Journal of Universal Computer Science, vol. 24, no. 10 (2018), 1472-1492
submitted: 21/4/17, accepted: 17/10/18, appeared: 28/10/18 © J.UCS

Towards Obtaining UML Class Diagrams from Secure Business Processes Using Security Patterns

Matías Zapata-Barra
(Computer Science and Information Technologies Department
University of Bío-Bío, Chillán, Chile
matzapat@alumnos.ubiobio.cl)

Alfonso Rodríguez
(Computer Science and Information Technologies Department
University of Bío-Bío, Chillán, Chile
alfonso@ubiobio.cl)

Angélica Caro
(Computer Science and Information Technologies Department
University of Bío-Bío, Chillán, Chile
mcaro@ubiobio.cl)

Eduardo B. Fernández
(Computer and Electrical Engineering and Computer Science Department
Florida Atlantic University, Florida, United States
fernande@fau.edu)

Abstract: The incorporation of security concepts on business processes models has turned out to be an interesting factor in the software development cycle, since it allows an early capture of security aspects, which will then be used in later stages. A way of complementing security incorporated in a business process is to link this kind of requirement with security patterns, due to the importance of these patterns on the software development process. This article tackles the procurement of UML classes that allow the advancement in a software development process with security requirements expressed as business processes as a base, using a BPMN extension and security patterns.

Keywords: BPMN, UML, BPMN-BPSec, Security Patterns, Security Requirement
Categories: D.2, D.2.1

1 Introduction

The absence of technology, be it either tools or mechanisms that are necessary for supporting software development, is one of the main vulnerability factors and a weakness of systems in general [Solinas, 09].

On the other hand, the development of organizations, many participants and an intensive use of communications and information technologies, carries with it an increased vulnerability, as it heightens the number of attack attempts on them and, most likely, one of these attacks will sooner or later succeed [Quirchmayr, 04].

In this scenario, security can no longer be considered an independent objective, forcing organizations to coordinate, deploy and direct many of their essential capacities in order to attain solutions that support security from an adequate perspective. A method for tackling this problem is to include security in early stages of business process models [Basin, 06; Herrmann, 06; Jürjens, 02; Mülle, 11; Rodríguez, 06, 07; Wolter, 08]. From this, it is possible to obtain Secure Business Processes, which can then be used as part of a software development process.

Transformations from a Secure Business Process specified with BPMN-BPSec [Rodríguez, 07] to a UML Class Diagram have been addressed in [Rodríguez, 10]. Adopting the idea of evolving models [Mellor, 02], this proposal attempts to utilize Security Patterns in order to obtain UML Class Diagrams, since these represent the best practices and experiences from experts. Moreover, these patterns materialize the mechanisms that allow the protection of confidentiality, integrity and availability of information in a system. Additionally, Security Patterns directly focus in solving security problems [Schumacher, 13].

In literature, two approaches can be observed in regards to tackling the utilization of Patterns in Business Processes. The first tackles Patterns in a general manner (without a focus on security), considering a Business Process model as a starting point, in which the patterns are applied [Bonillo, 06; Elgammal, 14; Forster, 07; Gschwind, 08; Schumm, 10; Anstett, 10][Samarütel, 16]. The second contemplates the use of Security Patterns over a Secure Business Process specification [Ahmed, 14] and [Khan, 12] [Varela-Vaca, 16] [Argyropoulos, 17]. These proposals do not consider the usage of security patterns in the transformation of secure business processes into UML classes. Previous works have shown transformations into secure UML classes from business processes [Rodríguez, 10]. This work aggregates security patterns with the objective of improving the obtained class models which, based on the results obtained from the validation, improves the representation of security and the obtained models' clarity. Consequently, the Method-Secure Business Process and Patterns (M-SecBP&P) method is proposed in order to obtain UML Class Diagrams from Secure Business Processes using Security Patterns.

The aim of this method is to link an early security requirements specification in a business process together with Security Patterns, whose selection is based on the specified security requirements, in order to generate UML Class Diagrams. The generated diagrams are validated in regards to their completeness, their comprehensibility and their usefulness for a software development process.

The remainder of this article is organized as follows: concepts related to security in business processes and security patterns are addressed in Section 2; a detail of the most important related work is shown in Section 3; the article's proposal is presented in Section 4, which consists on obtaining UML Class Diagrams using Security Patterns from a Secure Business Process; an illustrative example is shown in Section 5; the proposal's validation is addressed in Section 6; and finally, the work's conclusions are presented in Section 7.

2 Related Concepts

In order to contextualize the proposal, the basic concepts of Business Process Model and Notation (BPMN) [OMG, 15], security in business processes and security patterns are detailed in this section.

2.1 Basic BPMN Modelling Elements

The secure business processes in this proposal are created based on the BPMN standard. Thus, it is useful to present the basic elements for the modeling of these business processes, which is shown in Table 1. The use of BPMN has been decided because the basic modelling elements enable the easy development of simple Business Process Diagrams that will look familiar and be understandable by business users, business analysts, technical developers and business people.

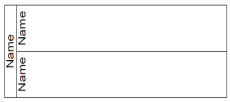
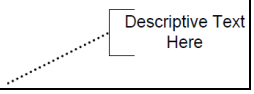
Element	Description	Notation
Activity	Generic term for work that company performs in a Process. The types of Activities that are a part of a Process Model are Sub-Process and Task.	
Group	Is a grouping of graphical elements that are within the same Category. This type of grouping does not affect Sequence Flows. Categories can be used for documentation or analysis purposes.	
Pool	Graphical representation of a Participant in a Collaboration. It also acts as a "swimlane" and a graphical container for partitioning a set of Activities from other Pools.	
Lane	A sub-partition of a Process, and sometimes of a Pool. It extends the entire length of the Process. Used to organize and categorize Activities.	
Message Flow and Message	A Message Flow is used to show the flow of Messages between two Participants. A Message is used to depict the contents of a communication between two Participants	
Data Object	Provide information about what Activities require to be performed and/or what they produce. Data Objects can represent a singular or a collection of objects.	
Gateway	Used to control divergence and convergence of Sequence Flows in a Process. Thus, it determines branching, forking, merging and joining of paths.	
Event	An Event is something that "happens" during the course of a Process. There are three types of Events: Start, Intermediate, and End.	
Text Annotation	Text Annotations are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram.	

Table 1: Basic BPMN Modelling Elements

2.2 Secure Business Process

Organizations possess distinct perspectives in regards to security underneath information systems. Every perspective has its own requirements, as well. Regardless, considering a high degree of abstraction, every system tends to have the same basic types of valuable assets and potential vulnerabilities of these assets [Firesmith, 04]. These vulnerabilities can be expressed as security requirements, which can be decomposed into sub-factors of quality. Through these sub-factors it is possible to generically identify security concepts [Firesmith, 04].

In literature, although various authors include topics related to security in Business Processes [Basin, 06; Herrmann, 06; Jürjens, 02; Mülle, 11; Rodríguez, 06, 07; Wolter, 08], this work will make use of the BPMN-BPSec extension, which is proposed in [Rodríguez, 07]. This selection is done because it allows to explicitly represent a subset of the security requirements presented in [Firesmith, 04] about Business Processes models described with BPMN. For this purpose, BPMN elements (shown in Table 1) where it is possible to define a list of requirements which conform the BPSec extension are used, as shown in Table 2.

Notation	Security Requirements	BPMN Element	Description
	Access Control	Pool, Lane, Group y Activity	Limited access to resources only to authorized user.
	Attack Harm Detection	Pool, Lane, Group, Activity, Message Flow, y Data Object	Detection, recording and reporting of an attempted attack.
	Audit Register	Pool, Lane, Group, Activity, Message Flow, y Data Object	Ability to collect and analyze information on the use of security mechanisms.
	Integrity	Message Flow y Data Object	Components protection and unauthorized intentional corruption.
	Non Repudiation	Message Flow	Necessity to avoid denial of any aspect of an interaction.
	Privacy	Pool, Lane y Group	Information protection, limiting access to unauthorized entities.

Table 2: Security Requirements – BPMN-BPSec [Rodríguez, 07]

Table 2, summarizes the security requirements that can be represented with BPMN-BPSec and the elements that can have these specifications in BPMN. It is important to mention that the BPSec extension allows the business analyst to express security requirements from their own perspective. These requirements can then be refined by a security expert so that they can be used in a software development process afterwards.

2.3 Security Patterns

A security pattern describes a recurrent and particular security problem. It is applied on specific contexts and presents a generic solution that has already been proven to work for such a problem [Schumacher, 13].

A patterns taxonomy is proposed in [Bonillo, 06], where they are grouped according to the following abstraction levels:

- Analysis Patterns: group of concepts that are part of a common construction in the world of conceptual modeling, they are relevant to a specific domain or can be adapted to others. The vision is conceptual and structured, identifying the nature of situations. This level does not define security patterns.
- Architecture Patterns: fundamental schemes of a system's organization, identifying a series of sub-systems and their respective responsibilities. The following security patterns can be found in this level: (I) SINGLE ACCESS POINT, (II) CONTROL POINT and (III) SECURITY SESSION.
- Design Patterns: lower level of abstraction than architecture patterns, which is closer to the code. Their use does not reflect the global structure of a system as is. The following security patterns can be found in this level: (I) AUTHORIZATION, (II) ROLE-BASED ACCESS CONTROL, (III) MULTI-LEVEL SECURITY and (IV) REFERENCE MONITOR.
- Interaction/Interface Patterns: successful solutions to problems related to user interface. They constitute a means for communication expressed in a simple notation, which can be understood by the design team. The following security patterns can be found in this level: (I) TOTAL ACCESS WITH ERRORS and (II) LIMITED ACCESS.

There are many Security Patterns that depend on the problem's context in literature. Since it is not possible to use them all, this work focuses on those Security Patterns shown in [Schumacher, 13]. The focus will primarily be set on Security Patterns oriented to Access Control in the design, architecture and interface levels.

Security Patterns are mainly classified as Architectural Patterns. This article uses the Security Patterns found most frequently in literature. The used patterns are: (I) AUTHORIZATION, (II) ROLE-BASED ACCESS CONTROL, (iii) MULTI-LEVEL SECURITY, (iv) REFERENCE MONITOR, (v) SINGLE ACCESS POINT, (vi) CONTROL POINT (vii) SECURITY SESSION, (viii) TOTAL ACCESS WITH ERRORS and (ix) LIMITED ACCESS [Fernandez-Buglioni, 13; Schumacher, 13; Rosado *et al.* 06]

3 Related Work

Works that utilize patterns in the Business processes context are described next. Special emphasis is given to those works that describe proposals that address the usage of security patterns related to business processes. A Systematic Literature Review [Kitchenham, 09] has been performed in order to identify these works, which allows their evaluation and interpretation, giving an emphasis on works that describe proposals that address the usage of security patterns related to business processes.

3.1 Patterns Proposed in Business Processes

Bonillo [Bonillo, 06] proposes a referential and integral theoretical framework, together with a methodology that covers from the requirements analysis to the monitoring of processes, supporting the stages of analysis, design, modeling and configuration, through the usage of patterns. The proposal is composed by two macro-processes: the first is related to the creation of the process itself, whereas the second corresponds to management, comprising the maintenance, the management itself and the monitoring of the process through management indicators. This proposal is only for the inclusion of patterns at the architecture level, which considers quality concepts, but it does not show how to evaluate these concepts.

Forster *et al*. [Forster, 07] describes a visual patterns language for the representation and execution of quality restrictions in business processes models. These restrictions are formally described through process patterns based on UML Activity Diagrams.

Gschwind *et al.* [Gschwind, 08] describes a business processes modeling tool extension that allows the integration of workflow patterns, giving a warning on the context and the consequences of its use, which allows the user to avoid possible errors when applying edition-time processes patterns.

Elgammal *et al.* [Elgammal, 14] presents an integral framework for the management of the fulfillment of business processes, with a focus on the design period as a first step towards the preventive support of the fulfillment of business processes. This framework possesses a fulfillment specification language based on patterns, which facilitates the formal specification of requirements for the fulfillment of business processes, automatically generating the fulfillment rules.

Schumm *et al.* [Schumm, 10a] presents a process views meta-model and also shows process views patterns, which specify elementary transformations for existing processes. In a later related work [Schumm, 10b], how to combine view patterns is shown, in order to help in the design, deployment, monitoring and analysis phases of business processes.

Awad *et al.* [Awad, 15] presents a framework for the proactive monitoring on execution time of business processes, called BP-Maas. This framework incorporates a wide range of fulfillment patterns for the abstract specification of restrictions in execution time of business processes.

Lohrmann and Reichert [Lohrmann, 15] describes an approach for the evaluation of process improvement patterns in specific scenarios considering real-world limitations, such as the role of high tier stakeholders or a system's adaptation cost.

Finally, Brambilla *et al*. [Brambilla, 12] presents a processes design methodology, supported by a set of tools, in order to include social features in business processes. An approach for supporting the design and implementation of "Social BPMN" solutions is presented. This extends BPMN's visual language in order to design processes with social interactions, gathering typical scenarios of socialization processes as reusable design patterns.

3.2 Security Patterns and Business Processes with security specifications

Ahmed and Matulevičius [Ahmed, 14] proposes a method that introduces security requirements in business processes through the collaboration between business

analysts and security analysts. Samarütel [Samarütel, 16] applies these patterns on business processes from aviation turnaround systems. In order to support this collaboration, a set of security risks patterns which have been proposed in [Khan, 12] is used. However, this work does not identify these patterns in the modeling phase, but rather explicitly presents them as selectable items from a drag and drop tool so that they can be integrated to the model.

Varela-Vaca [Varela-Vaca, 16] proposes OPBUS, which consists of a framework for the optimization of security in business processes, allowing risk identification and estimation in business processes based on flow control patterns.

Argyropoulos *et al.* [Argyropoulos, 17] proposes a framework for modeling business processes by using secure processes patterns.

Table 3 summarizes the above proposals. For each proposal, the language used for constructing the origin and the destination models, the capacity of the model generation of being automatized and the kinds of patterns used by the proposal are shown.

Proposal	**Origin**	**Destination**	**Automatized**	**Kinds of Patterns**
[Bonillo, 06]	BPMN	UML (no specific kind model of)	No	Patterns in General
[Forster, 07]	UML - Activity Diagram	UML - Activity Diagram	Yes	Visual Patterns Process
[Gschwind, 08]	BPMN	BPMN	Yes	Workflow Patterns
[Schumm, 10]	BPMN	BPMN	Yes	Process Viewing patterns
[Brambilla, 12]	BPMN	WebML	Yes	Socialization patterns
[Khan, 12]	BPMN	BPMN	No	Security Risk-Oriented Patterns
[Ahmed, 14]	BPMN	BPMN	Yes	Security Risk-Oriented Patterns
[Elgammal, 14]	BPMN	BPMN	Yes	Compliance Patterns
[Awad, 15]	BPMN	BPMN	Yes	Compliance Patterns
[Lohrmann, 15]	BPMN	BPMN	No	Process Improvement Patterns
[Varela-Vaca, 16]	BPM	BPM	Yes	Control-Flow Patterns
[Argyropoulos, 17]	BPMN	BPMN	Yes	Security Process Patterns

Table 3: Summary of Related Work

From the information presented in Table 3, it is possible to conclude that there are no works: (i) whose origin model has been built with BPMN-BPSec or some other security extension for BPMN, (ii) that use security patterns for generating destination models and (iii) whose destination models is a UML Class Diagram.

Although [Ahmed, 14] utilizes a business process security description based in the visual aspect proposed in [Rodriguez, 07], only security risks patterns are used and the transformation of the specifications into UML Class Diagrams does not exist.

In order to summarize, it can be said that there are no Works in literature where UML Class Diagrams are obtained from Secure Business Processes using Security Patterns for their generation. However, having such a perspective would give increased value to security specifications as it would allow incorporating security from a business process, which can then be used to create Security Patterns for the generation of UML Activity Diagrams. As such, the M-SecBP&P method is specified in order to cover this necessity.

4 M-SecBP&P: Proposal – Transformation of models by the use of Security Patterns

This section presents the M-SecBP&P proposal, which allows the selection of Security Patterns through the analysis of security requirements that are described in a business process model with BPMN-BPSec. In this sense, Section 4.1 presents the method used for selecting security patterns and their transformations, whereas Section 4.2 shows the relations between BPMN-BPSec and the architectural, design and interface security patterns.

4.1 Method for the selection of Security Patterns

The Method - Secure Business Process and Patterns (M-SecBP&P) - method's objective is to allow the selection and adaptation of security patterns by using the information obtained from a Secure Business Process specified with BPMN-BPSec.

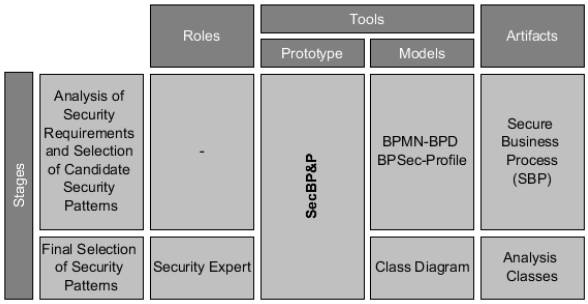


Figure 1: Overview of M-SecBP&P

An overview of M-SecBP&P is shown in Figure 1. M-SecBP&P is composed by a set of stages, roles, tools and artifacts, which are detailed next.

Stage-1: Security Requirements Analysis and Candidate Security Patterns Selection

The objective of this stage is to perform an analysis of the security requirements specified in the Secure Business Process, which is received as an input model and is specified with BPMN-BPSec.

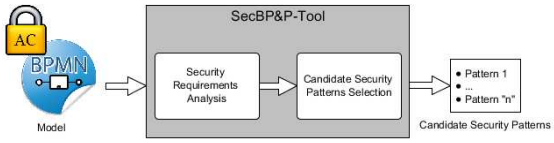


Figure 2: Security Requirements Analysis Scheme

Figure 2 shows the tasks that are performed in this stage. This stage does not have an associated role, since it is automatically performed by using the SecBP&P-Tool prototype, which is detailed in Section 5.

A Secure Business Process is taken as the input model. From here, the model's security requirements are analyzed to preliminarily select the Security Patterns. The ATL language [ALT, 17] has been used to perform this analysis. This language allows surveying a Secure Business Process diagram's internal logic and determining the kind of security requirement that has been specified in the diagram's elements.

The analysis is performed for every POOL in the business process, since this represents a minimal business entity. This is done considering the elements possessed by the POOL and its received messages as part of the POOL itself and not as independent entities. This allows the generation of a candidate security patterns list for each POOL.

Once the information on security patterns has been obtained for each POOL, a comparison between this information and the existing relation between the security requirements and security patterns (see Section 4.2) is performed, generating a list containing the security patterns that comply with the expressed requirements.

A security patterns list is obtained in this stage, which is automatically generated and contains the patterns that comply with all of the security requirements.

Stage-2: Final Selection of Security Patterns

The security expert gets involved in this stage, taking the responsibility of selecting the pattern that they deem the most convenient, based on the secure business process and the proposed requirements. This pattern is selected from a previous set of security patterns. Figure 3 illustrates the above.

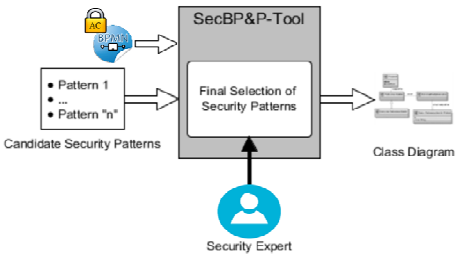


Figure 3: Security Patterns Selection Scheme

The output of this stage is a UML Class Diagram that considers the selected security pattern(s). The classes obtained as a result of the selection of the security pattern in this stage are complemented with the security classes generated by the BPMN-BPSec extension. This is applied to functional classes and, in case that these are related to security requirements, only those that are not already considered in the selected security pattern will be added. The transformation rules proposed in Rodriguez *et al* [Rodriguez, 10] are used for the transformation from a business process into a class diagram. These rules are summarized in Table 4.

BPMN Element	Class Diagram Element
Pool	Class
Lane	Class
Data Object, Message	Class
Activity	Operation
Security Requirement	Class

Table 4: Equivalence between BPMN and Class Diagram Elements summary [Rodríguez, 10]

These equivalencies allow the transition from a Secure Business Process into an equivalent Class Diagram. This Class Diagram is then used as the base for adapting the selected security pattern(s).

4.2 Relation between Security Patterns and Security Requirements

A security requirement can be interpreted differently depending on where it has been specified. For example, when the ATTACK HARM DETECTION security requirement is specified in a POOL, it implies keeping a registry of all of the activities performed by this entity. If this security requirement is taken to the software development context, it implies keeping a registry of the users that access the system and the functionalities that they make use of. On the other hand, when this security requirement is specified in a DATA OBJECT, it only implies keeping a registry of the activities that interact with the same DATA OBJECT.

Due to the above, Access Control Monitoring (ACM) can be related to a security requirement when it is specified either in a POOL, in a LANE or in a GROUP, since these represent an entity and/or a participant in BPMN. On the other hand, Resource

Monitoring (RM) can be related to a security requirement when it is specified either in an ACTIVITY, in a MESSAGE FLOW or in a DATA OBJECT, since these represent resources in BPMN, being these tasks or functions, messages and data respectively. This relation is shown in Table 5. This information is later used for the generation of the candidate patterns, keeping the BPMN elements in consideration (Stage-1: Security Requirements Analysis and Security Patterns Selection, in Figure 2).

	BPMN Elements					
	Access Control Monitoring			Resources Monitoring		
Security Requirement	Pool	Lane	Group	Activity	Message Flow	Data Object
Access Control	x	x	x	x		
Attack Harm/Detection	x	x	x	x	x	x
Integrity					x	x
No Repudiation					x	
Privacy	x	x	x			
Audit Register	x	x	x	x	x	x

Table 5: Types of BPSec Monitoring, adapted from [Rodríguez, 07]

Considering this relation, it is possible to associate every considered security requirement in the BPMN-BPSec proposal with the security patterns identified in literature, categorized in the architecture, design and interface levels. Thus, Table 6 presents the relation between the security requirements and patterns while considering the monitoring type (Access Control and/or Resources).

A detailed explanation of the relations presented in Table 6 is given next, taking the patterns that can be associated to each security requirement as a reference.

For the ACCESS CONTROL requirement, the patterns at the architecture level comply with identification, authentication and authorization. In regards to the design level, every pattern of this level coincides solely on authorization and can be done both for access control and resource monitoring. In regards to design-level patterns, they only coincide with authorization as these patterns validate the access of a user to the system's resources, its modules or a particular system. On the other hand, interface-level patterns comply only with authorization since they consider that the user has already accessed to the system before performing validations.

For the ATTACK HARM DETECTION requirement, the same strategy of applying a *Log* can be applied. By applying this strategy, the *Single Access Point* and *Control Point* architecture patterns can only implement access-control level detection. On the other hand, the *Security Session* pattern can implement both an access-control and a resource level detection. In regards to design-level patterns, the *Log* can be applied to the *Multi-level Security* and *Reference Monitor* patterns but only for resource monitoring. Finally, at an interface-level, the *Total Access with Errors* pattern can implement threat detection at a resources-level.

			Architecture-level Security Patterns			**Design-level Security Patterns**				**Interface-level Security Patterns**	
	Security Requirement BPMN-BPSec	Type of Monitoring	Single Access Point	Control Point	Security Session	Authorization	RBAC	Multi-level Security	Reference Monitor	Total Access with Errors	Limited Access
AC	Access Control (Identification)	ACM*	x	x	x						
	Access Control (Authentication)	ACM	x	x	x						
	Access Control (Authorization)	ACM	x	x	x	x	x	x	x	x	x
		RM**	x	x	x	x	x	x	x	x	x
AD	Attack Harm Detection	RM			x			x	x	x	
		ACM	x	x	x						
	Audit Register	RM	x	x	x			x	x		
I	Integrity	RM	x	x	x			x	x	x	x
NR	No Repudiation	RM			x			x	x	x	x
Px	Privacy (Confidentiality)	ACM	x	x	x	x	x	x	x	x	x

***ACM.**: Access Control Monitoring ****RM**: Resource Monitoring

Table 6: Relation between Security Patterns and BPMN-BPSec Security Requirements

For the AUDIT REGISTER requirement, it is possible to incorporate a *Log* in order to audit the system in regards to the access to the information, just like in the previous requirement. In regards to architecture-level patterns, due to their focus on verifying and validating the way in which a user accesses the system, the *Single Access Point and Control Point* security patterns can only be audited at an access-control level. On the other hand, *Security Session* allows auditing at a resource-control level. In regards to design-level patterns, *Multi-level Security and Reference Monitor* allow knowing who is accessing the resources. Finally, since they consider that the user has already accessed the system and do not validate this access, the *Total Access with Errors and Limited Access* interface patterns only allow auditing at a resources-level.

In the case of INTEGRITY, this requirement is associated to all patterns at the architecture and interface levels, since those patterns allow verifying that only authorized users can Access and/or modify the information that corresponds to them, based on their necessities. In regards to design-level patterns, this requirement is associated to the last two patterns of said level.

The NO REPUDIATION requirement is directly linked with resources monitoring but is not explicitly related with any specific pattern. However, since there are patterns that verify the access rights to resources, it is possible to add a *Log* with the objective of storing information referring to who accesses the system's information, so that it can be analyzed afterwards. By implementing this strategy, in regards to architecture patterns, only *Security Session* allows a resources monitoring, since it possesses the user's security information at every moment and it can be obtained whenever they access some resource. On the other hand, in the design-level, the *Multi-level Security* and *Reference Monitor* patterns are in charge of assigning resources and of intercepting requests made to them. Finally, in regards to interface patterns, none of them possess resource repudiation. *Total Access with Errors* verifies access rights every time that a user accesses a resource of the system, making it possible to apply the *Log* strategy. Although *Limited Access* focuses solely on showing the functionalities of every user whenever they access a resource, the system is the one that gives the access to resources, making it possible to apply the *Log* strategy.

In regards to the PRIVACY requirement, this one is associated with confidentiality and has a relation with every pattern since they all focus on assuring that the information is available only to authorized people.

Thus, the relations between security requirements and security patterns oriented to Access control have been stablished and explained.

5 Illustrative Example

The SecBP&P-Tool prototype was used for this example. The tool prototype's interface is shown in Figure 44. This tool supports the realization of the tasks related to the analysis of the security requirements, the presentation of candidate security patterns and the selection of a security pattern in order to generate a class diagram with the Security Pattern's adaptation.

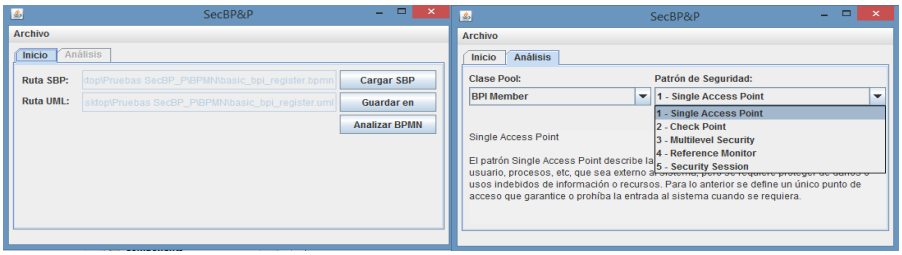


Figure 4: SecBP&P Prototype

The prototype was developed using Java 1.8 due to the ease of installing such an application in the user's environment. ATL Transformation Language [ATL, 2017] was used for processing the transformations from Business Process models into Class Diagram models. Lastly, the PlantUML plugin was used for generating the UML Class Diagram obtained through the transformation rules.

The illustrative example consists on a business process that includes security requirement. It covers the registration of a user in a webpage (see Figure). In such a process, the tasks to be realized by the BPI-Member POOL are to *complete the registration form, confirm the reception of the email address* and *log into BPI*. On the other hand, the BPI-Web POOL contemplates the tasks to *create a member profile, issue an email for confirmation* and *enable total access to the services*.

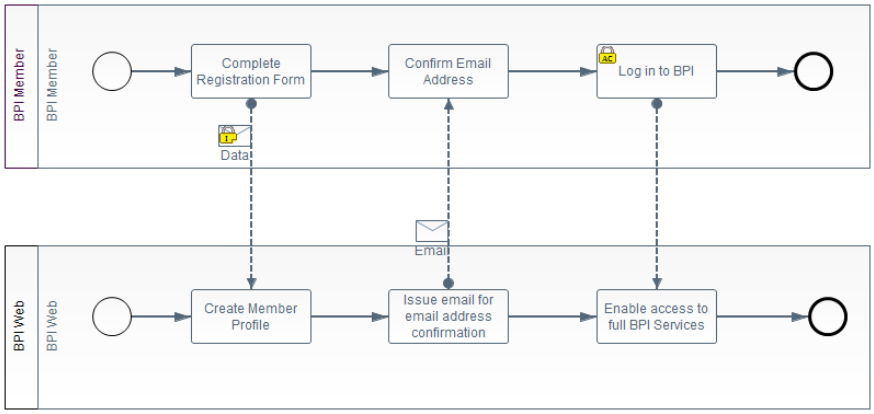


Figure 5: User Registration Example adapted from [BPMN, 15]

The security requirements specified in Secure Business Process are:

- INTEGRITY with audit register, in the "Data" message, this is for the BPI-Web and BPI-Member POOLs.
- ACCESS CONTROL in the "Log in to BPI" task, this is for the BPI-Member POOL.

By contrasting the information of the security requirements of both POOLs, it is possible to obtain the Candidate Security Patterns for each POOL, which in this case are:

- BPI-Member: Single Access Point, Check Point, Security Session, Multilevel Security and Reference Monitor.
- BPI-Web: Single Access Point, Check Point, Security Session, Multi-level Security and Reference Monitor.

The information that flows between POOLs can be classified in different security levels. Especially the information sent from BPI-Member to BPI-Web, which contains registration information that is more sensible, elaborated and complete than the email that is sent back for confirming the creation of the user. For this example, the "Multi-level Security" pattern was selected for both POOLs.

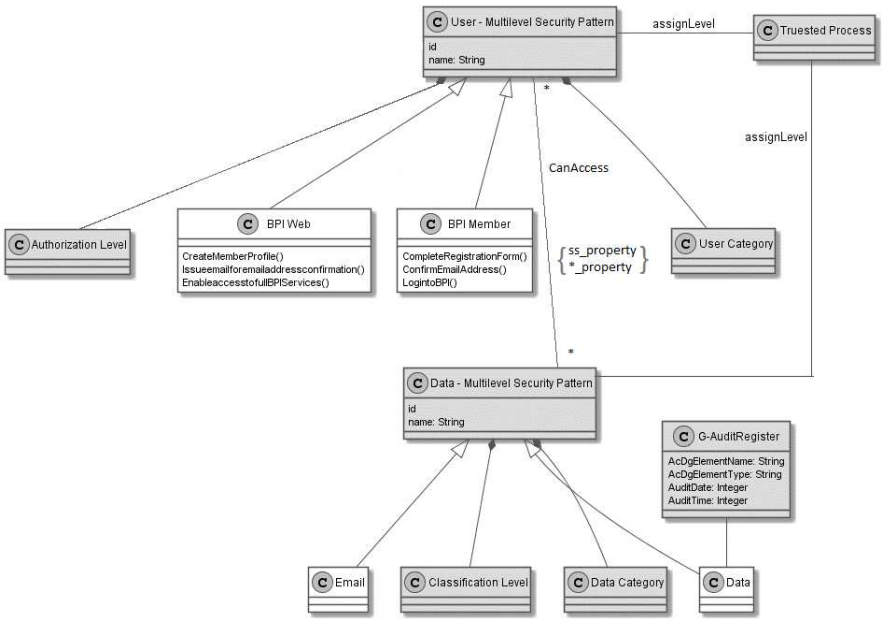


Figure 6: Security Pattern – Illustrative Example

The automatically obtained Class Diagram from the Secure Business Process specification is shown in Figure 6. This UML Class Diagram is the result of the selection of the security pattern (see Stage-2) where the grayed classes represent the security classes. With the exception of the G-AuditRegister class, which has been directly obtained from BPMN-BPSec, all other security classes correspond to the Multi-level Security pattern.

6 Validation

A quasi-experiment has been conducted for validating the models obtained with the SecBP&P proposal. A quasi-experiment is an empirical enquiry similar to an experiment, where the assignment of treatments to subjects cannot be based on randomization, but emerges from the characteristics of the subjects or objects themselves [Wohlin, 12].

The objective of the quasi-experiment was to demonstrate that the UML classes model generated by the M-SecBP&P proposal was more complete and easier to understand than one generated with a pattern-less method like the one proposed in [Rodríguez, 10].

The experimental objects used were a secure business process and a tool to make the transformations. The utilized Secure Business Process is shown in Figure 7. This business process represents a supplier's Purchase Order prosecution, who receives the payment information from a financial institution. The supplier sends the payment confirmation to the same financial entity afterwards, and then proceeds to process the

Purchase Order and ship the products. As it can be observed in Figure 7, this business process possesses three security requirements (which are represented by the padlocks).

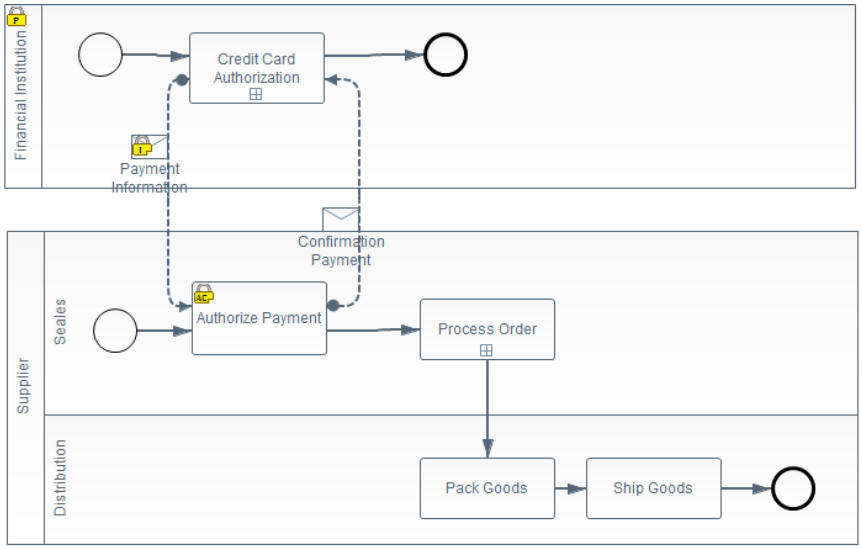


Figure 7: Purchase Prosecution with Credit Card Authorization BPMN adapted from [BPMN, 15]

A total of 26 subjects participated in the quasi-experiment, who on average had between 3 to 6 years of experience in the use of Class Diagrams. These subjects were selected among developers graduated from the Informatics Engineering career in the University of Bío-Bío.

An independent variable, otherwise known as main factor, is the origin of the Class Diagrams. It is also a nominal variable that takes two values: Model A (Diagram automatically obtained from the BPSec extension [Rodriguez, 10], shown in Figure 8) and Model B (Diagram automatically obtained from M-SecBP&P, shown in Figure 9). The dependent variables are: completeness, understandability and level of usefulness to initiate a software development process. The measurement of these variables is done through a questionnaire given to the participating subjects.

For the realization of the experiment, the subjects were divided in two groups of 13 subjects each. Each group was given the Class Diagrams in distinct order (first Model A and then Model B for the first group, and vice-versa for the second group), in order to avoid that the order in which the diagrams were shown would influence their answers. Thus, the subjects were given a questionnaire for obtaining information in regards to their knowledge and experience in regards to the experiment's relevant topics. This questionnaire consisted of five closed questions. Next, business process and the first transformation's resulting class diagram were presented to them, followed by a questionnaire inquiring about the variables to measure. The questionnaire consisted of one open and five closed questions. Afterwards, the same

business process and the second transformation's resulting class diagram were presented to them, followed by the same questionnaire as above. Finally, a questionnaire consisting of five closed questions was applied inquiring about the comparison between the two generated models.

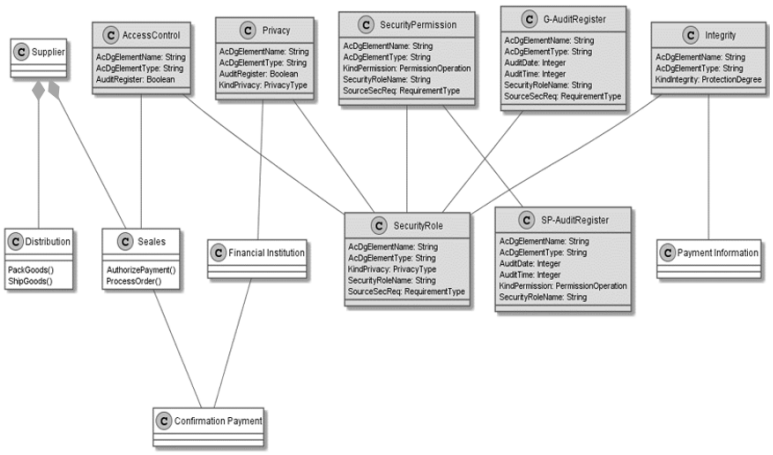


Figure 8: Model A: UML Class Diagram Obtained with the Proposal from [Rodríguez, 10]

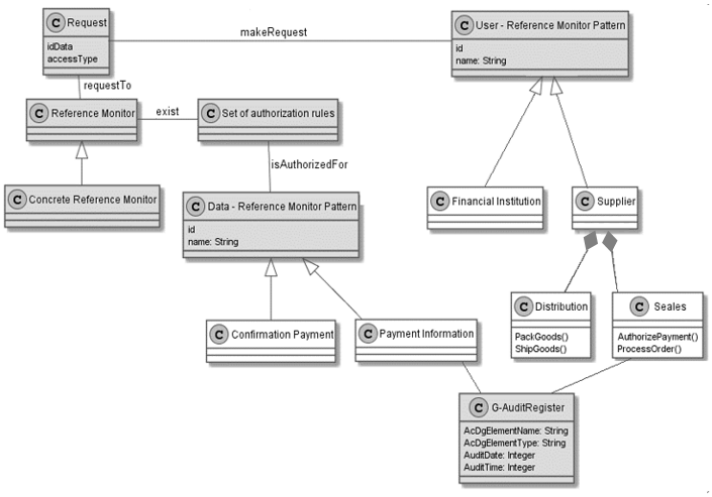


Figure 9: Model B: UML Class Diagram obtained with M-SecBP&P

The results of the survey's statistical analysis show that both models possess acceptable levels in regards to the dependent variables. However, when comparing the

generated models, a tendency of the Model B over the Model A in regards to security aspects completeness can be observed. This means that, according to the survey's respondents, the concepts of the Secure Business Process are easier to understand in the generated Class Diagram (Model B). In regards to comprehensibility, respondents perceive a similar level both for Model B and Model A. Finally, in regards to the usefulness for developing a system a big difference can be observed, as 70% of the respondents consider that the security aspects details given by Model B are better than those given by Model A in order to start developing a system.

The preference for Model B can be explained in the sense that the patterns are designed so that any developer can understand their structure's functioning, since they represent conceptual solutions to security problems.

Another interesting result is that no statistically significant relation could be detected between the respondents with knowledge on Security Patterns and those without. This enforces the idea that using Security Patterns in early stages of the development cycle, even if there is no security expert, is an adequate approach.

In regards to possible threats to the research's validity it is possible to mention the following: In regards to external validity, although the number of subjects was not as big, these subjects do represent the kind of professionals that this proposal is aimed for. There are plans on applying the quasi-experiment on a greater number of subjects in the future, in order to improve this aspect. In regards to internal validity, measures such as splitting the subject groups in two groups who analyzed the Class Diagrams in distinct order were taken, with the objective of avoiding the possibility that the order would influence their answers. In regards to construct validity, the selected measurements are often seen used for measuring effectiveness. Finally, in regards to conclusion validity, appropriate statistical tests were performed based on the data's nature.

7 Conclusions and Future Work

This article proposes the creation of UML Class diagrams, which are useful in software development, by combining Secure Business Process specifications together with Security Patterns. In order to achieve this, the M-SecBP&P method, which organizes and systematizes the required activities for achieving such a transformation, has been created.

A tool prototype that implements the tasks considered by this method has been created. The tool prototype allows the analysis of the Business Process Model and the selection of suitable Security Patterns in order to generate the desired UML Class Diagram, either as an image or as a standard format.

Previous work proposes the transformation from Secure Business Processes into UML Class Diagrams. It is important to consider, however, the usage of Security Patterns before generating the UML Class Diagram, as these patterns allow an adequate representation of the best practices and experiences of experts, materializing mechanisms that protect the confidentiality, integrity and availability of the information within a system.

The improvements of BPSecBP&P over the previous proposal have been validated through a quasi-experiment oriented to demonstrating that the UML classes model generated through the M-SecBP&P proposal is more complete and easier to

understand than one generated through a pattern-less method. User's perceptions have been satisfactory in regards to the generated model's completeness, comprehensibility and usefulness, demonstrating its adequacy as a starting point for the construction of a system. Additionally, it was detected that there was no statistically significant relation between users with knowledge on Security Patterns and those without, which reinforces the idea that the usage of Security Patterns is an adequate approach.

A Systematic Literature Review was conducted for this research. Its results reveal the absence of works that directly obtain UML Class Diagrams from Secure Business Processes with the use of Security Patterns.

Future work considers three aspects; (i) the ability to select more than one security pattern in order to generate an even better and more complete UML Class Diagram; (ii) include more security requirements into the BPSecBP&P method and (ii) to implement our proposal in real scenarios in order to improve the tool prototype.

Acknowledgments

This research is part of the following projects: BuPERG (DIUBB 152419 G/EF) and DIUBB 144319 2/R, both supported by "Dirección de Investigación de la Universidad del Bío-Bío".

References

[Ahmed, 14] Ahmed, N., & Matulevičius, R. : Securing business processes using security risk-oriented patterns Computer Standards & Interfaces. 36(4), 723–733, (2014)

[Argyropoulos, 17] Argyropoulos N., Mouratidis H. & Fish A.: Supporting Secure Business Process Design via Security Process Patterns. Enterprise, Business-Process and Information Systems Modeling. S.l.: Springer, pp. 19-33., (2017)

[ATL, 17] http://www.eclipse.org/atl/

[Awad, 15] Awad, A., Barnawi, A., Elgammal, A., Elshawi, R., Almalaise, A., & Sakr, S. : Runtime Detection of Business Process Compliance Violations: An Approach based on Anti Patterns In Proceedings of the 30th ACM/SIGAPP Symposium On Applied Computing - Enterprise Engineering Track (SAC 2015), Salamanca, Spain. (2015)

[Basin, 06] Basin, D., Doser, J., & Lodderstedt, T. : Model driven security: From UML models to access control infrastructures ACM Transactions on Software Engineering and Methodology (TOSEM). 15(1), 39–91, (2006)

[Bonillo, 06] Bonillo, P. : Metodología para la gerencia de los procesos del negocio sustenda en el uso de patrones Journal of Information Systems and Technology Management. 3(2), 143–162, (2006)

[BPMN, 15] BPMN Drawing Examples, Available since October 29, 2009; http://www.bpmn.org. (2015)

[Brambilla, 12] Brambilla, M., Fraternali, P., & Vaca, C. : BPMN and design patterns

for engineering social BPM solutions In Business Process Management Workshops. (pp. 219–230) (2012)

[Elgammal, 14] Elgammal, A., Turetken, O., van den Heuvel, W.-J., & Papazoglou, M. : Formalizing and appling compliance patterns for business process compliance Software & Systems Modeling. 1–28, (2014)

[Fernandez-Buglioni, 13] Fernandez-Buglioni, E. : Security patterns in practice: designing secure architectures using software patterns. John Wiley & Sons (2013).

[Firesmith, 04] Firesmith, D. : Specifying Reusable Security Requirements Journal of Object Technology. 3, 61–75, (2004)

[Forster, 07] Forster, A., Engels, G., Schattkowsky, T., & Van Der Straeten, R. : Verification of business process quality constraints based on visual process patterns In Theoretical Aspects of Software Engineering. First Joint IEEE/IFIP Symposium. (pp. 197–208) (2007)

[Gschwind, 08] Gschwind, T., Koehler, J., & Wong, J. : Applying patterns during business process modeling In Business process management. Springer. (pp. 4–19) (2008)

[Herrmann, 06] Herrmann, P., & Herrmann, G. : Security requirement analysis of business processes Electronic Commerce Research. 6(3–4), 305–335, (2006)

[Jürjens, 02] Jürjens, J. : UMLsec: Extending UML for secure systems development In UML 2002 — The Unified Modeling Language. Springer. (pp. 412–425) (2002)

[Khan, 12] Khan, N. H. APattern-Based Development of Secure Business Processes. Master's Thesis, University of Tartu. (2012)

[Kitchenham, 09] Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. : Systematic literature reviews in software engineering – A systematic literature review Information and Software Technology. http://doi.org/10.1016/j.infsof.2008.09.009, 51(1), 7–15, (2009)

[Lohrmann, 15] Lohrmann, M., & Reichert, M. : Effective application of process improvement patterns to business processes Software & Systems Modeling. 1–23, (2015)

[Mellor, 02] Mellor, S. J., Scott, K., Uhl, A., & Weise, D. : Model-driven architecture In Advances in Object-Oriented Information Systems. Springer. (pp. 290–297) (2002)

[Mülle, 11] Mülle, J., von Stackelberg, S., & Böhm, K. A security language for BPMN process models. KIT, University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association. (2011)

[OMG, 15] OMG : Business Process Model and Notation. Retrieved from http://www.bpmn.org (2015)

[Quirchmayr, 04] Quirchmayr, G. : Survivability and business continuity management In Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation-Volume 32. (pp. 3–6) (2004)

[Rodríguez, 10] Rodríguez, A., de Guzmán, I. G.-R., Fernández-Medina, E., & Piattini, M. : Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach Information and Software Technology. 52(9), 945–971, (2010)

[Rodríguez, 06] Rodríguez, A., Fernández-Medina, E., & Piattini, M. : Towards a UML 2.0 extension for the modeling of security requirements in business processes In Trust and Privacy in Digital Business. Springer. (pp. 51–61) (2006)

[Rodríguez, 07] Rodríguez, A., Fernández-Medina, E., & Piattini, M. : A BPMN extension for the modeling of security requirements in business processes IEICE Transactions on Information and Systems. 90(4), 745–752, (2007)

[Rosado, 06] Rosado G., Gutiérrez C., Fernández-Medina E. & Piatini M.: Security patterns and requirements for internet-based applications. Internet research, vol. 16, no. 5, pp. 519-536., (2006)

[Samarütel, 16] Samarütel S., Matulevičius R., Norta A. & Noukas R.: Securing Airline Turnaround Processes using Security-Risk Oriented Patterns. University of Tartu. In: Horkoff J., Jeusfeld M., Persson A. (eds) The Practice of Enterprise Modeling. PoEM 2016. Lecture Notes in Business Information Processing, vol 267. Springer, Cham. (2016)

[Schumacher, 13] Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P. Security Patterns: Integrating security and systems engineering. John Wiley & Sons. (2013)

[Schumm, 10a] Schumm, D., Anstett, T., Leymann, F., & Schleicher, D. : Applicability of Process Viewing Patterns in Business Process Management In Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International. (pp. 79–88) (2010)

[Schumm, 10b] Schumm, D., Leymann, F., & Streule, A. : Process viewing patterns In Enterprise Distributed Object Computing Conference (EDOC), 2010 14th IEEE International. (pp. 89–98) (2010)

[Solinas, 09] Solinas, M., Fernandez, E. B., & Antonelli, L. : Embedding security patterns into a domain model In Database and Expert Systems Application, 2009. DEXA'09. 20th International Workshop. (pp. 176–180) (2009)

[Varela-Vaca, 16] Varela-Vaca: OPBUS: A framework for improving the dependability of risk-aware business processes. AI Communications, vol. 29, no. 1, pp. 233-235.(2016)

[Wohlin, 12] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. *Experimentation in software engineering*. Springer Science & Business Media. (2012)

[Wolter, 08] Wolter, C., Menzel, M., & Meinel, C. : Modeling Security Goals in Business Processes. In Modellierung. (Vol. 127, pp. 201–216) (2008)