Refining Security Requirement Elicitation from Business Processes using Method Engineering

Kurt Sandkuhl[1], Raimundas Matulevičius[2], Naved Ahmed[2], and Marite Kirikova[3]

[1] University of Rostock, Institute of Computer Science
Chair Business Information Systems, Albert-Einstein-Str. 22, 18059 Rostock, Germany
Kurt.Sandkuhl@uni-rostock.de
[1]Institute of Computer Science, University of Tartu, Estonia
{rma, naved}@ut.ee
[3]Riga Technical University, Riga, Latvia
marite.kirikova@cs.rtu.lv

Abstract. A method defines a systematic process for problem solving including the required aids and resources. The transfer of method knowledge from the developers to other users requires a certain level of maturity and documentation of the method. Based on a method for security requirements elicitation from business processes (SREBP), we demonstrate how approaches from method engineering can be used to refine methods and improve transferability and maturity of method descriptions. The contributions of the paper are (1) to show how a component-based method view can be applied in method refinement, (2) the actual refinement process for SREBP integrating work procedure, cooperation principles and notation, and (3) initial experiences and lessons learned from refining SREBP.

Keywords: Security Requirements Elicitation, Method Engineering, Business Process Models, Method Knowledge Transfer.

1 Introduction

Security engineering plays an important role in lowering the risk of intentional harm to valuable assets (such as preventing and reacting to malicious harm, misuse, threats and security risks) [10]. Although the importance of introducing security engineering practices early in the development cycle has been acknowledged [13, 20], it is commonly overlooked when working with business process management. The reason is that while business analysts are expert in their domains, they have limited knowledge about the security domain [16].

There are several studies, which tries to target the above problem by enforcing security mechanisms. For instance, the UMLsec approach [12] introduces stereotypes to define secure systems from business processes expressed in activity diagrams. Elsewhere security extensions to the BPMN language are proposed to define access control, separation of duties and similar constraints [8], or to check business process compliance [18]. Although these (and similar) studies focus on (*i*) representing

Copyright © 2015 by the authors. Copying permitted for private and academic purposes.
This volume is published and copyrighted by its editors.

security aspects graphically or enforcing security mechanism to developed system, they are limited to provide the rationale for security requirements.

A method for security requirements elicitation from business processes (SREBP) is suggested in order to support identification of the security criteria and the guided derivation of the security requirements from the business processes [1] [4]. The method consists of two major stages. Firstly, it describes how to identify business assets and determine their security objectives. Secondly, it supports elicitation of security requirements from business process models that are captured at a level of granularity where data objects, resources and data flows are modelled [21].

The scope of this paper is to report on experiences and lessons learned from refining the SREBP method with concepts from method engineering. In general terms, a method defines a systematic process for problem solving, which encompasses the required aids and resources. Many technology and engineering disciplines use methods as a way to capture proven practices and to provide guidance for specific tasks. In computer science and business information systems, methods also address the development processes of various kinds of models, e.g. business process models or enterprise models, and the analysis and transformation of models. Development of methods is a complex process as methods should be grounded in solid experiences, and iteratively refined in many application cases in order to reach a sufficient maturity level. In particular when method knowledge is transferred from the developers of a method to new method users, both the method documentation and the method as such need to have a high level of maturity and detail.

Using a component-based method view proposed by Goldkuhl *et al.* [11], we report on the process of refining SREBP in order to ease the transfer of method knowledge to new SREBP users. The contributions of the paper are (1) to show how a component-based method view can be applied in method refinement, (2) the actual refinement process for SREBP integrating work procedure, cooperation principles and notation, and (3) initial experiences and lessons learned from refining SREBP.

The paper is structured as follows: Section 2 discusses the experienced challenges when transferring knowledge about SREBP. In Section 3 we introduce the SREBP method and the principles of the method engineering. In Section 4 we present the refinement of SREBP following the method development process. Section 5 discusses the observations. Finally conclusions and future work are given in Section 6.

2 Application Scenario

Work presented in this paper is based on the project "Improvement of IT-Security in Enterprises based on Process Analysis and Risk Patterns (ITSE)" which has university partners from three different countries: Estonia, Latvia, and Germany. The main goal of the project is to transfer the SREBP method to the practice of small and medium-sized enterprises (SME), including the creation of a set of guidelines. To illustrate its usefulness and completeness, SREBP will be used by all three university-partners in their regions. For this purpose, knowledge how to use the SREBP method has to be transferred from the SREBP developers (University of Tartu) to the other two university partners. The process of knowledge transfer was jointly designed by all partners and includes several steps:

1. Scientific publications about SREBP are provided to offer first information about the general idea and basic concepts;
2. A tutorial is developed by Tartu University encompassing not only SREBP as such but basic concepts from IT security and security engineering;
3. A joint workshop is organized which consists of teaching the actual tutorial, discussing the scientific publications and selecting pilot cases to apply SREBP;
4. During the pilot cases, the SREBP developers serve as coaches for the method users; based on the results from workshop and pilot cases, SREBP is refined and documentation is enhanced;
5. The refined SREBP version serves as basis for eliciting IT security requirements in actual SME cases.

When writing this paper, step 4 was still on going and step 5 was under preparation. However, the workshop on SREBP (step 3) resulted in some lessons learned and requirements to be taken into account during method improvement:

- Terms and concepts used in SREBP need to be documented in more detail to avoid ambiguity and misunderstandings. An example is the term “business asset”, which from an IT security perspective includes any resource or information to be protected. Whereas from an economics perspective only those resources are considered, which can be valued and reflected in the balance sheets.
- The prerequisites for using SREBP need to be made explicit. This includes both the competences that users should have (e.g., knowledge about IT security and process modelling) and the information, which should be included in the process models (e.g., resources and IT components accessed, processes to be represented).
- Whether or not the SREBP activities have to be performed in always the same order. What other different sequences make sense. An example is whether always the general view on value creation areas is needed first or SREBP can immediately start by analysing a specific process.
- SREBP uses four “business perspectives” which are also part of many existing enterprise or process modelling methods. But at closer inspection these are different from other approaches. Thus they need detailed explanation.

3 Background

This section summarises the background for our work, which consists of SREBP and its basic features. It also introduces relevant work from method engineering.

3.1 SREBP

As illustrated in Fig. 1, the SREBP method [1][4] consists of two stages: (*i*) business asset identification and security objective determination and (*ii*) security requirements elicitation. The main object of analysis during the SEBP application is the business process defined using the value chain diagrams (which visualise how the enterprise business functions are related in order achieve enterprise's goals) and business processes diagrams (where data objects, resources and data flows are modelled).

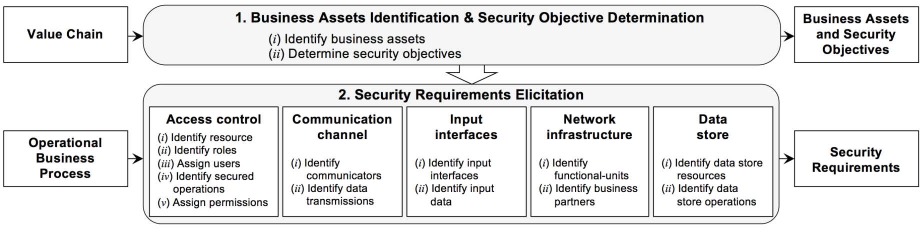
Value Chain
1. Business Assets Identification & Security Objective Determination
(i) Identify business assets
(ii) Determine security objectives
Business Assets and Security Objectives
2. Security Requirements Elicitation
Operational Business Process
Access control
(i) Identify resource
(ii) Identify roles
(iii) Assign users
(iv) Identify secured operations
(v) Assign permissions
Communication channel
(i) Identify communicators
(ii) Identify data transmissions
Input interfaces
(i) Identify input interfaces
(ii) Identify input data
Network infrastructure
(i) Identify functional-units
(ii) Identify business partners
Data store
(i) Identify data store resources
(ii) Identify data store operations
Security Requirements

Fig. 1. The SREBP method [1][4]

Business Assets Identification & Security Objectives Determination. The first stage starts with the analysis of the value chain 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 problem domain. It consists of two activities:

(*i*) *Identify business assets*: During this activity the central artefact (or artefacts) considered in the value chain is identified. 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.

(*ii*) *Determine security objectives*: The activity addresses determining of key security objectives – confidentiality, integrity and availability – for identified business assets. Here confidentiality describes a property of being made disclosed to unauthorised individuals, entities or processes. Integrity is a property of safeguarding the accuracy and completeness of the business asset. And availability describes the property of being accessible and usable upon demand by an authorised entity.

Security requirements elicitation. In [3], five security risk-oriented patterns are defined to derive security requirements. These patterns are based on the domain model for Information Systems Security Risk Management (ISSRM) [9] that supports the definitions of security concepts for asset-related concepts, risk-related concepts and risk treatment-related concepts. The patterns are used within five contextual areas (one pattern in each area), such as access control, communication channel, input interfaces, network infrastructure, and data store.

Application of the pattern within each contextual area consists of three steps. The first step is *pattern identification* in business process diagram. Pattern identification potentially could be performed using hierarchical level matching, business perspective matching, structural similarity and semantic similarity methods [2]. Once the pattern occurrences are identified in the business process model, the second step – *security model extraction* – is performed. The second step is performed following activities, which are different within the contextual area for each pattern. For example (see Fig. 1), to create a security model within the access control contextual area, one needs to (*i*) identify resource, (*ii*) identify roles, (*iii*) assign users, (*iv*) identify secured operations, and (*v*) assign permissions. The third step of pattern application is *security* *requirements derivation* from the security model. Typically, here the security requirements express a condition that needs to be made try by installing security countermeasures.

3.2 Method Engineering

The research area of method engineering offers a rich body of knowledge how to systematically develop, introduce and adapt "methods". Methods often are considered as prescriptive since they are supposed to provide guidance for problem solving or for performing complex tasks. This requires that a method would include *what activities* to perform, *how* to perform them (*procedure*), *what* results (*artefacts*) to develop, and *how* to capture these results (*notation*) [11]. All methods build on perspectives, values, principles, and categories (with definitions), which are expressed in the method and its elements and which show its underlying theories and rationality.

Different conceptualizations of the term "method" and related terms have been proposed. If there is a close link between procedure, notation, and concepts, the term method component is used [11][17]. The concept of method component is similar to the concept of method chunk [15] and [14] and the notion of method fragment [7]. Methods often consist of an integrated set of several method components, referred as methodology [6]. Then the components together form a structure called a framework.

For the purpose of refining SREBP, a component-based method view is considered favourable since SREBP includes several activities, which potentially can be performed in different order or even are optional in certain situations. The method conceptualisation proposed by Goldkuhl *et al.* [11] offer a component-based view and is selected. It states that a comprehensive method description should describe the perspective, framework, cooperation principles and all method components. Fig. 1 illustrates how these elements of the method conceptualisation are related:

- *Method components*: a method component should consist of concepts, procedure and notation. The *concepts* specify what aspects of reality are regarded as relevant in the modelling process, i.e. what is important and what should be captured a model. These relevant concepts should be named in the method component and explained if necessary. The *procedure* describes how to identify the relevant concepts in a method component. It may also cover prerequisites and resources. The *notation* specifies how the result of the procedure should be documented. As a rule, this must provide expressions for each concept and for the relationships between them.
- *Framework*: a method framework describes the relationships between the individual method components, i.e., which components are to be used and under what conditions, as well as the sequence of the method components (if any).
- *Forms of cooperation*: modelling tasks require a range of specialist skills or cooperation between different roles. These necessary skills and roles must be described, along with the division of responsibilities between the roles and the form of cooperation. The cooperation also includes responsibility assignment for the tasks or for method component, and organisation of collaboration.

- *Perspective*: every method describes the procedure for the modelling process from a particular perspective, which influences what is considered important when developing a model. This perspective often is related to the aims and purpose of the method.

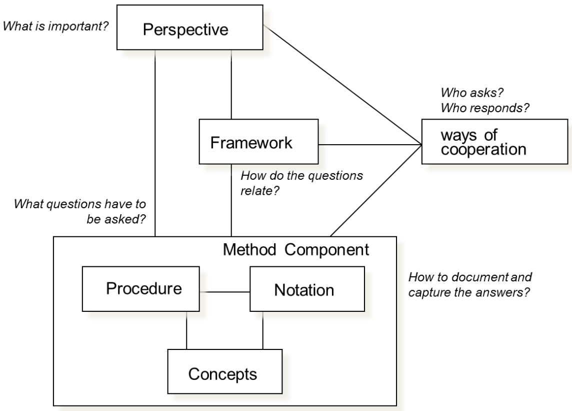

Fig. 1. Method components according to Goldkuhl *et al*. [11]

4 Refinement of SREBP

The case discussed in Section 2 showed that transferring knowledge about SREBP could be eased by making improvements in the method or its documentation. This section first performs an analysis of SREBP using Goldkuhl's conceptualization and then identifies what refinements of SREBP are made based on the analysis results.

4.1 Analysis of SREBP

The approach used in this section for analysing SREBP with respect to potential refinements is to check (a) whether the elements identified by Goldkuhl are defined for SREBP and (b) where the different elements were documented and if the documentation was detailed enough. Observations are listed in Table 1.

Table 1: Method elements and their status in SREBP

Method element		**SREBP status**
Perspective		Defined in scientific publications
Framework		So far, only the main process is defined in the tutorial but without alternative sequences
Method component		Not explicitly defined; potential components clearly visible in the tutorial
	Procedure	Defined in tutorial
	Concepts	Defined in several scientific publications
	Notation	Examples included in the tutorial
Cooperation Principles		Known by the method developers, but not explicitly defined

The first observation made during SREBP analysis is that there is no single SREBP document, which would contain all elements recommended by Goldkuhl *et al*. in an integrated way. This is not really surprising because SREBP was not developed with this method conceptualization in mind. The *perspective* of SREBP is explicitly defined in several publications and tutorial. Hence the focus on IT security requirements elicitation from business processes is made clear.

The framework defining the way of using method components in combination with each other and potential alternative sequences so far only is implicitly defined. In all examples showing the use of SREBP and in the overall description, the typical flow of activities is represented without commenting on alternatives. Here, dependencies between steps and possibilities to adapt according to different situations have to be made clear. Method components were not explicitly defined, but the method shows several clearly separable steps, which obviously could be considered as "components": the analysis of the overall value creation areas, the analysis of a business process and the use of security patterns – to name the most obvious examples. For each of these potential components, there foremost is a description of the procedure and in most case a way to represent the results of the steps, which often is not an explicitly defined notation but more a representation of results "by example". The important concepts are mentioned and discussed in the scientific publications but not exposed in the description of the procedures.

The cooperation principles are not documented at all in the written SREBP material. However, discussion with the method developers showed that they have a clear picture regarding the required competences and what roles need to be established and filled during analysis. This knowledge has to be made explicit.

4.2 Refinements of SREBP

Perspective. The major goal of the SREBP method is to identify the enterprise's assets, determines their security objectives, and elicits security requirements in order to reason on and ensure the security during the execution of business process. The method integrates security in processes to facilitate business analyst in understanding and deriving the security requirements from the business process models.

Cooperation. Typically security engineering requires a close collaboration between the *business analyst* (i.e., the specialist of the business domain) and *security analyst* (i.e., the specialist of the security domain). Being experts in business domain, business analysts have limited or no expertise in security engineering. They have to rely on the best security practices, information security standards, or security experts.

Business analyst introduces business context to security analyst. On one hand business analyst as an expert of business domain, describes organisation's work-flow. On another hand the goal of security analyst is to understand what business values are described in the business model. In other words security analyst (through collaboration with business analyst) identifies what business assets are, what security objectives (in terms of confidentiality, integrity, and availability) should be taken into account, and what are the IS assets to support the identified business assets.

Once the security requirements are derived from the business process models, they can be used to annotate the original business process model. The artefact that is returned to the business analyst is the business process model annotated with security requirements. But the feedback could also include security risk models. Another cooperation might be on security requirements trade-off analysis. However this activity is not emphasised in the SREBP method.

Method components. Refinement of the SREBP method components is illustrated in Table 2. In the first column this table includes all the major *concepts* used in the SREBP method. Majority of these concepts, like *business asset*, *security criterion*, information systems (*IS*) *asset*, *security risk*, and *security requirements* are taken from the domain model for the information systems security risk management (a.k.a., ISSRM) [9]. But the SREBP method also includes few concepts, like *security risk-oriented patterns*, which result from the use of the base ISSRM concepts.

Table 2: SREBP components

Concepts	Procedure	Notations
Value Chain	Created by the business analyst, expresses how the enterprise business functions are related in order achieve enterprise's goals	BPMN
Business Process Diagram	Created by the business analyst, expresses the use of the computerised information system. These diagrams should express the use of data objects, data flows and data stores.	BPMN
Business Asset	Identified from the value chain	Initially documented textually, later refined graphically depending on various security model notations
Security Criterion	Identified by understanding importance of the business assets	
IS Assets	Identified when analysing the business process diagrams	
Security Risk (and its major components)	Identified from the business process diagrams by instantiating the security risk-oriented patterns	Security risk-oriented BPMN [5]
Security Requirements	Identified from business process diagram by applying the security risk-oriented patterns and by instantiating pattern security parts	Documented textually as security requirements statements, and graphically using UML notations depending on the analysed contextual area
Security Risk-oriented Pattern	Artefact used to guide security risk requirements derivation from the business process diagrams. The patterns describe recurring security risks that arise within business processes. To mitigate the risks, the patterns recommend security requirements.	Documented textually in the structured template and graphically using the security risk oriented BPMN (see [3])
Pattern Occurrence	Identified in the business process diagram using security risk oriented patterns	Highlighted in the analysed business process diagram
Security Model	Derived from the business process model and the result of security risk-oriented pattern application.	Represented graphically using UML notations depending on the analysed contextual area and applied pattern

In the second column of Table 2 we define the <u>*procedures*</u> used to identify the relevant concepts. Hence the business analyst creates value chain and business process diagrams as the part of the organisations business process management. The asset-related concepts are identified from the value chain and business process diagrams, and security risk-related and risk treatment-related concepts are defined using the security risk-oriented patterns.

The third column presents the <u>*notations*</u> used to represent concepts. A notable set of concepts is expressed using the textual language, which is supported with the targeted graphical notations. Since SREBP is meant to consider business processes, majority of the notations are BPMN or security risk-oriented BPMN [3] [5]. However, the security requirements models are represented using UML.

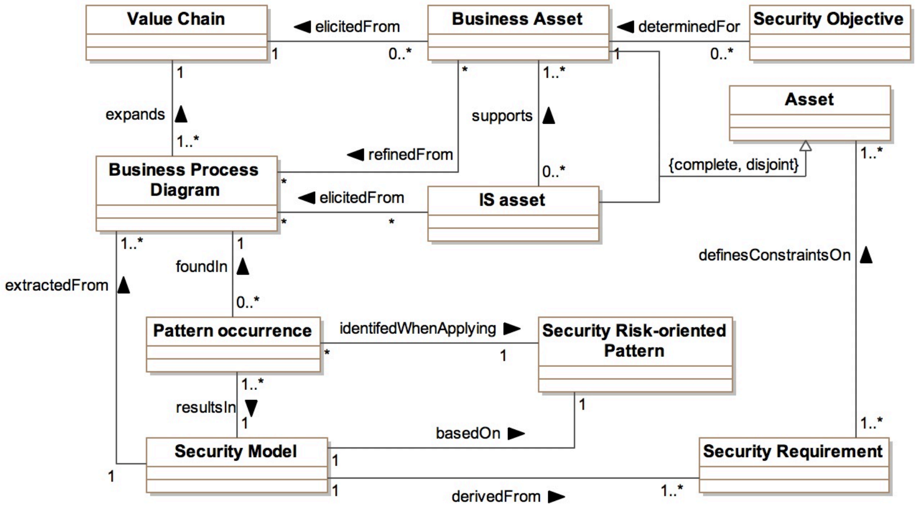

Fig. 2. The SREBP Framework

Framework. In Fig. 3 the relationships between the individual SREBP method components are described. Hence, the *Business Process Diagrams* expands separate actions represented in the *Value Chain* diagram. The *Business Assets* are elicited from the *Value Chain*. As described above the security analyst in cooperation with business analyst determines *Security Objectives* each identified *Business Asset. IS Assets* support *Business Assets*, which are also refined when considering the *Business Process Diagrams.* When applying *Security Risk-oriented Patterns*, *Pattern Occurrences* are found in *Business Process Diagrams*. *Pattern Occurrences* result in *Security Model*, which is extracted from *Business Process Diagram* based on the used *Security Risk-oriented Pattern*. *Security Requirements* are derived from the *Security Model* and they define the security constraints on the *Assets*.

Fig. 4 presents a high level SREBP process. As discussed in Section 3.1, the SREBP method consists of two stages. In the first stage one needs identify business assets and determine security objectives. In the second steps, the main activities include (*i*) identification of the patterns, (*ii*) extraction of security model based on the pattern occurrences, and (*iii*) derivation of the security requirements.

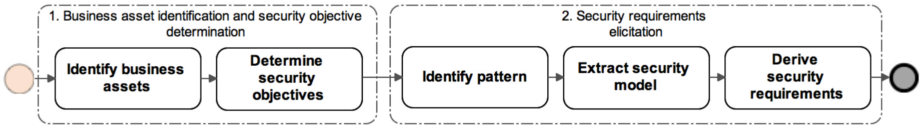
1. Business asset identification and security objective determination
Identify business assets
Determine security objectives
2. Security requirements elicitation
Identify pattern
Extract security model
Derive security requirements

Fig. 3. The SREBP Process

During the first step, one performs activities (like hierarchical level matching, business perspective matching, structural similarity and semantic similarity matching [2]) to identify patterns in the analysed business process diagram. Once the pattern occurrences are determined, one could extract the security model. It is important to note that one could select between analyses of the different contextual areas. Depending on the chosen contextual area (and its associated patterns) different activities for security model extraction could be performed (see Fig. 4). After extracting the security model, one derives and documents security requirements.

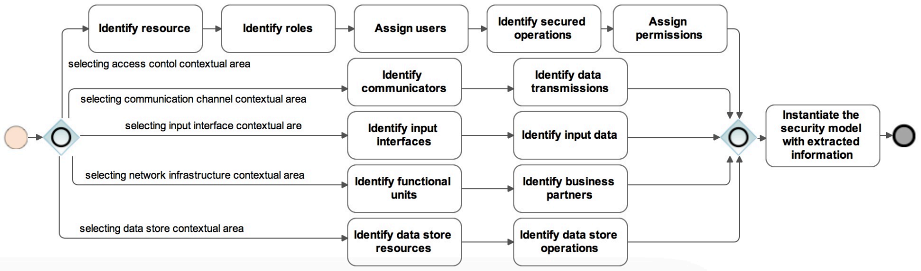
Identify resource
Identify roles
Assign users
Identify secured operations
Assign permissions
selecting access contol contextual area
Identify communicators
Identify data transmissions
selecting communication channel contextual area
selecting input interface contextual are
Identify input interfaces
Identify input data
Instantiate the security model with extracted information
selecting network infrastructure contextual area
Identify functional units
Identify business partners
selecting data store contextual area
Identify data store resources
Identify data store operations

Fig. 4. Expanded activity Extract security model

5 Discussion

Method analysis and method refinement described in Section 4 resulted in some observations, which will be discussed in this section. One observation concerns the suitability of Goldkuhl's method conceptualization. The general impression was that the method conceptualization by Goldkuhl *et al*. proved to be suitable and applicable. The method developers perceived the conceptualization and its way to decompose a method into different elements as helpful in the overall refinement process. The elements were used as a "checklist" for investigating what potential improvements and refinements are possible and could make sense. However, two elements of Goldkuhl's method view needed specific explanation. The term „perspective" had a tendency to confuse the method developers due to the more philosophical interpretation Goldkuhl *et al*. use in their work. An interpretation of perspective as the „purpose" of the method helped to avoid this confusion. Furthermore, the term "framework" was conceived as misleading as it could be interpreted as conceptual framework of the notation used. Thus, we clarified framework as giving an overview to method components and their inter-dependencies.

For the purpose of method knowledge transfer, in particular the "cooperation principles" and the "concepts" within a "method component" were considered as very valuable since speaking the same language (i.e. using the same concepts with an agreed-on meaning) and explicitly stating requirements with respect to the competences of the method users showed to be crucial for transferring the knowledge. SREBP previously primarily was used in a team at Tartu University who had been cooperating during many years. In such a situation, there is much implicit knowledge shared by the team members, which needs to be made explicit when transferring knowledge to outsiders. In this context, explanations of important concepts help to avoid misunderstandings. Furthermore, SREBP method users have to be expected to have at least a basic level of knowledge in IT-security and process modelling.

In Section 4.2 we have refined or intend to refine the SREBP method following the requirements listed at the end of Section 2. We acknowledge the need to provide more detailed explanations of the SREBP terms and concepts. We have started this process in Section 4.2, how due to the limited space he we include only limited explanations. A separate technical report needs to be prepared to clarify these terms. Next, we highlight the procedures and prerequisites needed to execute the SREBP method. For example, we stress that the *business process diagrams* need to be prepared in the way so that the used data and data stores would be represented in these diagrams. Finally, some activities of the SREBP method should be always executed in the same order (e.g., see Fig. 3). But some activities especially when analysing different contextual areas (e.g., see Fig. 4), could be executed in different order or even skipped from the analysis if the specific contextual area is outside the scope.

6 Summary and Future Work

Based on lessons learned from transferring knowledge about the SREBP method from method developers to new method users, the paper investigated refinement potential of SREBP and proposed changes in SREBP. The basis for identifying refinement potential was a decomposition of SREBP using a proven approach from method engineering, Goldkuhl *et al.*'s [11] component-based method view.

Future work in this area primarily has to focus on evaluating (a) the refined SREBP version as such in real-world cases and (b) the suitability of the documentation of the refined SREBP for knowledge transfer to new method users. The former is directed to the quality of SREBP to elicit security requirements whereas the latter addresses completeness and understandability of SREBP usage and its prerequisite. Thus, we started method transfer and method usage activities within the ITSE project.

Acknowledgements

This paper of the Baltic-German University Liaison Office is supported by the German Academic Exchange Service (DAAD) with funds from the Foreign Office of the Federal Republic Germany.

References

1. Ahmed, N., Deriving Security Requirements from Business Process Models, PhD thesis, University of Tartu, 2014
2. Ahmed., Matulevičius R.: A Taxonomy for Assessing Security in Business Process Modelling. Proceeding of RCIS, 2013: IEEE, 1-10
3. Ahmed, N., Matulevičius, R.: Securing Business Processes Using Security Risk-oriented Patterns. Computer Standards and Interfaces 36(4), 723–733 (2014)
4. Ahmed, N., Matulevičius, R.: A Method for Eliciting Security Requirements from the Business Process Models. In: CAiSE Forum, CEUR Workshop Proceedings, pp. 57– 64. CEUR-WS.org , 2015
5. Altuhhova, O., Matulevičius, R., Ahmed, N.: An Extension of Business Process Model and Notification for Security Risk Management. International Journal of IS Modeling and Design (IJISMD) 4, 93–113 (2013)
6. Avison, D. E. & Fitzgerald, G. (1995) Information Systems Development: Methodologies, Techniques and Tools. Berkshire, England: McGraw Hill.
7. Brinkkemper S., Method engineering: engineering of information systems development methods and tools, Information and Software Technology, 1995 37.
8. Brucker A., Hang I., Lückemeyer G., Ruparel R., SecureBPMN: Modeling and Enforcing Access Requirements in Business Processes, Proceedings of the 17th ACM symposium on Access Control Models and Technologies (SACMAT'12), pp 123-126
9. Dubois, E., Heymans, P., Mayer, N., Matulevičius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management. In: Intentional Perspectives on Information Systems Eng., pp. 289–306. Springer (2010)
10. Firesmith, D.: Engineering safety and security related requirements for software intensive systems. In: ICSE 2007 Companion, p. 169. IEEE (2007)
11. Goldkuhl, G.; Lind, M. and U. Seigerroth (1998) Method integration: the need for a learning Perspective. . IEE Proceedings, Software (Special issue on Information System Methodologies), Vol. 145, Nr 4.
12. Jürjens J., Developing Secure Systems with UMLsec from Business Process to Implementation, Verlässliche IT-Systeme 2001, DuD-Fachbeiträge 2001, pp 151-161
13. Jürjens, J.: Secure Systems Development with UML. Springer (2005)
14. Mirbel I., Ralytė J., Situational method engineering: combining assembly-based and roadmap-driven approaches, Requirements Eng. 11, 2006, pp 58–78.
15. Ralytė J., Backlund P., Kühn H., Jeusfeld M. A. (2006) Method Chunks for Interoperability. ER 2006, LNCS 4215, pp. 339 – 353, 2006, Springer-Verlag Berlin Heidelberg.
16. Rodriguez, A., Fernandez M, E., Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE-TIS(4) pp. 745–752 (2007)
17. Röstlinger, A. & Goldkuhl, G. (1994) På väg mot en komponentbaserad metodsyn. (in Swedish). Presented at "VITS Höstseminarium 1994", Linköping University, Linköping, Sweden.
18. Salnitri M., Paja E., Giorgini P., Preserving Compliance with Security Requirements in Socio-Technical Systems, Cyber Security and Privacy, CCIS 470, Springer, 2014, pp 49-61
19. Seigerroth U. (2011) Enterprise Modelling and Enterprise Architecture – the constituents of transformation and alignment of Business and IT, International Journal of IT/Business Alignment and Governance (IJITBAG), Vol. 2, Issue 1, pp 16-34, 2011.
20. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering 10(1), 34–44 (2005)
21. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2012)