

Use Case Application in Requirements Analysis using Secure Tropos to UMLsec - Security Issues

Mohammad Nazmul Alam
Department of CSE,
University of Science and
Technology Chittagong
USTC, Foy's Lake, Chittagong,
Bangladesh

Sohrab Hossain
Department of CSE,
University of Science and
Technology Chittagong
USTC, Foy's Lake, Chittagong,
Bangladesh

Kazy Noor E Alam
Siddiquee
Department of CSE,
University of Science and
Technology Chittagong
USTC, Foy's Lake, Chittagong,
Bangladesh

ABSTRACT

Information Systems Security is one of the most critical challenges presently facing nearly every one of the organizations. However, making certain security and quality in both information and the systems which control information is a difficult goal necessitating the mixture of two wide research disciplines which are typically separate: security engineering and secure software engineering. Security engineering has an extensive history, and has focused generally on providing advances in security models, techniques and protocols, but it remains in a steady state of the development. Secure software engineering, however, has emerged relatively recently, but is growing quickly and is paying attention on the integration of security into software engineering techniques; models and processes, in order to build up more secure information systems. The main aim of this paper is to show the requirements analysis using Secure Tropos to Umlsec. Secure tropos is a security oriented extension of tropos methodology and UMLsec is a security oriented extension of standard UML model. To do this we identify different transformation rules and we apply these rules by identifying different steps. We use kent Modeling Transformation Language as a Transformation Language to transform the secure tropos model to UMLsec model and then finally we use a case study to exemplify these rules.

General Terms

Transformation Language, Requirement Captures, Information Systems Security & Issue.

Keywords

Use Case, UMLsec, Tropos, Actor, Goal.

1. INTRODUCTION

Software is becoming more and more useful in our everyday life. We are using software systems in every aspects of the human society. Its vast usability day by day is creating more complex and the consequence the developer and the user facing new problems everyday in the real world situation. Widely use of information and sensitive data it is needed to keep secure the information that not only useful for the organizational purpose but also for individual users. So therefore security requirements are one of the most crucial issues for the development of the critical software systems. However it is also equally important that the systems are developed that meets the user needs [2]. If it is not fulfill the user criteria why they need it.

However traditionally, security would not be considered as a vital issue when developing the system. the software industry emphases on the reliability and performance of the software rather then security issues of the software and it takes a little

account after the definition of the system [2] as a result security may conflict and consequence leads to problems to human life, financial infrastructure, personal privacy and so on. Moreover security requirements differ from user requirements so it varies from one system to another [1]. but typically common principle of security is user authentication for confidentiality, Integrity for the preventing of unauthorized alteration and availability for the preventing of unauthorized withholding of information or resources [3]. Security engineering is now thought one of the branches in the software engineering development process. Traditionally it was different research area of investigation and that apart from software engineering and as a result security was not considered during the software development life cycle (SDLC) [2].

2. TROPUS METHODOLOGY

Tropos is a software development methodology adapted to describe both the organizational environment of a system and the system itself. Tropos adapts the i* modeling framework [4] which uses the concepts of actors, goals, tasks, resources and social dependencies for defining the obligations of actors (dependees) to other actors (dependers). Models in tropos are acquired as instances of a conceptual meta-model resting on the following concepts. An actor models entity that has strategic goals and intentionally within the system or the organizational setting. An actor represents a physical or a software agent as well as a role or position [4]. A Goal represents actor's strategic interests. There is hard goals and soft goal. There is no difference between hard goals and soft goals except the soft goal having no criteria for decide whether they are satisfied or not. Soft goals are typically used to model non functional requirements. A Plan (also known as task) represents, at an abstract level, a way of doing something. The execution of a plan can be means for satisfying a hard goal or a soft goal. A resource represents a physical or an informational entity. The main difference with an agent is that a resource has not intentionality. A dependency between to actor indicates that one actor depends, for some reason, on another in order to attain some goal, execute some plan, or deliver a resource.

The former actor is called the depender while the later is called the dependee. the object around which the dependence centres is called the dependum [2]. There are four main phases of software development that covers tropos methodology. These are early requirement analysis, late requirement analysis, Archetectoral design and detailed design. Early requirement analysis concerns with the initial study to understand the problem an existing organizational setting. The production of this stage is an organizational form, which identifies related actors, their particular dependencies and the

security constraints forced to those actors. The system of the organization is explained surrounded by its operational environment along with relevant functions and security requirement in the late requirement analysis stage, this explained models that contain a small number of actors, which include a number of dependencies and security constraint. These dependencies describe the system functional requirements, while security constraints delineate the system security requirement.

Architectural design, where the systems global architecture is defined in terms of subsystems, interconnected through data and control flows, within the framework, subsystems are represented as actors and data controlled, interconnections are represented as system actor dependencies. In addition, during this stage, different architectural styles are analyzed taking into account security and other non functional requirement of the system and secure capabilities identified and assigned to the different actors of the system to satisfy the secure entities.

Detailed design, where each architectural component is further defined in terms of input,output,control and the security aspects analyzed in the previous stages for this stage.tropos is using element of Agent Unified Modeling Language(AUML) to complement the feature of i*.

2.1 Secure Tropos

Tropos methodology does not have adequate concern about security during the development process of multi agent system. Secure tropos are introduced to enable the developers to add security constraint in the tropos methodology in the respect of security. According to Mouratidis [11] "secure tropos extends the tropos methodology by adding security concerns during the development of the Information Systems."and a security constraint defined as a "restriction related to security issues, such as privacy, integrity and availability, which can influence the analysis and design of a multi agent system under development by restricting some alternative design solution, by conflicting with some of the requirement of the system, or by refining some of the systems objectives"[2]. Secure dependency uses security constraints that must be fulfilled the dependency to be satisfied. Both the depender and the dependee have to agree for the fulfillment of

the security constraint in order for the secure dependency to be valid. That means the depender expect from the dependee to satisfy the security constraint and also that dependee will make an afford to deliver to dependum by satisfying the security constraint. There are three different types of secure dependency. In a dependee secure dependency the depender depends on the dependee and the dependee initiates security constraints for the dependency. In a depender secure dependency, the depender depends on the dependee and the depender initiates security constraint for the dependency. In a double secure dependency, the depender depends on the dependee and both the depender and the dependee introduce security constraints for the dependency. Both must satisfy the security constraint introduced to achieve the secure dependency. A secure entity represents a secure goal, a secure tasks or a secure resource. A secure goal represents the strategic interest of an actor with respect to security. Secure goals are mainly introduced in order to achieve possible security constraint that are imposed to an actor or exists in the system. However, a secure goal does not particularly define how the security constraints can be achieved, since alternative can be considered. A secure task or plan is defined as a task or plan that corresponds to a particular way for fulfilling a secure goal. A secure resource can be defined as an informal entity

that is related to the security of the multi agent system. Secure capability represents the ability of an actor or agent to achieve a secure goal.

2.2 UMLsec

UMLsec is UML extension for secure systems development [6] [7]. Security requirement such as secrecy, integrity and authenticity are presented as requirement elements by the UMLsec extension. The requirements are used to assess different kinds of diagrams and to show expected vulnerabilities. The security requirement is used when it fulfill the security policy and also it can make sure that the requirements meets the system that provide by the UML specification.UMLsec outlines the security engineering knowledge so it makes easier to understand the developers even non expert in this field [6].

The UML profile uses the typical UML extension mechanisms stereotypes, tagged values and constraints [2].Stereotypes are used together with tags to formulate security requirements and assumptions on the system environment. Constraints give criteria that determine whether the requirements are met by the system design. [6][7]

Stereotypes describe new types of modelling elements. Stereotype names are written in double angle brackets and are attached to a model element. The model element is then interpreted according to the extended meaning ascribed to the stereotype.

There are some stereotypes that used in UMLsec are <<secure Links>>, <<Secrecy>>, <<LAN>> etc. For example the stereotypes <<secure links>> means the security requirement

Links>>, <<Secrecy>>, <<LAN>> etc.For example the stereotypes <<secure links>> means the security requirement for communication is fulfilled by the physical layer. [6][7].

Table 1: Subset of the UMLsec profile

Stereotype	Base Class	Tags	Constraints	Description
Critical	Object, subsystems	Secrecy, integrity Authenticity		Critical object or subsystems
Data security	subsystem	Adversary	Secrecy, integrity And Authenticity	Basic data security requirement
Secure links	subsystem	Adversary	Secrecy, Integrity And Authenticity matched by links	Enforces secure communication links
LAN	Link, node			LAN connection
Internet	Link			Internet connection

3. TRANSFORMATION

Model transformation technologies are key enables for model-driven software development [8]. It enables to map models to other models and establish correspondence between different models. This works will illustrate how to write rules for a real example: a secure tropos and UMLsec model transformation. Each step presents a new mapping guideline and explains the concepts; where meaningful and different diagram illustrates the rule.

3.1 Run-through: Using TROPOS and UML for Security

Secure tropos is the security extension of tropos methodology that composed the security constraint into the system requirement modelling. It covers four main software development phases that is early requirement, late requirement analysis, architectural design and detailed design.

And UMLsec is the extension of UML that is used to capture the critical security in system development. The major diagrams in UMLsec are activity diagram, class diagram, sequence diagram, state chart diagram and Deployment diagram. According to Jurjens [10] activity diagram hold secure control flow coordination, a class diagram controls the security level during data exchange, a sequence diagram hold security critical interaction, a state chart diagram security preserved within object and a deployment diagram contains the physical security requirement. According to Mouratidis [9] Tropos model is mainly analysis model of the system and UML are mainly design model of the system.

3.2 Early Requirement for the Estate Agent

3.2.1. Purpose: The aim of the early requirement process is to determine the domain stake holder's with respect to major actors and the goals involved for the manual agent system. Actors sometimes will depend on others knows as dependencies. Dependencies on the other hand implies of developing the overall goal how the major actor is depending on the others. During the early requirement process we will just determine the actors and their task. But the dependencies involve with the estate agent will be describe in the later stage of late requirement. Once the actor's are found we can proceed to design the actor's diagram for the early requirement.

3.2.2 Actors involved with early requirement: The term actors does means that the task performed by someone to get the goal or the entity that has some goal. The system doesn't control the actors rather it's defined outside the system.

According our case study we have three major actors which will act for the overall implementation as a major actor. The actors are **Property owner**, **Property buyer** (who has preliminary got the offer) and the **Agent** (who is the communication media).

3.2.3 Overall description of the early requirement:

Early requirement starts with communicating with the estate agent from both parties. Both party on the other sense property buyer and the property owner. Property owner comes to the estate agent to sell, rent or mortgage of his/her house/flat/office/shop and the property buyer comes to buy, rent or mortgage of the house/flat/office/shop. Once the estate agent assures them that they would be able to perform their task they want then the both parties get the assurance and they think they came to the right place. According to the sectro tool the assurance can be a soft goal which we have named the **authentication**. Authentication is done by all parties.

Major goal involved with the property owner is enquiry for the property, property type, view property, legal issues which normally implies the final agreement and payment where the payment is obtaining through the agent.

There are some soft goal involved with the property owner is **authenticated owner** where the authentication done by the agent. **Authenticated agent** where the authentication is done by the property owner. Last soft goal involved with the property owner is **viewing property** after confirmation.

Major goal involved with property buyer is property **enquiry**, **buy property**, **rent**, **mortgage**, **view property for buy/rent/mortgage**, **final agreement** (legal issues) and **payment** where the buyer is paying to the agent for the specific task.

Soft goal involved with the property buyer is **authenticated buyer** where buyer is authenticating the agent to work for him to do the task he wants. **Authenticated agent** where the agent is agreed to perform the buyer task. As the agent is the communication media between both the parties all the major goals and the soft goals would be connected to the agent. Goal model describe the internal analysis of an actor. During the early requirement goal model, we have expended each actor and their goals. This is done normally expending each goal. Suppose rent is a goal if we expend the goal it will have several branches such as type of letting means house, flat, shop, office etc. When we have determined all the expansion necessary for the goals we have started the goal model of each actor.

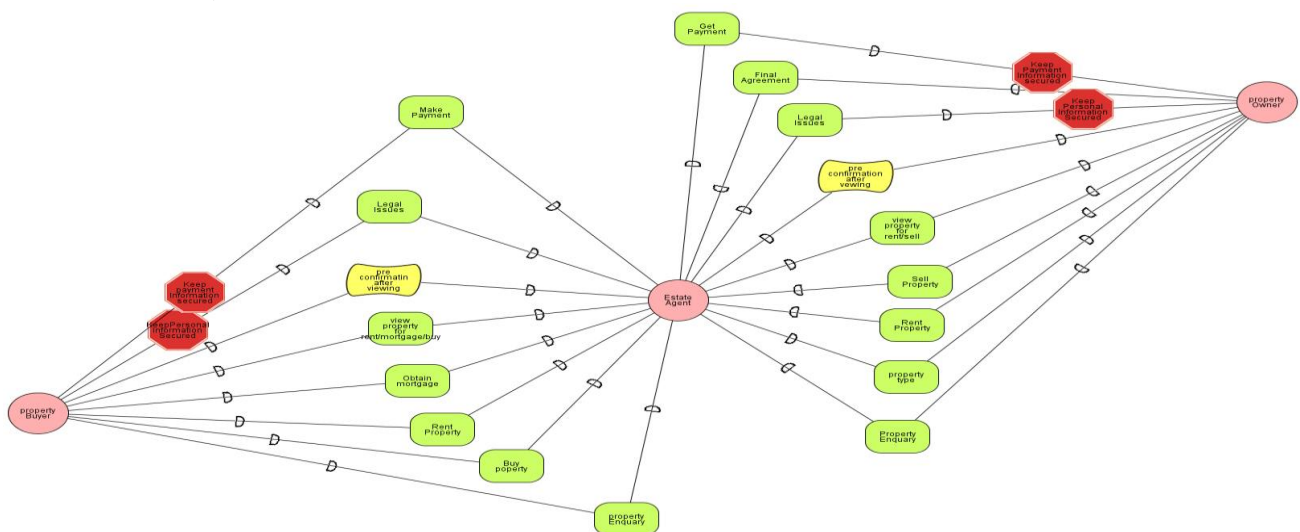


Fig 1: Early Requirement Diagram

3.3 Analysis of Secure Tropos requirement Model

After analysis of secure tropos requirement model we see in tropos, there is a set of actors who depends on the other actors to fulfill their goal. The object around which the dependence center is called the dependum[2]. Dependum could be a set of goal (soft goal and hard goal), Plan and resource where the dependees are required to perform all these activities. Secure tropos add the security constraint on those activities and dependencies where is appropriate. More precisely we can say secure tropos has:

- Actor (dependee and depender)
- Dependum (Plan, Resource, Hard goal, Soft goal)
- Security constraint,

3.4 Mapping Secure Tropos to UMLsec

This section provides the mapping guidelines to transform the secure tropos into UMLsec diagram and the following chapter will validate these mapping guidelines by showing the diagram followed by the case study.

3.4.1 Map the secure tropos early requirement analysis model into UMLsec use case diagram

Use case diagram in UMLsec captured the security requirement. Use case diagram is to capture the dynamic aspect of the system.

To map the UMLsec Use case diagram following items need to identify:

- use case which represents functionalities,
- Actors; an actor represents anything that interacts with the system. It can be only input the information to the system. It can be only receive the information from the system and or both [5].
- Relationships among the use cases and actors; a relationship indicates the interaction between the actors and the use cases.
- security requirement capture; Security requirement is applied by identifying the adversary of the system.

Step1

Identify the actors and use case in UMLsec and map it into secure tropos Actors and dependum. The functionalities of use case can be mapped into secure tropos Goal and Plan. It will be depend on which one will meet the requirement and secure tropos early requirement analysis will be considered in the Use case diagram. For example, in secure tropos we use three actors and their different goals and we map these as a actor and use case in UMLsec diagram.

Step2

Identify the critical actors, secure plan and secure goal to impose the security constraint and map it into security stereotype in UMLsec to capture the security requirement.

For example, in secure tropos actor Buyer buys the property from another actor Agent and they make payment to execute their goal. Here buy property is a dependum between those actors and another dependum make payment is invoked here to fulfill the goal where the security constraint will be used to keep the payment information secret. In UMLsec buyer and agent will be same as actor and buy property and make payment will be transformed as a use case where <<secure exchange>> security stereotype should be imposed to remain the payment information secret.

3.5 Transformation Language

We use Kent Modelling Transformation Language (KMTL) for transforming the Secure Tropos to UMLsec model. It was developed by the University of Kent. This approach has no predefined specification language but it was based on a set of java libraries including OCL support for defining activation constraints.

KMF, Model Driven Development concepts use the model, transformation and their instances [12].

```
Transformer Secure Tropos to UMLsec
tropos:secure tropos
uml:umlsec

Transformer secure tropos early
requirement to umlsec use case diagram
secure tropos:early requirement
umlsec:umlsec Use case

mapping secure tropos actor 2 umlsec
actor
sta:secure tropos actor
usa:umlsec actor
stdp:secure tropos dependum
uuc:umlsec use case
when
{
secure tropos actor 2 umlsec actor(sta,usa)
secure tropos dependum 2 umlsec use
case(stdp,uuc)
}
mapping secure tropos actor 2 umlsec
actor
sta:secure tropos actor[actor
name=tan,dependum=dp]
usa:umlsec[actor name=uan,use case=uc]
When
{
secure tropos actor name 2 umlsec actor
name(sta.tan,usa.uan)
and
secure tropos dependum 2 umlsec use
case(stdp.dp,uuc.uc)
}
```

Fig 2: Transformation of Secure Tropos to UMLsec Model

4. CASE STUDY

The case study that will be used to describe the security analysis and used to show the evidence of transformation of secure tropos model into UMLsec. We use an Estate Agent Management System as our case study. The main purpose of the Estate Agent is to perform the task of selling, letting (rent), mortgage and managing property. This task is to perform between seller and buyer. They don't sell their property directly rather they just make the arrangement to sell others property. On the other hand they are the communication media between seller and buyer. Once they knew that someone wants to sell their property then the agent property inspector inspect the house to gather the information regarding property. The agent makes a mutual agreement with seller that they will find the buyer to sale his/her house. Once the agreement done they put the property advertisement to the media for finding buyers.

For selling the follow some such as how to sell the house quickly, step-by-step guide to move the house, detailed guidance to sell the house right time and right place. For buying determine, terms of buying a property, type of buyers (whether first or second time). Step-by-step guide to buy the property, surveys explained (such as Valuation), can he/she afford the house? Why they want to buy. For renting the procedure can be guide to rent, determining tenant's right, type of rent furnished or unfurnished.

Finally we can now say that the term "Estate Agent" is a communication media who has the legal right to work for the individuals to sales, letting, buying other property.

4.1 Diagram of UMLsec Model

We are using these diagrams to comply the transformation rules between secure tropos and UMLsec. It does not imply the whole process of the case study. As our intention to show the transformation process rather than analysis of the whole process of the case study. We use here early requirement diagram to show the transformation of secure tropos model to UMLsec model.

We have already described about the secure tropos model of early requirement. Now we only explain here UMLsec model to verify the mapping rules how it has been derived from secure tropos model to UMLsec model. And please note that we have shown here part of the system to visualize only the transformation of the system rather than the whole process of the system.

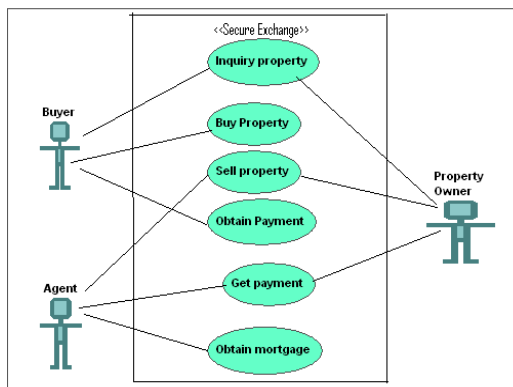


Fig 3: Use Case Diagram

UMLsec Use case Diagram in fig 3 We have identified actors and their functionalities and secure exchange.

5. CONCLUSION

The main contribution of this paper is the introduction of the transformation from secure tropos model to UMLsec model using the concept and notation of secure tropos model and UML diagrams. The presented approach has identified different steps for mapping the secure tropos to UMLsec diagrams. We have presented the secure tropos early requirement analysis model into use case diagram in UMLsec model as a transformation. Nevertheless the transformation

approach is not complete by all means. We need more attention to complete this effort. Further work includes providing a formal rules and guidelines for transforming the secure tropos model to UMLsec model.

6. REFERENCES

- [1] R. Anderson, "Security Engineering-A Guide to Building Dependable Distributed Systems", John Wiley & Sons, New York, NY.2001
- [2] H. Mouratidis and P. Giorgini (eds), "Integrating Security and Software Engineering: Advances and Future Vision", Idea group, IGI Publishing Group, 2006.
- [3] A. Jacobsson, "Privacy and Security in Internet-based Information Systems", Bleking Institute of technology, Sweden, Doctoral Dissertation Series No.2008:02, 2008.
- [4] Yu, E., K., "Modeling strategic relationship for process reengineering", PhD Thesis, University of Toronto, 1996
- [5] Terry Quatrani, "Visual Modeling with Rational Rose 2002 and UML", Addison-Wesley, 2004.
- [6] Jan Jürjens, "Secure Systems Development with UML", Springer publishing, 2003.
- [7] "Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification", 2008, <http://www.omg.org/spec/QVT/1.0/PDF>
- [8] Sebastien Demathieu, Catherine Griffin, Shane Sendall, (2005) "Model Transformation with the IBM Model Transformation Framework" Level: Intermediate
- [9] Mouratidis, H; Jürjens, J; Fox, J., "towards a comprehensive framework for secure systems development", in: Dubois, E; Pohl, K. (Eds) CAiSE 2006, LNCS 4001 pp 48-62.
- [10] Jan Jürjens (contributing P. Shabalin, S. Höhn, S. Meng) "Secure Enterprise Applications with UML" Software & Systems Engineering Informatics, TU Munich, Germany, 2003
- [11] secure tropos website: <http://securetropos.org/>
- [12] D.H Akehurst, et al. "Kent Model Transformation Language", University of Kent, Canterbury, 2006