A Coarse-Grained Comparison of BPMN Extensions for Security Requirements Modelling

Edgars Gaidels[1], Andrejs Gaidukovs[1] and
Raimundas Matulevičius[2]

[1] Institute of Applied Computer Systems, Riga Technical University, Latvia
{edgars.gaidels, andrejs.gaidukovs}@edu.rtu.lv
[2] Institute of Computer Science, University of Tartu, Estonia
rma@ut.ee

Abstract. Business Process Modelling Notation (BPMN) is the de-facto standard notation for representing business processes. At the same time BPMN lacks the proper security representation. Therefore, various extensions to express security concerns are proposed. In this paper we report on the comparison of two BPMN extensions to security requirements modelling. The approaches are mapped to taxonomy of the security requirements in order to explain how well they cover the security requirements categories.

Keywords: Business Process Modelling Notation (BPMN), Security Extensions, Security Requirements

CEUR-WS.org/Vol-2218/paper17.pdf

1 Introduction

Business process modelling languages (e.g., BPMN), does not allow security modelling by default. A native BPMN syntax [9] is insufficient for the representation of security concerns in a clear and common way. Some security-related details may be presented using the artefacts such as Data Objects, Groups or standard Text Annotation element. However, this way of representing security information lacks clarity and precision, restricting the usability and perception of the security models [11].

Some entities (e.g., business asset, IS asset, threat, threat agent and attack method) of the security risk management domain [1, 7] could be expressed using native BPMN syntax. However, when it comes to the modelling of the security requirements, the BPMN language is rather limited and additional extensions need to be developed. In this paper we compare two extensions of the BPMN to model security requirements: an extension for security requirements modelling in business processes [11] and SecBPMN [13]. The research goal is to explain how well these extensions are able to capture various categories of security requirements.

The paper is structured as follows. In Section 2 we overview existing BPMN extensions towards security and privacy modelling. Section 3 presents the taxonomy of the security requirements. This taxonomy is used to understand and to compare the coverage of the BPMN extensions (presented in Section 4) to express security requirements. The comparison results are given in Section 5. The paper concludes in Section 6.

2 BPMN Security Extensions

There exist a number of extensions of the BPMN to addresses various aspects of security or related aspects. For instance, in [6] considers how security BPMN could be used to manage risks factors. In [8], the BPMN constructs are annotated to capture and model trust. Elsewhere, in [3] the BPMN is enriched with the notations to capture information assurance and security modelling capabilities and in [15] the concept of compliance is introduced to restrict certain modifications behaviour of the business process. Another proposal to support process compliance is done in [2]; here authors introduce access control, separation of duty, binding of duty and need to know principles to the business process modelling.

In [1] the language extensions are prosed to manage security risks and to elicit security requirements. However this approach does not explicitly define how these requirements should be model along the business process. In [10] authors propose the extensions to capture privacy requirements and to reason about the selection of controls (i.e., privacy enhanced technology) to implement these requirements.

In this paper we specifically analyse the security requirements modelling and the extensions proposed to model these requirements using BPMN. In Section 4 an overview of extension for security requirements modelling in business processes [11] and SecBPMN [4] is given. But first we discuss taxonomy for the security requirements.

3 Security Requirements Taxonomy

A comprehensive taxonomy for security requirements is presented in [4]. Twelve categories of security requirements are defined as illustrated in Fig. 1. Hence, the *identification requirements* define the extent to which a system identifies its externals before interacting with them. The *authentication requirements* explain the extent to which a system shall verify its external before interacting with them, and *authorisation requirements* specify the access and usage of privileges of authenticated users.

The *immunity requirements* explain the extent to which a system shall protect itself from infection by unauthorised undesirable programs. The integrity requirements define the extent to which a system shall ensure that it is not intentionally corrupted via unauthorised creation, modification or deletion. The *intrusion detection requirements* consider the extent to which a system shall detect and record the attempts to access or to modify the system itself or the managed data. The *privacy requirements* explain the extent to which a system keeps its sensitive data and communications private from the unauthorised persons or programs [4].

The *non-repudiation requirements* define the extent to which a system shall prevent a party to one of its interactions from denying having participated in all or part of the interaction. The *security auditing requirements* enable security personnel to audit the status and use of the security mechanism. The *survivability requirements* consider the extent to which a system shall survive the intentional destruction. The *physical protection requirements* characterise the extent to which a system protects itself from the physical damage. Finally, the system maintenance security requirements define the extent to which a system shall prevent authorised modifications from accidentally defeating the security mechanisms [4].

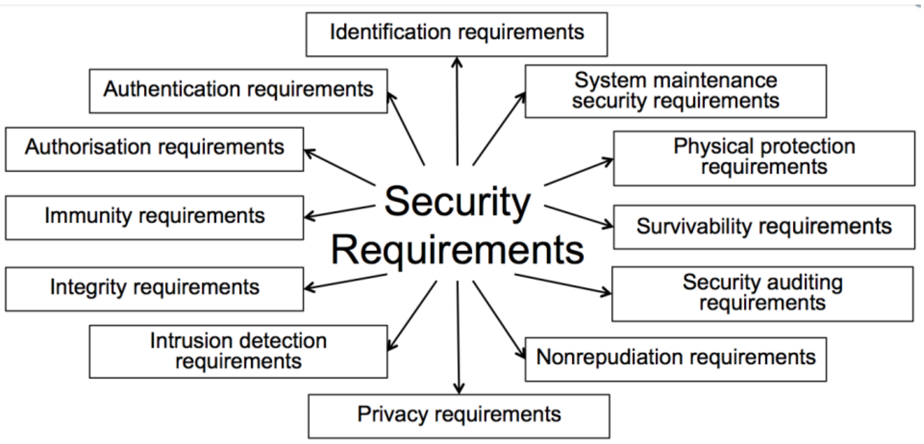

Identification requirements
System maintenance security requirements
Authentication requirements
Authorisation requirements
Physical protection requirements
Immunity requirements
Security Requirements
Survivability requirements
Integrity requirements
Security auditing requirements
Intrusion detection requirements
Nonrepudiation requirements
Privacy requirements

Fig. 1. Taxonomy of Security Requirements, adapted from [4]

In Section 4 we will use these security requirements categories and will map the security extensions of the BPMN language. But first, in Section 3 we present two BPMN extensions to security requirements modelling.

4 Description of BPMN Security Extensions

4.1 Security Requirements Modelling in Business Processes

In [11] Rodríguez et al. have proposed a security extension to the BPMN business process diagram. It targets the core elements, such as *pool*, *lane*, *activity* and *message flow*. The proposed extensions incorporate security into business process models from the business analyst’s perspective. The authors’ main concern is placed on the neglecting security requirement representation in the early design stage, leading to the incorporating of the security concerns at the later implementation or maintenance stages. Thus, the security concerns are incorporated at the analysis stage. The major concerns target security requirements such as *non-repudiation, attack harm detection, integrity*, *privacy* and *access control* as illustrated in Fig. 2.

Each security requirement in the *Secure Business Process Diagram* (SBPD) is depicted as a padlock symbol with the corresponding abbreviation in the middle. Similarly, in [12] authors propose security extensions for the UML 2.0; the rationale behind the selection of the padlock symbol is explained by the strong association between this symbol and the notion of security. Table 1 gives the overview of the concrete syntax of the security extensions.

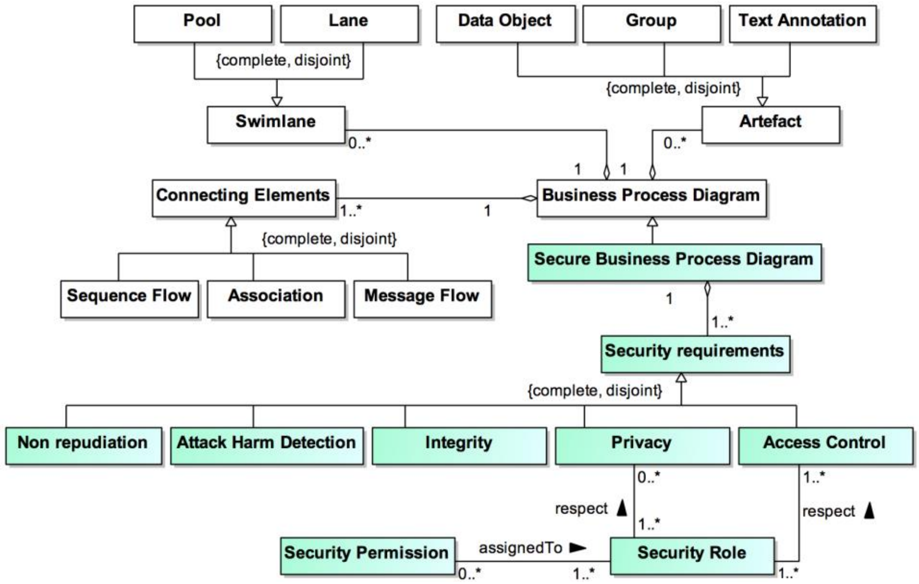


Fig. 2. Extension of the Business Process Diagram Abstract Syntax with the Security Requirement Concepts (adapted from [11])

Each entity from above marked with a letter χ could be additionally tagged with a specific value. For the protection degree, the χ could be replaced by *l* for *low*, *m* for *medium*, or *h* for *high*. For the privacy type, the letter χ can be replaced by *a* for *anonymity, c* for *confidentiality* or can be omitted. If *privacy type* is not specified, then both anonymity and confidentiality, are considered. Therefore, the extension instantly provides an ability to mark the already existing model objects with variety of security requirements, making the model in some extent more security-oriented in quite a simple manner. This means that several rules should be considered, as security elements are not universal, and are applicable specifically to the certain model elements as listed in Table 2.

4.2 SecBPMN

In [13] a SecBPMN framework is proposed to model and to verify business processes against compliance rules. SecBPMN uses security polices to ensure its functionality. This process is divided into two steps: (*i*) modelling, where a business process model is annotated with security concerns; (*ii*) verification, where the anotated BPMN model is verified using BPMN-Q language. BPMN-Q uses graphical queries that allow model checking against defined patterns. BPMN-Q uses relationships for designing queries. BPMN-Q queries can check and verify data object states, paths and flows of BPMN model.

Table 1. Concrete Syntax of the Security Extension (adapted from [11])

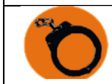
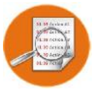
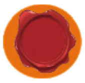
Notation	Name	Description	Tagged values
🔒	Security Requirement	Abstract class containing security requirements specifications. Each security requirement type must be indicated in some of its subclasses.	-
NR	Non-repudiation	It establishes the need to avoid the denial of any aspect of the interaction.	Auditing Values (comment);
AD	Attack Harm Detection	It indicates the degree to which the attempt or success of attacks or damages is detected, registered and notified.	Auditing Values (comment);
Ix	Integrity	It established the degree of protection of intentional and non-authorized corruption.	Auditing Values (comment); Protection Degree.
Px	Privacy	It indicates the degree to which non-authorized parts are avoided to obtain sensitive information.	Auditing Values (comment); Privacy type.
AC	Access Control	It establishes the need to define and/or intensify the access control mechanisms (identification, authentication and authorization) to restrict access to certain components in a business process diagram.	Auditing Values (comment);

Table 2. New security elements annotated to the business process diagram concepts (adapted from [11])

	Business Process Diagrams elements					
Security Elements	Pool	Lane	Group	Activity	Message Flow	Data object
Non-repudiation					X	
Attack harm detection	X	X	X	X	X	X
Integrity					X	X
Privacy	X	X	X			
Access control	X	X	X	X		
Security role	X	X	X			
Security Permissions				X	X	X

In this paper we are considering only the first step where security extensions – SecBPMN – are introduced. Each SecBPMN security annotation is always linked to one standard BPMN element, e.g., *data object*, *activity* or *message flow*. And then the annotation is formalized using one or more predicates [13]. Proposed security annotations elements with predicates are described in Table 3. Predicates have rather meaningful naming and it is understandable what particular goal is covered by the element.

Table 3. Security annotations (concrete syntax) of SecBPMN (adapted from [13])

Graphical syntax	Predicates
	AccountabilityAct (a: Activity, enfBy: {SecMechanisms}, monitored: {Users})
	AuditabilityAct (a: Activity, enfBy: {SecMechanisms}, frequency: Time) AuditabilityDO(do: DataObject, enfBy: {SecMechanisms}, frequency: Time) AuditabilityMF (mf: MessageFlow, enfBy: {SecMechanisms}, frequency: Time)
	AuthenticityAct (a: Activity, enfBy: {SecMechanisms}, ident: Bool, auth: Bool, trustValue: Float) AuthenticityDO (do: DataObject, enfBy: {SecMechanisms})

24	AvailabilityAct (a: Activity, enfBy: {SecMechanisms}, level: Float) AvailabilityDO(do: DataObject, enfBy: {SecMechanisms}, authUsers: {Users}, level: Float) AvailabilityMF (mf: MessageFlow, enfBy: {SecMechanisms}, level: Float)
🤝	ConfidentialityDO(do: DataObject, enfBy: {SecMechanisms}, readers: {Users},writers: {Users}) ConfidentialityMF (mf: MessageFlow, enfBy: {SecMechanisms}, readers: {Users},writers: {Users})
📄	IntegrityAct (a: Activity, enfBy: {SecMechanisms}, personnel: Bool, hardware: Bool, software: Bool) IntegrityDO (do: DataObject, enfBy: {SecMechanisms}) IntegrityMF (mf: MessageFlow, enfBy: {SecMechanisms})
⇏	NonRepudAct (a: Activity, enfBy: {SecMechanisms}, execution: Bool) NonRepudMF (mf: MessageFlow, enfBy: {SecMechanisms}, execution: Bool)
🔑	PrivacyAct (a: Activity, enfBy: {SecMechanisms}, sensitiveInfo: {Info}) PrivacyDO (do: DataObject, enfBy: {SecMechanisms}, sensitiveInfo: {Info})

Description of security goals predicates and explanation of methods and parameters:

- *Accountability* linked to activities, and shows that user performs current activity must be monitored; Includes only one predicate AccountabilityAct. Predicate has three parameters [13]:
  - activity – during execution of current activity must satisfy security aspect linked to this type of annotation,
  - enfBy - security mechanisms needed to satisfy security goal requirements for the activity,
  - monitored users.
- *Auditability* includes three predicates [13]:
  - AuditabilityAct means that all users manipulations must be tracked and stored during executions of current activity;
  - AuditabilityDO means that all users manipulations must be tracked and stored when CRUD (create, read, update, delete) operations performed on data objects;
  - AuditabilityMF means that all users manipulations needed for message flow communication (send , receive) must be tracked and stored;
- *Authenticity* includes two predicates, each predicate is linked to two specific type of BPMN elements [13]:

  - AuthenticityAct – linked to BPMN activity element. Means that identity of user must be verified before performing activity. This predicate uses following parameters:
    - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
    - trustValue - is the minimum set of access rights (or trust level) that user must have to perform current operation;
    - ident defines is anonymous user is allowed to perform current activity,
    - auth defines if user must be authorized.
  - AuthenticityDo includes two parameters:
    - do data object should prove that data object is genuine: object not changed by unauthorized users and every modification made to this object must be related to concrete user;
    - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
- *Availability* includes three predicates; each predicate is linked to two specific type of BPMN elements. It includes three predicates [13]:
  - AvailabilityAct means that current activity must be performed every time when activity included in business process;
  - AvailabilityDO means current data object do always available when authorized user is accessing this data object;
  - AvaliabilityMF means that message flow mf always available for communication;
- *Confidentiality* includes two predicates [13]:
  - ConfidentialityDO means that only authorized users are allowed to access current data object do. Includes following parameters:
    - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
    - readers – users with access to read from this data object;
    - writers – users with access to write to this data object;
  - ConfidentialityMF means that only authorizes allowed to use current message flow mf for communication
    - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
    - readers – users with access to receive from this message flow;
    - writers – users with access to send to this message flow;
- *Integrity* includes three predicates [13]:
  - IntegrityAct means that actions performed during this activity should be protected from corruption; also actor (person, hardware or software system) who perform operation must be protected from intentional corruption;
  - IntegrityDO means that current data object do should be protected from intentional corruption;

    - IntegrityMF means that every message mf should be protected from intentional corruption.
- *Non-repudiation* includes two predicates [13]:
    - NonRepudiationAct means that execution of activity must be proved; Includes two parameter:
        - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
        - execution – If is equal to true then activity execution must be verified, and if equal to false then non-execution must be verified;
    - NonRepudiationM means that usage of message must be verified. Includes two parameter:
        - enfBy - security mechanisms needed to satisfy security goal requirements for current activity;
        - execution – If is equal to true then message flow execution must be verified, and if equal to false then non-execution must be verified;
- *Privacy* includes two predicates [13]:
    - privacyACT means that users own data must be controlled by user when performing current activity;
    - privacyDO means that users own data must be controlled by user when using current data object.

5 Comparison

To compare both the SBPD [11] and SecBPMN [13] we check how they address different categories of the security taxonomy [4] presented in Section 2. Comparison results are listed in Table 4.

Firstly we note that three types of security requirements – *survivability*, *physical protection* and *system maintenance security* requirements – are captured neither in SBPD nor in SecBPMN. The *immunity* requirements are not considered in the SBPD approach; and the *identification* and *intrusion detection* requirements are not considered in SecBPMN.

In the SBPD approach, *integrity*, *privacy* and *non-repudiation* requirements are fully covered by introducing the corresponding extensions. The *intrusion detection* requirements are addressed by the attack harm detection extensions; although they have different naming, but their semantic correspondence is rather closely related. The *security auditing* requirements does not have the dedicated extension, but they are addressed by the auditing values introduced to other security extension types. Finally, as presented in Table 1 the access control extension covers both the *identification*, *authentication*, and *authorisation* requirements.

In the SecBPMN approach, the authentication, integrity, privacy and non-repudiation security requirements are covered by the corresponding security annotations. In addition, the SecBPMN auditability and accountability can be used to capture various aspects of the *security auditing* requirements. The *immunity* requirements could be partially addressed by the availability annotations taking into account that the executed business activities should be available (thus should have an immunity to security risks). Similarly, the availability annotations are used to define *authorisation* requirements to accessed data. Definition of the *authorisation* requirements is strengthened by considering confidentiality in SecBPMN.

Table 4. Comparison results

Security requirements	**SBPD**	**SecBPMN**
Identification	Access control	-
Authentication		Authenticity
Authorisation		Confidentiality Availability (*partially, regarding data access*)
Immunity	-	Availability (*partially, regarding activity execution*)
Integrity	Integrity	Integrity
Intrusion detection	Attack harm detection	-
Privacy	Privacy	Privacy
Non-repudiation	Non-repudiation	Non-repudiation
Security auditing	As auditing values of the security requirements	Auditability Accountability
Survivability	-	-
Physical protection	-	-
System maintenance security	-	-

6 Conclusion

In this paper we provide a coarse grained comparison of two security extensions to model security requirements. We consider how well they cover the taxonomy of the security requirements. An extension for security requirements modelling in business processes [11] broadens the possibilities of the BPMN in terms of describing security needs, however it is just a fraction of the potential security representations. As observed in [14] the extended concepts lack the ability to specify response procedures when a security function fails. It just leaves the model with the certain expectations towards the requirements, and it is challenging to find the application of the SBPD for describing more complex security scenarios. SecBPMN uses BPMN annotations with additional graphic elements and provides means for the security model verification. The weakness of the SecBPMN method is that it becomes overloaded with too many graphical elements, thus it becomes difficult to read and understand.

The comparison of the approaches to the security requirements taxonomy shows that they support modelling of some major concerns of the *software* security (e. g., privacy, non-repudiation, integrity, etc.). However their support for expressing physical security and security maintenance requirements is limited. It is also important to note that we did not consider the fine-grained analysis. It could potentially uncover limitations of these approaches to address important security concerns (e.g., reasoning for privacy enhanced technologies [10], data by minimization [5], etc). This analysis remains for the future work.

Acknowledgement. This research has been supported by the Estonian Research Council (grant IUT20-55).

References

1. Altuhhova, O., Matulevičius, R., Ahmed, N.: An extension of business process model and notification for security risk management. Int. J. Inform. Syst. Model. Design (IJISMD) 4(4), 93–113 (2013)
2. Brucker, A., Hang, I., Lückemeyer, G., Ruparel, R.: SecureBPMN: modeling and enforcing access control requirements in business processes. In: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, pp. 123–126. ACM, New York (2012)
3. Cherdantseva, Y., Hilton, J., Rana, O.: Towards SecureBPMN: aligning BPMN with the information assurance and security domain. In: Business Process Model and Notation. LNBIP, pp. 107–115. Springer, Heidelberg (2012)
4. Firesmith, D.G.: Engineering security requirements. J. Object Technol. 2(1), 53–68 (2003)
5. Fung B. C. M., Wang K., Chen R., and Yu P. S.. Privacy- preserving data publishing: A survey of recent developments. ACM Comput. Surv., 42(4):14:1–14:53, June 2010.
6. Marcinkowski, B., Kuciapski, M.: A business process modeling notation extension for risk handling. In: 11th International Conference on Information Systems and Industrial Management. LNCS, pp. 374–381. Springer, Heidelberg (2012)
7. Matulevičius R., Fundamentals of Secure System Modelling. Springer 2017, pp. 207
8. Menzel, M., Thomas, I., Meinel, C.: Security requirements specification in service-oriented business process management. In: International Conference on Availability, Reliability and Security (ARES 2009), pp. 41–49 (2009)
9. OMG, BPMN Specification - Business Process Model and Notation, http://www.bpmn.org/.
10. Pullonen P., Matulevičius R., Bogdanov D. (2017) PE-BPMN: Privacy-Enhanced Business Process Model and Notation. In: Carmona J., Engels G., Kumar A. (eds) Business Process Management. BPM 2017. Lecture Notes in Computer Science, vol 10445. Springer, Cham

11. Rodríguez, A., Fernández-Medina, E., Piattini, M.: A BPMN extension for the modelling of security requirements in business processes. IEICE Trans. Inf. Syst. E90–D, 745–752 (2007)
12. Rodríguez, A., Fernández-Medina, E., Trujillo, J., Piattini, M.: Secure business process model specification through a UML 2.0 activity diagram profile. Decis. Support Syst. 51, 446–465 (2011).
13. Salnitri, M., Dalpiaz, F., Giorgini, P.: Modelling and verifying security policies in business processes. Lect. Notes Bus. Inf. Process. 175 LNBIP, 200–214 (2014)
14. Sang, K.S., Zhou, B.: BPMN security extensions for healthcare process. Proc. - 15th IEEE Int. Conf. Comput. Inf. Technol. CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Secur. Comput. DASC 2015 13th IEEE Int. Conf. Pervasive Intell. Comput. PICom 2015. 2340–2345 (2015)
15. Schleicher, D., Leymann, F., Schumm, D., Weidmann, M.: Compliance scopes: Extending the BPMN 2.0 Meta model to specify compliance requirements. In: IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp. 1–8. IEEE (2010)