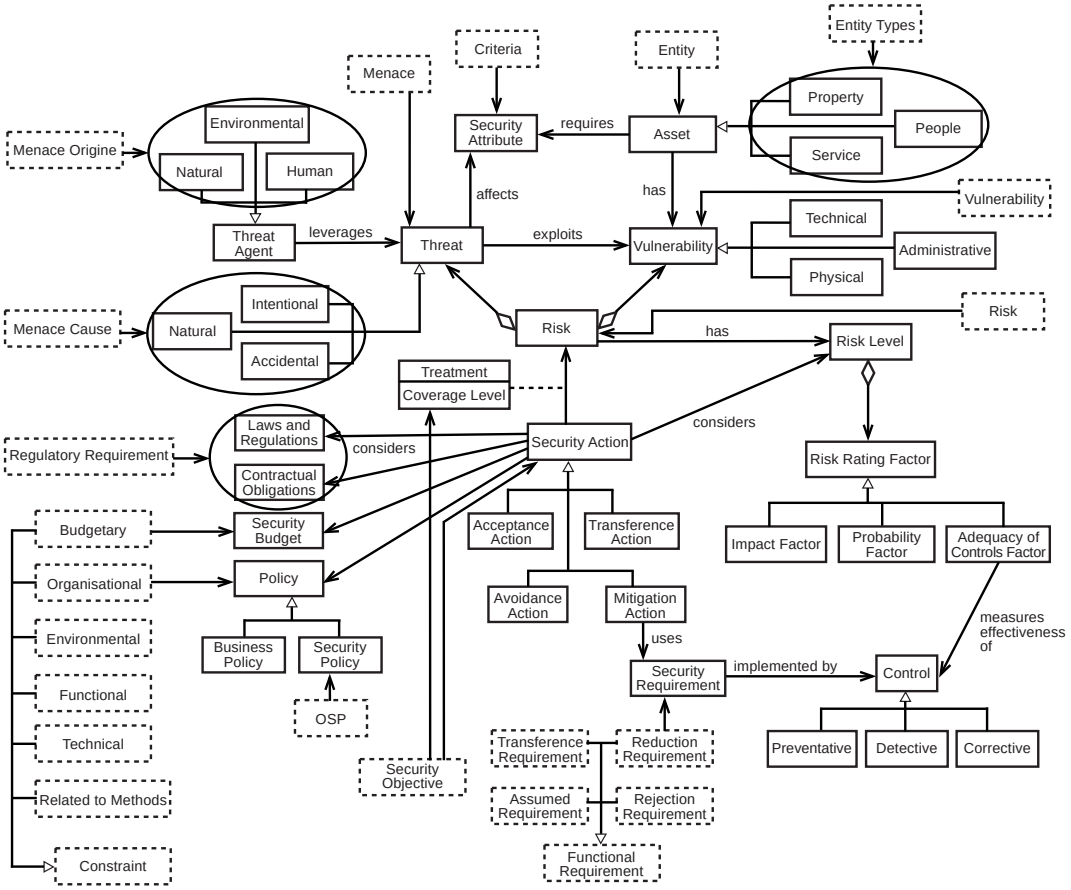


Figure 8.1: Mapping EBIOS concepts to proposed ontology

Second, and more aptly, mapping was undertaken from EBIOS to the low-level implementation of the ontology in the SASaCS tool. This low-level implementation was represented by the database schema actually used to store data in the tool. To depict the database schema and structure, an Entity Relationship Diagram (ERD) was used. The SASaCS database ERD and a description of its tables are given in Appendix A.

For the low-level mapping, relevant data from EBIOS and the tool were analysed and Tables 8.1, 8.2, 8.3, 8.4, 8.5 and 8.6 were created. These tables list all the XML elements in EBIOS methodology output, describe them, identify the ERD concept which they would map to (remember, the aim in the evaluation process is to conduct a conceptual mapping) and then provide a brief justification of the mappings chosen. It is worth noting that in the ERD, the term 'Risk action' (which is short for risk treatment choice/action) was used as opposed to 'security action'. This term therefore will mainly be seen in the mappings involving EBIOS now and CORAS in subsequent sections. There was no particular reason for this and both concepts are regarded as synonymous based on previous definitions (for example, in Appendix A).

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
<AssuranceRequirement>	Specification of the assurance provided by security functions implemented to cover *SecurityObjectives*	*None* - the ontology and ERD have not considered assurance aspects of security
ID	Unique identifier	
abbreviation	Abbreviated term for the requirement	
label	Name	
ccFamily	Respective Common Criteria Requirement Family	
description	Description	
justification	Justification	
baseID		
origine	Origin of requirement, e.g. CC (Common Criteria)	
<objectives>	*SecurityObjectives* which relate to the requirement	
<Constraint>	Constraint of, or faced by the organization	The ontology has no generic *Constraint* concept, but a mapping is possible at the ERD level to the *TreatmentFactor* table as both define limitations on a business
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*None*
theme	*Theme* describing type of constraint	*TreatmentFactor.treatment_factor_type_id* - this field would identify the respective theme/type
description	Description	*TreatmentFactor.treatment_factor_details*
coverLevel	Level of coverage of constraint by *Security Objective* . Options: {No Cover, Partial Cover, Complete Cover}	*None* - Coverage of Constraints is not accommodated in the ontology or ERD. Treatment factors/constraints are considered when determining the risk action, but no coverage value is stored
justification	Justification	*None*
<Criteria>	Characteristic of an Essential Element allowing the various sensitivities to be assessed	The corresponding ontology and ERD-level concept is *SecurityAttribute* . Mapping is direct as the concepts have the same meanings
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name of the criteria e.g. Confidentiality, Integrity, Availability	*SecurityAttribute.attribute_name*
description	Description	*None*
<Disaster>	Specific damage types relating to security criteria	*None* - the ontology and ERD have no related concepts
ID	Unique identifier	
label	Name of the damage e.g. disclosure, interruption	
criteria	*Criteria* which relate to the disaster	
description	Description	
<Entity>	Entity - An asset such as an organisation, site, personnel, etc	The corresponding ontology concept is *Asset* . *Asset* , *Risk* , and *ProjectRisk* tables provide a mapping at the ERD level
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*Asset.asset_name*
type	The relevant *EntityType*	*Asset.asset_type*
description	Description	*ProjectRisk.asset_details*

Table 8.1: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (1)

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
<EntityType>	Main Entity Types (Entity is a type of Asset)	The corresponding ontology concepts are *Property* , *People* , *Service* . These define main types of assets and are found in the ERD in *Asset* table, specifically *Asset.asset_type* field
ID	Unique identifier	*None* - New identifiers are generated in the tool database
type	General types of entities, e.g. Software, Hardware, Personnel, Network, System	*Asset.asset_type* - Mapping could either be done from the EBIOS types to the types used in the tool, or EBIOS types could be added (as new records) to the tool via *Asset.asset_type* field
description	Description	*None*
<Function>	Function - Process contributing to the operation of an activity of an organisation which creates, modifies, destroys or conveys information	The related ontology concept is the *Service* or *Property,* asset type. This is because a Function is viewed in EBIOS as more-or-less a type of asset i.e. something being threatened. For the ERD mapping, *Service* or *Property* as the *Asset.asset_type* with further details on Function mentioned in *Asset.asset_name*
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*Asset.asset_name*
shema	Unknown	n/a
sensitive	True/False	*None* - No related fields as all assets are considered as sensitive
description	Description	*ProjectRisk.asset_details*
<FunctionnalRequirement>	Security function to be implemented to contribute to covering one or more *SecurityObjectives* and therefore treating related ISS risks. Treatment may consist of reducing, rejecting, transferring or assuming the risks	The ontology has no related Functional Requirement concept that is able to fully map data from EBIOS. A partial mapping can be done between Requirements that are towards reducing risks, and the *Security Requirement* concept (i.e. risk mitigation). At ERD-level, full mappings can be achieved where Requirements for reducing risks map to fields in the *RiskAction* table, specifically *RiskAction.security_requirement* , and Requirements for rejecting, transferring or assuming risks map to *RiskAction.action_detail* . In this right, they are viewed as detailed ways to treat a given risk
ID	Unique identifier	*None* - New identifiers are generated in the tool database
abbreviation	Abbreviated term for the requirement	*None*
label	Name	*None*
ccFamily	Respective Common Criteria (CC) Requirement Family	*None*
description	Description	For a rejecting, transferring or assuming treatment, *RiskAction.action_detail* . For reducing treatment, *RiskAction.security_requirement*
justification	Justification	*None*
baseID	Unknown	n/a
origine	Origin of requirement, e.g. ISO17799, Common Criteria	*None*
<objectives>	*SecurityObjectives* covered by the requirement	There is only a partial mapping because the ERD just allows for a one-to-one relationship between the Functional Requirements (here regarded as detailed risk treatments, and represented by *RiskAction.security_requirement* and *RiskAction.action_detail*) and the Security Objectives (i.e. *RiskActions*) which they cover. (They are in fact, fields of a row in the same table - *RiskAction* table.)
<Impact>	Consequence for an organisation when a threat is accomplished	The corresponding ontology concept is *ImpactFactor* . At the ERD-level, there is no mappable concept however as EBIOS uses impact in a slightly different way - it is used to help judge sensitivity rating of an asset.

Table 8.2: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (2)

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name of impact e.g. financial, reputation, infringement of laws/regulations	*None* - Conceptually these might define specific types of *Impact* in the tool
description	Description	*None*
<Information>	Information - information or item of knowledge that can be represented in a form allowing its communication, recording or processing	The related ontology concept is the *Property,* asset type. This is because an Information element is viewed in EBIOS as more-or-less a type of asset i.e. something being threatened. For the ERD mapping, *Property* as the *Asset.asset_type* with further details on Information mentioned in *Asset.asset_name*
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*Asset.asset_name*
sensitive	True/False	*None* - No related fields as all assets are considered as sensitive
description	Description	*ProjectRisk.asset_details*
<Menace>	Menace/Threat - Possible attack of a threat agent on assets	The corresponding ontology and ERD-level concept is *Threat* . Mapping is direct as the concepts have the same meanings
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name of menace/threat e.g. fire, eavesdropping	*Threat.threat_name*
selected	True/False. Whether menace applies to current study	Only concepts with "selected = true" would be mapped as these ones relate to the current study
description	Description	*ProjectRisk.threat_details* - field mappable provided that the Menace has a respective Risk to be mapped
justification	Justification	*None*
descriptionMenaceElement	Description of threat elements	*None*
potentiel	Attack Potential value i.e. ease with which a threat agent can carry out an attack	*None*
<MenaceThemeList>	Lists related Attack Method Themes	*None* - Neither the ontology or ERD accommodate the overarching concept of Attack Methods
<SeverityScale>	Lists the *Criteria* affected by menace	Full mapping with *ThreatSecurityAttribute* table (EBIOS Criteria concept maps to *SecurityAttribute*)
<MenaceCauseList>	*MenaceCauses* related to menace	Partial mapping as the ERD does not allow a *Threat* to have multiple types (*Threat.threat_type*)
<MenaceOrigineList>	*MenaceOrigine* related to menace	Partial mapping as the ERD does not allow a *Threat* to have multiple origins (*Threat.agent_id*)
<MenaceCause>	Cause of a *Menace*	This concept relates to *Threat* types in the ontology (*Natural* , *Intentional* , *Accidental*) and ERD
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name of menace cause e.g. deliberate, accidental	*Threat.threat_type* - this field provides adequate mapping. EBIOS' "deliberate" and "accidental" menace causes map to *Intentional* , and *Accidental* threat_type respectively
description	Description	*None*
<MenaceOrigine>	Types of Threat Agent behind a *Menace*	This relates to *ThreatAgent* types in the ontology (*Environmental* , *Human* , *Natural*) and ERD
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name of types e.g. environmental, human, natural	*ThreatAgent.agent_type* - this field provides an exact mapping
description	Description	*None*

Table 8.3: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (3)

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
<Organization>	Organisation under analysis	*None* - There is no mappable concept because generic information such as organizational data was not recorded in the ontology or ERD
ID	Unique identifier	
presentation	Information about the organization	
shema		
<ConstraintList>	Lists *Constraints* related to organization	
<RegulatoryReferenceList>	Lists the regulatory references of the organization. These link to the *RegulatoryRequirements*	
<ActivityDomainList>	Main functional domains in the organization	
<ExternalActorList>	External parties that interact with organization	
<OSP>	Security rule previously determined	This concept is tentatively mappable to the ontology's *SecurityPolicy* because of the way (i.e. more as previous security rule which influences current decisions on security needs) EBIOS regards it. At ERD-level OSP maps to *TreatmentFactor*, with its type (*treatment_factor_type_name*) set to *SecurityPolicy*
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*TreatmentFactor.treatment_factor_details* - at ERD-level, a Security rule is a *TreatmentFactor*
description	Description	*None*
coverLevel	Level of coverage of rule by *SecurityObjective*. Options: {No Cover, Partial Cover, Complete Cover}	*None* - Coverage of security rules is not accommodated in the ontology or ERD. Treatment factors/constraints are considered when determining the risk action, but no coverage value is stored
justification	Justification	*None*
<RegulatoryRequirement>	Regulatory requirement applicable to organization	The corresponding ontology concept is *Laws and Regulations*, because both represent statutory and regulatory rules to which an organization must consider in determining their security needs. At ERD-level Regulatory Requirement maps to *TreatmentFactor*, with its type (*TreatmentFactorType.treatment_factor_type_name*) set to 'Laws and Regulations'
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*TreatmentFactor.treatment_factor_details* - at ERD-level, a Requirement is a *TreatmentFactor*
reference	Reference for more detail on requirement	*None*
coverLevel	Level of coverage of requirement given by *SecurityObjective*. Options: {No Cover, Partial Cover, Complete Cover}	*None* - Coverage of regulatory rules is not accommodated in the ontology or ERD. Treatment factors/constraints are considered when determining the risk action, but no coverage value is stored
justification	Justification	*None*
<Risk>	Risk - Combination of a threat and the losses it can cause	The corresponding ontology concept is *Risk*, and at ERD-level, *Risk* and *ProjectRisk* can be used. Mapping between concepts is direct as they have similar meanings
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*None*
menace	*Menace* related to risk	*Risk.threat_id* - this field would identify the respective menace/threat
description	Description	*ProjectRisk.general_risk_info*

Table 8.4: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (4)

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
sof	Attack Potential value i.e. ease with which a menace's cause can carry out an attack	*None*
coverLevel	Level of coverage of risk given by *SecurityObjective* . Options: {No Cover, Partial Cover, Complete Cover}	In the ontology, this maps to *Treatment* concept's *Coverage level* attribute. At the ERD level, a mapping is possible to the *ProjectRiskAction.coverage_level* field
justification	Justification	*None*
<ScenarioPotentiality>	Threat opportunity. (Lists composite Attack Potential values)	*None*
<RiskScenario>	Similar data to a *Risk*. (relates to formalised threat in EBIOS tool)	Mapping is similar to that done with the Risk concept above
ID	Unique identifier	
label	Name	
menace	*Menace* related to risk	
description	Description	
<ScenarioPotentiality>	Threat opportunity. (Lists Attack Potential values)	
<SecurityObjective>	Expression of the intention to counter identified threats or risks (depending on the context) and/or comply with the organizational security policies and assumptions	The corresponding ontology & ERD concept is *SecurityAction, RiskAction*. Contrary to the suggestion that Security Objective infers risk mitigation, because (i) there is no definitive statement of this in EBIOS documentation, and (ii) the Functional Requirements that cover/implement an Objective can accommodate risk rejection, transference, and assumption, Security Objective was considered and therefore mapped as a generic term to *SecurityAction/RiskAction*
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Name	*RiskAction.risk_action_name*
state	Unknown	n/a
baseID	Unknown	n/a
type	Proprietary EBIOS language data type	n/a
content	Description/content of Security objective	*RiskAction.action_remarks*
resistance	Defines the required strength level of the security objective	*None* - Strength levels were not considered in the ontology or ERD
resistance_justification	Resistance justification	*None*
coverLevel	Level of coverage of Objective provided by Functional Requirement. Options: {No Cover, Partial Cover, Complete Cover}	*None* - Coverage level not considered, as it was assumed in the ontology and ERD that *Security Requirements* and other detailed treatment methods fully covered their associated *RiskActions*
upstream_justification	Justification of coverage by security functional requirements	*None*
downstream_justification	Unknown	n/a
<SecurityObjectiveCover>	Aspects (Risk, Contstraint, OSP, etc.) covered by Objective	There is only a partial mapping because the ERD just allows for risks to be addressed/covered (using *ProjectRiskAction* table) by Objectives (here noted as *RiskActions*). There is no direct mapping from Constraints to *RiskActions*
<TargetSystem>	Target system under analysis	*None* - There is no mappable concept that attempts to capture and structure data related to a system under investigation in this way
ID	Unique identifier	
presentation	General description of Target system	

Table 8.5: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (5)

EBIOS XML elements/attributes	Description	Corresponding concept(s), table(s) and fields & the justification
<TargetSystemDomainList>	Lists general Activity Domains included in system	
<ElementList>	Lists Essential Elements (*Function* /*Information*) included in system	
<ConstraintList>	Lists *Constraints* on System	
<RegulatoryReferenceList>	Lists the regulatory references of the organization. These link to the *RegulatoryRequirements*	
<EntityList>	Lists security study target entities	
<Theme>	Main Constraint Themes	There is no corresponding generic concept in the ontology to capture all types of constraint to a business, however a few specifics are mappable, namely *Security Budget* , *Security Policy* and *Business Policy* . At ERD-level, the Theme concept can be mapped to *TreatmentFactorType* table.
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Constraint type, e.g. Budgetary, Organisational, Financial, Technical, Environment constraints, etc	*TreatmentFactorType.treatment_factor_type* - during the mapping either new types can be added to mirror EBIOS needs, or if relevant, they can be mapped to either *Security Budget* , *Security Policy* or *Business Policy*
description	Description	*TreatmentFactorType.treatment_factor_type_description*
<Vulnerability>	Vulnerability - Characteristic of an entity that can constitute a weakness or flaw in terms of information systems security	The corresponding ontology and ERD-level concept is *Vulnerability* . Mapping is direct as the concepts have the same meanings
ID	Unique identifier	*None* - New identifiers are generated in the tool database
label	Description	*Vulnerability.vulnerability_name*
menace	*Menace* related to vulnerability	*None* - The ERD does not allow for a direct linking between a Vulnerability and a Threat. Instead the Risk concept is used
<EntityTypeList>	*EntityTypes* related to Vulnerability	*None* - The ERD does not allow for a direct linking between a Vulnerability and a related type of asset

Table 8.6: Mapping EBIOS XML to the ontology and SASaCS tool's ERD (6)

The final step before analysing the mappings achieved was **mapping the security knowledge** (knowledge defined as, a set of meaningful case data) embodied in EBIOS output to the SASaCS tool and its ERD. To facilitate this mapping, a similar approach to that employed by Fenz et al. [57] was adopted. This involved referencing a few real-life examples of case study data output from EBIOS software, then describing how those actual data could be mapped to the SASaCS tool's ERD. Two examples are used in this research. Both highlight concepts central to an EBIOS study. The first is presented below in Code Snippet 3, and 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>
```

Code Snippet 3: EBIOS representation of an eavesdropping menace

The XML excerpt describes EBIOS knowledge of an eavesdropping menace faced by a system in an RA study. To map this knowledge, the basic mappings from the previous step (outlined in the tables) were used. At the high level therefore, a *Menace* in EBIOS was mapped to a *Threat* in the tool and ontology. Considering the lower level, the *selected* attribute of the *Menace* XML element was assessed first. This attribute defines whether or not a menace is selected from the dataset of menaces and thus whether it applies to the current RA study. A 'true' value indicates 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 are 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. None of the *justification*, *descriptionMenaceElement*, or *potentiel* attributes had mappable fields in the ERD.

The *MenaceThemeList* sub-element lists the EBIOS attack method themes from which the *Menace* was deduced. The related *Theme* element (theme id 'Theme.1014431415703') in this case represents 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 allows 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* store the types of security criteria (or attributes in the tool), whereas the *ThreatSecurityAttribute* table and the database record created provide 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 maps 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 8.2 pulls together the mapping example and displays a screenshot of the actual SASaCS tool database records (in their respective tables) that would be created 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

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

SecurityAttribute

attribute_id	attribute_name
1	Confidentiality

ThreatSecurityAttribute

threat_security_attribute_id	threat_id	attribute_id
14	48	1

Figure 8.2: Mapped *Menace* data in the SASaCS database

The next knowledge mapping example was based on the EBIOS SecurityObjective element. The XML snippet in Code Snippet 4, describes the security objective defined in the RA study. This objective was to treat the risk associated with the menace identified in the prior example.

To start, a *SecurityObjective* in EBIOS corresponds to a *RiskAction* record in the ERD. Analysing the XML element's attributes, *label* which is the name of a security objective, maps to *RiskAction.risk_action_name* in the ERD. Also, *con-*

```
<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.
      either passive or active." resistance="3" resistance_justification="" coverLevel=
      "SecurityRequirementCover.1076860509716" upstream_justification=""
      downstream_justification="">
  <SecurityObjectiveCovers>
    <SecurityObjectiveCover ID="SecurityObjectiveCover.1245667560533" reference=
      "RiskScenario.1248601769338" type="Risk" />
  </SecurityObjectiveCovers>
</SecurityObjective>
```

Code Snippet 4: EBIOS representation of a security objective

tent, defined as a description of the objective, maps to *RiskAction.action_remarks*. None of the other attributes allowed for a mapping because no related fields existed in the 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 mark 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 an ERD database record in the *ProjectRiskAction* table. In detail, the associated risk first needed to be available in the *ProjectRisk* table. Then, the *risk_id* of that risk and the *risk_action_id* of the current *RiskAction* (that is, *SecurityObjective*) entry would be copied to create a linking record in the *ProjectRiskAction* table.

Lastly, and more at a general level, because the EBIOS *SecurityObjective* element does not define a type (that is, whether it is geared towards risk mitigation, acceptance and so on) some manual intervention was required to complete the mapping to the ERD *RiskAction* table and thus provide data for the respective record's *action_type* field. A screenshot of the actual records in their tables within the SASaCS database is shown in Figure 8.3.

RiskAction

risk_action_id	risk_action_name	action_type	action_remarks	project_id	..
38	Eavesdropping protection...	Mitigate	The organization must...	9	..

ProjectRisk Action

project_risk_action_id	project_risk_id	risk_action_id	coverage_level	coverage_level_details
22	3	38	*NULL*	*NULL*

Figure 8.3: Mapped *SecurityObjective* data in the SASaCS database

The next section reflects on the general mapping undertaken with EBIOS. This includes the mapping examples above and the detailed mapping tables presented previously.

8.4 Discussing EBIOS Mapping

The principal aim of conducting the mapping process was to evaluate the compatibility of the SASaCS tool and embodied ontology with existing RM/RA approaches. Having completed the mapping of EBIOS, it can be seen that a majority of the main concepts and elements could be mapped, both at ontology and ERD levels. This has provided promising evidence to support the case for ontology and ERD compatibility. Of equal interest however are the number of concepts and element attributes that proved challenging to map. This is because these might indicate shortcomings and thus areas for improvement of the tool and/or ontology. 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. While both the tool and ontology include concepts mappable to the security objective and security requirement, neither accommodates the security assurance concept. For EBIOS mapping, this fact highlights a weakness in this research's models (specifically their ability to capture all security aspects) and hence affects compatibility.

From a general perspective, however, because the assurance concept was not prevalent in the range of popular RM/RA methodologies examined in Chapter 5, 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' SecurityObjective and the

tool's RiskAction. At a high level, *SecurityObjective* and *SecurityAction* (*RiskAction*) are semantically similar and thus allowed for a seamless mapping of concepts. When assessed in detail, however, as seen in the knowledge mapping attempted, there are a few differences (related to attributes and elements) which complicate the process. The most notable of these is presented.

The first difference was mentioned in the security objective example from the previous section. This deals with the inability to identify an appropriate action type (mitigation, transference, acceptance and so on) for the corresponding *RiskAction* database record without manual intervention.

The next difference is centred 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 point because it exemplifies a direct relationship between a security objective and aspects that are not risks. This relationship was not conceived previously and therefore is not represented in the tool or ontology. To take an example, in the tool and ontology, a security action/risk action is conceived with the prime aim of treating a risk. Aspects such as those mentioned above including constraints, regulatory requirements and security rules/policies, are mainly viewed as constructs that influence the risk's treatment. This is as opposed to constructs which independently give rise to security actions or risk actions.

The final difference to be discussed is EBIOS' use of a *coverLevel* attribute within the Security objective. This attribute describes the degree to which security functional requirements cover their respective security objectives. In the tool and ontology, a similar concept was applied in the relationship between *Risk* and *SecurityAction* (*RiskAction*), however this was not mirrored in the *RiskAction* to *SecurityRequirement* relationship. This decision was based on the assumption that security requirements and other detailed treatment methods would naturally fully cover a security action. In retrospect, this assumption may be somewhat premature as the monitoring of a coverage level in EBIOS suggests that full coverage may not always be attained. A review of this reality may therefore be

warranted.

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 method for each database *RiskAction* 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 main positives and difficulties incurred in the mapping of EBIOS to the tool and ontology identified, the next section presents the mapping done from CORAS to the tool and ontology.

8.5 Mapping CORAS to the Tool and Ontology

The CORAS methodology for conducting security risk analysis was discussed in Chapter 6. CORAS is the product of an EU research project targeted towards creating a tool-supported methodology for model-based risk assessment of security-critical systems [72]. To guide the mapping process in this section, the technique used in mapping EBIOS (that is, Fenz et al. [57]) was employed again, beginning with a **knowledge base analysis** of CORAS.

The CORAS method is accompanied by various conceptual models (see [46, 72]) which enabled quick identification and definition of its main concepts. These included: *assets* and *vulnerabilities* which have standard definitions; an *unwanted incident* which is an event that reduces the value of assets; a *threat*, defined as

a potential cause of an unwanted incident, this encompassed the human, or non-human cause; a *risk*, that is, the chance of occurrence of an unwanted incident; *risk value*, which is the value of a risk as derived from the likelihood and consequence of an unwanted incident; and lastly, *treatment*, which defines a means that has the objective of reducing a risk's assigned risk value. There are other concepts such as context, asset value, security policy and security requirement. These however were excluded as they were not common across CORAS models nor were they all (except asset value) used explicitly in the CORAS software (and thus its machine-readable output).

The **mapping of concepts and relations** from CORAS to the SASaCS tool and ontology was the next step. As with the EBIOS analysis, two levels of mapping were carried out. At the first level, the mapping was from the conceptual CORAS model (specifically the diagram in den Braber et al. [46]) to the ontology. This CORAS model was preferred instead of the XML elements from the machine-readable CORAS software output because it allowed for a more meaningful high-level mapping that still embodied the main concepts of the CORAS tool. The mapping of concepts is presented in Figure 8.4—boxes with dashed and solid lines represent CORAS and ontology concepts respectively. Specifics of the mapping are presented later in the detailed tables.

One noteworthy aspect is the mapping of the CORAS model *Threat* concept to both *Threat Agent* and *Threat* in the proposed ontology. This was done because their definition of a *Threat* itself ([46]) covered both ontology concepts. *Unwanted Incident* is another interesting concept as this and part of the meaning captured in *Threat* can be mapped to the single *Threat* concept in the ontology. More detail on these aspects is presented in the knowledge mapping and mapping reflections sections.

At the second level of mapping, emphasis was on the lower-level. This therefore included the CORAS XML elements and their corresponding fields in the SASaCS tool's ERD. After an analysis of CORAS software output, related ERD fields were identified where possible. The description of elements, the re-

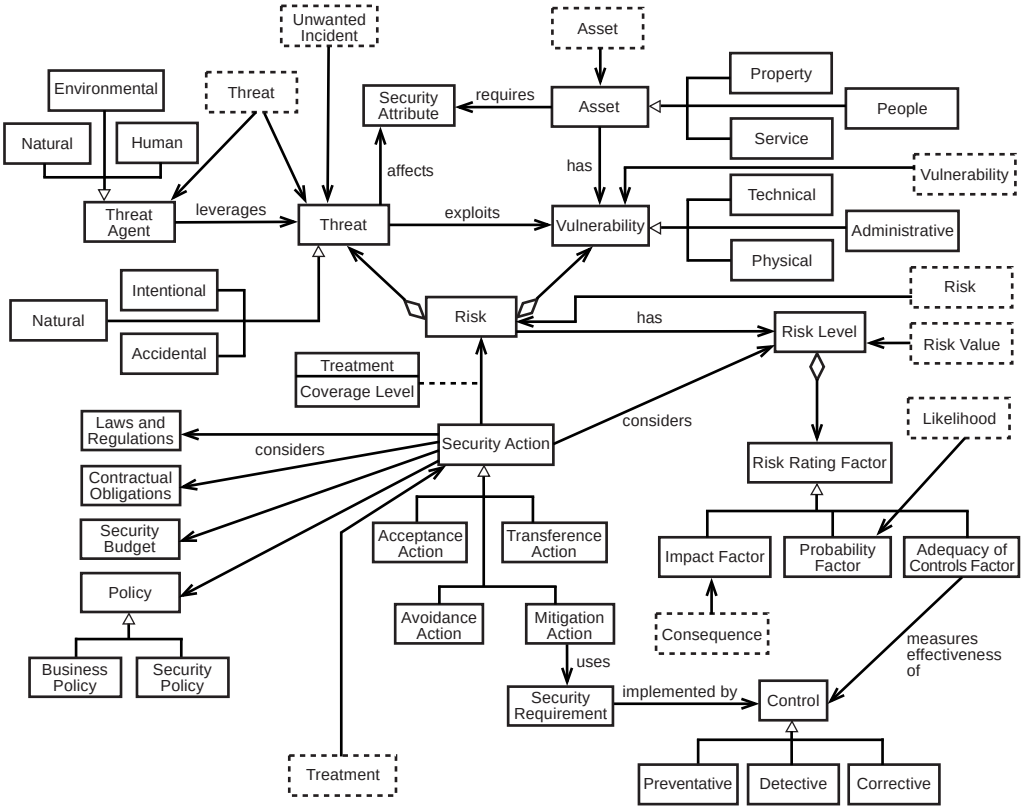


Figure 8.4: Mapping CORAS concepts to proposed ontology

sulting mappings and any justifications are shown in Tables 8.7, 8.8, 8.9, 8.10 and 8.11.

CORAS table types/column ids	Description	Corresponding concept(s), table(s) and fields & the justification
Asset Table	Table with list of assets. An asset is a physical or conceptual item in the system or organization which has value to the client	The corresponding ontology concept is *Asset* . *Asset* tables provide a mapping at the ERD level
assetId	Unique identifier for an asset	*None* - New identifiers are generated in the tool database
assetDescription	Description of asset	*Asset.asset_name*
assetCategory	Categorization of an asset. Options: {Human, Physical, Information, Organisational, Law and Regulation, Software, Other}	*Asset.asset_type* - Mapping could either be done from the CORAS categories to the types used in the tool (i.e. map "Human" to "People", and "Physical, Information, Organisational, Law and Regulation, Software" to "Property"), or CORAS categories could be added (as new records) to the tool via *Asset.asset_type* field in the *Asset* table
assetValue	Value or worth assigned to an asset. Usually based on definitions made in the Value Definition Table	*None* - Asset value not considered
Consequence and Frequency Table	Table with list of risks and the respective consequence and frequency values for each	The main concepts included in this CORAS table correspond to ontology concepts *Risk* , *ImpactFactor* , and *ProbabilityFactor* . At the ERD level, a mapping is possible to *Risk* , *ProjectRisk* and *RiskEstimate* tables. These aspects across both models contribute towards determining an appropriate risk level/value for a risk. CORAS data from this table is used to create a *Risk* record in the tool database and an appropriate *RiskEstimate* record
riskId	Unique identifier for a risk	*Risk.risk_id* - This would refer to the risk that is being valued
assetId	Asset which relates to risk	*Risk.asset_id* - Asset relating to the specific *Risk* identified by *Risk.risk_id*
incident	The event that may harm or reduce the value of the current asset	*Risk.threat_id* - The *incident* is used to locate the respective *threat_name* record in the *Threat* table. Once identified, the *threat_id* of that threat record is copied to the *Risk* table's *threat_id* field to aid in the creation of the *Risk* record
consequenceValue	The impact of an unwanted incident on the asset in terms of reduction or loss of asset value e.g. Insignificant, Moderate, Major	At ontology level, *ImpactFactor* relates as both attempt to capture the level of impact/consequence that results from an unwanted incident or threat occurring. For the ERD level, the *RiskEstimate.impact* field allows for a mapping, however companies intending to use the tool for comparison later, would first have to define the common set of valuation metrics then translate their individual metrics to the common metrics. To complete the mapping, the *ProjectRisk* table would also be required as this forms the link between *Risk* and *RiskEstimate*
frequencyValue	The likelihood or probability of the current unwanted incident occurring e.g. Rare, Possible, Certain	At ontology level, *ProbabilityFactor* relates as both attempt to capture the level of probability of an unwanted incident or risk occurring. For the ERD level, the *RiskEstimate.probability* field allows for a mapping, however companies intending to use the tool for comparison later, would first have to define the common set of valuation metrics then translate their individual metrics to the common metrics. To complete the mapping, the *ProjectRisk* table would also be required as this forms the link between *Risk* and *RiskEstimate*

Table 8.7: Mapping CORAS XML to the ontology and SASaCS tool's ERD (1)

CORAS table types/column ids	Description	Corresponding concept(s), table(s) and fields & the justification
HazOp Table	Table with output from a HazOp (HAZard and OPerability) analysis. Each entry identifies general threat scenarios	Conceptually, the HazOp table identifies threats faced by a system and therefore it is mapped to *Threat* at both the ontology and ERD levels. In practice however, this mapping is not done as it is assumed (based on literature) that the significant threats identified in this table are carried forward to the Scenario Table, and the mapping is conducted from there
hazopId	Unique identifier for hazard	
assetId	Asset relating to scenario, which is threatened	
reference	Reference to where threat is identified e.g. message in a sequence diagram	
hazopGuideword	Guide word used to direct and stimulate process	
hazopAttribute	Attribute of the component which the guide word relates to, e.g. storage capacity, bandwidth	
scenario	Description of the threat scenario, i.e. how the threat leads to the unwanted incident, either as text or as a reference to e.g. a sequence diagram	
FMEA Table	Table with output from a Failure Modes and Effects Analysis (FMEA). Each entry identifies general unwanted incidents	Conceptually, the FMEA table identifies threats faced by a system and therefore it is mapped to *Threat* at both the ontology and ERD levels. In practice however, this mapping is not done as it is assumed (based on literature) that the significant threats identified in this table are carried forward to the Scenario Table, and the mapping is conducted from there
fmeaId	Unique identifier for failure	
assetId	Asset whose value is reduced by the incident	
reference	Reference to where failure is identified e.g. message in a sequence diagram	
fmeaFailureMode	The way in which the component fails e.g. errors or defects in a process, design, or item	
incident	The event that may harm or reduce the value of the current asset	
frequencyValue	Frequency value assigned to the risk	

Table 8.8: Mapping CORAS XML to the ontology and SASaCS tool's ERD (2)

CORAS table types/column ids	Description	Corresponding concept(s), table(s) and fields & the justification
Risk Category Table	Table with list of risk categories. This allows risks to be grouped for treatment	At the ontology level, there is no corresponding concept. At the ERD level, the *ProjectRiskAction* table allows for risks to be directly listed/grouped according to their treatment method i.e. *RiskAction* . For an actual mapping however, risk categories may be simply expanded and each risk viewed on an individual basis. This may be done manually before mapping
riskCategoryId	Unique identifier for this risk category	*None*
riskIds	List of riskId (of risks) that belong to risk category	*None*
Risk Evaluation Table	Table with list of risks or risk categories, and their deduced priorities	The related ontology concept is *RiskLevel* as this concept also focuses on the valuation of risks. The ERD-level mapping from this general CORAS concept is to the *RiskEstimate* table. Because some relevant information on each risk has already been defined in the Consequence and Frequency Table, that previous mapping (to some fields in the *RiskEstimate* table) has to be considered to ensure the *riskValue* data in this table is mapped to the appropriate risk in the tool's *RiskEstimate* table. The *riskId* field in the Consequence and Frequency, and Risk Evaluation Table is key in doing this
riskOrCategoryId	The risk or risk category being evaluated	*RiskEstimate.project_risk_id* - The risk's unique id is used. This assumes that risk categories are not used and each individual risk is listed
riskValue	The level or value of a risk as derived from the likelihood and the consequence of an unwanted incident e.g. Low, Medium, High, which may come from the Consequence and Frequency Table	*RiskEstimate.risk_level* - This allows for a good mapping as CORAS and the ontology/tool define a summary risk level/value from aspects such as frequency, and impact. Companies intending to use the tool for comparison later, would first have to define the common set of valuation metrics for risk level/value then translate their individual metrics to the common metrics
Risk Evaluation Criteria Table	Table with list of risk evaluation criteria. This criteria specifies what level of risk the client is willing to accept	*None* - There is no related concept at ontology or ERD level that allows a mapping of this data
criteriaId	Unique identifier for risk evaluation criteria	
criteriaDef	Specification of the risk reduction in terms of frequency, consequence, or risk level and action to be taken when criteria is met (e.g. accept, monitor, treat)	
criteriaDescription	Description of risk evaluation criteria	
criteriaAssets	List of assets the risk evaluation criteria are applied for	
Scenario Table	Table with list threat scenarios. A threat scenario is a scenario which describes how a threat exploits a vulnerability leading to an unwanted incident	The corresponding ontology concepts are *Threat* and *ThreatAgent*. The *threat scenario* and *unwanted incident* concepts in CORAS are considered as two parts (or a lower-level breakdown) of a *Threat* concept in the ontology and ERD. These two concepts therefore map to the single *Threat* concept
scenarioId	Unique identifier for scenario	*None* - The respective - but not mappable - concept is *Threat.threat_id*

Table 8.9: Mapping CORAS XML to the ontology and SASaCS tool's ERD (3)

CORAS table types/column ids	Description	Corresponding concept(s), table(s) and fields & the justification
assetId	Asset which relates to scenario	*Risk.asset_id* - A respective *Risk* record would have had to be created or a mapping would not be possible for this attribute. This is because the ERD does not facilitate a direct link between *Threat* and *Asset*
reference	Reference to where risk is identified e.g. message in a sequence diagram	*None* - There is no allowance for these type of references to be made
threat	Threat which threatens the asset e.g. virus, hacker, unfaithful employee	*Threat.agent_id* - CORAS *threat,* in this regard maps to the ontology/ERD *ThreatAgent.* As such, a record would have to be created with the appropriate *ThreatAgent.agent_name* for the mapping
vulnerability	Vulnerability which may be exploited by a threat to lead to an unwanted incident	*Risk.vulnerability_id* - To find the respective *vulnerability_id* , the records in the ERD *Vulnerability* table would be searched using the CORAS *vulnerability* field's data. Once found, the *vulnerability_id* would be copied to the necessary *risk_id* field in the *Risk* record. This mapping would have to be done after the respective *Risk* record was created
incident	The event that may harm or reduce the value of the current asset	*Threat.threat_name* - The *scenario* and *incident* fields in CORAS are considered as two parts (or a lower-level breakdown) of a *Threat* concept in the ontology and ERD. These two concepts therefore map to the single *Threat* concept
scenario	Description of the threat scenario	Same as above. However, *incident* is used as the main concept and *scenario* the concept in detail
Treatment Identification Table	Table with list of treatments. A treatment is a means that is directed towards one or more risks with the objective of dealing with risk value	At the ontology level, *RiskAction* is the mappable concept as both define how to treat or handle risks. For the ERD level, the *RiskAction,* and *ProjectRiskAction* are used. One noteworthy observation is that the CORAS Treatment Identification Table lists all the possible treatments, whereas the tool and ERD only list the final treatments chosen to handle risks. A semi-manual process is needed therefore where a user identifies which treatments were actually chosen to handle a risk
treatmentId	Unique identifier for treatment	*None* - New identifiers are generated in the tool database
riskOrCategoryId	Risk or risk category being treated	*ProjectRiskAction.project_risk_id* - The risk's unique id is used - this assumes that the respective ProjectRisk record has been created to match the Risk record. Secondly, an assumption is made that risk categories are not used and each individual risk is listed
treatmentStrategy	Generic way in which risk may be treated. Options: {Reduce consequence, Reduce frequency, Transfer, Avoid, Retain}	*RiskAction.action_type* - CORAS treatment strategies map to the *RiskAction* types in the ontology and the ERD. Specifically, "Reduce consequence, and Reduce frequency" map to "Mitigate", "Transfer" maps to "Transfer", "Avoid" maps to "Avoid", and "Retain" maps to "Accept"
treatmentDescription	Description of treatment used to reduce risk level	*RiskAction.risk_action_name*
treatmentReferences	Reference to where treatment is documented e.g. treatment diagrams	*None* - There is no allowance for these type of references to be made
Treatment Evaluation Table	Table with list of treatments and evaluation data on their usefulness	*None* - No mapping is possible as the tool only accommodates chosen risk treatments as opposed to all risk treatments and their evaluation data
treatmentId	Unique identifier for treatment	

Table 8.10: Mapping CORAS XML to the ontology and SASaCS tool's ERD (4)

CORAS table types/column ids	Description	Corresponding concept(s), table(s) and fields & the justification
treatmentRiskReduction	Specification of the risk reduction in terms of frequency, consequence, or risk level	
treatmentMeetsRiskCriteria	Whether or not risk evaluation criteria are met. Options: {Yes, No, Partly, N/A}	
treatmentBenefit	Description of the benefits of this treatment for this risk	
treatmentCost	The cost of introducing this treatment for this risk	
treatmentPriority	The priority assigned to this treatment	
Value Definition Table	Table with list of elements used in the risk analysis and the CORAS system and ways in which they are to be valued/measured	*None* - This CORAS table is mainly used to give setup, or prerequisite data
valueType	Type of element to be valued e.g. Asset, Frequency, Consequence, Risk value	
valueDomain	Metric of valuation/measurement	
valueValues	Allowed set of values, (accommodates both qualitative and quantitative values)	
valueDescription	Description of the allowed values	
Vulnerability Table	Table with list vulnerabilities identified. A vulnerability is a weakness, flaw or deficiency by the system that opens for a threat to harm or reduce the value of assets	The corresponding ontology and ERD-level concept is *Vulnerability* . Mapping is direct as the concepts have the same meanings
vulnerabilityId	Unique identifier for a vulnerability	*None* - New identifiers are generated in the tool database
assetId	Asset which relates to scenario	*Risk.asset_id* - Asset relating to the specific *Vulnerability* . As an ERD *Risk* record would need to be created first, this field need not be mapped as the Consequence and Frequency Table also contains an assetId field which identifies the relevant asset
vulnerabilityQuestion	The means used to identify the vulnerability e.g. predefined list	*None*
vulnerability	A description of the vulnerability that may be exploited by a threat to lead to the unwanted incident	*Vulnerability.vulnerability_name*
scenario	Description of the threat scenario, i.e. how the vulnerability may be exploited to lead to an incident, either as text or as a reference to e.g. a sequence diagram	No mapping is done here because a mapping is also possible from the Scenarios Table

Table 8.11: Mapping CORAS XML to the ontology and SASaCS tool's ERD (5)

To conduct the **mapping of security knowledge** step, a scenario was prepared in the CORAS software and then exported to its XML format. The software's Help section and a case study by Fu et al. [64] which used CORAS, guided scenario preparation. The developed case produced practical examples of CORAS software output and provided data to be used for mapping to the SASaCS tool's ERD. Default settings were used in the CORAS software and customization of settings was kept at a minimum to maintain an objective mapping. Two examples were chosen which reference crucial stages within an RM process. These cover threat scenarios and risk identification and estimation. The mapping commences with the threat scenario XML in Code Snippet 5.

```
<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>
```

Code Snippet 5: CORAS representation of a threat scenario

Code Snippet 5 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 the SASaCS 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 8.5 gives a visual presentation of the mapped data as it is captured in the SASaCS 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 customer...		62

Figure 8.5: Mapped *Scenario* data in the SASaCS database

The second example of knowledge mapping uses the CORAS Consequence and Frequency Table. 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 in Code Snippet 6.

To map the risk defined in the <row> element in Code Snippet 6, the ERD's *Risk* and *ProjectRisk* tables were employed. After creating a new *Risk* database record, the CORAS *riskId* element's data were copied/mapped to the ERD *Risk.risk_id* field. For the CORAS row's *assetId*, the respective asset's

```
<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>
```

Code Snippet 6: CORAS representation of a risk

unique identifier (*asset_id* in the ERD *Asset* table) for 'Network1' was copied to *Risk.asset_id* field. 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. This sets up the foreign key relationship between tables.

To complete the ERD *Risk* record, the incident's respective *vulnerability* from the Scenario Table was used. Once the incident's vulnerability was found (recall 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 in mapping the security knowledge was transferring the consequence and frequency data. Assuming that metrics (that is, allowed values such as High or Moderate) for these factors were set to be the same in both CORAS and the SASaCS tool (note that metrics can be added to the tool using *PrioritizationScheme* table), the 'moderate' consequence in CORAS mapped to 'moderate' value for the *impact* field in the tool's *RiskEstimate* table. The 'likely' frequency then mapped to the 'likely' value for the *probability* field in *RiskEstimate*.

For the mapping above to be conducted, a *ProjectRisk* database record was required first. From the ERD as displayed in Appendix A, one can see 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 *project_risk_id* generated was copied to a new *RiskEstimate* record. The relevant

impact and *probability* values were then copied to that new *RiskEstimate* record also. As before, a screenshot is presented in Figure 8.6 to show the resulting mappings in SASaCS.

Risk

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

ProjectRisk

project_risk_id	project_id	risk_id	asset_details	agent_details	threat_details	...
88	68	RSK-1				

Prioritization Scheme

priority_id	priority_name	priority_description	rating_factor_type	project_id
7	likely	Possible that the...	probability	68
33	moderate	(1) May result in...	impact	68

RiskEstimate

risk_estimate_id	probability	impact	project_risk_id	probability_remarks	...
29	7	33	88		

Figure 8.6: Mapped *Consequence and Frequency* table data in SASaCS database

In keeping with the goal of this section, the mapping of the risks and risk values above focus purely on the mapping of knowledge to the tool's ERD. If looking towards using the mapped information for comparison of security actions later however, mapping companies must previously synchronize some of this information. Synchronization would be required on elements such as risks and risk ids to be used (recall that tool comparison is made largely based on common risks), and also the metrics for risk valuation. The latter of these aspects ensures that entities use similar valuation schemes and agree on the meanings of individual metrics.

8.6 Discussing CORAS Mapping

The mappings of CORAS and EBIOS to the SASaCS tool and embodied ontology provide another means of evaluating this research. In the CORAS mapping above, a majority of the high- and low-level core concepts and elements allowed for a translation across the different software tools. This might not be considered surprising as the CORAS method was used to aid in the creation of the ontology proposed. The reason for the similarity however was not thought to be due to the exact concepts from CORAS simply being copied to create the ontology.

Instead, this congruence was attributed to the very basic, standard RA concepts emphasized in CORAS which are viewed as equally critical in any RM/RA-based ontology. Generally, this mapping analysis therefore also supplied favourable evidence towards supporting compatibility of the tool and ontology. Although a good mapping was attained from CORAS, a few difficulties were incurred. The most significant of these are presented.

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 the notion of a threat might be understood in different ways. In the mapping from the previous section, a tool/ontology's *Threat* was stretched over two CORAS elements. These included threat *scenario* (defined as how the threat leads to or causes an unwanted incident) but primarily, the unwanted *incident* concept (which is the actual event that may harm or reduce the value of assets). Recall that in the CORAS Scenario Table mapping, these two elements are concatenated to form ERD's *Threat.threat_name*.

However, as the definitions of *Threat* (in tool/ontology) and unwanted *incident* (in CORAS) are largely the same, another way to do the mapping could be to map these two concepts/fields and discard data in the CORAS threat *scenario* element. The disadvantage of this would be losing data which provide more descriptive information on what actions (or causes) constitute a threat to an asset.

The current mapping choice for handling the threat *scenario* and unwanted *incident* elements (that is, appending them and then mapping to the *Threat.threat_name*) also has its shortcomings. These were discussed briefly in the knowledge mapping example and deal with the fact that if there are multiple threat *scenarios* that culminate in one unwanted *incident* (and all are listed as separate rows in the CORAS Scenario Table), a straight forward mapping would not be possible. In this case, the options would be to either (i) discard the *scenario* data during mapping and only use the *incident* data (this was feasible because the *incident* data are what link to a specific *Risk* in the Consequence and

Frequency Table) or (ii) to append the *scenario* and *incident* data and view each row in the Scenario Table as a separate ERD *Threat*. The latter of these options would complicate matters further as it would introduce a many-(threats)-to-one (risk) relation later in mapping that, at this moment, cannot be addressed by the tool's ERD.

In summary, the differences in threat representation between CORAS and the tool and ontology have highlighted new perspectives which slightly complicate mapping. Some of these however might easily be addressed through predefined mapping rules set up by companies, users or SASaCS tool designers. Examples of mapping rules for the *incident/scenario* versus *threat* could include only using data from the *incident* element to map to ERD's *Threat* record, or, each scenario has a different incident (and thus a different *Risk*) therefore *incident* and *scenario* data can be appended and then mapped to the ERD's *Threat* record.

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. As stated in the tables in Section 8.5, such a grouping of risks does not exist in the SASaCS tool's ERD. To enable for mapping therefore, risk categories needed to be broken down such that each risk is 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, users 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 at this point was because the SASaCS tool only accommodates actual treatments which were chosen to address a risk. Therefore, the treatment evaluation process documented in CORAS was taken to be complete with regards to the tool.

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 showed a chosen treatment. The *treatment-Priority* element in the Treatment Evaluation Table was considered to aid in mapping, however, because there was no predefined hierarchy of metrics (for example High, Medium or 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 transferred.

At this stage, this chapter has presented evidence to support an adequate mapping from the output (particularly the software's machine-readable output) from existing RM/RA approaches such as CORAS and EBIOS, to the SASaCS tool and underlying ontology proposed in this research. Taken broadly, this therefore fulfils the evaluation objective defined in Section 8.2 and affirms a reasonable level of tool and ontology compatibility. The next section reflects on the map-

ping process and very briefly compares the mappings undertaken. The goal was to identify any common weaknesses in the tool, situations in which mapping might be easier or other peculiarities that might lead to specific uses or future updates in the tool/ontology being necessary.

8.7 Reflecting on Mappings, the Tool and the Ontology

Both EBIOS and CORAS proved useful and insightful methods which aided in this research's evaluation. 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 [42]. 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. [57] 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 was 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 SASaCS tool/ontology to these or any other RM/RA methods. Other methods assessed in Chapter 6 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 as follows. (i) Allowing a *Security action/Risk action* to directly address aspects other than *Risks*, for example, laws and regulations, technical constraints and so on. The alternate term of 'Security action' would therefore be explicitly preferred to 'Risk action' from now on, and its meaning extended to '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)). (iv) Providing a 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 an RM/RA methodology, the SASaCS 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 the extensions, a first draft of an updated ontology is displayed in Figure 8.7. A draft of that ontology's respective ERD is also shown in Appendix A. As these are drafts, further tests will be needed to verify their rigour, identify any accompanying problems and make any other necessary updates. Readers should note that these new drafts do not replace current work as a basis for forthcoming chapters. When appropriate however, at times they are mentioned and debated.

One 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 8.7, or more detailed and thus focusing on data privacy, specific system types, or even organizational limitations.

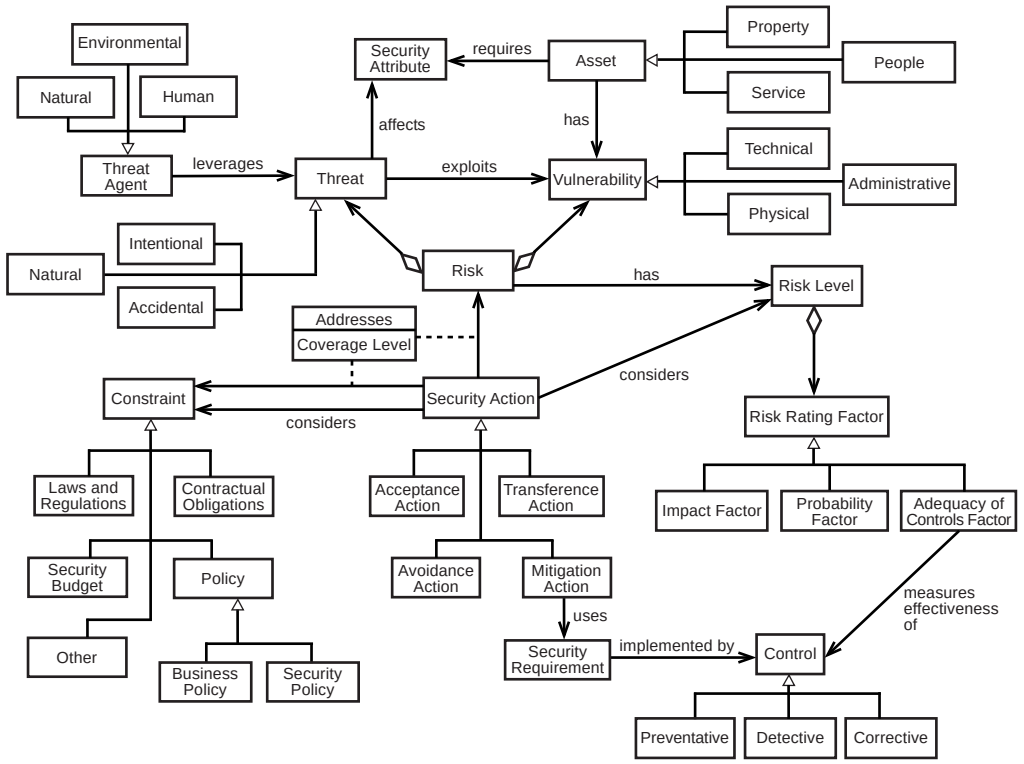


Figure 8.7: Draft of the new ontology

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 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.

8.8 Case Study

The last core section of this chapter rounds up the compatibility evaluation using a full case study. In previous sections a very detailed discourse was presented. Now the aim is to put that and other aspects of the Solution Model into a more real-

world context. In addition to further supporting the feasibility of this research's proposals, this would enable for a more thorough evaluation of the Model's proposals as they progressed from the initial Central Risk Catalogue to final SASaCS output. Readers are reminded that it is the original ontology and ERD that are used (and not the new drafts mentioned at the end of the previous section).

In the interest of continuity, the scenario presented in Chapter 4 was used *as a basis* for this section's case. It therefore begins much like that chapter where `Buyer` and `Supplier` are entering into a particular business situation. Unlike that chapter, however, to be consistent with the aims of this section and chapter, EBIOS and CORAS are used by entities for their RM purposes. Specifically, to analyse risk and determine security actions, `Buyer` used EBIOS and its software, whereas `Supplier` employed CORAS and its supporting tool. Finally, the types of output (security actions/requirements, influential factors and so on) from each RM method are taken to be generally consistent with those shown in Chapter 4. For example, in this case, `Buyer` does have a security action (a requirement) for a specific risk that is influenced by the risk's (monetary) impact, the Sarbanes-Oxley Act and a company security policy.

According to the Solution Model, regardless of the RM/RA method used, the starting point of the scenario should be a common risks base or catalogue. This point however is where one of the first difficulties surfaced. When the Model was first conceived it was assumed that the transferring of common risk data to RM/RA analyses would be done manually. During the completion of this study however, such a process actually proved somewhat tedious. This is especially in terms of accurate and consistent mapping of data from the common risks catalogue to the RM/RA methods and software.

If there was a risk to the confidentiality of WS messages in the Risk Catalogue system therefore, the problem was how that data and the related data on vulnerabilities, threats and assets, could be quickly, accurately and consistently entered into the RM/RA approaches and their software. Figure 8.8 depicts the area of focus in the 'Solution Model in action' diagram from Section 5.4.

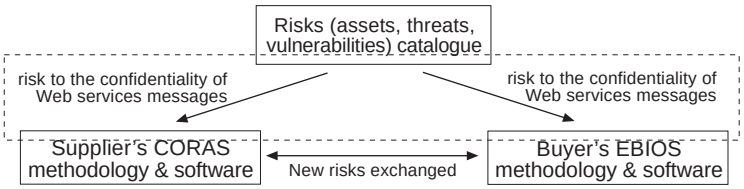


Figure 8.8: Area of focus in 'Solution Model in action'

Possibly the best solution to this problem resides in the automated mapping of data from the Central Risk Catalogue to the RM/RA method software, in this case represented by EBIOS and CORAS tools (used by `Buyer` and `Supplier` respectively). Two options were identified by which this could occur. First, adding an export capability to the Central Risk Catalogue system, which could output data on risks in the machine-readable formats of common RM/RA approach software. This is beneficial because it would be a central point where numerous RM/RA software formats could be generated. Also, it could take advantage of the 'Import' and 'Open File/Project' functionalities which are standard in a number of RM/RA software. For example, both CORAS and EBIOS tools have these capabilities.

One caveat noticed when assessing the Risk Catalogue export capability option is that unique identification numbers (IDs) for elements (for example, *Menace* IDs in EBIOS or *risk-analysis-result* IDs in CORAS) generated by the Central system might conflict with the same element IDs generated by the actual software at each company. There would therefore need to be some agreed allotment of ID ranges for the Catalogue-based option to function properly.

The second alternative suggests a more decentralized implementation where extensions could be added to the RM/RA software systems to enable them to read in and process Risk Catalogue system data. This would avoid the problem of conflicting IDs, but introduces the need to access, understand and edit various software. For this case, EBIOS and CORAS are good candidates in this regard as both are open source implementations (see [42] and [46] respectively).

Apart from the programming that would be necessary in both options, there is the question of how to map Risk Catalogue system data to EBIOS and

CORAS. This however can be largely addressed by reversing the mapping tables in Section 8.3 and 8.5 because the SASaCS tool's ERD is not dissimilar to that of the Risk Catalogue system. Essentially, one would now be going from SASaCS database records to EBIOS and CORAS software XML formats.

Having briefly digressed from the case study to discuss how transferring data from the shared risks catalogue could be addressed, the focus resumes at the RM/RA software stage (in Figure 8.8). This relates to the bottom two boxes. After `Supplier` and `Buyer` have agreed the risks to be used, they conduct their individual analyses. This generally encompasses the processes of risk estimation, risk evaluation and treatments. Code Snippet 7 and Code Snippet 8 give an initial idea of the risks data generated by each entity. This and most of the following examples are based around the risk to the integrity and confidentiality of Web services messages (thus, `Buyer` security requirement # 1) from Chapter 4. Hereafter, this is referred to simply as `Risk101`. From the code snippets, one can see exactly how different the representations of the same risk may be from company to company. As before, the + sign in the code indicates that there is additional data which is not displayed/expanded.

```
 <Risk ID="RiskScenario.1252746098288" label="Risk101" menace="Menace.1050437920519"
     description="The integrity and confidentiality of data in a Web services' message
     (in transit) is compromised" sof="AttackPotential.1070307963407" coverLevel=
     "SecurityObjectiveCoverValue.1078561424090" ... >
+  <ScenarioPotentiality potentiality="Potentiality.1076645892186">
 </Risk>
```

Code Snippet 7: EBIOS (`Buyer`) representation of the risk

```
<row>
  <cell columnId="riskId">Risk101</cell>
  <cell columnId="assetId">WSMessage</cell>
  <cell columnId="incident">Eavesdropping and tampering with data in a Web services'
     message (in transit)</cell>
  <cell columnId="consequenceValue">Medium</cell>
  <cell columnId="frequencyValue">Low</cell>
  <cell columnId="scenario"/>
</row>
```

Code Snippet 8: CORAS (`Supplier`) representation of the risk

With the RM/RA methodologies at each business complete, the next step was mapping the output data from `Buyer` and `Supplier` to the SASaCS tool.

This process was covered in detail in Sections 8.3 to 8.7 and therefore is not analysed in depth here. From a case study perspective however, one intriguing additional observation was made—that is, although RM/RA methods did not accommodate certain data expected by SASaCS, it did not mean that the data were not present in companies' considerations.

In the CORAS software output shown in Code Snippet 9 for example, it is apparent that a limited security budget influenced `Supplier`'s treatment strategy decision (see *treatmentDescription columnId*). Any automated mapping to SASaCS therefore should ideally capture these data as a unique Risk Treatment factor. This however was not possible because the machine-readable output of CORAS does not distinctly define such aspects in its XML structure. Here it is just in plain text.

```
<row>
  <cell columnId="treatmentId">TRT101</cell>
  <cell columnId="riskOrCategoryId">Risk101</cell>
  <cell columnId="treatmentStrategy">Retain</cell>
  <cell columnId="treatmentDescription">The unlikeliness of this risk and a limited
     security budget are the reasons for risk acceptance</cell>
  <cell columnId="treatmentReferences">Threat_Analysis09.doc</cell>
</row>
```

Code Snippet 9: CORAS representation of a risk treatment

A similar situation is present in `Buyer`'s EBIOS output regarding risk estimation. In this case, `Buyer` has used EBIOS to prioritize risks, however, because their technique is so elaborate it does not allow for a clear and reliable automated mapping to the risk level concepts in SASaCS.

To tackle these mapping issues a few other techniques were assessed but manual mapping proved to be the only dependable solution. This mapping involved noting the type of data requested by SASaCS (such as influential security policies or budgetary limitations) and using the tool's data entry screens to manually enter that data. This was easily done in this case through the creation of a *TreatmentFactor* record in SASaCS and then linking that record to the respective risk treatment, formally the *RiskAction* database record.

Regarding the manual risk estimation and prioritization mapping needed

for EBIOS mapping, a level of subjectivity would be introduced as users seek to map values in their analyses to the risk levels expected in SASaCS. To compensate for this subjectivity, detailed justifications and descriptions of chosen risk levels should be provided by parties. This information would be entered in the respective *RiskEstimate* database record's *risk_level_remarks*, *probability_remarks*, *impact_remarks* and *adequacy_of_controls_remarks* fields. Generally, at the end of mapping, companies' personnel should browse screens in the tool to ensure that all the required information has been transferred.

The next step in the case study was encoding each business' mapped data (now in SASaCS's database) to SADML documents. This process went without error. In Code Snippet 10, an example of the security risk under examination (`Risk101`) is presented. The marked-up risk data has the same basis across documents due to the use of the shared risks base in the beginning. SADML provides the common structure, tags, elements and attribute names. Different companies may add varying comments or descriptions however. The specific code in Snippet 10 is from `Buyer`.

```
<risk id="RISK101"><threats>
  <threat>
    <name>Eavesdropping and tampering with data in a Web services' message (in transit)</name>
    <threatAgent><agentName>Malicious party</agentName><comment /></threatAgent>
    <comment />
  </threat></threats><vulnerabilities>
  <vulnerability>
    <name>Circulating information in inappropriately secured formats</name>
    <asset>
      <dtype>property:data</dtype>
      <assetName>web service message</assetName>
      <comment>The data carried in the message is the key aspect</comment>
    </asset>
    <comment />
  </vulnerability></vulnerabilities>
  <riskComment>Violation of confidentiality using eavesdropping</riskComment>
  ...
</risk>
```

Code Snippet 10: SADML representation of the highlighted risk

The real difference in SADML documents across `Buyer` and `Supplier` is visible when it comes to the treatment of `Risk101`. As noted in Chapter 4, `Buyer` aims to mitigate this risk while `Supplier` accepts it. SADML Code Snippet 11 shows this and the respective treatment factors. On the left hand side is `Buyer`'s

document and on the right, `Supplier`'s. The + sign indicates that there is additional data which is not displayed here.

```
 <mitigationAction>
  <name>Protect against eavesdropping
    on Web service messages being
    transmitted between partners</name>
  <details>The organization must take
    measures to ensure there is no
    eavesdropping on data being
    transmitted between Web services
    across business parties.</details>
  <risks>
+  <risk id="RISK101">
  </risks>
+ <lawAndRegulationRefs>
  <contractualObligationRefs />
  <businessPolicyRefs />
+ <securityPolicyRefs>
  <securityBudgetRefs />
+ <securityRequirementRefs>
 </mitigationAction>
```

```
 <acceptanceAction>
  <name>The unlikeliness of this risk and
    a limited security budget are the
    reasons for risk acceptance</name>
  <details>Threat_Analysis09.doc</details>
  <risks>
+  <risk id="RISK101">
  </risks>
  <lawAndRegulationRefs />
  <contractualObligationRefs />
  <businessPolicyRefs />
  <securityPolicyRefs>
+ <securityBudgetRefs>
+ <riskActionImplementationDetailRefs />
 </acceptanceAction>
```

Code Snippet 11: SADML representations of companies' risk treatment choices

When compared to the original output from EBIOS and CORAS, one can appreciate the use of the standard format supplied by SADML. In this respect, SADML provides a bridge between different RM/RA methods and their software systems, that can then be used as a platform to compare high-level security actions across enterprises. It is worth noting that the benefits possible with SADML are largely due to its foundation in the well-researched ontology from Chapter 6.

With all stages in the case process completed, Figure 8.9 displays the output of the SASaCS comparison task presented to personnel at `Buyer` and `Supplier`. `Buyer` and `Supplier` are used instead of Company1 and Company2, and the screenshot is slightly modified to ease readability. This output covers the same information included at the end of the Requirements phase in Chapter 4's scenario. The real benefit associated with this output is the automation of various of the preceding steps taken to reach this point. These included (i) gathering data from RM/RA approaches (such as EBIOS and CORAS), albeit in a semi-automated fashion, (ii) allowing for influential factors in risk treatment to be defined in the initial stages, and finally, (iii) matching the security actions and requirements of companies based on risks.

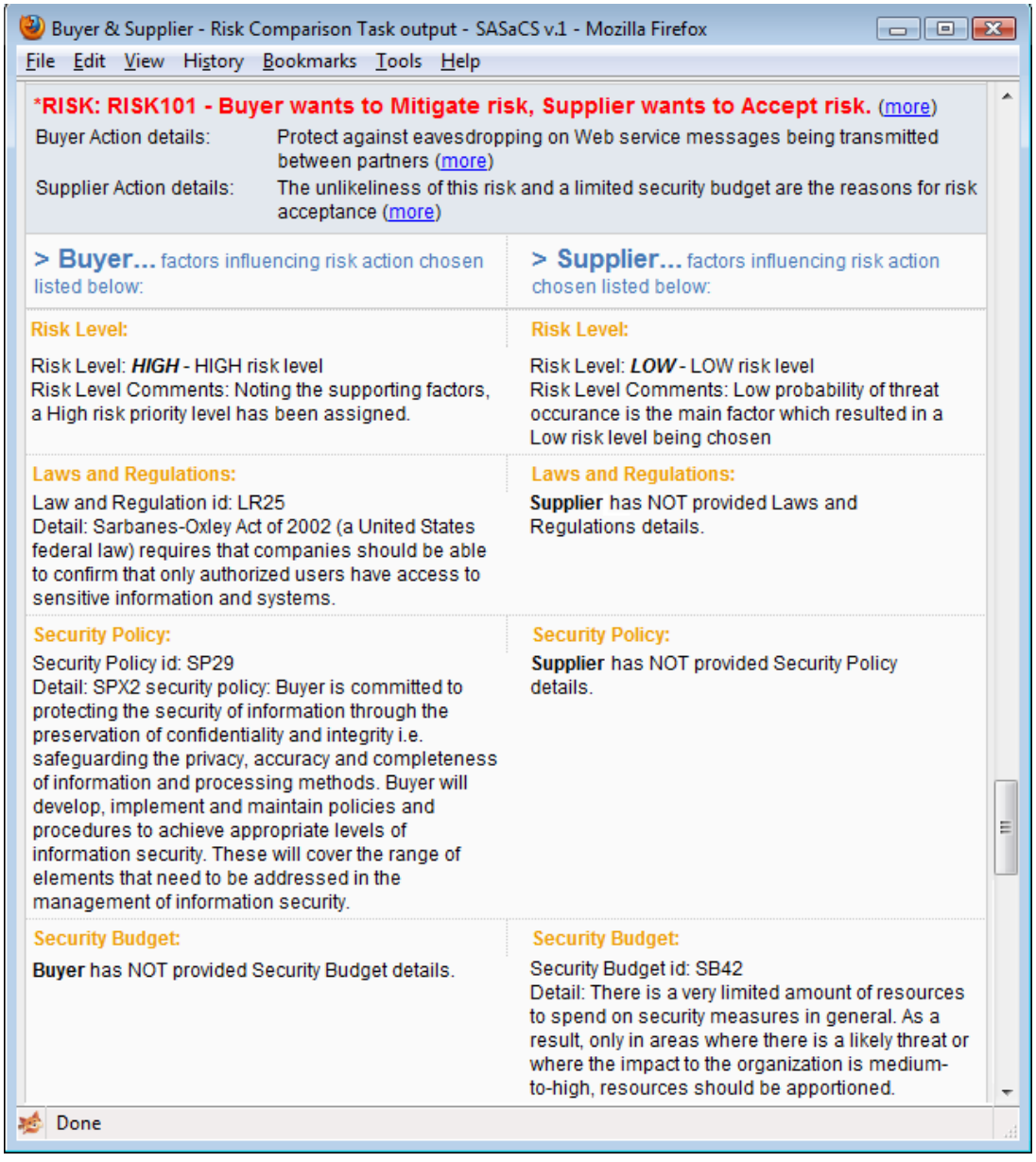


Figure 8.9: `Buyer/Supplier` Security Action Comparison output in Firefox

The output in Figure 8.9 also aids in reconciling semantic differences across RM/RA approaches as these issues are resolved by mapping rules earlier in the process. Furthermore, personnel from companies can refer to the ontology and the inclusive shared definitions/terminology at any point. This would be done to attain a clear understanding on terms in the context of the interactions. As parties come together therefore, they immediately can identify any conflicts in treatment choices and have the main factors supporting those conflicting choices displayed. This and the discussion above gives evidence to show that in many ways SASaCS has brought interacting enterprises closer together. This therefore allows for an easier transition between the Requirement Elicitation and Negotiation phases in BOF4WSS.

The shortcomings of SASaCS identified in this section's case study centred around the manual effort needed at a few stages to complete data mapping. This acted to limit some of the Solution Model's automation goals. To critically consider this point however, the level of automation available with a full system (with actual mapping functionality coded) would significantly bridge the disparity gaps and support a much easier negotiation on security actions between parties. A small degree of manual intervention therefore, even though not preferred, might be negligible. This is especially in business scenarios where there are large amounts of risks or security actions to be deliberated, and thus saving time at any point would result in substantial boosts in productivity. Considering all these aspects, this work and the findings thus far are seen to go a long way to positively answering the main research question posed in Section 5.2.3.

8.9 Summary

The focus of Chapter 8 was assessing the compatibility of the SASaCS tool and the ontology which it embodied, with RM/RA methodologies used by businesses today. As stated previously, a good level of compatibility was imperative given the necessary interactions between the tool and these approaches expected within

BOF4WSS. From the comparison and case study 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.

The following chapter covers the second stage of this research's evaluation. This stage provides an assessment of BOF4WSS and the Solution Model from a third-party perspective using interviews with industry-based security professionals. The findings from this evaluation act to give further insight into the applicability and strength of this research's proposals.

Chapter 9

Evaluating BOF4WSS and the Solution Model

Interviewing may be defined simply as a conversation with a purpose. Specifically, the purpose is to gather information. — Bruce L. Berg

9.1 Introduction

The second stage of the research's evaluation process is presented in this chapter. This stage involves an assessment of BOF4WSS and a second examination of the Solution Model (and resulting tool) proposed to support phase transition. Noting the limitations of a full evaluation (outlined in Section 1.3), the ideal option of a real-life case study evaluation for both proposals was not available. This left two alternate evaluation options.

The first option was the design and analysis of a virtual case study to see how the proposals (BOF4WSS, the Model and tool) would perform in addressing key issues. The second option relied on a third-party assessment of the proposals by conducting interviews with industry-based security professionals. Considering the critical feedback that could be gained from professionals in the field, the latter option was chosen. This would enable objective and detailed data to be gathered from knowledgeable persons in areas where proposals were ultimately

to be applied.

Formally therefore, this evaluation's aim was to utilize carefully structured interviews with industry-based security professionals to gather useful feedback on the proposals and their suitability for problems they target. This feedback could then be analysed and conclusions drawn to assess key research areas/statements. This chapter reports on that evaluation including the presentation of the research methods followed and an in-depth discussion of the research's findings.

9.2 Evaluation Method

To evaluate BOF4WSS and the Solution Model, a standard structure of research was followed. This included the definition of areas of interest and then the collection and analysis of relevant data to assess these areas. Rigid hypotheses were not preferred because this evaluation does not seek to thoroughly prove or disprove formal theory. Instead, the aim is to establish whether the information gathered supports the areas and proposals assessed, and if so, the degrees of support arising from the data gathered.

There were two core *areas* to be investigated for support in this evaluation, both related to original research aims and objectives in Chapter 1. The first was to investigate whether the framework proposed was an applicable, practical proposal which would aid businesses in reaching requisite levels of enhanced inter-organizational security and trust. The second was to examine if the Solution Model provided a viable process to greatly support transition between the Requirements Elicitation and Negotiation phases of the proposed framework.

To study these areas, a qualitative research strategy was chosen in which digitally-recorded, semi-structured interviews were employed. The interview data gathering technique was preferred as it allowed for a detailed study into the field and the gathering of descriptive, insightful data for analysis [130]. Semi-structured interviews enhanced this process because they allowed for a mixture of structure and flexibility in the questions asked. Therefore, in addition to asking

planned questions which directly related to the areas above, other interesting and associated observations could be explored.

To ensure the interview questions were clear and appropriate, pilots were used to refine them initially. Also, in the interest of gaining the highest quality feedback, interviewees were sent general documentation on the models at least a week before the interview. This allowed them time to review the proposals and gather their thoughts before the meeting.

As was mentioned, the target group for interviewees consisted of industry-based security professionals. To narrow this further, purposive sampling [12] (which is the use of special knowledge to select appropriate subjects) was applied. Within this general group therefore, individuals were selected that showed a good experience (demonstrated by job roles, certifications, qualifications and past project involvements) in the following pertinent fields: Web services technology, e-business and online business paradigms, security risk management, information assurance, security architectures and cross-enterprise interactions.

The interviewee selection process consisted of directly contacting persons with demonstrated experience (identified from company Web sites and/or articles published) and using the principal researcher's contacts within companies to help identify other suitable professionals. This targeted technique was adopted as opposed to more statistically random or quasi-random techniques to ensure that persons selected had a good degree of requisite experience and specialized knowledge.

Additionally, because the emphasis was on gathering in-depth information rather than surface-level data from as many persons as possible, only five professionals were interviewed. These professionals however had a total of 48 years experience in the security field. This small sample size allowed for a manageable yet very detailed amount of expert feedback to be gathered in the, on average, two-hour long interviews. Small sample sizes, greater depth of information and a focus on narrative data, are all key characteristics of purposive sampling [200]. Known limitations of this sampling technique however include possible bias in in-

terviewee selection and lack of wide generalizability of findings [12]. As subjects were selected based only on demonstrated experience and no knowledge of their personal opinions, bias was not viewed as a serious limitation here. Furthermore, wide and conclusive generalizations are not the goals of this evaluation but rather to gain some insight into the use of and support for research proposals.

Therefore, although there are noteworthy limitations of purposive sampling, the benefits possible with the technique were seen to outweigh the drawbacks in this case. This is especially considering the resource and time constraints on this doctoral project, and great amount of time taken even to set up interviews with the five subjects chosen. Common issues faced were the busyness and hectic schedules of professionals, coupled with the need for companies' legal departments to be involved to consider and approve interviewees' participation. Finally, to encourage honest and detailed feedback, the interviewees were told that their identities would be kept anonymous. This also avoided any further legal complications with their employers.

The overall goal of the interview process therefore was to present BOF4WSS and the Solution Model (particularly the core characteristics, possible areas of contention and novel aspects), and gather real-life, expert opinions and in-depth insights. This feedback would delve into the applicability (how suitable are the models for the situations and problems they target, what might the response from companies be, and so on) and strength (how well, if at all, are the problems addressed by models, what are their benefits and shortcomings, and so on) of the proposals based on security professionals' experiences. This information would then be used to aid in assessing the fulfilment of research aims in Chapter 1 and lower-level research questions such as the main one defined in Section 5.2.3.

Having conducted the interviews, recordings were then transcribed. As the focus was not towards a detailed linguistic or psychological analysis of data, precise transcripts (including pauses, meaningless repetitive interjections, digressions and indications of mood) were not produced; this course of action was supported by Rubin and Rubin [178]. Apart from this, the transcripts were accurate repre-

sentations which documented interviewees' feedback.

To analyse the data collected, the content analysis [12] data analysis technique was then applied. This provided a standard method to code, organize and index the transcribed interviews. Furthermore, it allowed for easy data retrieval, pattern identification and review, and basic counting to note any relevant quantitative observations [12]. A blend of deductive and inductive approaches to identifying themes in the data was favoured. This enabled themes to be identified which focused on investigating the predetermined areas for support (deductive) but also common themes that arose from data that were not conceived before (inductive).

With the research process outlined, the next section concentrates on the presentation and analysis of the research findings. This research interweaves the findings and analysis stages because it was felt that this would allow for a rich but also concise discussion. Berg [12] supports the viability of this combined option especially when compiling reports based on qualitative data. For reference, a list of the prepared interview questions is presented in Appendix B.

9.3 Presentation and Analysis of Findings

9.3.1 BOF4WSS

The first *area* to be investigated centres around whether the framework proposed is an applicable, practical proposal which would aid collaborating businesses in achieving desired levels of enhanced inter-organizational security and trust. To examine this, questions to interviewees concentrated on core principles and novel aspects of the framework which specifically aimed at addressing the outstanding research problems raised in Section 2.3.3. Four *themes* have been identified in which to present and analyse the data gathered.

The themes consider (i) the framework's emphasis on a highly collaborative approach to inter-organizational security, particularly where WS is concerned, (ii) the reality that BOF4WSS is detailed and at times prescriptive, (iii) the merit

of the framework's focus on higher layers (business-level for example) of security in WS-based cross-enterprise interactions, and (iv) the use of the Interaction Security Strategy (ISS) as a comprehensive security management structure, that could also foster trust across partners.

Using interviewees' feedback, the themes are assessed in terms of their use, strength and application. After that analysis, an additional section is presented with interviewees' general comments on the framework, before briefly summarizing the assessment thus far. In the presentation below, fictitious names are used for interviewees. This respects their anonymity while also allowing for a more vivid presentation of findings.

BOF4WSS and its highly collaborative approach

BOF4WSS emphasizes a highly collaborative approach to cross-enterprise security. This high degree of collaboration (manifested in dedication to working together, a good degree of information sharing, various meetings, and other time and investment commitments) was conceived specifically to address the shortcomings stemming from the isolated and individualistic approaches to securing e-business collaborations which use WS. Noting the amount of stress the framework places on this topic, it was chosen as one of the areas to evaluate within the interviews. The aim was to determine whether highly collaborative approaches such as the framework, might provide more adequate solutions for WS-based e-business interactions, as opposed to more individualistic approaches. The subsequent aim would be to then identify how applicable and practical such approaches are.

In response to questions posed regarding high degrees of collaboration as opposed to individual approaches to security, all professionals expressed that these types of approaches were preferred and yielded better security solutions. Interviewees indicated that solutions were likely to be more appropriate, skills and knowledge could be pooled, and finally systems could be designed and integrated more securely. This favourable opinion was upheld by professionals when questioned about BOF4WSS and its collaborative efforts towards security as well. An

interesting point put forward by one professional was that collaboration (especially initial meetings and willingness to work together) enabled him to be able to determine whether or not other companies were really committed to interactions and security or not. Collaboration was therefore being used as a tool to learn about potential partners and even their security postures *before* entering fully into business interactions with them.

Considering collaboration in the context of WS and BOF4WSS, John, a security professional of 10 years working for a leading international IT and consultancy services company, noted that collaboration is essential and needed at all levels (business, legal and technical agreements). Continuing the Lead Security Architect said, "... particularly with Web services, it has great promise but it's only going to work with that sort of collaboration". This view points to the importance of an increased amount of collaboration, even within the technology-driven WS world. Detailed feedback from other interviewees supported the importance of collaboration between companies in achieving inter-organizational security. Existing case study data (see Todd et al. [206]) also emphasizes the benefits of collaboration.

Even though supporters of collaboration, two professionals warned that it was important for businesses to maintain some degree of individuality (in terms of self-defense capabilities), or at least some safety net features (contract- or technical-based) within collaborations. These would protect individual companies if their partners inadvertently or intentionally became rogue. This point acts as a reminder that collaborative security approaches should not only focus on protecting the group of entities, but also protecting individual enterprises from the risks of being in the collaboration (Baker et al. [6] analyse and document some of these risks).

Having looked at the use of highly collaborative approaches in building cross-enterprise security solutions, the next step was to assess the application and practicality of such approaches and the framework in particular. From the feedback received, two opposing views were apparent. Three professionals regarded

high degrees of collaboration across companies as difficult to attain, whereas the others saw it as "quite practical" and not "too big a barrier". The main proponent for the former perspective was Mark, the Principal of Information Assurance in a global telecommunications and consultancy firm.

Drawing upon his 20 years in the security field, Mark stressed that collaboration was beneficial to have, but very difficult to attain. Additionally, making persons communicate, work together and readily share information (which are key activities in a collaborative process such as BOF4WSS) were not easy tasks. Prime reasons cited centred around the likely problems incurred when meshing teams from different companies with possibly different perspectives, processes, systems and organizational cultures. These issues are supported by literature in [7, 71].

Interestingly, John also showed an appreciation for the collaboration difficulties mentioned above but did not view them as too much of a barrier. Instead he noted, "yes it is intensive and costly to some extent and I think that's the only way to be really successful". In spite of these difficulties therefore, in his opinion, these approaches were not only practical but a necessity for success with security. Literature could be seen to support this 'security success via collaboration' perspective but primarily in closely knit business partnerships such as the extended enterprise (see Dynes et al. [49]).

Considering BOF4WSS in more detail, additional notable difficulties were identified by subjects relating to complexities in stakeholder arrangement and management (getting the right people together at the right time from across companies) and cross-border collaboration issues (in essence, normal collaboration issues exacerbated by ranges of cultures and perspectives) if/when the framework was applied internationally.

Speaking objectively, the aspects mentioned were somewhat overlooked by BOF4WSS due to the assumption that shared business aims, and goals for security would drive and support collaboration. When this assumption was put to subjects, some respondents agreed that shared aims would help. However, they

also expressed that there would need to be strong, mutually understood benefits for all companies, degrees of fairness ("Nobody wants to be the weak partner", Mark stated) and executive sponsorship from businesses. High-value projects and situations where there was positive history (and existing trust) between companies were also cited as scenarios in which high degrees of collaboration would be more practical. All of these driving factors would have implications for BOF4WSS and indicate situations in which it might be best used.

In summary, there was some consensus that the high degree of collaboration advocated by BOF4WSS would lead to a more adequate security solution for cross-enterprise interactions. According to the data however, its applicability may be limited (or at least, best suited) to business scenarios where either there is a strong commitment to businesses goals (and security is seen as an enabler to those), a substantial degree of executive sponsorship, they are high-value projects (amount stood to be gained or loss, motivated need to do whatever necessary to get job done), or there is existing trust between companies. The first two of these were previously mentioned in Section 3.12 as criteria for businesses adopting BOF4WSS. Conversely, the need for positive history and some degree of existing trust between companies was not envisaged before as a prerequisite to adoption. This was a significant finding because it suggested that even though the framework was aimed at building trust across partners, it might be more practical if some interaction history or degree of trust already existed.

Detailed and at times prescriptive framework

In seeking to create a comprehensive security-focused methodology (which supported companies from the planning to maintenance of cross-enterprise interactions using WS), a central objective of BOF4WSS was to provide detailed and occasionally prescriptive guidance. This guidance included the activities that might and should be conducted, possible ways in which they could be conducted, and their pertinence to attaining desired levels of layered security within the foreseen cross-enterprise interactions. With appreciation of the detailed level of

guidance and the possibility that it might not be well received by companies, it was chosen for assessment in the interviews. The objective was to ascertain its usefulness and applicability in aiding the creation of a security solution.

From an analysis of the data, it was seen that a majority of professionals found the detail in BOF4WSS of benefit to companies, and felt that enterprises would and should be open to it. One benefit referenced focused on the fact that detail would force people to consider all the factors and give structured ways—especially for inexperienced persons—to solve security problems. Another benefit seen in the framework was the visibility and ability to audit that it would bring to all aspects of the cross-enterprise scenario. According to Matthew, head of Information Security, Risk Management and Business Continuity at a UK educational institution, "An audit department would absolutely love this". This was stated because he felt that the framework would define a structure that audit departments, even though not security specialist, could use to track and monitor projects.

The main warning placed on the framework by professionals was that it should be wary of being detailed and prescriptive to the extent that companies were not allowed to adapt parts to the nature/culture of their enterprises. This could relate to tools, specific techniques or constituent methodologies. As is seen from Chapter 3 however, the framework appreciates these issues and either provides a set of options (such as a listing of risk management methods to determine security needs) or relies on industry standards and best practices (including use of ISO/IEC 27000 for security, or UML for modelling).

From the findings above therefore, it can be concluded that the detail provided by BOF4WSS should be useful to businesses and more of an advantage than a hindrance. This would not only apply to persons and businesses that lack experience in dealing with WS security issues within an e-business context, but also to entities seeking to have a framework to maintain structure, consistency and visibility throughout the complete process.

Appreciation of higher layers of security in cross-enterprise interactions

Another main aim of BOF4WSS is to emphasize holistic security solutions. Holism is used to refer to an encompassing approach that considers technologies, policies, processes, methodologies, standards and best practices for security. This aim particularly attempts to combat the overly reliant focus on technical mechanisms for security discussed in Section 2.3.3. The purpose of this theme therefore is to evaluate that aim and its merit in the context of cross-enterprise WS interactions.

Commenting on the data gathered, all interviewees displayed an appreciation of high levels of security and echoed the sentiment that technical approaches alone were insufficient. This finding therefore supported the framework's charter and literature in Singhal et al. [189] which highlighted the need for the higher layer of security with WS.

Speaking on this topic, John remarked that the challenge found in business today was achieving this higher level of engagement in projects, especially business ownership, and business and ICT alignment. Technology-level integration was not a problem but rather getting the engagement, involvement and buy-in for projects at the business levels. Lack of these higher level aspects, he noted, were the reasons many projects failed or stalled. Considering this challenge in terms of BOF4WSS, there is a focus on the higher layer, however no special mechanisms of encouragement to achieve it are provided. In the framework design it was envisaged that there would be a top-down drive for projects and therefore efforts were concentrated on supplying guidance for the necessary processes.

An additional concern lodged by two professionals was that even though the higher layer of security was important, the translation and implementation of these higher aspects to lower levels were equally important and not to be neglected. Paul, a Senior Security Researcher at global British-based IT company, warned that various things get lost in translation and imperfect implementations. This can be to some extent supported by difficulties highlighted in [182, 198]. Furthermore, Paul stated that, "you cannot solve problems at the highest level,

that's the thing, you do have to come down to the lowest level". As a result of these factors, he highlighted that it was key that security go through the entire process. Paul also stressed that the framework should maintain a balance between higher and lower layers of security and not overly emphasise either. This was an accepted perspective in BOF4WSS as it aims for holistic security.

The next question to interviewees centred on trust and whether the higher layer (and the activities therein such as jointly defining policies, agreeing on process for security, meetings and so on) in BOF4WSS might lead to increased trust across entities and their personnel. In response to this, a majority of professionals agreed on the likelihood of increased trust resulting. Common rationales presented linked to time spent together and commitment towards security that, once present, would be demonstrated to partners. Both of these lead to relationship building, which then may lead to trust. Todd et al. [206] is one documented real-world scenario where high-level activities such as joint risk assessments, "proved to be the foundation upon which mutual trust between the security communities ... has been built" [206] (p.50).

Mark was the least enthusiastic about the higher layer naturally achieving trust as he felt that trust was a very complex and difficult thing to attain—a view supported by Van Slyke and Bélanger [212]. This he attributed to human factors and the difficulty in predicting human behaviour. Aside from this however, respondents' feedback supported the possibility of increased trust across business partners.

Use of the Interaction Security Strategy (ISS)

The Interaction Security Strategy (ISS) is one of the more novel parts of BOF4WSS, in that it seeks to create and apply a cross-enterprise management structure not found to be used in practice. The first question to interviewees therefore was to gather their opinion on this strategy in terms of security and trust. Another point of interest was how the strategy compared to existing approaches, particularly contracts, as these seemed to be the main agreements structure used today by

companies.

The feedback gathered indicated that a majority of security professionals felt that the ISS was a valid and useful approach for cross-enterprise security and trust. Only Luke, a Senior Security Researcher with 4 years experience, disagreed as he was not sure about ISS positioning in the framework's process flow, or the level of security present in the ISS; he regarded it as too detailed.

One intriguing finding was that even though legal contracts formed the main agreements mechanism across companies, they were reported to cover security only very generally. For example, if in the UK or EU, they might only very briefly reference the Data Protection Act. Drawing on his 10 years experience, Matthew highlighted that contracts are not likely to cover security policies, continuity planning, or even ISO/IEC 27000 best practices. He emphasized that it was therefore important to strive for an extra layer of security (similar to the ISS) to be put in place. Generally supporting this point, a 2010 survey [166] has highlighted that roughly 40% of large business respondents do not ensure that their contracts with third party providers include security provisions. This is a telling aspect in terms of contracts and their lack of focus on security.

Additional advantages of the ISS identified by some interviewees linked to the flexibility it would allow and the pragmatic, actionable structure it provided over contracts. Contracts were seen to be very specific, hard to follow and often expressed in legal jargon. The key stipulation made by subjects was that the ISS was always in line with the contracts. This, they stated, would ensure synergy in agreements. In general therefore, professionals' feedback above is seen to support the ISS as a key tool in creating and instilling a cross-enterprise security solution. This would enhance the practical security provided today and support agreements in contracts.

The second question related to the ISS concentrated on its use as a mechanism to foster trust across businesses. Trust was hoped to be achieved by making security approaches (pertaining to the scenario) more predictable and transparent (these being two key attributes of trust [40, 212, 181]). From the resulting

interview data, a consensus was apparent as professionals all regarded the ISS as likely to foster trust. Reasons supplied included the clear guidance to businesses, and the ownership and understanding it supplied personnel with, considering that they aided in its creation. Both of these aspects link with intended goals of the ISS noted in Chapter 3. John's support for the ISS in this regard was motivated by its charter towards a joint security posture, something that he felt was more conducive to trust, rather than the "us and them" mentality he saw in some businesses today. This opinion can be related to collaboration in general and the reality that some parties might not be willing to collaborate to this extent.

The other salient view on the ISS and trust was held by Mark. He expressed the point that,"[the ISS] probably fosters trust in that it takes away distrust ... What you'd certainly find is that one of the major hurdles is getting over the distrust, doesn't mean that you've actually got trust once you've got over that". This view, albeit a solitary one in the context of respondents, draws attention to the precarious nature of trust and possible difficulty in gaining it across persons and enterprises. In general however, the ISS is seen to positively aid in this venture and provide a structure that could enhance currently used mechanisms.

General thoughts on the framework

With the framework's core principles and novel aspects assessed, the next three paragraphs highlight other noteworthy feedback (based on consensus, ideas related to research literature, or simply practicality) given by interviewees.

One view that arose with respect to creating security frameworks and methodologies generally was the inherent difficulty they faced in balancing complexity and being comprehensive, while also making them useful and consumable by businesses. John aptly summarizes this opinion in his remark, "getting the balance right is so important where it's rigorous enough to add value and to make sense, make the process more structured, and at the right level but not so verbose that it's not useful". He further stated that even though the real proof would be in the adoption of BOF4WSS, to him, it looked okay and seemed "light enough

... to be useful".

Another intriguing point which surfaced was that BOF4WSS did not appear to be specially suited to medium-to-high security or trust industries or business scenarios. Instead, interviewees felt that it was generic and according to Matthew, "would be good across the board". This perspective was of interest because the framework was originally targeted at businesses and scenarios that emphasize trust and medium-to-high levels of security (see Section 3.12). These cases were chosen as they were seen to justify the significant effort and resources needed to adopt BOF4WSS. Based on the data collected however, the framework might have wider scenario applications, subject to limitations from other findings.

The final significant point relates to framework applicability again but more from a higher perspective. In considering the application of BOF4WSS to scenarios, Paul expressed that asymmetries (whether due to size or bargaining power) in the market might limit the framework's use. This was because asymmetries lead to some enterprises looking to individually develop solutions to service as many generic customers as possible. This was as opposed to focusing on one-to-one collaborations and individual partner requirements (as emphasized in the framework). Although this was a notion only mentioned by one professional, the collaborative nature of BOF4WSS might suggest that it is better suited for symmetric-type interactions. These are interactions where each party has an influence and party-to-party negotiations, design and development is expected.

Summarizing framework analysis

Having presented and analysed the main findings related to the framework, below these are briefly summarized and used to investigate the degree of support for the core *area* defined at the beginning of Section 9.3.1.

The first theme investigated the high degree of collaboration desired by the framework. Based on that analysis, collaboration was seen to lead to more adequate and thus enhanced solutions as compared to those possible with individual or isolated approaches to security. Additionally, it was also concluded

that BOF4WSS (and to some extent, highly collaborative approaches in general) may be better suited to certain business situations and scenarios because of their nature (see the collaboration theme discussion for details). These findings strongly support the area being investigated but limit the target scenarios of the framework.

Considering the level of detail provided by BOF4WSS, a majority of interviewers saw this as a benefit to companies which would and should be welcomed. This was assuming that it allowed some degree of flexibility, which it can be said that BOF4WSS does (through the provision of various tool/technique options). Cited benefits of the framework included forcing companies to consider all the factors, aiding inexperienced persons (in what is arguably still a relatively immature field in terms of WS use for supporting complex business processes), and creating a level of visibility and ability to audit for cross-enterprise development and subsequent interactions. These aspects can all be seen to enhance current security approaches and therefore provide good support for the area studied.

Reflecting on the appreciation for higher layers of security in the context of WS in e-business, data showed a consensus in their merit and value within the overall security approach and solution. The main concern identified at this stage related to getting the necessary level of engagement, at what is essentially the business layer within companies. This is a problem not covered by the framework as it was assumed the necessary top-down drive for projects already existed. This top-down drive would be present in the applicable scenarios suited for BOF4WSS, highlighted in the sections above.

On the topic of trust, a majority of positive interviewee feedback acted to further support the framework's appreciation of and concentration on this higher layer. To recap, this layer involved getting companies together to interact, collaborate, and discuss and plan interactions security. Generally, these findings are therefore considered to provide a noteworthy degree of support for the area being investigated, both in terms of security and trust.

The ISS is in many ways a specialization of the higher layer security ap-

proach covered above, and interviewees also saw it as a useful approach in terms of cross-enterprise security. Its importance was accentuated particularly because there seemed to be no standard overarching management or guidance structure for businesses which pertained to security. Contracts were referenced, but it is known that these documents do not contain detail on security nor do they place it in an actionable language and context. Furthermore, findings indicated that trust between companies was likely to be fostered by the ISS. Interviewees linked this to the transparency and clear guidance for companies, and ownership and understanding implied as companies would have aided in the creation of the ISS. In terms of the area for support, the novelty in the ISS was seen to add to current approaches both in terms of security, and possibly also regarding trust.

Based on the preceding paragraphs and sections, it can be concluded that in the context of this evaluation there is significant support for the framework. This support is with respect to providing an applicable and practical approach to enable businesses to reach requisite levels of enhanced cross-enterprise security and trust. This therefore relates to the fulfilment of the second research aim in Chapter 1. Critically speaking, the majority of support for the use and viability of the framework relates to business scenarios where there is either a strong commitment to businesses goals, a great degree of executive sponsorship, they are high-value projects (and this value drives the need to do whatever necessary to complete the task properly), there is history and existing trust between companies, or there is symmetry in business interactions. Based on these characteristics and predefined target scenarios for the framework as defined in Chapter 3, specific candidate companies that should benefit most from BOF4WSS adoption would be the following.

– Large companies with smaller units (or subsidiaries) seeking to streamline online interactions using WS between these smaller units — As part of the same company, executive sponsorship and strong commitment from parent units would be a strong driver for smaller units to collaborate and bring interactions to fruition. These units would be focused towards symmetric

collaboration therefore there would be the need for both parties to engage in context-specific negotiations, design, customization and development. Also, assuming history between these units (given that it is the same company) there will already be a foundation of trust that can be exploited and built on.

– Partners in an extended enterprise setting, for example e-supply chains — Research in extended enterprises aided in the construction of this framework and a number of the criteria listed above meshes with needs in these types of business networks. As trust is already a key prerequisite in extended enterprises [40], if a group of businesses in such a network desired to switch from proprietary integration formats to WS for cross-enterprise interactions, BOF4WSS would be very useful. The long-term nature of these networks and strong commitment towards a shared goal and mutual benefits also support the framework's use. Furthermore, because these businesses tend to already be collaborators at the strategic and business level, collaborations in security using BOF4WSS would be a natural next step to protect inter-organizational interactions and individual enterprises. Symmetric interaction would also apply.

– Small and medium-sized enterprises (SMEs) seeking to build long-term partnerships — This relates in particular to small and medium-sized companies with past history, a strong commitment to partnerships, sustained symmetric interactions, and the desire to achieve shared business goals realized using WS. BOF4WSS would be of great applicability to these type of companies for two reasons. First, because there might be a lack of expertise and experience, the framework's detailed guidance would be very useful. Second, as there are fewer stakeholders, stakeholder arrangement and management should be less of a problem. To justify the time and resources necessary by BOF4WSS, long-term alliances are likely to be the most practical scenarios. In such situations companies can see their investment yielding returns in the long-term.

The next section presents the findings and analysis conducted regarding

the Solution Model.

9.3.2 The Solution Model

In this section, the second core *area* is examined to determine whether the findings support it, and if so, to what extent. This involves an investigation into whether the Solution Model provides a viable process to support transition between the Requirements Elicitation and Negotiation phases of the proposed framework. Similar to the evaluation of BOF4WSS above, questions to interviewees assessed novel characteristics and core precepts of the Solution Model.

For the presentation and analysis of data, four *themes* have been chosen. These are (i) opinions on transition problems highlighted, (ii) the premise that risks drive security actions and requirements, (iii) the likelihood of business partners sharing detailed information on common risks and their intended treatments; and (iv) the ultimate perceived use of the Model and tool. Data within these themes is analysed with respect to its application and scope.

Opinions on transition problems highlighted

The charter of the Solution Model was to address the transition problems that companies were likely to encounter in moving from the Requirements Elicitation to Negotiation phases in the framework. These problems were identified based on an informed scenario and relevant research literature. Considering their importance as a driving factor for the Model however, this theme assesses the issues again with the goal of determining exactly how serious they might be from professionals' perspectives.

Commenting on the feedback received, all but one security professional—namely, Luke—agreed with the transition issues highlighted. In response, Luke said he was unsure whether security would be considered at what he considered, an early stage in negotiations. In cases where there was agreement, professionals concurred with all of the transitional problems (such as semantics issues, difficulties understanding motivation for actions, and the arduous task of comparing

and negotiating actions) and substantiated their opinions by drawing on past experiences.

In terms of semantics issues during phase transition, John stressed the importance of spending time initially agreeing on terminology in projects, as words in the security domain were often misused. Paul and Matthew were two of the main proponents supporting the reality of disparity in formats of security actions and requirements. Relating to this, Matthew stated, "there are companies that might have a basic statement, they might have a graphical representation, they might have a few bits and pieces and in my experience actually getting those to marry together initially, is one of the hurdles you do have to get over". These aspects can be compared to the security mayhem discussed by Tiller [205].

One of the most interesting findings in the data related to the motivation behind security actions and requirements. On this topic, John noted that in addition to partners not supplying (or supplying little) motivational information initially, if they were asked to justify actions at a subsequent stage, they did not always have good reasons to support their security actions. He explained that in some situations where a predefined set of security actions (such as reused action lists or generic security checklists) were provided by companies, the original meaning might have be lost or the security landscape might have changed since their creation. Therefore in addition to the problems associated with businesses not communicating the motivation behind security actions, the reality exists that companies themselves might not be clear about reasons for their actions. This adds an extra level of complexity and discussions as companies meet in the Negotiations phase.

Another noteworthy observation from the data was that personnel involved in cross-enterprise negotiations may not always have a security background—they may be business-oriented persons for example. Matthew felt that some personnel have basic knowledge of security aspects but because they lacked core knowledge and experience in security, this tended to prolong the negotiations process. This is important because it highlights that even though it may be desirable for security

experts to be involved in negotiation, that might not always be the case. This lack of involvement however can affect the negotiations process negatively.

The findings presented and analysed in the previous paragraphs all help to support the reality of the problems faced as companies transition between BOF4WSS phases (or any general cross-enterprise negotiations task really). Mark's statement in response to the question about transition problems sums it up aptly as he expressed, "Oh, I've seen that, and you're exactly right, that is the way it happens, it takes months, possibly years in some circumstances". This quote captures the seriousness of the transition problems highlighted in this research.

Risks drive security actions and requirements

To ease difficulties in the initial matching and comparison of security actions and requirements across enterprises, the Solution Model proposed the use of a shared risks catalogue. A common risks base would be key to allowing for automated matching using a tool. Central to this proposal was the idea that risks are the core drivers for security actions. This notion was supported by literature surveyed in Chapter 6 and thus embodied in the resulting ontology. With appreciation of the importance of this notion to the Model and resulting software tool (SASaCS), it was chosen for assessment in the interviews.

Reporting on the data gathered, a majority of professionals supported the 'risk-driven' notion. Feedback ranged from, "it always stems from risks and understanding risks, risk management, risk evaluation, it really drives everything to be honest" to "driving security, a risk-based approach is something I firmly believe in". Cost factors were also mentioned by one interviewee but these still related to underlying risks and their mitigation cost/benefit savings. Interviewee feedback therefore can be seen to give support to findings in Chapter 6.

While accepting the role of risks as a driver for security, one interviewee expressed that a number of companies do not actually operate on a risk basis. Unfortunately, no examples were given as to what companies might do instead to define their actions. This reality is nonetheless a thought-provoking one in

terms of the Solution Model because even though it is not ideal (interviewees and research from Chapter 6 point to a risk-based approach being best), if it is widespread it might limit the adoption of the Model and tool.

The last important finding related to the communications benefit likely to result in using risks as a base for security-related discussions. Interviewee feedback identified that in using a risks base, security professionals and business persons (involved in negotiations) alike could understand what was at stake (impact to organization and so on). From this research's perspective, this is beneficial for two reasons. Firstly, if business-level personnel do engage in security negotiations (as alluded to in the theme above), using a language they will understand would give them the necessary insight into the process. Secondly, business persons are typically the budget holders (John and Mark stress this fact) therefore again, they have to understand the need for security for funds to be released to implement security actions.

Likelihood of sharing detailed information on risks and risks' treatments

The Solution Model and BOF4WSS require that business partners share a great amount of information on common risks faced, factors (including laws, organizational policies and so on) that influence/motivate security actions, and security actions themselves (whether they are geared towards risk mitigation or otherwise). With appreciation of the possible inherent difficulties accompanying this task (such as companies not wanting to share this information), the evaluation theme focuses on how realistic an expectation this is.

The conclusions from the data analysis in this theme were less clear and even in cases where professionals felt that information sharing was realistic, they still placed a number of conditions on sharing. For example, some stated that once the data requested were at a relatively high level and did not go into specific vulnerabilities or impacts to the organization, it would be feasible. This was an intriguing finding because the structure of the risks catalogue and data in

SASaCS does to some extent ask companies to define specific vulnerabilities that constitute a risk. This might therefore require the catalogue structure to be modified slightly to show less detail, or finding scenarios where parties were likely to be open and the structure could be accepted as is.

Supporting the opposite view, the feedback did observe that in some situations, companies might refuse to give much information to partners and cite confidentiality reasons. Overall however they were two prerequisites identified that would increase the likelihood of information sharing. These were, trust and an existing relationship between companies. Mark states, "a lot of companies, particularly in private sector are unlikely to do that unless you've got that trust". This shows a significance of existing relationships and trust to the Solution Model, similar to that necessary for the framework.

The ultimate use of the Model and tool

The SASaCS tool is a software implementation of the Solution Model. As such, it aims to streamline a number of tedious, repetitive and long-winded tasks, and thus significantly ease transition between the two framework phases. This directly relates to attaining an answer for the research question presented in Section 5.2.3. The evaluation of the Model, largely by way of the tool, was therefore imperative in these interviews. To conduct this evaluation, the tool prototype was demonstrated to interviewees and then questions were asked. Below, the feedback and analysis results are presented.

In response to questions regarding the tool's usefulness in supporting phase transition, interviewees felt that it was a very useful approach and system. John stated, "I think it would be really useful. Having seen it, I think the penny has dropped for me, I think this could be very powerful, very useful. I think this would help a lot". Furthermore he expressed, "And it would accelerate the adoption of technology solutions and this framework". John made this statement because he felt that in business today, collaborations are somewhat technology-focused and what inhibits projects is the discussion and agreement difficulties

arising from the business and legal sides. The tool, to him, was seen to help these sides by considering security at a higher level, communicable to people at this layer (business or legal professionals for example).

Mark was another professional who strongly supported the tool's usefulness. He commented, "a tool that helps bring that [core negotiation aspects] directly onto the table, it makes that time together far more productive". Such opinions as those mentioned here and above, give evidence to support the increased productivity achievable by using the tool (and the underlying Solution Model proposed). Matthew reinforces these point as he states, "I can think of projects that it probably would have shaved off months, in terms of the initial stages of that project, had they thought to do this earlier on".

When questioned about whether they (interviewees) would use the tool in such a negotiations scenario, a majority of subjects said that they would consider it—increased productivity being cited as the prime factor. Proponents also stated that the novel benefit with the Model and tool was that they laid out companies' security positions in a clear and direct format, and forced them to agree or disagree on positions/postures. Regarding the automated identification of conflicting security actions for risks, John stated, "you almost know straight away that the collaboration is not going to work unless someone changes their posture or they agree to something". The tool can therefore save time for companies in this regard (a feasibility level) also.

From a usability perspective, generally positive feedback was recorded. Perceived benefits related to good accessibility due to the use of a browser-based report format and the ease at which security actions from companies could be compared. Shortcomings mentioned included the need for increased flexibility in tool output (such as additional buttons and more options on screen). These are accepted as areas for improvement in moving from a prototype to construct a full version of SASaCS.

Even though interviewees affirmed the tool's usefulness in significantly supporting the phase transition, some noteworthy shortcomings were identified. Cri-

tiquing the higher level data present in the tool, Luke states, "it seems useful with the caveat that it might hide stuff away from the decision makers". To remedy this, he suggests a drill-down functionality to allow more detail to be seen on treatments or risks. This feature would be used by security professionals involved in negotiations, whereas business-oriented decision makers might be happy with the current higher level information. This is a useful suggestion but if implemented it would have to be optional. This is because, as was identified in the previous discussion theme, all companies might not be willing to share detailed information. Trust, to some extent, again becomes a factor.

Another observation mentioned was the dependence of the tool on the quality of the input data. "It is the input data's quality that is going to impact on the influence [of the tool]", Luke stresses. Matthew also supported this fact. This is obviously an issue, however little can be done beyond giving guides and on screen tooltips to companies and users. It is assumed that companies would appreciate the productivity benefits when quality data is provided and therefore use the Model and tool as suggested. Inadequate provision of information by some partners in a collaboration might even act as an indicator to other companies as to how serious partners are regarding collaboration and collaboration security.

Summarizing Solution Model analysis

In the following paragraphs, the findings presented and analysed above are summarized in a *theme-by-theme* fashion. The conclusions drawn are then used to determine the degree of support for the core *area* defined at the beginning of Section 9.3.2.

The first theme of analysis related to determining the severity of the transition problems that motivated the Solution Model's design. From the data, it was clear that a majority of professionals appreciated the problems (largely drawing on their own experiences) and viewed them as quite serious issues within projects. Additional issues were even mentioned relating to companies themselves not being clear on the exact motivation for security actions and inexperienced personnel

being involved in negotiations. Considering these points in light of the area under analysis, they can therefore be seen to support the seriousness of transition problems, especially relating to the great deal of time consumed and lack of productivity.

The Solution Model operates on the premise that security risks drive security actions and security requirements. The validity of this premise therefore directly affects the viability of the Model and resulting system/tool. Based on the data, most professionals supported this premise and viewed it as the best way forward. Furthermore, it was seen to have additional uses because the notion of a risk was viewed as a key communications tool that could give business persons the necessary insight into security. One contrary viewpoint to risks as a driver was that a number of companies actually do not operate on this basis. Without any clear indication of a standard, well-justified process to identify actions however, little could be done to address this issue. With respect to supporting the viability of the Solution Model therefore, the data was seen to strongly support a risks base to security actions.

For the Solution Model to work companies are required to share detailed information on risks related to the scenario, influential factors in risk treatment and defined security actions. On assessing the likelihood of that occurring, the analysis conclusions were not definitive. Some professionals regarded it as realistic, whilst others did not. Possibly the most noteworthy finding here was that trust and an existing relationship were cited as factors that might increase the likelihood of this information being shared. This is an acceptable prerequisite as it largely fits in with the updated target scenarios of BOF4WSS outlined at the end of Section 9.3.1. Assuming an atmosphere with trust and an existing relationship therefore, the interview findings can be seen to support an enhanced level of information sharing and thus to some extent, the viability of the Model.

In investigating the Solution Model by way of the tool, the most significant question would have to be centred around the ultimate strength of the process and tool itself. In response to this question, professionals gave very positive feed-

back and affirmed the usefulness of the tool in significantly easing cross-enterprise security negotiations. The Model and tool were especially seen to accelerate adoption of technology solutions and increase productivity by reducing time spent in negotiations. Additionally, one professional saw it as beneficial to the overarching framework such that it would accelerate the framework's adoption. This formed a salient point because it suggested that research into support systems (such as the Solution Model and tool) could positively impact the adoption of BOF4WSS.

Another important advantage is the fact that by requesting information on motivational/influential factors *before* companies meet, entities will have to find clear justifications to support their security actions. This directly helps to address the issues related to incomplete information and weakly justified motivational factors identified in the transition problems theme. Reflecting on the analysis area therefore, the findings and conclusions from this theme strongly support the viability of the Model in supporting phase transition. There might be some slight improvements that can be made (including drill down functionality, modifying structure of risks data in the catalogue and SASaCS) but these were not seen to seriously affect the use of the tool or viability of the Model.

In summary, the findings gathered provided a solid degree of support for the viability of the Solution Model in greatly aiding the transition between Requirements Elicitation and Negotiation phases of BOF4WSS. Trust and existing relationships between parties also played an important role, however this is acceptable as it coincides with the updated target scenarios of the framework. Generally therefore these findings positively answer the research question stated in Section 5.2.3.

Lastly, as this section represents the second evaluation of the Solution Model and tool (the first was the compatibility assessment), the findings and conclusions of the two evaluations were compared for any points of interest. One important observation was found. This was based on the fact that constraints (laws, obligations, policies and so on) were seen as an additional driver of security actions in Section 8.7, whereas in this chapter security professionals only

mentioned risks. Although this leads to no clear conclusion, because the Model and tool by nature should be comprehensive, they should arguably accommodate both cases. Critically speaking therefore, the viability of the Model and tool can be regarded as negatively affected because currently they only use a risks base (and thus will only automate handling of risk-based security actions). Possible ways that constraints could be included in automated handling were previously discussed in Section 8.7.

Even though the negative feedback mentioned above harms viability, the strong support for the risks base and the tool which was supplied by industry-based professionals was felt to outweigh this aspect. Future work towards automated handling of constraints will be pursued only to ensure that the Solution Model and tool are as comprehensive as possible. This would allow them to handle a greater number of situations in which they are required to support cross-enterprise negotiations.

9.4 Summary

This chapter hosted the second stage of this research's evaluation, and applied an interview-based analysis to assess BOF4WSS and the Solution Model and its tool. Findings were seen to support the framework and Solution Model as useful, viable and practical approaches in addressing the issues they target. There were however some limitations, particularly related to applicable scenarios for the framework and contentions regarding security actions and their core driving factors. These were important but were not viewed as factors that seriously undermined this research project's proposals.

The next chapter builds on the Solution Model and briefly explores a novel addition to further extend it. It is worth noting that the emphasis in Chapter 10 is not on defining an evaluated approach but on exploring the approach's use and considering some of the intriguing research issues which arise. These issues and the approach challenge current research thinking and also flow into future work.

Chapter 10

Exploring the Automated Reconciliation of Security Actions

The analysis of the way people make decisions (prescriptive theories) or the way people ought to make decisions (normative theories) is perhaps as old as the recorded history of mankind. — Evangelos Triantaphyllou

10.1 Introduction

Having completed the research's evaluation in the preceding two chapters, this chapter takes the opportunity to build on the favourable results attained and briefly explore an advancement of one of the research proposals, namely the Solution Model. The new area of interest specifically pertains to progressing the Model (and resulting tool) from just producing information reports to *automatically* reconciling conflicting security actions across businesses.

In detail therefore, the chapter's aim is to explore whether the Solution Model and SASaCS tool could be further enhanced to enable them to reconcile conflicting actions from companies. Instead of collating data and outputting information reports to companies' decision makers, the actual task of reconciling security actions would thus be moved from businesses' personnel to SASaCS.

The benefits of automated reconciliation particularly relate to the time saved by businesses in negotiations and the overall increased productivity that results. There are however some obvious reservations surrounding this type of approach. Of these, probably the most intriguing area stems from the question of, how a representative system-based reconciliation could be conducted that would lead to credible decisions. As this is a key foundational part of the new enhancement, this question has been selected as the focus for this chapter. The novelty of this work is twofold. Firstly, there is the attempt itself to automatically reconcile conflicting business-level security actions/decisions. Secondly, there is the process adopted to enable this task. This process is different in that it tries to include a number of decision factors not usually incorporated in a formal or numerical sense.

To aid in answering the aforementioned question, the core of the chapter begins with an analysis of how decisions—and particularly the security action reconciliation decision—could be formally (that is, mathematically and numerically) modelled. Formal modelling was necessary as it would allow for straightforward system/software processing. With the decision model defined and justified, a simple example is presented to demonstrate its application in conducting a reconciliation decision between three interacting companies.

Following the example, there is a detailed discussion of some of the main issues which arose in modelling and overall limitations of the decision model. As this is primarily an exploratory chapter, it is these issues and limitations which are of greatest interest. These are referenced again in terms of future work in the final chapter. Lastly, some first impressions on the decision model itself and its aims are presented and assessed. These impressions were gathered from security professionals as part of the general interview process presented in Chapter 9.

10.2 Decision Modelling

When interacting e-businesses come together at the Negotiations phase within BOF4WSS, they are likely to have a number of security actions that conflict with each other. The classic example is where one company wants to mitigate a particular shared risk while its partners in the scenario want to accept it. The question therefore is, how should companies go forward. In the preceding chapters of this thesis this problem has been studied in detail and a support model (Solution Model) and tool (SASaCS) were defined. The aims at that earlier stage were centred around getting all the required information before companies meet, structuring that information appropriately, and consequently using a system to output an informative report to aid a set of companies identify key discussion areas and make decisions.

Chapter 10 explores the progression of those aims into whether enhancements to the Model and tool might be able to support automated (system-based) decision making. This is with the understanding that some additional information to facilitate this process would be necessary from personnel. Having a system that could automatically reconcile conflicting security actions across companies would expedite phase transition for entities even more.

To allow for the system-based reconciliation of conflicting actions, this chapter investigates the formalization of the manual decision making process exemplified by companies' personnel in Section 4.3.2. Based on the informed guidance used in constructing the scenario, this is believed to be similar to typical processes used by real-world companies. To recap the case data, the decision making activity for businesses in the Negotiations phase consists of three core steps. In the first step, companies outline the factors supporting their security actions. Next, parties implicitly weigh and generally combine the importance and influence of those factors as they relate to the security action decision (this is similar to building an argument in support of the decision). Finally, analysts and security professionals compare security actions and their justifications (typi-

cally the strength of supporting factors and generally 'the argument') with other companies' respective decisions to reconcile conflicting actions.

In this chapter's context, formalization refers to defining the process using a mathematical model in which decision aspects are quantified. The real benefit of this activity is that it allows data to be processed by software (which implements the Model) at later stages. This software is what would handle the automated reconciliation.

Before continuing, a point worth stressing is that it is *security actions of conflicting types* (where for example, one company wants to *transfer* a risk whereas another wants to *accept* it) that are the focus of this work. This area was chosen because it was felt to be the first significant negotiations aspect that companies discuss before moving to consider specific, lower-level treatments (such as security requirements). It is however also accepted that conflicts across companies are possible even when their *action types* are the same. For example, three interacting companies may agree to mitigate a risk but have very different and possibly even contradictory ways of achieving this. Helping and supporting businesses with this problem is also an interesting area but not one of current emphasis in this research.

Lastly, considering the additional time and effort required from companies' personnel to facilitate the reconciliation process described next, it is unlikely that automated reconciliation will be suitable for application to all security action decisions. More details on this point will be given later.

To assist in formalizing the security action decision process, the research field of decision making and specifically Multi-Criteria Decision Making (MCDM) was referenced. Apart from the obvious correlation, this field was seen to be appropriate for this work from numerous perspectives. These included provision of structured methods to design mathematical decision models, a well-established literature base and finally a process that appreciates decisions with multiple inclusive factors/criteria. Furthermore, MCDM models are recognized techniques to support decision making and guide decision makers towards identifying a pre-

ferred course of action [11, 207], a goal of this chapter. There are however a few limitations in the application of MCDM to this research and these are discussed at the end of the section.

Returning to decision modelling and the application of decision making techniques involving numeric analysis, Triantaphyllou [207] identifies three essential steps. These are (i) determining relevant *criteria* (these are defined as the means used to judge an alternative) and *alternatives* (that is, final decision choices), (ii) attaching numeric values to the *relative importance* of criteria and to the *impacts of the alternatives* (also known as *performance*) on these criteria, and (iii) processing the numerical values to determine the ranking (preferences) of alternatives. Belton and Stewart [11] substantiate these steps but also supplement them by emphasizing the additional advantage of using numeric analysis to complement and challenge intuition. This thereby increases understanding of the problem and the final decisions made. Next, the three steps listed are used to define a proposed security action decision model.

The first task was determining the decision criteria and alternatives. For this step, the findings from the completed Security Action Analysis and Ontology Design components of the Solution Model were crucial. As presented in Chapter 6, when making a decision on a security action, salient motivational factors include Laws and Regulations, Contractual Obligations, Business Policies, Security Policies, Security Budgets (particularly very limited budgets) and the related Risk's Severity Level. In terms of the decision model therefore, these six factors can be seen to constitute the decision criteria. Using a similar process and with appreciation of the higher level focus (on action types) of this work, the alternatives identified were Mitigation, Acceptance, Transference and Avoidance security action decisions. These four generic alternatives are also consistent with initial assumptions on security action types in previous sections.

Having defined model aspects, the next step (according to [207]) was attaching numerical measurements. These would be used for calculations and final ranking of security action alternatives. First to be assessed were the criteria val-

ues. In MCDM, criteria values are typically used to represent relative weights of importance. For each criterion therefore, a value between 0 and 1 is stated that symbolizes the *relative* importance of the criterion to the decision maker. Relative means that values also relate to other criteria values stated such that their total sums to 1 (see Triantaphyllou [207] for further detail).

The determination of criteria weights can be done in a few ways, but one of the most commonly used time-tested techniques is based on pairwise comparison. This technique was proposed by Saaty [179] and extensive discussions on it are available in [179, 180, 207]. At a very basic level, this approach focuses on getting decision makers to compare pairs of criteria according to importance and rank them on a defined scale. Normalization methods are then applied to derive relative weight values for each criterion. Standard questions in the technique are therefore, comparing criterion X with criterion Y, whether X is absolutely more important than Y, whether X is moderately more important than Y, whether X is equally important to Y, whether X is moderately less important than Y, and so on.

In terms of this research's security action decision model, there are two options to determine criteria weights. The first option consists of each companies' decision makers using the pairwise comparison technique and entering the values themselves. Saaty in [180] supplies a comprehensive manual example, but such functionality could be built into any proposed software/system. This option has the benefit of directly drawing upon decision makers' perspectives and thereby possibly being a more representative model. Also, each company would have their own weights. The second option also involves pairwise comparison but looks at the provision of standard or default weight values for companies' use, which are based on the principal researcher's knowledge and consulted literature. This bypasses the need for additional work by companies (in conducting pairwise comparisons) by relying on a generic weighting which could be held constant across parties.

As these options each have their benefits, both are expected to be included

in the model (and resulting software) at some stage. This would allow for flexibility in that, if businesses are more concentrated on understanding and having a representative model, they could use a pairwise comparison system feature. However, if they are primarily interested in speeding up the process and using common weights across parties, the second option's feature could be chosen.

To give further insight into the pairwise comparison technique, it was used by this research to determine the standard weights for the six criteria presented above. Considering the substantial detail present in this method however, and the fact that the final weights are of more importance and novelty to this research, the process was included in Appendix C. Drawing on the findings in that appendix, the respective criteria weights are, Laws and Regulations (LR) at 0.409, Contractual Obligations (CO) at 0.285, Business Policies (BP) at 0.111, Security Policies (SP) at 0.116, Security Budgets (SB) at 0.053 and the related Risk's Severity Level (RL) at 0.026. The consistency ratio of 0.0982 (also discussed in Appendix C) indicates a good consistency of the comparison data entered and choices made (according to Saaty [179]).

Briefly commenting on the weights produced, one can see the great deal of importance associated with Laws and Regulations as they contribute just over 40%. Contractual Obligations are also key considerations with roughly 30%. A Risk's Severity Level or a limited Security Budget, however, only contribute relative weights of 2.6% and 5.3% respectively to a security decision. This was an interesting finding because even though Security Budget and Risk Level were crucial factors when looked at individually (if relying on *absolute* instead of *relative* weights for example), when compared to other criteria, they were often seen as notably less important.

Similar to the Risk Level and Budget values, the Business Policies and Security Policies of companies only gained small relative values, 11.1% and 11.6% respectively. This was noteworthy from the perspective that even though policies dictate a business' mission and operations, legal structures such as laws, regulations or contractual obligations are always paramount. Objectively speaking

however, these weights do not claim or profess to be perfect. Different decision makers may arrive at different weights and these are likely to all be valid given they are justified and maintain a good consistency ratio. The reality that different weights will lead to different final decisions/outcomes is also accepted. The advantage of subjectivity in that case is that the final decision will reflect the opinions of the decision makers who defined the weights and therefore would be more catered to their context.

Progressing from attaching numeric values to criteria, the next step was quantifying the impact (hereafter, *performance*) of the alternatives on the criteria. This sought to define how companies' felt about criteria as they pertained to a specific security action decision made. To allow for a more appropriate analysis and emphasis on criteria influence, there was a slight variation from the norm at this stage. Therefore, instead of the usual aim of determining how well an alternative fulfils criteria, the objective was determining how much an alternative was motivated or influenced by criteria. This change was not noted to have any negative side effects on modelling.

Unlike criteria weights, performance values are entirely supplied by businesses' decision makers near to decision time. There were two choices apparent in the literature ([207]) for entities to decide performance values. These were, pairwise comparison in terms of criteria (which leads to relative values) or allowing decision makers to specify absolute values. In the interest of not prolonging or further complicating the decision/transition phase for companies, the latter option was chosen. The Weighted Sum Model (WSM) [207] is an example of a commonly used method that employs absolute values. For this research absolute values in the range of 0 to 10 were allowed for entry by companies to define the extent to which an alternative was motivated by a criterion type.

To ease usability for companies' personnel, a Likert scale [162] would be provided (in the model and resulting software) listing five items, each with corresponding representative absolute values. These are: 1. *Very Important* (score of 10.0), 2. *Important* (score of 7.5), 3. *Moderately Important* (score of 5.0), 4.

Of Little Importance (score of 2.5) and 5. *Unimportant* (score of 0). In terms of a decision therefore, they would be applied as follows: "The Sarbanes-Oxley Act (the criterion) was *Very Important* (the performance) in making the Security action decision to mitigate a risk (the selected alternative)". Another example of the use of this scale and the values is shown in forthcoming paragraphs.

The last step in Triantaphyllou [207] focuses on processing the numerical values to determine the ranking of each alternative. For this task, the WSM method of processing numerical data was used. Other methods were considered but proved either to be too complicated or to require too much information from decision makers for this research's context. The Analytic Hierarchy Process (AHP) [179, 180] is a good example of a popular technique that was debated but later rejected because of its heavy emphasis on pairwise comparisons to determine all input values (both criteria weights and performance values). That emphasis would require a level of user input that would most certainly not streamline framework phase transition. The formula for WSM (sourced from [207]) is presented below. This pulls together all of the aspects and values defined previously.

$$A^*_{WSM-score} = \max_i \sum_{j=1}^{n} a_{ij} w_j, \; for\, i = 1, 2, 3, ..., m. \tag{10.1}$$

where $A^*_{WSM-score}$ is the WSM score of the theoretically best supported alternative, n is the number of decision criteria, a_{ij} is the actual performance value of the *i-th* alternative in terms of the *j-th* criterion, and w_j is the weight of importance of the *j-th* criterion. This formula can stand as the formal model to define the security action decision process, the $A^*_{WSM-score}$ score capturing which company's security action is best supported (has maximum value) and thus might be preferred. Below is a simple example using the proposals thus far.

10.3 A Situation Example

Suppose the situation from Section 4.3.2 where there are two businesses with conflicting security actions for a risk; `Buyer` vying to mitigate and `Supplier` to

accept. Additionally, assume a third company named `Distributor` that wants to insure against the risk and thereby transfer it to an insurance company. Its choice was motivated by the fact that their security policy states it must be handled, but a limited budget prohibits that at this time. `Distributor` has also cited a law that requires those types of risks to be addressed, but it allows for risk handling through insurance. Regarding performance values suppose that companies have individually defined the following.

For `Buyer`'s mitigation-based security action:

- The fact that such threats were evaluated and their possible impact on interactions would be very costly. Also, there is a medium risk likelihood as `Buyer` has been the victim of targeted attacks in the past. — Therefore, **Important** was selected to indicate that the Risk Severity Level criterion was Important in making the decision to mitigate the risk.
- Sarbanes-Oxley Act (SOX) of 2002 (a US federal law) requires that companies should be able to confirm that only authorized users have access to sensitive information and systems. — Therefore, **Very Important** was selected.
- `Buyer`'s security policy strongly advocates the protection of the integrity and confidentiality of all potentially sensitive communications. — Therefore, **Important** was selected.

For `Supplier`'s acceptance-based security action:

- There is very limited security funding. — Therefore, **Very Important** was selected.
- Deemed possibly unlikely that this risk would materialize as existing transport-level measures are thought to provide adequate security. — Therefore, **Very Important** was selected.

For `Distributor`'s transference-based security action:

- The security policy of `Distributor` states that security risks to confidentiality of company data classified as Private, must be handled. — Therefore, **Important** was selected.

– Cutbacks in the company have lead to a limited security budget at this point — Therefore, **Very Important** was selected.

– A law exists that emphasized that risk should be handled. The law permits that handling via insurance is an allowable alternative — Therefore, **Important** was selected.

The following decision matrix puts the data above, criteria weights and respective numeric performance values into context.

	Criteria					
	LR	CO	BP	SP	SB	RL
Alternatives	(0.409	0.285	0.111	0.116	0.053	0.026)
Mitigation (by `Buyer`)	10	0	0	7.5	0	7.5
Acceptance (by `Supplier`)	0	0	0	0	10	10
Transference (by `Distributor`)	7.5	0	0	7.5	10	0

To apply the WSM formula, the scores for the three alternatives are:

$$\begin{aligned} Mitigation &= 10 \times 0.409 + 7.5 \times 0.116 + 7.5 \times 0.026 = 5.155 \\ Acceptance &= 10 \times 0.053 + 10 \times 0.026 = 0.79 \\ Transference &= 7.5 \times 0.409 + 7.5 \times 0.116 + 10 \times 0.053 = 4.4675 \end{aligned}$$

Therefore the best supported alternative (in the maximization case) is Mitigation, `Buyer`'s choice. The SOX Act (a law) supporting their decision being a key factor due to the large weight assigned to Laws and Regulations.

This example presents a simple application of the decision model defined. From that illustration, it is apparent that the model works on the basis that the action with the 'strongest' support is preferred. This seeks to be similar to the manual negotiations process where the best justified or supported action is chosen. The next section continues discussion of the proposed model and presents its most notable limitations.

10.4 Discussion and Limitations

One of the greatest novelties about the newly proposed model is that it tries to formally accommodate a number of previously under-represented factors in the decision process. This advances existing literature and approaches where primarily, only risks and risk levels were scored and valued. Whilst it is understood that risks allow the easiest formal and numeric (especially monetary, in terms of loss potential) definition, the various other factors in a security action decision process should also be considered. This research attempts to provide a logical start to the high-level inclusion of such factors. Having discussed the proposed reconciliation model in detail in previous sections, its main limitations are now outlined. These discuss known practical limits in the decision model, but additionally areas that surround the further formalization of crucial decision factors.

To begin, one of the main limitations in the model is in its actual application. Ideally, companies will enter weight and/or performance values into the model (or a software tool embodying the model) at the same time they input risks and security action data. Apart from being convenient, this is particularly beneficial because the motivation data and decision model values will be fresh in decision makers' minds. Here also is the implicit assumption that companies' personnel will enter accurate values and not attempt to deceive the system to have their security actions chosen. The real limitation at this point therefore is that for the model to work, reconciliation-related data will need to be provided for *all* security action decisions. This is likely to lead to quite a long process for companies which will only pay off in situations where there are actually conflicts in matched security action decisions. Furthermore, spending all of that time somewhat contradicts this research's goal of easing phase transition. Arguably, companies could use the time and effort here to engage in the existing manual negotiations process.

Two possible solutions which attempt to minimize this problem have been identified. The first solution advocates that companies run the Comparison sys-

tem feature initially to determine conflicting security actions. Once these have been identified, companies should then enter reconciliation-related data for risks only related to those actions. A Reconciliation system feature (based on the model in this chapter) could then be applied on this smaller subset of actions to supply possible decision choices.

The next solution tries to be preemptive and thus relies on businesses initially selecting a set of risks (this could be done at the risk exchange stage in Figure 5.2 in Chapter 5) they want specially considered. Reasons for this might include perceived or likely difficulty in arriving at a common treatment decision across companies, or a desire for in-depth assessment of risks and the understanding of the resulting treatment/action. Once this smaller set of risks is selected, only security actions based on these risks would be passed through the Reconciliation system feature.

Unfortunately, neither option listed is ideal. This is because the former relies on decision makers remembering motivation and influences from an RM method applied days or weeks before, while the latter option does not guarantee that risks with actual conflicting actions will be emphasized. At this point, however, the former option is preferred as it focuses more on conflicting actions and their reconciliation, a core goal of this work. To aid in the value definition process, it is hoped that personnel will be able to use listed supporting factors (criteria) to remind them of possible motivation and influence (performance) values. Whilst option one is preferred, the latter option is not totally discarded as it has benefits in facilitating understanding of risks and security action decisions across companies. Application of this option might be followed up as a future system feature.

The other general limitation of the decision model is that it does not account for multiple factors/criteria of the same type. For example, if a company has four laws supporting a security action decision instead of one, the model should reflect this, potentially by a greater weighting or performance value. A greater influence or 'argument' would be the likely behaviour in real-world ne-

gotiations. Currently however, the decision model does not. Possible solution options to accommodate this include having extra parameters for each additional factor, or building such aspects into the performance Likert scale; for example, only allowing *Very Important* to be selected if three or more factors/criteria of the same type support a security action decision.

Another interesting aspect and possible restriction on the model is that it regards decision alternatives in an isolated manner. For example, assume there are ten companies in a scenario, nine desire to transfer a risk, but one company opts to mitigate it. Furthermore assume that the mitigation company has the maximum calculated value (that is, $A^*_{WSM-score}$). According to the model, all companies should adopt this decision. Even though this occurrence is a possibility, in the real-world it is probable that majority vote might triumph.

One way to tackle the isolation issue might be to sum decision values from companies with the same security action type. Then, compare these totals and choose the action with the maximum value. Provided the nine businesses above had a summed total greater than the total of the one, their action type would be selected. Albeit accommodating, there is one small caveat to even this technique however. This lies in the reality that the majority vote might always prevail even in situations where it might not be best. Future work would therefore have to investigate, monitor and balance this closely.

The penultimate limitation relates to the complexities of the security action decision process and the interrelation between its components not addressed by the model. An example of this is a situation where a company has a single action that addresses ten security risks. Arguably this action should receive an increased weighting or performance value simply because of the fact that it covers so many risks. Currently however, the model does not allow for this.

The problem in the model therefore is that it focuses on security actions on a risk-by-risk basis and not more generally as it is likely to be considered in the real world. There is also the argument that the specific risk or specific risk's severity level plays a part, instead of just noting the generic Risk Severity Level criterion.

Thus, possibly in situations where a risk has a severity level of 'High', this should be given a slightly greater (or lower, depending on the context) weighting than where a severity level of 'Low' supports a company's security action decision.

Additionally and more from an interrelation perspective, the model does not support links across factors, risks or treatments, nor is it retrospective. Concerning the last point, the model can suggest that companies mitigate a risk instead of accept it, but it does not look at the impact that decision might have on other risks or factors. For example, such a decision might mean that there is less money to spend on mitigating another risk, therefore that other risk may now need to be avoided or accepted. These are complex issues not addressed by the model but which represent real-life negotiations and discussions. Further comprehensive work is needed in this area to see how these aspects can be captured and to what extent.

The last debatable aspect of the model relates to decision makers. In MCDM techniques, there is typically a single, or group of decision makers concentrating on a specific decision. If it is a group, they first need to agree on input values (typically through consensus or voting) then enter them into the approach. The approach processes these values and selects a preferred alternative based on maximum scores. The same decision makers therefore provide all the input values.

In this research's model however, each company goes through the decision making process individually and then at the end supplies their security action summation score (formally, $\sum_{j=1}^{n} a_{ij} w_j$) to the system. This score is then compared with other companies' decision scores regarding the same risk and the maximum is chosen as the preferred or best supported alternative. Therefore, different decision makers supply input values. Although the use of separate decision makers seems like a useful and valid application of the MCDM technique, no literature could be found which also applies it. Further work therefore should encompass the evaluation of this particular application and its ultimate viability.

To briefly summarize this automated reconciliation section, there is still a

great deal of work to be done in creating a highly representative, formal decision model. This chapter provided a well-grounded start to that process by identifying a basic model which included a number of previously under-represented decision factors. Even though these factors are difficult to value and accommodate, they form key parts of the decision process and should be duly represented.

In looking towards automated reconciliation therefore, the biggest challenge will be in identifying the minimum level of data input and time commitments necessary, which leads to the greatest, most useful security action reconciliation results. After all, the focus is easing phase transition and not complicating or prolonging it further. There must also be an appreciation however that sometimes, automated reconciliation will simply not be possible or feasible. For example, take the situation where two or three companies have mandatory laws that support conflicting security actions. Or, consider the case where companies have very high scores or very close total scores. Boundaries will be needed in the system to flag situations and inconsistencies like these so that they could then be discussed by personnel. Output point (ii) in Figure 5.2 from Chapter 5 could be used as a channel for this.

Finally, as identified previously, there is additional scope beyond reconciliation for aiding understanding of security action decisions. Weighting and performance data provides a rich source of information which explicitly defines companies' perspectives. This information could be used to support complex or detailed negotiations processes, as opposed to streamlining phase transition.

10.5 First Impressions on the Decision Model

The automated reconciliation of security actions was explored for three main reasons. Firstly, the favourable feedback on the current SASaCS tool, next was the additional time likely to be saved by businesses in negotiations and finally, the overall increased productivity that could possibly result. The last two of these points were especially relevant noting the importance placed on time and

productivity by interviewees in Section 9.3.2. This is an interesting proposal from a research perspective, but because this tool is aimed at industry use, gaining some real-world feedback even at this early stage would be very useful. This would help to put the proposed enhancements of the tool into a practical context and give a first impression regarding feasibility.

To attain real-world feedback on the decision model enhancement, questions on the model and its aims were therefore posed to security professionals during the interviews mentioned in Chapter 9. Questions were included in the Solution Model section and can be seen in question 6 in Appendix B. Below a brief analysis is conducted on the feedback gathered. This analysis is in line with the assessment and discussions in the previous chapter.

At a general level, security professionals regarded the notion and process of automated reconciliation as 'interesting', but expressed that a great deal of analysis and proofing would be required. John, a security professional of 10 years working for a leading international IT and consultancy services company, summed up interviewees' views in his statement, "it's an interesting idea, but the exact nature of the formula or the risk factors, how that would work, I think I'd want to see more examples, to prove to me that it works and makes sense". Finally he added, "but I think it's an interesting idea worth exploring". This and similar views from most professionals are taken to support the feasibility of future investigations towards automation.

There was a single view not in support of full automation. This came from Mark, the most experienced security professional amongst interviewees. He strongly felt that the goal should be towards automation to aid in decision making and complementing understanding—not therefore, in providing definitive answers for security. Mark stated, "I'm not a firm believer in, you press the button for risk assessments and you get the answer out". This opinion was likely due to Mark's view that risk assessments and some aspects of security were an art and not a science, therefore human aspects still need to be present. This was a salient point to this work, because it acted as a gentle reminder of the continued need

for some human presence even in this level of the negotiation process.

Lastly, professionals agreed with the notion of *degrees of importance* of factors/criteria (such as Laws or Policies) in terms of a security action decision. For example, a relevant law may influence the treatment of a risk more than a related security policy. Interviewees' agreement therefore acted to directly support the performance or impact values (represented on the Likert scale) discussed earlier in this chapter.

In summary, a majority of interviewees viewed the proposal as interesting, but noted that it required a great deal of analysis and proofing. The main opposing perspective referenced the need for humans to actually make the reconciliation decisions (instead of a tool). This opinion was linked to the perception that risk assessments and aspects of security are more of an art (therefore somewhat subjective and mutable) than a science (strictly defined). Although this is a valid perspective, taking into account the positive feedback from other professionals, future work is likely to concentrate further on automation. Even if tool-based reconciliation is not used for definitive solutions to security action conflicts, a tool that could present an initial solution that then would need to be ratified by a human, would streamline negotiations even more than the Solution Model allows now.

10.6 Summary

The aim of Chapter 10 was to explore whether the Solution Model and SASaCS tool could be further enhanced to enable them to reconcile conflicting decisions across interacting businesses. To achieve that goal, an attempt was made at formally modelling the security action decision process which companies engaged in. For support of the formal modelling activity, the field of multi-criteria decision making was referenced. This was done both to guide modelling and with the aim of creating a more appropriate, grounded formula. Once this was completed and a model defined, a simple example was presented to illustrate model application.

Next, the model was discussed and its limitations highlighted. Even though the model itself is viewed as a novel proposal which challenges current research thinking, its limitations are slightly more important here. This is because they identify key issues for this (especially in terms of future work) and other research which attempts to formally define such a decision process. Lastly, the chapter presented feedback from industry-based security professionals. Generally, professionals showed interest in the model, however they noted that much more testing and analysis would need to be done. This supports feasibility of the ideas and thus the need for future research in this area.

Chapter 11 concludes this thesis and presents the main research contributions. Having discussed these aspects, the possible areas for future work are outlined. These cover topics that this doctoral project was unable to address due to resource and time constraints, but also new research ideas resulting from lessons learnt and current developments in the field.

Chapter 11

Conclusions and Future Work

Information security is, in the terms of the cliche, a journey, not a destination.
— Alan Calder and Steve Watkins

11.1 Introduction

A key goal of this thesis was the investigation of the e-business Web services security field, and the subsequent proposal and assessment of a novel solution approach to enhance existing security and trust. Having achieved this goal in previous chapters, this chapter ends this thesis by presenting project conclusions, main contributions to the research discipline and ideas for future work.

11.2 Conclusions and Discussion

In concluding any research project, one of the most important aspects is assessing the achievement of original research aims and objectives. As defined in Chapter 1, this project aimed to:

– Consider security in a business context and, with regard to its various components (policies, processes, technologies and so on), to develop a joint approach in which collaborating e-businesses could achieve an enhanced and a more comprehensive security solution for their Web services interactions; and

– Evaluate the suitability and strength of this approach's proposals in aiding businesses to reach the requisite levels of enhanced inter-organizational security and trust.

Commenting on the work completed, both of these aims were successfully accomplished through the fulfilment of their respective research objectives as listed in Section 1.2. To recap, the project commenced with the first objective and its examination of the e-business and WS security fields (Chapter 2). Having identified gaps in existing research relating to an overly reliant emphasis on technology for security and approaches to e-business security that were too individualistic, the next objective led to the development of BOF4WSS to address these issues (Chapters 3 and 4). The advantage of BOF4WSS was found in its provision of a joint approach in which collaborating e-businesses could achieve an enhanced and more comprehensive WS security solution.

The scope of the project was then reduced by the next objective to the more manageable research problem of supporting businesses' use of BOF4WSS. In particular, the problem area of interest centred on the transitional issues faced as companies moved between the Requirements Elicitation and Negotiation phases (Chapter 5). To address these challenges, the following two research objectives aimed towards the proposal and development of a detailed solution approach and prototype tool. These steps resulted in the Solution Model (Chapters 5 and 6) and SASaCS tool (Chapter 7)—two of the main contributions of this research.

The next goal focused on the evaluation of the Model and tool. Specifically, this objective resulting in an assessment of their compatibility with existing techniques (Chapter 8) and their ultimate usefulness (examined using interviews with security professionals) in supporting phase transition (Chapter 9). BOF4WSS was also assessed using the interview technique to gather expert feedback on its suitability and strength (Chapter 9). Fulfilment of the final research objective defined in Section 1.2 came through the critical analysis of the data gathered above. This allowed conclusions to be made regarding adequacy of the Solution Model and tool in supporting the framework, and viability of BOF4WSS and its

activities in enhancing inter-organizational security and trust (Chapters 8 and 9).

Summarizing the results attained in both evaluation chapters (that is, Chapters 8 and 9), there was a noteworthy degree of evidence to support the suitability and strength of this thesis' proposals (those proposals being, BOF4WSS, the Model and its tool). This evidence was found in the detailed specialist feedback from industry-based security professionals and the technical-level findings from the evaluation of the Model and tool's compatibility with existing approaches. The favourable results of this evaluation directly support the first research project aim and its goal of creating an enhanced security approach.

Briefly considering the research question in Section 5.2.3 which guided the Solution Model's creation, the evaluation findings show that the framework's phase transition process can be automated to a large extent. This automation is such that by use of the Model and tool, the transition process could be expedited and productivity for companies significantly increased.

Reflecting on interviewees' feedback in detail, they viewed various components of BOF4WSS (such as collaborative security, level of detail and the ISS) as beneficial and very likely to enhance current security and trust approaches. Feedback on the tool created to support the framework was also very encouraging. The main advantage which surfaced was its use in significantly easing security negotiations across companies and aiding productivity at a crucial and normally lengthy stage. One of the most salient points arising from that expert feedback was that the Model and tool were seen to provide a much needed bridge between business and technology, thereby accelerating the creation and adoption of technology solutions. This also meant that the Model and tool could accelerate the adoption of BOF4WSS itself by easing its application to business scenarios. These were positive and noteworthy findings originating from security professionals working within very related areas in industry.

It is worth noting that in both evaluation chapters, some negative results were apparent. For example, the compatibility evaluation showed flaws in the Model and tool because they did not provide fully adequate mappings. Addition-

ally, the theory that risks were the sole basis for security actions was strongly challenged by the EBIOS method. Even though these were notable concerns, as discussed previously they were not seen to significantly refute this research's proposals.

Before moving on to outline research contributions, there are a few important aspects to be stressed largely pertaining to the framework. The first aspect is in relation to interview results. This is to emphasize the point that BOF4WSS is likely to be best suited to certain situations and scenarios. Particularly, this includes business scenarios where there is strong commitment to businesses goals, a great degree of executive sponsorship, they are high-value projects (and this value drives the need to do whatever necessary to complete the task properly), there is history and existing trust between companies, or there is symmetry in business interactions. In situations where these aspects are absent, companies could still realize the enhanced security and trust possible with BOF4WSS. It is expected however that in these cases the framework might prove more difficult to apply and more effort will be needed to attain the maximum benefits.

The next two aspects embody limitations of the framework and were identified in project reflections. The first relates to the longevity of BOF4WSS and the perspective that it risks being outdated quickly. This is because BOF4WSS is arguably not as abstract as a framework/methodology should be. Therefore, even though identifying standards, laws, tools and technologies is beneficial as it gives e-businesses detailed guidance and insight into online WS interactions, it ties the framework too closely with current practices.

The risk of being outdated is a valid concern and the only solution to it that is in line with the original aim of the framework is to update BOF4WSS periodically. This would allow updates in relevant laws, tools and so on, and also enable any structural changes to be made based on field tests and adopting companies' feedback. Updating frameworks (and even more abstract frameworks) is an accepted reality as is exemplified in the various versions of the industry accepted model, TOGAF [202]—currently up to version 9. Furthermore, consid-

ering the volatility of the online security field, the updating of all security-focused models is imperative.

The second limitation results from the framework's basis on the Waterfall Model (WM). Even though this model is believed to be the most suitable (for reasons identified in Section 3.2), there are reservations about the time taken for overall project completion, and flexibility and turnaround time within individual phases. One possible way to address these issues is to incorporate quicker and more flexible development techniques within specific phases of the framework.

Additional benefits with more flexible techniques might also be attainable in the areas of project risk management (common with iterative methods) and purpose-built support tools (apparent in methods such as rational unified process). Hines et al. [80] provide a good start for this with regards to integrating agile methods in the WM. Such techniques will need to be evaluated in depth before being included in BOF4WSS to ensure that structure and benefits of the WM to large or critical system projects are not affected.

Having briefly reviewed the achievement of research aims and discussed results, the one noteworthy weakness in the investigation is mentioned. This weakness was introduced originally as a limitation in Chapter 1 and relates to the reliance on methods other than a thorough case study to evaluate BOF4WSS' proposals. Even though limitations precluded the use of this technique and interview-based assessment was very useful, a case study analysis still remains the ideal form of evaluation. That analysis has the benefit of allowing a real-life probe into BOF4WSS' use and application to an actual scenario. This would facilitate a comprehensive assessment of its acceptance by companies and their personnel. Furthermore, it would enable a thorough evaluation of the framework's suitability and strength in enhancing the cross-enterprise security solution proposed.

Overall, this project's investigation to provide an enhanced and more comprehensive security solution for collaborating e-businesses yielded positive results and research progress. This is both in terms of achieving the research aims and

objectives, and outlining proposals which could be favourably evaluated. The application of these proposals to real-world business scenarios is therefore seen to supply numerous advantages for companies regarding their security and also to some extent in fostering cross-enterprise trust. The next section continues the discussion on proposals as these also constitute the core novel aspects of this project.

11.3 Research Contributions

There are three main contributions of this research. The first contribution is the business-oriented framework for enhancing Web services security for e-business or BOF4WSS. This framework was developed particularly to address the outstanding need for a comprehensive, multilayered security approach for collaborating e-businesses that use WS. As was shown in Chapter 2, there currently exist a number of security advances such as [26, 74, 75, 187, 189, 194, 211, 217, 231]. The problem with these techniques however is that they are either too isolated (concentrating only on *one* business' view to security), or too technology-focused (only considering what technical mechanisms or standards are needed) to properly address the security and trust issues which WS introduces for collaborative e-business.

The prime novelty of BOF4WSS is the emphasis on providing an expanded formalization of a development methodology that stresses security and trust. This would accommodate multiple autonomous businesses working together. To address the outstanding issues from Chapter 2, BOF4WSS aims at considering the full nature of WS and its security implications within e-business, appreciating the real-time inter-organizational security issue now faced by interacting e-businesses, and finally promoting the use of a collaborative approach to provide enhanced levels of security and trust. In line with these aspects, the framework is furthermore initially targeted towards a special set of companies and interactions as discussed in Section 3.12.

As the conclusions from the previous section stated, the framework proposed was seen to achieve its aims. Critically speaking however, BOF4WSS was noted to be suited to a particular set of scenarios and situations—these were an update to those intended in Section 3.12. This reality is not seen as a negative, but rather a positive as it acted to provide a clearer target scope for framework use where its benefits should be better realized. The overall positive evaluation results gained reinforce this approach as a contribution to research and specifically, the e-business and WS security discipline. The results additionally help to highlight the need for higher-level, more comprehensive approaches to security and trust within the WS' e-business environment.

The next contribution is the Solution Model and its respective implementation (SASaCS). This model was created to ease the difficulties typically incurred by collaborating businesses as they met to discuss and negotiate on their joint security needs for a scenario. These aspects were largely considered within the context of BOF4WSS, however there might be wider applications as cross-enterprise negotiations between companies is a common-day activity. Future work might consider these other applications.

To address the security negotiation difficulties, the Model had three novel components. First, an ontology which provided a common understanding of the security needs domain to tackle semantic issues across companies. This model was the outcome of a detailed assessment of the existing security field and key security standards. The next component was a shared risks catalogue to enable security needs and actions across companies to be automatically matched and easily compared. Then finally, there was a purpose-built XML-based language which defined a common template/format. This would express security actions (and related factors) and also guide companies via a data entry screen interface on the security information they should prepare as they come together. Other possible applications of the language independent of this research include, using it as a document exchange format for business-level security information or for storage of risk management data from various RM techniques. Each of the com-

ponents above was novel in itself considering the lack of approaches to solve the specific problems (see Section 5.2.2) they targeted.

In terms of the evaluation, the Model and tool's use were assessed both at technical (Chapter 8) and interview-based (Chapter 9) levels. The favourable results received acted to cement proposals as noteworthy contributions to the research field and practical security advancements.

The last contribution encompasses the decision model which was aimed at automated reconciliation of security actions. This model and the limitations highlighted are important developments to the research field for two reasons. First, they provide initial steps towards a means to automatically reconcile conflicting, business-level security actions. This is critical in terms of the possibility for significantly increased productivity for businesses. Secondly, the decision model seeks to include various decision factors/criteria not usually incorporated within models in a formal or numeric sense. For example, seeking to numerically define influences of laws or strengths of security decision arguments.

As Chapter 10 was mainly exploratory, no evaluation was conducted. A few model limitations were however retrospectively identified. These add to the contribution itself by defining areas of contention that will need to be investigated further to progress the model and similar decision approaches in the security field.

11.4 Future Work

There are various interesting avenues for future research. In this section the most relevant are discussed. The first of these avenues looks towards improving this work and overcoming key research limitations faced.

The evaluation of BOF4WSS is one of the prime, initial areas for betterment. This is in terms of providing a more thorough evaluation of the framework's use and strength in enhancing inter-organizational security and trust. Future work would therefore include enlisting a set of entities willing to apply the framework to their WS-based business scenario. A scenario with 3–5 parties is

envisaged to allow for a sufficient and manageable amount of feedback for a detailed case study analysis of BOF4WSS. Furthermore, this analysis would enable future updates to the framework's proposals to be made based on new evaluation findings. It is hoped that as a result of the industry contacts (security professionals) made by the principal researcher during this project, along with the peer-reviewed publications covering the framework, there would be an increased chance of finding willing companies. During the case study evaluation, the Solution Model and SASaCS tool could also be applied. This would allow for another, more practical evaluation of these aspects to give additional feedback on their suitability, usage and strength in supporting collaborating parties.

The second area to improve current work would be considering agile methods (which are light-weight, quick and flexible) of development again. In Section 3.2 these were assessed but were regarded as inappropriate for a number of key reasons. Noting reservations about the time taken for overall project completion, and flexibility and turnaround time within individual phases within BOF4WSS however, the incorporation of agile methods might be imperative. Section 11.2 adds to this by presenting some advantages of these techniques.

Future work would therefore aim to determine in which phases agile methods might be properly applied, what are the best agile techniques to apply, and how useful is their application; this is specially within the context of the framework. It is expected that these methods will be more suited to areas such as Requirements Elicitation, Analysis/Architectural, Systems Design, and Development and Testing. This is because these methods tend to be very practical and hands-on. There is also very likely to be overlap, with one method covering numerous phases. The implications of this to the structure of the framework will need to be studied.

The final area for improvement relates to the Solution Model and addressing the problems identified in this project's evaluation. Proposed changes were presented in Section 8.7. These included widening the definition of the term 'Security action' and using Conflict types to combat the reality that risks alone do

not form the basis for Security actions. Questions going forward would therefore seek to investigate whether proposed changes in the Model and its implementation (SASaCS) are sufficient to tackle the problems. Special emphasis will also be placed on finding a common Security actions basis that encompasses risks and constraints. If this could be found, automatically matching all Security actions across disparate businesses would again be a feasible goal.

Progressing from the improvements that could be probed, there are a few natural extensions to this research for further work. The most innovative aspect is the automated reconciliation of conflicting security actions. The work to be done here would specifically focus on addressing the limitations and areas of contention identified in Section 10.4. For example, finding ways to account for multiple factors/criteria of the same type and appreciating the interrelation between decision model's components. Some suggested solutions were already discussed in that section. Once these issues are tackled, the model could be evaluated in an actual business scenario. The case study analysis expected in future work would supply a prime area where data from companies could be gathered for this assessment.

Other areas of interest for extensions include investigating where else system support might be appropriate for BOF4WSS and looking at how the framework might be updated. For the former aspect, the case study would aid significantly as it would highlight areas of difficulty and/or problematic phases where system support might be useful. This will be an intriguing endeavour when combined with the agile approaches to be considered for incorporation in the framework.

Returning to the latter area regarding framework updates, work would focus on one of BOF4WSS' limitations discussed in Section 11.2. The main problem there was the reality that the framework risks becoming outdated. As mentioned in that section however, the only way to address this issue is for future work to look into updating the framework periodically. Below the discussion continues by looking into open questions to the research community posed by this research and its results and conclusions.

The largely positive conclusions reached by this thesis serve as an indicator that more work needs to be done in these areas by the research community. Beyond the identified benefits, the evaluated framework shows that there is motivation for higher level and more comprehensive approaches to inter-organizational security and trust. As businesses look more towards WS to streamline cross-enterprise interactions, there will need to be more specialized approaches concentrated on holistic security. These could encompass specialized WS security risk management techniques and collaborative security approaches. Researchers might also seek to define approaches that have wider applicability than the situations targeted by BOF4WSS. Conversely, there is also the possibility of new work focusing, like BOF4WSS, on specific types of scenarios. For example, small and medium-sized enterprises (SMEs), particular industries (such as financial) or companies who require very high degrees of security and business-level trust.

As the Solution Model and SASaCS tool cover a very specialized target area, there are not many open questions for general research. One possibility however is the analysis of the use of very formal techniques such as OWL to define the ontology instead of the XML-based implementation in SADML. In this project, OWL was seen as surplus to requirements but its reasoning and logic capabilities might warrant its reconsideration especially if they could feed into another automated decision model. This leads to the other open question which focuses on the investigation of different approaches to creating a security actions decision model. This research outlines one method, but other techniques either based around OWL, or the use of techniques other than the Weighted Sum Model (which uses absolute values for criteria) could be assessed. These would all aid in the advancement of the research field.

Appendix A

SASaCS Tool

A.1 Introduction

This appendix briefly presents some of the implementation specifics of the SASaCS tool. Section A.2 displays screenshots of the user-friendly output report from the tool's comparison features. The following sections depict the database Entity Relationship Diagram (ERD) which was developed to support the SASaCS tool, and provide a description of the database tables. As the ERD is based on the ontology (from Chapter 6), common concepts can be taken to have similar meanings. In some cases however, ERD-level concepts have been modified slightly to better suit the low-level database implementation. Section A.4 presents a draft of an updated ERD which accommodates the shortcomings in the original design from Chapter 7.

A.2 Prototype Screenshots

The figures in this section supplement those provided in the main thesis to highlight the detailed output possible with SASaCS. Figure A.1 is a screenshot showing the data on a particular risk, namely `LR1`, which companies had decided to treat differently. As the tool is using the detailed output setting, the output (shown in the diagram) also displays the varying reasons which influenced

each company's individual security action decision. One of the prime benefits of the tool in this regard is its ability to immediately identify differences in risk treatment, and the likely reasons for these differences. This is also done in a user-friendly, colour-coded and unambiguous report format.

Figure A.2 enhances the comparison output by allowing companies to view the security action that addresses a particular risk from another perspective, that is, side-by-side with the other companies'. This enables security professionals and analysts to quickly visualize all the risks being addressed by a security action/risk action, along with various other relevant factors.

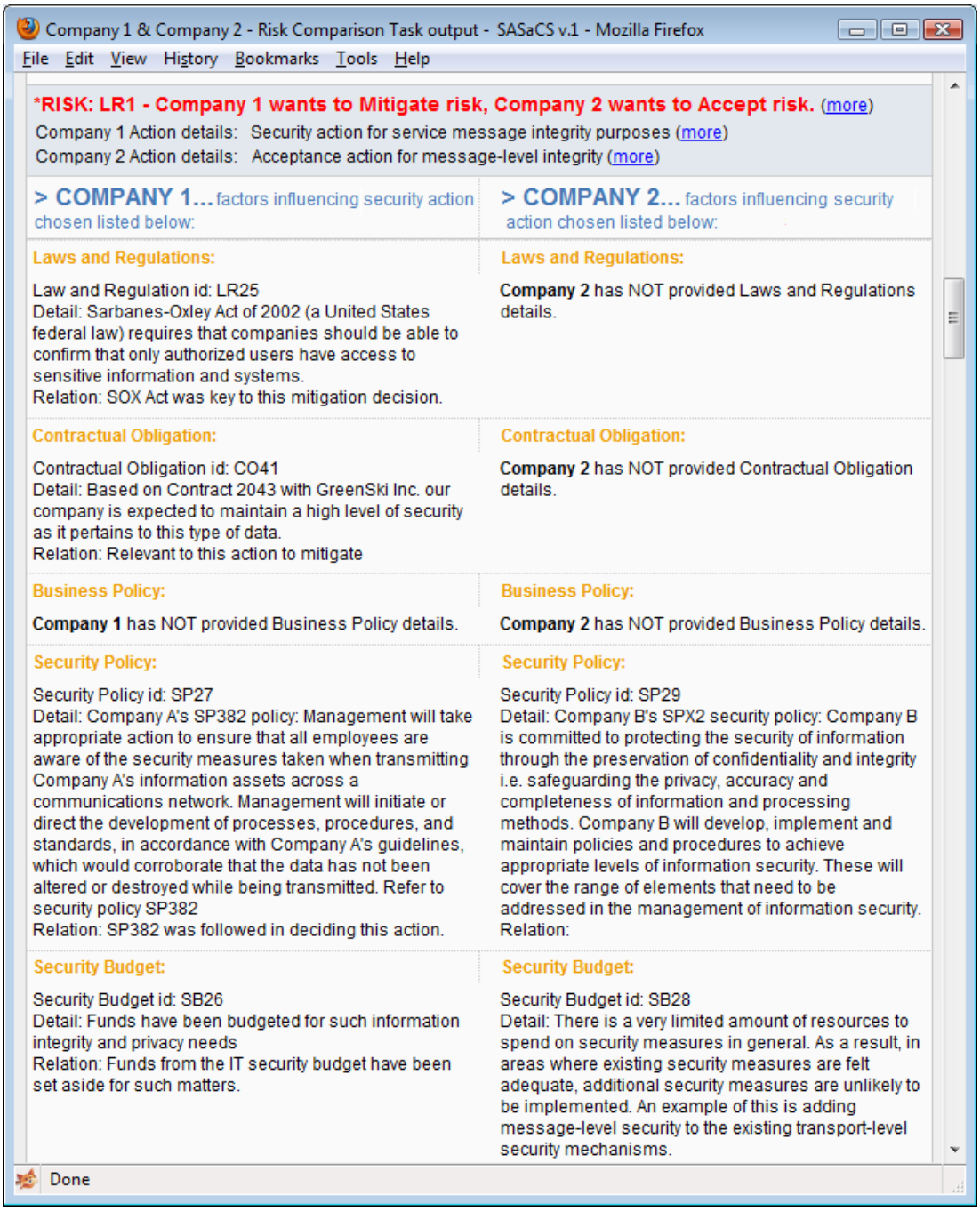


Figure A.1: SASaCS Security Action Comparison detailed output screenshot

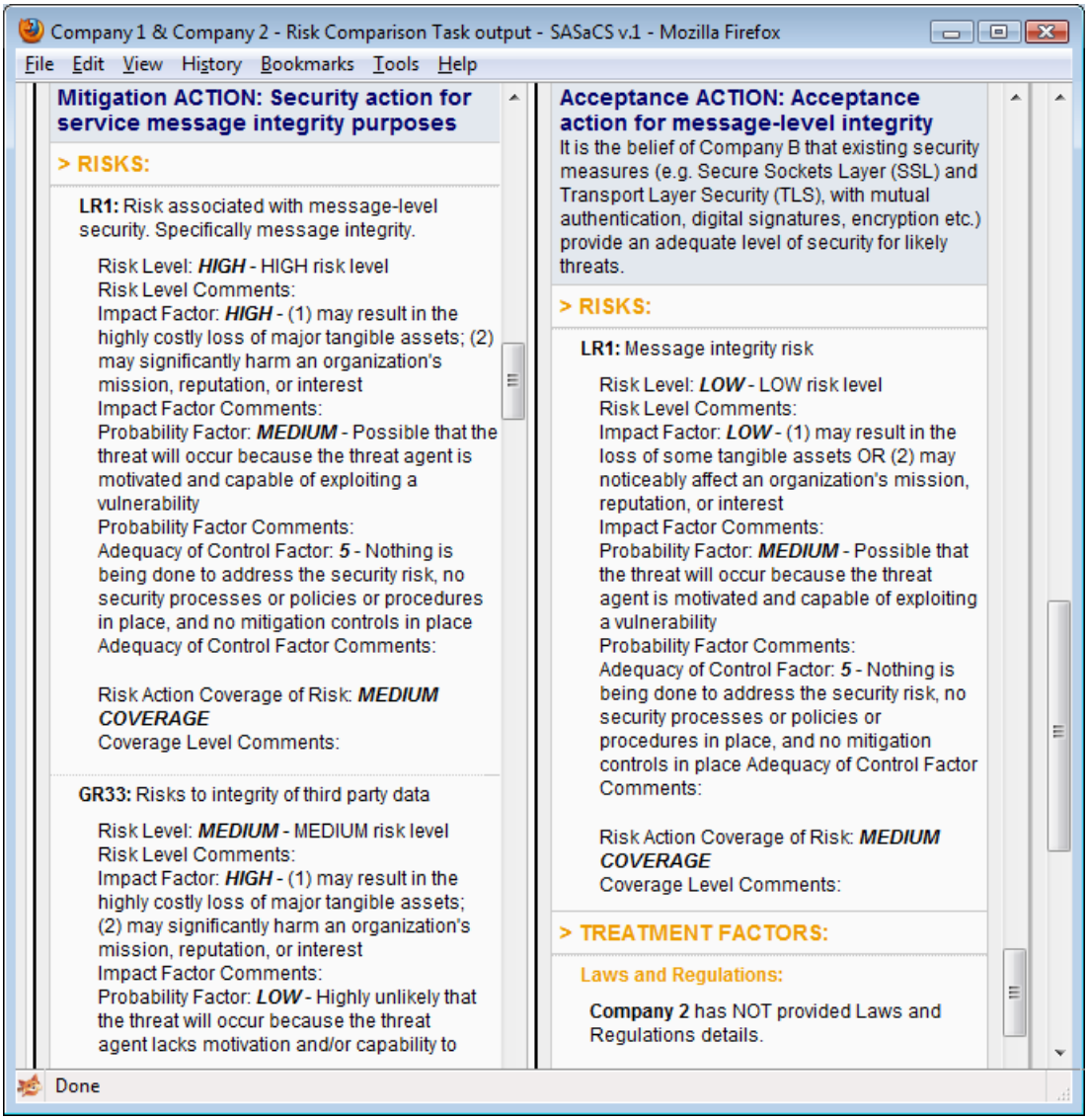


Figure A.2: SASaCS Security Action Comparison detailed output screenshot (2)

A.3 Entity Relationship Diagram

Asset: Defines the main assets

AssetSecurityAttribute: Allows the security attributes of an asset to be defined

ActionTreatmentFactor: Defines treatment factors (for example, laws, business policies and so on) that influence a security action/risk treatment choice

Project: Projects are defined for each RM/RA undertaking or scenario

ProjectRisk: Defines a risk that has been selected for use in a particular project

ProjectRiskAction: Defines the risks which a security action/risk treatment choice 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: Same definition as the ontology

RiskAction: This is taken to be the same as security action in the ontology. Readers should view security action, risk treatment choice and risk action (RiskAction) as the same

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: Same definition as the ontology

Threat: Same definition as the ontology

ThreatAgent: Same definition as the ontology

ThreatSecurityAttribute: Allows the security attributes which a threat affects to be defined

TreatmentFactor: Defines the elements that affect the treatment of a risk

TreatmentFactorType: Lists the generic types of treatment factors

Vulnerability: Same definition as the ontology

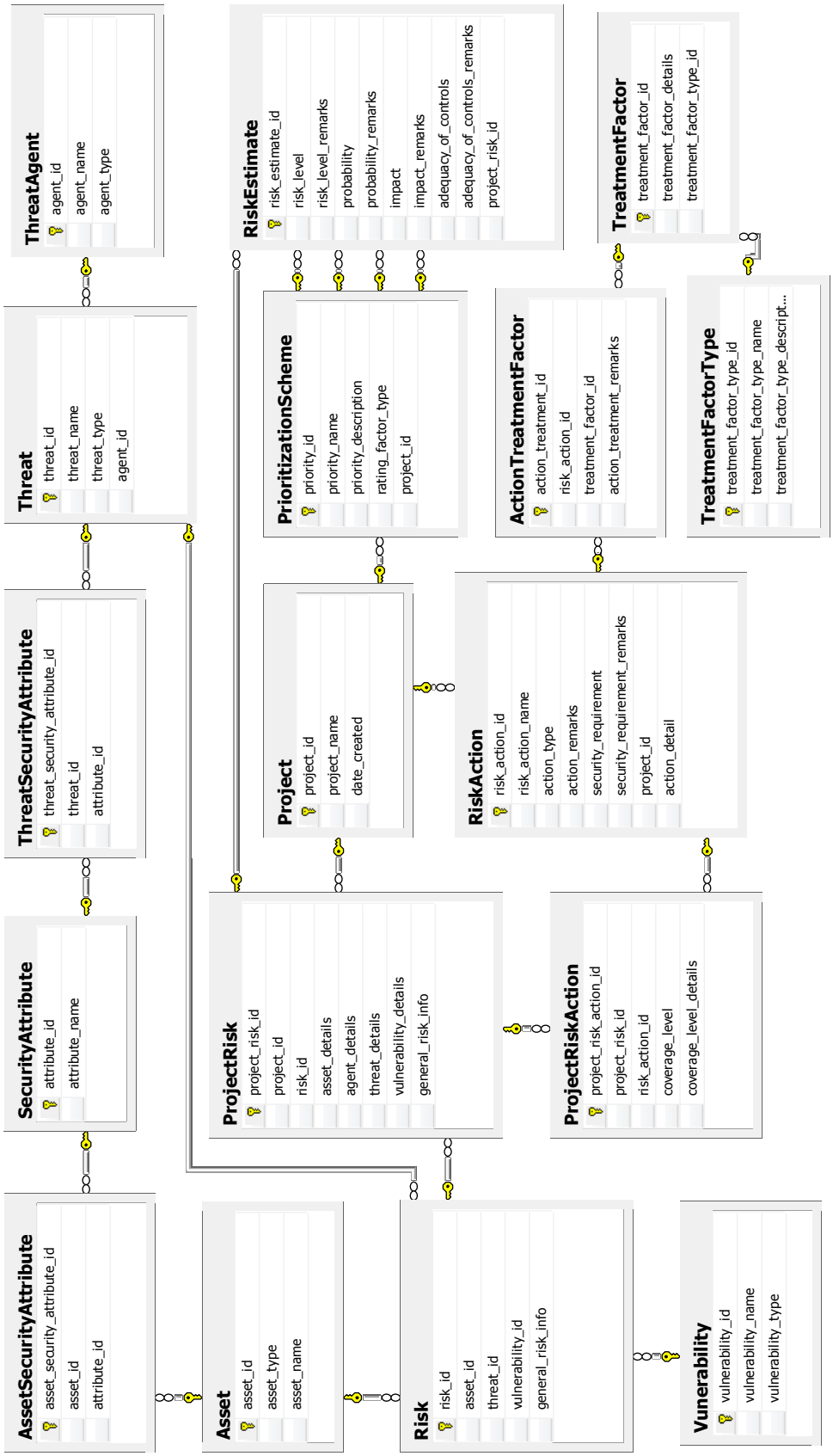


Figure A.3: Entity Relationship Diagram of the SASaCS tool

A.4 New Draft Entity Relationship Diagram

This section presents the draft of a new ERD to support the SASaCS tool. It accommodates updates suggested after the evaluation conducted in Chapter 8. The new or modified tables are described below.

Constraint: Defines limitations faced by the organization. These limitations can influence the treatment of a risk or give rise to a Security action themselves

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

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

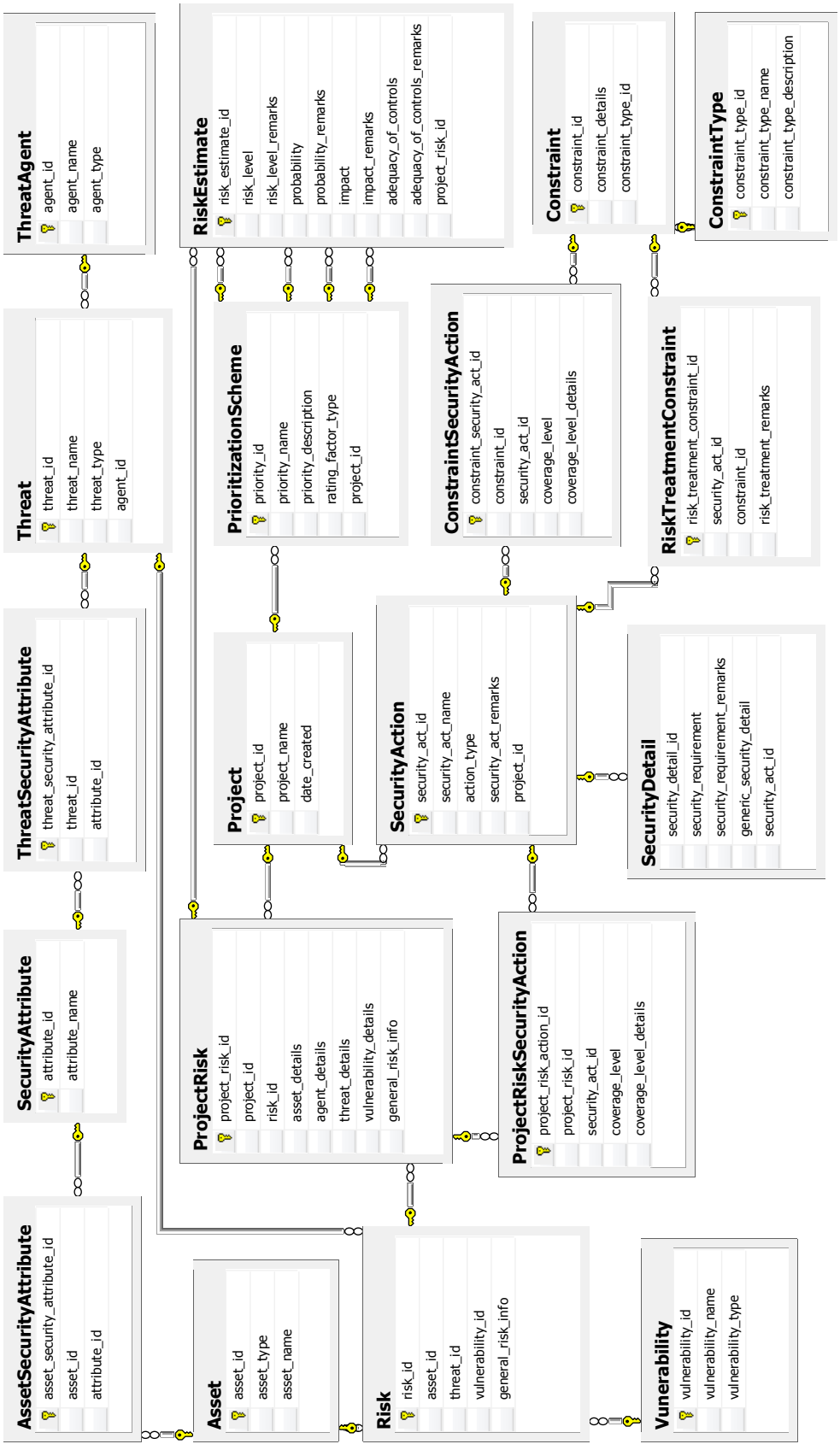


Figure A.4: New draft Entity Relationship Diagram of the SASaCS tool

Appendix B

Evaluation Interview Questions

B.1 Introduction and Interview Format

Appendix B was conceived to give further detail on the interview process used in this research project's evaluation stages. To start, this section outlines the interview format. The next section presents the list of predefined questions posed to interviewees.

In all cases, interviewees were sent preparatory documentation on BOF4WSS and the Solution Model and the problems they sought to address. This documentation was sent at least a week prior to the interview. Interviews were held at the workplace of the interviewee, in closed-door meeting rooms.

B.2 Question Sheet

SECTION 1: Questions related to the interviewee:

1. Could you give me a brief overview of your experience and background in security?
2. In any of your experiences, have you worked in any joint development or collaborative projects involving other companies?
3. Have you worked in any projects where security actions and requirements needed to be negotiated and agreed between all companies involved?

SECTION 2: Questions related to the evaluation of BOF4WSS:

1. Briefly introduce framework and its scope to the interviewee. Check if there are any questions.
2. Explain to the interviewee that, the framework argues that individual approaches to security across interacting enterprises are insufficient. It therefore advocates a highly collaborative approach where lots of information is shared, and time spent, towards companies jointly developing wholly secure interactions. Then ask the following questions:

   (a) What is your opinion on individual versus collaborative approaches to interactions security?
   (b) What are your general thoughts on the collaborative approach proposed?
   (c) How practical is it to expect such a high degree of collaboration (including sharing of information, attending meetings) across companies? Do you see any difficulties?
   (d) Can you think of any types of business scenarios, or types of companies or industries in which such a collaborative approach to security might be necessary or applicable?

3. Explain to the interviewee that, the framework is structured such that it provides detailed guidance on what companies should consider and at what points during planning and development. Give examples based on the activities of the Requirements Elicitation and Negotiation phases of BOF4WSS for example. Then ask the following questions:

   (a) Do you think that companies would be open to adopt a methodology that offered such detailed, and at times, prescriptive guidance?
   (b) Could adopting such a methodology be more a benefit or limitation to companies?

4. Explain to the interviewee that, the framework notes that particularly with scenarios where technologies such as Web services are used to support e-business interactions, industry and companies neglect the overarching high,

or business-level security (policies, processes, etc.) that should encapsulate and guide holistic security and secure interactions. Then ask the following questions:

   (a) Is there merit in focusing on these higher levels of security, or might technology-level integration alone be enough in these scenarios? Why?
   (b) Do you think that the time spent on this higher layer might be worth it, in respect to increased level of security or trust across businesses? If so, why? Or, why not?

5. Explain to the interviewee that, the Interaction Security Strategy (ISS) is a less rigid management structure that defines high-level cross-enterprise security directives to guide interactions, inclusive of relevant decisions internal to collaborating entities. Directives could include security strategies, policies, procedures and best practices. The ISS is used in addition to the security actions/requirements defined in legal contracts. Then ask the following questions:

   (a) Any questions on the ISS?
   (b) As compared to using only legal contracts, can you see any benefits in terms of security or trust, to this cross-enterprise strategy?
   (c) Are there any types of business scenarios, or types of companies or industries, that you think the ISS might be more applicable to?
   (d) Explain to the interviewee that, the ISS is jointly developed by companies and defines a security structure for all companies intended to make security approaches (regarding joint interactions) predictable and transparent. Could these factors foster trust between business partners?

SECTION 3: Questions related to the evaluation of the Solution Model for stage transition:

1. Briefly introduce the problems the Solution Model is intending to address. Then ask the following questions:

   (a) Any questions on this?

   (b) Do you agree or disagree with the problems highlighted?
   (c) Do you have any insights or past experience with the surrounding problem scenario to support or refute them as problems?

2. Explain to the interviewee that, to facilitate a matching of security actions across businesses, a common base is one possible solution. In the Solution Model, the common base used was a risks catalogue. This catalogue was shared by companies and from it, companies would select relevant risks to be factored in to their individual risk assessments. Then ask the following questions:

   (a) Do you see any issues with companies using a common risks base in that regard?
   (b) Explain to the interviewee that, in this proposal, there is the assumption that security actions and requirements always originate from some underlying risk. Risks therefore *always* drive security actions and requirements. Do you agree or disagree with this assumption?

3. Explain to the interviewee that, the Solution Model, and by large BOF4WSS, requires that businesses share a great amount of data on common risks faced, factors which influence risk treatment, security actions (even non-mitigation needs) with partners they are intending to collaborate with. This is the data that is matched and compared by the Solution Model's implemented tool. Then ask the following question:

   (a) Do you think that companies in real-world situations are likely to share this level of information with their partners?

4. Give the interviewee a demonstration of the tool and some of its main features (data entry, settings and comparison options, data output). Then ask the following questions:

   (a) How useful might be the tool and the output provided, in supporting the comparison and negotiation of security actions in business collaboration scenarios?

   (b) Are there any advantages or disadvantages noted?
   (c) Would you, as a security professional, use such a tool to aid in supporting the comparison and negotiation of security actions if given the option?

5. With regards to the tool's output report;

   (a) How user-friendly is the output report provided? Is it in a good, easy to use and understand format?
   (b) Do you have any suggestions to make the output more usable?

6. Explain to the interviewee that, the idea of automated reconciliation of conflicting security needs is being explored. One approach is to have companies add a degree of influence weighting or percentage value to each influential factor, then once these are totalled for a particular security actions, use the totalled value to compare across companies. Give the detailed example that was prepared. Then ask the following questions:

   (a) What do you think about the idea of degrees of importance of factors on a decided security action?
   (b) What are your thoughts on determining the strength of a security action in this way?
   (c) Do you think this would work?
   (d) What are your thoughts on this approach?

Thank interviewee for their time and assistance.

Appendix C

Applying the Pairwise Comparison-based Technique

C.1 Introduction

This appendix presents the application of the pairwise comparison-based technique proposed within Saaty [179, 180] to determine the weights of a specified set of decision criteria. These criteria are Laws and Regulations (LR), Contractual Obligations (CO), Business Policies (BP), Security Policies (SP), Security Budgets (SB) and the related Risk's Severity Level (RL). Since it is the use of the technique which is the focus, great detail is not given on the supporting mathematics. Readers however are encouraged to reference [179, 180] for proofs, theorems and in-depth discussions.

C.2 Determining Criteria Weights

As mentioned in Chapter 10, at a basic level the pairwise comparison technique gets decision makers to compare pairs of criteria according to importance and rank them on a defined scale. Saaty [179] then applies normalization methods to derive relative weight values for each criterion. One of the first key aspects therefore is the defined importance scale. This is shown in Table C.1.

Intensity of importance	**Definition**	**Explanation**
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgement slightly favour one activity over another
5	Essential or strong importance	Experience and judgement strongly favour one activity over another
7	Demonstrated importance	An activity is strongly favoured and its dominance demonstrated in practice
9	Absolute importance	The evidence favouring one activity over another is of the highest possible order of affirmation
2, 4, 6, 8	Intermediate values between two adjacent judgements	When compromise is needed
Reciprocals of above nonzero	If activity i has one of the above nonzero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i.	

Table C.1: Scale of relative importances (according to [179, 207])

The main purpose of the scale is to enable decision makers to indicate and abstractly quantify the extent to which one element (in this case, a criterion) is more or less important than another element (again, a criterion). Based on the table Definition and Explanation cell descriptions therefore, a respective numeric value would be chosen by decision makers. Once identified, the values are put into a pairwise comparison matrix. Formally, the list of available values are 9, 8, 7, 6, 5, 4, 3, 2, 1, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, $\frac{1}{5}$, $\frac{1}{6}$, $\frac{1}{7}$, $\frac{1}{8}$, $\frac{1}{9}$. The last nine values are the reciprocals of the former nine (according to the definition in the last row of Table C.1).

Following the pairwise comparison process and using the principal researcher's knowledge, the matrix below was created for the security action decision criteria. This process only needs to be completed once to lead to the weightings for each of the criteria. Weights would then apply to all security action decisions.

	LR	CO	BP	SP	SB	RL
LR	1	2	5	5	7	9
CO	$\frac{1}{2}$	1	4	4	6	8
BP	$\frac{1}{5}$	$\frac{1}{4}$	1	1	4	5
SP	$\frac{1}{5}$	$\frac{1}{4}$	1	1	4	6
SB	$\frac{1}{7}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{4}$	1	4
RL	$\frac{1}{9}$	$\frac{1}{8}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{4}$	1

To explain the choices behind some of the data in the matrix, the Laws and Regulations (LR) criterion is now considered. Reading the matrix row-to-column, the importance of LR was first compared to LR; this would be the first pair. As these are the same criterion, they are naturally of equal importance. Therefore, the value of 1 (from the scale) is input to the matrix where the LR row and LR column meets. Next, LR was compared to CO (thus, looking at where the row LR and column CO meet). Here, the value of 2 was chosen because it was felt that Laws and Regulations are slightly more important than Contractual Obligations (which are defined as agreements a company is previously legally binded to). This was based on the opinion that even though both are legally binding, Laws and Regulations define the mandatory rules of a country for all entities operating in that country to follow. Furthermore, it is more likely that a situation will be found where Contractual Obligations are drafted that appreciate a Law or Regulation, rather than a Law or Regulation being drafted that considers a company's Obligations.

In the comparison of LR to BP and LR to SP the value of 5 was selected. Even though an entity's Business Policy (BP) and Security Policy (SP) are of great importance, when these are compared to Laws and Regulations, it was felt that the latter criterion had greater importance in a decision. Apart from the reality that these policies are likely to be in line with (and in someways therefore, subordinate to) Laws and Regulations, there is the fact that not fulfilling a Law or Regulation will have a much greater impact than not fulfilling a policy.

The next pair is LR and SB and for this, the value of 7 indicating demonstrated importance, was chosen. Although a limited Security Budget (SB) is a serious restriction on a company's security posture, for all the reasons mentioned

above, Laws and Regulations still demand greater emphasis. A higher value was chosen as compared to CO, BP and SP particularly because surveys [210] have highlighted that companies are still spending too meagre amounts of their overall budgets on IT security. Therefore, as opposed to having truly limited funds in the business which then carries over to security spending, companies might be deliberately allocating the bare minimum or less to spending on security. Likely reasons for this are perceived importance or benefits, or visible return on investments.

The last criteria pair for comparison in the row is LR and RL. For this pair a value of 9 was chosen. Such a high importance level was selected because regardless of a Risk's Severity Level, a Law or Regulation would dictate its treatment. Furthermore, considering the other criteria, the Severity Level is usually assessed in terms of these. For example, a Risk's Severity Level is only as important to a decision as a company's Security Policy defines. A policy therefore could state that all risks of type X are to be accepted. Therefore even if there is a risk of type X that has a high Risk Severity Level, that risk would still be accepted. Aspects such as this lead to the selection of absolute importance of LR over RL.

The comparison process presented above was completed for the other rows to achieve the values from 1–9 shown in the matrix. A value of 1–9 in a cell (row, column intersection) indicates that a row element was equally or more important than a column element. Once these were identified, their reciprocals (which theoretically show lesser importance) were produced and included as stipulated by the scale in Table C.1. As an example, consider the LR and SB pairwise comparison result of 7 above. The respective step is to locate the cell (row and column intersection) which represents the reverse comparison and place the reciprocal value in that cell. As the reverse comparison cell would be SB (the row) and LR (the column), $\frac{1}{7}$ was input there. Such a procedure was used to input all other reciprocal values.

With the matrix completed, the next task mentioned in Saaty [179] is the

computation of a vector of priorities from the matrix; these priorities are also known as the relative criteria weights. To do this, Saaty suggests dividing the elements of each column by the sum of that column (therefore, normalize the column), and then adding the elements in each resulting row and divide this sum by the number of elements in the row. This, he notes, is a process of averaging over the normalized columns. Applying that process to the matrix above, the following results. The last column displays the priority vector.

	LR	CO	BP	SP	SB	RL	**Priority**
LR	0.464	0.527	0.437	0.438	0.315	0.273	0.409
CO	0.232	0.264	0.349	0.350	0.270	0.242	0.285
BP	0.093	0.066	0.087	0.088	0.180	0.152	0.111
SP	0.093	0.066	0.087	0.088	0.180	0.182	0.116
SB	0.066	0.044	0.022	0.022	0.045	0.121	0.053
RL	0.052	0.033	0.017	0.015	0.011	0.030	0.026
							$\sum = 1$

To discuss this process and explain the values above, the row LR and column RL intersection is used; here the value resulting is 0.273. According to Saaty [179], the first task is dividing the element by the sum of the column. This therefore gives the calculation below. (Note that the values used are from the first matrix.)

$$9 \div (9 + 8 + 5 + 6 + 4 + 1) = 0.273$$

Having made this calculation for all the cell values, the resulting row values are averaged. Taking the first row in the matrix above, the following calculation would be conducted. This gives the weight of importance of LR, that is, Laws and Regulations on the security action decision.

$$Priority/Weight = (0.464 + 0.527 + 0.437 + 0.438 + 0.315 + 0.273) \div 6 = 0.409$$

Lastly, considering the subjectivity behind this pairwise comparison-based process, Saaty [179] also defined the Consistency Index (CI) and Consistency Ratio (CR). These calculated values are used to measure and test the consistency of the data entered by decision makers into the pairwise comparison matrix (that

is, the first matrix in this appendix). Saaty notes that when the CI is close to zero and CR is equal to or less than 0.10, this is an indication of a good and acceptable consistency. In cases where CR yields a value greater than 0.10 however, a re-examination of the decision maker's pairwise judgements is recommended. Readers are directed to [179, 207] for detailed discourse on CI, CR and the mathematical reasoning supporting them.

To calculate the CI, the formula $(\lambda_{max} - n)/(n-1)$ is used, where n is the number of criteria, and λ_{max} is the value produced by adding the columns of the first matrix and then multiplying the resulting vector by the priority vector in the second matrix [207]. With the vector from the addition of columns in the first matrix being roughly (2.154, 3.792, 11.45, 11.417, 22.25, 33) therefore, λ_{max} has a value of 6.609. This leads to,

$$CI = \frac{(\lambda_{max} - n)}{(n-1)} = \frac{(6.609 - 6)}{6 - 1} = 0.1218$$

A value of 0.1218 is a fair indicator of consistency, however to confirm, the CR is considered. To calculate this, CI is divided by a Random Consistency Index (RCI). This index represents an average random consistency index derived from randomly generated reciprocal matrices [179, 207]. For matrices with size $n = 6$ the value 1.24 is defined as the RCI (see [179, 207]). Therefore,

$$CR = \frac{CI}{RCI} = \frac{0.1218}{1.24} = 0.0982 < 0.10$$

As the CR value is less than 0.10, this indicates that there is an acceptable consistency of the original matrix judgements and data. This finding acts to add some degree of reliability and credibility to the pairwise comparisons done above, and ultimately, the weightings proposed here.

Bibliography

[1] Active Endpoints. BPEL Open Source Engine - The ActiveBPEL Community Edition BPEL Engine — Active Endpoints, n.d. http://www.activevos.com/community-open-source.php (Accessed 21 August 2010). (Cited on page 115.)

[2] R. S. Aguilar-Savén. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2):129–149, July 2004. (Cited on pages 55 and 56.)

[3] C. Alberts and A. Dorofee. *Managing Information Security Risks : The OCTAVE Approach*. Addison-Wesley, Boston, 2003. (Cited on pages 47, 145 and 157.)

[4] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, Berlin, 2004. (Cited on pages 15, 17, 28, 35 and 69.)

[5] Anonymous. WS-CDL Eclipse, n.d. http://wscdl-eclipse.sourceforge.net/main.htm (Accessed 21 August 2010). (Cited on page 64.)

[6] W. H. Baker, G. E. Smith, and K. J. Watson. Information security risk in the e-supply chain. In Q. Zhang, editor, *E-Supply Chain Technologies and Management*, pages 142–161. Idea Group Inc., Hershey, PA, 2007. (Cited on pages 20, 53 and 233.)

[7] A. Baldwin, Y. Beres, S. Shiu, and P. Kearney. A model-based approach to trust, security and assurance. *BT Technology Journal*, 24(4):53–68, 2006. (Cited on page 234.)

[8] A. Barbir, C. Hobbs, E. Bertino, F. Hirsch, and L. Martino. Challenges of testing web services and security in soa implementations. In L. Baresi and E. Di Nitto, editors, *Test and Analysis of Web Services*, pages 395–440. Springer, Heidelberg, 2007. (Cited on pages 77 and 78.)

[9] BEA Systems, BMC Software, CA, Inc., IBM, Layer 7 Technologies, Microsoft, Novell, VeriSign. Web services federation language, 2007. http://www.ibm.com/developerworks/library/specification/ws-fed/ (Accessed 21 August 2010). (Cited on page 28.)

[10] R. Bebawy, H. Sabry, S. El-Kassas, Y. Hanna, and Y. Youssef. Nedgty: Web services firewall. In *IEEE International Conference on Web Services*, pages 597–601, Orlando, Florida, 2005. (Cited on pages 23 and 29.)

[11] V. Belton and T. J. Stewart. *Multiple Criteria Decision Analysis: An Integrated Approach*. Kluwer Academic Publishers, Boston, 2002. (Cited on page 259.)

[12] B. L. Berg. *Qualitative research methods for the social sciences*. Pearson International Education, London, fifth edition, 2004. (Cited on pages 6, 229, 230 and 231.)

[13] E. Berkman. B2B PARTNERSHIPS SECURITY - How to Practice Safe B2B. http://www.cso.com.au/article/80707/how_practise_safe_b2b (Accessed 21 August 2010), 2002. (Cited on page 95.)

[14] K. Beznosov, D. J. Flinn, S. Kawamoto, and B. Hartman. Introduction to web services and their security. *Information Security Technical Report*, 10(1):2–14, 2005. (Cited on pages 17 and 28.)

[15] K. Bhargavan, R. Corin, C. Fournet, and A. D. Gordon. Secure sessions for web services. *ACM Transactions on Information and System Security (TISSEC)*, 10(2), 2007. (Cited on page 13.)

[16] B. W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, May 1988. (Cited on pages 40 and 41.)

[17] R. J. Boncella. Web services and web services security. *Communications of the Association for Information Systems*, 14(18):344–363, 2004. (Cited on pages 2, 21, 23 and 94.)

[18] H.-J. Bullinger, K.-P. Fähnrich, and T. Meiren. Service engineering—methodical development of new service products. *International Journal of Production Economics*, 85(3):275–287, 2003. (Cited on page 41.)

[19] A. Calder and S. Watkins. *IT Governance: A Manager's Guide to Data Security and ISO 27001/ISO 27002*. Kogan Page Limited, London, fourth edition, 2008. (Cited on pages 51, 52 and 81.)

[20] M. Cambronero, G. Díaz, J. Pardo, and V. Valero. Using UML diagrams to model real-time web services. In *Second International Conference on Internet and Web Applications and Services*, 2007. (Cited on page 76.)

[21] M. Cambronero, G. Díaz, J. Pardo, V. Valero, and F. L. Pelayo. RT-UML for modeling real-time web services. In *IEEE Services Computing Workshops*, pages 131–139, 2006. (Cited on page 64.)

[22] E. Cerami. *Web Services Essentials*. O'Reilly, Farnham, 2002. (Cited on page 14.)

[23] N. Cerpa and J. Verner. Prototyping: some new results. *Information and Software Technology*, 38(12):743–755, 1996. (Cited on page 163.)

[24] CERT Coordination Center (CERT/CC). OCTAVE® information security risk evaluation. http://www.cert.org/octave/ (Accessed 21 August 2010), n.d. (Cited on pages 47 and 145.)

[25] D. Chaffey. *E-Business and E-Commerce Management: Strategy, Implementation and Practice*. Financial Times Prentice Hall, Harlow, 3rd edition, 2007. (Cited on page 11.)

[26] A. Charfi and M. Mezini. Using aspects for security engineering of web service compositions. In *IEEE International Conference on Web Services*, pages 59–66, Orlando, 2005. (Cited on pages 30, 32, 33 and 279.)

[27] S. Chatterjee. The waterfall that won't go away. *SIGSOFT Softw. Eng. Notes*, 35(1):9–10, 2010. (Cited on page 41.)

[28] S. Chatterjee and J. Webber. *Developing Enterprise Web Services: An Architect's Guide*. Prentice Hall PTR, Upper Saddle River, NJ, 2004. (Cited on pages 1, 19 and 69.)

[29] M. Chen. An analysis of the driving forces for web services adoption. *Information Systems and e-Business Management*, 3(3):265–279, 2005. (Cited on pages 1 and 17.)

[30] M. Chen and M. J. Meixell. Web services enabled procurement in the extended enterprise: An architectural design and implementation. *Journal of Electronic Commerce Research*, 4(4):140–155, 2003. (Cited on pages 12, 13 and 85.)

[31] D. C. Choua and K. Yurov. Security development in web services environment. *Computer Standards & Interfaces*, 27(3):233–240, 2004. (Cited on page 19.)

[32] CIECA. Security for Electronic B2B Transactions: CIECA Guidelines and Recommendations for Securing the Electronic Transmission of B2B Transactions. http://www.cieca.com/resources/Documents/SecurityforElectronic-B2BTransactions-2003-05-19.pdf (Accessed 21 August 2010), 2003. (Cited on page 95.)

[33] CISCO and IronPort. 2008 internet security trends: A report on emerging attack platforms for spam, viruses and malware, 2008. http://ironport.com/pdf/Trends_Report_IronPort_2008.pdf (Accessed 21 August 2010). (Cited on page 19.)

[34] ContentGuard. XrML... eXtensible rights Markup Language, n.d. http://www.xrml.org/ (Accessed 19 April 2010). (Cited on page 28.)

[35] R. Craig. E-com supply chain and smes. In Q. Zhang, editor, *E-Supply Chain Technologies and Management*, pages 34–53. Idea Group Inc., Hershey, PA, 2007. (Cited on page 12.)

[36] M. Curphey. Web services: Developers dream or hackers heaven? *Information Security Technical Report*, 10(4):228–235, 2005. (Cited on pages 2, 24 and 28.)

[37] M. Curphey and R. Arawo. Web application security assessment tools. *IEEE Security & Privacy*, 4(4):32–41, 2006. (Cited on page 78.)

[38] C. H. Davis and F. Vladica. Adoption and use of internet technologies and e-business solutions by Canadian micro-enterprises. In *Annual Conference of the International Association for Management of Technology*, Vienna, 2005. (Cited on page 11.)

[39] C. H. Davis and F. Vladica. The value of internet technologies and e-business solutions to micro-enterprises in atlantic Canada. In S. Barnes, editor, *E-Commerce and V-Business: Digital Enterprise in the Twenty-First Century*, pages 125–156. Butterworth-Heinemann, Oxford, 2nd edition, 2007. (Cited on page 12.)

[40] E. W. Davis and R. E. Spekman. *The Extended Enterprise: Gaining Competitive Advantage through Collaborative Supply Chains*. FT Prentice Hall, Upper Saddle River, NJ, 2004. (Cited on pages 12, 35, 36, 60, 82, 239 and 244.)

[41] W. S. Davis and J. Benamati. *E-Commerce Basics: Technology Foundations and E-Business Applications*. Addison-Wesley, Boston, 2003. (Cited on page 11.)

[42] DCSSI. Expression des Besoins et Identification des Objectifs de Sécurité (EBIOS) – Section 1 to 5. Technical report, Secrétariat général de la défense nationale, Direction Centrale de la Séccuritéc des Systècmes D'Information (DCSSI), 2004. (Cited on pages 183, 184, 214 and 219.)

[43] G. Decker, O. Kopp, F. Leymann, K. Pfitzner, and M. Weske. Modeling service choreographies using BPMN and BPEL4Chor. In Z. Bellahsène and M. Léonard, editors, *Advanced Information Systems Engineering*, volume 5074 of *Lecture Notes in Computer Science*, pages 79–93. Springer, Heidelberg, 2008. (Cited on pages 63 and 65.)

[44] G. Decker, O. Kopp, F. Leymann, and M. Weske. BPEL4Chor: Extending BPEL for modeling choreographies. In *IEEE International Conference on Web Services*, pages 296–303, Los Alamitos, CA, USA, 2007. IEEE Computer Society. (Cited on pages 63, 64 and 65.)

[45] O. Demirörs, Ç. Gencel, and A. Tarhan. Utilizing business process models for requirements elicitation. In *The 29th Conference on EUROMICRO*, pages 409–412. IEEE, 2003. (Cited on pages 44 and 46.)

[46] 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. Technical report, SINTEF, 2006. (Cited on pages 152, 154, 199, 200 and 219.)

[47] F. den Braber, I. Hogganvik, M. S. Lund, K. Stølen, and F. Vraalsen. Model-based security analysis in seven steps - a guided tour to the CORAS method. *BT Technology Journal*, 25(1):101–117, 2007. (Cited on pages 47, 95, 145 and 183.)

[48] S. Dynes, H. Brechbühl, and M. Johnson. Information security in the extended enterprise: Some initial results from a field study of an industrial firm. In *Fourth Workshop on the Economics of Information Security*, 2005. (Cited on page 20.)

[49] S. Dynes, L. M. Kolbe, and R. Schierholz. Information security in the extended enterprise: A research agenda. In *AMCIS 2007 Proceedings*, 2007. (Cited on pages 12, 20, 22, 35, 36, 127 and 234.)

[50] ebXML. ebXML - Enabling A Global Electronic Market, n.d. http://ebxml.org/ (Accessed 21 August 2010). (Cited on pages 13 and 18.)

[51] T. Erl. *Service-Oriented Architecture: Concepts, Technology, and Design*. Pearson Education, Upper Saddle River, NJ, 2005. (Cited on pages 2, 13 and 62.)

[52] European Network and Information Security Agency (ENISA). Glossary of risk management. http://www.enisa.europa.eu/act/rm/cr/risk-management-inventory/glossary (Accessed 30 July 2010), n.d. (Cited on page 145.)

[53] J. Euzenat. *Ontology Matching*. Springer, Berlin, 2007. (Cited on pages 138 and 150.)

[54] R. A. Falbo, F. B. Ruy, G. Bertollo, and D. F. Togneri. Learning how to manage risks using organizational knowledge. In H. Holz and G. Melnik, editors, *Advances in Learning Software Organizations (6th International Workshop, LSO 2004 Banff, Canada, June 20-21, 2004 Proceedings)*, volume 3096 of *Lecture Notes in Computer Science*, pages 7–18. Springer, Berlin, 2004. (Cited on pages 155 and 158.)

[55] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, 2nd edition, 2004. (Cited on pages 136 and 150.)

[56] S. Fenz and A. Ekelhart. Formalizing information security knowledge. In *4th International Symposium on Information, Computer, and Communications Security*, pages 183–194, Sydney, Australia, 2009. (Cited on pages viii, 151, 154 and 156.)

[57] S. Fenz, T. Pruckner, and A. Manutscheri. Ontological mapping of information security best-practice guidelines. In W. Abramowicz, editor, *Business Information Systems*, volume 21 of *Lecture Notes in Business Information Processing*, pages 49–60. Springer, Heidelberg, 2009. (Cited on pages 182, 183, 184, 185, 193, 199 and 214.)

[58] D. G. Firesmith. Engineering security requirements. *Journal of Object Technology*, 2(1):53–68, 2003. (Cited on page 143.)

[59] D. G. Firesmith. Security use cases. *Journal of Object Technology*, 2(3):53–64, 2003. (Cited on pages viii and 144.)

[60] D. G. Firesmith. Specifying reusable security requirements. *Journal of Object Technology*, 3(1):61–75, 2004. (Cited on pages 2, 152, 153 and 156.)

[61] S. Fischer and C. Werner. Towards service-oriented architectures. In R. Studer, S. Grimm, and A. Abecker, editors, *Semantic Web Services: Concepts, Technologies, and Applications*, pages 15–24. Springer-Verlag, Berlin, 2007. (Cited on pages 17 and 68.)

[62] Forum Systems. Anatomy of a web services attack, n.d. http://www.forumsys.com/resources/resources/whitepapers/Anatomy_of_Attack_wp.pdf (Accessed 21 August 2010). (Cited on pages 23 and 24.)

[63] H. Foster. WS-Engineer 2008: A service architecture, behaviour and deployment verification platform. In A. Bouguettaya, I. Krueger, and T. Margaria, editors, *Service-Oriented Computing ICSOC 2008*, volume 5364 of *Lecture Notes in Computer Science*, pages 728–729. Springer, Heidelberg, 2008. (Cited on page 64.)

[64] 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*, pages 250–256, 2008. (Cited on page 207.)

[65] D. Z. Garcia and M. B. Felgar de Toledo. A policy-based web service infrastructure for autonomic service integration. In *First Latin American Autonomic Computing Symposium (LAACS)*, Campo Grande, MS, 2006. (Cited on page 72.)

[66] D. Z. Garcia and M. B. Felgar de Toledo. A policy approach supporting web service-based business processes. In *First Brazilian Workshop on Business Process Management (WBPM 2007)*, Gramado, RS, Brazil, 2007. (Cited on pages 66 and 113.)

[67] M. Gerber and R. von Solms. From risk analysis to security requirements. *Computers & Security*, 20(7):577–584, 2001. (Cited on pages 143, 145 and 147.)

[68] C. Geuer-Pollmann and J. Claessens. Web services and web service security standards. *Information Security Technical Report*, 10(1):15–24, 2005. (Cited on page 28.)

[69] G. M. Giaglis. A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems*, 13(2):209–228, 2001. (Cited on pages 55 and 56.)

[70] Global Grid Forum. The Open Grid Services Architecture, Version 1.0, 2005. http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf (Accessed 21 August 2010). (Cited on page 14.)

[71] F. Goethals, J. Vandenbulcke, W. Lemahieu, and M. Snoeck. Different types of business-to-business integration: Extended enterprise integration vs. market B2B integration. In I. Lee, editor, *E-Business Innovation and Process Management*, pages 1–17. CyberTech Publishing, Hershey, PA, 2007. (Cited on pages 12, 17 and 234.)

[72] B. A. Gran, R. Fredriksen, and A. P.-J. Thunem. An approach for model-based risk assessment. In M. Heisel, P. Liggesmeyer, and S. Wittmann, editors, *Computer Safety, Reliability, and Security (Proceedings of 23rd International Conference, SAFECOMP 2004, Potsdam, Germany, September 21-24, 2004)*, volume 3219 of *Lecture Notes in Computer Science*, pages 311–324. Springer Berlin, Heidelberg, 2004. (Cited on pages 152 and 199.)

[73] T. Grandison. Conceptions of trust: Definition, constructs, and models. In R. Song, L. Korba, and G. Yee, editors, *Trust in E-services: Technologies, Practices and Challenges*, pages 1–28. Idea Group Publishing, Hershey, PA, 2007. (Cited on page 35.)

[74] C. Gutiérrez, E. Fernández-Medina, and M. Piattini. Web services enterprise security architecture: a case study. In *Workshop on Secure Web Services (SWS)*, pages 10–19, Fairfax, VA, 2005. (Cited on pages 1, 17, 30, 32 and 279.)

[75] C. Gutiérrez, E. Fernández-Medina, and M. Piattini. PWSSec: Process for web services security. In *IEEE International Conference on Web Services*, pages 213–222, Chicago, IL, 2006. (Cited on pages 30, 32, 33, 74, 75 and 279.)

[76] C. B. Haley, J. D. Moffett, R. Laney, and B. Nuseibeh. A framework for security requirements engineering. In *International Workshop on Software*

Engineering for Secure Systems, pages 35–42, Shanghai, China, 2006. ACM. (Cited on page 143.)

[77] B. Hartman, D. J. Flinn, K. Beznosov, and S. Kawamoto. *Mastering Web Services Security*. Wiley, Indianapolis, 2003. (Cited on pages 2, 24, 32 and 33.)

[78] F. Hartman and R. A. Ashrafi. Project management in the information systems and information technologies industries. *Project Management Journal*, 33(3):5–15, 2002. (Cited on page 44.)

[79] P. Hernacki and R. Pruett. Solving the supply chain puzzle. *Business Integration Journal*, pages 34–38, 2003. (Cited on page 12.)

[80] L. Hines, S. Baldwin, M. Giles, and J. Peralta. Implementing agile development in a waterfall project, 2009. http://www.ibm.com/developerworks/websphere/techjournal/0907_hines/0907_hines.html (Accessed 21 August 2010). (Cited on page 278.)

[81] L. Hodge and M. Mock. A proposed object-oriented development methodology. *Software Engineering Journal*, 7(2):119–129, Mar 1992. (Cited on page 41.)

[82] K. Hogg, P. Chilcott, M. Nolan, and B. Srinivasan. An evaluation of web services in the design of a b2b application. In *27th Australasian Computer Science Conference*, volume 26, pages 331–340, Dunedin, New Zealand, 2004. (Cited on pages 13 and 17.)

[83] I. Hogganvik and K. Stølen. Risk analysis terminology for IT-systems: Does it match intuition? In *4th International Symposium on Empirical Software Engineering*, pages 13–23, 2005. (Cited on page 148.)

[84] S. H. Houmb and G. Georg. The aspect-oriented risk-driven development (AORDD) framework. In *International Conference on Software Develop-*

ment (SWDC-REX), pages 81–91, Reykjavik, Iceland, 2005. (Cited on pages 150, 153, 154 and 155.)

[85] T. Humphreys, M. De-Soete, and C. J. Mitchell. Securing e-business. *ISO Focus*, 1(1):30–32, January 2004. (Cited on page 19.)

[86] innoQ. Web services standards overview, 2007. http://www.innoq.com/resources/ws-standards-poster/ (Accessed 21 August 2010). (Cited on pages 68, 69 and 114.)

[87] International Business Machines (IBM) Corp. Emerging Technologies Toolkit (ETTK) for Web Services, n.d. http://www.alphaworks.ibm.com/tech/ettk (Accessed 21 August 2010). (Cited on page 72.)

[88] International Business Machines (IBM) Corp. Rational Software Architect for WebSphere Software, n.d. http://www-01.ibm.com/software/awdtools/swarchitect/websphere/ (Accessed 21 August 2010). (Cited on page 64.)

[89] International Business Machines (IBM) Corp. Web Service Level Agreements (WSLA) Project, n.d. http://www.research.ibm.com/wsla/implementation.html (Accessed 21 August 2010). (Cited on page 72.)

[90] International Organization for Standardization (ISO). ISO/IEC guide 73 risk management – vocabulary – guidelines for use in standards. Technical report, 2002. (Cited on pages viii, 144, 145, 147, 148 and 149.)

[91] T. Ishaya and J. R. Nurse. Cross-enterprise policy model for e-business web services security. In D. Weerasinghe, editor, *Information Security and Digital Forensics*, volume 41 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 163–171. Springer, Heidelberg, 2010. (Cited on pages 33 and 50.)

[92] M. Jensen, N. Gruschka, R. Herkenhoner, and N. Luttenberger. SOA and web services: New technologies, new standards – new attacks. In *The Fifth European Conference on Web Services*, pages 35–44, Halle (Saale), Germany, 2007. (Cited on pages 1, 24 and 70.)

[93] Jini. Juni.org, n.d. http://www.jini.org/ (Accessed 17 April 2010). (Cited on page 14.)

[94] A. Jones and D. Ashenden. *Risk Management for Computer Security: Protecting Your Network & Information Assets*. Elsevier Butterworth-Heinemann, Amsterdam, Netherlands, 2005. (Cited on pages 144, 156 and 158.)

[95] S. Jones, M. Wilikens, and M. Masera. Trust requirements in e-business. *Communications of the ACM*, 43(12):81–87, 2000. (Cited on page 11.)

[96] J. Jürjens. *Secure Systems Development with UML*. Springer-Verlag, Berlin, 2005. (Cited on page 68.)

[97] S. Kairab. *A Practical Guide to Security Assessment*. Auerbach Publications, Boca Raton, FL, 2005. (Cited on pages 148 and 157.)

[98] J. Kajava, J. Anttila, R. Varonen, R. Savola, and J. Roning. Information security standards and global business. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 2091–2095, 2006. (Cited on page 127.)

[99] B. Karabacak and I. Sogukpinar. ISRAM: information security risk analysis method. *Computers & Security*, 24(2):147–159, 2005. (Cited on pages 47 and 145.)

[100] S. K. Katsikas, J. Lopez, and G. Pernul. Trust, privacy and security in e-business: Requirements and solutions. In P. Bozanis and E. N. Houstis, editors, *Advances in Informatics*, volume 3746 of *Lecture Notes in Computer Science*, pages 548–558. Springer, Heidelberg, 2005. (Cited on pages 21 and 22.)

[101] P. Keen, C. Ballance, S. Chan, and S. Schrump. *Electronic Commerce Relationships : Trust By Design*. Prentice Hall PTR, Upper Saddle River, NJ, 2000. (Cited on page 35.)

[102] A. Keller and H. Ludwig. The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11(1):57–81, 2003. (Cited on page 72.)

[103] M. Khalifa and J. M. Verner. Drivers for software development method usage. *IEEE Transactions on Engineering Management*, 47(3):360–369, 2000. (Cited on pages 40 and 41.)

[104] B. King and K. Reinold. *Finding the Concept, Not Just the Word: A Librarian's Guide to Ontologies and Semantics*. Chandos, Oxford, 2008. (Cited on pages 143 and 155.)

[105] C. R. Kothari. *Research Methodology: Methods And Techniques*. New Age International, Boston, second edition, 2008. (Cited on page 5.)

[106] D. J. Landoll. *The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments*. Auerbach Publications, Boca Raton, FL, 2006. (Cited on pages 145, 147, 148, 155, 156 and 158.)

[107] K. C. Laudon and C. G. Traver. *E-commerce: Business, Technology, Society*. Prentice Hall, New Jersey, third edition, 2007. (Cited on pages vii, 11, 12, 22 and 30.)

[108] K. C. Laudon and C. G. Traver. *E-commerce: Business, Technology, Society*. Prentice Hall, fifth edition, 2008. (Cited on pages 1, 11, 12 and 20.)

[109] J. Liberty and D. Hurwitz. *Programming .NET Windows Applications*. O'Reilly, Farnham, 2003. (Cited on page 174.)

[110] Liberty Alliance Project. Home - Liberty Alliance, n.d. http://www.projectliberty.org/ (Accessed 21 August 2010). (Cited on pages 28 and 69.)

[111] S. Lichtenstein. Developing internet security policy for organizations. In *30th Hawaii International Conference on System Sciences*, volume 4, pages 350–357, Wailea, HI, 1997. (Cited on page 31.)

[112] T. Lodderstedt, D. A. Basin, and J. Doser. SecureUML: A UML-based modeling language for model-driven security. In J.-M. Jézéquel, H. Hussmann, and S. Cook, editors, *UML 2002: The Unified Modeling Language*, volume 2460 of *Lecture Notes in Computer Science*, pages 426–441. Springer, Heidelberg, 2002. (Cited on page 68.)

[113] K. Lunn. *Software development with UML*. Palgrave Macmillan, Basingstoke, 2003. (Cited on page 166.)

[114] K. Mantell. From UML to BPEL: Model driven architecture in a web services world, 2005. http://www.ibm.com/developerworks/webservices/library/ws-uml2bpel/ (Accessed 21 August 2010). (Cited on page 76.)

[115] S. Masud. Use RosettaNet-based web services, part 3: BPEL4WS and RosettaNet, 2003. http://www.ibm.com/developerworks/webservices/library/ws-rose3/ (Accessed 21 August 2010). (Cited on page 19.)

[116] N. Mayer, E. Dubois, R. Matulevicius, and P. Heymans. Towards a measurement framework for security risk management. In *Workshop on Modeling Security (MODSEC) at International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Toulouse, France, 2008. (Cited on pages viii, 152, 153, 154 and 159.)

[117] P. Mayer, A. Schroeder, and N. Koch. A model-driven approach to service orchestration. In *IEEE International Conference on Services Computing*, pages 533–536, Honolulu, Hawaii, 2008. IEEE Computer Society. (Cited on page 55.)

[118] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla, and A. Murukan. Improving web application security: Threat modeling (Microsoft Corporation). http://msdn.microsoft.com/en-us/library/aa302419.aspx (Accessed 21 August 2010), 2003. (Cited on page 148.)

[119] M. J. Meixell. Quantifying the value of web services in supplier networks. *Industrial Management & Data Systems*, 106(3):407–422, 2006. (Cited on page 17.)

[120] D. A. Menascé and V. Almeida. *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*. Prentice Hall PTR, London, 2000. (Cited on page 11.)

[121] J. Mendling and M. Hafner. From WS-CDL choreography to BPEL process orchestration. *Journal of Enterprise Information Management*, 21(5):525–542, 2008. (Cited on page 64.)

[122] Z. Mi, C. Zunping, M. Ziji, and Z. Binyu. A security model design in web service environment. In *Fifth International Conference on Computer and Information Technology (CIT'2005)*, pages 736–740, Shanghai, China, 2005. (Cited on page 23.)

[123] Microsoft Corporation. Microsoft BizTalk Server, n.d. http://www.microsoft.com/biztalk (Accessed 21 August 2010). (Cited on page 76.)

[124] Microsoft Corporation. Microsoft Security Development Lifecycle (SDL) — SDL Process Guidance, n.d. http://www.microsoft.com/security/sdl/ (Accessed 13 October 2010). (Cited on page 36.)

[125] Microsoft Corporation. Microsoft SQL Server, n.d. http://www.microsoft.com/sql/default.mspx (Accessed 21 August 2010). (Cited on page 174.)

[126] Microsoft Corporation. Microsoft Visual Studio, n.d. http://www.microsoft.com/visualstudio/en-us/default.mspx (Accessed 21 August 2010). (Cited on page 174.)

[127] Microsoft Corporation. Sample supply chain scenario, n.d. http://msdn.microsoft.com/en-us/library/bb950139(BTS.10).aspx (Accessed 21 August 2010). (Cited on page 85.)

[128] J. Misrahi. Validating your business partners. In H. F. Tipton and M. Krause, editors, *Information Security Management Handbook*, volume 1, pages 123–131. Auerbach Publications, Boca Raton, FL, sixth edition, 2007. (Cited on pages 51 and 53.)

[129] A. Munteanu. Information security risk assessment: The qualitative versus quantitative dilemma. In *Managing Information in the Digital Economy: Issues & Solutions - Proceedings of the 6th International Business Information Management Association (IBIMA) Conference*, pages 227–232, 2006. (Cited on page 147.)

[130] M. D. Myers. *Qualitative research in business and management*. SAGE, London, 2009. (Cited on page 228.)

[131] J. Myerson. Use SLAs in a web services context, part 1: Guarantee your web service with a SLA, 2004. http://www.ibm.com/developerworks/library/ws-sla/ (Accessed 21 August 2010). (Cited on page 72.)

[132] S. Nachtigal and C. J. Mitchell. Modelling e-business security using business processes. In *The International Conference on Security and Cryptogra-*

phy (SECRYPT 2006), pages 459–464, Setubal, Portugal, 2006. INSTICC Press. (Cited on pages 11 and 20.)

[133] J. D. Naumann and A. M. Jenkins. Prototyping: The new paradigm for systems development. *MIS Quarterly*, 6(3):29–44, 1982. (Cited on page 41.)

[134] Net-Square Solutions Pvt. Ltd. wsChess - web services assessment and defense toolkit, n.d. http://net-square.com/wschess/index.shtml (Accessed 21 August 2010). (Cited on page 78.)

[135] R. C. Newman. Cybercrime, identity theft, and fraud: Practicing safe internet - network security threats and vulnerabilities. In *The 3rd Annual Conference on Information Security Curriculum Development (InfoSecCD Conference'06)*, pages 68–78, Kennesaw, Georgia, 2006. (Cited on page 19.)

[136] H. R. M. Nezhad, H. Skogsrud, B. Benatallah, and F. Casati. Securing service-based interactions: Issues and directions, 2005. http://dsonline.computer.org/WAS (Accessed 21 August 2010). (Cited on page 26.)

[137] J. R. Nurse. Enhancing web services security for e-business using cross-enterprise policies. Master's thesis, University of Hull, UK, 2006. (Cited on pages vii, 33, 34, 50 and 85.)

[138] 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) 2009*, pages 286–291. IEEE Computer Society, 2009. (Cited on page xi.)

[139] 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*, 2(4):253–276, 2009. (Cited on page xi.)

[140] J. R. Nurse and J. E. Sinclair. Supporting the comparison of business-level security requirements within cross-enterprise service development. In

W. Abramowicz, editor, *Business Information Systems*, volume 21 of *Lecture Notes in Business Information Processing*, pages 61–72. Springer, Heidelberg, 2009. (Cited on page xi.)

[141] J. R. Nurse and J. E. Sinclair. A case study analysis of an e-business security negotiations support tool. In *Electrical Engineering and Applied Computing*, Lecture Notes in Electrical Engineering. Springer, 2010. (Cited on page xi.)

[142] J. R. Nurse and J. E. Sinclair. Evaluating the compatibility of a tool to support e-businesses' security negotiations. In *The International Conference of Information Security and Internet Engineering (ICISIE), under World Congress on Engineering (WCE) 2010*, volume 1, pages 438–443. Newswood Limited, International Association of Engineers, 2010. (Cited on page xi.)

[143] J. R. Nurse and J. E. Sinclair. An evaluation of BOF4WSS and the security negotiations model and tool used to support it. *International Journal On Advances in Security*, 3(3), 2010. (Cited on page xi.)

[144] 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 (ICIW) 2010*, pages 13–18. IEEE Computer Society, 2010. (Cited on page xi.)

[145] J. R. Nurse and J. E. Sinclair. A thorough evaluation of the compatibility of an e-business security negotiations support tool. *International Journal of Computer Science*, 37, 2010. (Cited on page xi.)

[146] Object Management Group (OMG). Catalog of UML profile specifications, n.d. http://www.omg.org/technology/documents/profile_catalog.htm (Accessed 21 August 2010). (Cited on page 55.)

[147] Object Management Group (OMG). UML profile for modeling qos and fault tolerance characteristics and mechanisms, version 1.1, n.d.

http://www.omg.org/cgi-bin/doc?formal/08-04-05.pdf (Accessed 21 August 2010). (Cited on pages 56, 66, 68 and 75.)

[148] Object Management Group (OMG). UML profile for schedulability, performance, and time specification, version 1.1, n.d. http://www.omg.org/cgi-bin/doc?formal/05-01-02.pdf (Accessed 21 August 2010). (Cited on page 64.)

[149] Object Management Group (OMG). Unified modeling language™ (UML®), n.d. http://www.omg.org/spec/UML/index.htm (Accessed 21 August 2010). (Cited on page 45.)

[150] Oracle Corporation. Oracle BPEL Process Manager, n.d. http://www.oracle.com/technology/bpel (Accessed 21 August 2010). (Cited on page 76.)

[151] Oracle Corporation. Oracle SOA Suite 11g, n.d. http://www.oracle.com/technologies/soa/soa-suite.html (Accessed 21 August 2010). (Cited on page 64.)

[152] Organization for the Advancement of Structured Information Standards (OASIS). Web services business process execution language version 2.0, 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (Accessed 21 August 2010). (Cited on pages 14 and 76.)

[153] Organization for the Advancement of Structured Information Standards (OASIS). OASIS, n.d. http://www.oasis-open.org/ (Accessed 21 August 2010). (Cited on pages 2, 28 and 69.)

[154] C. Ouyang, M. Dumas, A. H. M. ter Hofstede, and W. M. P. van der Aalst. From BPMN Process Models to BPEL Web Services. In *IEEE International Conference on Web Services*, pages 285–292, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on page 76.)

[155] S. Padmanabhuni and H. Adarkar. Security in service oriented architecture: Issues, standards and implementations. In P. Periorellis, editor, *Securing Web Services: Practical Usage of Standards and Specifications*, pages 1–21. Information Science Reference, Hershey, PA, 2008. (Cited on pages 14, 21, 23, 25, 27 and 28.)

[156] M. P. Papazoglou. Service-oriented computing: Concepts, characteristics and directions. In *Fourth International Conference on Web Information Systems Engineering*, pages 3–12. IEEE, 2003. (Cited on page 13.)

[157] M. P. Papazoglou. *Web Services: Principles and Technology*. Prentice Hall, Harlow, Essex, 2007. (Cited on pages v, vii, 1, 2, 14, 15, 16, 18, 28, 64, 65, 69, 72, 74, 75, 77 and 85.)

[158] M. P. Papazoglou and P. Ribbers. *e-Business: Organizational and Technical Foundations*. John Wiley & Sons Ltd., Chichester, West Sussex, 2006. (Cited on pages 2, 13, 19, 45, 55 and 85.)

[159] M. P. Papazoglou and W.-J. van den Heuvel. Business process development life cycle methodology. *Communications of the ACM*, 50(10):79–85, 2007. (Cited on page 18.)

[160] T. R. Peltier. *Information Security Risk Analysis*. Auerbach Publications, Boca Raton, FL, 2001. (Cited on page 156.)

[161] T. R. Peltier, J. Peltier, and J. A. Blackley. *Information Security Fundamentals*. Auerbach Publications, Boca Raton, FL, 2005. (Cited on pages 147 and 148.)

[162] R. M. Perloff. *The Dynamics of Persuasion: Communication and Attitudes in the 21st Century*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, second edition, 2003. (Cited on page 262.)

[163] Pi4 Technologies Foundation. Pi4SOA, n.d. http://pi4soa.wiki.sourceforge.net/ (Accessed 21 August 2010). (Cited on pages 64 and 111.)

[164] J. T. Pollock and R. Hodgson. *Adaptive Information: Improving Business Through Semantic Interoperability, Grid Computing, and Enterprise Integration*. Wiley-Interscience, Hoboken, NJ, 2004. (Cited on page 150.)

[165] M. E. Porter. Strategy and the internet. *Harvard Business Review*, 79(3):60–78, 2001. (Cited on page 12.)

[166] PricewaterhouseCoopers LLP and Infosecurity Europe. Information Security Breaches Survey 2010: Executive Summary, 2010. http://www.pwc.co.uk/pdf/isbs_survey_2010_executive_summary.pdf (Accessed 30 July 2010). (Cited on pages 19 and 239.)

[167] O. Prokein, T. Faupel, and D. Gille. Web services as an enabler for virtual organizations. In S. Barnes, editor, *E-Commerce and V-Business: Digital Enterprise in the Twenty-First Century*, pages 245–269. Butterworth-Heinemann, Oxford, 2nd edition, 2007. (Cited on pages 13, 18, 23 and 24.)

[168] E. P. Pulier and H. Taylor. *Understanding Enterprise SOA*. Manning Publications, California, 2005. (Cited on page 17.)

[169] P. Ratnasingam. *Inter-Organizational Trust for Business-To-Business E-Commerce*. IRM Press, Covent Garden, London, 2003. (Cited on page 35.)

[170] R. Richardson. 2007 CSI computer crime and security survey, 2007. http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pdf (Accessed 21 August 2010). (Cited on page 19.)

[171] R. Richardson. 2008 CSI computer crime and security survey, 2008. http://www.cse.msstate.edu/ cse6243/readings/CSIsurvey2008.pdf (Accessed 21 August 2010). (Cited on page 19.)

[172] A. Rodríguez, E. Fernández-Medina, and M. Piattini. Security requirement with a UML 2.0 profile. In *The First International Conference on Availability, Reliability and Security*, pages 670–677, 2006. (Cited on pages 56 and 68.)

[173] A. Rodríguez, E. Fernández-Medina, and M. Piattini. A BPMN extension for the modeling of security requirements in business processes. *IEICE - Transactions on Information and Systems*, E90-D(4):745–752, 2007. (Cited on page 56.)

[174] S. Röhrig and K. Knorr. Security analysis of electronic business processes. *Electronic Commerce Research*, 4(1-2):59–81, 2004. (Cited on pages 57, 58, 59 and 143.)

[175] RosettaNet. RosettaNet Home, n.d. http://www.rosettanet.org/ (Accessed 21 August 2010). (Cited on pages 13 and 19.)

[176] J. Roy, M. Barik, and C. Mazumdar. ESRML: a markup language for enterprise security requirement specification. In *IEEE INDICON*, pages 509–512, Kharagpur, 2004. (Cited on page 173.)

[177] W. W. Royce. Managing the development of large software systems: concepts and techniques. In *The 9th International Conference on Software Engineering*, pages 328–338, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press. (Cited on page 40.)

[178] H. J. Rubin and I. S. Rubin. *Qualitative Interviewing: The Art of Hearing Data*. Sage Publications, London, 2005. (Cited on page 230.)

[179] T. L. Saaty. *The Analytic Hierarchy Process*. McGraw Hill, New York, 1980. (Cited on pages vi, 260, 261, 263, 298, 299, 301, 302 and 303.)

[180] T. L. Saaty. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1):83–98, 2008. (Cited on pages 260, 263 and 298.)

[181] B. S. Sahay. Understanding trust in supply chain relationships. *Industrial Management & Data Systems*, 103(8):553–563, 2003. (Cited on pages 35 and 239.)

[182] F. Satoh, Y. Nakamura, N. K. Mukhi, M. Tatsubori, and K. Ono. Methodology and tools for end-to-end SOA security configurations. In *IEEE Congress on Services - Part I*, pages 307–314. IEEE Computer Society, 2008. (Cited on page 237.)

[183] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. Wiley, Indianapolis, 2004. (Cited on page 31.)

[184] M. Schumacher and U. Roedig. Security engineering with patterns. In *The 8th Conference on Pattern Languages of Programs (PLoP)*, Monticello, Illinois, 2001. (Cited on pages 58 and 59.)

[185] P. H. Schurr and J. L. Ozanne. Influences on exchange processes: Buyers' preconceptions of a seller's trustworthiness and bargaining toughness. *The Journal of Consumer Research*, 11(4):939–953, 1985. (Cited on page 24.)

[186] J. Sherwood, A. Clark, and D. Lynas. *Enterprise Security Architecture: A Business-Driven Approach*. CMP Books, San Francisco, CA, 2005. (Cited on pages 12, 36, 60, 66, 67 and 77.)

[187] N. Sidharth and J. Liu. IAPF: A framework for enhancing web services security. In *31st Annual International Computer Software and Applications Conference*, pages 23–30, Beijing, China, 2007. (Cited on pages 19, 28, 29, 70, 78 and 279.)

[188] Siemens Enterprise Communications Ltd. CRAMM, n.d. http://www.cramm.com/ (Accessed 30 July 2010). (Cited on page 47.)

[189] A. Singhal, T. Winograd, and K. Scarfone. Guide to secure web services (NIST Special Publication 800-95). Technical report, National Institute of

Standards and Technology (NIST), 2007. (Cited on pages vii, 1, 2, 17, 24, 26, 28, 30, 32, 70, 115, 237 and 279.)

[190] D. Skogan, R. Groenmo, and I. Solheim. Web service composition in UML. In *Eighth IEEE International Enterprise Distributed Object Computing Conference*, pages 47–57, 2004. (Cited on pages 55 and 76.)

[191] Software Engineering Institute, Carnegie Mellon University. Secure software development life cycle processes, 2009. https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/sdlc/326-BSI.html (Accessed 13 October 2010). (Cited on pages 36 and 37.)

[192] Software Engineering Institute, Carnegie Mellon University. TSP-Secure, nd. http://www.cert.org/secure-coding/secure.html (Accessed 13 October 2010). (Cited on pages 36 and 37.)

[193] I. Sommerville. *Software Engineering*. Addison-Wesley, Essex, eighth edition, 2007. (Cited on pages viii, 40, 41, 80, 162, 163, 165, 166 and 167.)

[194] 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. (Cited on pages v, 23, 25, 29, 30, 32, 33, 51, 57, 58, 59, 61, 69, 70, 77, 85, 94, 95, 104, 106, 107, 115, 116, 128 and 279.)

[195] G. Stoneburner, A. Goguen, and A. Feringa. Risk management guide for information technology systems (NIST special publication 800-30). Technical report, National Institute of Standards and Technology (NIST), July 2002. (Cited on pages v, 47, 92, 93, 145, 146, 147, 156, 158 and 159.)

[196] B. Suh and I. Han. The IS risk analysis based on a business model. *Information and Management*, 41(2):149–158, 2003. (Cited on pages 47, 145, 147 and 148.)

[197] R. Swart, K. Forcht, D. Olsen, B. Marshall, and M. Harris. Security at the edge: Rethinking security in light of web services. *Issues in Information Systems*, 6(2):103–109, 2005. (Cited on pages 1, 2, 13 and 24.)

[198] M. Tatsubori, T. Imamura, and Y. Nakamura. Best-practice patterns and tool support for configuring secure web services messaging. In *IEEE International Conference on Web Services*, pages 244–251, Athens, Greece, 2004. IEEE Computer Society. (Cited on pages 76 and 237.)

[199] Techinsurance, Inc. Contractedge.com - contracts, n.d. http://www.contractedge.com/contracts.html (Accessed 21 August 2010). (Cited on pages 101 and 108.)

[200] C. Teddlie and F. Yu. Mixed methods sampling: A typology with examples. *Journal of Mixed Methods Research*, 1(1):77–100, 2007. (Cited on page 229.)

[201] O. Tettero, D. J. Out, H. M. Franken, and J. Schot. Information security embedded in the design of telematics systems. *Computers & Security*, 16(2):145–164, 1997. (Cited on page 143.)

[202] The Open Group. TOGAF™Version 9, 2009. http://www.opengroup.org/togaf/ (Accessed 21 August 2010). (Cited on pages 36 and 277.)

[203] The SANS™Institute. Lab anti-virus policy. http://www.sans.org/resources/policies/Lab_Anti-Virus_Policy.pdf (Accessed 21 August 2010), 2006. (Cited on pages 85 and 95.)

[204] The Security Service. CRAMM User Guide (Version 5.1), 2005. http://dtps.unipi.gr/files/notes/2009-2010/eksamino_5/politikes_kai_diaxeirish_asfaleias/egxeiridio_cramm.pdf (Accessed 21 August 2010). (Cited on page 47.)

[205] J. S. Tiller. *The Ethical Hack: A Framework for Business Value Penetration Testing*. Auerbach Publications, Boca Raton, FL, 2005. (Cited on pages 20, 49, 127, 148, 158 and 246.)

[206] M. Todd, E. Zibert, and T. Midwinter. Security risk management in the BT HP alliance. *BT Technology Journal*, 24(4):47–52, 2006. (Cited on pages 127, 128, 233 and 238.)

[207] E. Triantaphyllou. *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers, Dordrecht, 2000. (Cited on pages vi, 259, 260, 262, 263, 299 and 303.)

[208] T. Tsiakis, E. Evagelou, G. Stephanides, and G. Pekos. Identification of trust requirements in an e-business framework. In *The 8th WSEAS International Conference on Communications*, Athens, Greece, 2004. (Cited on page 35.)

[209] B. Tsoumas and D. Gritzalis. Towards an ontology-based security management. In *20th International Conference on Advanced Information Networking and Applications*, volume 1, pages 985–992, 2006. (Cited on pages 152 and 153.)

[210] UK Department of Business, Enterprise and Regulatory Reform (BERR). 2008 Information Security Breaches Survey, 2008. http://www.pwc.co.uk/pdf/BERR_2008_Executive_summary.pdf (Accessed 21 August 2010). (Cited on pages 19, 46, 51 and 301.)

[211] W.-J. van den Heuvel, K. Leune, and M. P. Papazoglou. EFSOC: A layered framework for developing secure interactions between web-services. *Distributed Parallel Databases*, 18(2):115–145, 2005. (Cited on pages 30, 31, 33 and 279.)

[212] C. Van Slyke and F. Bélanger. *E-Business Technologies: Supporting the Net-Enhanced Organization*. Wiley, New York, 2003. (Cited on pages 35, 238 and 239.)

[213] H. Van Vliet. *Software Engineering: Principles and Practice*. John Wiley & Sons Ltd., Chichester, third edition, 2008. (Cited on pages 40, 41 and 163.)

[214] J. L. Vivas, J. Lopez, and J. A. Montenegro. Grid security architecture: Requirements, fundamentals, standards and models. In Y. Xiao, editor, *Security in Distributed, Grid, Mobile, and Pervasive Computing*, pages 255–288. Auerbach Publications, Boca Raton, 2007. (Cited on page 28.)

[215] A. Vorster and L. Labuschange. A new comparison framework for information security risk analysis methodologies. *South African Computer Journal*, (37):98–106, 2006. (Cited on pages 145, 146 and 147.)

[216] I. Walden. Privacy and data protection. In C. Reed and J. Angel, editors, *Computer Law: The Law and Regulation of Information Technology*, pages 459–504. Oxford University Press, New York, sixth edition, 2007. (Cited on page 51.)

[217] P. Wang, K.-M. Chao, C.-C. Lo, C.-L. Huang, and M. Younas. A fuzzy outranking approach in risk analysis of web service security. *Cluster Computing*, 10(1):47–55, 2007. (Cited on pages 1, 23, 30, 32, 33 and 279.)

[218] Web Services Interoperability Organization (WS-I). Security challenges, threats and countermeasures version 1.0, 2005. http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.pdf (Accessed 21 August 2010). (Cited on pages 2, 24, 70 and 115.)

[219] Web Services Interoperability Organization (WS-I). WS-I, n.d. http://www.ws-i.org/ (Accessed 21 August 2010). (Cited on pages 68 and 69.)

[220] P. Wongthongtham, E. Chang, and T. Dillon. Ontology modelling notations for software engineering knowledge representation. In *Inaugural IEEE International Conference on Digital EcoSystems and Technologies*, pages 339–345, Cairns, Australia, 2007. (Cited on page 155.)

[221] World Wide Web Consortium (W3C). Web service use case: Travel reservation, 2002. http://www.w3.org/2002/06/ws-example (Accessed 21 August 2010). (Cited on page 15.)

[222] World Wide Web Consortium (W3C). Overview of W3C technologies. http://www.w3.org/2003/Talks/0521-BudapestW3CTrack-IH/Overview.html (Accessed 21 August 2010), 2003. (Cited on pages vii and 16.)

[223] World Wide Web Consortium (W3C). Web services glossary, 2004. http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/ (Accessed 21 August 2010). (Cited on page 13.)

[224] World Wide Web Consortium (W3C). Web services choreography description language ver. 1.0. http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/ (Accessed 21 August 2010), 2005. (Cited on pages 14, 63 and 64.)

[225] World Wide Web Consortium (W3C). W3C, n.d. http://www.w3.org/ (Accessed 21 August 2010). (Cited on pages 2, 28 and 69.)

[226] Y. Yan and M. Klein. Web services vs. ebXML, an evaluation of web services and ebXML for e-business applications. In R. G. Qiu, editor, *Enterprise Service Computing: From Concept to Deployment*. IGI Publishing, Hershey, PA, 2006. (Cited on pages 17, 18 and 19.)

[227] S. S. Yau and Z. Chen. A framework for specifying and managing security requirements in collaborative systems. In L. T. Yang, H. Jin, J. Ma, and T. Ungerer, editors, *Autonomic and Trusted Computing*, volume 4158 of

Lecture Notes in Computer Science, pages 500–510. Springer, Heidelberg, 2006. (Cited on page 140.)

[228] W.-C. Yau and G. R. K. Rao. Web services security in e-business: Attacks and countermeasures. In G. Radhamani and G. R. K. Rao, editors, *Web Services Security And E-business*, pages 165–183. IGI Publishing, Hershey, PA, 2007. (Cited on pages 24, 25 and 29.)

[229] W. D. Yu, D. Aravind, and P. Supthaweesuk. Software vulnerability analysis for web services software systems. In *IEEE Symposium on Computers and Communications*, pages 740–748. IEEE, 2006. (Cited on page 78.)

[230] W. D. Yu, R. B. Radhakrishna, S. Pingali, and V. Kolluri. Modeling the measurements of QoS requirements in web service systems. *SIMULATION*, 83(1):75–91, 2007. (Cited on pages 69 and 116.)

[231] J. Zhang. Trustworthy web services: Actions for now. *IT Professional*, 7(1):32–36, 2005. (Cited on pages 1, 15, 17, 28 and 279.)

[232] F. Zhao. *Maximize Business Profits Through E-partnerships*. IRM Press, London, 2006. (Cited on pages 11 and 12.)

[233] Y. C. Zhou, X. P. Liu, E. Kahan, X. N. Wang, L. Xue, and K. X. Zhou. Context aware service policy orchestration. In *IEEE International Conf. on Web Services*, pages 936–943, 2007. (Cited on pages 17 and 24.)

[234] O. Zimmermann, M. R. Tomlinson, and S. Peuser. *Perspectives on Web Services: Applying SOAP, WSDL, and UDDI to Real-World Projects*. Springer, Berlin, 2003. (Cited on pages 13, 15, 18 and 75.)