Specification of Security Requirements for Business Collaborations

Bart Orriens

January 18, 2006

Abstract

Service-oriented computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing business collaborations. In order to realize this vision security is a critical issue that must be addressed. Current work in this area is low level in nature without any connection to high level security requirements. In this report we present a layered specification of security properties for business collaboration. The approach supports specification of high level security objectives, the mechanisms to achieve those, and the measures needed to implement these mechanisms. Moreover, dependencies among objectives, mechanisms and measures are made explicit as such creating a traceable path from objectives to measures.

Infolab Technical Report Series, no. 25

January 2006

1 Introduction

Recently there has been increasing focus on service-oriented computing (SOC), the new emerging paradigm for distributed computing and e-business processing, to deliver flexible and adaptable corporate business services by utilizing existing services across organizational boundaries. *Business collaboration* refers to a cooperation between multiple enterprizes working together to achieve a common business goal. In order to realize the vision of utilizing services as fundamental elements for developing applications [18] for business collaboration, security is a critical issue that must be addressed. Businesses will be averse to participating in cooperations that do not take place in a trusted environment, for example to avoid problems concerning denial of actions, unauthorized reading of information, and so on.

Therefore, for the successful adoption of SOC within the business collaboration domain, the paradigm must provide the means to make cooperation between enterprizes secure. At the moment, the most successful manifestation of SOC can be found in web services technology. A web service is a specific kind of service that can be unambiguously identified (generally by means of a URI) and whose service description and transport utilize open Internet standards, such as XML-based SOAP messages. Unfortunately, most work in the web service security arena concentrates solely on low-level security provisions; leaving the issue of relating these to higher level business requirements unaddressed. In this report we briefly introduce our business collaboration context framework, which provides the context required for business collaboration development and management. We also show how this context can be described using collaboration models. Subsequently, we explain how the introduced collaboration models can be augmented to support the specification of security requirements from high level goals to low-level security measures such as provided by current web service security solutions.

The remainder of this report is structured as followed: we first introduce a running example based on a complex insurance claim handling scenario in section 2. Next, in section 3 we briefly discuss our framework for business collaboration context; after which we explain our model driven approach for the definition of this context in section 4. After that, in section 5 we analyze the role of security in business collaboration, and show how the specification of security requirements can be facilitated. Finally, we present conclusions in 6 and outline future work.

2 Example

To exemplify the ideas presented throughout this paper an example inspired by the case study in [10] is used. The example describes a complex multi-party scenario, which outlines the manner in which a car damage claim is handled by an insurance company (AGFIL). AGFIL cooperates with several contract parties to provide a service level that enables efficient claim settlement. The parties involved are Europ Assist, Lee Consulting Services, Garages and Assessors. Europ Assist offers a 24-hour emergency call answering service to policyholders. Lee C.S. coordinates and manages the operation of the emergency service on a day-to-day level on behalf of AGFIL. Garages are responsible for car repair. Assessors conduct the physical inspections of damaged vehicles and agree repair upon figures with the garages. The scenario outline is as followed (more details are introduced in the remainder of this paper where needed):

The policyholder (customer) phones Europ Assist using a free-phone number to notify a new claim. The claim is received by a call handler within Europ Assist's telephone assistance department. After verification of the customer' credentials to ensure that the provided policy details are valid and the occurred loss is covered, the call handler finds an approved repairer nearest to the customer's location. The customer is notified that this repairer will arrive at the scene shortly, if necessary with a replacement car and towing service. The call handler subsequently contacts the selected repairer to notify him of the incident. If the repairer is not available, another one will be selected and contacted. The customer is kept posted of such changes by phone. Once the repairer is on its way, the call handler contacts AGFIL to inform them of the made claim.

Upon receipt of the claim a claim handler will be assigned within AGFIL. The claim handler will gather all related claim information like customer records, claim history, etc. to Lee C.S. After that the claim handler will fill out the claim details on a claim form, which is subsequently stored pending further developments. Lee C.S. in the meanwhile has one of its consultants working on the claim. The first thing this consultant does, is contact the garage to inquire about the status of the car. The garage has picked up the car while the previous was going on and has worked out an estimate of the car repair cost. If this cost was below $500 then the garage will have started repairs. But if the costs were higher, the consultant at Lee C.S. contacts an assessor to go to the garage and check out the car for him -or herself. This assessor makes an independent estimate of the repair costs and negotiates a final price with the garage.

The result of the assessment is next reported back to the consultant at

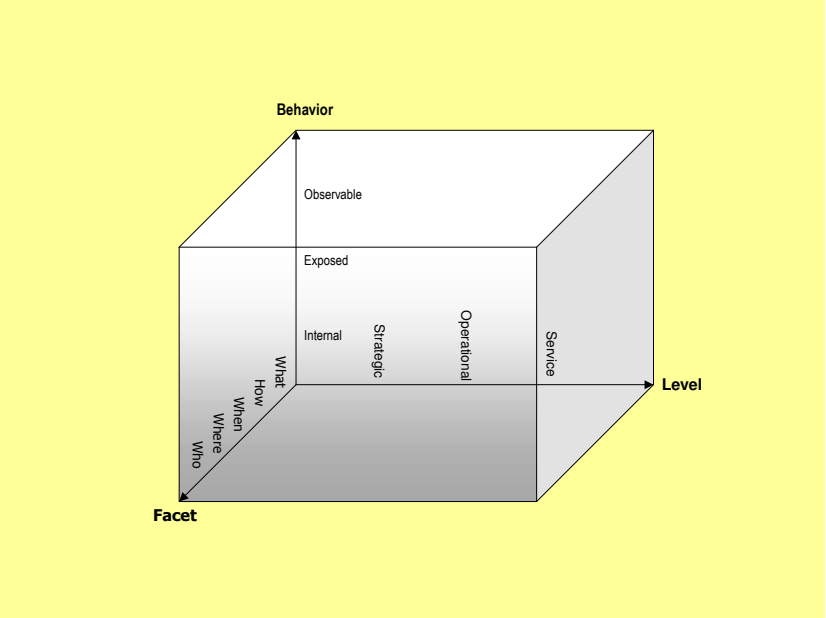

Fig. 1: Business Collaboration Context Framework (BCCF)

Lee C.S. The consultant reads the report and approves repair. An approval notification is sent to the garage, which consequently starts repairs on the car. Lee C.S' consultant also informs the claim handler at AGFIL of the final repair cost estimate upon which the claim handler incorporates the new information in the claim form. Once the garage has completed its repairs on the customer's car, an invoice is communicated to the consultant at Lee C.S. The consultant checks the invoice to see if it matches the earlier received cost estimate. Once the invoice is approved, the consultant sends the invoice onwards to AGFIL. The claim handler receives the invoice and adds it to the claim form. Payment for the claim is also issued.

3 Business Collaboration Context Framework

At the heart of our approach stands the Business Collaboration Context Framework (BCCF). The BCCF captures the context in which business collaboration development and management takes place by adopting a three dimensional view. Through this three dimensional view modularization of the definition and management of business collaborations is achieved. An overview of the framework is shown in Fig. 1.

As the figure illustrates we modularize the business collaboration context along three dimensions in the BCCF, being *behavior*, *level* and *facet*. We briefly discuss these in the following. For more information the reader is referred to [15, 16].

3.1 Behavior

The first dimension, **behavior**, places emphasis on the different behaviors that an enterprize exhibits in business collaboration; where consequently the purpose and target of development and management varies. The behavior dimension encompasses three types of behavior captured in three corresponding so-called collaboration aspects (inspired by among others [9, 19, 22]): observable, exposed and internal behavior expressed in the *conversation*, *participant public behavior* and *internal business process* aspect respectively.

The observable behavior constitutes the externally visible behavior between participants in a business collaboration; and is expressed in the *conversation aspect*. Captured in the *participant public behavior aspect* the exposed behavior describes how an individual participant can publicly behave in a business collaboration (i.e. its potential collaboration behavior). In contrast, the internal behavior (specified in the *internal business process aspect*) is also individual to each participant; however, it is only of interest to this particular participant, i.e. it can not be observed by other participants.

3.2 Level

The second dimension, **level**, recognizes the fact that the different business collaboration behaviors of an enterprize take place at several levels; where consequently the domain, degree of abstraction and the type of developers in development and management varies. In the BCCF three layers of abstraction are identified (inspired among others by [14, 23]): the *strategic*, *operational* and *service* level spanning from high level requirements to technical realization of collaboration behaviors.

At the strategic level the focus is on behavior that is abstract in nature, describing the purpose and high level requirements an enterprize has with the behavior. The operational conditions under which enterprizes exhibit their behavior are part of the operational level. This level establishes how high level strategic behavior (private, exposed and observable) will be operationalized. The technical realization of operational behavior is done at the service level, describing how the services provided by the IT-infrastructure support the operational activities.

3.3 Facet

The third dimension, **facet** captures the fact that the collaboration behaviors conducted by enterprizes affect many different parts. Facets represent these different parts of a business collaboration behavior that can be observed; and where consequently the focus and type of developer involved in collaboration development and management varies. Five facets are distinguished (inspired by among others [6,21,23]): *what*, *who*, *where*, *when* and *how* facet.

The *what* facet emphasizes the structural view of a collaboration behavior, focusing on what things are used to perform a collaboration behavior. The *how* facet takes a functional standpoint, and thus concentrates on how a collaboration behavior is conducted. The *who* facet concerns the participant(s) conducting the collaboration behavior. The location(s) at which the behavior is carried out are expressed in the *where* facet, whereas its temporal dimension is covered in the *when* facet.

4 Modeling the BCCF

To capture the three dimensions of collaborations aspects, levels and facets of BCCF we employ two types of model: meta models and models, both of which are defined for individual levels. Meta models provide design guidelines in terms of classes and their relationships, where depending on the collaboration aspect being modeled additional constraints are placed on the meta-model. Models represent a particular application design, and are derived by populating a meta model's *classes*.

Every meta model consists of six classes, where each class captures a particular facet; i.e. for *what*, *how*, *where*, *who*, *when* and *why* facet. Every class constitutes a set of logically related *attributes*. *Associations* connect the classes expressing dependencies among facets. *Mappings* define dependencies among levels by providing links between classes that describe the same facet at different perspectives (illustrated by the arrows between facets at different perspectives in Fig. 1).

Snippets of exemplary models for the AGFIL application are illustrated in Fig. 2, showing its strategic, operational and service model respectively; where the models are represented based on UML conventions. In order to distinguish different facets, we represent them in different shapes in their UML models (see also legend in Fig. 2): *what* facet is shown as folded corners, *how* facet as rounded rectangles, *who* facet as octagons, *where* facet as plaques, and *when* facet as hectagons. For more information the reader is referred to [15, 16]; where [17] contains the most recent details.

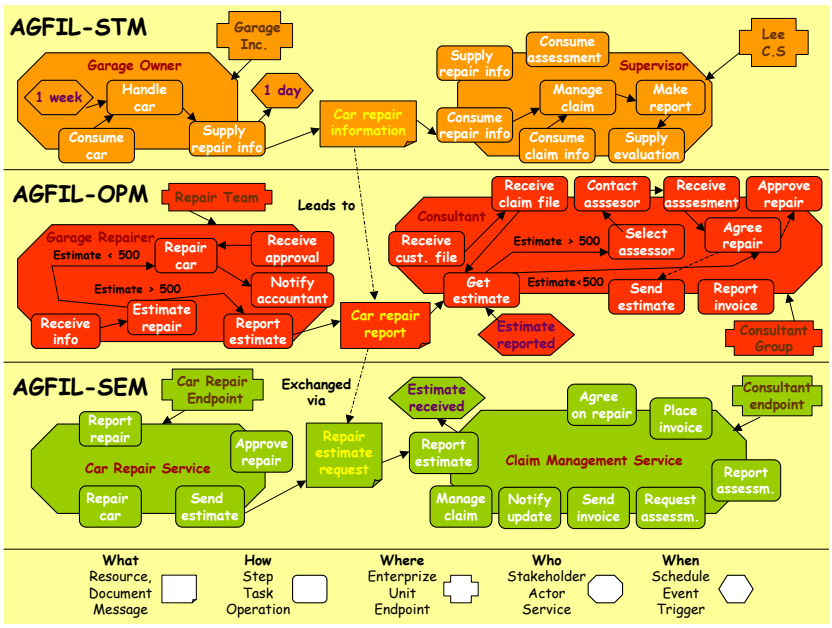

Fig. 2: AGFIL Collaboration Models

4.1 Strategic Models

At strategic level, strategic models like the AGFIL-STM in Fig. 2 capture purpose and high level requirements of business collaborations, akin to requirements analysis [4, 22]. Strategic models are expressed in terms of resources, steps, stake holders, enterprizes, and schedules. Resources such as `car repair information` provide abstractions for means such as financial, human and informational capital. Resources are used and produced by steps which represents high level functions.

Steps are of type 'internal' (like `handle car` presented inside the stakeholder boundary in Fig.2); or type 'communication' representing resource supply and consumption e.g. `consume repair information`. Stake holders like `garage owner` describe the participants involved who are responsible for carrying out defined steps. Stake holders belong to an enterprize, where enterprizes are manifestations record the information about the participating enterprize where behavior is carried out. Stake holders and their enterprizes are bound by schedules reflecting temporal constraints, like the deadline of `1 week` for `handle car`.

4.2 Operational Models

At operational level, operational models like the AGFIL-OPM in Fig. 2 depict how high level strategic behavior is realized in terms of operational activities. These are expressed in terms of documents, tasks, actors, units, and events. Documents (like `car repair report` represent the flow of information in a collaboration behavior. Documents are used and produced by tasks. Tasks represent specific business functions, and are of type 'internal' or 'communication' (represented inside or on the boundary of the actors respectively), e.g. `collect claim form` and `report invoice` respectively.

Actors such as `garage repairer` and `consultant` are responsible for carrying out tasks. Actors instantiate the **Actor** class and belong to units such as `repair team` unit, whose abstract definition is provided by the **Unit** class. In order to assess progress, keep logs to ensure non-repudiation, and etceteras, events are published and subscribed to by actors. Events describe business occurrences which have properties such as 'date', 'time', 'severity'.

4.3 Service Models

At service level, operational models are translated into service models that specify how the described operational behavior is realized using the services offered by the IT-infrastructure. Service models are defined in terms of messages, operations, services, endpoints, and triggers. Messages represents containers of information (e.g., `repair estimate request`), consisting of meta-data and actual data. Messages function as the inputs and outputs of operations such as `place invoice`.

Operations, just as steps and tasks at strategic and operational level respectively, can be dependent on one another. Additionally, they can be of type 'internal' or 'communication'. Operations are grouped in services (e.g. `car repair service`, which constitute collections of logically related operations. Services themselves are provided by endpoints (like `claim handling endpoint`) and have properties 'network location' and 'type'. To express technical occurrences triggers like `claim request acknowledged` can be defined on the basis of the **Trigger** class.

4.4 Mappings between Models

For the specification of dependencies between different collaboration behaviors at different levels, we employ vertical mappings. Vertical map-

pings are realized by providing links between the classes in different meta-models and instance models at different perspectives. The vertical mappings are based on the implicit links that exist between classes that describe the same facet at different levels in the same collaboration behavior. We define the following mappings:

Resources at strategic level are mapped to documents at operational level. Documents themselves are mapped to messages using *exchangedVia* relations. Steps are mapped via *decomposedIn* relations to tasks; whereas tasks are *realizedBy* operations. Stake holders *control* actors, where each actor is *representedBy* a service. Enterprizes are *organizedIn* units, where each unit itself *offers* one or more endpoints. Schedules are *splitInto* events, where each event *causes* multiple triggers.

5 Security in Business Collaboration

Security in general can be viewed as being concerned with establishing the capacity "to be able to avoid being harmed by any risk, danger or threat" [5]. Interpreted in the context of business collaboration security deals with providing assurance to enterprizes that their cooperation is taking place in a secure manner. In other words, security is concerned with providing peace of mind for the businesses involved, where they can rely on the fact that their collaboration is safe from risks like impersonation, unauthorized use of resources, and etceteras.

When put into the business collaboration context as presented in section 3, it follows that for business collaboration security can be perceived at a strategic, operational and service levels, each of which represents a level of abstraction with its own content and meaning regarding security. In the remainder of this section we shall discuss the role of security and the specification of security requirements at the different levels. An overview of the requirements that can be specified is provided in Figure 3.

5.1 Strategic Level Security

As observed at an abstract strategic level a business collaboration constitutes a cooperation between enterprizes making use of each other's business services to exchange resources to further their business goals. At this level security specification deals with the definition of the threats that can jeopardize the successful completion of these resource exchanges. Threat analysis here is aimed at 1) identification of the threat, 2) measuring the magnitude of the potential loss if this threat is not dealt with, and 3) the

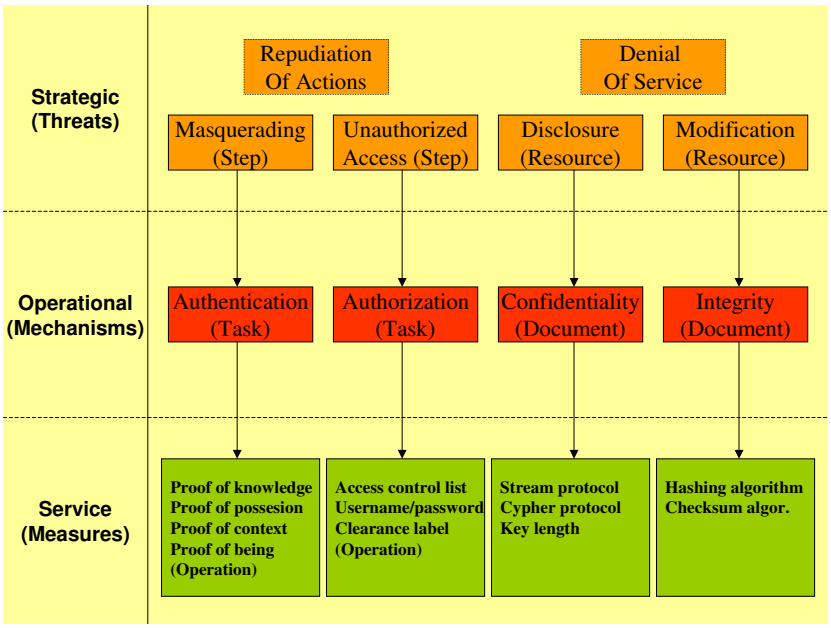

Fig. 3: Security in Business Collaboration

probability that the loss will occur. For business collaboration commonly six types of security threat are identified:

- *Masquerading*

  A first threat is that of masquerading, which is the threat of an entity pretending to be another entity. In the context of business collaboration this involves one stake holder assuming the identity of another stake holder and subsequently act like it by performing the real stake holder's steps. Alternatively, it can be the case that an outside party is trying to infiltrate the collaboration by stealing the identify of one of its stake holders.

- *Unauthorized Access*

  Masquerading is usually the means to an end for attackers to gain access to resources to which they are ordinarily not entitled. They do so by performing steps which they are not allowed to carry out . In this sense masquerading is related to the risk of *unauthorized access*, which pertains to the usage of resources by stake holders or outside parties who are not allowed to do so.

- *Unauthorized Disclosure*

  A third threat in the business collaboration environment is concerned with the disclosure of resources to unauthorized parties. Resources exchanged among business collaboration participants is often of a

sensitive nature, e.g. electronic patient records, insurance claims, payment roll information, etc. It is not difficult to see that the consequences of such private information falling into wrong hands could spell disaster.

- *Unauthorized Modification*

  Rather than dealing with the wrongful disclosure of information the fourth threat, unauthorized modification, focuses on the alteration of resources as they are being exchanged between stake holders. Think for example of some party making unauthorized changes to your product order, or more extreme modifying blueprints of the airplane you are constructing.

- *Repudiation of Actions*

  Repudiation of actions is the fifth threat, and expresses the danger against accountability where stake holders can deny having performed certain actions. Typical situations where repudiation is an issue, is in the domain of bank transactions, bid offers in auctions, and so on.

- *Denial of Service*

  The sixth and final threat is denial of service, capturing the possibility that authorized participants are unable to access a business service due to unavailability. Another option is that requests for service usage are blocked or delayed, which can be viewed as an extreme case of unauthorized modification.

In this report we focus on prevention of the first four security threats. The fifth and sixth threat are not taken into consideration, as the defensive arrangements required for their neutralization are largely outside the scope of the field of security (such as monitoring and logging for repudiation, and load balancing, service pooling, etceteras for denial of service).

5.2 Operational Level Security

Based on the threat analysis at the strategic level security measures are determined at the operational level to establish defenses capable of tackling the identified threats. At this level a business collaboration constitutes the sending and receiving of appropriate documents by enterprizes to further the state of the business collaboration. As such, the security mechanisms that will be employed, are to provide protection for this document communication. In correspondence with the described threats in subsection 4.1 we identify the following security mechanisms:

- *Masquerading → Authentication*

  Countering identity theft requires an unambiguous way in which the identity of a participant can be established. For this purpose an authentication mechanism is employed, which revolves around using a combination of something an actor knows, has and possesses to check its identity claim. Authentication can be done directly by the actor interested in verification of the identity, or indirectly where this process is delegated to a trusted third party. Only when the actor has provide proper authentication for a task, he/she is allowed to perform it.

- *Unauthorized Access → Authorization*

  In order to be able to verify access to business services, it must be clear who is allowed to what. This necessitates the presence of an authorization mechanism, which limits and controls access to information and resources by addressing the range of things an actor is allowed to do. The most common approaches for authorization are grounded on a discretionary, mandatory or role based scheme [11], depicting for each task the scheme used to authorize actors. Authorization often requires authentication, however, this is not always the case. For example, entrance tickets give you the right to access something, but do not require any identification by themselves.

- *Disclosure → Confidentiality*

  The security mechanism used to prevent unauthorized disclosure of resources is confidentiality, which helps avoid situations where unauthorized parties can observe information. Confidentiality is usually associated with some form of encryption mechanism to make documents unreadable; where a distinction is often made between cryptographic and hashing techniques. Cryptographic techniques provide means of converting information from its normal, comprehensible form into an incomprehensible format, rendering it unreadable without secret knowledge. Hashing techniques produce a document digest, which appears random to attackers and does not leak information about the content itself.

- *Unauthorized Modification → Integrity*

  In order to ensure that information can not be modified during its exchange without this being notice, the mechanism of integrity is required. Integrity deals with ensuring that any alteration to a resource will be detectable. This mechanism is passive in the sense that modification can not be prevented. Rather its purpose of detection is focused on identifying modifications to documents based upon which

appropriate action can be taken. Integrity, like confidentiality, often utilizes some form of hashing technique. Because the digest produced in hashing is (in theory) unique for any document, any change (even a single letter) will result in a different digest. Checksum algorithms like cyclic redundancy checking are another option, providing a simple form of protection against document modifications.

Together these four security mechanisms provide the building blocks required to tackle the threats present in the business collaboration environment. Furthermore, these mechanisms are often used in conjunction to further strengthen security defenses, for example using confidentiality and integrity to protect authentication or authorization information, or utilizing authentication to first establish identity upon which an authorization mechanism is grounded.

5.3 Service Level Security

The security mechanisms selected at the operational level must subsequently be realized at the service level via security measures. This level is the domain of the service oriented computing paradigm. In this paradigm a business collaboration is viewed as a set of interacting technical services, where these interactions are message based and facilitate the communication of information among services. In order to meet the business driven security demands at this level, the message based interactions, i.e. message exchanges, must be adequately protected.

For this purpose we have identified the following security measures; where these are discussed grouped in accordance with the security mechanism they realize:

- *Authentication → Proof of knowledge, possession, context, being*

  Authentication of actors at a service level involves establishing the identity of the service requester on a per-operation basis. Four types of proof can be used to verify a requester's identity, being proof of knowledge, proof of possession, proof of context and proof of being. The first and last one provide a direct means of authentication; whereas the second and third are indirect in nature (that is, the actual authentication is done by an external party).

  Proof of knowledge requires the requester to provide some specific knowledge like a username/password combination, date of birth, pin, etc. This allows the direct authentication of the requester without

any intervening parties; Similarly, proof of being is a direct authentication means e.g. dna, facial scan, iris scan, thumbprint, and so on. Although these checks involve physical features, the resulting scanning information is digital in nature and can thus be utilized at a service level to facilitate authentication.

Proof of possession involves the requester demonstrating his/her ownership of something such as a birth certificate, passport, credit card, or a kerberos or X.509 certificate. These are typically granted by parties other than the one offering the service. Proof of context concerns the requester proving that he/she is trying to access the operation from a certain location and/or at a particular time; which are also usually not determined by the service provider itself.

- *Authorization → Access control list, username/password, clearance*

  Authorization revolves around determining whether an actor can perform a particular task. As observed in subsection 5.2 the most common approaches for authorization are grounded on a discretionary, mandatory or role based scheme. Accordingly, at the service level we identify three characteristics to realize each of these schemes.

  Opting for the discretionary scheme for a task's authorization at operational level, means specifying at service level for the related operations that these define access control lists depicting who can/cannot access the operation. The mandatory scheme is facilitated by defining clearance labels like the security levels used in the military for each operation. The role based scheme is enabled via the specification of what roles can perform what operation; and which requester can play which role.

- *Confidentiality → Stream protocol, cypher protocol, key length*

  Achievement of confidentiality at operational level is possible via the application of cryptographic and/or hashing techniques to a document. If the former is chosen, then at service level the message transporting this document will utilize some sort of stream protocol to realize this (like RC4). In addition, the length of the key to be used as input for the encryption process must be specified. In the case hashing is chosen to protect the document, a cypher protocol such as RSA or DEA will be employed.

- *Integrity → Hashing algorithm, Checksum algorithm*

Integrity of documents at operational level is supported at service level through the usage of hashing and/or checksum algorithm. This depends on the choice made at operational level, that is, whether hashing or checksum was chosen as the mechanism to establish document integrity. Hashing algorithms here are the same as discussed just now for confidentiality. Examples of checksum algorithms include the different CRC based protocols.

Observe that the above described measures at service level are not intended to be exhaustive in nature. The authors are aware that many other measures exist; the above is therefore intended to be of illustrative nature to show how operational security mechanisms may be realized at a technical service level.

6 Conclusions

In this technical report we addressed the issue of specification of security requirements for business collaborations. This work is motivated by the lack of support thereof in the current research with regard to relating high level security objectives to concrete security measures; as most work (like [7, 1, 8]) focuses on provision of low level security measures without taking higher level, business driven security requirements into consideration.

To remedy this situation we introduced our generic framework for capturing the business collaboration context; and explained how this context can be described via the usage of various meta models and models. After that we explained how these meta models and models can be augmented to facilitate security requirement specification at strategic, operational and service level. Furthermore, we established relations between the requirements at these different levels; as such enabling traceability of strategic security objectives to operational security mechanisms to service level security measures (and vice versa).

A caveat concerns the defined security properties: these are not intended to be exhaustive in nature nor do the authors expect them (and the relations between them) to be final. As the authors are not themselves experts in the field of security, more work to further develop the (currently basic) support for security requirement specification is required. However, we believe that the presented approach provides a first step on the road to comprehensive security requirement specification for business collaboration.

References

[1] Atkinson, B., G. Della-Libera, S. Hada, M. Hondo, P. Hallam-Baker, J. Klein, B. LaMacchia, P. Leach, J. Manferdelli, H. Maruyama, A. Nadalin, N. Nagaratnam, H. Prafullchandra, J. Shewchuk, D. Simon, Web Services Security (WS-Security), *http://www-106.ibm.com/developerworks/webservices/library/ws-secure/, 2002*

[2] S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Malhotra, A. Nadalin, N. Nagaratnam, M. Nottingham, H. Prafullchandra, C. von Riegen, J. Schlimmer, C. Sharp, J. Shewchuk, Web Services Policy Framework (WS-Policy), *http://www-106.ibm.com/developerworks/library/specification/ws-polfram/, September 2004*

[3] M. Bartel, J. Boyer, B. Fox, B.rian LaMacchia, E. Simon, XML-Signature Syntax and Processing *http://www.w3.org/TR/xmldsig-core/, 2002*

[4] P. Bresciani et al, Tropos: An Agent-Oriented Software Development Methodology, *Autonomous Agents and Multi-Agent Sytems, Vol. 8, No. 3, pp. 203-236, 2004*

[5] Cambridge Learner's Dictionary, *http://dictionary.cambridge.org*

[6] B. Curtis et al, Process Modeling, *Communications of the ACM, Vol. 35, No. 9, pp. 75-90, 1992*

[7] G. Della-Libera, B. Dixon, P. Garg, S. Hada, P. Hallam-Baker, M. Hondo, H. Maruyama, N. Nagaratnam, A. Nash, R. Philpott, H. Prafullchandra, J. Shewchuk, D. Simon, E. Waingold, R. Zolfonoon, Web Services Secure Conversation Language (WS-SecureConversation), *http://www-106.ibm.com/developerworks/library/specification/ws-secon/, 2002*

[8] G. Della-Libera, P. Hallam-Baker, M. Hondo, T. Janczuk, C. Kaler, H. Maruyama, N. Nagaratnam, A. Nash, R. Philpott, H. Prafullchandra, J. Shewchuk, E. Waingold, R. Zolfonoon, Web Services Security Policy (WS-SecurityPolicy), *http://www-106.ibm.com/developers/library/ws-secpol/, 2002*

[9] R. Dijkman et al, Service-oriented Design: A Multi-viewpoint Approach, *International Journal of Cooperative Information Systems, Vol. 13, No. 4, pp. 337-368, 2004*

[10] P. Grefen et al, CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises, *International Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, pp. 277-290, 2000*

[11] W. van den Heuvel, K. Leune, M. Papazoglou, EFSOC: A Layered Framework for Developing Secure Interactions between Web-Services, *Kluwer Academic Publishers, Vol. 13, No. 12, pp. 1-38, 2005*

[12] T. Imamura, B. Dillaway, E. Simon, XML Encryption Syntax and Processing, *http://www.w3.org/TR/xmlenc-core/, 2002*

[13] K. Leune, M. Papazoglou, W. van den Heuvel, Specification and Querying Security Constraints in the EFSOC Framework, *Proceedings of the 2d International Conference on Service Oriented Computing, New York City, USA, 2004*

[14] Object Management Group, Model Driven Architecture, *http://www.omg.org/docs/ormsc/01-07-01.pdf, July 2001*

[15] B. Orriens et al, Bridging the Gap between Business and IT in Service Oriented Business Collaboration, *Proceedings of the IEEE International Conference on Services Computing, Orlando, Florida, USA, July 2005*

[16] B. Orriens et al, Establishing and Maintaining Compatibility in Service Oriented Business Collaboration, *Proceedings of the 7th International Conference on Electronic Commerce, Xi'an, China, August 2005*

[17] B. Orriens, Modeling The Business Collaboration Context, *INFOLAB Technical Report Series, No. 28, Tilburg, The Netherlands, January 2006*

[18] M. Papazoglou, G. Georgakapoulos, Introduction to the Special Issue about Service-Oriented Computing, *Communications of the ACM, Vol. 46, No. 10, pp. 24-29*

[19] C. Peltz, Web services orchestration: a review of emerging technologies, tools, and standards, *Hewlett Packard White Paper, January 2003*

[20] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, Role-Based Access Control Models, *IEEE Computer, 1992*

[21] A. Scheer, Architecture for Integrated Information Systems - Foundations of Enterprise Modeling, *Springer-Verlag New York, Secaucus, NJ, USA, 1992*

[22] P. Traverso et al, Supporting the Negotiation between Global and Local Business Requirements in Service Oriented Development, *Proceedings of the 2d International Conference on Service Oriented Computing, New York, USA, 2004*

[23] J.A. Zachman, A framework for information systems architecture, *IBM Systems Journal, Vol. 26, no. 3, pp. 276-292, 1987*