Nurse, Jason R. C. and Sinclair, Jane E. (2009) *Securing e-Businesses that use Web Services - a Guided Tour Through BOF4WSS.* International Journal on Advances in Internet Technology, 2 (4).

Securing e-Businesses that use Web Services — A Guided Tour Through BOF4WSS

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

Abstract

Security in Web services technology itself is a complex and very current issue. When considering the use of 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 achieving a level of security beyond just technologies, that is trusted, endorsed and practiced by all businesses involved. In an attempt to address these problems, our research has previously introduced BOF4WSS [1], a business-oriented development methodology, specifically geared to guide e-businesses in defining, and achieving agreed security levels across collaborating enterprises. As that work was only an introduction, the aim of this paper is to provide detailed insight into what exactly BOF4WSS advocates and how these activities and processes aid in building security and trust.

Keywords: Security, Web services, e-business, systems development methodology, cross-enterprise interactions

1. Introduction

E-business has become the fastest growing means of conducting business in today's economy. In achieving the online business-to-business (B2B) collaboration between e-businesses, the use of services-oriented computing, by way of Web services (WS) technology, is playing an increasingly significant role [2]. The novel benefit is rooted in its ability to allow for seamless integration of business processes across disparate enterprises. This is due to the use of standardized protocols and open technologies [3]. One author [4] even states that the facilitation and automation of these business processes is the ultimate goal of Web services. As WS' use expands however, securing these services becomes of utmost importance.

In an attempt to address new security challenges accompanying WS, standard-setting bodies have proposed numerous pioneering standards. As WS matures, the move from lower level security details such as standards and technologies, to higher level considerations however, is imminent [5]. Security, irrespective of the context, is a multilayered phenomenon encompassing aspects such as practices, processes and methodologies. This factor is especially true with WS which, as authors [6] note, substantially complicates the security environment for e-businesses.

Considering this, and with special appreciation of the inter-organizational security issue now facing businesses interacting using WS, in previous work we have introduced the Business-Oriented Framework for enhancing Web Services Security for e-business (BOF4WSS) [1] to address some of these issues. At its core, this framework supplies a cross-enterprise development methodology that can be used by businesses—in a joint manner—to manage the comprehensive concern that security in the WS environment has become. Building on the introduction to BOF4WSS given in that work therefore, this paper presents thorough coverage of the framework, its ideas, the tasks involved, and also their justifications.

The remainder of this extended paper is structured as follows: Section 2 contains a brief review of the security advancements in WS use for e-business with the aim of identifying outstanding security issues, and therefore paving the way for BOF4WSS. Next in Section 3, a detailed discussion of the framework, including its novelty and use, is given. Conclusions and future work will be outlined in Section 4.

2. Web Services Security within e-Business

2.1. State of the Art

Albeit a promising implementation technology for the Service-Oriented Architecture (SOA), and an increasingly used enabler of e-business, WS comes at a high price of an unstable security foundation. The literature identifies numerous challenges [5], [7], but the most pertinent for our research is the reality that WS adds significant complexity to the e-business security landscape [6]. This complexity makes security 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 mean an increased, real-time security risk for its partners—both immediate and extended.

To address the new security challenges mentioned above, consortiums such as OASIS and W3C have developed and ratified numerous pioneering standards (as can be seen in [5]). 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. Beyond addressing the perceived inadequacies of the current standards base, researchers are now targeting the more general components of a security solution such as best practices and processes. These actions give life to a prediction made by NIST, which emphasized that as WS technology matured, methodologies and recommended practices for security would become the next step in the goal of developing secure systems [5].

Some of the most pertinent, and noteworthy proposals focusing on these higher layers are: [8], which builds on existing technologies and the theory of Aspect-Oriented Programming, to provide a framework for securing WS compositions (necessary in collaborative e-business) using the WS-Security and WS-Policy standards; [9] aims to provide a methodical development approach for constructing security architectures for WS-based systems; [10] which provides integrated WS design strategies and best practices for end-to-end security; [11] – 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; and lastly the Event-driven Framework for Service Oriented Computing in [12] – 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 [12].

2.2. Outstanding Security Issues

WS security approaches should aim to be thorough in planning, developing and maintaining an adequate solution. Standard security components encompass technologies, but as recent literature [13] in the study of security in general has emphasized, it also includes policies, processes, methodologies, and best practices. To WS' detriment however, this fact does not appear to be unanimously shared as any attention on these other aspects is being drowned out by the proliferation of various new technology standards. One can easily see this fact when comparing the few higher layer approaches mentioned in [1] to the vast number of standards and technical systems highlighted in [5]. It may therefore be very tempting to regard such mechanisms as the 'solutions' to the WS security problem. Whilst the work of technologists is valuable to building security and trust however, alone they cannot form the entire solution. 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 in the goal of reliable, comprehensive, multilayered security. This perspective is supported by authors in [5] and they identify processes such as effective risk management, and defence-in-depth through security engineering, as critical to developing robust, secure systems.

A final concern regarding standards is that there are already too many available [14]. Therefore, as opposed to benefiting WS, this plethora of sometimes overlapping standards ultimately confuses developers and acts to complicate secure WS implementation and use. The importance of these factors is magnified when assessing WS use for the already complex field of e-business.

To briefly assess the aforementioned research in [8], [9], [10], [11], these are all seen to successfully complement available technologies, and provide useful security approaches. Their main caveat however is that they consider security predominantly from one company's internal viewpoint, i.e., what should a company 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 readily facilitates. In [12], even though this 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 for our purposes as mentioned before.

Looking beyond these advancements, an intriguing research area which has received little emphasis is at the level of *cross-enterprise interaction* (i.e. interactions spanning, and including collaborating businesses and their internal systems). Specifically, we refer 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 be solely at the technical level but look generally at a number of other fundamental aspects (e.g. security directives, policies, government regulations, best practice security standards, business risk considerations, and negotiations necessary) 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 [6].

The basic notion behind such a proposal can been seen in the largely exploratory research study done in [15]. In that article, the authors accepted the comprehensive security dilemma e-businesses face and proposed a generic model to enhance security. In many respects our research's general proposals are an extension of that exploratory work, to delve into the intricacies of what would constitute such a comprehensive security approach.

Further to the previously mentioned goals, this new approach would also aim towards facilitating the increased trust in business partners, their systems, and the overall service interactions, as an intrinsic objective. The importance of trust in e-business (with or without WS) is stressed by several authors [16] 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 [17], [18]. This approach would aim to foster trust between partners, their systems (which are no longer 'black boxes' to partners), and the overall service interactions, by stressing these and related factors.

With regards to security and trust in general, the approach could be seen to facilitate a level of 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 complete security solution. In discussing the general topic of WS' usage for B2B, Alonso et al. [19] 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". 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. Typical areas in which cross-enterprise security might be a such an important goal would be businesses with substantial and long-term investments. Also, companies bound to strict contractual or government regulations that must be enforced. And lastly, businesses that deal with mission-critical systems, such as the health or banking sectors.

In summary, there are a number of unaddressed issues as e-businesses look towards creating, and maintaining a comprehensive, trustworthy WS security solution. Primarily these stem from (i) an overly reliant emphasis on technology, alluding to standards and systems as the complete solution to WS security, and (ii) an overly isolated security stance, focusing on the process *one* company should follow to secure itself internally, therefore ignoring the comprehensive security issue introduced by WS use. As was previously mentioned in Section 1, to address these issues, BOF4WSS was proposed. The goal of the next section therefore is to expand on the introduction in [1] and provide an in depth look at the inner workings and activities in BOF4WSS

3. BOF4WSS

3.1. Overview

To address the outstanding security issues above, and strengthen available solutions, the Business-Oriented Framework for enhancing Web Services Security for e-business (BOF4WSS) displayed in Figure 1 was conceived. As is illustrated, 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.

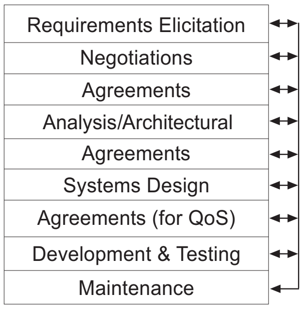

Figure 1. BOF4WSS Overview

The Waterfall Model (WM) methodology in particular was the main influence for the framework's design. This can be seen when comparing BOF4WSS's phases to those of the WM i.e. system feasibility study, requirement analysis and project planning, system design, detailed design, coding, testing and integration, installation, and maintenance [20]. The WM was preferred to other methodologies due to the transparent, well-organized, highly documented, and strongly disciplined process it can bring to this large inter-organizational development project [20], [21]. Some practitioners even view the structure possible with the WM as an ideal fit for the corporate (and somewhat bureaucratic) world, and a key reason why the WM is here to stay [22].

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 as a foundation however, because literature [23], [24] 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 and also, later sections.

Despite the benefits listed, it is accepted that the WM is not perfect and does have shortcomings. For example, researchers have identified that it freezes requirements too early, lacks flexibility in the original model when traversing stages, and results in excessive documentation [20]. As opposed to adopting a different methodology however, BOF4WSS addresses these shortcomings by allowing for flexibility through bottom-up progression and feedback (shown on the right in Figure 1), and stressing the involvement of key stakeholders throughout the entire process. 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 can, and may change at subsequent stages closer to design. The inclusion of the Negotiations and various Agreements stages at the points specified is necessary due to the inter-organizational process, and the importance of companies discussing and agreeing on goals.

The prime novelty in BOF4WSS is the emphasis on providing an expanded formalization of a development methodology that focuses on security, which can accommodate multiple autonomous businesses working together. As will be seen below, the framework and its phases give detailed guidance on what should occur and how, and its pertinence 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 to a company simply mean interactions that occur in transit (i.e. 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 below.

Returning to the point regarding the 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 reuse existing methods and practices—both from industry and academia—thus concentrating on the compilation of these into a coherent, well-defined process instead of reinventing standardized parts of the proverbial security wheel.

Another main design goal of the framework is to promote/utilize Web services specifications and tools wherever, and whenever useful. This is done to provide companies that adopt BOF4WSS with a practical methodology that pulls together key WS-specific specifications and tools from the plethora of technologies available, and shows exactly where and how they can fit into the development of a Web services solution. To date, the authors are not aware of such a broad methodology as BOF4WSS, which aims to fit together some critical pieces of the WS security puzzle in the context of cross-enterprise, highly structured, extensible (by allowing different approaches to be plugged in), business-oriented framework.

BOF4WSS's close alignment with Web services, security, and cross-enterprise development, differentiate it from somewhat related, existing frameworks and models such as TOGAF [25] (a detailed method and a set of supporting tools for developing an enterprise architecture), SABSA [26] (a framework for delivering cohesive information security solutions to enterprises), and the Web services development lifecycle [4]. These are all very adequate, de facto approaches, but aim at a much more generic level than is of interest in this research. Otherwise, there are various similarities between these models and BOF4WSS, including identification and involvement of key stakeholders, definition of conceptual data models for foreseen interactions, phase inputs and outputs, and architectural design and technical level implementations.

To support the largely textual description of the framework's activities below, 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, e.g. performance, scalability, maintainability, and so on.) At some stages however, in the interest of completeness, this paper does attempt to give some guidance on these areas. This is particularly when they relate to key 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.

3.2. Requirements Elicitation Phase

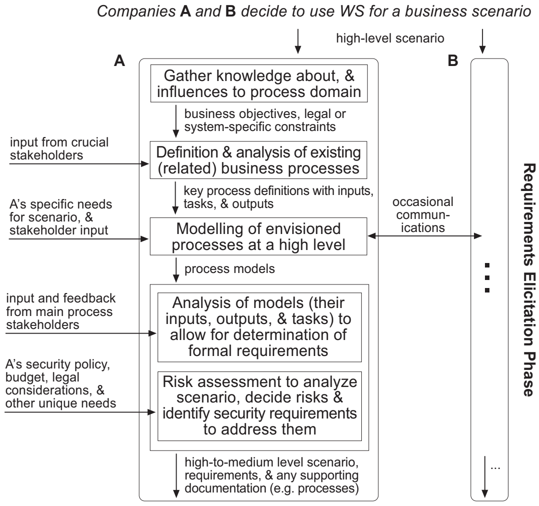
Companies A and B decide to use WS for a business scenario
high-level scenario
A
B
Gather knowledge about, & influences to process domain
business objectives, legal or system-specific constraints
input from crucial stakeholders
Definition & analysis of existing (related) business processes
key process definitions with inputs, tasks, & outputs
A's specific needs for scenario, & stakeholder input
Modelling of envisioned processes at a high level
occasional commun-ications
process models
input and feedback from main process stakeholders
Analysis of models (their inputs, outputs, & tasks) to allow for determination of formal requirements
A's security policy, budget, legal considerations, & other unique needs
Risk assessment to analyze scenario, decide risks & identify security requirements to address them
high-to-medium level scenario, requirements, & any supporting documentation (e.g. processes)
...
Requirements Elicitation Phase

Figure 2. Workflow model of the Requirements Elicitation phase

The **Requirements Elicitation phase** is displayed in Figure 2 and assumes two companies, A and B; more companies however are possible. Within this first phase, each company works largely by itself, analyzing internal business objectives, constraints, security polices, relevant laws and regulations and so on. This is done to determine their high-level requirements for the expected WS business scenario. Typically, a company team should be assembled that would be responsible for project management, system development, cross-enterprise communications, and generally steering and championing the project from inception to fruition.

To aid in the Requirements Elicitation process, the phase utilizes the methods proposed by [27], which focus on the definition and analysis of business process models to elicit requirements (functional, quality, and security-specific). This approach is preferred as it is a tested technique that also has an innate emphasis on business processes—i.e. the culmination of service interactions. During these methods, as with some of the subsequent stages, the framework heavily stresses the involvement of stakeholders, and especially top management buy-in (i.e. support). Validated by studies in [28], these are critical success factors in managing and developing information systems.

As illustrated in Figure 2, the approach in [27] 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. Legacy applications are an important consideration at this point because these are likely to supply critical functionality in the foreseen scenario. These systems will therefore have to be thoroughly understood, and their business functionality/logic rationalized and defined. This is particularly useful in the next task as legacy applications functions are packaged, modelled and included in envisioned processes.

Crucial persons (i.e. stakeholders) of reference for the information mentioned will be top executives, domain experts, project managers, systems analysts and end users. For the modelling activity in this second task, the Unified Modeling Language (UML) 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 (i.e. there are no ‘current processes’ specifically related to the envisioned interactions), this task will not be as relevant. Instead, the aim will be to consider how their internal processes will integrate with this 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 [29]. The choice between these methods will largely be dependent on benefit versus cost, and whether legacy systems can adequately fulfill new business goals.

Generally, the processes defined in this task are expected to be high-level, and mainly cover internal (i.e. known) as opposed to external (i.e. 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 is required to enable useful processes to be defined. Also, again UML is suggested for (i) the reason above, 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 for high- or medium-level processes to be defined.

The last task in the approach proposed in [27] is the actual requirements determination. This is accomplished through analysis of the newly defined process models. By assessing the inputs, outputs, and tasks involved, general requirements (functional and quality-based) for each stage of the process can be defined at 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, but businesses should make an effort to give some idea of their desires for system quality. To elicit security-specific requirements, the authors mainly analyze the access restrictions of the actors (users or applications) on the processes, and process inputs and outputs. In these last two stages, BOF4WSS heavily involves the previously highlighted stakeholders.

In addition to the security requirements identified above, a scenario risk assessment is strongly suggested to provide more detailed and extensive security information. This assessment, as opposed to the one above which focuses primarily on access restrictions in processes, enables for a comprehensive, security-driven analysis of the scenario. The assessment is strongly suggested primarily to combat the unfortunate reality that a significant number of businesses simply do not carry out formal security risk assessments to identify key risks faced [30]. Or, if companies do engage in risk assessments, studies show that major gaps in risk assessment coverage often are apparent that could result in significant risks being overlooked [31]. To aid in this process, there are a range of assessment methods. BOF4WSS suggests well-documented, and internationally validated, time-tested techniques such as NIST Special Publication 800-

30: Risk Management Guide [32], OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) [33], and CORAS (Construct a platform for Risk Analysis of Security Critical Systems) [34].

Generally, some of the crucial factors considered in a chosen technique should include risks (constituted of assets, threats, vulnerabilities) and their priority levels (i.e. severity and impact if risks materialize), organizational security policies (which 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 needs expected to be met by new business partners. All of these factors significantly aid in the determination of the security that should be factored in during these envisioned WS communications. These requirements 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—i.e. *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*.

3.3. 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 discussion and negotiations. Figure 3 displays the workflow. The purpose is to use the inputs to this stage as a basis to chart an **agreed** path forward in terms of scenario and business requirements, and high-to-medium level process definitions. 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 is it to be secured, how will security be applied, and so on. Specifically the two main tasks in this phase therefore are: Discussion and negotiation on (i) functional and quality requirements, and then (ii) security actions and requirements. 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 thus 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.

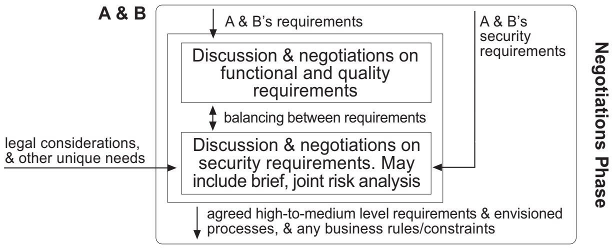
A & B
A & B's requirements
A & B's security requirements
Discussion & negotiations on functional and quality requirements
balancing between requirements
legal considerations, & other unique needs
Discussion & negotiations on security requirements. May include brief, joint risk analysis
agreed high-to-medium level requirements & envisioned processes, & any business rules/constraints
Negotiations Phase

Figure 3. Workflow model of the Negotiations phase

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), and 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 in [35] clearly highlights that in forming these extended networks or partnerships of companies, this integration task is formidable. Regardless however, this is 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*.

3.4. Agreements Phase

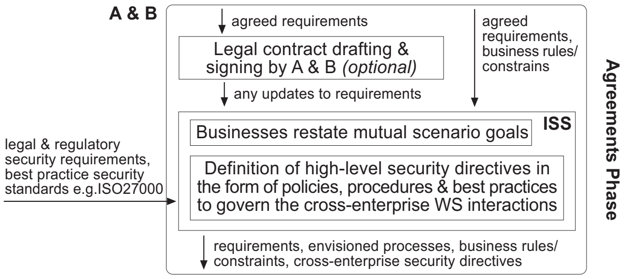
A & B
agreed requirements
agreed requirements, business rules/ constrains
Legal contract drafting & signing by A & B (optional)
any updates to requirements
ISS
Businesses restate mutual scenario goals
legal & regulatory security requirements, best practice security standards e.g.ISO27000
Definition of high-level security directives in the form of policies, procedures & best practices to govern the cross-enterprise WS interactions
requirements, envisioned processes, business rules/ constraints, cross-enterprise security directives
Agreements Phase

Figure 4. Workflow model of the Agreements phase

The **Agreements phase** depicted in Figure 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, or to have only one main contract at a later stage when details of interactions are finalized. 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 agreement and definition of requirements in the Negotiations phase makes the latter of these tasks (i.e. defining the groundwork) less complex and arduous.

This legal document is followed by the Interaction Security Strategy (ISS) which, as opposed to the contract, 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. Figure 4 shows that the central activities in this stage are: (i) the restating of the businesses' mutual goals for the scenario—this will provide a clear vision for the strategy; and (ii) the actual definition of the security strategy's directives. In addition to the use of requirements, and business constraints, when defining these directives, the framework emphasizes consideration of two aspects, i.e. 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. This is 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 fulfill the statutory need, but also because a number of these laws stress principles of good, reliable security that should be practiced by businesses.

Some of the most relevant laws businesses should consider include the Sarbanes-Oxley Act (SOX) of 2002 (U.S.)—this emphasizes the maintenance of adequate internal controls to ensure the accuracy of financial information [10], [36]; 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 [36]; Data Protection Directive 95/46/EC of 1995 (E.U.)—this is targeted towards personal data, ensuring that it is adequate, accurate, and processed lawfully, amongst other things [10]; and Gramm-Leach-Bliley Act (GLB) of 1999 (U.S.)—mainly aimed at financial institutions, this act 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 [36]. 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 to legal and regulatory requirements, the ISS emphasizes the incorporation of best practice security 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 [30] 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 4 shows, they form a key input into this stage. This standards set in particular, is targeted at the provision of an internationally recognized, organization independent framework for effective, extensive information security management [37]. Themes addressed include the definition of essentials (in terms of specifying information security policies, conducting in-depth risk assessments, and so on) for the creation of an adequate information security management system (formally, the ISMS)—this is covered in ISO/IEC 27001:2005; a code of recommended best practices for planning and implementing the ISMS—see ISO/IEC 27002:2005; and detailed guidelines for the information security risk management process to support the ISMS—see ISO/IEC 27005 [37]. The creation and maintenance of a well-conceived, and thorough internal security management system for an organization is the fundamental objective of this standards set.

To put the ISS directives (e.g. laws, standards, and policies) into context, Figure 5 is included below. This illustration, based on work in [38], shows how each aspect covered by the ISS fits in to provide a layered model for the e-business security environment.

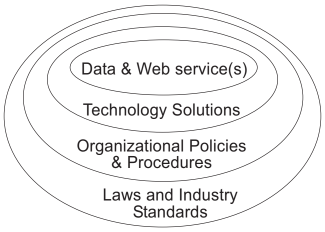
Data & Web service(s)
Technology Solutions
Organizational Policies & Procedures
Laws and Industry Standards

Figure 5. The e-business security environment (based on [38])

Looking directly at the ISS' emphasis on standards during this Agreements phase, there are countless benefits. As mentioned prior, the term *cross-enterprise interactions* denote 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 which can demonstrate adherence to guidelines, can be issued with a certificate to show customers and business partners that their systems and practices are secure to an international standard [37].

At the *external* level these standards also prove useful as certain clauses (e.g. ISO/IEC 27001, Control A.6.2) deal specially with external parties, and attempting to maintain the security of an organization's information assets as they are accessed, processed, communicated to, or managed by external parties [37]. 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 brief look is taken at some examples of what the ISS could cover. The first example is the specification of best practices each company should abide by internally. One best practice might be related to ensuring companies maintain sufficient logs of system events; 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 i.e. what procedures should companies 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 updating 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 constituting of persons from both 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, i.e. 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 (see Section 2.2), and its importance should not be neglected. This paper does note other, more direct methods to assess a business partner's commitment to security, such as audits, on-site visits, and questionnaires (as suggested in [36]), 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; this opinion is supported by [39]. 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 this Agreements phase the following documents and information are prepared to be 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).*

3.5. Analysis/Architectural Phase

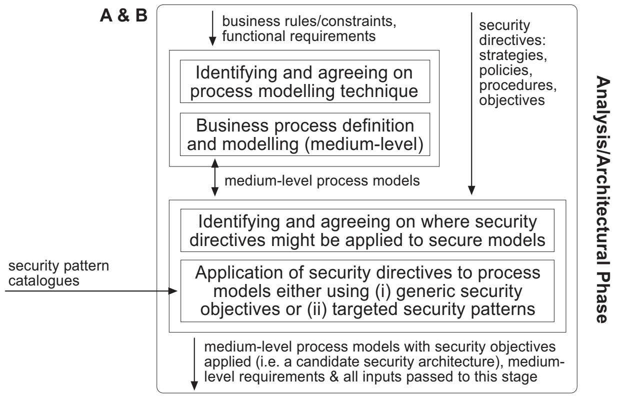
A & B
business rules/constraints, functional requirements
security directives: strategies, policies, procedures, objectives
Identifying and agreeing on process modelling technique
Business process definition and modelling (medium-level)
medium-level process models
Identifying and agreeing on where security directives might be applied to secure models
security pattern catalogues
Application of security directives to process models either using (i) generic security objectives or (ii) targeted security patterns
medium-level process models with security objectives applied (i.e. a candidate security architecture), medium-level requirements & all inputs passed to this stage
Analysis/Architectural Phase

Figure 6. Workflow model of the Analysis/Architectural phase

Following on from agreements, next is the **Analysis/Architectural phase**. The workflow of this stage is given in Figure 6. This phase's purpose to enable companies to take the agreed requirements, and define conceptual (medium-level) 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 analyze 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. With this in place, the directives (policies, best practices, and so on) from the ISS can then be applied to secure the models. This two-stage method to securing business processes is adopted from research done in [40], which focused on decomposing processes into flows with inputs and outputs, then applying derived security objectives (these are encompassed in our security directives) to secure process components. [41] is an example of other work which adopts a similar, stepped approach to secure e-business processes during design.

To define the medium-level business process models needed, various standard modelling techniques are available (see [29], [42], [43]). Some of the most popular of these are UML (inclusive of its many specialized profiles), Data Flow Diagrams (DFD), Integration Definition for Function Modeling (IDEF) techniques (e.g. IDEF0, IDEF1x, IDEF3), and the Business Process Modeling Notation (BPMN). The UML 2.0 extension for SOA, UML4SOA [44], is a recent proposal from research groups which also provides an interesting technique. This profile however appears to be only targeted at service orchestration (i.e. internal, as opposed to cross-enterprise systems). Yet another option is the UML profile in [45] for Web service 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 (e.g. BPEL, albeit an older version)—a necessary task in future stages.

Having mentioned the SOA, the framework notes that businesses may or may not model processes in terms of *services* at this point. The notion of a *service* here refers to its abstract meaning i.e. distinct units of logic [46]. This is therefore the conceptual prerequisite to actual technology-based Web services. If modelling in terms of abstract services, businesses for example might start defining functionality or processes to package together to form referable units of logic. If initial modelling in previous phases define components to encapsulate legacy system functionality, these could be starting points to regard as services. The benefits of service modelling at this point, is that it could give early insight to where services are likely to fit in, and secondly, that it forces businesses to view processes in terms of services early on. If a company is looking towards full adoption of an SOA framework internally, the latter of these benefits is more crucial.

Although services modelling is an option, there is arguably no need for companies to rush into the services creation task as yet. This is because the forthcoming Design phase which covers a lower level of analysis, addresses this concern in detail. As with many other parts of BOF4WSS however, the final decision is left up to businesses and what suits their ideology and situation best. For the more standardized techniques above, [47] provides a brief outline of the software and tools available to support modelling. The importance of tools cannot be stressed enough, as these are critical in streamlining and easing the modelling process for companies.

Since it is almost certain that companies would have engaged in process modelling at some point before, they are likely to have preferred techniques; because of this, first an agreement is required on the technique that they will use. As can be concluded from Figure 6, the framework does not stipulate that any particular method be used. It however does advice businesses to carefully deliberate the benefits, and shortcomings of the options available. Any assessment should bear in mind: (i) the goal of this phase i.e. 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 (e.g. security or scheduling constraints) at lower level, and therefore having standard ways to state these aspects may be beneficial; and (iii) the impending need to translate (with, or without tool support) these process models into more WS-specific formats, for external and internal usage. Regarding the latter two points, businesses for example might find it useful to know that firstly, there have been proposed extensions to UML to account for security, and secondly, 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 7 is one guide that can be supplied to companies to give a summary view of UML and BPMN with respect to the options for modelling security. Information on 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 [48], [49], [50] respectively.

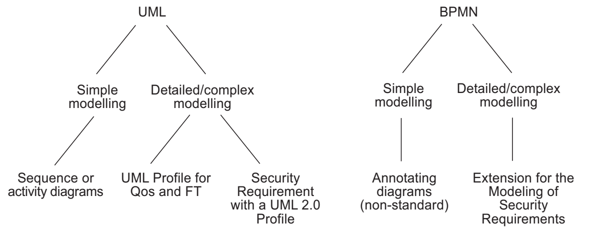
UML
Simple modelling
Detailed/complex modelling
Sequence or activity diagrams
UML Profile for Qos and FT
Security Requirement with a UML 2.0 Profile
BPMN
Simple modelling
Detailed/complex modelling
Annotating diagrams (non-standard)
Extension for the Modeling of Security Requirements

Figure 7. Options for modelling security with UML and BPMN

Researchers in [42] and [43] have investigated into the nuances of a number of popular process modelling techniques, and their findings would be a first point of reference (used 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 [42]. Purposes could range from functional (task-focused) to informational (data flow-based), or from process development to simply enabling for 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 [43]. For more on BPMN and UML4SOA, see [29] and [44] respectively. All of the information on these techniques from resources identified above can be used by businesses to aid in the selection of the most appropriate process modelling approach to suit their specific organizations and needs.

Once the modelling technique has been agreed, Figure 6 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 (i.e. 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 process at an abstract but holistic level, which is comprehensible by all members of the companies' teams, as opposed to only systems designers or software developers. Conceptual process definition can allow companies to analyze processes, weigh alternatives, and assess process inter-relations. 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.

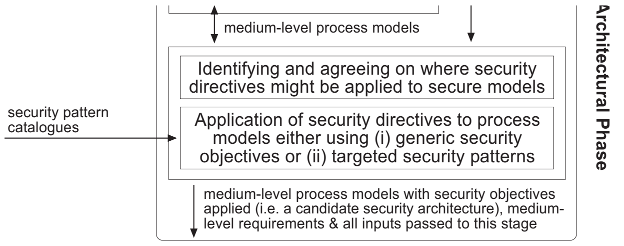
medium-level process models
Identifying and agreeing on where security directives might be applied to secure models
security pattern catalogues
Application of security directives to process models either using (i) generic security objectives or (ii) targeted security patterns
medium-level process models with security objectives applied (i.e. a candidate security architecture), medium-level requirements & all inputs passed to this stage
Architectural Phase

Figure 8. Identification and application of directives in the Analysis/Architectural phase

After defining the cross-enterprise process models, the next general task presented in Figure 8 is to apply the security 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 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 security directives to the processes they will affect. The framework accepts that not all directives may be process-specific or -related (for e.g. monthly updates on ISS). When the matching has been completed, there are two methods in which directives can be actually applied to the process models, these are either (i) through the use of generic *security objectives* (as done in [40]) or (ii) by employing targeted *security patterns* (see [10]). These two methods are preferred in BOF4WSS because they provide decent 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 9 diagrammatically presents the general process.

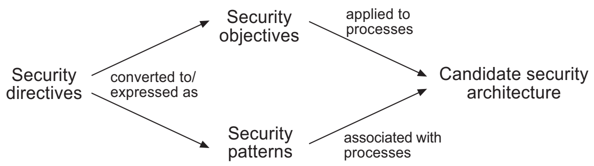
Security directives
converted to/ expressed as
Security objectives
Security patterns
applied to processes
associated with processes
Candidate security architecture

Figure 9. Process from security directives to security architecture

To use the first approach (i.e. [40]) in its original form, companies will have to ensure that process-related security directives are stated with regard to the *security objectives* of confidentiality, integrity, availability, and accountability. This however is not a limitation, because the framework does appreciate and support the desire of businesses to add other, possibly relevant objectives that reflect the directives. These additions might include objectives on nonrepudiation, authentication, and authorization for example. After this is complete, individual process components (i.e. inputs, outputs, activities, and actors—users of process activities) are assigned rating values (for e.g. 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 an assignment; if a data value α 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 [10]. 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" [51]. 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 8. Authors in [10] have investigated this topic in detail, and have provided an extensive listing of existing and new patterns spanning the Web, Business, Web services, and Infrastructure and Quality of Services tiers of a typical company's systems. Using the example of data value α from the previous approach above, personnel at companies 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 Design Phase. The goal at this Architectural stage therefore (as illustrated in Figure 9 and also done in [10]) is mainly the identification of relevant security patterns.

To briefly compare the security objectives [40] and security patterns [10] 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 for the security priorities (e.g. 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 companies chose 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 overly complex problems [51]. An additional benefit of using the pattern catalogue in [10] specifically is that it largely is geared towards Web services 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*.

3.6. 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 10. At this point, the agreement is in the form of a more thorough legal contract reflecting detailed expectations of the parties included in the envisaged business 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. During contract drafting, it is accepted that requirements may change, and therefore any updates made are 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 interactions to the ISS. Many authors [17], [26] support this and similar views, and highlight 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*.

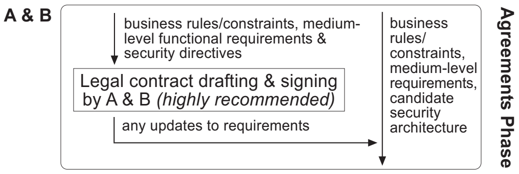
A & B
business rules/constraints, medium-level functional requirements & security directives
Legal contract drafting & signing by A & B (highly recommended)
any updates to requirements
business rules/ constraints, medium-level requirements, candidate security architecture
Agreements Phase

Figure 10. Workflow model of the Agreements phase

3.7. Systems Design Phase

The **Design phase** next is analogous to a company's internal systems design process (for e.g. see [10]) 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. In Figure 11 the specific tasks in this stage are presented diagrammatically. As is shown, the first activity is for the teams from each business to jointly define the low-level process 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 can be seen, and also the specific tasks which constitute each process activity. 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. Work in [46] is one commendable reference that examines moving from business processes to service models and designs, which 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 12 is used.

This diagram shows a simplified medium-level process

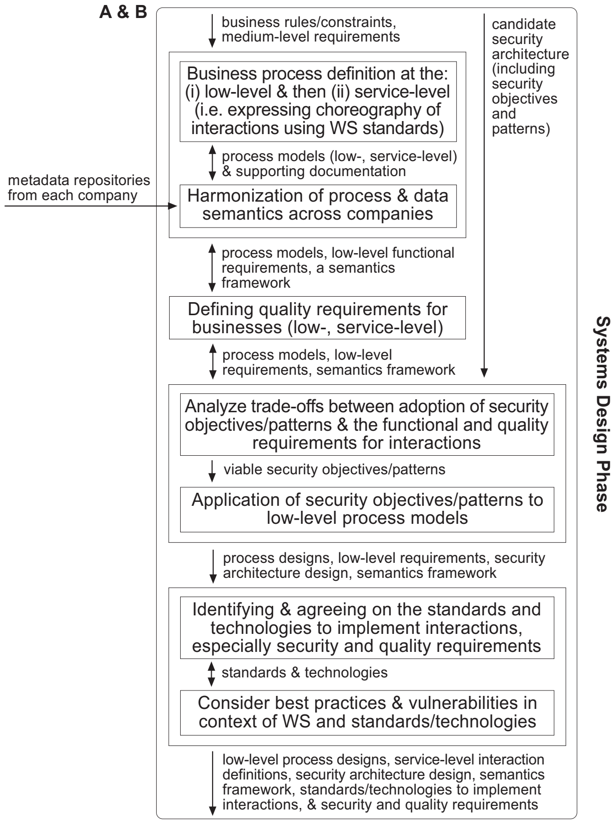
A & B
business rules/constraints, medium-level requirements
candidate security architecture (including security objectives and patterns)
Business process definition at the: (i) low-level & then (ii) service-level (i.e. expressing choreography of interactions using WS standards)
process models (low-, service-level) & supporting documentation
metadata repositories from each company
Harmonization of process & data semantics across companies
process models, low-level functional requirements, a semantics framework
Defining quality requirements for businesses (low-, service-level)
process models, low-level requirements, semantics framework
Systems Design Phase
Analyze trade-offs between adoption of security objectives/patterns & the functional and quality requirements for interactions
viable security objectives/patterns
Application of security objectives/patterns to low-level process models
process designs, low-level requirements, security architecture design, semantics framework
Identifying & agreeing on the standards and technologies to implement interactions, especially security and quality requirements
standards & technologies
Consider best practices & vulnerabilities in context of WS and standards/technologies
low-level process designs, service-level interaction definitions, security architecture design, semantics framework, standards/technologies to implement interactions, & security and quality requirements

Figure 11. Workflow model of the Systems Design phase

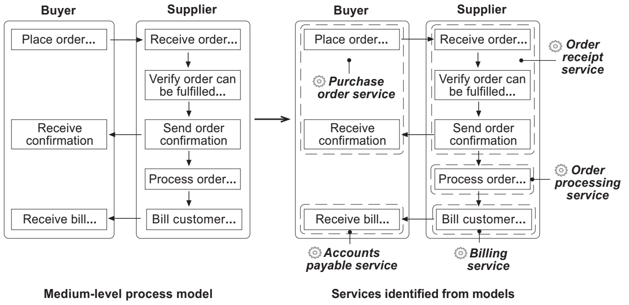

Figure 12. An example of moving from processes to services

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 purported might be primarily of the interactions *between* companies, or the interactions *between and also within* the businesses. To use Figure 12 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 companies throughout the framework's activities is largely left to the individual teams, BOF4WSS stresses that openness and transparency could foster trust between these companies. This trust will be a key ingredient to successful future business interactions.

Building on the low-level process definitions, Figure 11 shows that the following task 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 Web services 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 (i.e. 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 WS message exchanges can take place [52]. 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 [52]; 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 [53], [54]. 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. ebXML's Business Process Specification Schema (BPSS) is another popular option that can specify business transactions and their choreography [29]; this method is not preferred because of BOF4WSS's aim to primely utilize WS-specific technologies.

In deciding whether to use WS-CDL or BPEL4Chor, the framework highlights the following factors for consideration by businesses. This paragraph assesses WS-CDL and the next, BPEL4Chor. 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. (Notably however, a concise research survey in [55] provides a counter-argument to this assumption as they state that interest in this specification has dwindled.) A second advantage of WS-CDL is that, from the WS-CDL document defined, companies are largely able to generate BPEL workflow templates for their *internal* process flows, that reflect the global business agreement [4], [56]. Third, because WS-CDL leaves actual implementation decisions and details to companies, it allows them the flexibility to use preferred internal technologies. A high-level example is given in [52], 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 [57] to model processes in the Architectural phase, research work in [58] has investigated a method for translating those models into WS-CDL documents. This could therefore be plugged in, and used by companies to automate document creation. Lastly, there is some (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 [59], Pi4SOA [60] and LTSA WS-Engineer [61]. These could be employed by companies to assist in creating and testing the WS-CDL definitions.

At its core, the second approach i.e. BPEL4Chor, defines extensions to BPEL to enable the definition of choreographies [53]. 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 [53]. A second noteworthy factor is that if businesses have used BPMN to model processes in previous stages, research in [54] 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. [53] should be referenced 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, companies can chose their technologies of preference.

Along with the low-level process definition shown in Figure 11, harmonization of process and data semantics across companies is critical. In this paper however, this activity is not covered as it would necessitate an extensive discussion that digresses considerably from the overall 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 in [4]. 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, ebXML process definitions, and RosettaNet's Partner Interface Processes (details of each, available in [4]).

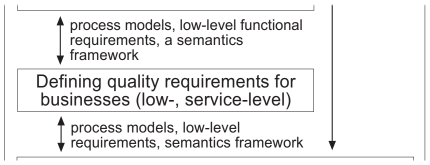
process models, low-level functional requirements, a semantics framework
Defining quality requirements for businesses (low-, service-level)
process models, low-level requirements, semantics framework

Figure 13. Definition of quality requirements task in the Systems Design phase

Following the definition of processes and harmonization of semantics, the goal switches to the determination of the quality requirements at these lower levels. For ease of reference Figure 13 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, businesses, especially their analyst and systems designers play central roles. Business teams need to decide details such as availability (or uptime) of systems and services, acceptable latency levels, performance expectations by parties, and more general aspects including usability, scalability, and even maintainability of envisioned systems. [62] has compiled an appropriate listing of WS quality of service attributes that can be used as a starting point by teams. It is important that these requirements and their relation to processes be well thought through 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 of service characteristics (see [48]) that can be used.

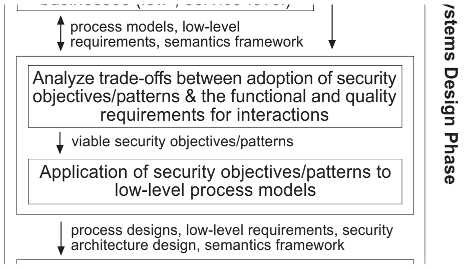
process models, low-level requirements, semantics framework
Analyze trade-offs between adoption of security objectives/patterns & the functional and quality requirements for interactions
viable security objectives/patterns
Application of security objectives/patterns to low-level process models
process designs, low-level requirements, security architecture design, semantics framework
stems Design Phase

Figure 14. Security analysis and application tasks in the Systems Design phase

The next step in BOF4WSS (an excerpt is shown in Figure 14) 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 analyzing 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. Work in [26] yields 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 (i.e. a security requirements), cost (i.e. a general limitation) and usability (i.e. a quality requirement), the author states, “To obtain higher security . . . will cost more. To increase security often impacts upon usability, and visa versa” [26]. Another brief example is a company’s use of three security patterns to ensure the integrity of messages passed between it and its business partners. From a security perspective, this is ideal (the more security the better), but from a performance perspective, it is unlikely to be accepted because excessive security will undoubtedly negatively affect processing time. Looking at the security objectives method, even though businesses may desire to have every message secured to the highest priority level in terms of confidentiality and integrity, financially, this may simply not be realistic noting cost of certificates, software and so on. These are the types of factors to be assessed in this step.

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 14 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) to another (as was likely done in the Architectural phase), 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. To give an example of possible output, a snippet of a UML process model with patterns applied is included in Figure 15.

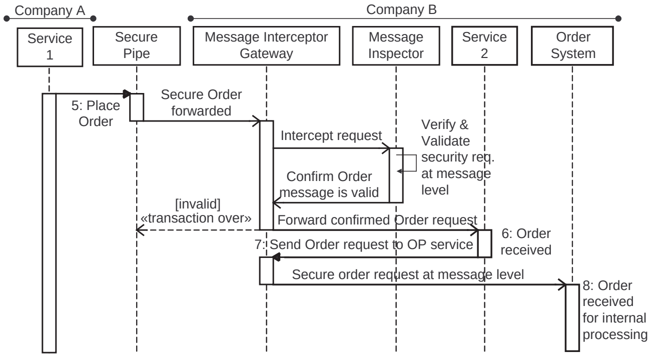
Company A
Company B
Service 1
Secure Pipe
Message Interceptor Gateway
Message Inspector
Service 2
Order System
5: Place Order
Secure Order forwarded
Intercept request
Verify & Validate security req. at message level
Confirm Order message is valid
[invalid] «transaction over»
Forward confirmed Order request
6: Order received
7: Send Order request to OP service
Secure order request at message level
8: Order received for internal processing

Figure 15. UML process model with patterns applied

In this UML sequence diagram, three security patterns have been applied to protect order-related communications/messages between systems at Company A and Company B. These are (i) secure pipe—for securing a basic connection between trading parties; (ii) message interceptor gateway—a central location to manage security enforcement tasks; and last (iii) message inspector—this is responsible for the verification and validation of the security elements in the data or message delivered. These patterns were sourced from [10], and more examples of their use and application can be gathered from that reference.

Due to its versatility and extensibility, UML again forms one of the better techniques for the modelling task. In Figure 7, it was shown that for simple modelling, sequence or activity diagrams are useful; an example of which is seen above. To facilitate detailed modelling, one suggested option is the UML profile for security, quality and fault tolerance requirements. This profile is defined in [48] and provides a standard mechanism for expressing security. Another noteworthy option still within the structured confines of UML can be found in [49]. 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 [63] and SecureUML [64] are two additional, more detailed security-related extensions to UML that might also be of interest to businesses.

With regards to BPMN, the inclusion of security aspects in models is much less researched and standardized when compared to advances in UML. Authors in [50] state that BPMN “does not explicitly consider mechanisms to represent the security requirements". Because of this fact, BOF4WSS primarily suggests annotations to models and detailed supporting documentation when engaged in simple modelling. Detailed modelling as highlighted in Figure 7 can be partially addressed by recent research work—see [50]. In that article, researchers recognize the need to have the capability to include security in models, and therefore develop an extension to BPMN for modelling security requirements in business process. Albeit not fully complete, this work provides an invaluable start for companies in moving from detailed notes to formal, standards-based models. The caveat to adopting this approach however is its newness. This means that there may be changes or updates in future security modelling notations (thus the possible need to reconstruct process models), and also that it lacks tool support (the need for tools to streamline and ease process modelling should not be overlooked). For all of the other modelling techniques, where no special accommodation is made for modelling security actions or requirements, the framework advocates annotating the models and making detailed supporting documentation. This is not ideal because requirements are not directly and practically applied to models however, it should provide teams with enough relevant information to proceed.

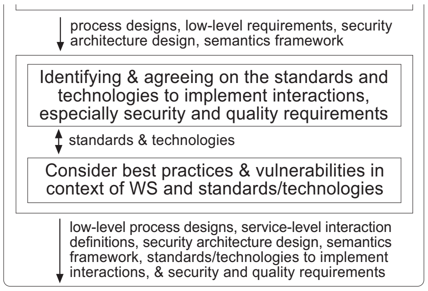
process designs, low-level requirements, security architecture design, semantics framework
Identifying & agreeing on the standards and technologies to implement interactions, especially security and quality requirements
standards & technologies
Consider best practices & vulnerabilities in context of WS and standards/technologies
low-level process designs, service-level interaction definitions, security architecture design, semantics framework, standards/technologies to implement interactions, & security and quality requirements

Figure 16. WS standards agreement and assessment tasks in the Systems Design phase

The penultimate task in the Design phase depicted by Figure 16, is identifying and agreeing on the standards that will be used to implement the services, and especially the security and quality-of-service (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. Authors in [14] allude to this fact as they discuss the "What's missing" in Web services technology. The main interoperability problems they identified stem from the existence of too many standards (over sixty already), the tweaking of standards by individual companies, and the numerous versions of even the basic WS standards [14]. Authors accept that WS-Interoperability (WS-I) [65] profiles can address some of these problems, however they note that this is only possible if companies make their Web services compatible to 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 technologies should take the lead at this point. Whereas 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 provides companies with key information sources which they can reference. Sources range from published texts [4], [10], [19], [66] for introductory- and intermediate-level material, to the actual standards Web sites i.e. W3C, OASIS, Liberty Alliance Project and WS-I, for up-to-date, definitive information.

To identify security standards, the work of [10] is particularly relevant if companies have used their security pattern catalogue in previous stages in the framework. The reason for this is because within their catalogue, also supplied is a list of standards and technologies that can implement the respective patterns. For example, to implement the message inspector pattern mentioned above and shown in Figure 15, [10] suggests options of XML Encryption, XML Signature, SAML and XKMS to name a few. Information on these and other security standards and technologies can be found in NIST guidelines such as [5]. One of this article's core purposes is to provide companies with "practical, real-world guidance on current and emerging standards applicable to Web services" [5]. 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 [67]. This covers a number of WS QoS aspects, mentions standards which are used to implement them, and also discusses techniques to improve Web service quality.

A final short 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. If there is an absence of tools, regardless of the benefits of standards proposed, these standards cannot be used. Common reasons for little, or no tool support include newly developed/ratified standards (i.e. they lack maturity), and rejection of standards from key tool provider companies such as Microsoft Corporation (with .NET), or Sun Microsystems Inc. (with J2EE).

Having agreed to some degree on the standards and technologies to be employed, BOF4WSS (see Figure 16) advises companies to consider (i) the common vulnerabilities and pitfalls in WS and the mechanisms chosen, and (ii) the best practices in using the WS standards/tools, and implementing them as securely as possible. Both of these factors may have been analyzed 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 i.e. common vulnerabilities and pitfalls in WS, two prime sources are documents from organizations such as NIST (see [5]) and WS-I (see [68]). These give information on common attacks, risks, and typical security challenges.

For the second task in Figure 16, 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. [5] provides general guidance in addressing threats and on secure implementation tools and technologies. [10] gives various best practices and design strategies. [68] identifies typical countermeasures (technologies and protocols) to mitigate common WS threats. Finally, [69] 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 16, and is highlighted because depending on vulnerabilities or best practices, teams may often opt to use different, more robust standards, or technologies with extensive guidance (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)*.

3.8. Agreements (for QoS) Phase

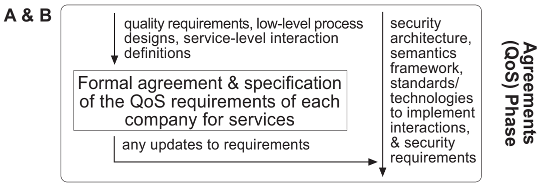
A & B
quality requirements, low-level process designs, service-level interaction definitions
Formal agreement & specification of the QoS requirements of each company for services
any updates to requirements
security architecture, semantics framework, standards/ technologies to implement interactions, & security requirements
Agreements (QoS) Phase

Figure 17. Workflow model of the Agreements (for QoS) phase

With the low-level process designs, and service-level interactions defined, the **Agreements phase** concentrates on the agreements necessary at the quality-of-service (QoS) level. During the task shown in Figure 17, the goal is to specify the mutual understanding of the priorities, responsibilities, and guarantees expected by each business with respect to the other entity, regarding the actual Web services. This phase directly extends work on quality requirements in the Design phase, and in the end, results in a set of formal and contractual agreements. As done before, QoS requirements typically assessed include service availability needs (e.g. a service uptime of 99.98%), performance requirements (e.g. 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 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, companies can reference [4].

Another option proposed by the research community in [70] is to make use of accepted policy standards such as WS-Policy to specify a service's quality requirements. 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 in [71]. Broadly, this framework allows for the specification and monitoring of SLAs for WS. It enables service users and providers (i.e. companies in BOF4WSS context) to define a variety of SLAs, specify the SLA parameters (e.g. availability, 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—see [72]. 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)*.

3.9. 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 18)

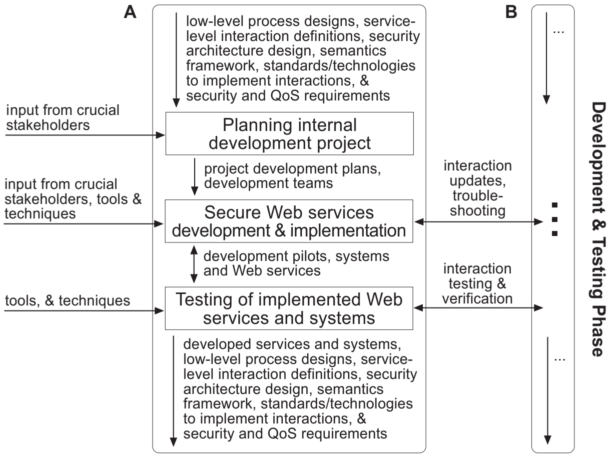
A
B
low-level process designs, service-level interaction definitions, security architecture design, semantics framework, standards/technologies to implement interactions, & security and QoS requirements
input from crucial stakeholders
Planning internal development project
project development plans, development teams
input from crucial stakeholders, tools & techniques
Secure Web services development & implementation
interaction updates, trouble-shooting
development pilots, systems and Web services
tools, & techniques
Testing of implemented Web services and systems
interaction testing & verification
developed services and systems, low-level process designs, service-level interaction definitions, security architecture design, semantics framework, standards/technologies to implement interactions, & security and QoS requirements
Development & Testing Phase
...
...

Figure 18. Workflow model of the Development and Testing phase

is centered on the actual development, implementation, deployment and testing of services and systems at the companies. Because of this factor, it is mainly carried out by companies individually, with each company working on their own systems development. 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 (i.e. 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 paper'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. As a result, this phase is much less strictly prescribed, with Figure 18 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 pointer 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 are instrumental in aiding in this internal process are [4], [9].

In the former work, [4] 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 for SOA-based deployment. 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, the framework additionally suggests the integration of PWSSec [9]—a detailed development process for secure Web services. The novelty behind this process 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 [73]. This text provides some useful guidelines that can be applied within the planning task, related to the 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. If we consider the Analysis phase in [4] 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 [4]) 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 on WS-CDL (and BPEL4Chor), these process definitions can be converted to the BPEL advocated in [4]. For the more security-specific PWSSec [9], the medium- and low-level security requirements, and security patterns identified from BOF4WSS can be reused in PWSSec's Requirements and Architectural stages. [9] also uses UML and the profile for security ([48]) for some of its modelling; one would recall that 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 (e.g. UML, BPMN). To recap, BPEL allows for the specification of business process behaviour based on Web services. Amongst other things, it is an execution language which can be run by software engines to orchestrate message, control and data flows. If companies have modelled process in UML or BPMN therefore, techniques such as [45], [74], [75] that offer some aid in translating these models to executable processes (in BPEL) are quite ideal. Specifically, [74] works with the translation of BPMN models to BPEL definitions, whereas [45], [75] aim at transforming their UML variants and extensions (which may be have been used by companies) to their respective BPEL process 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 (e.g. service policy specifications, federated security), this task is still far from trivial.

Researching security configurations for WS, [76] highlights 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, and 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 [4]. Companies that use that methodology therefore can receive more information on it there. Regarding security monitoring, the key is to install softwares to maintain adequate logs, audit trails and records that can be referred to as required. Authors in [10] highlight that having these audit trails has even become a requirement of some laws e.g. SOX. Intrusion detection, or prevention software may also be of interest to businesses. Fortunately, some of the softwares mentioned are implementations of typical, higher-level security patterns, and therefore are very likely to be included in developed systems.

The final task within this phase is the testing of the developed Web services 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* (i.e. both *internally*, and *externally*, across companies). Testing can occur from three main perspectives; functional (do Web services do what they should), quality (are the set performance, usability, scalability, etc. requirements met), and security (is there adequate protection in place for Web services and systems). Guidance on testing the functional and quality requirements is given in lifecycle methodology [4] 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 [26]. Conversely, this however does not mean that testing is impossible, or 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 i.e. identify requirements (these may be in terms of needs, 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, [77] offers a number of strategies and guidelines. These are both generic (i.e. just highlight the use of test suits, test patterns and so on), and targeted (i.e. 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 [78] regarding various guidelines on software vulnerability analysis for WS. These include checking for cross-site scripting, services traversal, DoS attacks, and access validation attacks. Actually, businesses can 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; this is likely because WS testing is a discipline still in its infancy [77]. One tool that has surfaced (and been referenced in academia [69] for its use) however, is wsChess [79]. WsChess is described by its makers as a freely available toolkit for WS assessments and defense [79]. Authors in [69] give a brief example of how wsChess can be used to probe for vulnerabilities and formulate attacks against Web services. To assess Web applications that may constitute part of the WS systems, a number of tools are available. [80] is a perfect 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 analyzers, Web application scanners, runtime analysis tools, and configuration management tools, to assist companies with their tool selection [80]. The tools and techniques supplied here and those available from other sources should be used wherever possible to enable for a thorough, adequate testing of the developed Web services systems. This testing activity completes the Development and Testing phase; the outputs of this phase 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).*

3.10. Maintenance Phase

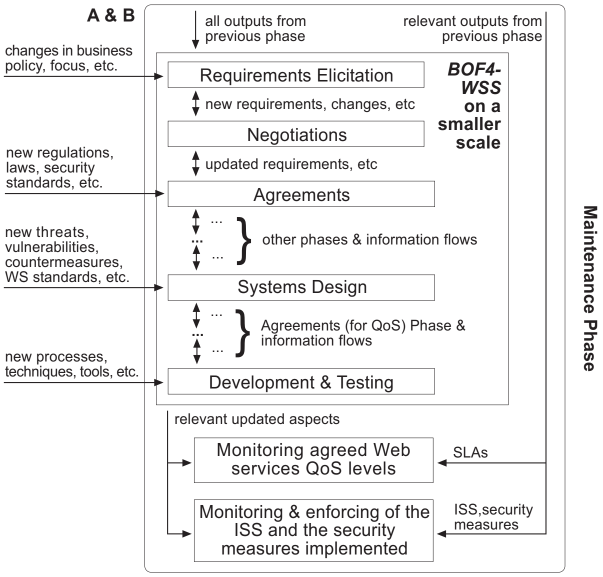
A & B
all outputs from previous phase
relevant outputs from previous phase
changes in business policy, focus, etc.
Requirements Elicitation
BOF4-WSS on a smaller scale
new requirements, changes, etc
Negotiations
new regulations, laws, security standards, etc.
updated requirements, etc
Agreements
new threats, vulnerabilities, countermeasures, WS standards, etc.
...
other phases & information flows
Systems Design
...
Agreements (for QoS) Phase & information flows
new processes, techniques, tools, etc.
Development & Testing
relevant updated aspects
Monitoring agreed Web services QoS levels
SLAs
Monitoring & enforcing of the ISS and the security measures implemented
ISS,security measures
Maintenance Phase

Figure 19. 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 19. It is important to understand 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 full BOF4WSS process. The advantage of this is the experience they bring and the avoidance of having to deal with too steep of a learning curve. One team already mentioned in BOF4WSS is the security team from the first agreements stage. These personnel are entrusted with the responsibility of monitoring the *internal* and *external* environments, and considering new threats, laws, and security requirements, and how these will be included in system and WS interactions 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 true even when these updates are agreed by both companies involved. Changes should not be made in isolation without first analyzing 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 19). By reiterating this process for new needs in the form of updates and changes, it allows modifications to be made in a structured and controlled context. Repetition of previous phases is not uncommon during software maintenance as noted in [23]. Because the BOF4WSS process has been discussed in detail previously, it is not covered here or in Figure 19. Instead, Figure 19 is used to display some of the key **new** inputs (i.e. in addition to the ones outlined in previous phases) which are very likely to be incurred. Examples are, changes in the business policies (reflecting possibly new goals, aims), new regulations (therefore new, mandatory security 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 19 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 Web service performance levels such as service response times, and system downtime and latency. SLAs also are 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. Authors in [37] when they describe information security 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 (e.g. audit trails, logs) from monitoring and detection softwares 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. This relates to 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 (by a third party possibly), and on-site visits.

A final noteworthy aspect shown in Figure 19, 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.

3.11. Summarizing BOF4WSS

As can be seen from the preceding in-depth discussion of BOF4WSS, the framework provides a detailed guidance model for inter-organizational cooperation. Beyond this, the next aim in this research (discussed in Section 4) is to drill down into the framework's specifics and provide a practical implementation base. This includes investigation into how stages of the architecture can be expanded, when or where can existing mechanisms be used, and lastly in the provision of suitable infrastructure and tool support to aid in framework use.

Reflecting on BOF4WSS in its entirety, specially with regard to its use by companies, it is obvious that this is not a process to be taken flippantly. 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 (e.g. [17]), and how security—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*. 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. At that point, 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 however, existing companies may choose to more focus on initial phases of BOF4WSS i.e. identifying risks and negotiating on security actions, and then ensuring that companies share the same goals with regards to cross-enterprise security. The ISS would be very relevant in this regard.

To utilize this approach, companies will have to be prepared to work together and devote resources—financial and nonfinancial (e.g. time, skills, experience)—to this venture. Many changes in how the businesses worked before WS adoption will be necessary. However as stated in [66] concerning WS in general, "the potential benefits—both financial and strategic—to adopting Web services are sufficiently large to justify such [business] changes." The same fact is true when focusing on security specifically.

Another crucial factor supporting the highly involved approach to security central to BOF4WSS, is the emerging legislative requirement-base. These regulations (partially shown in [10]) demand that companies now look both *internal* and *external* (i.e. business relationships) in their considerations of security. In [6], authors commenting on the new security responsibilities in WS, state that "risks must be assessed and managed across a collection of organizations, which is a new and very challenging security responsibility". They also make the point that to ensure collective WS offerings between businesses are secure, elements such as strategies and structured approaches to security must be used [6]. All these requirements fuel the need for a security approach such as BOF4WSS.

3.12. Limitations

There are two known, noteworthy limitations of the framework. 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 gives e-businesses detailed guidance and insight into online WS interactions, it ties the framework too closely with current practices. This 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 [25]—currently up to version 9. Furthermore, considering the volatility of the online security field, the updating of all security-focused models is vital.

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 above), there are reservations about the time taken for overall project completion, and flexibility and turnaround time within individual phases. Possible ways to address these issues include attempting to incorporate quicker and more flexible development techniques within specific phases of the WM-based framework. Additional benefits with these 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. [81] 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 however, to ensure that structure and benefits of the WM to large or critical system projects are not affected.

4. Conclusion and Future Work

In this paper, we extended the work in [1] by engaging in a detailed discussion of our cross-enterprise development methodology, BOF4WSS. This discussion included a step-by-step analysis of its nine phases, where we presented the activities involved, justified the guidance given, and highlighted how the activities proposed could aid in building the requisite security and trust across collaborating e-businesses. Throughout this work, our main contention was that because of the very nature of Web services technology, the security of interacting e-businesses was now a much broader, and more 'real-time' issue than ever before. The broad issue was as a result of the ease in which threats and attacks by way of WS, could propagate from poorly secured companies to the systems of their unsuspecting business partners. Whereas the 'real-time' issue refers to the speed in which attacks can spread between interacting companies.

The novelty of our approach is that it considers the full nature of WS, and its security implications (technical and otherwise); recognizes and targets the 'live' inter-organizational security issue now faced by interacting e-businesses; and finally, promotes the use of a joint approach, where businesses work closely together and follow a structured process, to achieve enhanced levels of security and trust across partners. Our approach therefore aims to be a facilitator of, instead of a panacea to the security of e-businesses which use WS. Similarly, the goal is to provide another important piece of the security puzzle that is complementary to existing approaches.

In future work, the first area of interest is the provisioning of systems support for the framework itself. As can be seen, BOF4WSS is a quite extensive process. To aid in its use therefore, we intend to further examine each stage and the interface between stages, and provide support wherever applicable. One area already identified (through an initial exploratory investigation), concerns the outputs the *individual* Requirements Elicitation phase and their immediate usefulness as inputs to the *joint* Negotiations phase. This is of interest because of the inherent difficulty (relating to security actions/requirements format, prioritization schemes, and so on) in attempting to quickly and easily compare, and negotiate on the high-level security actions/requirements of each business as they are passed from the Requirements Elicitation, to Negotiations phases. To address this issue we are currently investigating into a tool-based approach to streamline security actions/requirements comparison and negotiations. The first steps in our work can be seen in [82], [83].

Once the framework with added systems support is complete, our next goal will be its evaluation to determine how well BOF4WSS's aims of enhancing security and trust across businesses, have been achieved. Noting the complexity and scope of the framework, the evaluation process necessary is far from trivial. In an attempt at a thorough evaluation which appreciates these difficulties, a three-stage process is planned.

First, industry-based security professionals would be interviewed to get their views on the suitability, and application of the framework. This stage provides useful and quick feedback from a variety of experienced and expert sources, on the framework and the activities it proposes. The next stage adopts a more practical perspective and focuses an evaluation of the systems support developed, to ascertain the actual degree of support supplied to the framework; specific scenarios are envisioned for use, which test numerous aspects of tool support. Initial work in this area can be seen in [84].

The last stage would be the full application of BOF4WSS to a real-life case scenario to critically evaluate its suitability and strength in achieving its goals. This would involve engaging a small set of companies to use the framework in their business scenario, then monitoring them, and constructing a case study from the observations, difficulties, uses and so on. The case could then be analyzed in-depth to make inferences on the applicability and effectiveness of the framework in real-world scenarios. This three-stage process would enable key aspects of BOF4WSS to be evaluated and substantiated conclusions made on its proposals.

References

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

[2] J. Zhang, "Trustworthy web services: Actions for now," *IT Professional*, vol. 7, no. 1, pp. 32–36, 2005.

[3] M. Chen, "An analysis of the driving forces for web services adoption," *Information Systems and e-Business Management*, vol. 3, no. 3, pp. 265–279, 2005.

[4] M. P. Papazoglou, *Web Services: Principles and Technology*. Harlow, Essex: Prentice Hall, 2007.

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

[6] B. Hartman, D. J. Flinn, K. Beznosov, and S. Kawamoto, *Mastering Web Services Security*. Indianapolis: Wiley, 2003.

[7] R. J. Boncella, "Web services and web services security," *Communications of the Association for Information Systems*, vol. 14, no. 18, pp. 344–363, 2004.

[8] A. Charfi and M. Mezini, "Using aspects for security engineering of web service compositions," in *IEEE International Conference on Web Services*, Orlando, 2005, pp. 59–66.

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

[10] 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.

[11] 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*, vol. 10, no. 1, pp. 47–55, 2007.

[12] 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*, vol. 18, no. 2, pp. 115–145, 2005.

[13] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*. Indianapolis: Wiley, 2004.

[14] S. Fischer and C. Werner, "Towards service-oriented architectures," in *Semantic Web Services: Concepts, Technologies, and Applications*, R. Studer, S. Grimm, and A. Abecker, Eds. Berlin: Springer-Verlag, 2007, pp. 15–24.

[15] T. Ishaya and J. R. Nurse, "Cross-enterprise policy model for e-business web services security," in *Information Security and Digital Forensics*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, D. Weerasinghe, Ed. Heidelberg: Springer, 2010, vol. 41, pp. 163–171.

[16] 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.

[17] E. W. Davis and R. E. Spekman, *The Extended Enterprise: Gaining Competitive Advantage through Collaborative Supply Chains*. Upper Saddle River, NJ: FT Prentice Hall, 2004.

[18] C. Van Slyke and F. Bélanger, *E-Business Technologies: Supporting the Net-Enhanced Organization*. New York: Wiley, 2003.

[19] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Berlin: Springer-Verlag, 2004.

[20] M. Khalifa and J. M. Verner, "Drivers for software development method usage," *IEEE Transactions on Engineering Management*, vol. 47, no. 3, pp. 360–369, 2000.

[21] H.-J. Bullinger, K.-P. Fähnrich, and T. Meiren, "Service engineering—methodical development of new service products," *International Journal of Production Economics*, vol. 85, no. 3, pp. 275–287, 2003.

[22] S. Chatterjee, "The waterfall that won't go away," *SIGSOFT Softw. Eng. Notes*, vol. 35, no. 1, pp. 9–10, 2010.

[23] I. Sommerville, *Software Engineering*, 8th ed. Essex: Pearson Education Ltd., 2007.

[24] H. Van Vliet, *Software Engineering: Principles and Practice*, 3rd ed. Chichester: John Wiley & Sons Ltd., 2008.

[25] The Open Group, "TOGAF^{TM}Version 9," 2009, http://www.opengroup.org/togaf/ (Accessed 8 February 2010).

[26] J. Sherwood, A. Clark, and D. Lynas, *Enterprise Security Architecture: A Business-Driven Approach*. San Francisco, CA: CMP Books, 2005.

[27] O. Demirörs, Ç. Gencel, and A. Tarhan, "Utilizing business process models for requirements elicitation," in *The 29th Conference on EUROMICRO*. IEEE, 2003, pp. 409–412.

[28] F. Hartman and R. A. Ashrafi, "Project management in the information systems and information technologies industries," *Project Management Journal*, vol. 33, no. 3, pp. 5–15, 2002.

[29] M. P. Papazoglou and P. Ribbers, *e-Business: Organizational and Technical Foundations*. Chichester, West Sussex: John Wiley & Sons Ltd., 2006.

[30] 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 8 February 2010).

[31] R. S. Coles and R. Moulton, "Operationalizing IT risk management," *Computers & Security*, vol. 22, no. 6, pp. 487–493, 2003.

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

[33] C. Alberts and A. Dorofee, *Managing Information Security Risks : The OCTAVE Approach*. Boston: Addison-Wesley, 2003.

[34] 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*, vol. 25, no. 1, pp. 101–117, 2007.

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

[36] J. Misrahi, "Validating your business partners," in *Information Security Management Handbook*, 6th ed., H. F. Tipton and M. Krause, Eds. Boca Raton, FL: Auerbach Publications, 2007, vol. 1, pp. 123–131.

[37] A. Calder and S. Watkins, *IT Governance: A Manager's Guide to Data Security and ISO 27001/ISO 27002*, 4th ed. London: Kogan Page Limited, 2008.

[38] K. C. Laudon and C. G. Traver, *E-commerce: Business, Technology, Society*, 3rd ed. New Jersey: Prentice Hall, 2007.

[39] W. H. Baker, G. E. Smith, and K. J. Watson, "Information security risk in the e-supply chain," in *E-Supply Chain Technologies and Management*, Q. Zhang, Ed. Hershey, PA: Idea Group Inc., 2007, pp. 142–161.

[40] S. Röhrig and K. Knorr, "Security analysis of electronic business processes," *Electronic Commerce Research*, vol. 4, no. 1-2, pp. 59–81, 2004.

[41] S. Nachtigal, "eBPSM: A new security paradigm for e-business organisations (e-business process security model)," in *The Ninth International Conference on Electronic commerce*, Minneapolis, MN, 2007, pp. 101–106.

[42] G. M. Giaglis, "A taxonomy of business process modeling and information systems modeling techniques," *International Journal of Flexible Manufacturing Systems*, vol. 13, no. 2, pp. 209–228, 2001.

[43] R. S. Aguilar-Savén, "Business process modelling: Review and framework," *International Journal of Production Economics*, vol. 90, no. 2, pp. 129–149, July 2004.

[44] P. Mayer, A. Schroeder, and N. Koch, "A model-driven approach to service orchestration," in *IEEE International Conference on Services Computing*. Honolulu, Hawaii: IEEE Computer Society, 2008, pp. 533–536.

[45] D. Skogan, R. Groenmo, and I. Solheim, "Web service composition in UML," in *Eighth IEEE International Enterprise Distributed Object Computing Conference*, 2004, pp. 47–57.

[46] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River, NJ: Pearson Education, 2005.

[47] J. Recker, "Process modeling in the 21st century (BPTrends columns & articles)," May 2006, http://www.bptrends.com/publicationfiles/05-06-ART-ProcessModeling21stCent-Recker1.pdf (Accessed 8 February 2010).

[48] 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 8 February 2010).

[49] 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*, 2006, pp. 670–677.

[50] ——, "A BPMN extension for the modeling of security requirements in business processes," *IEICE - Transactions on Information and Systems*, vol. E90-D, no. 4, pp. 745–752, 2007.

[51] M. Schumacher and U. Roedig, "Security engineering with patterns," in *The 8th Conference on Pattern Languages of Programs (PLoP)*, Monticello, Illinois, 2001.

[52] World Wide Web Consortium (W3C), "Web services choreography description language version 1.0," 2005, http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/ (Accessed 8 February 2010).

[53] G. Decker, O. Kopp, F. Leymann, and M. Weske, "BPEL4Chor: Extending BPEL for modeling choreographies," in *IEEE International Conference on Web Services*. Los Alamitos, CA, USA: IEEE Computer Society, 2007, pp. 296–303.

[54] G. Decker, O. Kopp, F. Leymann, K. Pfitzner, and M. Weske, "Modeling service choreographies using BPMN and BPEL4Chor," in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, Z. Bellahsène and M. Léonard, Eds. Heidelberg: Springer, 2008, vol. 5074, pp. 79–93.

[55] A. Barker and J. van Hemert, "Scientific workflow: A survey and research directions," in *Parallel Processing and Applied Mathematics*, ser. Lecture Notes in Computer Science, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, Eds. Heidelberg: Springer, 2008, vol. 4967, pp. 746–753.

[56] J. Mendling and M. Hafner, "From WS-CDL choreography to BPEL process orchestration," *Journal of Enterprise Information Management*, vol. 21, no. 5, pp. 525–542, 2008.

[57] 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 8 February 2010).

[58] 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*, 2006, pp. 131–139.

[59] Anonymous, "WS-CDL Eclipse," n.d., http://wscdl-eclipse.sourceforge.net/main.htm (Accessed 8 February 2010).

[60] Pi4 Technologies Foundation, "Pi4SOA," n.d., http://pi4soa.wiki.sourceforge.net/ (Accessed 19 December 2008).

[61] H. Foster, "WS-Engineer 2008: A service architecture, behaviour and deployment verification platform," in *Service-Oriented Computing ICSOC 2008*, ser. Lecture Notes in Computer Science, A. Bouguettaya, I. Krueger, and T. Margaria, Eds. Heidelberg: Springer, 2008, vol. 5364, pp. 728–729.

[62] 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.

[63] J. Jürjens, *Secure Systems Development with UML*. Berlin: Springer-Verlag, 2005.

[64] T. Lodderstedt, D. A. Basin, and J. Doser, "SecureUML: A UML-based modeling language for model-driven security," in *UML 2002: The Unified Modeling Language*, ser. Lecture Notes in Computer Science, J.-M. Jézéquel, H. Hussmann, and S. Cook, Eds. Heidelberg: Springer, 2002, vol. 2460, pp. 426–441.

[65] Web Services Interoperability Organization (WS-I), "WS-I," n.d., http://www.ws-i.org/ (Accessed 8 February 2010).

[66] S. Chatterjee and J. Webber, *Developing Enterprise Web Services: An Architect's Guide*. Upper Saddle River, NJ: Prentice Hall PTR, 2004.

[67] W. D. Yu, R. B. Radhakrishna, S. Pingali, and V. Kolluri, "Modeling the measurements of QoS requirements in web service systems," vol. 83, no. 1, 2007, pp. 75–91.

[68] 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 8 February 2010).

[69] N. Sidharth and J. Liu, "IAPF: A framework for enhancing web services security," in *31st Annual International Computer Software and Applications Conference (COMPSAC)*, Beijing, China, 2007, pp. 23–30.

[70] 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.

[71] A. Keller and H. Ludwig, "The WSLA framework: Specifying and monitoring service level agreements for web services," *Journal of Network and Systems Management*, vol. 11, no. 1, pp. 57–81, 2003.

[72] International Business Machines (IBM) Corp., "Emerging Technologies Toolkit (ETTK) for Web Services," n.d., http://www.alphaworks.ibm.com/tech/ettk (Accessed 8 February 2010).

[73] O. Zimmermann, M. R. Tomlinson, and S. Peuser, *Perspectives on Web Services: Applying SOAP, WSDL, and UDDI to Real-World Projects*. Berlin: Springer, 2003.

[74] 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*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 285–292.

[75] 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.

[76] 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*. Athens, Greece: IEEE Computer Society, 2004, pp. 244–251.

[77] A. Barbir, C. Hobbs, E. Bertino, F. Hirsch, and L. Martino, "Challenges of testing web services and security in SOA implementations," in *Test and Analysis of Web Services*, L. Baresi and E. Di Nitto, Eds. Heidelberg: Springer, 2007, pp. 395–440.

[78] W. D. Yu, D. Aravind, and P. Supthaweesuk, "Software vulnerability analysis for web services software systems," in *IEEE Symposium on Computers and Communications*. IEEE, 2006, pp. 740–748.

[79] Net-Square Solutions Pvt. Ltd., "wsChess - web services assessment and defense toolkit," n.d., http://net-square.com/wschess/index.shtml (Accessed 8 February 2010).

[80] M. Curphey and R. Arawo, "Web application security assessment tools," *IEEE Security & Privacy*, vol. 4, no. 4, pp. 32–41, 2006.

[81] 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 8 February 2010).

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

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

[84] ——, "Evaluating the Compatibility of a Tool to Support E-Businesses' Security Negotiations (to appear)," in *The International Conference of Information Security and Internet Engineering (ICISIE 2010), part of World Congress on Engineering (WCE) 2010*, London, UK, 2010.