A Secure Business Process Modelling For Better Alignment between Business and IT

Youseef Alotaibi
Department of Computer Sciences, Umm Al-Qura University, Makkah, Saudi Arabia, Email: yaotaibi@uqu.edu.sa

Abstract
The existing information system (IS) development methods do not meet the requirements to resolve security-related IS problems and they fail to provide for the successful integration of security and systems engineering during all development process stages. Security should be considered during all software development process and the requirements specifications should be identified. This paper aims to propose an integrated security and IS engineering approach in all software development process stages by using i* language. The proposed framework is divided into two separate parts: modelling information technology system and modelling IS security. The results indicate that considering security IS goals in the whole system development process can have a positive influence on system implementation, better meet business expectations and positively impacts on business process performance.

Key words: Business process; Business Process Modelling; Security; i* Model; UML

1. Introduction
Security plays a crucial role in business PM [20]. However, the literature shows that it is quite challenging to add security into BPs for several reasons. Firstly, the integration of security into a developed business PM is not very well understood. Secondly, security properties are complicated and error-prone when integrated manually [2]. Furthermore, a lack of experienced IS developers can result in security leaks. Therefore, IS developers need to have concrete guidelines and appropriate tools to develop secured business PM applications [3].

Security is considered a non-functional requirement (NFR) by many in the software engineering community. Non-functional requirements represent constraints, such as authorized and unauthorized accesses, where the systems are operating [12]. Therefore, security requirements must be defined after identifying the system. However, there are many challenges in implementing security into IS. Firstly, security requirements are often complicated to analyse and model. The requirement of separate functional and non-functional requirements is a major problem in analysing the NFRs whereas the NFRs could be related to one or set of the functional requirements at the same time. However, when the NFRs are stated separately from the functional requirements, the relationship between them cannot be seen easily. Secondly, IS developers may lack knowledge on developing and modelling a secured system [5][19].

Security should be considered throughout the entire business development process and requirements specifications should be identified. If security is only considered in certain stages of the development process, the security requirements will conflict with the system’s functional requirements [10]. Therefore, the issue of security must be taken into account with functional requirements during the system development stages in order to limit conflict. This can be done by defining them in the early stages of system development and making attempts to anticipate them. However, when security is only added in the late stages of system development, the chance of more conflicts occurring is increased, which may require a lot of money to address.

The literature shows that there are many commercial methods, such as ITBPM, OCTAVE, CRAMM, EBIOS, and MEHARI available to IT security officers to perform a risk analysis of the security problems and define the security solutions [18]. However, these existing methods do not meet the requirements to resolve security-related IS problems and they fail to facilitate the successful integration of security during all development process stages.

Thus, in order to create a secured business PM helping to improve the alignment between business and IT, this paper will propose an integrated security and IS engineering approach in all development process stages using the i* language. In our proposed secured business PM framework, there are five stages in software development to create a secured IS: (1) early requirements stage; (2) late requirements stage; (3) architectural design stage; (4) detailed design stage; and (5) implementation stages. The results show that considering security IS goals in the whole system development process can have a positive influence on
system implementation and better meet business expectations.

This paper is organized as follows. Section 2 outlines the related work. Section 3 presents our secured business PM framework. Finally, section 4 summarizes the paper.

2. Related Work

Security is an essential aspect of all information processing activities and all organizations have to develop actively mechanisms and tools to maintain and ensure the security and integrity of their information resources [23].

The literature shows that only a few approaches consider security requirements as a primary part of all software development processes. For example, applied the process-oriented approach to represent security requirements as harmonious goals and used them throughout the software system development. This proposed non-functional requirements (NFRs) framework uses security requirements and permits system developers to consider design decisions which are related to the non-functional requirements [9].

In [1], the authors demonstrated the application method for BP remodelling to achieve better realization and representation of the non-functional processes. This approach was adapted from the developed software engineering method to link the NFRs to the conceptual models. In a case study of BPs in healthcare, the authors used a cancer registration process in Jordan to show how the NRF graphing technique containing the goal operations and interaction analysis and goal evolution can be applied to create a NFR model for BPs by using RAD.

In [14], the authors proposed the requirements specification framework involving the separation of the delegation and trust relationships. This was undertaken to bridge the gap between the functional and trust requirements of the IT system and its trust management and privilege management infrastructure. This framework supported the automatic verification of IS requirements identified in the formal modelling language called the Delegation Logics and trust management [17]. The health care process was used as a case study to validate this framework.

In [18], the authors employed the requirements engineering approach to model and map IS security goals at an early stage of the software development process. In reference to alignment between business and IS, these approaches used five major steps: (1) identifying organizational environments; (2) derivation of security goals; (3) detecting security requirements from goals; (4) detecting constraint and security requirements; and (5) analysing risks at the architectural level.

In [16], the authors proposed an extension of UML, called UMLsec to provide security features in the UML model, such as access control and confidentiality. The study used four different UML diagrams: (1) class diagrams to guarantee that the exchange of data conforms to the security levels; (2) state chart diagrams to avoid the indirect flow of information from high to low values with the object; (3) interaction diagrams to guarantee the accuracy of important security interactions between the objects; and (4) deployment diagrams to guarantee that the physical layer can meet the security requirements in communication.

In [11], the authors proposed SecureUML, an expressive UML-based language, to construct their security design models. This method combines the design specifications for distributed systems with the security policy specifications. It is used to automate the model analysis semantically and meaningfully. This security policy contains the declarative aspects, such as static access control information and the programmatic aspects, which are based on the dynamic information known as the authorization satisfaction constraints. The SecureMOVA tool was used to implement this approach.

BPs are an essential source for software systems engineering [6][8] and much attention has been given to BPs in the area of IS security and software engineering. In [24] the authors proposed a formal meta-model integrating the modelling processes and the process-related Role Based Access Control (RBAC) models which contain the role, role hierarchies and the tasks’ duty constraints. This proposed approach was used to extend the UML2 activity models, which are known as Business Activities. The library and runtime engines were implemented to manage the Business Activities runtime models and enforce the software system’s policies and constraints. The gap between systems development and systems security may result in a lack of understanding of the security risks. In [13], the authors examined how to integrate security as the functional requirement in BPs analysis and modelling. They extended the semantic approach to the secure collaborative inter-organization e-Business processes proposed in [15]. They proposed the secure activity resource coordination (SARC) framework to create business PM characterized by the secure exchange of information across organizational boundaries. This framework has been evaluated against the Enriched Use Case and the UML activity diagram standard.

In [21], the authors proposed the Model Driven Development (MDD) approach for ISs development. This approach was particularly focused on the security
requirements where the security was modelled along with all other BP aspects. They identified a set of the rules and a checklist to automatically transfer and obtain the set of UML analysis classes and use cases from the secure business process (SBP) model. The proposed approach was validated through a case study of a real BP in the area of payment for electrical energy consumption.

In [19], the authors adapted use cases to propose an abuse case model that captured and analysed security requirements. This model identified the specifications of every interaction between the system and one or a set of actors, as such interactions can negatively affect the system. The misuse case concept describes functions which the system should not allow [22]. Furthermore, the miss-actor concept is defined as someone who accidentally or intentionally starts the misuse case. In this approach, security is considered by analysing a security-related misuse case.

However, these approaches have several drawbacks. They only provide guidance, for instance, as to how security can be handled during certain stages of the software development process. In other words, existing methods of IS development do not meet the requirements necessary to resolve security-related IS problems, and they fail to provide successful security integration during the development process stages. For example, the approach in [16] is only applicable throughout the design stage while the approach in [19] can only be used in early requirements analysis. Table 1 summarizes the existing literature on software development process stages.

A further limitation our review of the existing work highlights is that most of the previously mentioned approaches only deal with specific security requirements, goals and constraints. For example, UMLsec proposed in [16] focuses on access control security requirements and integrates them into the model-driven software development process. Table 2 summarizes the literature on existing security goals.

<table>
<thead>
<tr>
<th>References</th>
<th>Security Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integrity</td>
</tr>
<tr>
<td>[19]</td>
<td></td>
</tr>
<tr>
<td>[16]</td>
<td>√</td>
</tr>
<tr>
<td>[14]</td>
<td></td>
</tr>
<tr>
<td>[18]</td>
<td>√</td>
</tr>
<tr>
<td>[1]</td>
<td></td>
</tr>
<tr>
<td>[13]</td>
<td></td>
</tr>
<tr>
<td>[21]</td>
<td>√</td>
</tr>
<tr>
<td>[20]</td>
<td>√</td>
</tr>
</tbody>
</table>

Table 2: Related Work on Existing Security Goals

Therefore, we propose a security approach covering all stages of the software development process and considering all security requirements, such as access control and encryption, security goals, such as integrity and secrecy, and security constraints, such as authorized and unauthorized access in this paper.

3. Secured Business PM Framework

Many IS security problems can occur when an organization’s assets need to be protected from threats and attacks. However, it is a complex task to protect these assets since the business environment changes rapidly. Business organizations comprise complex business structures that are evaluated and updated within the customer structures and demands. These structures consist of processes, models, strategies and sets of activities worked together to achieve business goals. For better alignment between IS and business, IS security problems have to be addressed by managing security in the form of defining, analysing, modelling and mapping the IS attacks and identifying suitable security requirements in order to respond to these attacks in five different IS development stages: the early requirements stage, late requirements stage, architecture design stage, detailed design stage and implementation stage.

This paper aims to present a requirements engineering-based approach for business and IS analysts to better understand security problems, define
that can influence the
ure and loss of customer
on and entities can access the
IS assets constraints or
s, and to detect security
d on the already accepted
gement. IS
otect the business assets and
economic value to the
e IS assets need to be
protected to ensure business continuity, reduce
examples of IS assets. Th
ystem, Ethernet network, and system administrator are
on the IS as the IS assets support them. The operating
organizations, the business assets management depend
such as hardware, software, network and people. In
which are part of IS and their operating environment
into six major components:

- **Business Assets**

Business assets are anything which have an
economic value to the organization such as
formation, process, knowledge and people. They are
central to the realization of the organization’s business
goals and objectives. Technical plans and structural
calculation processes are examples of business assets.
It is an important to protect the business assets and
survival of the organization.

- **Information System Assets**

Information System assets are any components
which are part of IS and their operating environment
such as hardware, software, network and people. In
organizations, the business assets management depend
on the IS as the IS assets support them. The operating
system, Ethernet network, and system administrator are
examples of IS assets. The IS assets need to be
protected to ensure business continuity, reduce
business damage, and increase return on investment
and business opportunities.

- **Constraint**

Constraints are the restriction that can influence the
analysis and design the under development system. In
other words, they are the restrictions on the IS assets
such as fail, success, authorized, and unauthorized
access.

- **Security Goals**

Security goals are concerned with safeguarding
business and IS assets which protect the organizations
or individuals which depend on them. The security
goals are mapped into different IS assets constraints or
restrictions. They aim to ensure the integrity,
availability, and secrecy of the information and
processes in the organization, where integrity
safeguards the completeness and accuracy of the
information and processing methods and ensures that
the information remains unchanged from the source to
destination entity, availability ensures that the
authorized users have access to the IS when required,
and secrecy or confidentiality ensures that the
information is only accessed by authorized person,
entities, or processes.

- **Security Mechanisms and Techniques**

Security mechanisms and techniques include access
control, safeguard, encryption and decryption used to
reduce the risks that affect the business and IS assets
and solve the constraints or restrictions. Where the
access control is the selective restriction of access to
the information by authorized persons or entities only;
safeguard is the procedures or mechanisms that protect
against threats, limit the information security incident
impact, decrease the vulnerability and remove
incidents such as anti-virus software or backups;
encryption is the process of encoding the information
in such way that hackers cannot read it while only the
authorized person and entities can access the
information; and decryption is the process of decoding
the information.

- **Risks**

Risks are the combination of the probability of an
event and its consequence. Risks, such as the threat,
vulnerability, impact, and attack work against the
security goals; where the threat exploits the
vulnerability and causes unwanted incident or attack
such as information modification, system hacking,
thief, and deleted files. Vulnerability is a weakness of
the organization or system which can be exploited by
threat such as a lack of access control. Impact
describes the business assets level by negating security
criteria such as a loss of process integrity and IS assets
level, such as component failure and loss of customer
is the potential negative consequence of the risk which
can harm the business and IS assets when the threat is
accomplished. Attack is any attempt to expose, destroy, modify, steal, or gain unauthorized access to or make unauthorized use of the business and IS assets.

Modelling IS security level (Part 2) is divided into five different IS development stages: early requirements stage, late requirements stage, architecture design stage, detailed design stage and implementation stage, as shown in figure 2.

The early requirements stage focuses on understanding the problems by studying the setting of the existing organizations. In this stage, the business environment and assets are identified and the IS security goals and constraints are derived. Therefore, the organization model is the output of this stage. The late requirements stage focuses on modelling the “to-be” security model by adding and analysing security requirements and constraints. The architectural design stage focuses on defining the system’s global architecture, dividing the existing actors into sub-actors and linking them to the security goals as the second level in this stage [9].

Figure 1: Secured Business PM Framework Approach.
3.1.1 Early Requirements Stage

The early requirements stage focuses on understanding the problems by studying the setting of an existing organization. There are two main steps in this stage. In the first step, the business environment and assets are identified, while the IS security goals and constraints are derived in the second step. In other words, step 1 is where the BPs can be modelled by using the \(i^*\) modelling language and, thus, the security requirements can be linked within it whereas step 2 defines the IS security goals and how to link them within the BPs. Therefore, the organization model is the output of this stage.

- **Defining the Business Environment and Assets Step**

Business assets are anything which have an economic value to the organization such as information, process, knowledge, and people. They are the central in the realization of the organization’s business goals and objectives. Technical plans and structural calculation process are example of business assets. It is important to protect the business assets to ensure the survival of the organization. In this step, the business environment and assets are defined using \(i^*\) modelling. Firstly, business managers and system users have to explain their existing BPs, outline using descriptive words the problems, and the requirements and changes that have been found. Then, business analysts analyse the model and pass it to the business and system modellers who follow this procedure.

- **Deriving Information System Security Goals and Constraints Step**

After deciding which BPs need to be implemented and all BP assets are defined, the IS security goals need to be derived and defined to protect the proposed BP assets. Firstly, business analysts have to check the model BP and describe what BP activities have to be secured according to the business managers and system user’s requirements and explain any problems which were identified in their existing systems. For example, customer information must be secure to ensure customer privacy, and the relationship between the purchase department and manufacturer actors must be secure and access to the data warehouse must be secure. The next step is that the IS security goals and constraints are derived using the \(i^*\) model.

The literature shows that several methodologies can be used to protect the BP assets such as availability, secrecy and integrity. “Availability” indicates the usability and accessibility of the BP assets upon a request from the business authorities. “Secrecy”, which is also referred to as “confidentiality”, identifies that information will neither be disclosed nor available to unauthorized entities, authorities, processes or...
individuals. “Integrity” identifies the completeness and accuracy of the BP assets [17].

After defining the BP, the IS security goals have to be inserted into the BP. At this level, the IS security goals confirm the definition of the organization’s soft goals in the i* modelling language.

### 3.1.2 Late Requirements Stage

The functional, non-functional and security requirements of the system “to-be” are described in the late requirements stage. The “to-be” system introduces one or a set of actors that have a set of dependencies with other organizational actors identified in the early requirements stage. Thus, the late requirements stage focuses on modelling the “to-be” security model by adding and analysing the security mechanisms or techniques, such as access control, safeguard, encryption and decryption used to reduce the risks affect the business and IS assets and solve the constraints or restrictions; where the access control is the selective restriction of access to the information by authorized persons or entities only; safeguard is the procedures or mechanisms to protect against the threat, limit the information security incident impact, decrease the vulnerability and remove incidents, such as anti-virus software, backup or digital signatures (ensure the integrity and authentication of information); encryption is the process of encoding the information in such way that the hackers cannot read it while only authorized person and entities can access the information; and decryption is the process of decoding the information.

In this stage, the security techniques are used to confirm that all the business managers’ and system users’ requirements have been met. For example, the recovery or backup techniques could be used to ensure the data availability. Also, the digital signature or password pin could be used to ensure the authorization.

### 3.1.3 Architectural Design Stage

In the architectural design stage, the existing actors are divided into sub-actors and linked them to the security goals. For example, the “ensure data availability” and “ensure data privacy” goals are linked to the “availability manager” and “privacy manager” internal actors, respectively. Furthermore, the “ensure data integrity” goal is linked to “integrity verification manager” actor, which is linked to the “check data integrity” task, whereas the “check authorization” goal is linked to “access control manager” actor which is linked to the “access control” task.

### 3.1.4 Detailed Design Stage

The detailed design stage focuses on defining the architecture elements being defined in the previous stages in more detail in relation to inputs, outputs, controls and security. In other words, developers identify the actors’ interactions in detail throughout the detailed design stage, taking the security-related aspects derived from previous stages into account.

This stage contains 2 steps: (1) using the UML sequence graphical diagram for the actor interaction diagram; and (2) generating the final IS requirements and development using the UML state chart diagram.

- **Using UML Sequence Graphical Diagram for the Actor Interaction Diagram Step**

Sequence diagram shows the flow of events. It consists of objects represented in the usual way (as named rectangles with the name underlined), messages represented as solid-line arrows, and time represented as a vertical progression. The UML enables the user to expand the field of view and shows how an object interacts with other objects. In this expanded field of view, we have included an important dimension time. The key idea here is that the interactions among objects take place in a specified sequence, and the sequence takes time to go from beginning to the end. Extending downward from each object is a dashed line called the object’s lifeline. Along the lifeline is a narrow rectangle called activation. The activation represents an execution of an operation the object carries out. The length of the rectangle signifies the activation's duration where duration and time in general are represented in a rough and ordinal way. This means that each dash in a lifeline usually does not stand for a specific unit of time but is intended to give a general sense of duration.

**Guidelines to Translate i* Model Technique to UML Sequence Graphical Diagram**

In order to make the proposed secured business PM more understandable, we identify several guidelines that need to be followed to translate the i* model techniques used in the previous stages to UML sequence graphical diagram as follows:

Each actor in the i* model technique is mapped as the object lifeline on the UML sequence diagram. Each goal in the i* model technique is mapped as the message to associate between objects on the UML sequence diagram. Each security goal or task in the i* model technique is mapped as the message to associate objects on the UML sequence diagram.

- **Generating Final IS Requirements & Development Using UML State Chart Diagram Step**

State chart diagram captures the changes. It presents the states of an object that appear along with the transitions between the states and shows the starting point and endpoint of a sequence of state changes. The state chart diagrams show the flow of events and change of states from one state to another. It allows the system analysts and modellers to modify
any requirement package at any state chart stages. It has several fundamental UML symbols. For example, the icon for a state is a rounded rectangle, and the symbol for a transition is a solid line with an arrowhead. A solid circle stands for the starting point of a sequence of states, and a bull's-eye represents the endpoint.

In this step, the final IS requirements and development is generated using the UML state chart diagram. This generated UML state chart diagram gives a precise picture of the system requirements, and it passes into the implementation stage. At this stage, the secured BP is completed, analysed, modelled and developed, and is ready to develop the UML class diagram for implementation.

Guidelines to Translate i* model Technique to UML State Chart Diagram

In order to make the proposed secured business PM more understandable, we identify several guidelines that need to be followed to translate the i* model techniques used in the previous stages to UML state chart diagram as follows:

- Each actor in the i* model technique is mapped as the state on the UML state chart diagram. In other words, each object in the UML sequence diagram is mapped as the state on the UML state chart diagram.

- Each goal in the i* model technique is mapped as the association between states on the UML state chart diagram. In other words, each message or association in the UML sequence diagram is mapped as the association on the UML state chart diagram.

- Each security goal or task in the i* model technique is mapped as the message to associate states on the UML state chart diagram.

3.1.5 Implementation Stage

In the implementation stage, the UML class diagram is generated as the first step for the Java coding. Also, it helps on the analysis side too. It enable IS analysts and modellers to talk to clients in the clients' terminology and, thus, stimulate the clients to reveal important details about the problem to solve it.

At this stage, the secured BP is completed, analysed, modelled and developed, and is ready to be implemented using one of different implementation languages, such as Java, C++, and C. Before starting the implantation, the classes have to be checked. If there is any error or anything need to be changed, the model is sent back to the IS analysts and modellers. Otherwise, the model is implemented and it has to be tested, simulated and run in a similar environment before use in the real business environment.

Guidelines to Translate i* Model Technique to UML Class Diagram

In order to make the proposed secured business PM more understandable, we identify several guidelines that need to be followed to translate the i* model techniques used in the previous stages to UML class diagram as follows:

Each actor in the i* model technique is mapped as the class on the UML class diagram. In other words, each state in the UML state chart diagram is mapped as the class on the UML class diagram. Each dependency between two actors in the i* model technique is mapped as the association between these match two classes on the UML class diagram. Each goal in the i* model technique is mapped as the class operation on the UML class diagram. Each security goal or task in the i* model technique is mapped as the class operation on the UML class diagram.

3.1.6 Stages Summary

In summary, as guidelines for the users to follow, this secured business PM framework has several steps to be followed. Firstly, business managers and system users have to explain their existing BPs, outline the problems that have been found, as well as the requirements and changes using descriptive words.

Then, the users have to follow stage 1: early requirements stage. The business analysts analyse the BP and pass it to the business and system modellers. In the first stage, the users have to define the BP activities as it is in the business environment and try to reengineer the BP using i* model technique; where the person and departments, such as customer, head officer, account office and selling department are considered as an actors, while activities such as place order, make payment, and check payment are considered as goal. The second stage is that the users have to analyse the BP activities to consider and ensure those BP activities which could be at risk and must be secured. For example, the customer information must be secure to ensure customer privacy. Thus, the users should derive and map the system goals, such as integrity and availability into each activity considered in the previous analysis step using i* model technique where the security goal, such as integrity, availability and reliability are considered as the soft goal. Soft goal dependency is quite different to hard goal dependency. The soft goals refer to goals where there are no straightforward criteria to decide whether the condition has been met or not.

In the second stage: late requirement stage, the users should map the security mechanisms and techniques, such as encryption, decryption, and access control for each security goals defined and derived in stage 1 in order to ensure these security goals, where these security mechanisms and techniques are considered as tasks in the i* model. The task
dependency is used when any activities are performed by the organizational actors. For example, the backup procedure and recovery security mechanisms can be used to ensure the data availability. Furthermore, encrypt and decrypt data mechanisms can be used to ensure the data privacy.

In the third stage, the architectural design stage, as there are many security goals derived and defined in business PM. There are many security mechanisms and techniques that have been used in the system, each of these security goals and security mechanisms and techniques has to be related into one actor who will take the full responsibility to manage and control it. For example, in order to ensure the data integrity, the data availability and the data privacy, three new actors named as integrity manager, availability manager, and privacy manager are defined.

In the fourth stage, the detailed design stage, architecture elements that have been defined in the previous stages will be defined in more detail in relation to inputs, outputs, controls and security. In order to do so, there are two steps that have to be followed. Firstly, the UML sequence diagram is used to model the actor interaction between the system actors. There are some steps needed to be considered in order to generate the UML sequence diagram. For example, each actor in the i* model technique is mapped as the object life line on the UML sequence diagram. Furthermore, each goal, security goal or task in the i* model technique is mapped as the class operation on the UML class diagram. Each security goal or task in the i* model technique is mapped as the message to associate between objects on UML sequence diagram.

Secondly, the final IS requirements and development is generated using the UML state chart diagram. There are some steps that need to be considered in order to generate the UML state chart diagram. For example, each actor in the i* model technique is mapped as the state on the UML state chart diagram. In addition, each goal in the i* model technique is mapped as the association between states on the UML state chart diagram. Each security goal or task in the i* model technique is mapped as the association between states on the UML state chart diagram. This generated UML state chart diagram gives an accurate picture of the system requirements. It is sent to the IS developers, who will check this model, for implementation. If there is anything needed to be changed or there is an error, they send it back to the IS analysts and modellers to modify. Otherwise, model will be pass into the implementation stage. At this stage, the secured BP is completed, analysed, modelled and developed, and is ready to develop class diagram for implementation purpose.

In the last stage, the implementation stage, the UML class diagram is generated as the first step for the Java coding. There are some steps that need to be considered to generate the UML class diagram. For example, each actor in the i* model technique is mapped as the class on the UML class diagram. Moreover, each dependency between two actors in the i* model technique is mapped as the association between these match two classes on the UML class diagram. Each goal, security goal, or task in the i* model technique is mapped as the class operation on the UML class diagram. At this stage, the secured BP is completed, analysed, modalled, and developed, and is ready to be implemented using one of different implementation languages such as Java, C++, and C. Before starting the implantation, the classes have to be checked. If there is any error or anything that needs to be changed, the model is sent back to the IS analysts and modellers. Otherwise, the model is implemented and it has to be tested, simulated, and run in similar environment before used in real business environment.

4. Conclusion & Implications

Security must be considered throughout the entire business development process and requirements specifications should be identified. In this paper, we presented an integrated security and IS engineering approach throughout all the software development process stages by using the i* language in order to solve the second challenge: creating a secured business PM. We have divided our proposed framework into two separate parts: modelling the information technology system, and modelling the IS security.

Modelling IS security consists of five major stages: (1) early requirements stage; (2) late requirements stage; (3) architectural design stage; (4) detailed design stage; and (5) implementation stage. In the early requirements stage, the business environment and assets are identified and the IS security goals and constraints are derived, whereas in the late requirements stage, the “to-be” security model is modelled by adding and analysing the security requirements and constraints. Furthermore, in the architectural design stage, the existing actors are divided into sub-actors and the security goals are linked. The detailed design stage where the architecture elements are defined in more detail contains two steps: (1) using the UML sequence graphical diagram for actor interaction diagrams; and (2) generating the final IS requirements and development using UML state chart diagram. In the implementation stage, the UML class diagram is generated as the first step for the Java coding.

The results show that considering security IS goals in all system development process can have a positive influence on system implementation and better meet business expectations.

Two major implications can be derived from this paper for IS developers and business organizations.
First, for IS developers, the paper shows how system security goals can be derived from the business environment and defined during the whole system development process which leads an improved system. Second, for the business organization, it can increase customer confidence and trust which can lead to an increase in the company’s profit.

However, because of the paper page limitation, the paper has one limitation; we have not validated our proposed framework within any existing business process as a case study.

References


