

Presentation and Validation of Method for Security Requirements Elicitation from Business Processes

Naved Ahmed and Raimundas Matulevičius^(✉)

Institute of Computer Science, University of Tartu,
J. Liivi 2, 50409 Tartu, Estonia
{naved,rma}@ut.ee

Abstract. In recent years, the business process modelling is matured towards expressing enterprise's organisational behaviour. This shows potential to perform early security analysis to capture enterprise security needs. Traditionally security in business processes is addressed either by representing security concepts graphically or by enforcing security constraints. But such security approaches miss the elicitation of security needs and their translation to security requirements for system-to-be. This paper proposes a method to elicit security objectives from business process models and translate them to security requirements. As a result, the method contributes to an alignment of business processes with the technology that supports the execution of business processes. The approach applicability is illustrated in few examples and its validity is reported in the comparative study.

Keywords: Security in business processes · Business process modelling · Requirements engineering

1 Introduction

There has been several attempts to engage the relatively matured security requirements engineering in business processes. However, the majority of studies either focusses on the graphical representation of security aspects in business process models [15, 22] or enforces the security mechanisms [10] or both [23]. These studies analyse major problems when addressing security engineering in business process modelling. Firstly, security requirements are specified in terms of security architectural design (i.e., security control) and missing the rationale about the trade-offs of the security decision. Secondly, the requirement elicitation is either missing or haphazard: this leads to miss some critical security requirements. And finally, due to the dynamic and complicated nature of business processes, the studies only address varying aspects (i.e., authorisation, access control, separation of duty or binding of duty) but not the overall security of business processes. These problems can be overcome by eliciting security objectives from

business processes and by transforming them to the security requirements where the technology supports the business processes execution.

In this paper we analyse *how to determine security objectives from the business process models and to translate them to security requirements*. In a previous study [1] we have presented a method for security requirements elicitation from business processes (SREBP). The goal of this paper is to highlight what analysts need to do in order to define security models and to elicit security requirements using the SREBP method. In addition we present a comparative analysis of the coverage of security requirements sets elicited using the SREBP method and the security quality requirements engineering (a.k.a., SQUARE) approach. To show the generalisation of the SREBP application and to understand whether the results could be replicated, we briefly report on two SREBP (and SQUARE) application cases.

The rest of the paper is structured as follows. In Sect. 2 we presents the SREBP method using the land management example. In Sect. 3, validity of the proposal is analysed. Section 4 presents some related studies. Finally, Sect. 5 concludes the paper and presents some future work.

2 The SREBP Method

2.1 Illustrative Example: Land Management System

To perform the security requirements elicitation one needs to collect the knowledge of enterprise *value system* from the *value chain* and the *business functions*. Figure 1 illustrates a value chain for the land management system (LMS) example. It organises the enterprise business functions and relates them to each other (as enterprise cooperates to achieve the business goals). In Fig. 2 a detailed workflow of Prepare Plan process is given. The process has two business partners (Lodging Party and Planning Portal) expressed as swimlanes, while Registry is identified as an information system.



Fig. 1. Land management systems - value chain

Similarly to Prepare Plan, other sub-processes (e.g., Lodge Plan, Examine Plan, Approve Plan and Update Plan) are also expanded to the operational models. But in Sect. 2.2, we will present the SREBP method using the Prepare Plan process (as illustrated in Figs. 1 and 2).

2.2 Security Requirements Elicitation Method

In [2], we have presented a set of security risk-oriented patterns for securing business processes. Based on these patterns, the SREBP method helps deriving

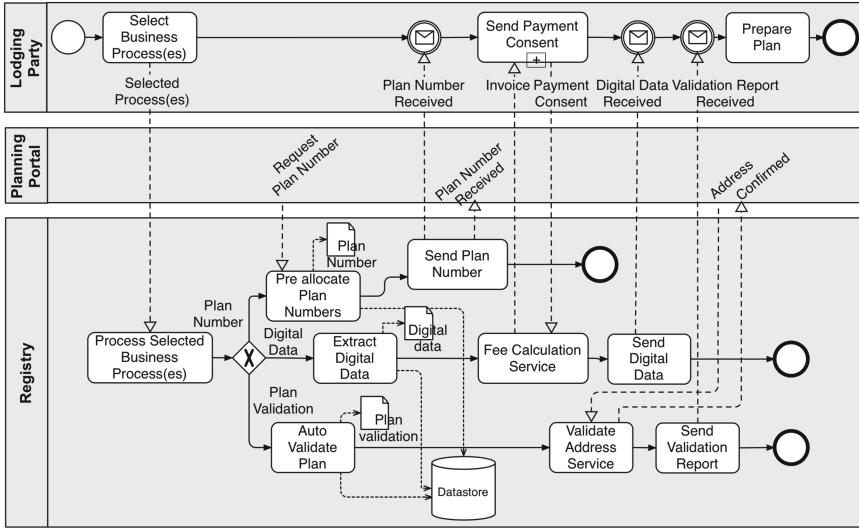


Fig. 2. Operational business process - prepare plan

security requirements as constraints that have to be respected when executing business processes. The first stage (see Fig. 3) is *business asset identification and security objective determination*. In the second stage, the *elicitation of security requirements* is done from the system’s contextual areas.

2.3 Stage 1: Business Assets Identification & Security Objectives Determination

The first stage starts with the analysis of the *value chain* (see Fig. 1) from which the assets that must be protected against security risks are determined. The stage requires collaboration between security analysts and the stakeholders from the analysed enterprise. It consists of two activities:

(i) *Identify business assets:* During this activity the central artefact (or artefacts) considered in the value chain is identified. Typically, further details of this artefact are considered in the business process model, like *Prepare Plan Process* in Fig. 2. The enterprise’s value chain can either have a single artefact used in all the processes or comprised of multiple artefacts in each operational business process. In the LMS case, *Plan* is identified as a protected asset, since, it is the central artefact used in all the business processes (see Figs. 1 and 2).

(ii) *Determine security objectives:* The activity addresses determining of key security objectives – confidentiality, integrity and availability – for identified business assets. In the LMS case, we define the following security objectives for business asset *Plan*: *i*) *Plan* should be confidential, i.e., no unauthorised individual should read it and its relevant data; *ii*) *Plan* should be integral, i.e., the *Plan* and its relevant data should not be tempered; and *iii*) *Plan* and its relevant data should be available to the business partners at anytime.

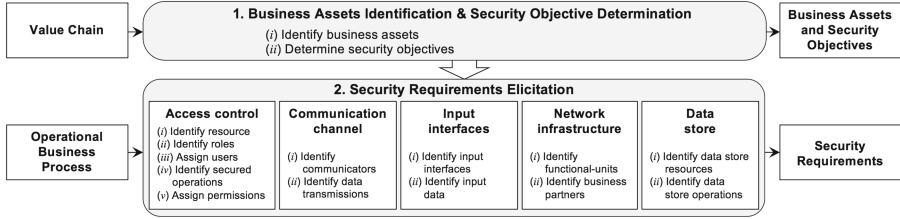


Fig. 3. Security requirements elicitation method

2.4 Stage 2: Security Requirement Elicitation

At the second stage, the security requirements elicitation is performed at five contextual areas: *access control*, *communication channel*, *input interfaces*, *network infrastructure*, and *data store*. It is important to note that each artefact–*data* or *process* – separately considered and protected at each area, contributes to the security of the business asset (i.e., *Plan*) identified at the first stage.

Access Control specifies how the business assets could be manipulated by individuals, applications or their groups. The major concern is to protect the confidentiality of identified business asset, in our example the *Plan*, when it is being manipulated by the IS asset, (i.e., the *Registry*). The security threat arises if the access to the *Plan* and its properties, like (*Plan Number*, *Digital Data*, and *Plan Validation*) is allowed to users without checking their access permissions. The risk event would: *i*) negate confidentiality of *Plan*, *ii*) lead to the *Plan* unintended use, and *iii*) harm the *Registry*'s reliability. A way to mitigate the security risk is the introduction of access control mechanism, for example the Role-Based Access Control (RBAC) model. The RBAC model is elicited by performing the following activities:

(i) Identify resource: Hence, the business asset (i.e., *Plan*) is defined as a resource that needs to be protected from the unauthorised access. The protected resource is characterised by its attributes that add value to the asset. For example, in Fig. 4, *Plan Number* and *Digital Data Number* are attributes of *Plan* derived from the operational business process models.

(ii) Identify role: Roles are determined from the operational business process. The swimlanes are considered as outside role while the lanes of an information system corresponds to the internal role. We consider both outside and internal roles, since they both could access the secured business asset i.e., *Plan*. These roles (e.g., *Lodging Party* and *Planning Portal*) are modelled using `«role»` stereotype in RBAC security model (see Fig. 4).

(iii) Assign users: This activity assigns roles to users, which are instances of some role. Usually it is not possible to elicit concrete users from the operational business process. This, potentially, requires expertise of and collaboration with the domain experts.

(iv) *Identify secured operation*: An operation is an executable set of actions that can change the state of the protected resource. In this activity, any business activity (including both the task and sub-process) from the operational business processes that accesses the protective resource is identified as secured operation. For instance, Pre allocate Plan Numbers, Send Plan Number, Fee Calculation Service, and etc. are secured operations which manipulate properties Plan Number, Digital Data and Plan Validation (Fig. 4).

(v) *Assign permissions*: Permissions characterise role privileges to perform operations on the protected resource. In this activity, permissions specify the security actions –namely, *Create*, *Read* and *Update*– over secured operations that the role can perform to change the state of the protected resource. For example, Lodging Party role has the permission to create resource Plan.

By executing these activities, an RBAC security model (Fig. 4) is developed. Based on this model, the security requirement *check for the access rights* is evolved to the following context specific security requirements.

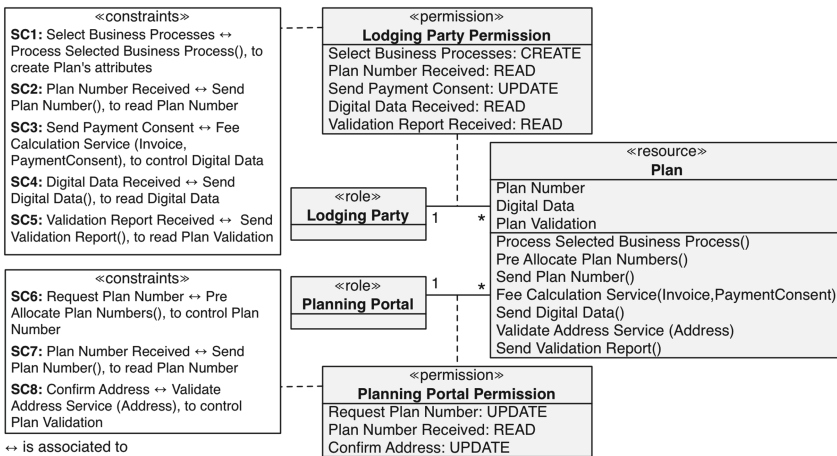


Fig. 4. RBAC Security model - prepare plan business process

- RQ1.** Lodging Party should be able to:
1. create or initialize the Plan Number, Digital Data and Plan Validation.
 2. read the Plan Number, Digital Data and Plan Validation.
 3. update the Digital Data.
- RQ2.** Planning Portal should be able to:
1. update the Plan Number and Plan Validation.
 2. read the Plan Number.

The security model (i.e., Fig. 4) defines how authorised parties should access the protected resources. However, it does not support capturing scenarios like *entailment constraints* [11], *delegation constraints* [4] and *usage control* [19]. These

requirements could be determined in collaboration between business and/or security analysts. For example, the following entailment constraints could be defined:

RQ3. Fee Calculation Service should be performed by different users assigned to the Lodging Party.

RQ4. Pre allocate Plan Numbers and Send Plan Number should be performed by the same user with Planning Portal's role.

RQ3 defines that there should exist at least two users in the Registry with the same role, to finish executing the task Fee Calculation Service: the first user issues the Invoice and the second user approves the Payment Consent. Requirement RQ4 highlights the concept binding of duties.

Communication Channel is used to exchange data between business partners (e.g., Lodging Party and Planning Portal) and system (e.g., Registry). Here, data, like Selected Business Process(es), Payment Consent and etc., need to be protected when they are transmitted over the (untrusted) communication channel, i.e., Internet. The communication channel could be intercepted by the threat agent and the captured data could be misused (i.e., read and kept for the later use or modified and passed over) by the threat agent. This could lead to the loss of the channel reliability, and could negate the confidentiality and integrity of the Plan. To mitigate the risk, in this contextual area one performs two activities:

(i) *Identify communicators:* Communicators are the entities that transmit or receive data. Operational business processes are considered to identify the information system of an enterprise and their business partners who exist outside of an enterprise but transmit/receive data to/from the enterprise. In Fig. 5, we illustrate a security model for communication channel between Registry and Lodging Party using a UML interaction diagram. Registry is modelled as LMS's information system that communicates with the Lodging Party identified as LMS's business partner.

(ii) *Identify data transmission:* One needs to determine the business asset and/or its relevant data transmitted or received between the identified communicators over the untrusted communication channels, i.e., Internet. For example, Selected Process(es) and Plan Number are communicated between Registry and Lodging Party, thus, they require to be protection.

In order to ensure the secure transmission of business assets or its relevant data, the above activities results in the following security requirements for the Lodging Party and Registry and correspondingly for other entities that communicates with Registry:

RQ5. Registry should have unique identity in the form of key pairs (public key, private key) certified by a certification authority.

RQ6. Lodging Party and Planning Portal should encrypt and sign Selected Process(es), Plan Number, and other using keys before sending it to Registry.

A security requirements implementation could be fulfilled by the standard transport layer security (a.k.a., TLS) protocol [3] as illustrated in Fig. 5. As the first contact, the Lodging Party sends Registry a handshake message, which includes

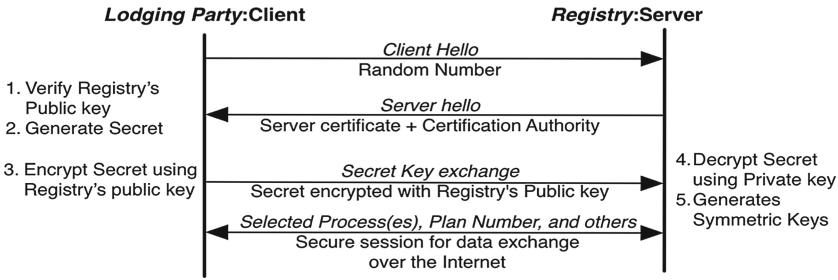


Fig. 5. TLS Protocol implementation, adapted from [3]

a random number. Following RQ6, the Registry responds with its public key and the information about the certification authority. After verification of the Registry’s public key, the Lodging Party generates the secret and sends it to the Registry encrypted with the Registry’s public key. The Registry then decrypt the secret using the private key and generates symmetric session keys. The keys enable Lodging party and Registry to establish a secure session for data exchange. Following RQ7, encryption keeps the transmitted data (e.g., Selected Business Process(es), Payment Consent and etc.) confidential and signing it ensures that the received data is not tempered. The secure communication continues until it is not explicitly terminated by Lodging Party or Registry.

Input interfaces ensure that the input data submitted by business partners are correct and complete. In this contextual area two activities are suggested:

(i) *Identify input interfaces*: The activity identifies the input interfaces of information system from the operational business processes that has incoming message flows. The input interfaces are those activities of information system that receives input from the enterprise stakeholders.

(ii) *Identify input data*: The activity identifies the input data received by the input interfaces from the enterprise’s business partners.

In LMS (see Fig. 2), we identify Process Selected Business Process(es) and Fee Calculation Service as input interfaces of Registry that receives the Selected Process(es) and Payment Consent from Lodging Party. The threat agent can exploit the vulnerability of the input interfaces by submitting the data with a malicious scripts. If happening so the availability and integrity of any activity (e.g., Send Digital Data) after the input interface (e.g., Fee Calculation Service) may be negated. To avoid this risk the following security requirements must be implemented for the identified input interface:

- RQ7.** Fee Calculation Service should filter Payment Consent.
- RQ8.** Fee Calculation Service should sanitize Payment Consent to transform it to the required format.
- RQ9.** Fee Calculation Service should canonicalize Payment Consent to verify against its canonical representation.

Input filtration [6] (RQ8) validates the input data against the secure and correct syntax. The string input should potentially be checked for length and

character set validity (e.g., allowed and blacklisted characters). The numerical input should be validated against their upper and lower value boundaries. *Input sanitization* (RQ9) should check for common encoding methods used (e.g., HTML entity encoding, URL encoding, etc.). The *input canonicalization* [6] (RQ10) verifies the input against its canonical representation.

Network infrastructure secures the infrastructure where the information system is deployed and where it executes its tasks. The enterprise's information system is composed of several small functional units, which can either be deployed at single location or multiple locations connected through internet. The goal of this contextual area is to guarantee availability of these functional units to the enterprise user or their partners. Two activities are performed within this contextual area:

(i) *Identify functional-unit*: A functional-unit is an activity or sub-process implemented on independent network infrastructure to provide certain functionality of an enterprise's information system. An information system can comprised of one or more functional-units. In LMS case, their information system (i.e., Registry) illustrated in Fig. 2 is consists of three functional-units (i.e., Pre allocate Plan Numbers, Fee Calculation Service and Validate Address Service) deployed on three independent network infrastructure connected through internet to form a single information system (i.e., Registry) for LMS. Later, we demonstrate the security requirements elicitation of Pre allocate Plan Numbers functional-unit using a UML security model (see Fig. 6).

(ii) *Identify business partner*: Business partners are the external entities that can access the network infrastructure in order to communicate with the enterprise information system. The access includes any request type necessary to receive or send data. In LMS, we identify Lodging Party and Planning Portal as external entities that communicate with Registry in Prepare Plan process.

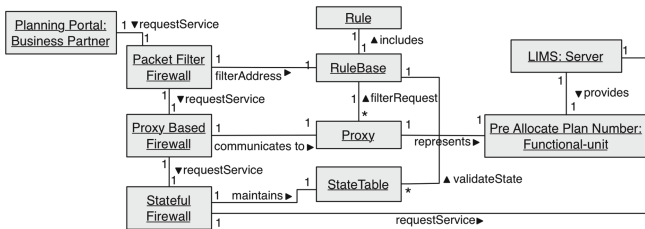


Fig. 6. Firewall architecture, adapted from [24]

In Fig. 2, Registry has a functional-unit Pre allocate Plan Numbers offered to Planning Portal through the communication channel. The threat agent may exploit the hosts in the channel and hack them because of the protocol (e.g., TCP, ICMP or DNS [5]) vulnerability; i.e., the ability to handle an unlimited number of requests for service. When receiving simultaneously multiple requests, the server i.e., Registry, will not be able to handle them, thus, the services become unavailable.

The successful denial of service attacks could also provoke the loss of partner's (e.g., **Planning Portal**) confidence on **Registry**. The above activities helps to develop a UML security model (see Fig. 6) that defines three types of firewalls [24] – **Packet Filter Firewall**, **Proxy Based Firewall** and **Stateful Firewall**. The security model introduce the following requirements to mitigate the risks to the functional-units of **Registry**:

RQ10. Pre allocate **Plan Numbers** should establish a rule base (i.e., a collection of enterprise' constraints used by different firewalls) to communicate with **Lodging Party** and **Planning Portal**.

RQ11. **Packet Filter Firewall** should filter the **Planning Portal's** address to determine if it is not a host used by the threat agent.

RQ12. **Proxy Based Firewall** should communicate to the proxy which represents Pre allocate **Plan Number** to determine the validity of request received from **Planning Portal**.

RQ13. **State Firewall** should maintain the **state table** to check the **Planning Portal's** request for additional conditions of established communication.

It is important to notice that the communication between the **Planning Portal** (and also **Lodging Party**) and the **Registry** is bidirectional. The similar requirements must be taken into account when **Registry** sends messages (e.g., **Fee Calculation Service** sends **Invoice**) back to the business party.

Data Store is used to define how data are stored and retrieved to/from the associated databases (e.g., **Data store** in Fig. 2). If the threat agent is capable of accessing and retrieving the data, their confidentiality and integrity would potentially be negated, thus, resulting in the harm of the business asset (i.e., the **Plan**) and its supporting IS assets (i.e., the **Registry**). To avoid unauthorised access to the datastore we introduce a RBAC model. In this contextual area, the RBAC model is developed using the following activities:

(i) *Identify Datastore resource:* In this context, **Datastore** is identified as a single collective resource. The identified business assets and their related data in the operational process models are modelled as the resource attributes. In Fig. 7, the attributes **Plan Number**, **Digital Data** and **Plan Validation**, actually, represents the attributes of business asset **Plan**.

(ii) *Identify Datastore's operations:* The activity identifies operations that save or retrieve the data, identified in previous activity, from **Datastore**. These operations are modelled as operations of **Datastore's** resource in the RBAC model as illustrated in Fig. 7.

Once the resource and operations are modelled, the activities *identify role* and *assign permissions* are performed as described in the *access control* contextual area. This results in a security RBAC model for enterprise's **Datastore** given in Fig. 7. The security model helps to elicit the following **Datastore's** requirements that ensure the integrity and confidentiality of stored business assets.

RQ14. The **Registry** should audit the operations after the retrieval, storage or any other manipulation of data in the **Data store**.

Auditing (supported by the access control policy) is the process of monitoring and recording selected events and activities [18]. It determines who performed what operations on what data and when. This is useful to detect and trace security violations performed on the Plan Number, Digital Data and Plan Validation.

RQ15. The Registry should perform operations to hide/unhide data when they are stored/retrieved to/from the Data store.

A possible RQ15 implementation is cryptographic algorithms. The encryption offers two-fold benefits: (i) the data would not be seen by the Data store users (e.g., database administrator) where the circumstances do not allow one to revoke their permissions; (ii) due to any reasons if someone gets physical access to the Data store (s)he would not be able to see the confidential data stored.

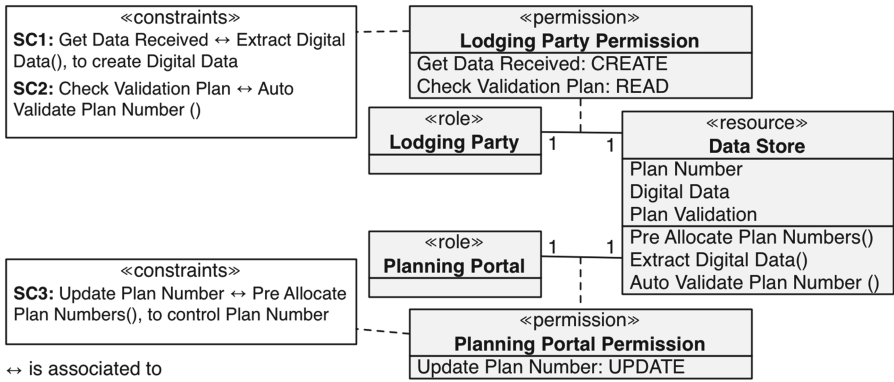


Fig. 7. RBAC security model - data store auditing

3 Validation

3.1 Validation Design

The research question of this validation is to determine which method – SREBP or SQUARE – results in a higher *coverage* of resulting security requirements. We have consequently applied both method on the business processes modelled using BPMN. These models were used as the input for both methods. The business process models included the activities whose execution is supported by the information system or its architecture. Hence we have received the results after applying the methods on the same set of business process models. Both methods were applied by two persons, but these were different in the studied cases.

The SQUARE method [14] is developed as a systematic and flexible approach to elicit security requirements from various sources. Its major steps are: agreement on definition, identification of security goals, selection of elicitation techniques, development of artefacts, risk assessment, elicitation of security requirements,

requirements categorisation, prioritisation, and inspection. We selected SQUARE with the purpose to compare it to SREBP. The SQUARE input was limited to the value chain and process models. We note that if applying the SQUARE in the broader context, the outcome potentially would be different.

The received sets of security requirements were confronted to the security requirements categories (see Sect. 3.2) in order to identify their coverage regarding these categories. The received results were compared to answer the validation research question.

3.2 Coverage of Security Requirements

In this study coverage of the security requirements is estimated as the aggregation of the different security requirements categories, defined following the existing literature [9, 24]. *Identification* requirements are security requirements that associate an individual or application with its unique identity before any interaction with the information system. *Authentication* requirements are security requirements that recognise and validate the individual's identity before interacting with the information system. *Authorisation* requirements are the security requirements that describe the role or individual authorised to access the business assets or its related data in the information system. *Accounting* requirements are security requirements to record security related actions or events (e.g., unauthorised access or communication to an information system or its datastore) and make the information available about these actions or events. *Audit* requirements are security requirements to analyse the information captured using security accounting requirements and verify against a set of valid rules to indicate if there is any security violations happened. *Non-repudiation* requirements are security requirements that capture and maintain the evidence to identify the individuals participated in an activity (e.g., transaction or interaction) to provide protection if they deny their involvement. *Immunity* requirements are the security requirements to specify the ability of an information systems to protect itself from unauthorized access undesirable programs (e.g., viruses or application-specific attacks). *Data exchange* requirements are security requirements to protect the confidential business data from unauthorised access during transmission over internet. We assume that these categories mutually covers the 100 % of security requirements – therefore, each category contributes 12,5 % coverage to the total.

In this study we apply both SQUARE and SREBP to elicit security requirements from two business processes. In order to assess the coverage of each category, we use a 5-grade scale (0, 25, 50, 75 and 100 %). The coverage is assessed by analysing how many of the asset's attributes was addressed by the security requirements. If none of the asset's attributes had been addressed, it was given the value 0 % and if all attributes were addressed, the value 100 % was assigned. Similarly, the values 25 % (few attributes addressed), 50 % (half of the attributes addressed), 75 % (more than half but not all attributes addressed), were assigned for each criterion.

3.3 Laboratory Information Management System

Analysis of laboratory information management system (LiMS) was executed by two researchers, including the first author of this paper. The value chain (see Fig. 8) comprised of 7 business processes –namely Offer Quote Process, Project Registration Process, Quality Check Process, Check Inventory Process, Prepare Samples Process, Process Samples Process, and Deliver Samples Process. The business processes are expanded to business process models. Once the security requirements are derived using SREBP and SQUARE, their *coverage* is compared as illustrated in Fig. 9. It is important to note that although the SREBP resulted in higher number of security requirements than SQUARE, we do not take this numbers into account when comparing the security requirements coverage, because our goal is to understand the coverage regarding each requirements category.

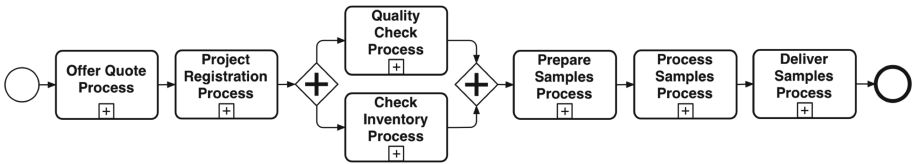


Fig. 8. LiMS Value chain

Methods		SREBP										SQUARE										DIFFERENCE
Business Assets	Categories	Identification	Authentication	Authorization	Accounting	Audit	Non-repudiation	Immunity	Data exchange	TOTAL	Identification	Authentication	Authorization	Accounting	Audit	Non-repudiation	Immunity	Data exchange	TOTAL			
		12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	100	
Offer Quote	Reqs.	2	40	14	18	12	2	59	6	153	2	2	10	6	0	0	13	4	37	116		
	%age	75%	100%	58%	100%	67%	100%	100%	67%	83%	75%	33%	58%	75%	0%	0%	75%	33%	44%	40%		
	Coverage	9.38	12.50	7.29	12.50	8.33	12.50	12.50	8.33	83.33	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	39.56		
Project	Reqs.	2	46	22	29	18	1	66	10	194	2	2	12	6	0	0	13	4	39	155		
	%age	75%	100%	58%	100%	67%	100%	100%	67%	83%	75%	33%	58%	75%	0%	0%	75%	33%	44%	40%		
	Coverage	9.38	12.50	7.29	12.50	8.33	12.50	12.50	8.33	83.33	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	39.56		
Sample Quality	Reqs.	2	50	7	18	12	0	75	2	166	2	2	11	6	0	0	15	4	40	126		
	%age	75%	100%	58%	100%	67%	0%	100%	67%	71%	75%	33%	58%	75%	0%	0%	75%	33%	44%	27%		
	Coverage	9.38	12.50	7.29	12.50	8.33	0.00	12.50	8.33	70.83	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	27.06		
Purchase Order	Reqs.	2	46	14	15	10	1	66	4	158	2	2	13	6	0	0	15	4	42	116		
	%age	75%	100%	58%	100%	67%	100%	100%	67%	83%	75%	33%	58%	75%	0%	0%	75%	33%	44%	40%		
	Coverage	9.38	12.50	7.29	12.50	8.33	12.50	12.50	8.33	83.33	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	39.56		
Sample Plate	Reqs.	2	40	11	29	18	1	57	2	160	2	2	11	6	0	0	15	4	40	120		
	%age	75%	100%	58%	100%	67%	100%	100%	67%	83%	75%	33%	58%	75%	0%	0%	75%	33%	44%	40%		
	Coverage	9.38	12.50	7.29	12.50	8.33	12.50	12.50	8.33	83.33	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	39.56		
Process Sample Sheet	Reqs.	2	46	18	36	24	1	66	2	195	4	4	22	12	0	0	27	8	77	118		
	%age	75%	100%	58%	100%	67%	100%	100%	67%	83%	75%	33%	58%	75%	0%	0%	75%	33%	44%	40%		
	Coverage	9.38	12.50	7.29	12.50	8.33	12.50	12.50	8.33	83.33	9.38	4.17	7.29	9.38	0.00	0.00	9.38	4.17	43.77	39.56		
Sample Result	Reqs.	2	42	16	21	14	0	57	8	160	6	6	35	18	0	0	39	12	116	44		
	%age	75%	100%	50%	100%	67%	0%	100%	67%	70%	75%	33%	67%	75%	0%	0%	75%	33%	45%	25%		
	Coverage	9.38	12.50	6.25	12.50	8.33	0.00	12.50	8.33	69.79	9.38	4.17	8.33	9.38	0.00	0.00	9.38	4.17	44.81	24.98		
TOTAL	Reqs.	2	46	14	15	10	1	66	4	158	2	2	13	6	0	0	15	4	42	116		
	%age	75%	100%	57%	100%	67%	71%	100%	67%	80%	75%	33%	59%	75%	0%	0%	75%	33%	44%	36%		
	Coverage	9.38	12.50	7.14	12.50	8.33	8.93	12.50	8.33	79.61	9.38	4.17	7.44	9.38	0.00	0.00	9.38	4.17	43.92	35.69		

Fig. 9. Coverage of the LiMS security requirements

The comparison illustrates that SREBP method reaches a coverage of almost 80% of coverage in addressing security of the LiMS business assets; whereas the coverage achieved using SQUARE is close to 44%. The differences are mainly due to different target audience of these methods, SREBP facilitates business analysts while SQUARE targets security analysts. Hence, SQUARE independently

addresses the security comprehensively although its integration with business processes require more efforts to elicit security requirements. SREBP is asset-driven and security requirements are specified in details satisfying the majority of their security objectives. Instead, SQUARE focusses on the technology that supports the execution of business processes and represents security requirements at general level. This shifts the priority from acquiring the security objectives towards implementing security controls.

3.4 Football Federation Information Management System

When considering the football federation (FF) case we wanted to understand whether the result received in LIMS can be repeated. It is important to note that the FF case was also executed by two different persons than LiMS (the second author of the paper acted only as the supervisor during this case execution).

The value chain illustrated in Fig. 10 suggested five business assets (i.e., Player, Team, Umpire, Game and Timetable). The further security requirements elicitation was performed using SREBP and SQUARE. As illustrated in Fig. 11 the comparison illustrates that SREBP method reaches a coverage of almost 82 % in addressing security of the FF business assets; whereas the coverage achieved using SQUARE is close to 47 %. This results highly corresponds to the results of the LiMS analysis.

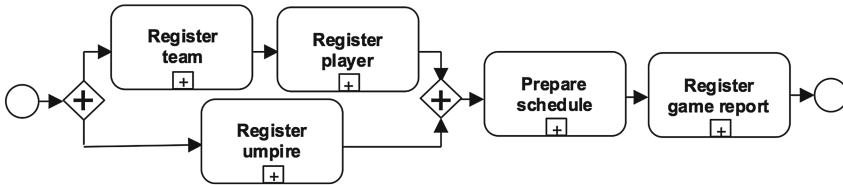


Fig. 10. FF Value chain

3.5 Threats to Validity

The validity of results may be affected by few threats. Firstly, the risk of researchers' familiarity with the concepts (e.g., BPMN, attack trees, UML and etc.) affects the results. The researchers personal interpretation of the problem and constructed models has an impact on the objectivity of results. Therefore, we adapted principles of: *i*) data triangulation [21], where several sources are used to collect data (i.e., threats and vulnerabilities), and this limits the effects of the interpretation of one single data source; *ii*) peer debriefing [21], where peer having different expertise allows avoiding the risk of being biased during the application of both methods. Similarly, peer debriefing helped reducing the risk of construct validity where peer focusing on what the researchers has in mind and that the actual problem has been investigated.

Methods		SREBP										SQUARE										DIFFERENCE
Business Assets	Categories	Identification	Authentication	Authorization	Accounting	Audit	Non-repudiation	Immunity	Data exchange	TOTAL	Identification	Authentication	Authorization	Accounting	Audit	Non-repudiation	Immunity	Data exchange	TOTAL			
		12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	100	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	100			
Player	Reqs.	2	12	12	15	3	1	33	12	90	1	4	5	5	1	0	6	5	27	63		
	%age	75%	100%	83%	75%	75%	75%	100%	67%	81%	25%	58%	58%	92%	8%	0%	75%	50%	46%	38%		
	Coverage	9.38	12.50	10.35	9.38	9.38	9.38	12.50	8.33	81.20	3.13	7.31	7.31	11.47	1.04	0.00	9.38	6.25	45.89	35.31		
Team	Reqs.	2	18	18	21	6	2	44	22	133	1	5	5	5	1	0	6	5	28	105		
	%age	75%	100%	83%	75%	75%	75%	67%	67%	77%	25%	58%	58%	92%	8%	0%	75%	50%	46%	31%		
	Coverage	9.38	12.50	10.35	9.38	9.38	9.38	8.33	8.33	77.03	3.13	7.31	7.31	11.47	1.04	0.00	9.38	6.25	45.89	31.14		
Umpire	Reqs.	2	12	14	15	6	2	51	14	116	1	5	5	5	1	0	6	5	28	88		
	%age	75%	100%	83%	100%	75%	75%	100%	67%	84%	25%	58%	58%	92%	8%	0%	75%	50%	46%	39%		
	Coverage	9.38	12.50	10.35	12.50	9.38	9.38	12.50	8.33	84.32	3.13	7.31	7.31	11.47	1.04	0.00	9.38	6.25	45.89	38.43		
Game	Reqs.	4	24	14	30	3	1	66	24	166	1	4	5	5	0	1	6	4	26	140		
	%age	75%	100%	83%	100%	75%	75%	100%	67%	84%	25%	58%	58%	92%	0%	25%	75%	50%	48%	37%		
	Coverage	9.38	12.50	10.35	9.38	9.38	12.50	8.33	8.33	84.32	3.13	7.31	7.31	11.47	0.00	3.13	9.38	6.25	47.98	36.34		
Timetable	Reqs.	2	20	12	21	3	1	48	18	125	1	4	5	5	2	1	6	4	28	97		
	%age	75%	100%	67%	100%	75%	75%	100%	67%	82%	25%	58%	58%	92%	25%	25%	75%	50%	51%	31%		
	Coverage	9.38	12.50	8.33	12.50	9.38	9.38	12.50	8.33	82.30	3.13	7.31	7.31	11.47	3.13	3.13	9.38	6.25	51.11	31.19		
TOTAL	%age	75%	100%	80%	90%	75%	75%	93%	67%	82%	25%	58%	58%	92%	10%	10%	75%	50%	47%	35%		
	Coverage	9.38	12.50	10.35	10.94	9.38	9.38	11.46	8.33	81.72	3.13	7.31	7.31	11.47	0.78	0.78	9.38	6.25	46.41	35.31		

Fig. 11. Coverage of the FF security requirements

Concerning external validity, the findings are independent of their application and the results are largely overlapping. Potentially, the methods could be generalised and repeated on other case studies. However, different organisations have their specific security goals, which need to be considered separately.

We started by applying SREBP method to reduce learning effects. There are no carry-over effects to SREBP application as participants were not familiar with the process models. However, the SQUARE method benefitted carry-over effects as participants became familiar with the domain. We have adapted similar approach in verifying security requirements, by verifying SREBP requirements first; this avoids any carry-overs to SREBP but SQUARE requirements are verified later therefore any carry-over benefitted SQUARE.

4 Related Work

Fabian et al. [8] conducted a thorough study comparing the security requirements engineering methods. For instance *goal-oriented approaches*, such as Knowledge Acquisition in Automated Specification (KAOS) [12], Secure i^* [7], and Secure Tropos [16], facilitate the requirements elicitation and specification by providing the rationale for a particular requirement. *UML based approaches*, like Misuse cases [25] or SecureUML [13], focus on the system design. In the SREBP method SecureUML is used to define security requirements of *access control* and *data store* contextual areas, UMLsec is applied to create requirements models within *communication channel* and *business service* contextual areas.

Security in business processes is integrated in several ways: security objective elicitation, security requirements modelling, security risk-driven approaches and security requirements conformance checking. In [26] a generic security model specifies security goals, policies, and constraints based on a set of basic entities, attributes, interactions, and effects. In [10] business process elements are used to express the common security requirements. These studies guarantee that security

constraints are not violated by achieving the security goals. However, they do not define graphical notations and do not guide elicitation.

A formal descriptive language [23] is used to derive security requirements that assign security level to business process components. In [22] BPMN is extended with a specific padlock symbols to annotate business processes with early security requirements. Similarly, in [20] two new artefacts – operating condition and control case – are proposed to express the constraints, which help mitigate risk and facilitate the early discovery of security requirements. An annotation language [17] embedded in business process models is proposed to express security requirements as structured text annotations. In comparison to this related work where the focus is placed on representing security requirements (graphically) on the process models, SREBP suggests a novel approach to elicit these requirements and define them as the business rules.

5 Conclusion and Future Work

In this paper, we presented the SREBP method for eliciting security requirements from the business processes. Its strength lies in its general description of security goals and the systematic analysis of the contextual areas. We have defined the application guidelines and compared it to the SQUARE method. The study illustrates that SREBP is rather generalisable to different problems. We could also conclude that the method contributes with a relatively complete (with respect to the security requirements categories) set of security requirements. We also illustrate that the achieved result is rather repeatable in different cases.

As the future work, SREBP has to be strengthened with analyses of threat likelihood, vulnerability and impact levels. This would help prioritise the security requirements and support business analyst in deciding, which security requirements should be implemented in case of limited time, resources, or finances. It is also important to continue the SREBP validation regarding its correctness (i.e., proving that some formally defined criteria are satisfied) and usability (i.e., investigating method acceptance in practical settings).

References

1. Ahmed, N., Matulevičius, R.: A method for eliciting security requirements from the business process models. In: CAiSE Forum and Doctoral Consortium **2014**, 57–64 (2014)
2. Ahmed, N., Matulevičius, R.: Securing business processes using security risk-oriented patterns. *Comput. Stan. Interfaces* **36**(4), 723–733 (2014)
3. Apostolopoulos, G., Peris, V., Saha, D.: Transport layer security: how much does it really cost? In: *Proceedings IEEE INFOCOM 1999 The Conference on Computer Communications*, vol. 2, pp. 717–725 (1999)
4. Atluri, V., Warner, J.: Security for workflow systems. In: Gertz, M., Jajodia, S. (eds.) *Handbook of Database Security*, pp. 213–230. Springer, US (2008)
5. Chang, R.: Defending against flooding-based distributed denial-of-service attacks: a tutorial. *Commun. Magazine, IEEE* **40**(10), 42–51 (2002)

6. Clarke, J., Fowler, K., Oftedal, E., Alvarez, R.M., Hartley, D., Kornbrust, A., O'Leary-Steele, G., Revelli, A., Siddharth, S., Slaviero, M.: *SQL Injection Attacks and Defense*, 2nd edn. Syngress Publishing, Burlington (2012)
7. Elahi, G., Yu, E.: A goal oriented approach for modeling and analyzing security trade-offs. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) *ER 2007*. LNCS, vol. 4801, pp. 375–390. Springer, Heidelberg (2007)
8. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. *Requirements Eng.* **15**(1), 7–40 (2010)
9. Firesmith, D.G.: Engineering security requirements. *J. Object Technol.* **2**(1), 53–68 (2003)
10. Herrmann, P., Herrmann, G.: Security requirement analysis of business processes. *Electronic Commerce Research* **6**(3–4), 305–335 (2006)
11. Hummer, W., Gaubatz, P., Strembeck, M., Zdun, U., Dustdar, S.: Enforcement of entailment constraints in distributed service-based business processes. *Inf. Softw. Technol.* **55**(11), 1884–1903 (2013)
12. van Lamsweerde, A.: Engineering requirements for system reliability and security. In: Broy, M., Grunbauer, J., Hoare, C.A.R. (eds.) *Software System Reliability and Security*, vol. 9, pp. 196–238. IOS Press, Amsterdam (2007)
13. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: a UML-based modeling language for model-driven security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) *UML 2002*. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
14. Mead, N.: identifying security requirements using the security quality requirements engineering (SQUARE) method. In: Mouratidis, H., Giorgini, P. (eds.) *Integrating Security and Software Engineering*, pp. 44–69. Idea Publishing Group, Hershey (2006)
15. Menzel, M., Thomas, I., Meinel, C.: Security requirements specification in service-oriented business process management. In: *ARES*, pp. 41–48 (2009)
16. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. *Int. J. Soft. Eng. Knowl. Eng.* **17**(02), 285–309 (2007)
17. Müllle, J., von Stackelberg, S., Bohm, K.: Modelling and transforming security constraints in privacy-aware business processes. In: *SOCA*, pp. 1–4 (2011)
18. Natan, R.B.: *Implementing Database Security and Auditing: Includes Examples for Oracle, SQL Server, DB2 UDB Sybase*. Digital Press, Newton (2005)
19. Park, J., Sandhu, R.: The UCON-ABC usage control model. *ACM Trans. Inf. Syst. Secur.* **7**(1), 128–174 (2004)
20. Pavlovski, C.J., Zou, J.: Non-functional requirements in business process modeling. In: *APCCM*, pp. 103–112. Australian Computer Society, Inc. (2008)
21. Robson, C.: *RealWorld Research - A Resource for Social Scientists and Practitioners-Researchers*. Blackwell Publishing, Oxford (2002)
22. Rodríguez, A., Fernández, M.E., Piattini, M.: A BPMN extension for the modeling of security requirements in business processes. *IEICE-TIS* **E90-D**(4), 745–752 (2007)
23. Röhrig, S., Knorr, K.: Security analysis of electronic business processes. *Electron. Commer. Res.* **4**(1–2), 59–81 (2004)
24. Schumacher, M., Fernandez, E.B., Hybertson, D., Buschmann, F., Sommerlad, P.: *Security Patterns: Integrating Security and Systems Engineering*. Wiley, New York (2006)
25. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. *Requirements Eng.* **10**(1), 34–44 (2005)
26. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. *JSA.* **55**(4), 211–223 (2009)