How to Model a Secure Information System (IS): A Case Study

Youseef Alotaibi and Fei Liu

***Abstract*—The existing information system (IS) developments methods are not met the requirements to resolve the security related (IS) problems and they fail to provide a successful integration of security and systems engineering during all development process stages. Hence, the security should be considered during the whole software development process and identified with the requirements specification. This paper aims to propose an integrated security and IS engineering approach in all software development process stages by using *i** language. This proposed framework categorizes into three separate parts: modelling business environment part, modelling information technology system part and modelling IS security part. Booking hotel room management process is used as a case study to validate the proposed framework. 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.**

***Index Terms*—Case study, information system, requirement engineering, software development process, security goals.**

I. Introduction

Information system (IS) has been used almost in all aspect of human society, such as military, health science, telecommunication companies, e-commerce etc. Since using the IS has been arise, the concept of security to secure these systems requires to be arise. Because many systems may contain a private data to be available only to authorized people, the security concepts have to be added into the IS. For example, booking hotel rooms order management process contains the customers' personal information and their credit card information and thus this system must to be secured to save the customers' privacy.

The security is considered as a non-functional requirement by the software engineering [1]. Although, the non-functional requirement introduces the quality features, it represents the constraints, such as authorized and unauthorized accesses where the systems must be operated [2], [3]. Therefore, the security requirements must be defined after identifying the system. However, there are several challenges to have a better support for the security engineering. Firstly, the security requirements are commonly complicated to be analyzed and modelled. There is one main problem in analyzing the non-functional requirements which is the requirement of separate the functional and non-functional requirements while the non-functional requirements could be related to one or set of the functional requirements at the same time. However, when the non-functional requirements are stated separately from the functional requirements, the correspondence between them cannot be seen easily. Secondly, the IS developers may have lack knowledge to develop and model a secure system [4], [5].

The security has to be considered through all business development process and identified with the requirements specification. If the security only considers in the certain stages of the development process, the security requirements will conflict within the system functional requirements. Therefore, the security requires to be taken into account within the functional requirement during the system development stages in order to limit the conflict cases and that can be done by defining them in the early stages of the system development and trying to overcome them. However, when the security only adds in the late stages of the system developments, the chance of having more conflicts could be increased and it may require a lot of money to overcome them.

Literature shows that there are many commercial methods, such as ITBPM, OCTAVE, CRAMM, EBIOS, MEHARI, etc available to IT security officers in the organizations to be used to perform the risk analysis of the security problems and define the security solutions [6]–[8]. However, these existing methods of the IS developments are not met the requirements to resolve the security related IS problems and they fail to provide a successful integration security during all development process stages.

Thus, we propose an integrated security and IS engineering approach in the all development process stages by using *i** language. There are four stages of the software developments to have a secure IS in our proposed framework approach: (1) early requirements stage, (2) late requirements stage, (3) architectural design stage and (4) details design stage. Booking hotel room management process has been used as a case study in order to validate our proposed framework. 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 the business expectations.

The remainder of this paper is organized as follows: section II describes the related work of modeling secure IS; section III presents our proposed framework approach; section IV describes the proposed framework validation with the help of a case study; and the conclusion and future research directions are presented in section V.

Manuscript received March 24, 2012; revised April 27, 2012.

A. Y. Author is with La Trobe University, Bundoora, VIC, 3086, Australia (phone: +61 405099952; e-mail: yaalotaibi@students.latrobe.edu.au).

F. L. Author is with La Trobe University, Bundoora, VIC, 3086, Australia (phone: +61 3 9479 1949; e-mail: f.liu@latrobe.edu.au).

II. Related Work

Literature shows that there are only a few approaches considered the security requirements as a primary part of all software development processes. For example, in [1], the authors applied the process oriented approach to represent the security requirement as harmonious goals and used them throughout the software system development. This non-functional requirements proposed framework is represented and used the security requirements as the classes of the non-functional requirements and it permits the system developers to consider the design decisions which are related into the represented non-functional requirements.

In [9], the authors proposed an approach to reuse the existing descriptions of the business process to analyze the security requirements and derive the essential security measures. This proposed approach contained four major steps: (1) identifying the general security objectives of the business process, (2) examining the constructs security objectives, such as actors, (3) examining whether these specifications are consistent or not, and (4) creating the list of the essential security measures for every business process component.

In [7] and [10], the authors proposed the requirements engineering approach to model and map the IS security goals at the early stage of the software development process in the context of the alignment between the business and IS. These approaches contain five major steps: (1) identifying organization environments, (2) derivation of information security goals, (3) detecting security requirements from goals, (4) detecting constraint and security requirements, and (5) analyzing risks at the architectural level.

In [11], the extension of the Unified Modelling Language (UML) which calls UMLsec was proposed to contain the model of the security features, such as access control and confidentiality. There are four different UML diagrams used in [11]: (1) class diagrams to guarantee that exchanging of data obeys the security levels, (2) state chart diagrams to avoid the indirect information flow from the high to low values with the object, (3) interaction diagrams to guarantee the accuracy of the important security interactions between the objects and (4) deployment diagrams to guarantee that the physical layer can meet the security requirement on communication. Moreover, in [12] the UML was extended to model security and the authors presented the security modelling language called the SecureUML. The authors described how the UML could be used to identify the access control related information in the whole application design and used this information to create a complete access control infrastructures automatically.

In [5], the authors adapted the use cases to propose an abuse case model which used to capture and analyze the security requirements. This model identified as the specification of complete interaction type between the system and one or set of actors and this interaction can negatively affect the system. The misuse case concept which describes the non allowed function by the system defined in [13]. Furthermore, the mis-actor concept defined as someone who accidentally or intentionally starts the misuse case. In this approach, the security is considered by analyzing the security related misuse case.

In [14], the obstacle concept was used in the KAOS framework to capture undesired system properties, identify and relate the security requirements into other system requirements. There are two set of techniques which bases on the temporal logic formalization utilized because of obstacle goals satisfaction and requirements.

All of pervious mentioned approaches above provide the first step to integrate the security concept within the software engineering and they are useful in modelling security requirements. However, these approaches has some drawbacks since they only have a guide about how can the security handled during the certain stage of software development process. For example, the approach in [11] is applicable throughout the design stage while the approach in [5] is used throughout the early requirements analysis. Hence, we will propose a security approach covering all software development process which can help limiting the conflict cases by defining them at the early stage in the system development and trying to overcome them. Table 1 summarizes the literature of existing software development process stages.

TABLE I: Related Work of Existing Software Development Process Stages [15].

Reference	Year	Software Development Process Stages			
		Early Requirement	Late Requirement	Architecture Design	Detail Design
[5]	1999	√			
[11]	2001			√	
[16]	2002	√		√	
[9]	2002	√			
[12]	2002			√	
[7]	2007	√			
[17]	2008			√	
[18]	2009	√		√	
[10]	2011	√			
[19]	2011			√	√

III. Proposed Framework

There are many IS security problems happened when the origination assets require to be protected from the threats and attacks. However, it is a complex task to protect organization assets since the business environment has been changed rapidly. The business organizations contain complex business structures that are evaluated and updated within the customer structures and demands which consist of processes, models, strategies and set of activities which work together to achieve the business goals. To better alignment between IS and business, the IS security problems have to be addressed by managing the security in the form of defining, analyzing, modelling and mapping the IS attacks and identifying the suitable security requirements in order to respond to these attacks in four different IS development stages: early requirements stage, late requirements stage, architecture design stage and detail design stage.

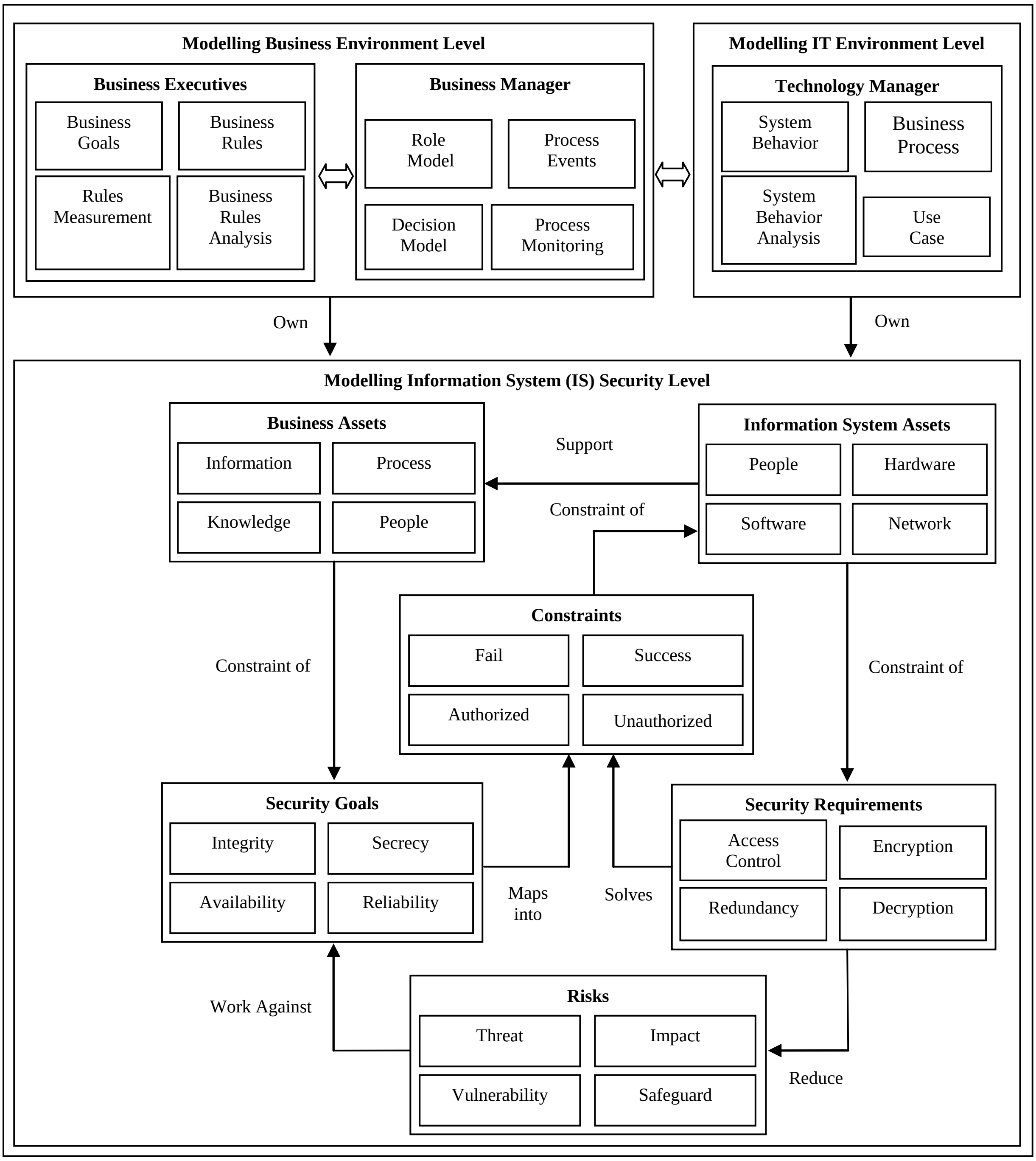

Fig. 1. Proposed framework approach

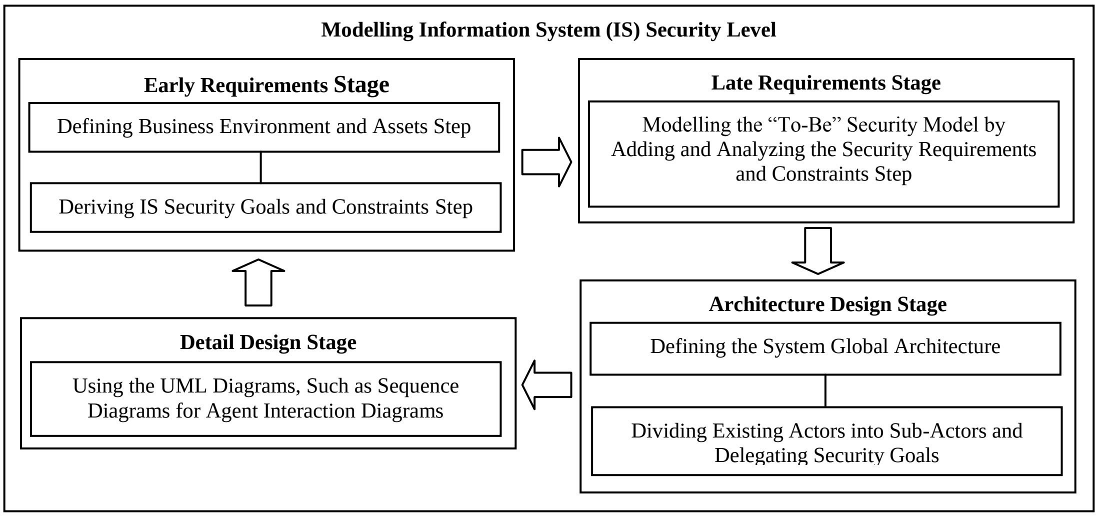
Modelling Information System (IS) Security Level
Early Requirements Stage
Defining Business Environment and Assets Step
Deriving IS Security Goals and Constraints Step
Late Requirements Stage
Modelling the "To-Be" Security Model by Adding and Analyzing the Security Requirements and Constraints Step
Architecture Design Stage
Defining the System Global Architecture
Dividing Existing Actors into Sub-Actors and Delegating Security Goals
Detail Design Stage
Using the UML Diagrams, Such as Sequence Diagrams for Agent Interaction Diagrams

Fig. 2. Modelling information system (IS) security level

This paper aims to present a requirement engineering-based approach for the business and IS analysts to better understand the security problems and define their associated security goals and detecting the security requirements and constraints from the goals. We have categorized our proposed framework into three separate parts: modelling business environment part, modelling information technology system part and modelling information system security part as shown in figure 1. Part 1 divides into two levels: the business decision level and the business process modelling level where each level is made up of four business components. The business decision level consists of the business goals, the business rules, the rules measurement and the business rules analysis. The business process modelling level consists of the role model, the process events, the decision model and process monitoring. Part 2 consists of the system behaviour, the business process, the system behaviour analysis, and the use case. Part 1 and 2 are not in the scope of this paper and for more details please refer to [20].

Part 3 describes how to define, model and analyze the attacks on the IS and business organization as the security is the major element in IS for this proposed approach scope. It identifies the qualities expected from IS, such as reliability, safety, usability and etc. Part 3 is divided into four different IS development stages: early requirements, late requirements, architecture design and detail design stages as shown in figure 2.

The early requirements stage focuses on understanding the problems by studying the setting of existing organizations. In this stage, the business environments and assets are identified and the IS security goals and constraints are derived. Therefore, the organization model is the output of this stage. However, the late requirements stage focuses on modelling the "to-be" security model by adding and analyzing the security requirement and constraints. Furthermore, the architectural design stage focuses on defining the system global architecture, such as mobile agent and client and server in subsystems that interconnect to each other throughout the data and control flows. Next, the existing actors are divided into sub-actors and the security goals are delegated as the second level in this stage. The detail design stage focuses on defining the architecture elements that has been defined in the previous stages in more details in inputs, outputs, controls and security aspects by using the UML sequence diagram for the agent interaction diagram [21].

IV. Case Study

To validate this proposed framework approach, booking hotel room management process is used as the case study where the systems' goal is to implement the process of booking the hotel room automatically in order to save the customer's time and reduce the number of staff which will, in turn, have a positive effect on company revenue. The *i** modelling language has been used to model the proposed business model [13].

The *i** modelling framework is an agent-oriented requirements modelling language appropriate for the early phase of system modelling to understand the system's problems. It is used for strategic actor relationships and the intentional model. This framework contains two important components: the Strategic Dependency Model (SDM) and the Strategic Rationale Model (SRM). The SDM is used to describe the network of relationships between actors. The SDM is a component where every node represents an actor and every link between two nodes shows that one actor is dependent on the other actor. It also provides a description of the external relationships between the actors. The aim of the SDM is to provide indications about why the business process is organized in a certain way. However, it cannot adequately support the exploration, suggestion and evaluation of other solutions for the process, which the SRM can do [10].

The SRM is used to support and describe why actors can have different ways to organize their work, such as a different configuration for Strategic Dependency networks. SRM has four main nodes: goal, soft goal, resource and task, and two main links which are mean-ends links and task decomposition links. These are used to model the internal relationships between actors. This model can systematically explore possible new business process designs [18], [22].

A. Early Requirements Stage

The early requirements stage focuses on understanding the problems by studying the setting of existing organizations. There are two main levels in this stage. In the first level, the business environments and assets are identified while the IS security goals and constraints are derived in the second level. In other words, level 1 is where the business processes can be modelled by using the *i** language and thus the security requirements can be linked within it whereas level 2 defines the information system security goals and how to link them within the business processes. Therefore, the organization model is the output of this stage.

A.1. Defining the Business Environment and Assets Step

Figure 3, using the *i** diagram, shows that the booking hotel room management process contains several activities. It consists of six different organizational actors: "customers", "head office", "reservation department", "administration office", "database" and "account office". There are four different kinds of dependencies between the actors: the business goal dependency, the soft goal dependency, the task dependency and the resources dependency. There are seven different business goal dependency categories in our case study: (1) "Place Booking" supports the customers to place their booking order with the head office actor, (2) "Manage Customer Claim" defines how the customers can lodge a claim with the administration office, (3) "Make Payment" supports the customer in making the payment to the account office, (4) "Check Availability" checks that there is an available room in the hotel database, (5) "Update" enables the company's database to be updated after booking any rooms types, (6) "Manage Finance" helps the administration office to manage the finance on the account office department, (7) "Confirm Payment" confirms that the account office has received the customers payment after the booking service is processed.

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. The task dependency is used when any activities are performed by the organizational actors. For example, there is one task dependencies in our case study: “Structure Calculations” which is done by the head office actor for the reservation department actor. The resource dependency is used to describe the dependencies between different organizational actors. For example, there are three resource dependencies in our case study. The head office actor is dependent on certain resources, such as it has to provide the “technical plans” and “models” to the reservation department actor and the reservation department actor has to provide an “estimate” to its customers.

A.2. Deriving Information System Security Goals and Constraints Step

After deciding which business processes need to be implemented and all business process assets are defined, the IS security goals need to be derived and defined in order to protect the proposed business process assets. The literature shows that several methodologies can be used to protect the business process assets, such as availability, secrecy and integrity [23]. “Availability” indicates the usability and accessibility of the business process assets upon a request from the business authorities. “Secrecy”, which is also referred to as “confidentiality”, identifies the neither information which will not be disclosed nor will it be available to unauthorized entities, authorities, processes or individuals. “Integrity” identifies the completeness and accuracy of the business process assets.

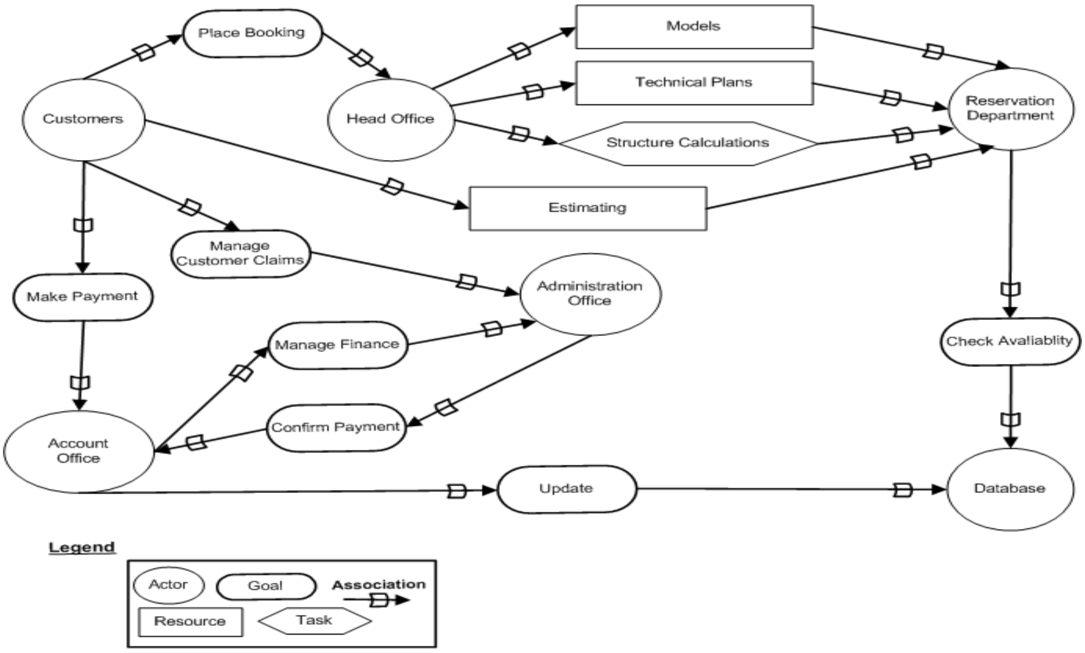

Fig. 3. Definition of the business environment and assets step

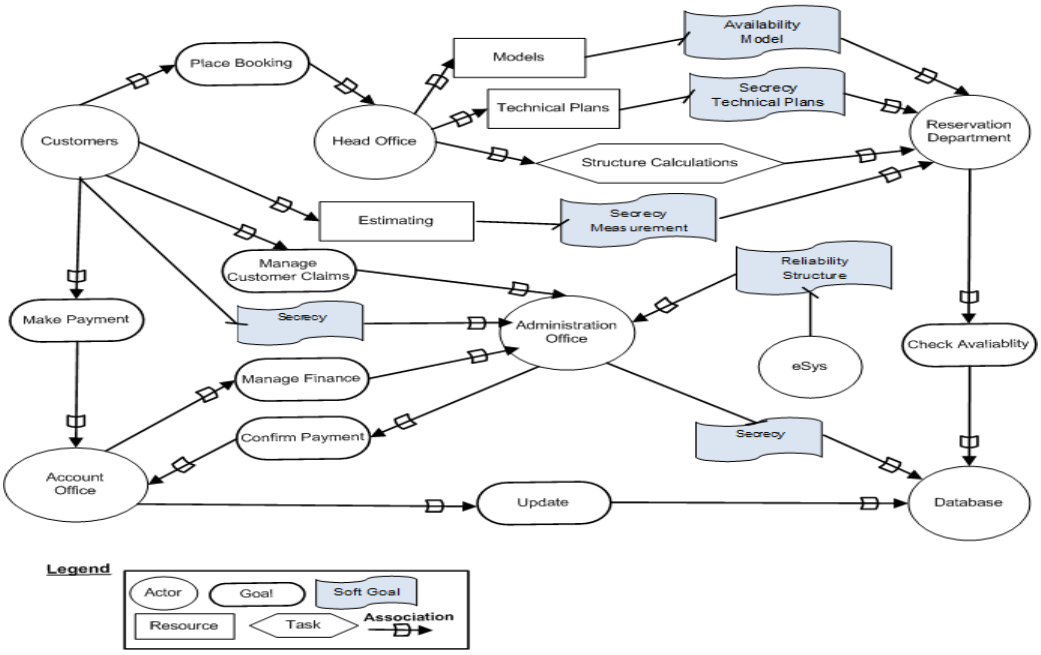

Fig. 4. Derivation of IS security goals and constraints step

After defining the business process, the IS security goals have to be inserted into the business process. At this level, the IS security goals confirm the definition of the organization’s soft goals in the *i** language. The literature on requirements engineering shows that it is easy to map IS security goals into business requirements [3], [12]. Figure 4 shows our case study’s soft goals, such as secrecy measurement which is the soft goals for estimating. These security goals show how the business process and sensitive information about customers is secured. For instance, the secrecy measurement security goal contributes to the customers’ trust and confidence, and the availability model security goal contributes to the company’s confidence. In addition, the security goals may be represented as security dependencies in some cases when the actors indicate security issues rather than the companies’ soft goals. A new actor, called electronic System “eSys”, is introduced in order to satisfy the reliability of the hotel’s structure and define the stakeholders who have security concerns in our case study.

B. Late Requirements Stage

The functional, non-functional and security requirements of the system “to-be” are described at 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 analyzing the security requirements and constraints.

In our case study, the main aim of the hotel is to improve the customers’ trust and confidence and assist the reservation department to provide good service and the account office to confirm the customers’ payments. Therefore, the hotel system depends on the eSys to have an automatic service and thus the eSys introduces as a new actor in our case study and analyses by using the same concepts used to analyze other actors in our case study. Any goals which cannot be met by the system’s actors or can be met in a better way by the eSys are assigned to the eSys actor.

The main goal of the eSys is to automate services in order to satisfy these dependencies between actors. Several sub-goals must be met, as shown in figure 5, to fulfill the automatic service goal in the eSys as follows: “structure calculation”, “provide customer information” and “estimate tools use”. Every sub-goal can be further analyzed by employing mean end analysis. For instance, the “estimate tools use” goal is met in the fulfillment of the “record technical plan”, “estimate technical plan”, “update technical plan” and “validate technical plan” sub-goals.

From a security point of view, there are three major security goals which need to be considered by the eSys: integrity, privacy and availability. These security goals are shown in figure 5 as “keep data integrity”, “keep data privacy” and “keep data available”. Furthermore, the eSys has to satisfy the “share information only if customer accepts” security goal. These security goals can be satisfied by three major goals: “ensure data integrity”, “ensure data privacy” and “ensure data availability” respectively while the “keep data privacy” security goal is also fulfilled by the “block system access” goal.

These three major goals are divided into different tasks and sub-goals. For example, the “ensure data integrity” goal is divided into “check data integrity” task which can be achieved by considering the “use authorization code message” and the “use digital signature” tasks. Moreover, the “ensure data privacy” goal is divided into three different tasks: “encrypt data”, “decrypt data”, and “access control” which can be achieved by performing the “check password” task, and “ensure customer accept” sub-goal. In addition, the “ensure data availability” goal is achieved by considering two different tasks: “recovery” and “backup procedure”.

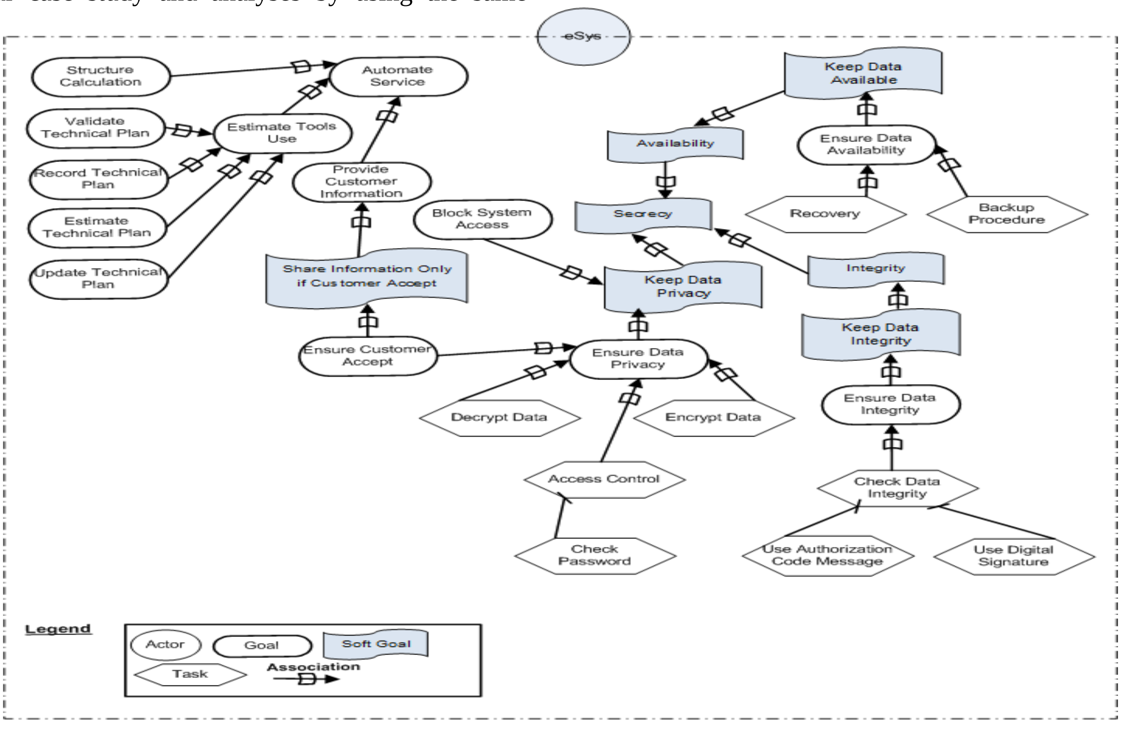

Fig. 5. Late requirements stage

C. Architectural Design Stage

There are two main steps in the architectural design stage. At the first level, the system's global architecture, such as the mobile agent and the client/server, is defined in subsystems which interconnect to each other throughout the data and control flows. Thus, the Architectural Style Selection Diagram (ASSD) proposed in [24] is used to model these architecture styles and system security requirements and goals. For further information on how to select the best architecture style, refer to [24], as this is not within the scope of this research. At the second level, the existing actors are divided into sub-actors and the security goals are delegated.

C.2. Dividing Existing Actors into Sub-Actors and Delegating Security Goal Level

After evaluating different architecture styles and selecting one of them, the existing actors are divided into sub-actors and the security goals are delegated. The eSys actor is decomposed to the internal actors and the responsibility for fulfilling the eSys goals is delegated to these internal actors, as shown in figure 6. For example, the "ensure data availability" and "ensure data privacy" goals are delegated to the "availability manager" and "privacy manager" internal actors, respectively. Furthermore, the "ensure data integrity" goal is delegated to "integrity manager" and "integrity verification manager" is delegated to the "check data integrity" task, whereas the "access control" goal is delegated to "access control manager".

The "block system access" goal is delegated to the "system access manager" actor while the "acceptance manager" and "customer broker" actors are introduced into the eSys actor in order to satisfy the security goals of obtaining customer information together in the "share information only if the customer accepts". In addition, the "structure calculation" goal is delegated to the "structure calculation manager".

Finally, "validate technical plan", "record technical plan", "estimate technical plan" and "update technical plan" goals are delegated to the "technical plan validate manager", the "technical plan record manager", the "technical plan estimate manager" and the "technical plan update manager" actors, respectively.

D. Detail Design Stage

The detail design stage focuses on defining the architecture elements that have been defined in the previous stages in more detail in relation to inputs, outputs, controls and security. In other words, the system developers identify the actors' interactions in detail throughout the detail design stage, taking the security-related aspects derived from previous stages into account. In this stage, the UML sequence diagram is used to model the agent interaction between the system actors, as shown in figure 7 [25], [26]. This diagram illustrates the interactions with arrow lines between the customer, head office, administration office, eSys System, privacy manager, access control manager, database and reservation department actors which are graphically shown by the rectangles at the top of diagram.

The customers place their booking with head office and then the payment is checked. If the payment is accepted, the booking is accepted. Otherwise, the booking order is cancelled. The security rules which are similar to the business rules as defined by the UML are introduced. These security rules are placed on notes and attached to the related actor interactions. Next, the administration office sends the eSys access request to the eSys system and then the incoming request is decrypted with the aid of the privacy manager. At the next step, after providing the authorization information, this authorization information is checked to ensure that it is valid, and authorization clearance is provided. Otherwise, authorization clearance is rejected.

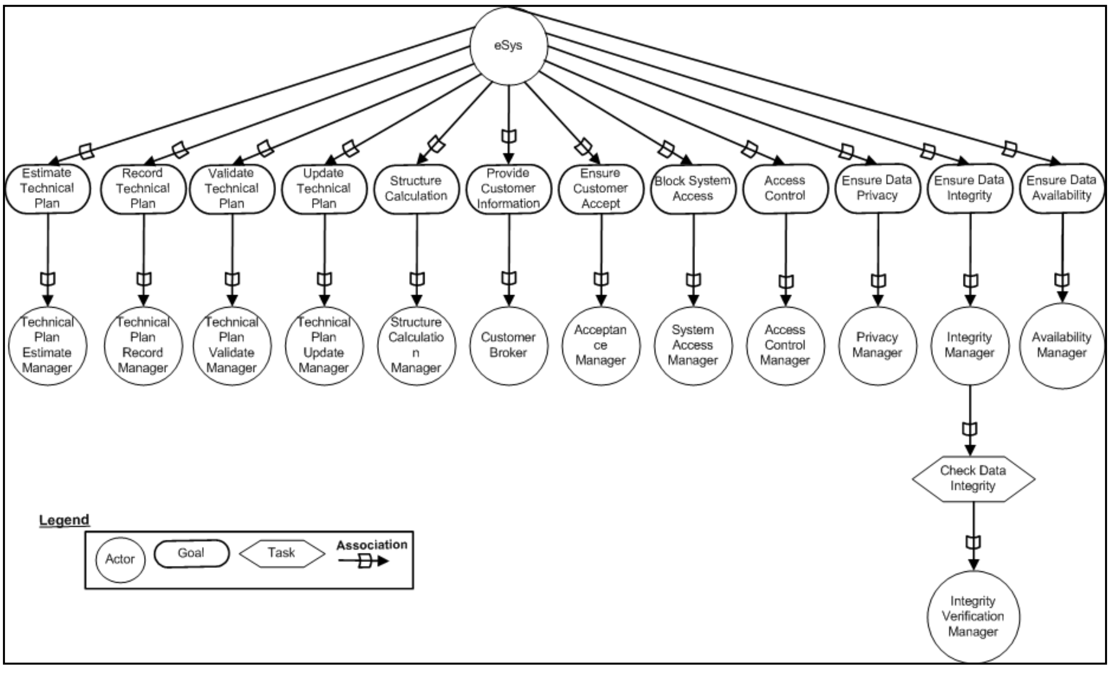

Fig. 6. Dividing existing actors into sub-actors and delegating security goal level

The eSys system sends the eSys access reply to the administration office. The eSys access request is accepted if the authorization clearance is provided. Otherwise, the eSys access request is rejected. After accepting the customer's payment, the order is progressed by checking the rooms' availability in the database. If there is any room available, the booking is managed and the booking confirmation is sent to the customer.

V. Conclusion and Implications

Security can play a crucial role in business processes and e-commerce. However, the literature shows that it is quite challenging to add security into business processes for several reasons. Firstly, the integration of security into a developed business process is not very well understood. Secondly, security properties are complicated and error-prone when integrated by hand. Furthermore, the lack of experience of IS developers can lead to security leaks. Therefore, IS developers need to have concrete guidelines and appropriate tools to develop secure applications.

Security must be considered throughout the entire business development process and requirements specifications should be identified. In this paper, we present an integrated security and IS engineering approach throughout all the software development process stages by using the *i** language. We have divided our proposed framework into three separate parts: modelling the business environment, modelling the information technology system and modelling the information system security.

Modelling IS security consists of four major stages: (1) early requirements stage; (2) late requirements stage; (3) architectural design stage; and (4) details design stage. At the early requirements stage, the business environment and assets are identified and the IS security goals and constraints are derived, whereas at the late requirements stage, the "to-be" security model is modelled by adding and analyzing the security requirements and constraints. Furthermore, at the architectural design stage, the existing actors are divided into sub-actors and the security goals are delegated while at the detail design stage, the architecture elements are defined in more detail by using the UML sequence diagram for the agent interaction diagram.

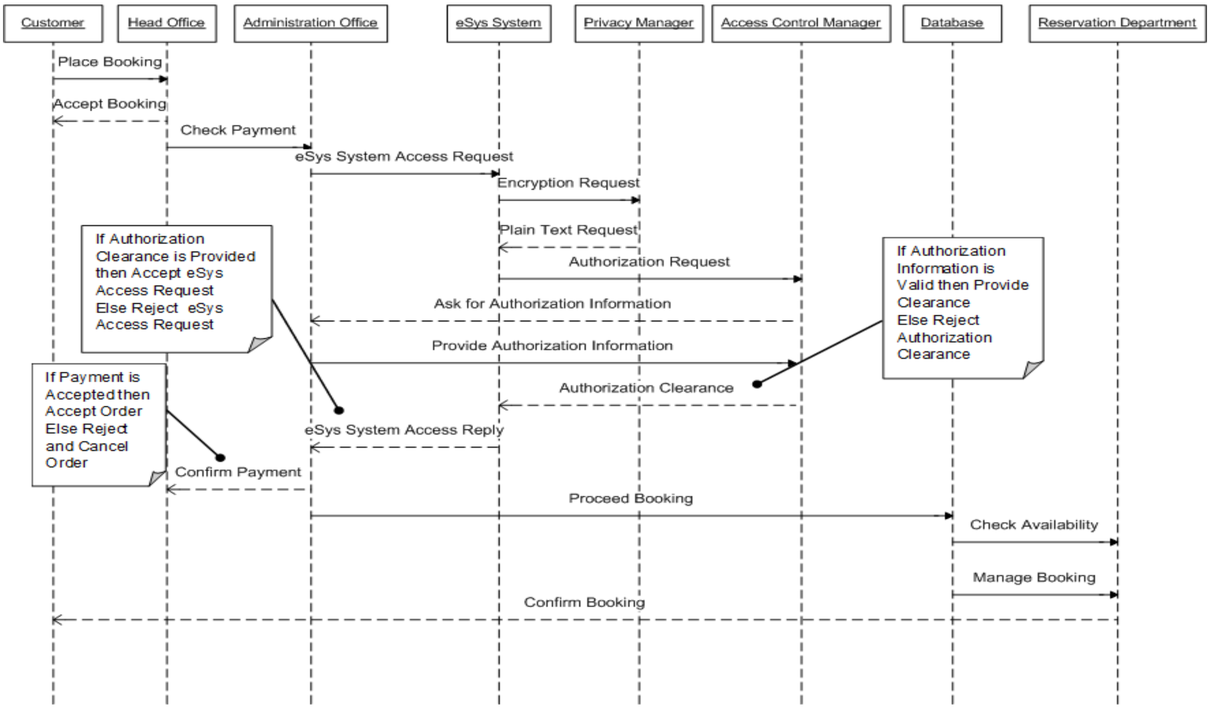
Customer
Head Office
Administration Office
eSys System
Privacy Manager
Access Control Manager
Database
Reservation Department
Place Booking
Accept Booking
Check Payment
eSys System Access Request
Encryption Request
Plain Text Request
Authorization Request
If Authorization Clearance is Provided then Accept eSys Access Request Else Reject eSys Access Request
If Authorization Information is Valid then Provide Clearance Else Reject Authorization Clearance
Ask for Authorization Information
Provide Authorization Information
Authorization Clearance
If Payment is Accepted then Accept Order Else Reject and Cancel Order
eSys System Access Reply
Confirm Payment
Proceed Booking
Check Availability
Manage Booking
Confirm Booking

Fig. 7. Sequence diagram for agent interaction diagram

Booking hotel room management process was used as a case study in order to validate our proposed framework. 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.

Two major implications can be derived from the study for IS developers and business organizations. First, for developers, the study 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 the hotels' profit. However, the paper has one limitation; we only tested our proposed framework on one business process. Thus, in the future, it could be possible to test our framework with more than one business process in different business sectors.

References

[1] L. Chung and B. A. Nixon, "Dealing with non-functional requirements: three experimental studies of a process-oriented approach," in *Proceedings of the 17th international conference on Software engineering*. ACM: Seattle, Washington, United States. 1995, pp. 25-37.

[2] C. B. Haley, J. D. Moffett, R. Laney, and B. Nuseibeh, "A framework for security requirements engineering," in *Proceedings of the 2006 international workshop on Software engineering for secure systems*. ACM: Shanghai, China. 2006, pp. 35-42.

[3] E. Yu and L. Cysneiros. "Designing for privacy and other competing requirements." in *2nd Symposium on Requirements Engineering for Information Security (SREIS' 02)*. Raleigh, North Carolina. 2002.

[4] M. Backes, B. Pfitzmann, and M. Waidner, "Security in business process engineering." *Business Process Management, Springer Berlin / Heidelberg*, 2003: pp. 1019-1019.

[5] J. McDermott and C. Fox. “Using abuse case models for security requirements analysis.” In *Proceedings 15th Annual Computer Security Applications Conference*, (ACSAC '99). 1999.

[6] R. J. Anderson, Security Engineering: “A guide to building dependable distributed systems.” 2008.

[7] N. Mayer, E. Dubois, and A. Rifaut, “Requirements Engineering for Improving Business/IT Alignment in *Security Risk Management Methods Enterprise Interoperability II*” Springer London. 2007, pp. 15-26.

[8] H. Mouratidis and J. Jurjens, “From goal-driven security requirements engineering to secure design.” *International Journal of Intelligent Systems*, vol. 25, no. 8, pp. 813-840, 2010.

[9] S. Rohrig and S. S. Ag, “Using process models to analyze health care security requirements,” in *International Conference Advances in Infrastructure for e-Business, e-Education*, e-Science, and e-Medicine on the Internet. Italy. 2002.

[10] A. Ullah and R. Lai, “Managing Security Requirements: Towards Better Alignment between Information Systems and Business,” in *15th Pacific Asia Conference on Information System (15th PACIS)*, Queensland University of Technology (QUT) in Brisbane, Australia. 2011.

[11] J. Jürjens, “Towards Development of Secure Systems Using UMLsec Fundamental Approaches to Software Engineering,” *Springer Berlin / Heidelberg*. 2001, pp. 187-200.

[12] T. Lodderstedt, D. Basin, and J. Doser, “SecureUML: A UML-Based Modeling Language for Model-Driven Security,” in *the Proceedings of the 5th International Conference on the Unified Modeling Language*, Springer Berlin / Heidelberg. 2002, pp. 426-441.

[13] G. Sindre and A. L. Opdahl, “Eliciting security requirements with misuse cases.” *Requirements Engineering*, 2005, vol. 10, no. 1, pp. 34-44.

[14] A. Dardenne, S. Fickas, and A. V. Lamsweerde, “Goal-directed concept acquisition in requirements elicitation,” in *Proceedings of the 6th international workshop on Software specification and design*. IEEE Computer Society Press: Como, Italy. 1991, pp. 14-21.

[15] Y. Alotaibi and F. Liu, “A Novel Framework to Model a Secure Information System (IS),” *2012 International Conference on Information and Computer Applications (ICICA 2012)*. Hong Kong. 2012.

[16] L. Liu, E. Yu, and J. Mylopoulos. “Security and privacy requirements analysis within a social setting.” in *Proceedings on 11th IEEE International Requirements Engineering Conference*, 2003.

[17] G. Goluch, A. Ekelhart, S. Fenz, S. Jakoubi, S. Tjoa, and T. Muck, “Integration of an Ontological Information Security Concept in Risk-Aware Business Process Management.” in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008.

[18] C. Wolter, M. Menzel, A. Schaad, P. Miseldine, and Meinel, C. “Model-driven business process security requirement specification.” *Journal of Systems Architecture*, vol. 55, no. 4, pp. 211-223. 2009.

[19] A. Rodrǵuez, E. Fernandez-Medina, J. Trujillo, and M. Piattini, “Secure business process model specification through a UML 2.0 activity diagram profile.” *Decision Support Systems*, 2011, vol. 51, no. 3, pp. 446-465.

[20] Y. Alotaibi and F. Liu, “Business Process Modelling Towards Derivation of Information Technology Goals,” in *Proceedings of the 45st Annual Hawaii International Conference on System Sciences*. Maui, Hawaii, US. 2012.

[21] M. G. Object, “OMG Unified Modeling Language (OMG UML),” *Superstructure*, vol. 2, on. 1.2. November 2007.

[22] A. Van Lamsweerde and E. Letier, “Handling obstacles in goal-oriented requirements engineering.” *IEEE Transactions on Software Engineering*, 2000, vol. 26, no. 10, pp. 978-1005.

[23] E. Yu, “Modelling Strategic Relationships for Process Reengineering,” PhD Thesis, Department of Computer Science, University of Toronto. 1995.

[24] E. S. K. Yu, “Towards modelling and reasoning support for early-phase requirements engineering.” in *Proceedings of the Third IEEE International Symposium on Requirements Engineering*. 1997.

[25] G. Sindre and A. L. Opdahl. “Eliciting security requirements by misuse cases.” in *Proceedings of 37th International Conference on Technology of Object-Oriented Languages and Systems*, TOOLS-Pacific 2000.

[26] E. S. K. Yu, J. Mylopoulos, and Y. Lesperance, “Modelling the Organization: New Concepts and Tools for Re-Engineering.” *IEEE Expert*, 1996: pp. 16-23.

Youseef Alotaibi received the Bachelor of Computer Science degree from Teacher College at King Abdulaziz University, Jeddah, Saudi Arabia and Masters of Information Technology (Computer Network) degree from La Trobe University, Melbourne, Australia. He is currently doing a PhD degree on Computer Science at La Trobe University, Melbourne, Australia. He is awarded Saudi Arabian Higher Education Ministry and Umm Al-Qura University Postgraduate Research Scholarship for his master and PhD studies.

He is currently a lecturer in the Department of Information Technology and Computer Science at Makkah College, Umm Al-Qura University, Makkah, Saudi Arabia. Previously, he worked as an assistant lecturer in the same University. His research areas include Business Process Modelling, Reengineering, Information System, System Security and Electronic Commerce.

Mr. Alotaibi is a member of IEEE (Computer Society).. He has authored about 5 conference and journal papers. He was an invited reviewer for HICSS-45 Conference.

Fei Liu received the Bachelor of Mathematics degree from Zhejiang University, Hanzhou, China and Masters of Computer Science and PhD degree from La Trobe University, Melbourne, Australia. She was awarded La Trobe University Postgraduate Research Scholarship for her PhD study.

She is currently a senior lecturer in the Department of Computer Science and Computer Engineering, La Trobe University. Previously, she worked as a lecturer in the University of South Australia and Royal Melbourne Institute of Technology. She also worked in Ericsson Australia as a software engineer. Her research areas include Automated Reasoning, Semantic Web and System Analysis and Design

Dr. Liu is a member of IEEE (Computer Society).. She has authored/co-authored more than 50 conference and journal papers.