Software and Systems Modeling manuscript No.
(will be inserted by the editor)

Enhancing Secure Business Process Design with Security Process Patterns

Nikolaos Argyropoulos · Haralambos Mouratidis · Andrew Fish

Received: date / Accepted: date

Abstract Security is an important non-functional characteristic of the business processes used by organisations for the coordination of their activities. Nevertheless, the implementation of security at the business process level can be challenging due to the limited security expertise of process designers and the delayed consideration of security. Security patterns can, therefore, be a useful tool, as they capture expert knowledge and proven solutions and can be integrated to business processes with minimal security-related knowledge. Moreover, security requirements engineering approaches have been established as a structured way of reasoning about security during their early development stages of information systems and can, thus, provide valuable input during process design. This work introduces a set of process-level security patterns which are used to enhance an existing framework for the design of secure business processes. Through the framework's application, a system's composition as well as its security requirements are elicited, using security-oriented goal models. The goal model is then automatically transformed into a complete business process model, the security of which is implemented by the introduced set of security process patterns. Thus, the patterns introduced in this work, enhance the framework's functionality by providing a structured way of operationalising security at the business process level of abstraction. The framework is applied to a real-life information system and the effectiveness and usability of the proposed patterns is evaluated via a workshop-based experiment. The evaluation indicates that non-experts are able comprehend and utilise the developed patterns to construct secure business process designs.

Keywords Security Requirements Engineering · Business Process Modelling · Security Process Patterns · Business Process Security

N. Argyropoulos, H. Mouratidis, A. Fish
Centre for Secure, Intelligent and Usable Systems,
School of Computing, Engineering and Mathematics,
University of Brighton, Brighton, United Kingdom
E-mail: {n.argyropoulos, h.mouratidis, andrew.fish}@brighton.ac.uk

1 Introduction

Business processes are essential instruments utilised by organisations for the coordination of their activities in order to produce value in the form of products and services [42]. Information systems often play an essential role in supporting the execution of business processes. Therefore, it is critical that such systems are designed in a structured manner, taking into account the functional characteristics of the processes they will support.

The elicitation of information systems' requirements is often conducted at an organisational level of abstraction in order to capture not only technical but also social aspects of a system's environment (i.e., participating stakeholders, interdependencies, strategies). Goal modelling is a very common approach for the elicitation of systems' requirements, as goal modelling languages are equipped with high-level concepts (e.g., actors, goals, dependencies) essential for such type of system analysis. Nevertheless, business process models are designed in a lower, operational level of abstraction, as their main purpose is to capture the choreography of activities between different process participants. Therefore, bridging the abstraction gap between organisational level goal models and operational level business process models can be a challenge.

Aligning system requirements, as captured by goal models at the organisational level, with process activities at the operational level, provides traceability between system models of different abstraction levels [14]. Additionally, it helps provide justification for design choices, as decisions regarding the structure and contents of the process can be linked to the satisfaction of specific goals. Therefore, such an alignment leads to more robust and context-aware operationalisations of security at the business process level [34]. Therefore, to enhance the alignment between an organisation's strategic goals and its operations, there needs to be a well-defined interconnection between a system's requirements and its process models. For discussion on goal-oriented requirements engineering (GORE) and related topics, see [15].

In addition to the functional characteristics of a business process, there also exists a number of non-functional aspects that need to be taken into consideration. Security is one of the most important of such non-functional aspects due to the potential impact of its shortcomings for organisations in terms of finances, reputation and legal compliance [29]. Since the consideration of security during the early design stages of systems is considered highly beneficial [24], specialised security-oriented extensions have been developed for the majority of the established process modelling languages. Nevertheless, capturing the context and rationale behind general and security-related design choices made during process design, is outside of the scope of process modelling languages [13].

Another obstacle in the design of secure business processes is the disconnect between security experts and the system developers [22]. Since the main concern of system developers is functionality, security is underprioritised and implemented in an ad-hoc manner during the later development stages. Security patterns are often utilised as a way to overcome such issues, as they are able to provide to non-experts with standardised and proven solutions to common security-related issues [16]. Patterns can encapsulate security expertise and standardise proven solutions to recurring problems [22], which can facilitate a systematic and structured approach towards the operationalisation of security by non-experts [28]. However, the issue of over- and underspecification of security patterns often makes it difficult for non-experts to identify and integrate patterns of the appropriate abstraction level to their business process models.

To address the aforementioned challenges, this work extends a framework, developed in our previous work [7,4], with a security process pattern library. The framework supports the process of creating secure business process models, using security-oriented goal models as the starting point. The introduced patterns are integrated into business process models to operationalise different types of security requirements (e.g., confidentiality, integrity, availability) and are expressed at an abstraction level which is generic enough to be able to be instantiated by different types of security implementing technologies. Therefore, the developed framework facilitates (i) the elicitation of functional and security requirements of information systems at an organisational level of abstraction using goal models, (ii) the creation of business process model skeletons sourcing from the information captured at the goal model level and (iii) the refinement of the business process model skeletons to secure business process models using a set of security process patterns. A supervised workshop session, followed by a questionnaire is used to evaluate the perceived usability of the introduced patterns and compare them to ad-hoc approaches. Moreover, the overall framework is applied to a real life information system as a proof-of-concept. This paper significantly improves and extends an earlier version [9], by presenting: (i) a more specific conceptual basis upon which our approach was constructed, together with formal specifications of the transformations; (ii) a real case study and its impact upon the introduced patterns; and (iii) an extended discussion and lessons learnt from the evaluation of the approach.

The rest of this paper is structured as follows; Section 2 introduces our security process patterns and then Section 3 presents, via a real-life example, a framework that utilises them for the creation of secure business process designs. Section 4 presents the evaluation of the proposed set of security process patterns, while Section 5 compares the contributions of our work to related literature. Finally, Section 6 concludes with a short discussion of this work and its future directions.

2 Security Process Patterns

A pattern, in the context of software development, is a reusable package which incorporates expert knowledge and represents a recurring structure, activity, behaviour or design [44]. A security pattern is a well-understood solution to a recurring information security problem and can be expressed either as a structural pattern, which incorporates designs that can be implemented in the final product or a procedural pattern, which represent high level directions for improving the process of developing security-critical software systems [22]. During the requirements and analysis phases of the system development lifecycle, the majority of the proposed design pattern focus on security attacks while patterns for implementing countermeasures are less well-represented [44]. Therefore, as part of this work we introduce a number of structural process design patterns aiming to facilitate the operationalisation of countermeasures for the main types of security requirements, at a business process model level of abstraction. Such patterns are generic enough to be implementation-agnostic but able to specify a basic sequence of activities and interactions between process participants which can lead to the satisfaction of a system's security requirements. The use of patterns can be considered within the context of requirements reuse (see [41]).

The basic structure of each of the proposed patterns is captured using BPMN collaboration diagrams [31] and includes the activities required for the operationalisation of a security implementing technology. Definitions from international standards [19, 40] for each type of security requirement (i.e., authentication, authorisation, confidentiality, integrity, availability) were utilised to identify the basic functionality that each pattern should describe. Furthermore, literature sources (i.e., [43, 12] were utilised to identify how such functionality can be expressed in the context of a business process model.

The security-implementing activities included in each pattern are annotated with a padlock symbol at their top left corner to visually communicate their security-oriented nature. Corresponding activities exist at the user's lane describing any required interaction with the system's security implementing activities (e.g., username and password input). The security-constrained activity or data object, which created the need for the implementation of security, is marked with a bold black border in order to be easily distinguishable from other activities or objects. A series of message exchanges between the two lanes are also included to capture the communication between the user and system side during the interaction with the various mechanisms and for communicating the success or failure of the operation (e.g., *"Access Granted"*). Finally relevant start and end events along with gateways that split the process flow are also modelled within each pattern. An overview of the BPMN 2.0 concepts utilised for the construction of the patterns is presented in Fig. 1.

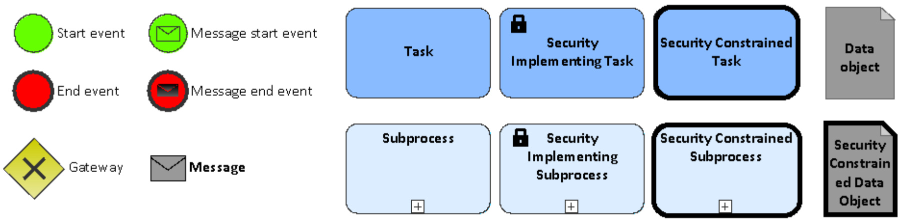
Start event
Message start event
End event
Message end event
Gateway
Message
Task
Security Implementing Task
Security Constrained Task
Data object
Subprocess
Security Implementing Subprocess
Security Constrained Subprocess
Security Constrained Data Object

Fig. 1 Overview of BPMN 2.0 elements used in patterns

The activities contained within each pattern are not dependent on the implementation of a specific mechanism but rather on the type of the security requirement at hand. Therefore, the pattern operationalising a specific type of security requirement (e.g., authentication) can be instantiated by a number of different mechanisms (e.g., smartcard, biometrics, username/password). It is also the case that one pattern can be reused within another pattern. For instance, the pattern for Authentication is reused within the Authorisation pattern since its functionality is required for the completion of the authorisation process.

2.1 Authentication

Authentication, in the context of a business process, entails the verification of a credential of a subject using security mechanisms [43]. Therefore, a process participant is required to have a verified identity before performing a specific activity or accessing a resource. To realize the authentication requirement, as illustrated in Fig. 2, every time a user submits a request to the system for accessing an authentication-constrained resource or activity, the system should check that request and ask for the user's authentication data. Once the user submits the authentication data in the appropriate form (e.g., username/password, biometric data) the system should check its validity and, if valid, allow the user to access to the constraint resource or activity.

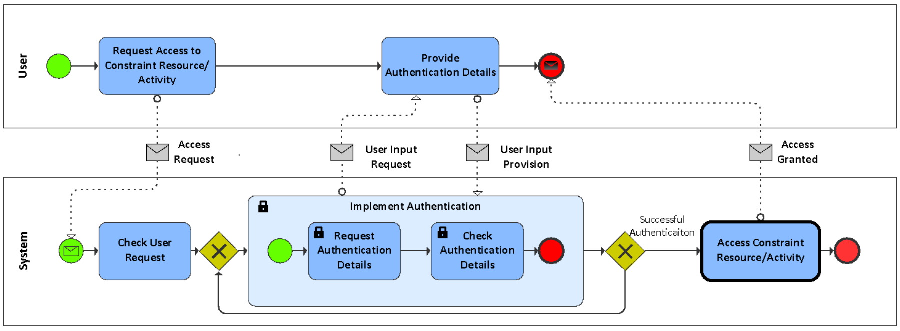
User
Request Access to Constraint Resource/ Activity
Provide Authentication Details
Access Request
User Input Request
User Input Provision
Access Granted
System
Check User Request
Implement Authentication
Request Authentication Details
Check Authentication Details
Successful Authenticaiton
Access Constraint Resource/Activity

Fig. 2 Authentication pattern

2.2 Authorisation

Authorisation, in terms of a business process model, requires the restriction of access to assets based on certain business or security requirements of an entity [19]. Therefore, only process participants with the appropriate permissions can access a resource or perform an activity that is authorisation-constrained. As shown in Fig. 3, to realise the authorisation requirement, first a user requests access to authorisation-constrained activities or resources and the authentication process takes place in order for the user's identity to become known to the system. After the successful authentication, the role and/or the permissions attached to the user's account are checked and, if appropriate, the user gains access to the constraint activity or data object.

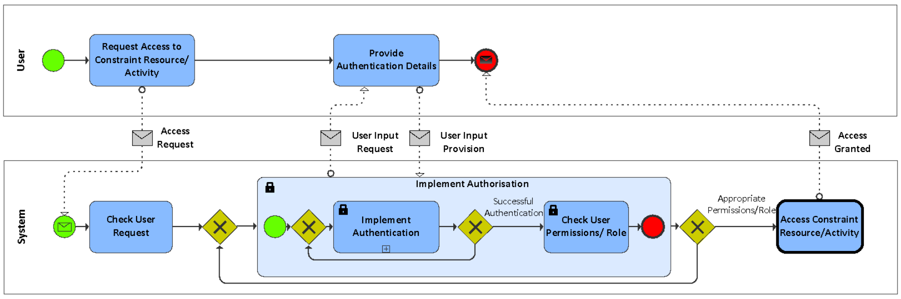
User
Request Access to Constraint Resource/ Activity
Provide Authentication Details
Access Request
User Input Request
User Input Provision
Access Granted
System
Check User Request
Implement Authorisation
Implement Authentication
Successful Authentication
Check User Permissions/ Role
Appropriate Permissions/Role
Access Constraint Resource/Activity

Fig. 3 Authorisation pattern

2.3 Confidentiality

Confidentiality, in terms of business process models, is a property of a data object and involves the identification of authorised entities that can access it [12]. As shown in Fig. 4, to achieve confidentiality in a business process, if the user is not already authorised, the authorisation process takes place as previously described. Next, a secure communication channel is created between the user and the system through which the confidentiality-constrained data object can be transferred.

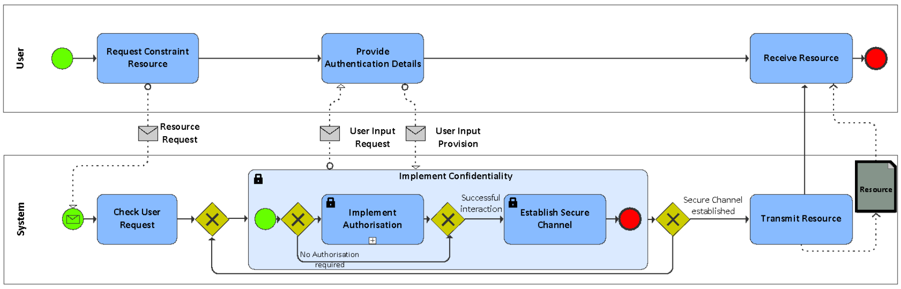

Fig. 4 Confidentiality pattern

2.4 Integrity

Integrity is concerned with ensuring that information is protected from improper modifications so as to avoid intentional or accidental unauthorised changes to system data [40]. As illustrated in Fig. 5, to achieve integrity, after an integrity-constrained data object has been transferred to the system, the system's copy of the resource needs to be compared to the original by data validation techniques.

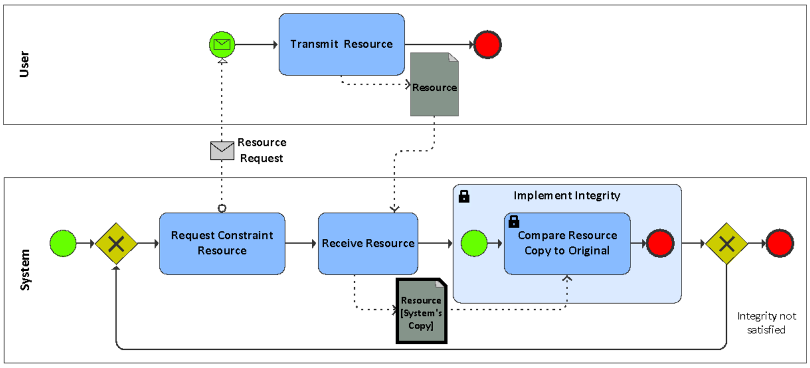

Fig. 5 Integrity pattern

2.5 Availability

Availability describes the property of system resources being accessible and usable upon demand by an entity [19]. Therefore, the pattern for availability, presented in Fig. 6, is utilised to ensure that critical resources are always available to process participants. To realise that requirement, when a requested resource is not available, the system has to maintain backups, using a number of available implementation technologies, from which the data object can be retrieved and be made available to the user.

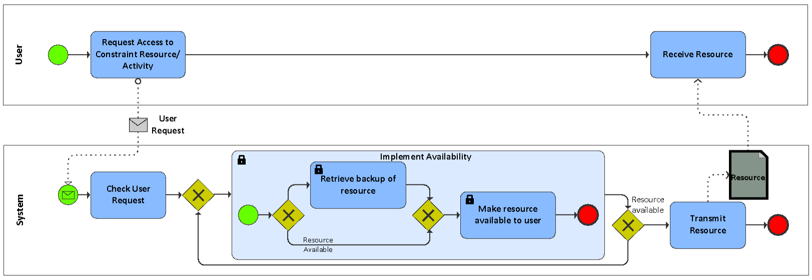

Fig. 6 Availability pattern

3 Secure Business Process Design Framework

The introduced process patterns are an important component of a broader framework that facilitates the creation of secure business process designs. Initial descriptions of some of the framework components have been introduced in our previous work [7,4,6] and described in Fig. 7. In this paper we enhance this framework by incorporating the developed patterns as a means for guiding the operationalisation of security at the business process level.

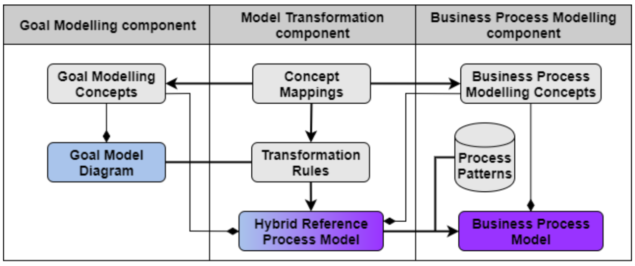

Fig. 7 Components of secure business process design framework

More specifically, the final output resulting from the application of this framework is a business process model which contains both functional and security implementing activities. Such security implementing activities originate from analysis performed at a high level organisational view of the system captured via goal-models, using a security requirements engineering approach. The goal model, capturing participating actors, their goals, tasks and resources, apart from facilitating the elicitation of security requirements, will also provide a means of automatically producing a business process skeleton via a set of model transformation rules. This process skeleton can finally be refined into a complete and secure business process model using the security process patterns introduced in Section 2.

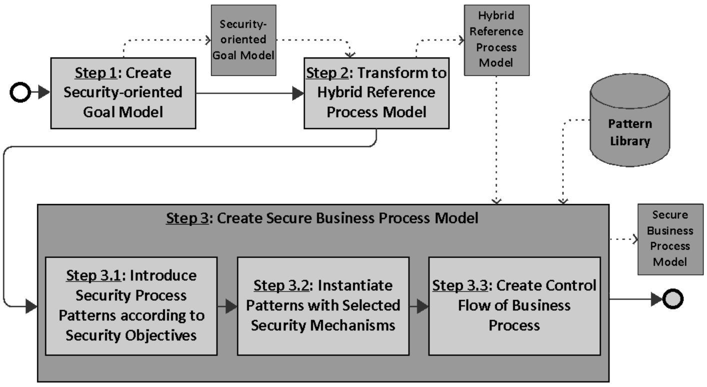
Security-oriented Goal Model
Hybrid Reference Process Model
Step 1: Create Security-oriented Goal Model
Step 2: Transform to Hybrid Reference Process Model
Pattern Library
Step 3: Create Secure Business Process Model
Secure Business Process Model
Step 3.1: Introduce Security Process Patterns according to Security Objectives
Step 3.2: Instantiate Patterns with Selected Security Mechanisms
Step 3.3: Create Control Flow of Business Process

Fig. 8 Steps for the application of framework

The steps for the application of our framework are described in Fig. 8. *Step 1* uses the *Goal Modelling component* to create a security-oriented goal model that captures a high abstraction view of the system to-be. Once such model has been created, a series of model transformation steps are applied in *Step 2* using the *Model Transformation component* to create a hybrid reference process model. That model acts as a mid-level artefact which maps security requirements and proposed countermeasures to specific parts of the business process, thus creating a security-annotated process skeleton. *Step 3* makes use of the *Business Process Modelling component* to refine the hybrid reference process model in order to create the final output of our framework, a secure business process model. During this step, the proposed security process patterns are integrated *(Step 3.1)* and instantiated *(Step 3.2)* in the final business process model and the process flow is manually constructed *(Step 3.3)*.

A more detailed overview of the framework's application, incorporating the security patterns introduced in Section 2, will be demonstrated via a working example of the Greek electronic prescription system[1]. The purpose of that system is to facilitate the creation and archiving of electronic prescriptions created by medical practitioners and used by patients to receive medication. The system's functionality and security requirements are described in the relevant act of the Greek Parliament [17], but for the purposes of this example a simplified version of the system was modelled, containing only a subset of its original specifications. The selection of the particular system for the application of the framework was based on its socio-technical nature, since it involves both human (e.g., patients, medical practitioners) and information systems (e.g., e-Prescription platform) as participants and has some security-critical aspects, since it deals with sensitive information exchanges between its participants (e.g., medical records, treatment plans). Additionally, the portion of the system modelled through the framework's application is complex enough to demonstrate the value added by the application of the proposed security patterns.

The application of each of the framework's components creates a different abstraction level model of the system, which is then used as input for the next component, with the secure business process model of the e-prescription process being the final output.

3.1 Goal Modelling component

Secure Tropos [27] is a security-oriented extension of Tropos [11], a goal-oriented requirements engineering method. The main motivation behind the creation of Secure Tropos was the lack of a methodology to support the capturing, analysis and reasoning of security requirements from the early stages of the development process. As such, Secure Tropos combines concepts from requirements engineering for representing general concepts and security engineering for representing security-oriented concepts, which are presented in detail in [26].

[1] https://www.e-prescription.gr/

The creation of security-oriented goal models for the elicitation of requirements, threats and implementation mechanism alternatives for the system to-be is the starting point of the framework proposed by this work. The ability of Secure Tropos to capture and analyse such concepts in an explicit and structured manner is the main reason for its selection as the method of choice for performing the organisational level modelling required by our framework. More specifically, the advantages of Secure Tropos, compared to other security-oriented GORE approaches are:

i. its ability to perform social analysis during the early requirements stage by capturing actors, their goals, resources and interdependencies,
ii. the simultaneous consideration of security along with the other requirements of the system-to-be, via the provision of a number of different modelling views, each capturing different aspects of the system's design (e.g., organisational view, security requirements view, security attacks view).
iii. the support for not only the requirements but also the design stages of the development lifecycle, through the mapping of abstract security constraints and threats to specific security implementation mechanism alternatives.

An example of a Secure Tropos Security Requirements view diagram is presented in Fig. 9. This view depicts node-link diagrams enclosed in circular containers representing system actors, with different types of nodes and connections to model both organisational and security related elements.

The entities interacting within that system, namely the *"e-Prescription system"*, the *"Medical Practitioner"* and the *"Patient"* are represented as actors. Each of them has a set of goals to achieve by interacting with each other. Their goals are decomposed into sub-goals and finally into plans which represent simple activities each actor has to perform (e.g., *"Issue prescription"*). Certain goals can be delegated to other system actors in order to be satisfied (e.g., *"Diagnose Patient"* goal is delegated to the *"Medical Practitioner"* from the *"Patient"*). Such goals are modelled as part of the dependency relationships arrows and as darker green rectangles with rounded edges at the actor that will satisfy them. Resources are also identified to represent documents created or required by plans or goals in order to be fulfilled (e.g., *"Prescription"*). Similarly to goals, resources can also be delegated from one system actor to another, in case they are required for the satisfaction of an actor's goals or plans (e.g., the *"Prescription"* resource is delegated from the *"e-Prescription system"* to the *"Patient"* for the satisfaction of the *"Fill Prescription"* plan). Such delegations are modelled through dependency relationships which include the delegated resource within the dependency relationship arrows. A delegated resource also appears as a dark yellow rectangle within the depender actor's container.

Security constraints are connected to goals, plans or resources in order to restrict their functionality in favour of achieving a security objective. For instance, in the system modelled in Fig. 9, *"Only authorised medical practitioners can issue prescriptions"* is an *"Authorisation"* type constraint, while the *"Maintain confidentiality of Patient Records and Treatment Plan"* is a

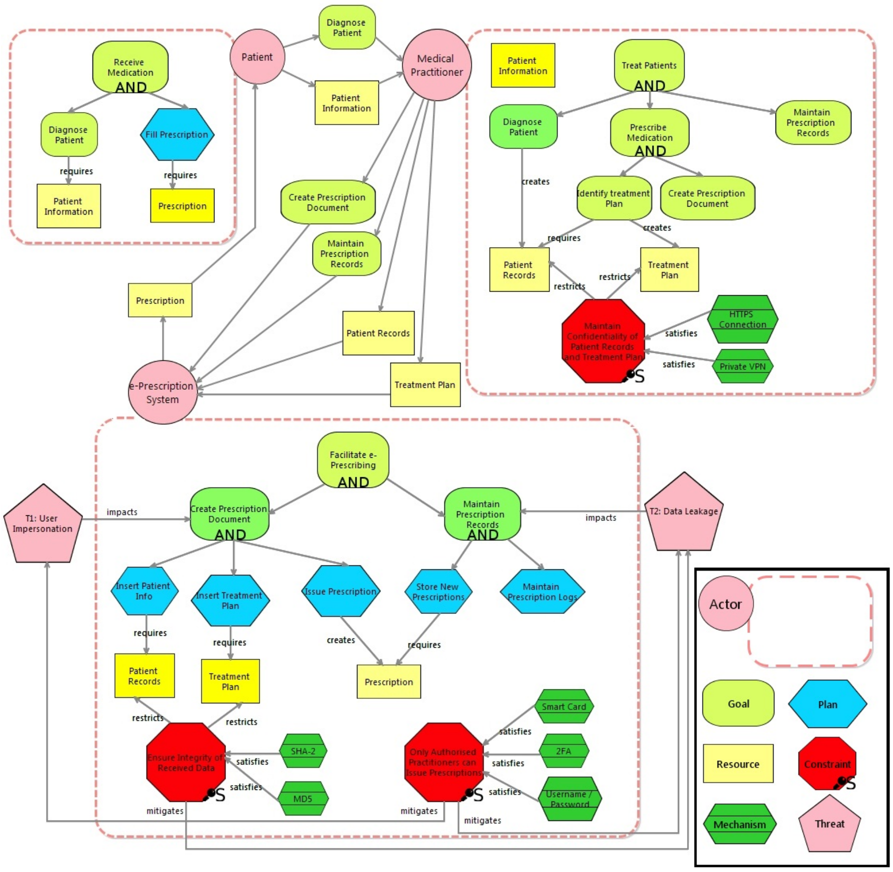

Fig. 9 Security Requirements view of e-Prescription system

"Confidentiality" type of constraint. Threats (e.g., *"User Impersonation"*) are also identified and connected to entities they can potentially impact. Since a threat can impact entities belonging to one or more system actors, they are placed outside actor's containers at the Security Requirements view diagram. To achieve the system's security objectives and mitigate the identified threats, a number of security implementing mechanisms are introduced. For example the security objective of *"Authorisation"* can be satisfied by the implementation of *"2-Factor authentication"*, *"Smart Cards"* or *"Usernames and Passwords"*. In practice, system designers and security experts are encouraged to propose any mechanism that may fit the needs of the system at this stage, since the final decision regarding the mechanisms that will be implemented in the final business process will take place at a later time.

3.2 Model Transformation component

In order to transfer elements of the organisational structure, high-level requirements and constraints captured at the goal model to the operational level, a linkage between the two levels of abstraction needs to be created. This linkage is a crucial step for the creation of operational level artefacts, in the form of business process models, which are aligned with organisational level artefacts identified through our analysis during the initial phases of this framework. To achieve that, during the model transformation phase, we introduce an intermediate model called *hybrid reference process model*. This model includes concepts from both goal and process models (*hybrid*) and captures all the security-related information elicited from the Goal Modelling and Decision Support components of the framework. For reference, the (partial) Secure Tropos metamodel used, which captures relevant relationships between concepts, is shown in Fig. 10.

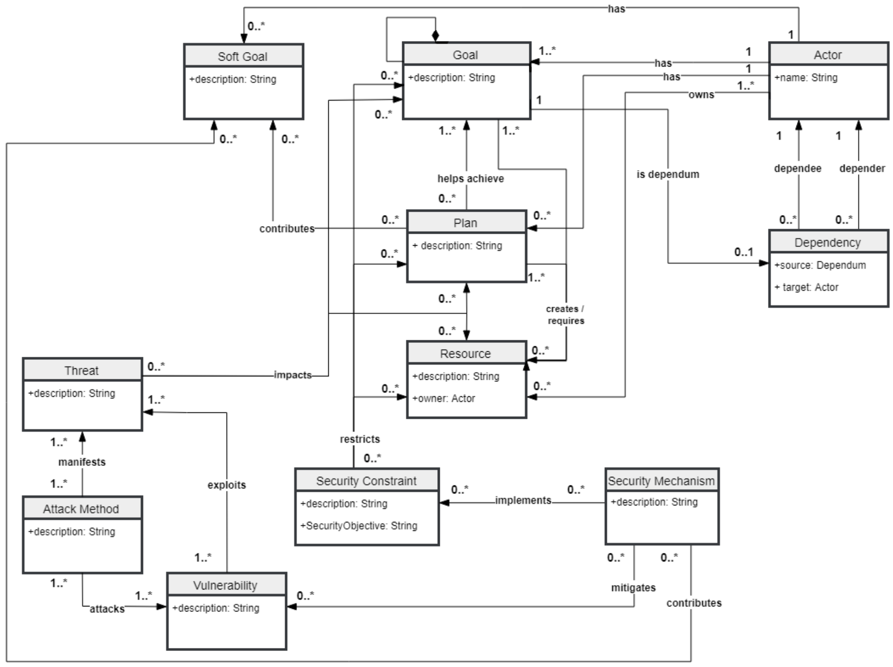

Fig. 10 Metamodel of (relevant) Secure Tropos concepts

The hybrid reference process metamodel used is shown in Fig. 11, where the concepts inherited from Secure Tropos are included in the dashed-line container. More details can be found in [3], but we note that there is a relationship change in that Security Mechanisms mitigate Threats rather than mitigating Vulnerabilities that exploit Threats, since Vulnerabilities are not needed in the hybrid. The model produced as a result of the application of the Model Transformation component can be later instantiated into a number of similar but slightly different business process models (*reference model*), according to the specific security needs of each instance.

The process related concepts (i.e., lanes, activities, data objects) included in the hybrid reference process model are transformed from their corresponding goal model concepts (i.e., actors, goals, plans, resources) and also inherit the Secure Tropos concepts capturing security-related analysis (i.e., constraints, objectives, mechanisms, threats). By capturing such connections between goal and process model level concepts via the hybrid reference process model we can trace changes at the high-level requirements of an organisation to specific parts of its business processes and vice-versa.

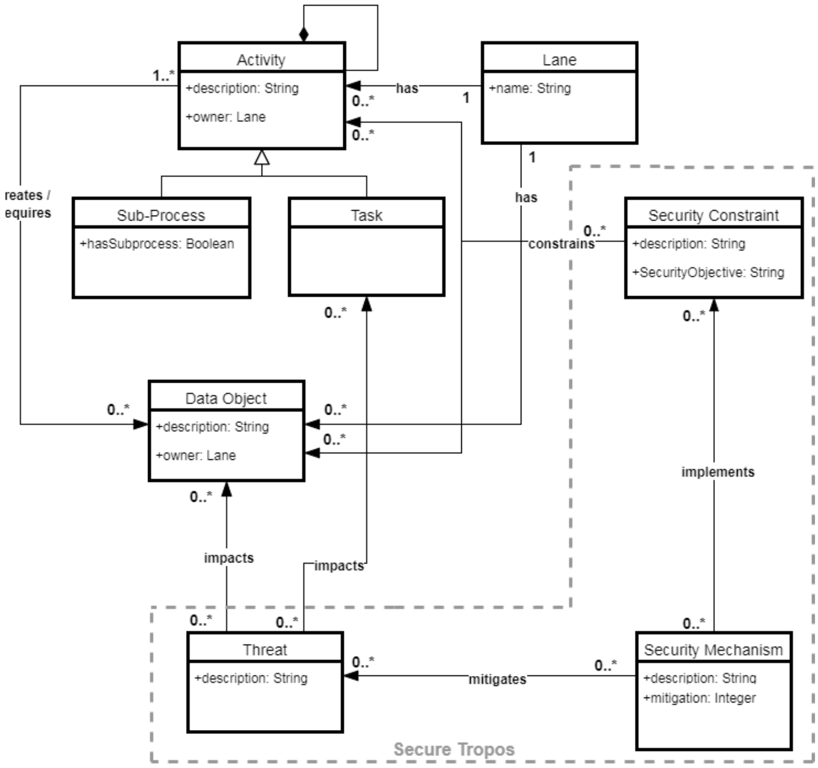

Fig. 11 Metamodel of the hybrid reference process model

To identify conceptual similarities between goal and process modelling concepts and create explicit transformation rules we use the meta-models and concepts definitions provided by Secure Tropos [27] and BPMN 2.0 [31]. More specifically, a *lane* in BPMN 2.0 is described as a container for organising and categorising activities [31], usually performed by a specific entity (e.g., process participant, information system). Since an *Actor* is also used as a container for goals and plans to be achieved by an entity in the context of goal models, we can transform the actors included in the goal model to lanes of the same name in the hybrid reference process model.

In a similar manner we can map the goals of each actors and the plans used to achieve them, to process activities. An *Activity*, according to the definition of BPMN 2.0, is a generic container for work performed by an entity [31] and can take two distinct forms, a *Sub-Process* and a *Task*. The difference between sub-processes and tasks is that the former can be broken down into a finer level of detail while the latter captures atomic activities that cannot be further decomposed. Similarly in goal models, goals are used as containers for capturing the intentions of system actors and can be further decomposed to a finer level of detail, while plans express atomic actions that need to be performed for the achievement of a goal. Thus, by transforming goals to sub-processes and plans to tasks in the hybrid reference process model, we can transfer information regarding the intentions of each actor and use them to generate the main activities to be included at the business process level.

The exchange of information assets in physical or digital form is one of the fundamental components of a business process. For this purpose the concept of *Data Objects* is included in BPMN 2.0 and defined as entities providing information which activities require in order to be performed and/or they produce as a result of their execution [31]. Similarly, at the goal model level resources are used to capture information entities which are required for or created from the fulfilment of a goal or the performance of a plan. Therefore, due to the conceptual similarities between the two concepts, the resources included in the goal model can be transformed into data objects in the hybrid reference process model. In this way, information captured at the goal model regarding data-related assets can be transferred to the business process model. A graphical depiction of the mappings described so far can be seen in Fig. 12.

Apart from the business process model concepts, the hybrid reference process model inherits a number of concepts from the Secure Tropos goal model. More specifically, concepts used to capture security aspects (i.e., security constraints, security mechanisms, threats), connected with goals, plans and resources of the goal model are transferred to the hybrid reference process model and connected to the corresponding activities and data objects.

In addition to the mappings between concepts of Secure Tropos and BPMN 2.0, a series of transformation steps can be defined (see Table 1) for automating the process of creating a hybrid reference process model starting from a security oriented goal model. Each of the transformation steps are to be applied iteratively for each of the components included in the security requirements view of the Secure Tropos goal model created during the previous phases of this framework.

We adopt a layered transformation construction, so that elements of the goal model are transformed in stages (called steps). For instance, Step 1 creates a lane for each actor, then Step 2 populates the lanes with activities (sub-processes and tasks) that are the transformation of goals of the actor

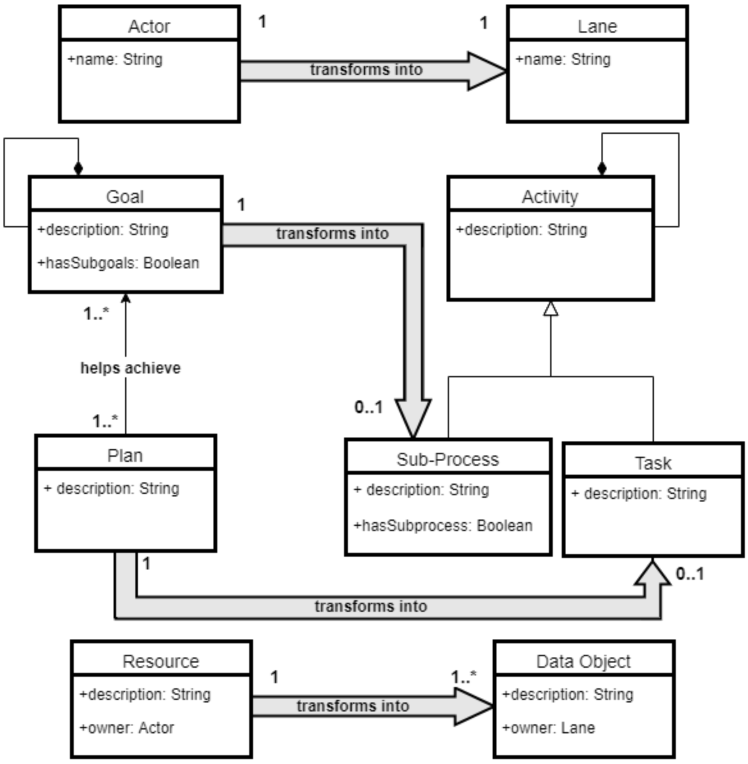

Fig. 12 Object mappings (from top to bottom): (1) actors to lanes, (2) goal and plans to activities (sub-processes and tasks), (3) resources to data objects.

for the lane. In Table 1, we provide a natural language specification of the transformation steps, followed by a formal specification and then an example application.

3.2.1 Formal specification

We provide specifications of the transformations in each Step in Table 1, subdividing some of them (e.g 4 into 4a,4b,4c) to simplify the reading. The specifications are provided in a variant of the Z-notation, similar to the one used in [10], where: a pre-condition lies above a double line and is preceded by the symbol $\triangle$, whilst the post-condition lies below the line and is preceded by $\bigtriangledown$; within the pre-condition boxes, a separator ; is used to separate parameter set memberships from other variable set memberships and to separate these from other constraints. Here, ψ denotes the transformation function that takes objects in the source goal model and creates objects in the target hybrid model; ":" indicates the type; "." is object oriented navigation; $\in$ means set membership (overloaded to also mean set containment for convenience); "|" denotes disjunction (slightly abusing notation, interpreting a

Table 1 Summary of steps for the goal-to-hybrid reference process model transformation

Step 1	each **(actor)** of the goal model is transformed into a **(lane)** in the hybrid model.
Step 2	each leaf-level **(goal)** of the goal model is transformed into a **(sub-process)** in the hybrid model. each leaf-level **(plan)** of each goal in the goal model: is transformed into a **(task)** within the sub-process for the goal in the hybrid model.
Step 3	each **(resource)** of the goal model is transformed into a **(data object)** in the hybrid model.
Step 4	each **(security constraint)**, each **(security mechanism)**, and each **(threat)** which is connected (directly or indirectly) to a goal, plan or resource of the goal model is copied in the hybrid model and is connected to the corresponding activities (sub-processes or tasks) or data objects.

matching of ordered elements in pre- and post-conditions to save space). For example, $g : Goal, a : Actor, g = a.has$ means that g is a goal, a is an actor, and a is connected to g via the has relationship, so that a has goal g, whilst $g.hasSubgoals = false$ means that the $hasSubgoals$ attribute of g is false (i.e. it is a leaf node in the graph of the model). Also, $(g|p|r) \in c.restricts$ in pre-condition and $(\psi(g)|\psi(p)|\psi(r)) \in \psi(c).restricts$ in post-condition is interpreted as "if c restricts g then $\psi(c)$ restricts $\psi(g)$, and if c restricts p then $\psi(c)$ restricts $\psi(p)$, and if c restricts r then $\psi(c)$ restricts $\psi(r)$.

Step1

$\triangle\ a : Actor$

$\triangledown\ \psi(a) : Lane$

Step2a

$\triangle\ g : Goal; a : Actor; g = a.has; g.hasSubgoals = false$

$\triangledown\ \psi(g) : Subprocess, \psi(g) = \psi(a).has$

Step2b

$\triangle\ g : Goal; p : Plan, g.hasSubgoals = false, g = p.helpsachieve$

$\triangledown\ \psi(p) : Task, (\psi(p) \in \psi(g).activity$

Step3

$\triangle\ r : Resource, , a : Actor, r.owner = a$

$\triangledown\ \psi(r) : DataObject, \psi(r).owner = \psi(a)$

Step4a

$$\triangle\ c : SecurityConstraint; (g|p|r) \in c.restricts$$

$$\nabla\ \psi(c) : SecurityConstraint; (\psi(g)|\psi(p)|\psi(r)) \in \psi(c).restricts$$

Step4b

$$\triangle\ m : SecurityMechanism;$$
$$(t \in m.mitigates.exploits | c \in m.implements)$$

$$\nabla\ \psi(m) : SecurityMechanism;$$
$$(\psi(t) \in \psi(m).mitigates | \psi(c) \in \psi(m).implements)$$

Step4c

$$\triangle\ h : Threat; (g|p|r) \in h.impacts$$

$$\nabla\ \psi(h) : Threat; (\psi(g)|\psi(p)|\psi(r)) \in \psi(h).impacts$$

3.2.2 Application

The application of the model transformation component at the e-Prescription system's goal model produces the hybrid reference process model illustrated in Fig. 13. More specifically, the actors introduced during the organisational level analysis of the system (i.e., *Patient, Medical Practitioner* and *e-Prescription System*) are transformed into business process lanes with the same name. Next, activities are created and placed in the corresponding lanes, originating from the leaf-level goals and plans of each system actor. For instance the *"Diagnose Patient"* leaf-level goal is transformed into a sub-process with the same name in the *Medical Practitioner*'s lane. In a similar manner, the relevant resources (i.e., *"Prescription"*), previously introduced at the goal model, result in data objects in the hybrid reference process model, connected as inputs or outputs to the activities that create or require them. For instance, since the *"Prescription"* resource is created by the plan *"Issue Prescription"* at the goal model level, a data object with the same name is the output of the corresponding activity in the hybrid reference process model.

The constraints connected to a goal, plan or resource of the goal model are now transferred into the hybrid reference process model and connected to the corresponding activity or data object (e.g., *"Only authorised medical practitioners can issue prescriptions"*) is connected to the *"Issue Prescription"* activity). The security objective that is satisfied by each security constraint will eventually help the process designers to select the appropriate process pattern which will be integrated in the final business process model. Similarly, threats identified at the goal model level are also transferred and connected to the corresponding elements of the hybrid reference model which they impact. For instance, the *"Data Leakage"* threat, impacting the goal *"Maintain*

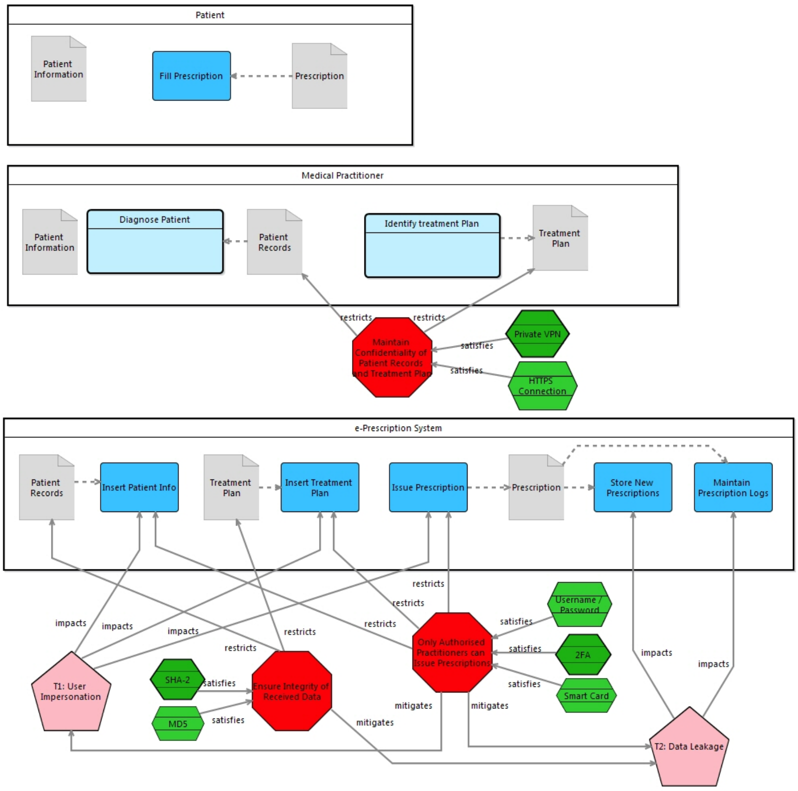

Fig. 13 Hybrid reference process model of the e-Prescription system

Prescription Records" in the goal model, is now connected to the process activities *"Store New Prescriptions"* and *"Maintain Prescription Logs"* of the hybrid reference process model, which resulted from the transformation of the leaf-level children nodes of the impacted goal.

Finally, the security mechanisms, proposed at the goal model for the implementation of each security constraint, are also transferred in the hybrid reference process model to maintain the information regarding the range of potential configurations of security countermeasures at the process level. For instance, *"Smart Cards"*, *"2-Factor Authentication"* or *"Usernames and Passwords"* are amongst the security mechanisms that can be selected for the implementation of the authorisation-related security constraint (i.e., *"Only authorised medical practitioners can issue prescriptions"*).

The model transformation process can be automatically performed using the SecTro modelling platform[2]. The platform supports the creation of all modelling views of the Secure Tropos framework, including the Security Requirements view used by the Goal Modelling component of our framework. Ad-

[2] http://www.sense-brighton.eu/research/sectro-tool/

ditionally, the transformation rules described in this section are implemented within the SecTro modelling platform, therefore allowing the creation of a hybrid reference process model in a fully automated manner.

3.3 Business Process Modelling component

The Business Process Modelling component uses the hybrid reference process model as input for creating secure business process designs. For each security-constraint activity or resource of the hybrid reference process model, one of the proposed security mechanism is selected to be integrated within the final business process model. To provide a structured approach towards security operationalisation for process designers, the Business Process Modelling component uses the security design patterns, as introduced in Section 2. Other than the security pattern integration, the Business Process Modelling component is also where the final business process model is created.

The process skeleton captured by the hybrid reference process model is refined by manually adding BPMN process elements. More specifically, after the security process patterns have been instantiated and integrated, process designers construct the control flow of the process by:

1. Manually order (since ordering information is not available in the goal model) and connect (via relationships) activities contained in the hybrid reference process model.
2. Decompose (if necessary) existing sub-processes into tasks and add any new activities desired.
3. Add any necessary getaways, start and end events within each lane and sub-process.
4. Add any additionally desired message exchanges between lanes (anything desired in addition to those introduced by the security process patterns will need to be manually added).

Fig. 14 presents the final business process model of the e-Prescription system. In the *"e-Prescription System"* lane the business process design pattern for the requirement of *"Authorisation"* (see Fig. 3), has been introduced before the security constraint activities *"Insert Treatment Plan"* and *"Issue Prescription"*, denoted with a bold-line border. The authorisation pattern has been instantiated to implement the *"2-Factor Authentication"* security mechanism. Therefore, activities of the authorisation pattern which were abstractly defined before, are now presented as more explicit declarations (i.e., *"Implement 2-Factor Authentication"*) to reflect the implementation of the selected security mechanism. The same process was followed for the *"Confidentiality"* requirement connected to the *"Patient Records"* and *"Treatment Plan"* data objects, where the pattern for *"Confidentiality"* (see, Fig. 4) has been instantiated to implement the *"HTTPS"* security mechanism. Finally, the Integrity pattern (see Fig. 5) has been instantiated with the *"SHA-2"* security mechanism and associated with the integrity-constraint data objects *"Patient Info"* and *"Treatment Plan"*.

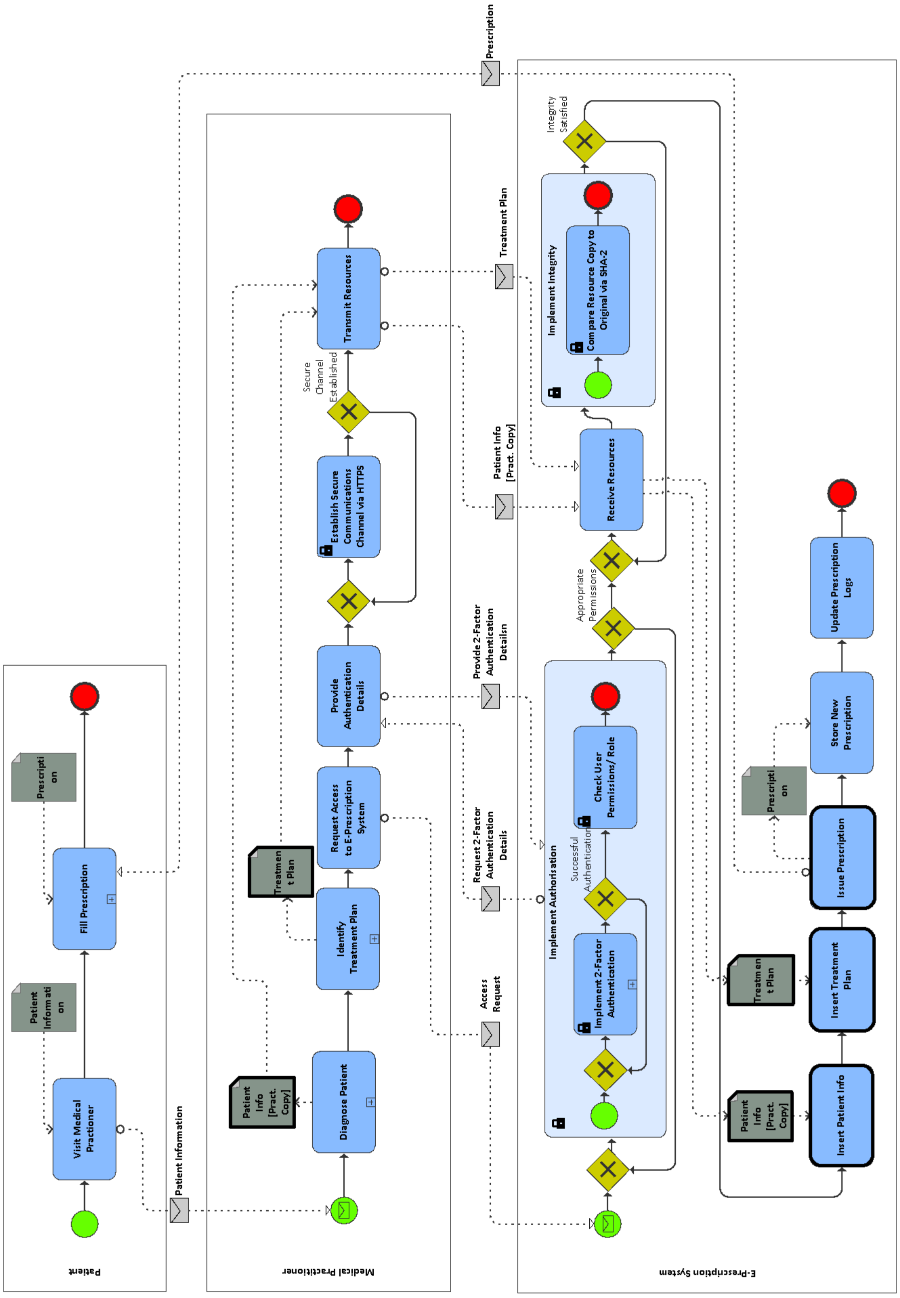

Fig. 14 Business Process Model of the e-Prescription System

Therefore, the inclusion of the security process patterns increases the amount of automation during the integration of security in a business process model, thereby reducing the overhead of process designers in terms of time, effort and prerequisite security-related expertise. Additionally, the patterns also enhance the modelling consistency by realizing the same security requirement in a consistent manner both within a model and across models.

Other than the introduction of the instantiated business process design pattern for the operationalisation of the identified security constraints, start and end events have been manually added at each lane of the final business process diagram to denote the beginning and end of each of the contained subprocesses. Additionally, message exchanges have been added between lanes for transferring relevant data objects and the activities contained within each of the model's lanes have been ordered and connected with each other to create a control flow. The ordering and connecting of activities and data objects is also a manual task since the goal model, which provided us information regarding the basic structure of the system, is inherently not equipped to capture information regarding temporal dimensions of the system, such as the order of execution of its plans.

4 Evaluation

Apart from the application of the security process patterns to the e-Prescription system, as reported in the previous sections, the work was evaluated through two different approaches. Initially, through a specific experiment focusing on the security process patterns, and secondly through the application of the security process patterns on a complex case study, as part of the Secure Business Process Framework illustrated in Fig. 7. In the rest of this section we provide details, discussion and results from those two evaluation approaches.

4.1 Experiment

An experiment was conducted in order to i) evaluate the perceived understandability and ease-of-use of the proposed security process patterns and ii) compare their implementation to ad-hoc security integration in business process models. Overall, thirty (30) postgraduate students (MSc and PhD level) from two different universities (i.e., University of Brighton, UK and Pantheon-Sorbonne University, France), in the areas of information systems design and information security, completed the experiment, in two separate supervised workshop sessions, each with a duration of approximately thirty minutes.

A brief introduction to familiarise the participants with business process modelling concepts and BPMN diagrams was provided at the beginning of each session. Next, a brief business process model, shown in Fig. 15, was presented to the participants.

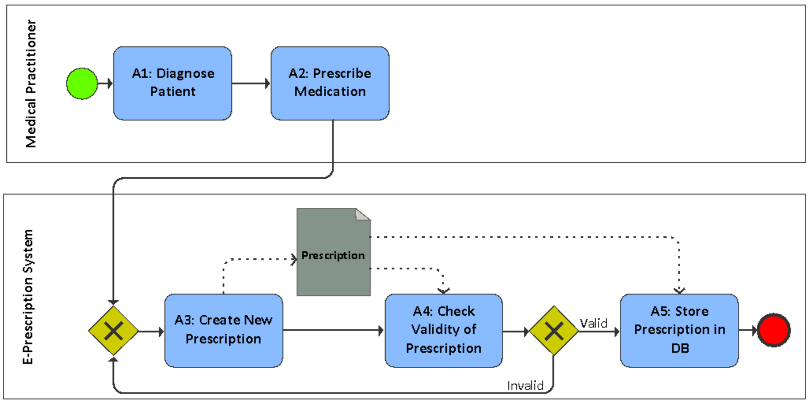

Fig. 15 Business Process Model of Evaluation Experiment

During the first scenario the participants were asked to redesign the provided process model by introducing any activities they considered necessary, in an ad-hoc manner, in order to satisfy the authentication constraint *"Only registered medical practitioners can create a new prescription"*. Only after the first scenario was completed, the participants were presented with the authentication pattern, as introduced in Fig. 2. For the completion of the second scenario, they were asked to instantiate and introduce the pattern to the business process model of Fig. 15, in order to, once again, satisfy the same security constraint.

After both parts of the experiment were completed a short questionnaire was distributed in order to capture the opinions of the participants regarding their experience. The questionnaire entries were phrased as statements accompanied by a 5-point Likert scale, ranging from strongly disagree to strongly agree, from which the responders selected the option best reflecting their opinion. The statements provided to the participants were the following:

- "I found it difficult to identify which activities I needed to add to the process model (Fig. 15) in Scenario 1."
- "I found it easier to create a business process model in Scenario 2 than in Scenario 1."
- "The contents and structure of the business process pattern (Fig. 3) were easy to understand."
- "I found it easy to integrate the business process pattern into the business process of Fig. 15."

At the end of the questionnaire form there was also the option of providing free-form comments and remarks[3].

The participants' responses to the above statements are summarised as follows:

[3] The questionnaire and a summary of the responses can be accessed in: http://www.sense-brighton.eu/process-patterns-questionnaire/

- 10 out of 30 (33%) either agreed (9) or strongly agreed (1) that it was difficult to identify the security related activities needed to be added in the process, in an ad-hoc manner.
- 15 out of 30 (50%) either agreed (10) or strongly agreed (5) that it easier to create a secure business process model using the provided process pattern compared to the ad-hoc security implementation.
- 20 out of 30 (66%) either agreed (15) or strongly agreed (5) that the provided process pattern was easy to understand,
- 18 out of 30 (60%) either agreed (13) or strongly agreed (5) that the provided process pattern was easy to integrate to the provided business process model.

4.2 e-Government System Case Study

As mentioned above, the work was evaluated as part of the Secure Business Process Framework. This section does not aim to provide details of the whole framework's application to the case study, which is outside the scope of this paper, but rather focusses on the elements of the application that are related to the security process patterns. For other aspects and analysis related to this case study please see [3].

The case study involves an e-government system (SPA) of the Municipality of Athens, Greece. More specifically, the selected system is used for the administration of swimming pool facilities used by Athenian citizens and was part of the VisiOn European project, in which the second author participated.

The case study was developed and performed in close cooperation with two analysts of DAEM S.A the organisation in charge of developing all information systems for the municipality of Athens. Both of them were experts in system analysis and design, while one of them was also a security expert. Both of them were familiar with goal modelling, security requirement elicitation with Secure Tropos and process design using BPMN due to their previous participation at the VisiOn project. The communication of the stakeholders with the author initiated during June 2017 and regular teleconferences were performed until the completion of the case study in September 2017.

The BPMN 2.0 Collaboration diagram that describes the SPA system was constructed in close cooperation with the analysts of DAEM. The automatically generated hybrid reference process model allowed us to identify: i) the basic structural characteristics of the process (lanes, activities, information objects), ii) the types of security constraints and the specific process elements they restrict and, iii) the security mechanisms to be implemented to satisfy each constraint. First the business process design patterns, were made available to the analysts. Next we matched each security constraint to its corresponding pattern. For instance an important security constraint of the SPA Certificate copies shall not be modified after issuing", was operationalized by the Integrity pattern, which in turn was instantiated by the Checksum security mechanism, as selected during the decision support process. The instantiated patterns were manually introduced into the business process diagram, for each constraint activity or data object.

Next, a manual refinement of the process model was performed which focused on introducing control flow elements, such as start and end events, gateways, additional activities and message exchanges between lanes. After some iterations, a final version of BPMN 2.0 collaboration diagram describing the functionality of the SPA system was developed, as presented in Fig. 16.

4.3 Evaluation Reflections and Lessons learned

The different evaluation activities, presented above, facilitated the refinement of the developed secure process patterns to the state presented in this paper. Each evaluation method provided valuable insights which led to the improvement of the patterns in an iterative manner.

The experiment allowed us to get an indication of the perceived usability and understandability of the proposed process patterns. It also indicated that such patterns are a preferable alternative to ad-hoc approaches, thereby confirming the literature consensus that patterns provide more structure and guidance to process designers. Another insight gained from our evaluation, especially the experiment, was that even non-experts in the area of information security were able to sensibly make use of the provided patterns in order to create consistent models within a reasonable timeframe. This indication is also aligned with literature findings, suggesting that patterns facilitate reusability and model consistency while also reducing the overhead for process designers in terms of time and prerequisite domain knowledge.

Nevertheless, the generalisability of the experiment's results is limited since the participants only worked with a small subset of the proposed patterns and a simple process model. Another aspect that has to be considered is the potential of bias introduced by learning effects, since the participants familiarized themselves with the process model of Fig. 15 during the first scenario, thus, potentially making it easier for them to apply the pattern in the same model during the second scenario. Other threats to the experimental validity include the diverse backgrounds of the participants, since their information security and business process modelling experience varied, while also English was not the native language of a number of participants. Nonetheless, to minimize the effects of such factors, the workshop sessions, during which the experiment was performed, were supervised and any participant enquiries regarding the experiment were answered.

An exit interview was arranged with the DAEM case study involved stakeholders to provide us with qualitative insights regarding the perceived applicability and effectiveness of the security process patterns. The main aim of this was to: (i) capture the analysts experiences regarding the design of the SPA business process using the security process patterns; (ii) identify what they perceived as relevant benefits and shortcomings. The Goal Question Metric (GQM) template [38] was utilized to structure each question of the interview

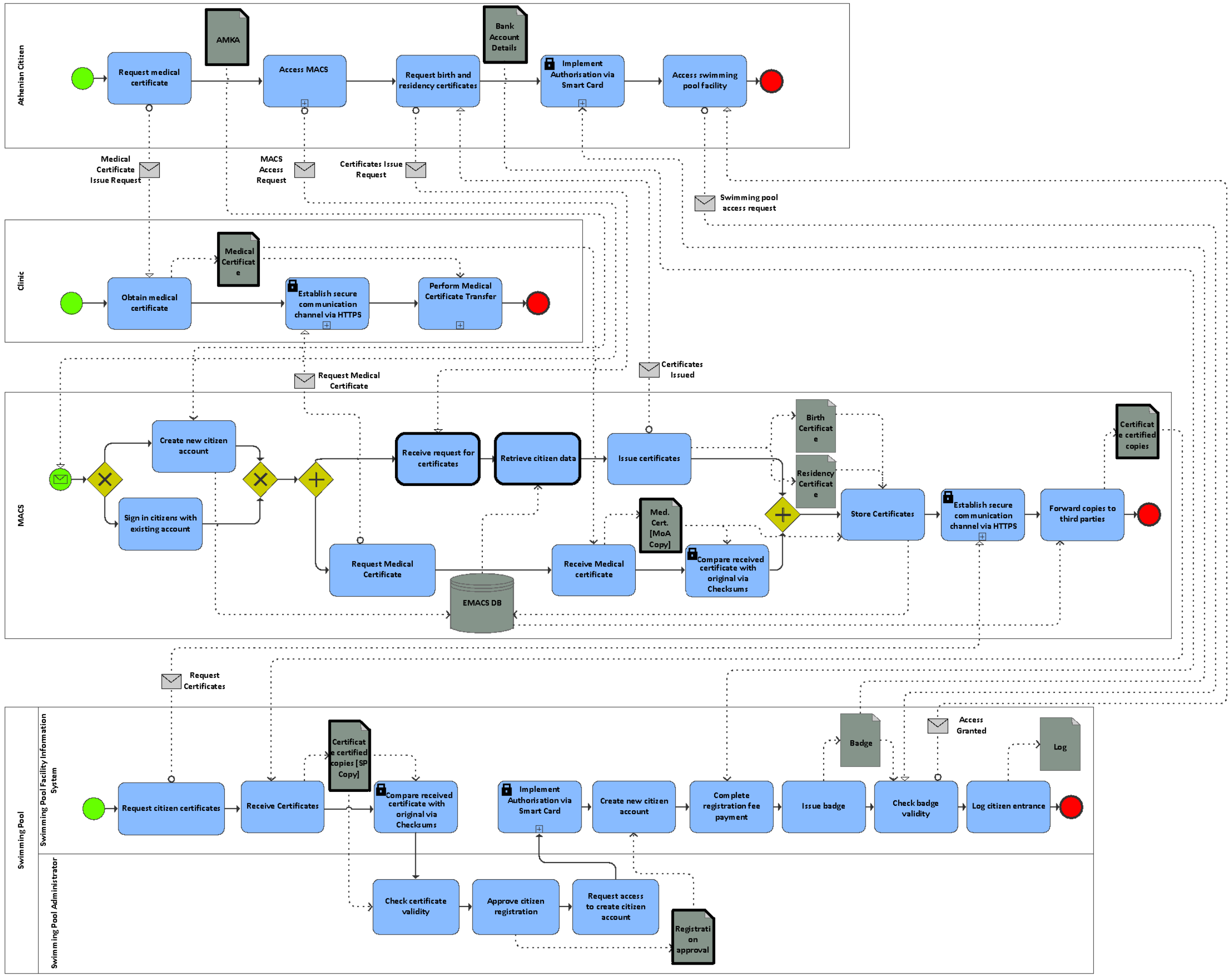
Athenian Citizen
Request medical certificate
AMKA
Access MACS
Request birth and residency certificates
Bank Account Details
Implement Authorisation via Smart Card
Access swimming pool facility
Medical Certificate Issue Request
MACS Access Request
Certificates Issue Request
Swimming pool access request
Clinic
Obtain medical certificate
Medical Certificat e
Establish secure communication channel via HTTPS
Perform Medical Certificate Transfer
Request Medical Certificate
Certificates Issued
MACS
Create new citizen account
Sign in citizens with existing account
Receive request for certificates
Retrieve citizen data
Issue certificates
Birth Certificat e
Residency Certificat e
Store Certificates
Establish secure communication channel via HTTPS
Forward copies to third parties
Certificat e certified copies
Request Medical Certificate
EMACS DB
Receive Medical certificate
Med. Cert. [MoA Copy]
Compare received certificate with original via Checksums
Request Certificates
Swimming Pool
Swimming Pool Facility Information System
Request citizen certificates
Receive Certificates
Certificat e certified copies [SP Copy]
Compare received certificate with original via Checksums
Implement Authorisation via Smart Card
Create new citizen account
Complete registration fee payment
Issue badge
Badge
Check badge validity
Access Granted
Log citizen entrance
Log
Swimming Pool Administrator
Check certificate validity
Approve citizen registration
Request access to create citizen account
Registrati on approval

Fig. 16 Business process model of the SPA system

as it allows us to specify: (i) the focus of the question, (ii) the objective of the question, (iii) the variable measured, (iv) the subjects participating and (v) the context of the question.

The following illustrates the relevant GQM for the Security process patterns:

- **Analyse** the developed security process patterns
- **for the purpose of** quantitative evaluation
- **with respect to** the perceived usability and complexity of the process patterns
- **from the point of** view of the system designers and security expert
- **in the context of** refining the security process pattern to a final version

Based on the feedback provided by the analysts, the process patterns were useful to the analysts since they provided a structured and predefined way to implement the different types of security constraints. They were also at an appropriate level of abstraction which matched the abstraction level of the final business process model. Some concerns regarding the patterns were focused on their placement within the process model, which was not always obvious, and the additional complexity they introduced to the final process model, which led to the analysts preferring to introduce them as collapsed sub-processes to keep the model manageable.

The final e-government case study, which constituted the last step of the process patterns evaluation process, facilitated the creation of the final version. In particular, the earliest version of such patterns, could only be applied to process lanes existing within the same pool. The latest version of the patterns, as presented in this paper were extended to include message exchanges across process lanes, allowing them to be applicable in a broader range of scenarios, where the participating lanes do not belong in the same pool.

5 Related Work

Kienzle et al. [22] have created a pattern repository including both structural and procedural patterns for web service security, expressed through a textual template. Mouratidis et al. [28] introduce security patterns to describe security implementing techniques (e.g., agent authenticator), expressed using Tropos, an agent-oriented software engineering approach, and a textual description template. Rosado et al. [34] link security requirements to architectural and design security patterns in order to guide the implementation of security in the area of web services. High-level architectural patterns and mid-level design patterns of security implementing mechanisms (e.g., secure message router, credential tokenizer) are matched to specific types of security requirements of web service applications. Ahmed et al. [1] identify potential risks and security requirements at the process level by matching process fragments with security-risk patterns used to capture common security requirements. A comprehensive survey of works in the area of security design patterns is provided by Laverdiére et al. [23], where a number of desirable properties of security design patterns and a template for pattern description are developed.

Salnitri et al. [35] introduce SecBPMN which extends BPMN 2.0 in order to perform security-related annotation of business processes. The security requirements captured via such annotations are formalised by a series of predicates which, similar to security process patterns, encapsulate security-related information. Li et al. [25] introduce a method for constructing goal models which are able to capture and analyse attack patterns depending on the contextual environment of the system. Kalloniatis et al. [20,21] introduce the PriS framework for the design of privacy-aware processes, starting from goal models. A set of privacy process patterns are used by PriS for the incorporation of privacy requirements into business processes, which are refined and expressed in BPMN 2.0 in [6].

The above works ([22,28,34]) provide patterns which aim to capture specific types of security countermeasures or, in the case of [1], use process patterns to identify where security-related violations can occur within the process. In contrast, each of the patterns presented in our work captures the operationalisation of one type of security requirement and can accommodate its implementation by any suitable security implementing technology. Therefore, their implementation-independent nature allows a higher degree of generalisability and flexibility compared to countermeasure-specific patterns. Moreover, similar to the works of [35,25,20,21], our framework also uses goal models but it provides explicit steps for transitioning from them to the operational level of abstraction. Thus, the added value it provides relates to its ability to map both security requirements and security countermeasures, captured at a high abstraction level, to specific business process elements via a structured and automated model transformation process. Therefore, it facilitates the alignment between security requirements at the organisational level and the operationalisation of security countermeasures at the process level.

The literature is full of examples of works related to business processes, security modelling and a combination of both. In this paper we primarily focus on presenting works that we think are directly related to the security process patterns presented above. For a full review of works related to business process modelling and security modelling, with more detailed comparison than here, please see [3], while for a review of works related to security patterns please see [30].

In [36], SecureBPEL is introduced as a process specification language emphasizing in the security aspect of business processes, aiming to bridge the gap between the early requirement analysis and the development of secure workflows. This method is essentially an extension of the BPEL execution standard enriched with constructs from the Secure Tropos goal-oriented security requirements engineering framework. Such concepts are used to enforce delegation and trust requirements in web services used to support the designed business process, thereby extending the functionalities of traditional BPEL. SecureBPEL offers a way of deriving process skeletons based on requirements specified early in the development process, which can be then refined to produce secure workflows with minimal effort.

The M-BPSec framework [33] aims to create secure business process specifications by transforming computationally independent models (CIMs) to platform independent models (PIMs) by the application of predefined transformation rules. At the CIM level, business analysts can express their security requirements at a high level of abstraction, on the business process model via a series of padlock symbols. The secure business process can either be modelled using UML activity diagrams (UML-AD) or BPMN.

In the same context of model transformation, the SECTET framework [2] is developed for the implementation of security in business process. The first step in the framework is the creation of a platform independent model (PIM) using a UML profile, called SECTET UML, to capture the initial business requirements. SECTET-PL, a domain-specific predicative language, is also introduced for the definition of security policies and is integrated with the UML modelling component of the framework. For the transition to a platform specific model (PSM) a series of transformation rules are defined in QVT. Using these rules XACML security policies can be generated from the requirements model.

The Sec-MoSC framework is another security-oriented BPMN extension introduced in [39]. Sec-MoSC aims to integrate security requirements with BPMN process models by introducing the concepts of NF-Attribute, NF-Statement and NF-Action. The NF-Attribute expresses the security requirements of a specific process fragment, the NF-Statement quantifies that requirement (e.g., High, Medium, Low) while the NF-Action models mechanisms that can be implemented to satisfy such requirements. After the security annotated model is refined it can be automatically translated to BPEL execution code with security configurations sourcing from the parameters set at the process model level.

The work presented in [32] introduces BPMN-sec, a BPMN extension focusing on the security aspect of business processes outsourced to the cloud. In BPMNsec two main types of stakeholders are involved, namely a user-side and a cloudside, each controlling different parts of the process. UML Activity Diagrams (ADs) have been the focal point of a number of security-related UML extensions. In [37] UML ADs are utilised to capture misuse cases. In such mal-activity diagrams malicious actors and their actions are modelled along with the process they negatively impact.

6 Conclusion

Designing secure business processes can be a challenging endeavour since system developers often have limited knowledge regarding the analysis and implementation of security. Process patterns, encapsulating expert knowledge and proven solutions, can be a way to overcome the lack of security expertise during a system's development process. Identifying security process patterns of the appropriate abstraction level and granularity is another challenge, since over-specified patterns may be not flexible enough to fit the specific context of the system at hand, while high-level architectural patterns may be too generic.

The contributions of this work can be summarised as follows: (i) the development of a set of security process patterns which are evaluated through a workshop-based experiment, and (ii) the integration of the developed patterns as a component to an existing framework for the design of secure business processes, the functionality of which was demonstrated through its application to an existing information system.

The proposed set of patterns can be utilised for the integration of security in business process models. Their most important characteristic is the level of abstraction at which they are expressed, as it allows them to capture the steps required for the operationalisation of security requirements at the business process level of abstraction, in an implementation-agnostic manner. The perceived usability and understandability of the proposed patterns has been positively evaluated through a workshop-based experiment. The participants of this experiment also indicated that designing secure processes via the proposed set of patterns was preferable to ad-hoc approaches to security.

Moreover, this collection of patterns was integrated into a broader framework to enhance its functionality. This framework uses Secure Tropos goal models to capture a socio-technical view of an information system in order to elicit its security requirements and propose potential security implementing mechanisms. The created goal model is automatically transformed to an intermediate process skeleton through the application of a series of transformation rules. Finally, it is refined into a secure business process model through the integration and instantiation of the proposed security design patterns and some manual refinement of its control flow. The application of the proposed framework to a real-life e-Prescription system allowed us to also demonstrate how the introduced patterns can be seamlessly integrated into structured approach. The framework application facilitated the creation of a secure business process design for the system at hand, through a series of well-defined steps, supported by established modelling languages and an automated model transformation functionality.

A key future work direction is the further refinement and extension of the proposed pattern library. In addition to that, the privacy process patterns, introduced by our previous work [6], will be added to the pattern library of our framework so it will be able to cover the analysis and operationalisation of both security and privacy countermeasures in business process models. Regarding the overall framework, new components have been developed as part of our ongoing work, providing it with additional analysis and verification capabilities. More specifically, in [5] we have presented an approach for a risk-based decision support process for the selection of security mechanisms. This approach is the basis for a decision support component that will allow process designers to select an optimal set of security mechanisms. Such mechanisms will be used to instantiate the process patterns and be integrated into the final process model. Furthermore, in [8], we introduce a series of attribute-based security verification algorithms that can be applied to a business process model and verify the satisfaction of its security requirements. The output of this work will also form a new component which will allow the security verification of the process model produced via the application of our framework.

There are other future directions of research, such as extending the automatic transformations components presented, and identify the limits of what can be automatically transformed versus parts that must be manually identified (e.g. due to information not being specified in the goal models). Within this context, one may also consider the specification of model transformations in accordance with several alternative approaches (e.g. [45, 18]). It will also be valuable to further evaluate the usability of goal modelling approaches for business process designs.

References

1. Ahmed, N., Matulevičius, R.: Securing business processes using security risk-oriented patterns. Computer Standards & Interfaces **36**(4), 723–733 (2014)
2. Alam, M.: Model driven security engineering for the realization of dynamic security requirements in collaborative systems. In: International Conference on Model Driven Engineering Languages and Systems, pp. 278–287. Springer (2006)
3. Argyropoulos, N.: Designing secure business processes from organisational goal models. Ph.D. thesis, University of Brighton (2018)
4. Argyropoulos, N., Alcañiz, L.M., Mouratidis, H., Fish, A., Rosado, D.G., de Guzmán, I.G.R., Fernández-Medina, E.: Eliciting security requirements for business processes of legacy systems. In: IFIP Working Conf. on The Practice of Enterprise Modeling, pp. 91–107. Springer (2015)
5. Argyropoulos, N., Angelopoulos, K., Mouratidis, H., Fish, A.: Decision-making in security requirements engineering with constrained goal models. In: SECurity and Privacy Requirements Engineering, 2017 1st International Workshop on (SECPRE 2017). IEEE (2017)
6. Argyropoulos, N., Kalloniatis, C., Mouratidis, H., Fish, A.: Incorporating privacy patterns into semi-automatic business process derivation. In: Research Challenges in Information Science (RCIS), 2016 IEEE 10th Int. Conf. on, pp. 1–12. IEEE (2016)
7. Argyropoulos, N., Mouratidis, H., Fish, A.: Towards the derivation of secure business process designs. In: Int. Conf. on Conceptual Modeling, pp. 248–258. Springer (2015)
8. Argyropoulos, N., Mouratidis, H., Fish, A.: Attribute-based security verification of business process models. In: Business Informatics (CBI), 2017 IEEE 19th Conference on, vol. 1, pp. 43–52. IEEE (2017)
9. Argyropoulos, N., Mouratidis, H., Fish, A.: Supporting secure business process design via security process patterns. In: Enterprise, Business-Process and Information Systems Modeling - 18th International Conference, BPMDS 2017, 22nd International Conference, EMMSAD 2017, Held at CAiSE 2017, Essen, Germany, June 12-13, 2017, Proceedings, pp. 19–33 (2017)
10. Bottoni, P., Fish, A., Parisi-Presicce, F.: Spider graphs: a graph transformation system for spider diagrams. Software and Systems Modelling **14**(4), 1421–1453 (2015)
11. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems **8**(3), 203–236 (2004)
12. Cherdantseva, Y., Hilton, J.: A reference model of information assurance & security. In: The 8th International Conference on Availability, reliability and security (ARES), pp. 546–555. IEEE (2013)
13. Decreus, K., Poels, G.: A goal-oriented requirements engineering method for business processes. In: Forum at the Conf. on Advanced Information Systems Engineering (CAiSE), pp. 29–43. Springer (2010)

14. Decreus, K., Poels, G., Kharbili, M.E., Pulvermueller, E.: Policy-enabled goal-oriented requirements engineering for semantic business process management. Int. J. of Intelligent Systems **25**(8), 784–812 (2010)
15. Dubois, E., Mouratidis, H.: Guest editorial: security requirements engineering: past, present and future. Requirements engineering **15**(1), 1–5 (2010)
16. Fernandez, E.B., Pan, R.: A pattern language for security models. In: In Proc. of PLoP, vol. 1 (2001)
17. Greek-Parliament Act 3892: Electronic registration and fulfilment of medical prescriptions and clinical test referrals (2010). [In Greek]
18. Guerra, E., de Lara, J., Kolovos, D., Paige, R.: A visual specification language for model-to-model transformations. In: IEEE Symposium on Visual Languages and Human-Centric Computing (2010)
19. ISO: ISO/IEC 27000 Information technology Security techniques Information security management systems Overview and vocabulary. Tech. rep. (2014)
20. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Using privacy process patterns for incorporating privacy requirements into the system design process. In: 2nd Int. Conf. on Availability, Reliability and Security (ARES'07), pp. 1009–1017. IEEE (2007)
21. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the pris method. Requirements Engineering **13**(3), 241–255 (2008)
22. Kienzle, D.M., Elder, M.C.: Security patterns for web application development. University of Virginia technical report (2002)
23. Lavérdiere, M., Mourad, A., Hanna, A., Debbabi, M.: Security design patterns: Survey and evaluation. In: 2006 Canadian Conf. on Electrical and Computer Engineering, pp. 1605–1608. IEEE (2006)
24. Leitner, M., Miller, M., Rinderle-Ma, S.: An analysis and evaluation of security aspects in the business process model and notation. In: 8th Int. Conf. on Availability, Reliability and Security (ARES'13), pp. 262–267. IEEE (2013)
25. Li, T., Paja, E., Mylopoulos, J., Horkoff, J., Beckers, K.: Security attack analysis using attack patterns. In: Research Challenges in Information Science (RCIS), 2016 IEEE 10th Int. Conf. on, pp. 1–13. IEEE (2016)
26. Mouratidis, H., Argyropoulos, N., Shei, S.: Security requirements engineering for cloud computing: The Secure Tropos approach. In: Domain-Specific Conceptual Modeling, Concepts, Methods and Tools, pp. 357–380. Springer (2016)
27. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. Int. J. of Software Engineering and Knowledge Engineering **17**(2), 285–309 (2007)
28. Mouratidis, H., Weiss, M., Giorgini, P.: Modeling secure systems using an agent-oriented approach and security patterns. Int. J. of Software Engineering and Knowledge Engineering **16**(03), 471–498 (2006)
29. Neubauer, T., Klemen, M., Biffl, S.: Secure business process management: a roadmap. In: 1st Int. Conf. on Availability, Reliability and Security (ARES'06), pp. 457–464. IEEE (2006)
30. Nhlabatsi, A., Bandara, A., Hayashi, S., Haley, C., Jurjens, J., Kaiya, H., Kubo, A., Laney, R., Mouratidis, H., Nuseibeh, B., Tun, T., Washizaki, H., Yoshioka, N., Yu, Y.: Security patterns: Comparing modeling approaches. In: Software engineering for secure systems: Industrial and research perspectives, pp. 75–11 (2011)
31. Object Management Group: Business Process Model Notation (BPMN) Version 2.0. Tech. rep. (2011)
32. Rekik, M., Boukadi, K., Ben-Abdallah, H.: Bpmn meta-model extension with deployment and security information. In: 13th International Arab Conference on Information Technology ACIT (2012)
33. Rodriguez, A., Fernández-Medina, E., Piattini, M.: M-bpsec: a method for security requirement elicitation from a uml 2.0 business process specification. Advances in Conceptual Modeling–Foundations and Applications pp. 106–115 (2007)
34. Rosado, D.G., Gutiérrez, C., Fernández-Medina, E., Piattini, M.: Security patterns and requirements for internet-based applications. Internet research **16**(5), 519–536 (2006)
35. Salnitri, M., Dalpiaz, F., Giorgini, P.: Designing secure business processes with secbpmn. Software & Systems Modeling pp. 1–21 (2016)

36. Séguran, M., Hébert, C., Frankova, G.: Secure workflow development from early requirements analysis. In: on Web Services ECOWS'08, IEEE Sixth European Conference, pp. 125–134. IEEE (2008)
37. Sindre, G.: Mal-activity diagrams for capturing attacks on business processes. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 355–366. Springer (2007)
38. van Solingen (Revision), R., Basili (Original article 1994 ed.), V., Caldiera (Original article 1994 ed.), G., Rombach (Original article 1994 ed.), H.D.: Goal Question Metric (GQM) Approach. American Cancer Society (2002)
39. Souza, A.R., Silva, B.L., Lins, F.A., Damasceno, J.C., Rosa, N.S., Maciel, P.R., Medeiros, R.W., Stephenson, B., Motahari-Nezhad, H.R., Li, J., et al.: Incorporating security requirements into service composition: From modelling to execution. In: Service-Oriented Computing, pp. 373–388. Springer (2009)
40. Stonebumer, G., Goguen, A., Fringa, A.: Risk management guide for information technology systems. Recommendations of the National Institute of Standards and Technology (2002)
41. Toval, A., Nicols, J., Moros, B., Garcia, F.: Requirements reuse for improving information systems security: A practitioner's approach. Requirements Engineering **6**, 205–219 (2001)
42. Weske, M.: Business process management: concepts, languages, architectures. Springer Publishing Company, Incorporated (2010)
43. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process security requirement specification. Journal of Systems Architecture **55**(4), 211–223 (2009)
44. Yoshioka, N., Washizaki, H., Maruyama, K.: A survey on security patterns. Progress in informatics **5**(5), 35–47 (2008)
45. Zivkovic, S., Kühn, H., Karagiannis, D.: Facilitate modelling using method integration: An approach using mappings and integration rules. In: European Conference on Information Systems (ECIS) (2007)