POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Contextual, Requirements-Driven, Adaptive Access Control

Relatore: Prof. Carlo Ghezzi
Correlatore: Prof. Bashar Nuseibeh,
Dott. Liliana Pasquale,
Dott. Luca Cavallaro,
Dott. Mazeiar Salehie,
Dott. Raian Ali.

Tesi di Laurea di:
Claudio Menghi,
matricola 755045

Anno Accademico 2011-2012

To my great aunt, Maria Regina,
for all of her unending support.

Sommario

I sistemi di sicurezza includono tra i loro obiettivi principali quello di proteggere dati e risorse contro modifiche, accessi e divulgazioni non lecite, consentendone, allo stesso tempo, l'accesso e la disponibilità agli utenti autorizzati. Le regole di controllo di accesso specificano sotto quali condizioni un particolare utente può utilizzare una determinata risorsa presente nel sistema [MC11]. Nonostante la definizione e la manutenzione di tali regole non sia semplice, la loro gestione avviene con tecniche ad-hoc, lasciando il sistema vulnerabile ad attacchi e violazioni [HA09].

Un sistema di controllo di accessi adattativo, dovrebbe essere capace di modificare le politiche di controllo di accesso in relazione ai requisiti del sistema, al valore delle risorse da proteggere e al contesto nel quale l'applicazione lavora. I requisiti del sistema rappresentano funzionalità e proprietà che il software deve soddisfare [Mof99] e descrivono le ragioni per cui le politiche di controllo di accesso sono definite e utilizzate [TS94]. La considerazione dei requisiti facilita l'analisi del problema di sicurezza, evidenziandone le soluzioni senza considerarne i relativi dettagli implementativi [Mof99]. In tale analisi rivestono un ruolo fondamentale i requisiti di sicurezza, la cui corretta determinazione rappresenta una condizione necessaria al fine di proteggere le risorse da possibili attacchi. Il valore di tali risorse riveste un ruolo centrale nella selezione delle politiche di controllo di accesso da applicare durante l'esecuzione del sistema, influenzando sia i requisiti del sistema stesso, sia il comportamento di eventuali attaccanti [MSLPIORA11]. Per esempio, se il valore delle risorse aumenta è necessario selezionare delle politiche più restrittive per garantirne la protezione delle risorse da potenziali attacchi e soddisfare i requisiti di sicurezza dell'applicazione. Il contesto è un altro fattore fondamentale nella determinazione delle politiche di controllo di accesso da applicare. Quando il contesto cambia, diversi requisiti (includendo quelli di sicurezza) possono essere attivati o disattivati e delle funzionalità appropriate devono essere selezionate per soddisfarli [RA10]. Requisiti, risorse e contesto sono solitamente trascurati dai sistemi di sicurezza attualmente disponibili sul mercato, rendendoli, di conseguenza, poco flessibili.

Questa tesi presenta un approccio innovativo, capace di considerare i requisiti del sistema, il valore delle risorse da proteggere e il contesto nel quale l'applicazione lavora, come elementi chiave per lo sviluppo di un sistema di sicurezza adattativo. Il modello dei requisiti del sistema viene esteso al fine di considerare tali elementi, e tradotto in una rete causale utilizzata per analizzare il rischio di eventuali attacchi e l'utilità delle diverse misure di sicurezza. Basandosi su tale analisi vengono selezionate, durante l'esecuzione dell'applicazione, l'insieme delle misure di sicurezza da applicare.

L'approccio presentato è supportato da*SecuriTAS* (a tool to engineer adaptive security) un tool che consente lo sviluppo di sistemi di sicurezza adattativi. SecuriTAS permette di utilizzare il modello definito durante la fase di analisi dei requisiti (STrioM), durante la fase di esecuzione del software, al fine di selezionare dinamicamente l'inisieme delle misure di sicurezza necessarie per proteggere il sistema. Per raggiungere tale obiettivo il modello sviluppato (STrioM) viene convertito in una rete causale (FCN). Le risorse da protegge e il contesto di esecuzione dell'applicazione vengono continuamente monitorate e utilizzate per riconfigurare la FCN, mentre le misure di sicurezza vengono applicate per mezzo di opportuni attuatori.

Nel caso di un problema di controllo di accesso l'insieme delle misure di sicurezza è costituito dalle politiche di controllo di accesso. Applicando la meodologia proposta a tale problema è possibile progettare un sistema di controllo di accesso adattativo. Viene descritto come applicare la metodologia proposta a tale caso e progettato un prototipo di una applicazione capace di regolare dinamicamente l'accesso ad un particolare ufficio.

Abstract

Security systems are usually used to protect data and resources against unauthorized modifications, accesses, and disclosures but, at the same time, resources availability must be guaranteed to the legitimate users. Access control policies are rules that specify the access privileges of each user [MC11], and are among the most used technique to provide assets' security. Defining and managing access control policies is far from being a trivial process. Traditionally, access control policies have been specified in ad-hoc manner, leaving the system vulnerable to security breaches [HA09].

A self-adaptive access control system should be able to modify the access control policies according to the value of the assets that need to be protected, the context in which the application is working and the system requirements. Unfortunately, these elements are usually neglected in the access control systems development. Requirements are "something called or demanded, a conditions which must be complied with" [Mof99] and represent the reasons why access control policies are specified and enacted [TS94]. Consider requirements in the access control policies specification, has the advantage of ensuring that other solutions to the problem, which are not necessarily regarded so naturally as low level policies, are considered [Mof99]. Security requirements are a particular kind of requirements which concern with the protection of assets from harms. Assets have a central role in security and may influence other security concerns, such as threats, attacks, vulnerabilities, risks, security goals, and security controls [MSLPIORA11]. When an asset become more valuable, the system has to select more restrictive access control policies to satisfy security requirements. Another factor that impact on the system requirements is the context in which the application is working [RA10]. More precisely, when the context changes different security concerns can be activated or deactivated. However, requirements, assets and context are usually neglected in developing of adaptive security systems.

This thesis presents an approach that considers requirements, assets and context, as keys elements in the development of an adaptive systems. First

of all, the requirement model is extended to take care of the different security concerns. Secondly, it is translated into a causal network to analyze, at run-time the risks of attacks and the utility of the different security controls. Based on this analysis the set of security controls to apply at run-time are selected.

The approach presented is supported by a tool called *SecuriTAS* (a tool to engineer adaptive security). This tool translate the model developed during the requirements analysis (STrioM) into the relative fuzzy causal network (FCN), which is used to dynamically select, at run-time, the set of security controls to apply. Assets and context are monitored by means of suitable sensors. When assets or context change, the value of each node of the FCN is updated, and the fuzzy causal network is re-evaluated. SecuriTAS uses a set of effectors to enact the set of security controls necessary to protect the system.

In the access control case, the set of security controls necessary to protect the system consists of the set of access control policies to apply. This approach makes possible to design an adaptive access control system, able to modify the access control policies in relation with the context, the value of the assets, and the requirements of the system. The thesis provides a demo of an application able to dynamically regulate the access into a particular office.

Acknowledgements

The work presented in this thesis would not have been possible without the help and support of the people listed below.

First and foremost I want to thank Prof. Carlo Ghezzi who proposed me to develop this thesis at Lero. I cannot find words to express my gratitude to my supervisor, Prof. Bashar Nuseibeh who suggested me useful advices during this work. I am also grateful to the whole Lero team: Liliana Pasquale, Raian Ali, Mazehiar Salehie, Inah Omoronyia, Luca Cavallaro for participating in many lively discussions and making lots of thoughtful comments and suggestions, I would not have been able to complete this work without their support.

I also would like to thank The Irish Software Engineering Research Centre (Lero) for its hospitality, and for its dynamic and supportive work environment. I am also grateful to United Technologies Research Centre (UTRC) for their collaboration and their constructive observations.

I am indebted to my friends and all people who supported me in different ways during my studies. In particular, I want to remember my fellow students, the Irish and the homeland friends and express my deepest gratitude to my flatmates Fabio and Oscar for their friendship and advices.

Additionally, I would like to thank my family: my parents Fulvia and Roberto and my brother Michele, who have always encouraged and believed in me.

Finally, it is with immense gratitude that I acknowledge Maria, for all times she sustained me throughout the duration of my studies, for her great patient and unwavering love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Imagination is more important than knowledge."

Albert Einstein

In the last few years, automatic processing of information and data pervades every aspect of the modern enterprises. The importance of data managed by various organizations is continuously growing up and, consequently, the necessity to protect physical and information assets against potential attacks is becoming primary goal for the companies [CIN05]. In this environment, it becomes extremely important to correctly configure the technology instruments devoted to guarantee the secure access/usage of several assets, such as buildings, computer based information systems, files and money.

Access control is one of the major security mechanisms used to achieve confidentiality, integrity and privacy and refers to the regulation over who can reach and interact with each resource [SS94]. Access control is usually regulated by means of policies, which are rules that specify the access privileges of each user [MC11]. Defining access control policies is far from being a trivial process. Traditionally, access control policies have been specified in a doc-manner, leaving the system vulnerable to security breaches [HA09]. In particular, critical assets, requirements and contextual factors are usually not considered systematically in the access control policies specification [HA09].

Requirements are "something called or demanded, a condition which must be complied with" [Mof99]. Security requirements are concerned with the protection of valuable assets [CIN05] and are aimed to satisfy the top level security objectives such as guaranteeing confidentiality, integrity, availability and accountability (CIAA) for the resources under protection [PCoPfIS99]. Early understanding and specification of access control policies are keys to effectively satisfy security requirements. Even if several work has already considered requirements in the definition of the access control policies [FLLD01, CIN05, FP09, MZ08], the role of the assets that need to be protected, and the context in which the application is working, is often neglected.

An asset is anything that has a value for the organization and need to be protected [CPP02]. Assets have a central role in security and may influence other security concerns, such as threats, attacks, vulnerabilities, risk, security goals, and security controls [MSLPIORA11]. For example, if an asset becomes more valuable, the motivations of attackers, the threat level and the likelihood of successful (harmful) attacks increase consequently. In a similar way, if a new asset is added into the system, certain threats may be influenced, and security goals may become more critical. Finally, when an asset is removed from the system, certain threats and attacks may be not possible anymore. These changes may affect other security concerns, such as the set of security requirements. As a consequence, when assets change, the system has to modify its access control policies accordingly. To this aim, fostering asset awareness is fundamental. Assets must be continuously monitored, and used to select the set of policies to apply to continue to satisfy the system security requirements.

The context has a strong influence on the system requirements and vulnerabilities. It may activate a set of requirements and enable a set of security functions that may be necessary to meet the activated requirements [Ali10]. In a similar way, vulnerabilities depend on the context [EYZ10]. When the context changes, different vulnerabilities can be activated and deactivated, and these changes can also have an impact on the access control policies necessary to protect the system. For this reason, security systems must also be context-aware and adapt their behavior according to the current context conditions, such as their location of use and the collection of nearby people and objects [ST94]. Despite several works have already considered the role of the context in the definition of the access control policies [GI97, AS08, MMMA00, CLS^{+}01, CCB08b, CCB08a], they do not explicitly consider requirements and assets in their approaches.

This thesis describes an approach that allow to analyze security problems, by considering requirements, the value of the assets that need to be protected and the context in which the application is working, from the early stages of the software development process. This approach is based on a conceptual model (STrioM Security Trio Model [MSLPIORA11]) able to capture the different concerns of a security problem: goals, requirements, tasks, vulnerabilities, context, assets, security controls, including their mutual relations. STrioM uses an *asset model* to describe assets and their relationships, a *goal model* to describe, users, functional and non-functional requirements, tasks, and security controls, and a *threat model* to describe threats and attacks. Context is used to activate/deactivate portions of the previous models according to the environmental condition in which the application is working. This model is then translated into a *fuzzy causal network (FCN)* and is used, at run-time, to analyze system security in different situations, and enable, when necessary, a set of security controls to mitigate the security threats. The assets that need to be protected and the context in which the application is working, are monitored by means of suitable sensors. When the context or the assets change, a certain set of requirements or vulnerabilities might be activated and the available security controls might be more or less effective to reach activated requirements and mitigate existing vulnerabilities.

The approach described in this thesis has been developed in a prototype tool, *SecuriTAS*[1]. This tool develops the activities of the *MAPE* (Monitor Analyze Plan Execute) loop [HS06]. By using sensors, SecuriTAS collects data from the environment in which the application is working (Monitor). The accumulated data are cleaned, filtered and pruned and exploited to infer trends and information (Analyze). These information are used to decide how to act on the executing system and the security measures to apply(Plan). Finally, the security measures selected are enacted on the system using suitable effectors (Execute). Using a realistic system as a case study, it is highlighted how SecuriTAS can be used to enforce dynamic access control. SecuriTAS has been used to develop an adaptive access control system able to regulate the access to an office depending on the context, the criticality of the assets present in the office, and the system requirements.

1.1 Research Context

Security, and in particular access control, requirements engineering and self-adaptive software are the main areas related to this thesis.

Requirements engineering is the branch of software Engineering concerned with the real world goals for, functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families [NE00].

[1]Liliana Pasquale, Claudio Menghi, Mazeiar Salehie, Luca Cavallaro, Inah Omoronyia and Bashar Nuseibeh, "SecuriTAS: A Tool for Engineering Adaptive Security" [LPN12]

Requirements engineering addresses the elicitation, the analysis and the interpretation of the stakeholders' needs. In particular, requirement engineers try to capture, investigate and specify the stakeholders requirements, perceived as goals, to understand what the system has to do and why. An important aspect is that requirements specification must provide all information necessary to the developers to implement a system that complies with the stakeholders' needs. Security requirements are a particular type of requirements that must be satisfied to achieve the CIAA security goals.

Security concerns the protection of assets against unauthorized disclosure, transfer, modification, or destruction, whether accidental or intentional [IEE01]

Security is generally recognized as having a critical role in software systems, as security incidents can be costly. For example, Nick Leeson's trading resulted in losses of over £800 million and caused the bankruptcy of Barings Bank; similarly, John Rusnak defrauded the Allied Irish Bank of a similar amount in 2002 [CIN05]. One principle of security is that it is not absolute: it is just a factor to balance against the cost of loss and fraud [GD07]. As computing technology becomes more tightly integrated into the fabric of everyday life, it is imperative that security mechanisms become more flexible and less intrusive [CFZA02]. To this end, security systems can benefit from self-adaptation [CCB08b, CLS+01].

A Self-adaptive software is able to modify its own behavior in response to changes in its operating environment. By operating environment, we mean anything observable by the software system, such as end-user input, external hardware devices and sensors, or program instrumentation [OGT+99].

Developing software systems able to detect the changes and independently take decisions is a primary goal of adaptive system engineering. In particular, an adaptive security system is a kind of a self-adaptive system, used to protect assets against threats and attacks and satisfy security requirements.

Adaptive security is intended to indicate that security policies and mechanisms can change in some automated or semi-automated fashion in response to events [Mar]

Access control systems are a particular type of security systems that can benefit from adaptive security [CCB08b, CLS+01].

Access control is concerned with permitting only authorized users (subject) to access services and resources (targets) [DBSL02].

Organizations use access control systems to regulate who can interact with a set of critical resources. In physical security, access control refers to the regulation of access to a property, a building, or a room, while in computer security, access control refers to the usage of sensitive and valuable files and information. Access control is usually regulated by means of hight-level rules called access control policies, that determine how accesses are controlled and access decisions determined [SSS94].

1.2 Research Questions

Traditionally, access control policies have been specified in ad-hoc manner, leaving the system vulnerable to security breaches [HA09]. Context information, asset values and system requirements, are usually neglected in the definition of access control policies. For this reason, this thesis is aimed to investigate the advantages of considering such elements in the policies specification, and to develop an adaptive access control system able to select, at runtime, the set of access control policies that prevent unauthorized access.

This thesis develops an approach to guide software engineers in building adaptive access control systems, from the requirements analysis to the enactment of the system at run-time. First, this thesis describes a model that captures the interplay between the different elements involved in a security problem. Then, this model is used to to select the access control policies that have to be applied. The analysis of the following questions is necessary to meet these objectives:

- *Why access control and requirements are related?*
  The problem of access control can be tackled considering the application and the security requirements of the system. In this way, on the one hand, it is possible to guarantee that the users are granted the access to the resources they need to satisfy the application requirements. On the other hand the system can ensure that unauthorized accesses to the protected resources are prevented.

- *Which is the role of the context in the access control problem? How context analysis affects the design of access control systems?*
  Context impacts on the different security concerns requirements. When context changes, different security concerns (i.e., requirements and vulnerabilities) can be activated or deactivated. Proper access control policies are enforced to satisfy security requirements, and, as a consequence, they must change accordingly. For example, in a building management system, in case of fire (Context changes), the access control system has to modify its policies to guarantee the access to the fireman and put the fire out (requirements changes).

- *How assets' value and criticality influence the access control system?*
  Security refers to the protection of valuable assets against intentional harm. As the value of the assets changes, different security requirements may be activated or deactivated. This is reflected in different choices of the access control policies. For example, suppose that a new expensive device is placed inside a specific office. As the value of the assets to protect increases, the system has to apply more restrictive access control policies to satisfy security requirements.

- *Which models/methodologies could be used to describe an adaptive security problem at the requirements level?*
  This work is based on two different approaches. It leverages the approach proposed by Salehie et al. [MSLPIORA11] which consider assets as first-class entities in engineering secure systems. Using an asset model the approach describes assets and their relationships. This model is related to the requirements of a system, expressed through a goal model and the objectives of an attacker, expressed through a threat model. This model is translated into a fuzzy causal network (FCN) and used to modify the behavior of the system at run-time. The second approach has been proposed by Ali et al. [RA10]. It provides a contextual goal model that allows to analyze the impact of the context on the system requirements. The context in which the application is working is used to activate/deactivate requirements. The set of functionalities to apply at run-time are selected according to the activated requirements.

- *How the requirements model can be used to develop an adaptive security system?*
  Security systems, at runtime, use suitable sensors to monitor their context and the value of the assets that need to be protected. These elements impact on the other security concerns and on the system requirements. The system has to choose amongst variants to meet these requirements. In other words, the system needs a reasoning mechanism to derive the set of security controls necessary to protect the system. The set of security controls selected are applied using effectors.

- *How to decide among different access control policies? Which kind of reasoning technique can guide the decision mechanism?*
  At run-time, systems need to select among different security controls according to the context and the value of the assets under protection.

To this end, it is necessary to select a reasoning technique which allows computers to reason automatically based on the information coming from the different sensors present in the environment.

1.3 Contribution of the Thesis

The contributions of this thesis are the following:

- It provides an approach to guide the engineer from the requirements analysis to the enactment of the system at run-time. This approach is composed of two different phases. During the first phase, the system requirements are analyzed and refined. In particular, the asset, threat and requirement models are designed. In the second phase, the model is used at run-time to select the set of security controls that maximize the system security.

- It describes *Security Trio Model (STrioM).* This model is used during the requirements modeling to capture the different aspects involved in a particular security problem and is made by three different submodels: assets, goals and threats model. *Asset model* represents assets and their relationships, *goal model* describes, users, functional, non-functional requirements, tasks, and countermeasures, and *threat model* describes threats and attacks. Context is used to activate/deactivate portions of the previous models according to the environmental condition in which the application is working.

- It describes the *Fuzzy Causal Network (FCN)*: a reasoning mechanism that allows to use the previous model at run-time to select the set of security controls to protect the system. Each concept of the STrioM is translated into a node of the fuzzy causal network, while arcs are used to represent the relations between the different concepts. The fuzzy causal network is used to estimate the impact of the different security controls on risks and attacks.

- It presents *SecuriTAS* (a tool to engineer adaptive security) a tool to regulate the system behavior at run-time. The model developed during the requirements analysis (STrioM) is converted into a ("fuzzy causal network FCN") and used, at run-time to dynamically select the set of security controls to protect the system. The context in which the application is working and the assets to protect are monitored by means of suitable sensors. When these elements change, SecuriTAS update the FCN accordingly and starts the reasoning. Based on the results obtained, SecuriTAS uses effectors to apply the security measures selected.

- It describes how SecuriTAS can be used to realize an adaptive access control system by implementing the *MAPE (Monitor, Analyze, Plan, Execute* loop. The monitor component of the MAPE loop is used to filter and correlate sensor data. The analyzer uses the data coming from monitoring to update the values of the nodes of the FCN. The planning component of the MAPE loop detects the utility of the different configurations of security controls and select the most useful once. Finally, executer enacts the set of security controls selected using suitable effectors.

- It applies SecuriTAS on a concrete case study to evaluate its advantages and limitations. SecuriTAS has been used to develop an adaptive access control system able to regulate the access to an office depending on the context (people present inside the office), the criticality of the assets present inside, and the system requirements. Such system is illustrated by means of a demo. The *AET62 NFC Reader* is used to authenticate the different users, to detect the assets present inside the office and their value, and to monitor the context in which the application is working.

1.4 Structure of the Thesis

The reminder of the thesis is structured as follows:

- CHAPTER 2 details the current state of the art in the areas of adaptive, context-aware and asset-centric access control and of goal-oriented requirements engineering.

- CHAPTER 3 describes the conceptual approach used in this work. This approach was presented by Salehie et al. [MSLPIORA11] and is composed of two main phases: the design of the Security Trio Model (STrioM) and its use at run-time to select the set of security controls to apply. The chapter presents both STrioM and the fuzzy causal network: the reasoning technique used at run-time to select the set of security controls to apply.

- CHAPTER 4 extends the approach proposed by Salehie et al. [MSLPIORA11] with an explicit representation of context. This chapter also analyzes how it is possible to customize the extended approach in order to realize an adaptive access control system.

- Chapter 5 presents SecuriTAS: a Tool for Engineering Adaptive Security. It allows to use STrioM at runtime to analyze changes in security concerns and select the best set of security controls necessary to protect the system. The chapter provides an overview of the tool, a general description of the software architecture, a presentation of the main system components and, finally, a survey on the design and implementation choices.

- Chapter 6 applies SecuriTAS on the ICSRC (Ireland Computer Science Research Centre) case study and describes several scenarios to highlight the advantages and limitations of the approach and the tool proposed in this thesis.

- Chapter 7 summarizes the main contributions of the thesis, discusses its limitations and outlines possible directions for future works.

Chapter 2

State of the art

"Before everything else, getting ready is the secret of success."

Henry Ford

The restriction of access is a mechanism by which organizations protect their assets [CIN05] against potential harms. Enterprises use access control systems to regulate who can interact with the different resources present in the environment [TS94]. Even if access control depends on the context in which the application is working, on the assets that need to be protected, and on the system requirements, an approach able to consider these aspects together in the design of an access control system, is still missing. This chapter describes the state of the art concerned with requirement-driven, context-aware and assets-centric access control. SECTION 2.1 *provides a survey on the adaptive access control literature and analyzes the works that consider requirements, context and the asset value in access control.* SECTION 2.2 *describes the state of the art on goal-oriented requirements engineering, summarizing its advantages. Finally,* SECTION 2.3 *relates the thesis contribution, with the works present in the literature.*

2.1 Adaptive access control

This section reviews the state of adaptive access control. Section 2.1.1 introduces the access control problem. Section 2.1.2 briefly discusses requirements-driven access control literature. Section 2.1.3 describes the works related to context-aware access control. Finally, section 2.1.4 presents the state of the art concerned with asset-centric access control.

2.1.1 Main concepts

Access control is a critical step in securing IT systems, as it aims to prevent unauthorized access to sensitive information [MZ08]. The main goal of access control systems is to regulate access to both informational (logical access control) and physical (physical access control) resources [McA05]. Physical Access Control refers to the practice of restricting entrance to a property, a building, or a room to authorized persons [Edw11]. A possible way to achieve physical access control is through human surveillance, for example a receptionist or a guard, physical devices, such as locks and keys, or other technologies, such as access control systems. Logical access control regulates the access to logical resources, such as files, programs, folders etc, by verifying and validating authorized users, authorizing user access to IT systems and data, and restricting transactions (read, write, execute, delete) according to the user's authorization level [INF].

According to S.Sandhu [SSS94], three security aspects are related to the access control problem: authentication, auditing and authorization. Authentication aims to verify the identity of a user, process, or device, before granting access to some critical resources of the system [SHF$^+$01]. In the physical world authentication systems are used to verify the user identity. Typically these system are based on something the user knows (e.g., a password), something the user has (e.g., an access card), or something the user is (e.g., a fingerprint) [MJaR$^+$10]. Authorization is the process of granting rights to participants to perform an interaction, for instance to access a resource [WMM08], and is usually based on the users' roles. Auditing concerns with a posteriori analysis of all the requests and activities of users in the system. Auditing requires the registration (logging) of all users requests and activities for their later analysis [SS94].

Another important distinction analyzed in [dV11] is the difference between *policies*, *models* and *mechanisms*. Security policies define the (high-level) rules according to which access control must operate. Security models provide a formal representation of the access control security policies and its functioning. Finally, security mechanisms define the low level (software and hardware) functions that implement the controls imposed by the policy and formally stated in the model.

Samarati et al. [dV11] group access control policies in three main classes: discretionary, mandatory and role-based. Discretionary policies (DAC) are based on the identity of the requestor and specify what requestors are (or are not) allowed to do [dV11]. The term discretionary derives from the user's possibility of passing their privileges to other users. This kind of policies do not impose any restriction on the usage of information. Indeed, once the user receives the requested data, he/she can disseminate them. For example, a user who is able to read some data can pass them to other unauthorized users, who might read them without an authorization from the data owner. Discretionary policies rise from cooperative environments, such as the academies.

Mandatory policies (MAC) are based on regulations mandated by a central authority [dV11]. More precisely, this authority assigns a security level to each user and object present in the system. The security level associated with an object (e.g., papers, proposals) reflects the sensitivity of the contained information and the potential damages that could result from its unauthorized disclosure (e.g., top secret, secret, confidential, unclassified). On the other hand, the security level associated with a user, also called clearance, reflects the user's trustworthiness in not disclosing sensitive information to other users not authorized to see it. In the simplest case, the security level is conceived as an element of a hierarchical ordered set (Actor level). Mandatory policies are implemented by preventing information stored in high-level objects to flow into low-level objects, and they rise from rigid environments, such as the military one.

Finally, role based access control policies (RBAC) are an alternative to discretionary and mandatory policies, whose decisions are based on the roles of the individual users inside the enterprise [SR00]. Since more than one person can have the same role, RBAC manages the access for all the individuals belonging to a particular category. One of the most significant disadvantages of RBAC is the direct connection between the permissions and the role of the users. In particular, dividing people into categories based on roles makes it difficult to define granular access controls for each single resource.

2.1.2 Requirements-driven access control

Several approaches explicitly consider requirements in the definition of access control policies.

Liu et al. [LYM03] propose the use of i* [Yu09] Strategic Rationale for defining policies. i* is a modeling language suitable for an early phase of system modeling to specify system requirements. Liu et al. suggest to use the actor boundaries to derive policies. More precisely, role based access control policies can be defined deriving roles from actors definition and permissions from the tasks within the boundary. For example, an actor can be represented by a Family Doctor. The goal of this actor is to provide a regular clinical service. In order to achieve this goal, the actor needs to open new medical records (task). Therefore, it is necessary to grant the permission on the medical record to the Family Doctor (Policy). The approach proposed by Liu et al. provides a systematic way to specify access control policies, but this activity is not supported by a tool.

Crook et al. [CIN05] provide a tool, based on Formal Tropos [FPMT01, CKM01], to derive access control policies from their organizational context and verify these policies. A policy is verified by instantiating domain concepts, and checking whether the policy is consistent with that instantiation. Instantiation and verification is achieved by translating the model into an intermediate language, which is then interpreted by the model checking tool NuSMV [CCGR00] to automatically instantiate objects and ensure model constraints are enforced. A model constraint is represented using the first order predicate language.

Also Fontaine et al. [FLLD01] describes an approach to derive access control policies from their organizational context and verify these policies, but they integrate the KAOS methodology [DvLF93] with the Ponder policy language [spr]. KAOS is a methodology for goal-oriented requirements elaboration. It provides a specification language, an elaboration method and tool support. Ponder is a declarative, object-oriented language to specify and manage security policies for distributed systems. The key contribution of this work is the transformation of operationalized goals into access control policies.

Another model-driven approach for the specification and analysis of access control policies is proposed by Massacci and Zannone [MZ08] and supported by a framework. Their framework is built on the top of SI*, a modeling language used to capture and analyze functional and security requirements. Basically the goal model is extended with different types of relation: own, provide and request. Own indicates that an actor has full authority concerning and disposition over his entitlement. Provide indicates that an actor has the capabilities to achieve the goal. Request indicates that an actor intends to achieve the goal. The relation between the actors within the system are captured by the notions of delegation and trust. Assignment of responsibilities among actors can be made by execution dependency (when an actor depends on another actor for the achievement of a goal) or permission delegation (when an actor authorizes another actor to achieve the goal). Furthermore, the relation trust of execution, models the fact that an actor may prefers to appoint other actors to achieve assigned duties. Massacci and Zannone describes how to use such extended model to specify and analyze access control policies.

The works previously described do not consider the access control system as an adaptive system. More precisely, the access control policies are defined in a static way. The context in which the application is working and the assets that need to be protected, are not considered as key elements in the decision of the access control policies to enact and, therefore, are neglected during the requirements analysis.

2.1.3 Context-aware access control

The understanding of the context in which the application is working plays a key role in the development of an adaptive access control system. When the context changes, the set of security controls necessary to protect the system must be modified accordingly. Several works consider the role of the context in access control.

Giuri and Iglio [GI97] provide suitable mechanisms for the definition of content-based access control policies. More precisely, they propose an extended definition of permission (am, o, exp), where o is an object on which the access is regulated, am is an access mode valid for each element of o, exp is a logical expression evaluated when an access is required. The semantics of this privilege is: a user can make use of the object o in access mode am, if the expression exp is verified, where the expression exp refers to the content of the object. For example, the restricted privilege *(delete, PatientRecord, PatientRecord.State = "discharged")* represents the privilege to delete, in the PatientRecord table, every row that represents a discharged patient. However, the definition of context used by Giuri and Iglio is not enough general, since the context is only related to the content of a particular object.

Bertino et al. [AS08] point out how the access to critical data depends on environmental parameters, such as time and location. In particular, they developed a spatio-temporal role based access control model (GSTRBAC) and they propose a formal framework to compose complex spatial constraints. Their framework uses a predicative first order logic (Alloy [Jac02]) to model policies and discover possible conflicts. Their specification model, formally captures the policy constraints and assertions, and provides a conflict resolution mechanism using the Alloy constraint analyzer. However, Bertino et al. consider only time and location as context elements.

Another model, named Generalized Role Based Access Control (GR-BAC) and proposed by Covington et al. [MMMA00, CLS^{+}01], extends the traditional Role-Based Access Control (RBAC) model, considering object and environment roles. Their model incorporates these aspects in the traditional RBAC and defines three types of roles (Object, Environment and Subject roles). Object roles are based on any classifiable property of an object, such as the date of creation or the object type. An object is any resource in a system. For example, in a home resources may include different appliances, such as a dishwasher or stereo, media objects, such as movies, and sensitive digital information, such as medical records or income tax returns. Environmental roles consider the state of the environment before grant the permission at the user, as for example time or location. Finally, subject roles refer to the classical RBAC, which use the role of the user to select if he/she can interact with the resource.

Zhang et al. [ZP04] documents that GRBAC may not be feasible in practice, since the potential large amount of environmental roles make the system hard to maintain. Furthermore, they explain that by defining too many roles in the system, GRBAC loses the advantage that RBAC provides. Therefore, they propose Dynamic Role Based Access model (DRBAC), which includes seven different elements: users (the entities whose access has to be controlled) roles (the function that the specific user has in the organization), permission (the approval to access one or more RBAC protected resources), envs (the set of context information in the system), session (the set of interactions between subjects and objects), UA (the mapping that assigns a role to a user), PA (the mapping that assigns permissions to a role). In DRBAC state machines maintain the role subset for each user and the permission subset for each role. A state machine consists of state variables, which encode its state, and events, which transform its state. In DRBAC, there is a Role State Machine for each user, and a Permission State Machine for each role. The role and permission are used as state variables respectively. The Context Agent collects context information and generates predefined events to trigger transitions in the state machines. State machines are able to manage a potential large amount of environmental roles.

The previous works, consider context as an extra condition that must be satisfied to activate a given security rule or transaction. However, the specification of the security policies is completely parameterized by the organization, so it is not possible to handle simultaneously several security policies associated with different organizations. Kalam et al. [KBB$^+$03] suggest a model, named Organization based access control (OrBAC), able to specify the security policies at the organizational level, which is independently of the implementation of this policies. OrBAC defines a policy depending on: how this organization is employing subject (this is modeled through the concept Role), how this organization is using objects (this is modeled through the concept View), how this organization is performing actions (this is modeled through the concept Activity), how this organization is defining contexts that apply to users who are performing actions on objects (this is modeled through the relationship Define). Therefore, a permission has the form: Permission (org, r, v, a, c), where permission represents prohibition, obligation and recommendation, org is the organization, r is the role, v is the view, a is the activity, and c is the context. For example, the security policy of the Limerick hospital may include the facts Permission (Limerick Hospital, physician, medical-record, consulting, urgency). Where, Limerick Hospital is the organization, physician is the role of the user, medical-record is the view (object), consulting is the activity, and urgency is the context.

However, the Kalam et al. [KBB$^+$03] model includes the possibility to specify both permissions, prohibitions, obligations and dispensations. Therefore, some conflicts may occur. Cuppens and Cuppens-Boulahia [CCB08b, CCB08a] describe a strategy to detect conflicts in OrBAC policies. They suggest to combine the bottom-up approach of Datalog [Ulf82] with the top-down strategy as defined in the SLG algorithm [Tom97] to evaluate OrBAC policies. Thus because Datalog rules guarantee the decidability of query evaluation and its termination in polynomial time, but in some cases, it would not be possible to express most contextual conditions, such as temporal or spatial contexts. Combining the bottom up approach of Datalog with the top-down strategy of SLG algorithm, enables the evaluation of OrBAC policies.

The previously described works consider context in the management and design of access control policies. However, they do not consider why access control policies should be defined and enacted. For example, why a specific user has to interact with a specific resource, which assets the access control system has to protect, and which security requirements must guarantee, are neglected in the previous approaches.

2.1.4 Asset-centric access control

The restriction of access is a mechanism by which organizations protect their assets [CIN05] against potential harm. Identifying assets and their value are usually the initial steps of security requirements engineering [HLMN08]. Jaatun et al. [JT08] suggest an approach that can be used to identify and prioritize assets in any software engineering project.

Firesmith [Fir04] describes an asset-based risk-driven analysis approach to specify security requirements. In particular, this work considers assets, attacks, attackers, threats, security goals and, policies. What missing is a run-time methodology to dynamically adapt the system behavior, according to the context and the value of the assets.

Salehie et all. [MSLPIORA11] promote assets as a first-class entities in engineering software systems. They provide an approach able to consider the different aspects involved in a general security problem. They represent assets and their relationships by means of an asset model, use the KAOS [Lam09] goal model to represent functional and non-functional requirements, and a threat model to capture threats and attacks. From these models, a causal network is generated and used at run-time, to analyze the consequences of asset-relevant changes. This work describes a general approach able to analyze any security problem, but does not consider the context in which the application is working, and it is not specifically focus on the access control problems.

Since policies define restrictions of access to valuable information assets, and such access is required to carry out tasks, assets have to be considered as first element in the access control policies definition [CIN05, CCB08b, CCB08a, ZP04]. What is commonly not considered, and is missing in the literature, is an approach that consider the run-time value of the assets and their relationships to select the set of access control policies to apply.

2.2 Goal-oriented Requirements engineering

This section provides an overview on goal-oriented requirements engineering. Section 2.2.1 describes the main concepts related to goal oriented requirements engineering, while section 2.2.2 presents the main motivations of using goal model.

2.2.1 Main concepts

"Requirements engineering is the process of discovering, documenting and managing the requirements for a computer-based system. The goal of requirements engineering is to produce a set of system requirements which, as far as possible, is complete, consistent, relevant and reflects what the customer actually wants" [SS97].

Identifying the different actors and the users' requirements is the first step of software development [DR77]. An actor is an entity that has strategic goals and intentionality within the system or the organizational settings [Ali10]. An actor can represent different agents, roles, systems etc. and has the possibility to autonomously decide what and in which way reach his/her goals [ADG10a, FP09, BPG$^+$04].

Different definitions of goals are present in the literature. According to van Lamsweerde [vL01] a goal is an objective the system under consideration should achieve. Anton [Ant96] defines a goal as the hightest level object of the business, organization or system. Rolland et al. [RSA98] describe a goal as something that some stakeholder hopes to achieve in the future.

Users' goals are related to each other and decomposed in sub-goals by means of AND-Decomposition or OR-Decomposition [YLM04]. When a goal is AND-Decomposed it is necessary to reach all subgoals for its achievement. Conversely, when a goal is OR-Decomposed it is sufficient to reach one of its subgoals for its achievement. In other words, using OR-Decomposition, a set of alternatives to reach the refined goals are described.

By means of goals decomposition, requirements are defined and elicited [vL01], to describe both the functional and non-functional characteristic of the system to be. Functional requirements represent the operations that the system is expected to deliver [vL01], while non functional requirements represent qualities of the system such as performance, usability, security [SL07, LK95]. In particular, security requirements are a particular sub-class of non-functional requirements, concerned with the protection of the valuable assets [CIN05].

As context changes, different set of requirements can be activated or deactivated. To reach these goals different tasks can be executed from the system. Schilit and Theimer [ST94] describe the context-aware computing as a software able to adapt its behavior according to the location of use, the actions of nearby people and objects, as well as changes to those objects over time. Dey [Dey01] defines the context as any information that is considered relevant to characterize the situation of an entity as a person, a place, or an object, or the interaction between a user and the application, including the users and the application themselves. Schilit et al. [SAW94a] consider the context in terms of attributes of the physical environment, and in particular the physical location of users. Ali [ADG10b, ADG10a, ADG09, Ali10] shows the advantages of using context in the goal modeling oriented techniques as instrument to determine the set of requirements relevant to the system.

Goal oriented techniques rationalizing changes in access control systems, provide awareness if any requirement is going to be denied, and also aid decision support at runtime.

2.2.2 Goal Model: Main motivations

The protection of assets from harm is, first of all, a problem of deciding who can interact with the assets that need to be protected. On the one hand, there is the users' necessity to access to the different resources, on the other hand, the system has to guarantee security, giving at each actor the least possible privileges (least privilege principle) [CIN05]. Security, functional and other non-functional requirements provide rational for revoking or granting access permission to assets in the system boundary. Any change in the access control policies should be justified by reasons from requirements to answer WHY the change is essential. Therefore, a requirements model can support rationalizing changes in access control systems, provide awareness if any requirement is going to be denied.

The main advantages of considering requirements, using a goal-oriented approach [vL01, YM98] are presented in following.

- *Acquiring requirements.* The main advantage of goal-oriented requirements engineering is the ability to capture "Why" a system is necessary. More precisely, goal modeling provides a way to describe who, how and why will interact with the system.

- *Explaining requirements to the stakeholders.* To detect the system requirements, the cooperation between the software engineers and the final stakeholders is necessary. Goal modeling techniques provide instruments understandable by a large audience, from software engineers, to the final customers.

- *Refining requirements.* Goal modeling allows the refinement of high level goals and the clarification of the potential ambiguous requirements. The refinement explains the requirements to stakeholders by answering "why do we need to do this" and "how do we reach this".

- *Managing conflicts.* Different users may have different goals and priorities. Goal-oriented techniques provide a natural, high level way to detect and manage conflicts between different requirements.

- *Avoiding irrelevant requirements.* Goal modeling allows one to distinguish relevant aspects of software from the irrelevant ones. A simple model understandable by both the stakeholders and the software engineers, is the way to understand which requirements the system has to satisfy and which ones are not relevant for the application.

- *Separating stable from volatile aspects.* Goals and users requirements evolve during time. Goal modeling techniques help to detect which parts of the software system are stable and which ones evolve during time.

- *Analyzing context.* Humans behavior depends on context. According to the environmental conditions, users can have different goals and can choose different ways to satisfy them. Identifying the set of goals important in each context is necessary to design a flexible software able to adapt his behavior in different environmental conditions.

- *Reducing the gap between physical world and software world.* Understanding the users requirements in the different environmental conditions provides a way to understand how the software system impacts on the users' behavior and activities. In particular, it is possible to analyze the relation between the physical world, in which the application is working, and the application itself.

2.3 Towards a Solution: asset-centric, context-aware adaptive security

This section provided a brief overview on the literature related to the main aspects involved in this work. It described several works that consider requirements in the access control policies definition [FLLD01, MZ08] but, in all of them, the role of the assets and the environment on which the application is working is neglected. On the other hand, it presented other works that consider context [CCB08b, CCB08a, ZP04, MMMA00], but ignore the role of the system requirements and the value of the assets that need to be protected. Finally, it described an approach proposed by Salehie et al. [MSLPIORA11] by which promote assets as a first-class entities in engineering secure software systems. This approach provides a model able to capture the different aspects involved in a security problem, and a way to use this model in the run-time adaptation. However, this work does not focus on the access control case and the role of the context in which the application is working is not considered.

This thesis aims to adapt the Salehie et al. [MSLPIORA11] approach, to realize an adaptive access control system, able to consider the assets, and the system requirements in the run-time adaptation.

Chapter 3

Requirements-driven Adaptive Security

"Security is about trade-off, not absolutes."

Ravi Sandhu

This chapter describes a requirements-driven approach that will be used in the rest of the thesis. The approach was presented by Salehie et. al. [MSLPIORA11] and conceives assets as first-class entities during requirements modeling and runtime adaptation. This approach is composed of by three main phases. During the first phase, the system requirements and the security concerns are designed. In the second phase, STrioM is used as input to build a fuzzy causal network (FCN), which is used at runtime to analyze system security in different situations, and enable, when necessary, a set of security controls to mitigate security threats. SECTION *3.1 provides an overview of the mentioned approach.* SECTION *3.2 presents STrioM the extended requirements model used to describe a security problem. Finally,* SECTION *3.3 describes the fuzzy causal network (FCN) and how is used at runtime.*

3.1 Overall approach

The approach presented by Salehie et. al. [MSLPIORA11] conceives assets as first-class entities during requirements modeling and runtime adaptation. To this end, the requirements model is extended to consider the main security concerns including threats, attacks and vulnerabilities. The extended model (named STrioM) is translated into a fuzzy causal network (FCN). Finally, the FCN is used at runtime to analyze the security risks and evaluate the set of security controls necessary to protect the system. The three main phases of the approach are reported in Fig 3.1.

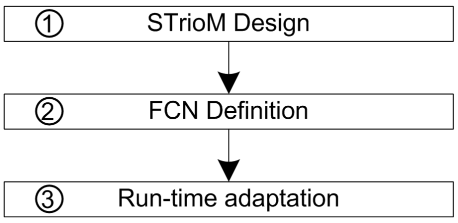


Figure 3.1: The three phases to model, analyze, and apply adaptive security at runtime

- Security Trio Model (STrioM) Design
  In the first phase STrioM is used to identify requirements. Functional and non-functional requirements are described using the KAOS [Lam09] goal model, while the behaviors of potential attackers is captured using threat (anti-model)model [vL04b]. Finally, asset are specified in terms of KAOS entities by means of an asset model.

- Fuzzy Causal Network (FCN) Definition
  In the second phase STrioM is used as input to build the Fuzzy Causal Network (FCN). The elements and relationships identified in the asset, goal, and threat models are translated into nodes and edges of the FCN. Nodes are used to represent security concerns, such as requirements, threats and attacks. Links identify positive and negative causal relationships between security concerns. Each node and link is associated with an appropriate value. The value of a node reflects the degree to which the "concept" is active in the system at a particular time. A directed edge E_{ij} from concept C_i to concept C_j measures how much C_i causes C_j .

- Run time adaptation
  The FCN previously described is used to apply adaptive security at runtime. The system uses appropriate sensors to monitor the values of the assets that need to be protected, and tunes the fuzzy causal network accordingly. The analysis mechanism calculate the utility value of any possible security controls configuration and the best one is selected. The outcome of the previous analysis is used to (re-)configure the running system.

3.2 Security Trio Model (STrioM)

This section presents STrioM a model that represents the different security concerns (assets, threats, attacks, vulnerabilities, security goals/requirements/controls) together with the functional and non-functional requirements of the system. STrioM is composed of three different models: asset, goal and threat model (Fig. 3.2).

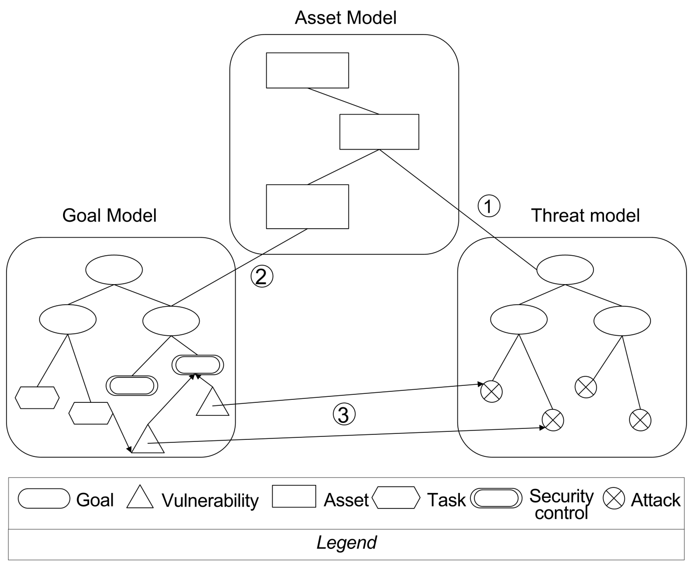


Figure 3.2: STrioM: Security Trio Model

- The Asset Model represents assets and their relationships. An asset is related with another on that contains or use it. For example, in a mobile smart phone asset model, the asset mobile phone is related with the SIM card. Relationship between two assets represents the fact that the value of one asset affects the value of the other one. In the previous example, the higher the SIM credit, the higher the mobile smart phone value.

- The Threat model is used to describe the behavior of threat agents, also called attackers. Attackers (e.g., Random hackers, malicious applications or malicious employees) can exploit vulnerabilities, to compromise the system and reach their goals. Threat goals are the motivations of threat agents and are iteratively decomposed into subgoals. Threat goals are finally satisfied by means of attacks (e.g., phishing, malware, root exploit), which exploit system vulnerabilities.

- The Goal Model is used during the early requirements analysis to explain the objectives of a software system [ADG10a]. The goals of the system are gradually decomposed until functional and non functional requirements are identified. Requirements can be finally reached by means of executable operations (tasks). Security goals/requirements are a particular kind of goals/requirements related to the protection of the system. The set of task used to meet security requirements are called security controls. Security controls, such as "PIN" or "Fingerprint", are used to mitigate system vulnerabilities. Vulnerabilities are system weakness which can be exploited by attackers and facilitate their success [MSLPIORA11]. On the one hand, vulnerabilities may result after the execution of particular tasks, such as the installation of new software, input validation errors, execution of malicious programs like viruses, or malware.. The goal model explicitly represents security controls and vulnerabilities in terms of KAOS entities.

Asset, threat and goal models are connected by means of the following links.

- Link 1: assets values influence threat goals. Adding an asset or increasing the value of an existing asset can motivate threat agents.
  For example, in a mobile smart phone, a potential attacker may be interested in "stealing credit". The threat goal "steal credit" is directly dependent with the amount of money present on the SIM card.
- Link 2*:* assets values impact on security goals. When an asset is added or removed from the system, certain security goals may be changed accordingly.
  For example, in a mobile smart phone, the goal "confidentiality" depends on the value of the information stored on the phone, such as "credit card info", "email contacts" and "user location data".
- Link 3: attacks exploit vulnerabilities. The success of an attack may depend on the presence of different vulnerabilities V1, V2,... Vn. The higher is the probability of these vulnerability, the higher is the probability of attack to succeed.
  For example, the vulnerability "no encription" and "no authentication" is necessary to "access data on stolen phone".

3.2.1 Asset Model

The asset model represents assets and their relationships. The example used to illustrate this model was described by Mazeiar et. all. [MSLPIORA11] and is illustrated in Fig. 3.3. The example describes the set of assets related to a mobile phone. Asset model is made by:

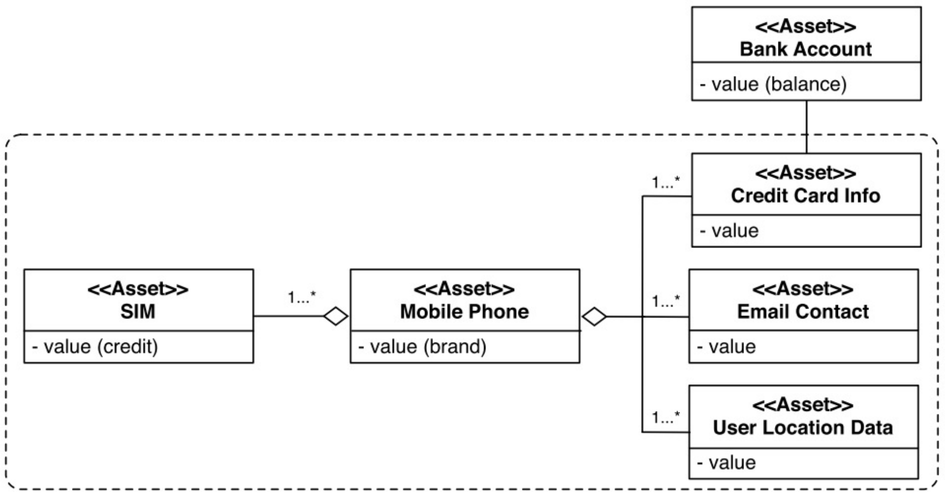


Figure 3.3: Asset Model, an Example

- Asset: is anything that has a value for the organization [CPP02]. Assets can represent both physical resources, such as computers, screens, devices, and logical, such as files.
  In the mobile phone example assets are: "mobile phone", "SIM", "credit card info", "bank account", "email contact" and "user location data".

- Asset value: represents the asset criticality for the company or a person. This value may be related to the economic value of the resource, such as its price, or to other factors, such as the importance for the enterprise.
  In the example, the value of the SIM is inferred from the credit value, while an approximative value is selected for the user location information.

The link following described is used to connect assets together:

- Asset to asset: is normally represented as a line, with a diamond on the end, and on the line the cardinality is reported.
  In the example, Mobile Phone has one or more SIM and SIM is a part of one Mobile Phone.

3.2.2 Goal model

The goal model represents the main objectives the system must achieve/-maintain and decomposes them into functional and non functional requirements. The example used to illustrate this model was described by Mazeiar et. all. [MSLPIORA11], and is illustrated in Fig. 3.4. This model is made by:

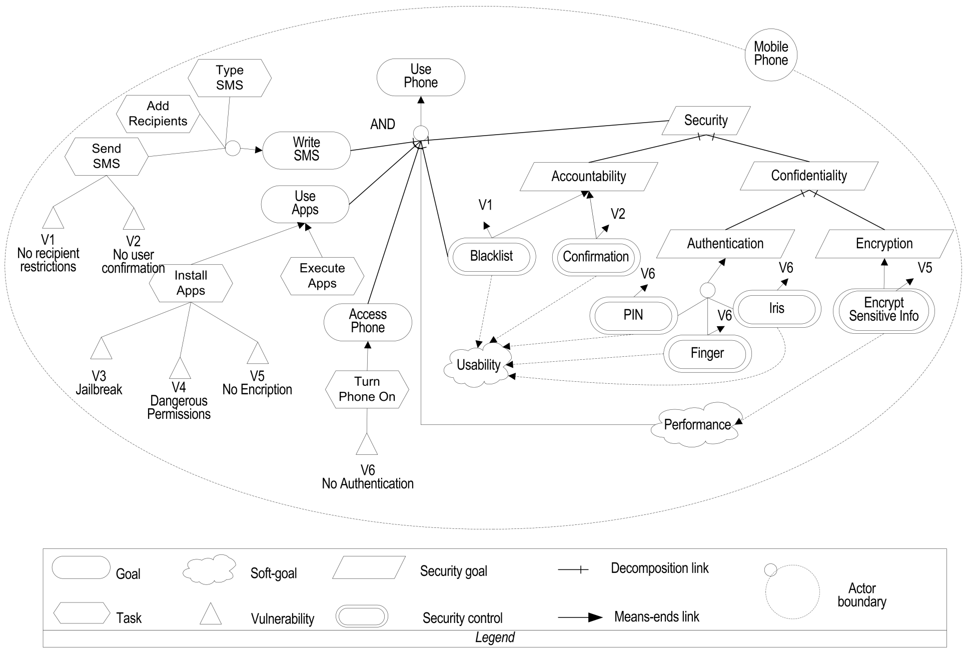


Figure 3.4: Goal Model, an example

- Actor: is an entity that has strategic goals and intentionality within the system or the organizational settings [Ali10]. He is represented by a circumference with the name of the actor inside.
  In the example, the owner of the mobile phone is the only actor present in the system.

- Goal: is something that some stakeholder hopes to achieve in the future [RSA98]. In other words, goals represent actor's strategic interest that have clear-cut definition and clear-cut criteria to judge if they are satisfied [Ali10].
  "Write SMS" and "use apps" are two different goals identified in the mobile phone example.

- Non-functional goal: are a particular subset of goals which describe a property or characteristic that a software system must exhibit or a constraint that it must respect, other than an observable system behavior. In particular, non-functional goals represents actor's strategic interest that has no clear-cut definition and/or no clear-cut criteria to judge if it is satisfied [Ali10].
  In the example, the non-functional goals identified are "usabiliy" and "performance".

- Security goal: are a particular kind of goals concerned with the protection of the system from attacks.
  In the example, "Accountability" and "Confidentiality" are two security goals identified.

- Task: is an executable process that represents a way of doing something [Ali10].
  "Install Apps" and "Execute Apps" are two different tasks identified in the example.

- Security control: is a protection mechanism employed to secure the system [EYZ10]. More precisely, security controls are a particular type of task used to mitigate vulnerabilities, and satisfy the security requirements.
  "PIN", "Fingerprint", "Iris" are possible security controls used to mitigate the vulnerability "no authentication".

- Vulnerability: is a weakness or a back-door in the IT system which allows an attacker to compromise its correct behavior [EYZ10].
  An example of vulnerability presented in the mobile phone case is "V1: No recipient restrictions".

The previous security concerns are connected by means of a set of links:

- Means-end: goals (including security goals) are finally decomposed into functional and non-functional requirements and are satisfied by means of tasks (security controls for security goals). Means-end links are used to connect tasks and security controls with their corresponding requirements.
  In the example, the tasks "Install Apps" and "Execute Apps" allow users to reach the goal "Use Apps".

- Decomposition: goals and tasks are linked together by means of AND or OR-decomposition. Using these links a particular goal is refined into sub-goals. For example a goal G can be AND-decomposed or OR-decomposed into subgoals G1, G2, ... GN. A goal that is AND-decomposed is satisfied only if all its subgoals are satisfied. A goal that is OR-decomposed is satisfied only if at least one of its subgoals is satisfied. The same principle is applicable for the task decomposition.
  In the example, the goal "Security" is decomposed using OR-decomposition in "Accountability" and "Confidentiality".

- Contribution to non-functional goal: represents the relation between certain tasks/security controls and non-functional goals. This link is used when the specified tasks/security controls positively/negatively contribute to the corresponding non-functional goal.
  In the example, the security control "encrypt sensitive info" contribute negatively on the soft-goal "performance".

- Security control to vulnerability: when a security control is connected with a vulnerability, if the security control is active, the vulnerability is mitigated or removed from the system.
  In the example, the use of "PIN" is a security control used to mitigate the vulnerability "No authentication".

- task to vulnerability: tasks may activate vulnerabilities. This means that when a task or a security control is executed certain vulnerabilities may be brought into the system.
  For example, the task "install apps" may activate the vulnerabilities "no encryption", "dangerous permissions" or "jailbreak".

Furthermore, several links are used to relate the goal, asset and threat model together.

- Asset to Security Goals: represents the relation between the assets that need to be protected and the security goals that aim to protect it.
  In the example the value of the "Credit Card Info", "Email Contact" and "User Location Data" impact on the Security Goal "Confidentiality".

- attack to vulnerability: the success of an attack may depend on the presence of different vulnerabilities V1, V2,... Vn. The higher is

the probability of these vulnerability, the higher is the prob- ability of attack to succeed.
For example, the vulnerability "no encription" and "no authentication" is necessary to "access data on stolen phone".

3.2.3 Threat Model

Security is based on finding a tradeoff between different security goals and other functional and non-functional goals. Analyzing threats and attacks allows the system engineer to detect who, how and why can exploit vulnerabilities. The threat model provides a graphical way to describe these elements. More precisely, threats can be viewed as malicious actors' goals [GE07b], can be reached by means of attacks (tasks) and violate security goals, such as confidentiality, integrity availability and accountability. The example used to illustrate this model was described by Mazeiar et. all. [MSLPIORA11], and is illustrated in Fig. 3.5. This model is composed of:

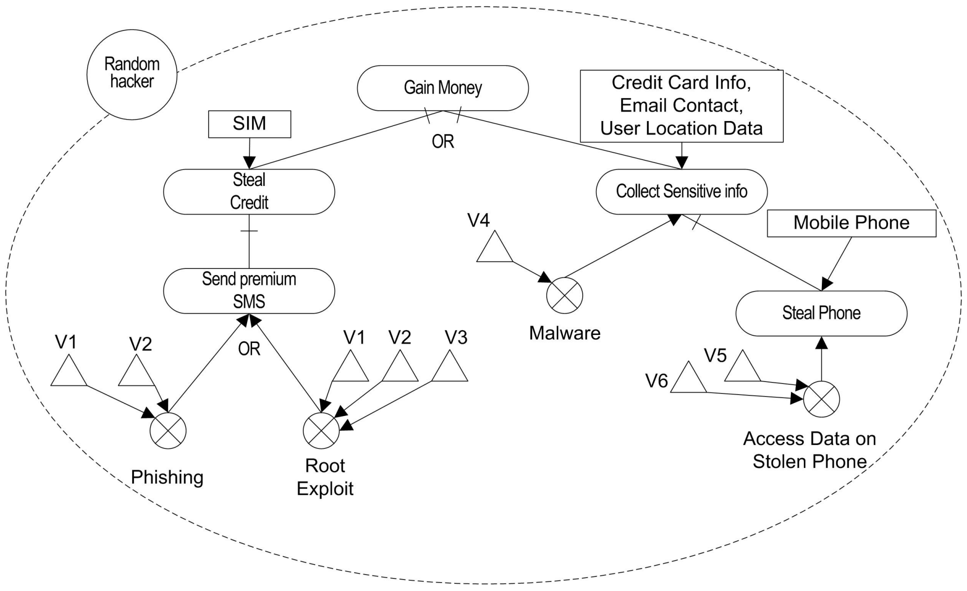


Figure 3.5: Threat Model, an example.

- Threat agent: is an entity with malicious intentions, such as random hacker, malicious application or malicious employee. Threat agents are represented by a circumference with the name of the actor inside.
In the example, "Random hacker" is the threat agent identified.

- Threat goal: is a motivation of a threat agents.
  In the example, "Gain money" and "Steal credit" are two different threat goals identified".

- Threat or Attack: is a potential way in which an attacker can violate the security of a component of the system [EYZ10]. By means of attacks a particular threat agent tries to exploit the vulnerabilities of the system to meet his goals.
  In the example, "Phishing" or "Root exploit" are two possible attacks used to reach the threat goal "Send premium SMS"

The previous entities are connected using a set of links:

- Means-end: connects threat goals and attacks. The satisfaction of the threat goals connected to a particular attack is related to the success of the attack itself.
  For example, the satisfaction of the threat goal "Send premium SMS" is related to the success of the attacks "Phishing" or "Root Exploit".

- Decomposition: threat goals are connected together using AND or OR-decomposition. AND or OR-decomposition is used to refine a particular threat-goal into sub-threat-goals. A threat goal that is AND-decomposed is satisfied only if all its subgoals are satisfied. A threat goal that is OR-decomposed is satisfied only if at least one of its subgoals is satisfied. These links can be also used to decompose attacks in sub-attacks.
  In the example, "gain money" is OR-decompose in "collect sensitive info" and "Steal credit"

Several links connect the threat model and the goal model.

- Vulnerability-attack: are the links between vulnerabilities and attacks. If a vulnerability is connected to a particular attack, the presence of the vulnerability is a necessary requirement for the success of the attack.
  In the example, the success of attack "Phishing" is related with the presence of the vulnerabilities "V1:No recipient restrictions" and "V2:No user confirmation"

- Threat-asset: this link is used to connect an assets to a threat. An increment in the value of an asset is reflected in the value of all threats with which it is connected.

For example, "steal credit" is in relation with the amount of money present on the "SIM". The higher is the credit on the "SIM" card, the higher is the value of the threat goal "steal credit".

3.3 Fuzzy Causal Network

The interest in fuzzy causal networks has increased in the recent years for their ability to support both the qualitative (e.g., Fuzzy Causal Maps [Kos86]) and quantitative (e.g., Bayesian decision networks [doi09]) analysis and decision making. FCN can use observable quantities, latent variables, unknown parameters or hypotheses in the decision process. A fuzzy causal network is a dynamic system whose topological structure is a directed graph [ZLZ06] composed of nodes and edges.

- Node represents a concept whose state varies with time. In particular, a state is represented by a real number that specifies the value of the node at the specified time. The aggregation function associated to each node specifies how nodes reacts when their inputs change.
- Link is used to connect nodes and represents a causal relation from the tail to the head of the edge. The weight associated to a particular link indicates how strong is the relation between the nodes. Nodes which are not connected represent variables which are conditionally independent of each other.

If the value of a specific node changes at the time t, all the network is updated and the process is iterated until a final equilibrium state is reached [Kos88]. Each node takes as input a particular set of values from the nodes that are associated with it and aggregates them with its current value according to a specific function (e.g., sum, average, max). Several motivations make the FCN useful in the STrioM run-time analysis:

- FCN provides quantitative and qualitative analysis. Since some informations are imprecise or incomplete, quantitative analysis may not be always feasible. Some data, such as risk or threat are intrinsically uncertain, others, such as assets are subject to imprecise evaluations.
- Zhou et. al. [ZLZ06] demonstrate that FCN under certain conditions converges to its limit cycle or static state in O(n) steps, where n is the number of nodes of the FCN. This makes the fuzzy causal reasoning adoptable in the run-time evaluation of security controls, when the value of the assets to protect and the other security concerns changes.

3.3.1 Building the fuzzy causal network

The FCN is built starting from STrioM. The different nodes of the network represent different security concerns (i.e., assets, security goals, vulnerabilities, threats, etc.,) of the FCN. Each node has a specific meaning, specified as a fuzzy value (into the range [0,1]) except the utility. The semantics of each node of the network is described in Table 3.1.

Node	Meaning
Asset	Value
Threat Goals	Threat level
Goals	Satisfaction level
Attacks	Probability of success
Security controls	Strength
Vulnerability	Probability of presence
Utility	Value

Table 3.1: Semantics of the nodes of the FCN generated from STrioM model

The nodes associated with assets, goals (including functional, non-functional, and security goals), and threats represent the value of assets, the satisfaction of goals and the level of threat, respectively. Attacks are associated with their probability of success, while security controls are associated with their effectiveness (strength). Vulnerabilities are associated with their to be brought by the system. The causal network also includes three other nodes: partial risks, risk and utility. These nodes are not present in STrioM and are used to assess risk and the utility of any possible configuration of security controls. The partial risk depends on the value of the targeted assets and the probability of the attack. The risk node aggregates contribution of all partial risk nodes. Finally, the utility node express the effectiveness of the security control configuration selected, depending on its benefits and costs. Costs indicate how much security controls hurt other functional and non-functional goals, while benefits depend on how much they mitigate the risk. The FCN uses three type of nodes to describe the different security concerns:

- CHANCE NODES represent uncertain domain entities significant for causal reasoning (oval in the diagram). Except the security controls, security concerns (derived from STrioM) are chance nodes in the FCN. Partial risk and risk are also chance nodes.

- DECISION NODES indicate decisions to be made (rectangle in the diagram). In this case each security controls are translated into a set of decision nodes in the causal network.

- UTILITY NODES correspond to the fitness value of the network configuration (hexagon in the diagram).

Links are used to connect the different nodes present in the FCN. Each causal link of the FCN is labeled with a signed weight, which represents the strength of causal relationship between two nodes of the network.

- POSITIVE CAUSAL LINKS are used to connect two entities A and B when an increase of the value of A causes an increase of the value of B (A $\xrightarrow{+}$ B).

- NEGATIVE CAUSAL LINKS are used to connect two entities A and B when an decrease of the value of A causes an decrease of the value of B (A $\xrightarrow{-}$ B).

A high/low weight for a positive or negative causal link means that an increase of A may cause a great/small increase or decrease of B. In this thesis weights are specified using labels in the [0,1] interval.

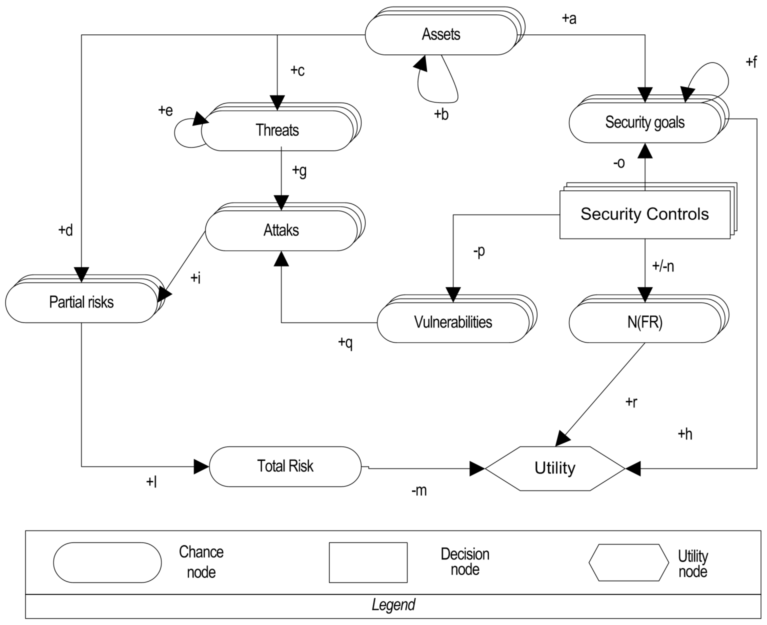


Figure 3.6: The abstract model of the fuzzy causal network

The semantics of the links of the FCN (see Figure 3.6) is the following:

- *+a*: security goals are directly influenced by the value of assets present in the system. The higher are the value of the assets to protect, the higher is the value of the associated security goals.
  For example, the "Confidentiality" goal is affected by the "User location and data". More accurate are the information on the "User location and data" stored in the phone, more important is for the system guarantee their confidentiality.

- *+b*: the value of an asset also depends on its related assets (positive causality). When the value of an asset changes, this value is propagated onto its related assets.
  For example, the value of the "Mobile phone" increases with the credit present on the "SIM".

- *+c*: the value of assets has a positive impact on threats. Assets and threats connected are directly dependent. When the value of an asset increases, the value of the corresponding threat increases as a consequence, since adding an asset or increasing the value of an existing asset can motivate threat agents.
  For example, the threat goal "Steal credit" depends on the money present in the "SIM".

- *+d*: the value of each partial risk depends on the value of the targeted asset. Since partial risk is the probability of an attack multiplied by the possible resulting loss, as the value of assets increase the risk of an attack consequently increases.

- *+e*: threats can influence other threats. If a threat goal A is connected to a thread goal B, the higher is the value of A, the higher is the threat level of B.
  For example, if the value of the threat goal steal credit increases, the value of the connected threat goal gain money increases as a consequence.

- *+f*: a security goal can influence another security goal. The satisfaction level of a security goal is estimated by considering the satisfaction level of security goals connected.
  For example, the higher the value of the node "accountability", the higher is the valued of the node "security".

- *+g*: threats goals can influence attacks. Threat nodes propagate their values on the nodes representing their sub-threats and, finally, to the

nodes representing their underlying attacks.
For example, "Steal Mobile Phone" propagates his value to "Access Data On Stolen Phone".

- *+h*: security goals positively impact on the utility of the system. The utility is estimated by considering the satisfaction of security goals as the benefit of a specific configuration of security controls.

- *+i*: partial risks depend on the value of an attack. The higher is the probability of an attack, the higher is the relative partial risk.
For example, the partial risk of Malware depends on the relative attack probability of the malware attack.

- *+l*: the value of the risk is calculated by summing the value of partial risks.
For example, the total risk of the system depend on the value of the risks of malware, phishing, root exploit and access data on the stolen phone.

- *-m*: the risk has a negative impact the utility. When the risk increases the utility of the current configuration of security controls decreases.

- *-n*: the security controls can have a negative impact on non-functional requirements.
For example, "Encrypt Sensitive Info" has a negative impact on "Performance".

- *+o*: security controls influence the satisfaction level of security goals. The more severe is the set of selected security control, the higher the satisfaction level of the corresponding security goals will be.
For example, "confidentiality" goal is affected by the value of the security control "PIN", "Iris", "Finger", "Encrypt Sensitive Info". If all of them are activated, the satisfaction level of "confidentiality" is higher than when only "PIN" is enabled.

- *-p*: vulnerabilities are mitigated by enabled security controls. If the system selects more restrictive security controls the value of the system vulnerabilities decreases.
For example, security control "Encrypt Sensitive Info" is used to mitigate the vulnerability "No encryption".

- *+q*: the presence of a set of vulnerabilities influences the probability of an attack. Since attacks exploit vulnerabilities, the higher is the

value of the vulnerabilities, the higher is the probability of success of an attack.
For example, the success of "Access Data On Stolen Phone" depends on the presence of vulnerability "No encryption". The higher the probability of this vulnerability, the higher the probability of "Access Data On Stolen Phone" to succeed.

- *+r*: the utility of the system depends on the satisfaction of functional and non-functional requirements. The less penalized the functional and non-functional requirements are by the application of a configuration of security controls, the greater is the utility of that configuration.

3.3.2 Setting the fuzzy causal network

The analysis and decision-making are performed using the reasoning mechanism of the FCN. Before using the causal network two different steps have to be performed: configuring the aggregation functions and initialization and tuning.

- Configuring the aggregation functions: the reasoning mechanism of the FCN computes the value of each node by aggregationg the contributions of its incoming links according to a specific aggregation function, such as *Minimum, Maximum, Average and Sum.* Indeed, the system designer has to select for each node of the FCN the relative aggregation function. When different security concerns are connected to the same node (e.g., assets and security controls are connected to the security goals) the reasoning mechanism first aggregates the contributions coming from the same link types, and then combines them together. In this case the system designer has to select several different aggregation functions, one for each type of input and one to merge the results together. It is underly that the aggregating functions are used after applying weighting to each input. This means that the incoming contributions of each node can be tuned by adjusting the weights of the corresponding links.

- Initialization and tuning: once the aggregation functions have been selected the values of the nodes and weights of causal links should be initialized. The value of the different nodes of the FCN represent the initial condition of the network. Instead, the weights of the FCN links represents the strength of causal relationship between the different security concerns. Nodes and weights values should be tuned based on existing qualitative and quantitative evidence. The initial values can be provided by stakeholders, domain experts, and existing evidence (e.g., statistical data).

The FCN weights and aggregation functions selected respectively for each link and node of the FCN are used to perform a reasoning mechanisms, called sensitivity analysis. By generating different combinations of assets values, the impact on the different security concerns is analyzed, and, in particular, the utility of any configuration of security controls is evaluated. Two main activities are executed during the sensitivity analysis. First, the weights of the fuzzy causal network are tuned based on the mapping between assets and utility values. Second, the effects of the different aggregation functions on the FCN reasoning are investigated. More precisely, system designers analyze how the different aggregation functions impact on the FCN reasoning and on the final behavior of the system. On the one hand, if risk is calculated by using the maximum function more effective security controls are selected and may have an adverse impact on other functional requirements. On the other hand, the average function may lead to the selection of security controls that are less effective, but slightly penalize other functional requirements. If security requirements are deemed more important, the Maximum function is selected, otherwise the Average function is chosen.

3.3.3 Using the FCN at run-time to support adaptive security

At runtime, when the value of an assets changes, the FCN should be re-evaluated. The new value of the assets is propagated through the network. Then, all the configurations of security controls are evaluated. The FCN propagates the causal effects of the different configurations through links towards the utility node. After a few number of iterations the node values do not change anymore, and the network shows the updated values of risk and utility. The configuration with the best utility is selected. The flow chart of Fig 3.7 shows the adaptation process that is executed every time the value of an asset change.

- STEP 1 the FCN is derived from security model, the first configuration of security controls is selected Scc(0), while the Bscc (best security control configuration) and the best utility value are setted respectively to null and $-\infty$.
- STEP 2 the FCN reasoning is executed and the utility (u(k)) of the current security control configuration Scc(k) is calculated.

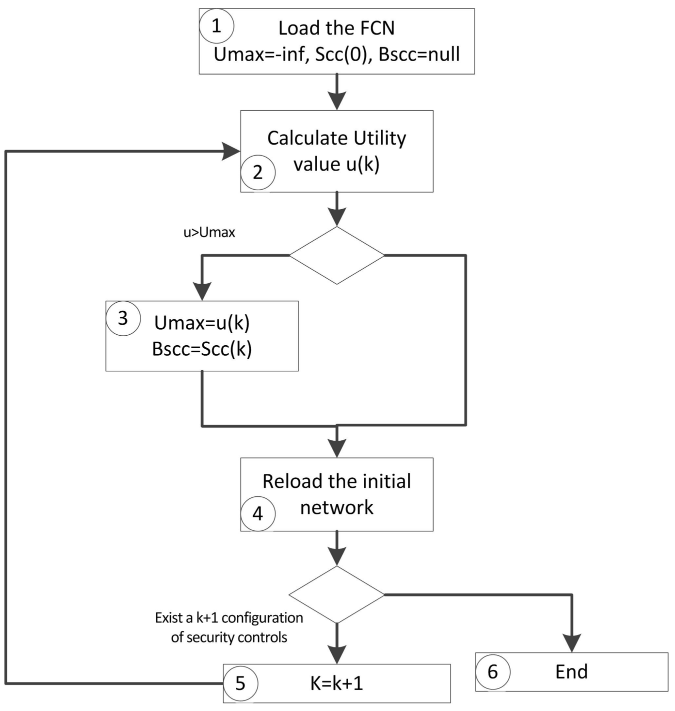


Figure 3.7: FCN adaptation process flow chart

- STEP 3 is executed when the utility calculated at the step 2 is grater than the best utility founded until the current configuration. The utility and the best security control configuration are updated (Umax=u(k), Bscc=Scc(k)).
- STEP 4 the initial condition of the FCN is restored, since the value of the FCN nodes were updated at the step 2.
- STEP 5 loads the next (k+1) configuration of security controls.
- STEP 6 when all configurations of security controls are analyzed the security control with the best utility is stored in the Bscc variable.

Chapter 4

Requirements-driven adaptive access control

"What we know is not much. What we do not know is immense."

Pierre-Simon Laplace

The restriction of access is a mechanism by which organizations protect their assets [CIN05] from potential harm. Traditionally, these systems are assumed to be relatively static. This assumption is no longer valid, as computing technology is becoming more tightly integrated into everyday life. The proliferation of smart gadgets, mobile devices, and sensors enabled the construction of pervasive computing environments, transforming regular spaces into intelligent spaces [ZP04]. Such intelligent spaces require new access control systems able to reconfigure their security controls in some automatic or semi-automatic way in response to environmental changes, named context. As security concerns can be affected by environmental changes, especially in a dynamic scenarios like access control, context must be explicitly represented/modeled and must drive the re-configuration of security controls. However, the approach proposed by Salehie et. al. [MSLPIORA11] does not explicitly consider context in the representation of security concerns and does not allow to re-configure security controls when the context changes. For this reason, this section extends the approach proposed by Salehie et. al. [MSLPIORA11] with an explicit representation of context, as described in SECTION *4.1. This section also applies the extended approach on a dynamic access control case of study, as described in* SECTION *4.2.*

4.1 Adding context to Requirements-driven Adaptive Security

Security systems usually ignores the context in which the system operates. However, as computing technology becomes more integrated into everyday life, it is imperative that security systems become more flexible. When the environmental conditions (context) in which the application is working change, different requirements and security concerns may be activated/deactivated. The effectiveness of the security controls used to protect the system changes as a consequence. Therefore, security system success is related on its ability to reconfigure its security controls in some automatic or semi-automatic way, in response to environmental changes.

Salehie et. al. [MSLPIORA11] present an approach able to guide system designers in developing adaptive security systems, from the requirements analysis to their enactment at run-time. The approach promotes assets as first-class entities in engineering secure software systems, and is based on three different steps. During the first step, a model (STrioM) is designed to capture the interplay between the different concerns involved in a security problem. In particular, an asset model is related to the requirements of the system, expressed through a goal model, and the objectives of an attacker, expressed through a threat model. During the second step, the model is translated into a fuzzy causal network (FCN) that is used at run-time to reconfigure the security controls that are applied on the system based on the assets to be protected.

Context may also influence the selection of proper security controls. When the context changes, different requirements and security concerns (i.e., assets, vulnerabilities, tasks, threats, etc.) can be activated/deactivated. For example, in an emergency security system, people may be informed about the nearest emergency door by means of voice instructions. However, goal "Give instructions by voice" is only enabled in a specific context, for example, when the "place is not noisy". On the other hand, for meeting its requirements, the system may perform a set of functionalities that cause changes in the context. For these reasons, an adaptive security system cannot be separated from its execution context. To add context to the approach proposed by Salehie et. al. it is necessary to:

- *Extend STrioM* with a new element representing a contextual factor. Suitable links must also be defined to connect each contextual factor to the security concerns that may be affected by its changes.
- *Analyze the impact of context on the definition of the FCN.* Since

STrioM is enriched with a notion of context and is used to derive the nodes and links of the FCN, it is also necessary to analyze how the context can affect the design of the FCN. In particular, the FCN needs to be restructured to include the nodes and links corresponding to both the contextual factors and their connections to the other security concerns that need to be added in the STrioM model. The set of fuzzy values that the FCN nodes representing the context can assume must also be identified, together with their activation functions.

- *Analyze the impact of context on the adaptation of the system at run-time.* In the approach proposed by Salehie et al. the FCN is only re-evaluated when assets change in order to assess the utility of any configuration of security controls. In this way, the configuration of security controls with the best utility can also be selected. However, when context is taken into account, its changes can also affect other security concerns and indeed cause modifications in the values of the nodes of the FCN. For this reason, the FCN should be re-evaluated even when context changes. In particular, it is necessary to understand how context modifications can be propagated into the FCN and when/how the FCN should be re-evaluated.

The following sections describe, respectively, each one of these three aspects, to extend the Salehie et al. approach with context.

4.1.1 Contextual STrioM

Context has been defined in the literature in different ways, since specific definition of context strongly depends on the domain it is used in. In this thesis context is considered as any information that can be used to characterize the specific environmental situation that may affect security concerns. A *security concern* represents any concept that is deemed relevant in securing the system. Context can be a factor in deciding what goals/requirements the system has to meet, which threat goals an attacker wants to reach. It can also enable system vulnerabilities, tasks or security controls. On the other hand, the system itself may cause changes in context. An early analysis of the mutual influence between context and the other security concerns may improve the quality of the software and facilitate the development of adaptive security systems. Despite its central role, context is ignored in STrioM. For this reason, this thesis extends the STrioM model with a representation of the context, as follows:

- Context: is any information that can be used to characterize the situation of a security concern, and is represented using blue oval-shaped element (Fig 4.1).

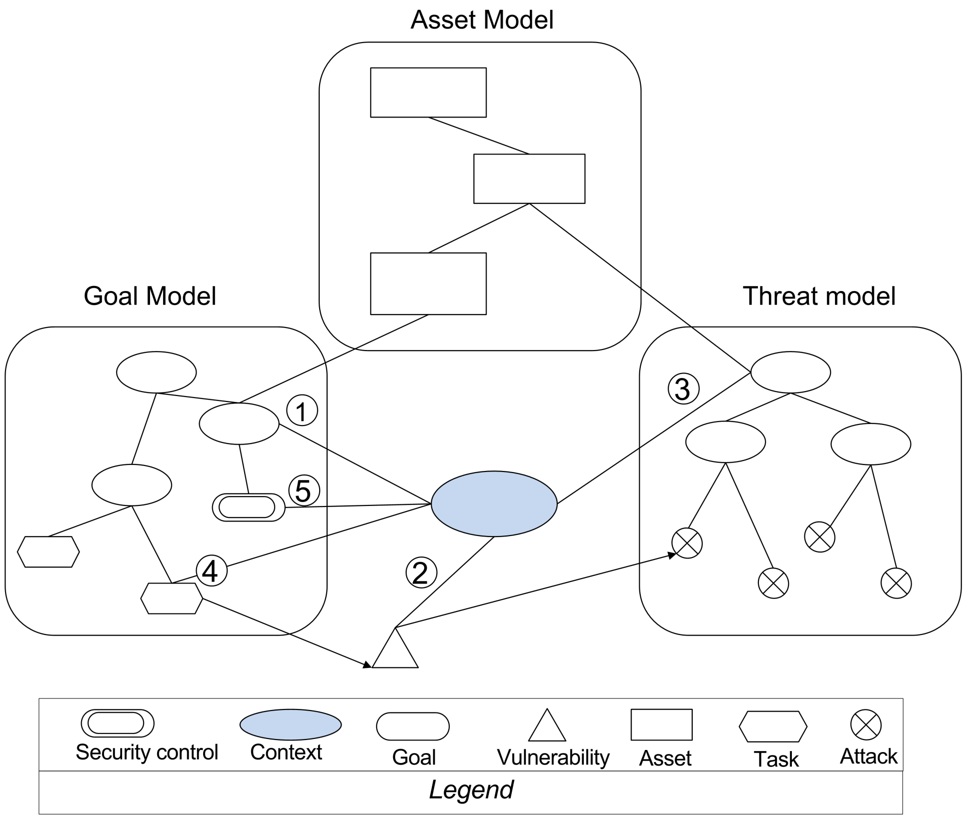


Figure 4.1: Contextual STrioM

A set of links is used to connect the context to the other security concerns:

- Context-Goals. The context has been widely defined in requirements engineering as the "partial state of the world that is relevant to an actor's goals [RA10]". Context might be considered to determine the set of goals relevant to a system and derive the alternatives the system can adopt to reach these goals (see link 1 in Fig. 4.1).
For example, in an emergency security system, people may be informed about the nearest emergency door by means of voice instructions. "Give instructions by voice" (Goal) can be adopted only when the "place is not noisy" (Context).

- Context-Vulnerabilities. Vulnerabilities are weaknesses that attacks can exploit. Vulnerabilities may be activated in different environmental situations. For this reason, the nodes representing contextual factors must be related with the those representing the vulnerabilities they activate (see link 2 in Fig. 4.1).
  For example, depending on whether a skype call is executed in a "public network" or in a "home network" (Context), different values for the vulnerability "no encription" can be identified.

- Context-Threats. Threats are the motivation of the threat agents or anti-goals [vL04b]. When the context changes different threats can be affected.
  For example, if "there is an open day" (Context), a the new threat "disturb the guests" can be activated.

- Context-Tasks. Context may affect the tasks as it may cause changes on the state of the system or resources a task acts on.
  For example, the task "open windows to circulate air" aims to change an object, that is the windows, into two different states, "closed" and "open" respectively (context). If the windows is already "opened" (context) the task "open windows to circulate air" cannot be executed.

- context-Security controls: context strongly influences security controls. Laws, enterprise policies and technological constraints regulate the set of security controls that can be used in different environmental conditions.
  For example, the security control "revoke the permission to the student", which modifies the permission stored in a database, cannot be executed if "the database is out of order" (Context)

4.1.2 Contextual fuzzy causal network

A fuzzy causal network is a dynamic system whose topological structure is a directed graph [ZLZ06] made by nodes and links. Its structure is inferred from the elements and links represented in the asset, goal, and threat models. Each node of the FCN represents a security concern. Therefore, it is necessary understand which role is played by the context and indeed detect which kind of node (chance, decision, utility) should be associated with the context.

As described in section 3.3.1 security concerns can be mapped to three types of node: chance, decision or utility nodes. Chance nodes represent uncertain domain entities significant for causal reasoning. Decision nodes indicate decisions to be made. Finally, utility nodes represent the fitness value of the network configuration. Context is mapped to chance nodes, since it is used to describe environmental conditions that belong to the domain space. The value associated with the FCN node "Context" represents the satisfaction level of context, or, in other words, it reflects the degree to which the "context" is active in the system at a particular time. For example, the noiser the place, the higher the relative FCN node.

To connect these nodes with the other security concerns, positive and negative links can be used. Links represent the causal relation from tail (context) to the head of the link (goals, threat goals etc.). Each causal link is labeled with a signed weight, which represents the strength of the causal relationship between the context and the other security concern. Both positive and negative causal links can be used to connect context with the other security concerns.

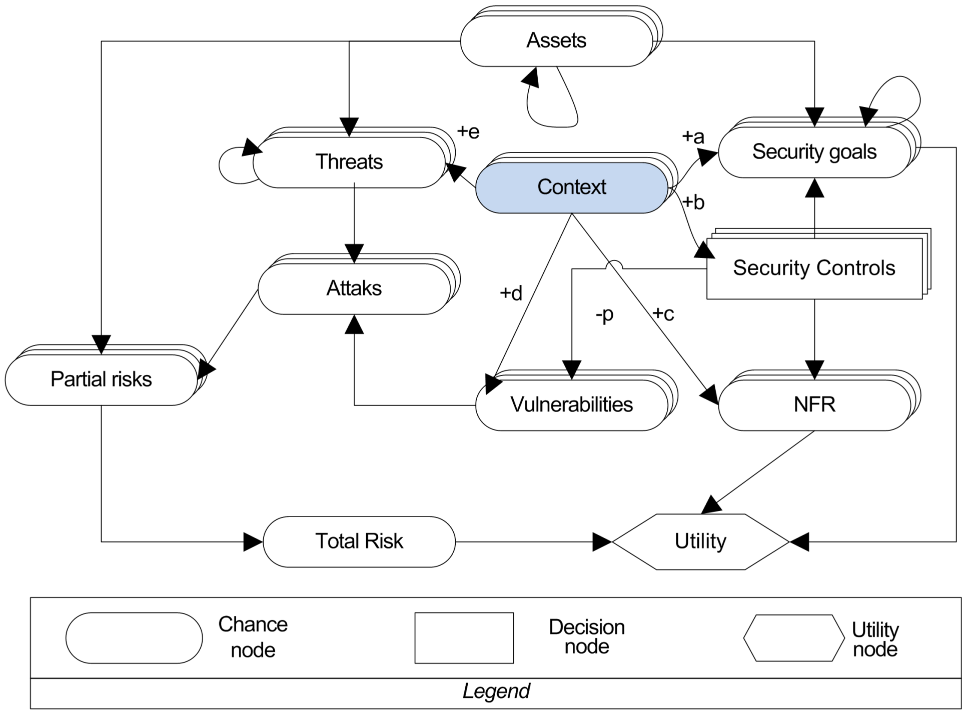


Figure 4.2: The abstract model of the fuzzy causal network

The analysis and decision-making are performed using causal reasoning on the FCN links. Before using the causal network two different steps have to be performed: configuring the aggregation functions, and the initialization and tuning. The following paragraphs analyze how context impacts on these two steps.

Configuring the aggregation functions

The reasoning mechanism of the FCN uses particular functions called aggregation functions to evaluate each node of the FCN. Aggregation functions combine numerical values from the input links into a single representative node value. Context nodes do not need to be associated with aggregation functions, since their value is directly monitored by suitable probes that are available in the system. An appropriate function that convert the physical quantities measured by sensors into a context values, must be designed. The values generated can be fuzzy (in [0,1]) or crisp (0 or 1). However, context is connected to other nodes, such as goals, threats, tasks, security controls and vulnerabilities. Therefore, it is interesting to investigate how context influences their aggregation functions.

- Activation context. In this case, the context can only assume values 0 or 1 (activated/deactivated). This value is used to activate (deactivate), if the links have a positive (negative) causal effect, the nodes that are associated with that context node. The aggregation function of the node connected to context, is designed accordingly.
  For example, in an emergency security system, people may be informed about the nearest emergency door by means of voice instructions. "Give instructions by voice" (Goal) can be adopted only when the "place is not noisy" (Context). "Place is not noisy" is an activation context that can only be associated with values 0-1.

- Fuzzy context. In this case, the context can assume fuzzy values, comprised between 0 and 1. In case of positive causal link, the higher is the value of the context, the higher will be the value of the corresponding security concern. In case of negative causal link, the opposite relation is verified. The aggregation functions of the connected security concerns are designed accordingly.
  For example, the task "open windows to circulate air" aims to change an object, that is the windows, into different states (context), "closed", "vasistas", "partially open" and "opened". The different context values can be represented using fuzzy values.

Initialization and tuning
Once the aggregation functions have been selected the values of the nodes and the weights of the causal links should be initialized. The value of the different nodes of the FCN represents the initial condition of the network. Instead, the weights of the FCN links represents the strength of the causal relationship between different security concerns. The values of the nodes and weights should be tuned based on existing qualitative and quantitative evidence. Context does not strongly affect the sensitivity analysis procedure described in section 3.3.1. That procedure allows system designers to analyze the impact of assets on the other security concerns and on the utility of any possible configuration of security controls. Obviously, when context is considered, this procedure must be reiterated for each combination of context values. This makes the reasoning procedure of the FCN more expensive in terms of time and effort.

4.1.3 Contextual run-time adaptation

The run-time adaptation process presented in section 3.3.3 must also be extended with a notion of context. In summary, the process previously described is composed of four steps.

- Step 1: the system monitors the assets present in the environment and changes the values of the nodes of the FCN accordingly.
- Step 2: the new value of the assets is propagated through the network.
- Step 3: the FCN reasoning is started and the utility of all configurations of security controls is calculated.
- Step 4: the configuration of security control with the best utility is selected.

To consider context, it is necessary to modify steps 1 and 2. In step 1, each contextual factors monitored in the environment must be reflected on a specific node of the FCN representing the context to which an appropriate fuzzy value is assigned. In the step 2, the context value is propagated through the network. If the context type is activation, its is used to activate or deactivate the connected representing other security concerns. If the context type is fuzzy, the value of the context node is normally propagated on the other connected nodes of the network.

4.2 Adaptive Access Control

Context may influence access control policies in different ways. When the context changes (i.e. time, users' location, lighting, temperature etc.), requirements, vulnerabilities, tasks, threats, attacks etc. may be activated or deactivated. For example, in a university, if "the professor [p] is not in the office" (context) the goal of the student [s] "speak with the professor", is deactivated. The different security concerns impact on the set of security controls applied to protect the assets managed by the system. In the previous example, the access control system "revokes the permission to enter to the student" (aecurity control). In other words, granting and revoking permissions does not only depend on "who the user is" but also on "where the user is" and "why the user needs to enter in a specific location".

Assets have a central role in adaptive access control. While the system is executing assets may evolve dynamically: new assets can be added to the system, an asset's value can change, or existing assets may not be under protection anymore. Assets influence other security concerns, such as threats, attacks, vulnerabilities, risks, security goals, and security controls. When the value of an asset increases, tighter policies and authentication measures might be necessary to satisfy the security requirements. In an opposite way, when the assets value decreases, less restrictive policies are chosen to reduce their impact on the business continuity.

The approach of Salehie et. al. [MSLPIORA11], which promotes assets as first-class entities in engineering secure software systems was estended in section 4.1 to consider the context in which the application is working. This section customizes the approach to the analysis of an access control problem.

4.2.1 STrioM for Adaptive Access Control

Access control systems are used by organization to protect their assets. Designing access control systems is not trivial, since it is not only a problem of deciding who can access to a specific location, but includes elements from the world besides the security system and their mutual interplay. On the one hand, context influences security systems: if "the window in the office [o] is opened" (context) the security system can activate "security cameras". On the other hand, security systems modify context: if "the window in the office [o] is opened" (context) the security system can "close the window" (security control). The previous examples highlight how different aspects must be considered in the design of adaptive access control systems.

This thesis mainly focuses on three aspects of access control described by Samarati et. all. [SS94]: authentication, auditing and administration. Authentication aims to verify the identity of a user, process, or device, before granting access to some critical resources of the system. Authorization is the process of granting rights to participants to perform an interaction, for instance to access a resource. Auditing concerns is used to monitor the system. Authentication and administration can be considered as security controls, used to regulate the access to the different resources. Auditing is commonly used to infer contextual information[1]

In this thesis an adaptive access control system is defined as a system able to modify its access rules (authorization), and authentication methodologies when context and assets changes (auditing). To do so, the system analyzes the impact of assets and context modifications on the other security concerns. This section, describes how contextual STrioM and its sub-models (asset, threat and goal model) can be customized to analyze an access control problem. It refers to the e

Asset Model

There are no substantial difference with the asset model described in section 3.2.1.

Goal Model

Security, functional and other non-functional requirements as well as contextual factors provide the rational for revoking or granting access permissions to assets. In the following it is described how the security concerns provided in the goal model can be used to analyze an adaptive access control problem.

- *Actors*: are the users identified in the particular case of study. In the access control case, actors represent users (or their role) that need to access to a particular resource (asset) present in the system. In a task such as "grant permission to enter in the office [o] to the user [u]" the user [u] corresponds to an actor described within the boundary.
  For example, "bachelor, master and PhD student" are possible actors of a university case of study.

- *Goals*: in the access control case, requirements are used to describe the reasons why the users have to access to specific resources (physicals or logicals).

[1]The approach is usable both for physical and logical access control. In chapter 6 to evaluate the approach it is considered a physical access control system.

For example, the student has to access to the professor's office (T1-001) to "speak about an exam", or the student has to access to the paper x for his/her thesis

- *Tasks*: in access control tasks are used to grant or revoke permissions to the different users on specific locations (physical access control) or resources (logical access control). In particular, this kind of tasks have the following structure: Grant [Actor, Resource]. Resources can refer to different areas of the enterprise buildings (physical access control), such as offices, departments, buildings. They can also refer to assets containing intellectual properties (logical access control), such as folders or files.
  For example, the goal "the professor shall regularly meet students" and "meetings shall take place in the professor's office" can be satisfied only if the tasks "grant enter[S,T1-001]" and "grant enter[P,T1-001]" are executed.

- *Context*: is any information that can be used to characterize environmental conditions that can affect the security concerns.
  In the previous example, the goal "meetings shall take place in the professor's office" is activated only if "the professor is in the office".

- *Vulnerabilities*: are activated when particular tasks are executed. As only authentication, authorization and auditing are considered, the vulnerability "intruder can access to the office" can be activated when the task "grant enter[S,T1-001]" is executed or when the authentication measures are not sufficient to correctly authenticate the users.

- *Security Controls*: are a particular kind of task that are used to mitigate the vulnerabilities brought by the system and satisfy the security requirements. In the access control case, security controls include: managing permissions and change authentication procedure. Managing permissions includes security controls with the following structure: Revoke [Actor,Resource]. Authentication procedures may include fingerprints, smart cards, password.
  "For example, revoke enter [S,T1-001]" and "revoke enter [P,T1-001]" are two different security controls used to restrict the access to the office T1-001 to the student (S) and the professor (P), respectively.

Threat Model
There are no substantial difference with the threat model described in sec-

tion 3.2.3.

4.2.2 The fuzzy causal network for Adaptive Access Control

To use the fuzzy causal network presented in section 3.3 and its contextual extension (section 4.1.2) it is necessary to consider the relation between security controls and functional requirements. In the approach proposed by Salehie et. al., risk, non functional requirements and security goals influence the value of the utility node. If a strengthener configuration of security controls is selected the total risks decreases, the satisfaction level of security goals increase, and the satisfaction of non functional requirements decreases. If non functional requirements are not considered the system automatically select the stronger configuration of security controls, this because the contributions of security goals and risk is in the same direction (increase the utility). For example, if iris authentication is used, the risk of threat decreases and the value of the security goals increases. This has a positive impact on the utility. However, the iris also impacts on the usability of the system, which has a negative impact.

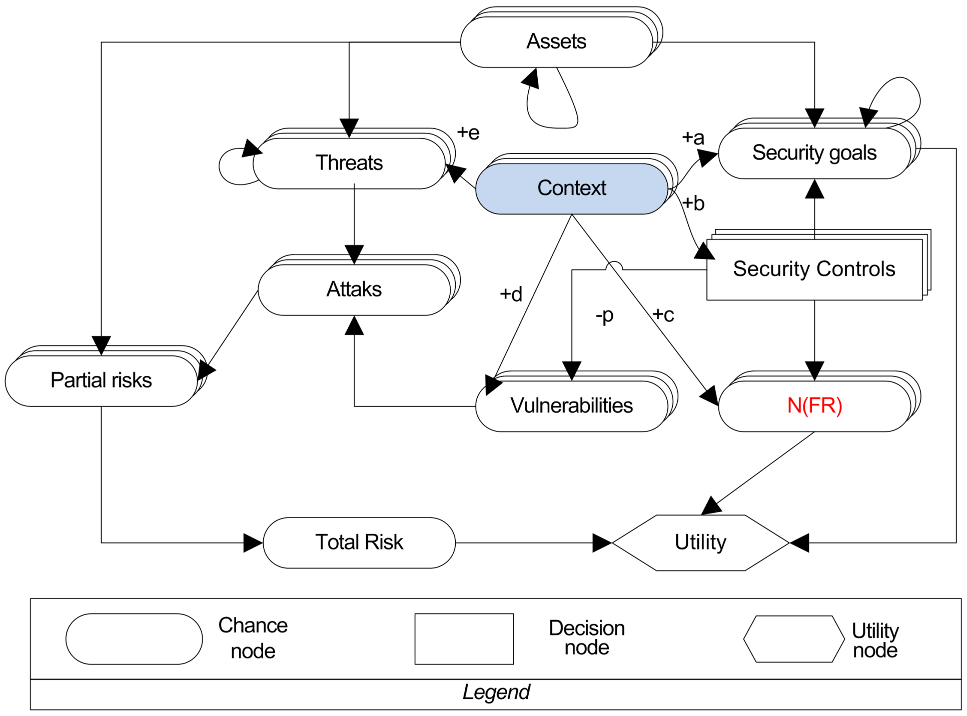


Figure 4.3: Adaptive Access Control: the abstract model of the fuzzy causal network

Using the Salehie et. al. approach in the access control problem, security controls, such as revoke enter [Student, T1-001], have only a positive influence on the utility of the system. Indeed, the system automatically revoke the permissions at all the users of the system. Therefore, it is necessary to explicitly represent the links between the security controls and functional requirements. For example, if the permission is revoked to the student, the student cannot reach his/her goals, such as "speak with the professor". Fig. 4.3 presents the fuzzy causal network where the node NFR is used to represent also the functional requirements (FR).

4.2.3 Run time adaptation for Adaptive Access Control

Run-time adaptation does not present relevant differences, compared to the process described in section 4.1.2. It is pointed out that the reasoning process tries all the combinations of the users permissions and the authentication procedures choosing the most useful.

Chapter 5

SecuriTAS: A Tool for Engineering Adaptive Security

" Simplicity is the soul of efficiency."

Austin Freeman

In this section we describe SecuriTAS, a tool to manage adaptive security that can be plugged in any system. It allows us to use the STrioM model in the runtime adaptation process to identify and apply a proper set of security controls to protect the system. SecuriTAS provides different interfaces that simplify its integration with any system. SecuriTAS provides a monitor interface to receive notification on changes in assets and/or context from the controlled system. It also uses an effector interface to apply the selected security controls in the system. SECTION 5.1 *presents the architecture and the implementation of SecuriTAS.* SECTION 5.2 *describes how SecuriTAS can interact with other systems, especially with a physical access control system. Finally,* SECTION 5.3 *presents the graphical interface of SecuriTAS.*

5.1 SecuriTAS

This section describes the architecture of SecuriTAS (section 5.1.1) and illustrates its implementation (section 5.1.2).

5.1.1 Architecture

SecuriTAS is composed of the Graphical Modeler and the Adaptation Manager (Fig 5.1).

The GRAPHICAL MODELER[1] is a visual editor to create the asset, goal, and threat models. After the asset, goal, and threat models are completed, the Graphical Modeler generates the corresponding FCN XML file (see Link 7 Fig. 5.1) and map each security control to the corresponding security function (see Link 6 Fig. 5.1) implemented in the system through the Security Controls Mapping file. This file is used at run-time to associate the abstract security controls selected by the adaptation manager to protect the system with its real security functionalities.

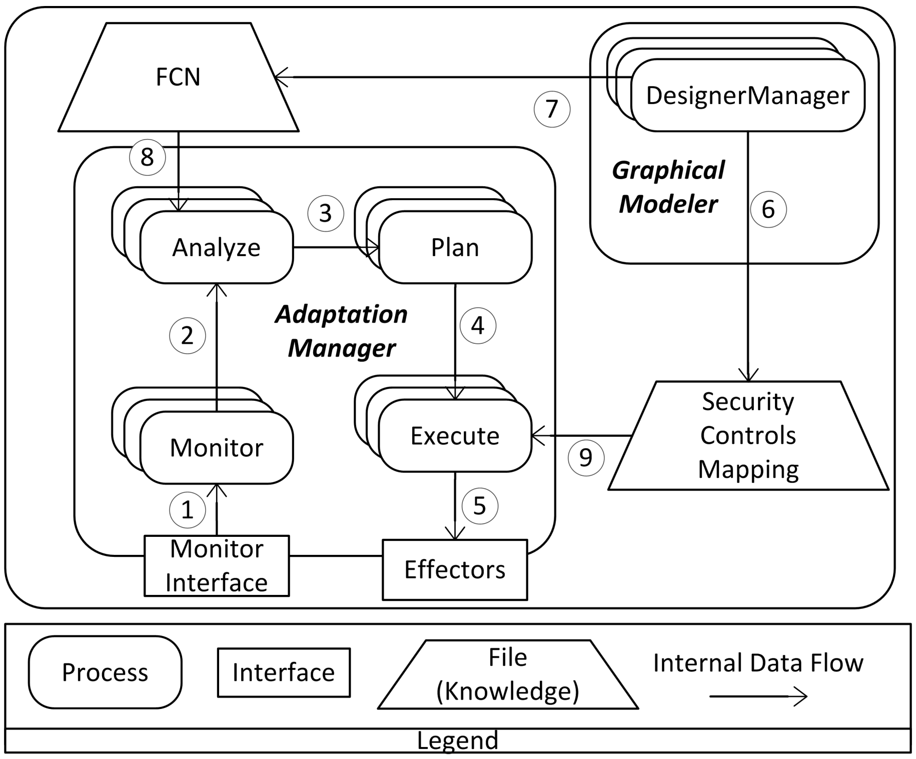


Figure 5.1: SecuriTAS: component diagram

The ADAPTATION MANAGER modifies the system behavior at run-time by checking, rejecting or carrying out adaptations. It implements the activities of the MAPE loop to apply adaptive security at runtime. *Monitor* is

[1]The graphical modeler was developed by Pasquale et. al. [LPN12] and it is customized to consider context.

responsible for collecting, filtering and organizing data from sensors. *Analysis* loads the FCN model generated by the graphical modeler (see Link 8 Fig. 5.1) into a Fuzzy Causal Network and updates its nodes based on the current value of assets and context. *Planning* determines which security controls need to be changed, by selecting the configuration of security controls with the best utility. Finally, *execute* is responsible for applying the security controls selected by planner (see Link 9 Fig. 5.1). To do so, it uses the information present inside the "Security Controls Mapping" XML file generated by the graphical modeler. The monitoring and effectors interfaces are used from the adaptation manager to interact with the other applications present in the system. Customers can use their own applications to monitor the environment. When these applications detect that assets or context changes they call the monitor interface to communicate at the adaptation manager that it is necessary to re-evaluate the security controls used to protect the system. In the same way, customers can use their own applications to protect the system. The effector interface is used from the adaptation manager to modify the behavior of other applications.

5.1.2 Implementation

SecuriTAS is a Java EE (Java Enterprise Edition) server application. It uses JBoss as Java EE application Server, and it is divided in three different tiers: presentation, logic and data tier (Fig 5.2).

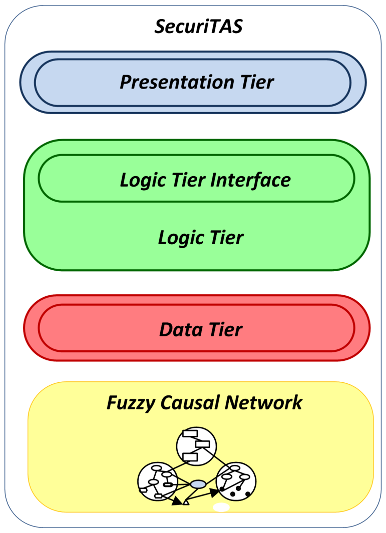


Figure 5.2: SecuriTAS: layer architecture

- The *presentation tier* is the top level of the application. It is used to show at the system administrator the actual state of the FCN reasoning. In this way the administrator can understand why a set of security controls is selected for a certain context and assets' state. It is composed of a set of JSP (Java Server Pages) and HTML pages. The JSP page are made by HTML and JavaScript code.
- The *logic tier* performs the monitor, analysis, planning, and execution activities necessary to support adaptive security. It offers a set of interfaces (API), which are called from the customer applications when the context or assets change. On the other hand, it uses an appropriate interface to apply the set of security controls selected. It is implemented using Java Servlet and Entity Java Beans.
- The *data tier* loads the FCN. More precisely, it generates the FCN using the information stored in the FCNS XML file, and the activation functions selected by the system designer.

In the following are described the implementation choices that have been made to develop SecuriTAS. In particular, the XML used to store the fuzzy causal network, the commands to load and evaluate it.

FCN XML

The FCN generated by the graphical modeler is stored as XML file based on the following DTD (Document Type Definition). A fuzzy causal map is composed of concepts and connections (Line 3). Each concept has several attributes: act, fixed, input, name, output. The concept activator (act) is used to indicate the type of concept node and its activator, such as asset, attack, context (Line 6). The attribute fixed indicate if the value of the node can change (false) or not (true) during the reasoning process (Line 9). Input (Line 10) and output (Line 12) are used respectively to define the initial input and output value of the concept. The connections are used to link the concept together. The attribute from (Line 15) and to (Line 17) indicate respectively the tail and the head of the connection. Finally, fixed or weighted (Line 18) indicate whether the links has/has not a wight associated.

```
1 <!ELEMENT maps ( map ) >
2 <!ATTLIST map name CDATA #REQUIRED >
3 <!ELEMENT map ( #PCDATA | concepts | connections )* >
4 <!ELEMENT concept ( description? ) >
5 <!ELEMENT description ( #PCDATA ) >
6 <!ATTLIST concept act ( ASSET | ATTACK | CONTEXT | CR |
SECUTIRYCONTROL | ORSGOAL | PRISK | RISK | THREAT |
UTILITY | VULNERABILITY ) # REQUIRED >
<!ATTLIST concept fixed ( false | true ) # REQUIRED >
<!ATTLIST concept input (# PCDATA) # REQUIRED >
<!ATTLIST concept name CDATA # REQUIRED >
<!ATTLIST concept output (# PCDATA) # REQUIRED >
<!ELEMENT concepts ( # PCDATA | concept )* >
<!ELEMENT connection ( param ) >
<!ATTLIST connection from CDATA # REQUIRED >
<!ATTLIST connection name CDATA # REQUIRED >
<!ATTLIST connection to CDATA # REQUIRED >
<!ATTLIST connection type NMTOKEN # FIXED ''WEIGHTED'' >
<!ELEMENT connections (# PCDATA | connection )* >
<!ELEMENT param EMPTY >
<!ATTLIST param name NMTOKEN # FIXED ''weight'' >
<!ATTLIST param value (# PCDATA) # REQUIRED >
```

Fuzzy Causal Network

To perform the analysis (See Fig. 5.1) the fuzzy causal network must be loaded from the XML and translated into a java causal network. The fuzzy causal network is implemented as Java Fuzzy Cognitive Map (FCM) provided by the JFCM Java library (http://jfcm.megadix.it/). The FCM is composed of concepts and connections. To instantiate a new concept it necessary to execute the command:

```
Concept c1 = new Concept(''name'', ''description'', ''activation
function'', ''input'', ''output'', ''fixed output'');}
```

Where activation function is the function used to evaluate the node, input and output are values between 0 and 1 which represent the initial input and output of the concept, and fixed output is true when the concept does not change its value when the FCN is evaluated. To instantiate a new connection the following command is executed:

```
FcmConnection conn1 = new WeightedConnection(''c1 to c4'',
''Asset to Threat'', 0.8);}
```

Where “c1 to c4” is the name of the connection, “Asset to Threat” and 0.8 are respectively its description and weight. The create a new fuzzy causal map it is necessary to instantiate a new CognitiveMap object.

```
CognitiveMap map = new CognitiveMap();
```

Then concepts and connections are added to the network. The following command is used to connect the concept “c1” with the concept “c4” using the connection “c1 to c4”. “c1”, “c4” and “c1 to c4” are respectively the name of the two concepts, and the connection.

```
map.connect(‘‘c1’’, ‘‘c1 to c4’’, ‘‘c4’’)};
```

Using Fuzzy Causal Network in adaptation

As previously described, during analysis, the FNC is loaded from the XML file. The FCNXmlFileManager is the class in charge of loading the XML file of the fuzzy causal network. Each concept and connection is added at the CognitiveMap as previously described.

```
CognitiveMap cognitiveMap=FCNXmlFileManager.loadXml(
new ConfFileManager().getInitialMap());}
```

The method loadXml of the FCNXmlFileManager class returns the java fuzzy causal network associated with the FCN stored in the XML file. To do this, the method getInitialMap() of the class ConfFileManager() is used to get the location of the XML FCN file. Before starting the FCN reasoning, the CognitiveMap is updated with the actual values of context and assets:

```
CausalNetworkManager.setAssetsValue(cognitiveMap,assetInput);
CausalNetworkManager.setContextValue(cognitiveMap,contextInput);
```

The CausalNetworkManager is the class designed to manage and update, the fuzzy causal network. During the executing phase the fuzzy causal network is evaluated and the configuration of security controls with the best utility selected. Since the FNC is modified during the reasoning process, before starting the reasoning the CognitiveMap is temporally copied in a map (initialMap). Indeed, every time a new configuration of security controls has to be evaluated, the network has to be set to its initial conditions (stored inside the initialMap). In the following, the main commands used to select the security controls configuration to apply are described.

```
 1 while(generator.hasNextConf())
 2 {
 3 runner.setMap(mpa);
 4 cognitiveMap=generator.loadNextConfiguration(map);
 5 runner.converge();
 6 if(bestConfUtility<CausalNetworkManager.getUtility(map))
 7 {
 8 bestConfUtility=CausalNetworkManager.getUtility(map);
 9 secContBestConf=CausalNetworkManager.getSecCont(map);
10 }
11 CausalNetworkManager.restInitMap(map,initialMap);
12 }
```

Line 1: the generator, which is responsible to generate all possible configurations of security controls checks wheter there are other configurations that need to be evaluated.

Line 3: the runner (is used to run the FCN to evaluate the utility of the

configuration of security controls generated in the previous step) is updated with the map that contains the FCN.
Line 4: the next possible configuration of security controls is loaded by the generator.
Line 5: runner.convergence() evaluates the fuzzy causal network for the configuration of security controls selected in the previous step.
Line 6: controls the utility of the new configuration.
Line 8: the value of the best utility is updated.
Line 9: the value of the best security controls configuration is updated. If the utility is greater than the one obtained for the previous configuration, the best configuration of security controls is updated and set to the current one.
Line 11: the initial configuration of the causal network is restored.

5.2 SecuriTAS and access control

This section describes how SecuriTAS can be integrated with other software systems. In particular, in this thesis SecuriTAS is used to realize an adaptive access control system. Section 5.2.1 describes the adaptive access control system architecture. Section 5.2.2 describes the main implementation aspects of the adaptive access control system.

5.2.1 Architecture

The Adaptive Access Control System is composed of: the Access Control Manager, the Security Control Manager and SecuriTAS (Fig. 5.3).

Access Control Manager
Classical access control managers provide several functionalities, such as authentication, authorization, centralized policy administration, advanced session management, and logging. In this thesis it is developed a simple prototype of an access control manager, which offers only authentication and authorization. These functionalities are usually provided by the *Identity Manager*. When a user wants to access to a particular location, his/her data are sent to the Access Control Manager (more precisely to the identity manager) that authenticates the user. When the user is authenticated, the system checks if he can enter in the location. In the first case the system opens the door by using a specific effector and shows a message on the screen outside the door (see Link 5 in Fig. 5.3)[2]. In the other case, the door is not

[2]In this thesis the screen is emulated using an applet.

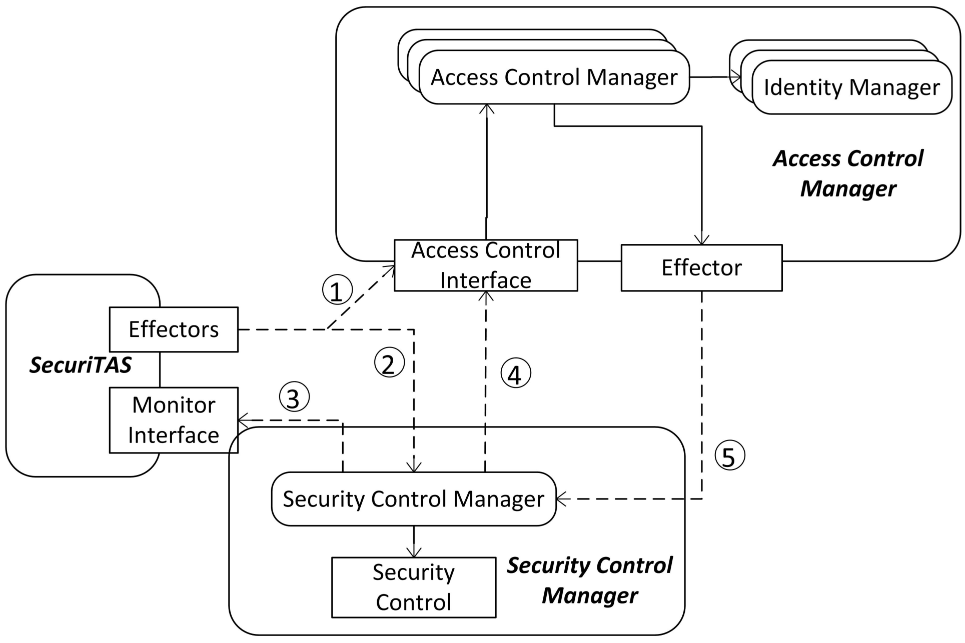


Figure 5.3: Component diagram of the Adaptive Access Control System

opened and a message is showed on the screen.

Security control manager

The security control manager is used to manage the devices present in the environment. In this thesis the AET62 NFC Reader is used to monitor who accesses to/exits from the office and which assets are moved to/removed from the office. When a card is swapped on the reader, the system detects if the card is associated with an asset or a user. In the first case, the asset information, such as the asset value, is sent to SecuriTAS (see Link 3 in Fig. 5.3) to select the set of security controls to protect the system. In the second case, the user login information is sent to the Access Control Manager (see Link 4 in Fig. 5.3), which authenticates the user and detects if he/she can enter in the location and applies a set of appropriate security controls.

SecuriTAS

SecuriTAS is responsible for the system adaptation. It modifies the authentication mechanisms (i.e., PIN, fingerprint, password) and the access control rights. In the first case, the new combination of authentication methodologies is communicated at the Security Control Manager (see Link 2 in

Fig. 5.3). In the second case, the new access control rights are sent to the Access Control Manager (see Link 1 in Fig. 5.3).

5.2.2 Implementation

The Adaptive Access Control System proposed is composed of the Access Control Manager and the Security Control Manager. The Access Control Manager is used to authenticate the different users and store login information, such as fingerprint, password, smart card information. The Security Control Manager is used to manage the different devices present in the environment. In this thesis, the Security Control Manager manages the AET62 NFC Reader[3], which is used to monitor who accesses to/exits from a particular location, such as an office, and which assets are moved to/removed from the office. If an employee swipes his/her smart card on the NFC reader, he/she can enter only in case he/she has the permission to access to that location. In case an employee is in a specific location and swipes his/her card, it means that he/she exited from that location. Smart cards are also used to tag assets in the system. For instance, in case an asset is outside a particular location and its card is swiped, it means that this asset is moved to that location.

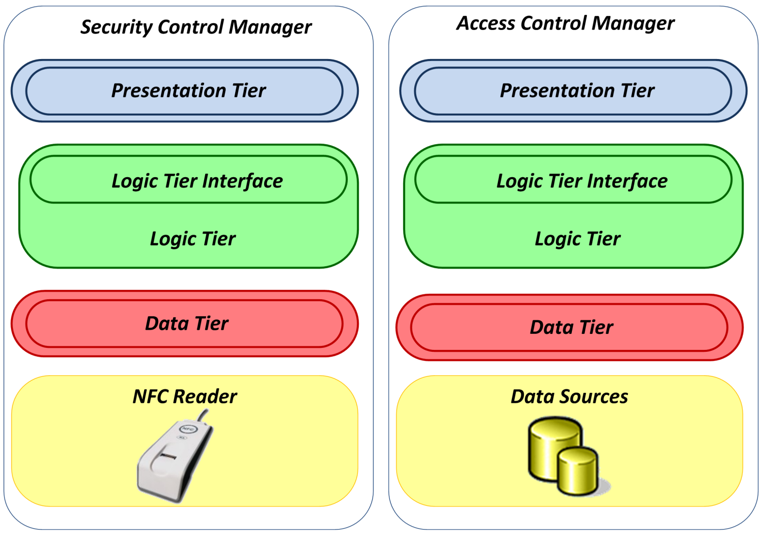


Figure 5.4: Adaptive Access Control: layer architecture

The Access Control Manager is a Java EE (Java Enterprise Edition) server application. It uses JBoss as Java EE application Server, and it is

[3]http://www.acs.com.hk

divided in three different tiers: presentation, logic and data tier(Fig. 5.4).

- The presentation tier is the top level of the Access Control Manager. It is used to show the actual state of the system to the administrator. The system state includes the permissions of the different users and the status of the login devices. This layer it is implemented as a set of JSP (Java Server Pages) and HTML pages.
- The logic tier performs the logical decisions and evaluation. It authenticates the different users, and offers a set of interfaces (API), which are called from SecuriTAS (more precisely from the adaptation manager) to modify the users' permissions. It is implemented using Java Servlet and Entity Java Beans.
- The data tier manages the information stored in a MySql database, such as the users biometric information, and the access control lists that specify who can access to the different locations to be protected.

The Security Control Manager is a Java Applet application which manages the AET62 NFC Reader. It is organized into three tiers: presentation, logic and data tier (Fig. 5.4).

- The presentation tier emulates a screen that can be located outside a particular office or area of the building. It shows to the system's users if they can/cannot enter in a particular location once they are authenticated. It is implemented as a java applet.
- The logic tier calls the proper functions of SecuriTAS and Access Control Manager based on the card swapped on the NFC Reader. If the card is associated with an asset, it means that a new assets is put inside the location protected by the NFC Reader. The information regarding the asset is forwarded to SecuriTAS that applies a specific adaptation to protect the system. If the card is a user's smart card, the login tier calls the Access Control Manager to detect if the user can enter in the location.
- The data tier provides a set of interfaces to communicate with the different devices, such as the AET62 NFC Reader, that is used in this thesis. For example, this interface can be used to read users login information or to check if the reader is connected or available.

Access control manager

The access control manager is developed as a simple Java EE (Java Enterprise Edition) server application. The class *User* and *Role* are respectively used to represent the users of the system, and the different roles they can assume. These classes are implemented as java EJB (Enterprise JavaBean). For example, Claudio (user) is associated with the student role. The class *User* contains the username of the user (primary key), smart card code, and a pointer to the role of the user in the enterprise. The class *Role* has three different attributes: a string which represents the name of the role (primary key), a boolean, that is true if the users with that role can access to the particular location, and the collection of users with the specified role. When a particular user swaps his/her card on the NFC reader, the Security Control Manager reads the code of the card. If the card does not represent an asset, it communicates the code of the user (stored on the card) to the access control manager using the Access Control Interface. The access control manager receives the smart card code of the user and detects his/her role. If the user with the role identified can access to the location, the access control system opens the door, otherwise it shows a message on the screen located outside the door. More precisely, the following instructions (written in pseudo code) are executed:

```
Get the smart card code (from the Access Control Interface)
Get the user associated with that code.
Get the role of the user.
if the user with that role can access to the location.
        open the door.
else
        show a message on the screen outside the door.
```

Security control manager

The security control manager is used to manage the AET62 NFC Reader (Fig. 5.5). This device combines the ACS' ACR122U Near Field Communication (NFC) Reader and UPEK's swipe fingerprint sensor, and provides a two-factors authentication system. However, only NFC smart cards (Milfare 4k smart card) are used in this thesis to authenticate the different users. Only two bytes of these smart cards are used. The byte (0x00) indicates if the card refers to a user (0x00) or an asset (0x01). If the card is associated with an asset the second byte indicates the value of the particular asset. Indeed, if the card is associated with a user, it indicates the code of the user[4]. In this prototype a maximum of 16 users can be managed. The following code is used to manage the NFC reader.

```
NfcManager nfcManager=new NfcManager();
nfcManager.setConnection();
while(true)
```

[4]At each user is associated a unique smart card code.

Figure 5.5: AET62 NFC Reader

```
 4 {
 5 nfcManager.waitForCardPresent();
 6 nfcManager.authenticateBlock((byte) 0x04);
 7 byte[] cardCode=nfcManager.readBlock((byte) 0x04,(byte) 0x10);
 8 if(cardCode[0]==(byte) 0x01)
 9 {
10 monitorInterface.updateAsset(cardCode);
11 writeMessage(``New asset detected, the network is updated.'');
12 }
13 else
14 {
15
16 if(accessControlInterface.checkAccessPossibility(cardCode))
17 {
18 String name=accessControlInterface.getUsername(cardCode);
19 writeMessage(``Hi,'' +name+'' The door is open. ``);
20 door.openDoor();
21 Thread.sleep(5000);
22 door.closeDoor();
23 }
24 else
25 writeMessage(``Hi,'' +name+ ''you cannot access to the room ``);
26 }
27 nfcManager.waitForCardAbsent();
28 }
```

Line 1: the NfcManager, which is responsible of managing the AET62 NFC reader is created.
Line 2: the NfcManager instantiates the connection with the NFC reader, in case no connection has been established yet.
Line 5: the NfcManager waits that a user swaps a card on the reader.
Line 6: when the card is detected the NfcManager authenticates the block

number 4[5].
Line 7: the NFC Nanager reads 10 byte of the block number 4.
Line 8-12: if the first block is equal to 0x01 the card represents an asset. In this case, it is necessary to call SecuriTAS to evaluate the set of security controls to apply. monitorInterface.updateAsset communicate the new value of the asset to SecuriTAS. Furthemore, (*Line 11*) the message "hi" plus the name of the user plus "you cannot access to the room" is written on the screen (More precisely on the applet that emulates a screen outside the door).
Line 14-23: in case the card does not represent an asset, the card is used to identify a user. The NFC Manager uses the remote call checkAccessPossibility(cardCode) to communicate at the Access Control Manager that the user wants to access to a particular location. If the user can access the remote call returns true. In this case the message "Hi "+name+"The door is open" is written on the screen (*Line 19*) and the door is open (*Line 20*). The user has 5 second to enter in the location (*Line 21*), before the door the door is closed (*Line 22*). If the user cannot access to the room the message "Hi "+name+" you cannot access into the room" (*Line 25*) is written on the screen. Finally, the system waits until the card is removed from the reader (*Line 27*).

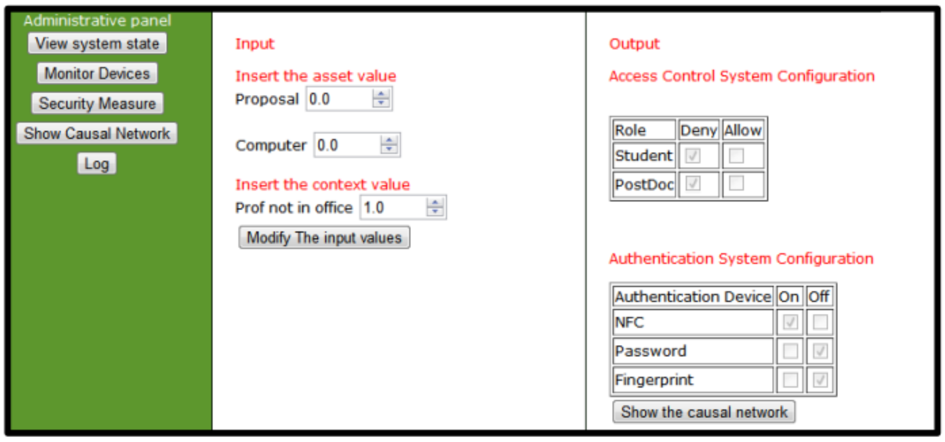
5.3 Graphical interface

This section provides a briefly overview on the graphical interface of SecuriTAS. As previously described, SecuriTAS is still a prototype. At the moment, the graphical interface of the access control system and the adaptation manager are still joined together. From the dashboard provided (Fig. 5.6) it is possible to:

- *View the system state.* It shows the current state of the system: the context, the value of the assets that need to be protected, and the security controls necessary to protect the system. The administrator can manually modify the value of the assets and the context by pushing the button "modify the input values". In this way SecuriTAS can be used as a stand alone application.

- *Monitor devices.* It shows the value of the assets that need to be protected and the context. In the future this page can show also the

[5]The Milfare 4k smart card is composed of four blocks of 1k. Before access to the data stored in a particular block it is necessary to authenticate the block of the card.


Administrative panel

View system state

Monitor Devices

Security Measure

Show Causal Network

Log

Input

Insert the asset value

Proposal 0.0

Computer 0.0

Insert the context value

Prof not in office 1.0

Modify The input values

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc	✓	

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

Show the causal network

Figure 5.6: SecuriTAS Dashboard

state of the devices connected and topological information about the assets location.

- *Security measures.* It shows the state of the security controls controls used to protect the system.
- *Show the causal network.* It shows the actual state of the causal network (Fig. 5.7): the value of the nodes and the links between them. The administrator can understand why a particular set of security controls is selected to protect the system.

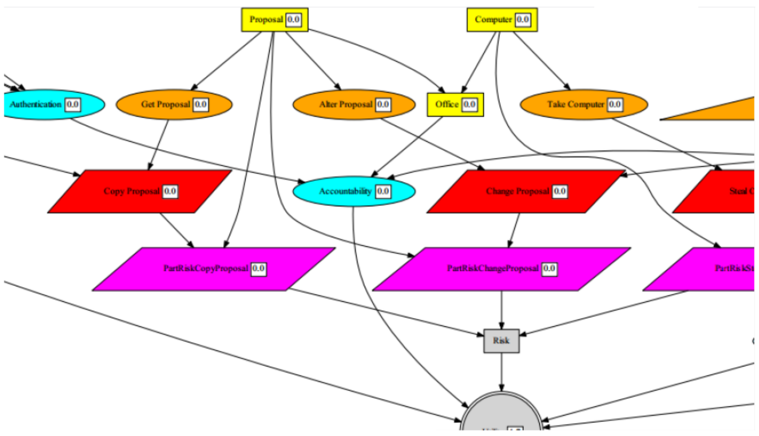


Figure 5.7: Partial view of the Fuzzy Causal Network

Chapter 6

Evaluation

" Experience without theory is blind, but theory without experience is mere intellectual play."

Immanuel Kant

The following chapter shows a feasible case study to evaluate SecuriTAS and the approach described in chapter. 4. SECTION 6.1 *describes the case study set in a realistic research centre, namely the Irish Computer Science Research Centre (ICSRC).* SECTION 6.2 *uses several scenarios to compare the behavior of an adaptive access control system with that of a non-adaptive one. Finally,*SECTION 6.3 *presents several considerations on the evaluation results.*

6.1 The ICSRC case study

ICSRC case study aims to regulate the access of the employees to a particular office present in its research centre building. The access control system has to be able to dynamically apply a set of policies that minimally impact on the activities performed in the research centre and still guarantee the security requirements. More precisely the access control system has to consider what and why should be protected and how to manage the consequences of changes. In other words, the access control system should be requirements-aware, which means that it should be able to consider the security requirements for defining and maintaining the security policies.

The ICSRC wants to guarantee the security of the assets present in the building. To this end, critical assets are identified and tracked, and access control policies will be adjusted when these assets change. Security codes are used to classify assets. For information assets the security codes represent the sensitivity of the information, for physical assets they represent the value of the object itself. Table 6.1 presents the codes and the meaning of the symbols used to classify information and physical assets.

The access control policies applied on the system must be based on the role of the people who can access to the ICSRC building. In this case study three roles are considered: student, post-doc and professor. The purpose of this system is to regulate the access to a particular office of the building (T1-001).

Information Asset		**Physical Asset**	
Code	*Meaning*	*Code*	*Meaning*
TS	Top Secret	VH	Very High
S	Secret	H	High
C	Confidential	M	Medium
U	Unclassified	L	Low

Table 6.1: Security code and meaning of both information and physical assets.

6.1.1 Modeling requirements and security concerns

STrioM is used to model the requirements and security concerns for the IC-SRC case study. To this aim, it is necessary to represent the asset, goal and threat model, as described in the previous chapters.

Asset Model

Assets are the main entities the access control system must protect. In this case of study the asset model takes into account two different aspects: the value of an asset and its location. The assets value is used to describe its criticality. Table. 6.1 details the list of security code used to classify assets values. Locations are represented in the asset model as new security concerns that are connected to the assets.

For simplicity, in the ICSRC case study one location (the office T1-001) is considered, together with two assets: a physical asset (Computer) and an information asset (Proposal). The asset model is represented in Fig. 6.1.

Goal Model

In this section, a different goal model is designed for each role considered in the case study. As described in the previous chapters, the goal model represents the security requirements, necessary to protect assets, as well as other non-functional requirements, such as performance and usability.

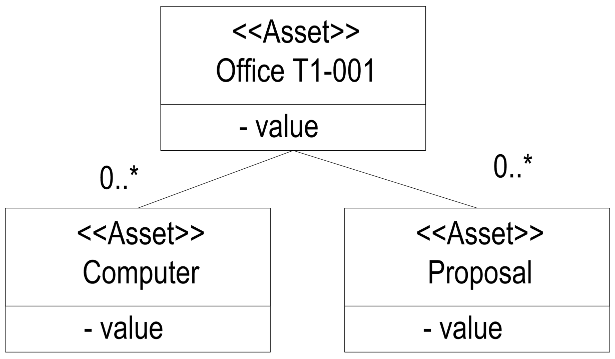


Figure 6.1: ICSRC: Asset Model

Professor Goal Model

The goal model for the professor role (Fig. 6.2) considers the following goals:

- *Manage projects and researches*: one of the main goal of the professor is the management of projects and researches. In particular, to reach this goal the professor supervise students, and use devices present in a particular office.
- *Meet students in an office*: the professor should meet the students in a particular office of the building (T1-001 in this case).
- *Use the devices present in the office*: the professor may have the necessity to use the devices present in a particular office.

To satisfy these goals the system has to execute two different tasks:

- *Grant [P,T1-001]*: grants to the professor the permission to enter in office T1-001.
- *Grant [S,T1-001]*: grants to the student the permission to enter in office T1-001.

These tasks can activate the vulnerability:

- *Intruder can access to the office*: the tasks grant [P,T1-001] and grant [S,T1-001] could allow an intruder to access to the office.

Two security controls that are used to mitigate this vulnerability, impact on the user's goals:

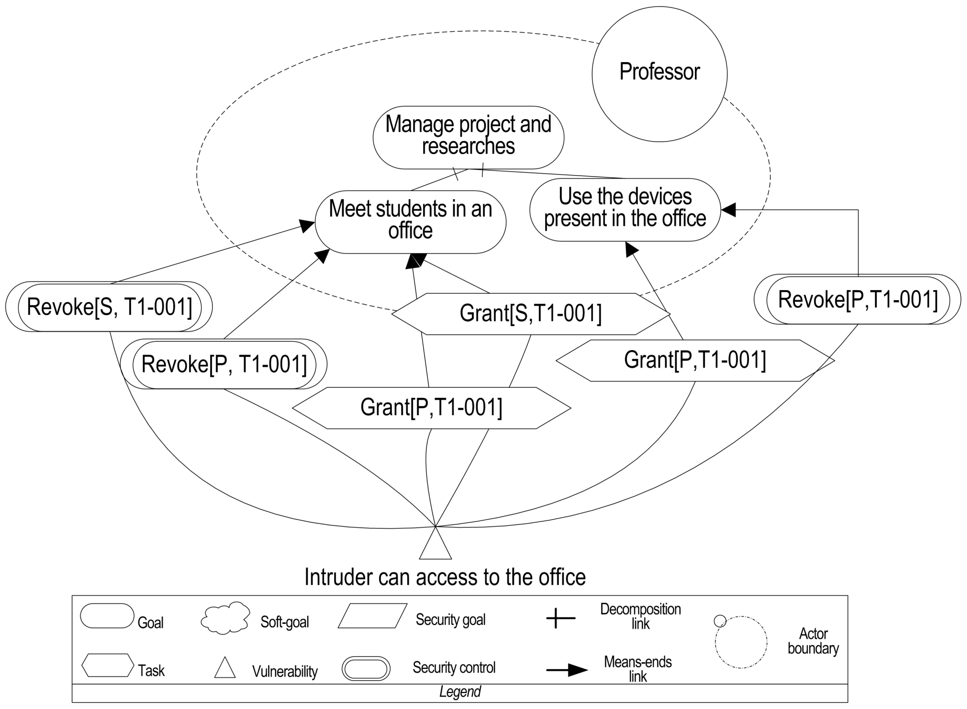


Figure 6.2: ICSRC: Professor Goal Model

- *Revoke [P,T1-001]*: revokes to the professor the permission to enter in office T1-001.
- *Revoke [S,T1-001]*: revokes to the student the permission to enter in office T1-001.

Post-doc Goal Model

The goal model for the post-doc role (Fig. 6.3) considers the following goals:

- *Manage his/her research*: one of the main goal of the post-doc is the management of his/her research. In particular, to reach this goal the post-doc has to use devices present in a particular office and meet the professor in his/her office.
- *Meet professor in an office*: the post-doc should meet the professor in a particular office of the building. In this case the professor's office (T1-001).

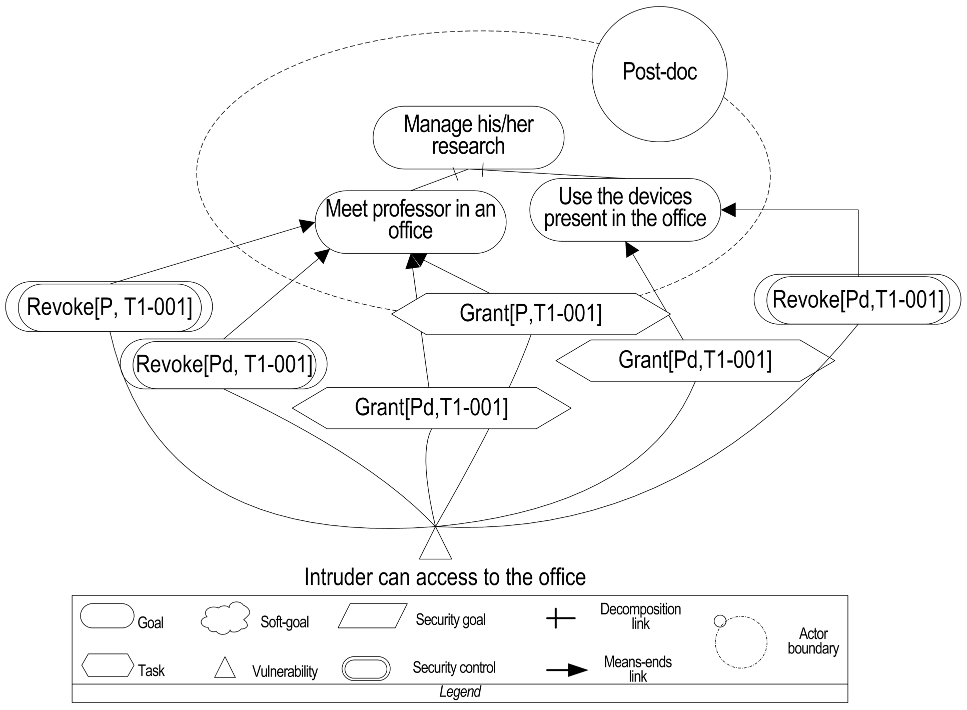


Figure 6.3: ICSRC: Post-doc Goal Model

- *Use the devices present in the office*: the post-doc may have the necessity to use the devices present in the professor's office (T1-001).

To satisfy these goals the system has to execute two different tasks:

- *Grant [Pd,T1-001]*: grants to the post-doc the permission to enter in office T1-001.

- *Grant [P,T1-001]*: grants to the professor the permission to enter in office T1-001.

These tasks can activate the vulnerability:

- *Intruder can access to the office*: the tasks grant [Pd,T1-001] and grant [P,T1-001] could allow an intruder to access to the office.

Two security controls that are used to mitigate this vulnerability, impact on the user's goals:

- *Revoke [Pd,T1-001]*: revokes to the post-doc the permission to enter in office T1-001. This security control negatively impact both on the goal “meet professor in an office and on “use the devices present in the office”.

- *Revoke [P,T1-001]*: revokes to the professor the permission to enter in office T1-001. This security control negatively impact on the goal “meet professor in an office.

Student goal model

The student goal model (Fig. 6.4) includes the following goals:

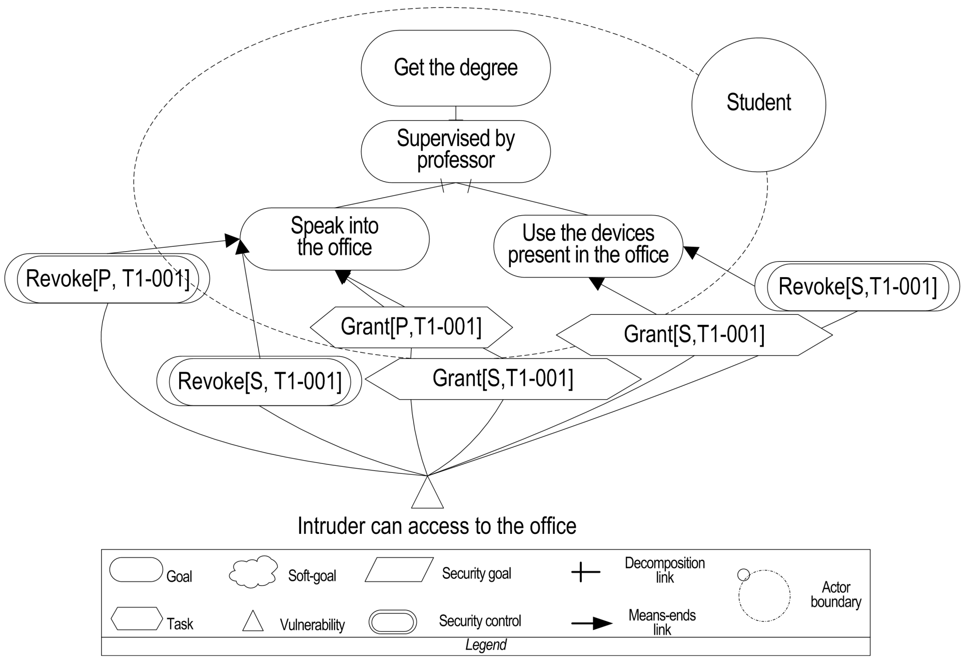


Figure 6.4: ICSRC: Student goal model

- *Get the degree*: the main goal of each student is to get the degree. To reach this goal the student may need to complete the exams and deliver a thesis project. To deliver a thesis project the student must be supervised by a professor.

- *Supervised by professor*: to deliver the thesis project the student needs to be supervised by a professor to discuss the project. The student may need to speak with the professor or use the devices present in the professor’s office.

- *Speak into the office*: if both the professor and the student have the access to a particular office of the building they can meet inside it.
- *Use the devices present in the office*: to finish his thesis the student may have the necessity to use a particular device (e.g., computer, sensors) that are present in a particular office.

To satisfy these goals the system has to execute two different tasks:

- *Grant [P,T1-001]*: Grant to the professor the permission to enter in the office T1-001.
- *Grant [S,T1-001]*: Grant to the student the permission to enter in the office T1-001.

These tasks can activate the vulnerability:

- *Intruder can access to the office*: the tasks grant [P,T1-001] and grant [S,T1-001] could allow an intruder to access to the office.

Two security controls that are used to mitigate this vulnerability, impact on the user's goals:

- *Revoke [P,T1-001]*: revokes to the professor the permission to enter in office T1-001.
- *Revoke [S,T1-001]*: revokes to the student the permission to enter in office T1-001.

Security and non-functional requirements

The security and the non-functional requirements of the system are illustrated in Figure. 6.5 and are described in the following:

- *Accountability*: is related to the ascertain responsibility of actions on a specific resource.
- *Authentication*: is the process used to verify the identity of a person that wants to access to the office. This goal can be reached using several types of authentication devices, such as PIN, fingerprint and iris readers.
- *Authorization*: is used to determine who can access to the office (regulates the permissions of the T1-001 office).

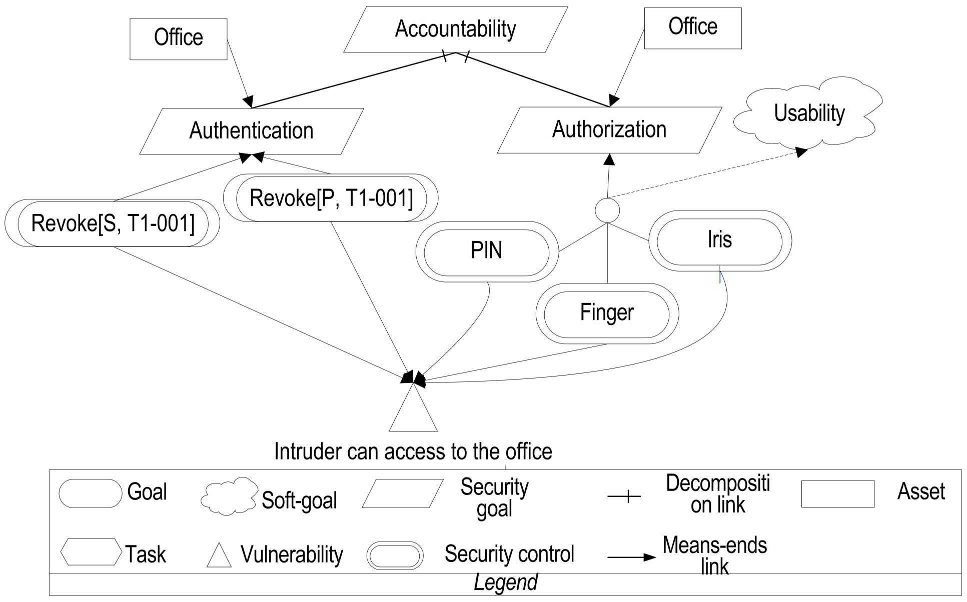


Figure 6.5: ICSRC: Security Goal Model

- *Usability*: security controls (PIN, finger and iris) have a negative impact on the system usability. Usability represents how specified products can be used effectiveness, efficiency and satisfaction in a specified context of use [924]. For example, using iris is less usable than PIN since scan the iris is less comfortable than writing a PIN code.

Several security controls are used to reach security goals:

- *Pin*: personal Identification Number is a secret code used to authenticate students, post-docs and professors.
- *NFC Smart Card*: NFC smart cards contains a unique code used to authenticate students, post-docs and professors.
- *Fingerprint*: are used to scan the fingerprint and authenticate students, post-docs and professors.
- *Revoke [S,t1-001]*: revokes to the student the permission to enter in the office T1-001.
- *Revoke [P,t1-001]*: revokes to the professor the permission to enter in the office T1-001.

These security controls handle the following vulnerability:

- *Intruder can access to the office*: the security controls revoke [P,T1-001] and revoke [S,T1-001] are used to mitigate the vulnerability “intruder can access to the office”.

Threat Model

The threat model of the ICSRC case of study is shown in Figure 6.6. It is characterized by the following threats:

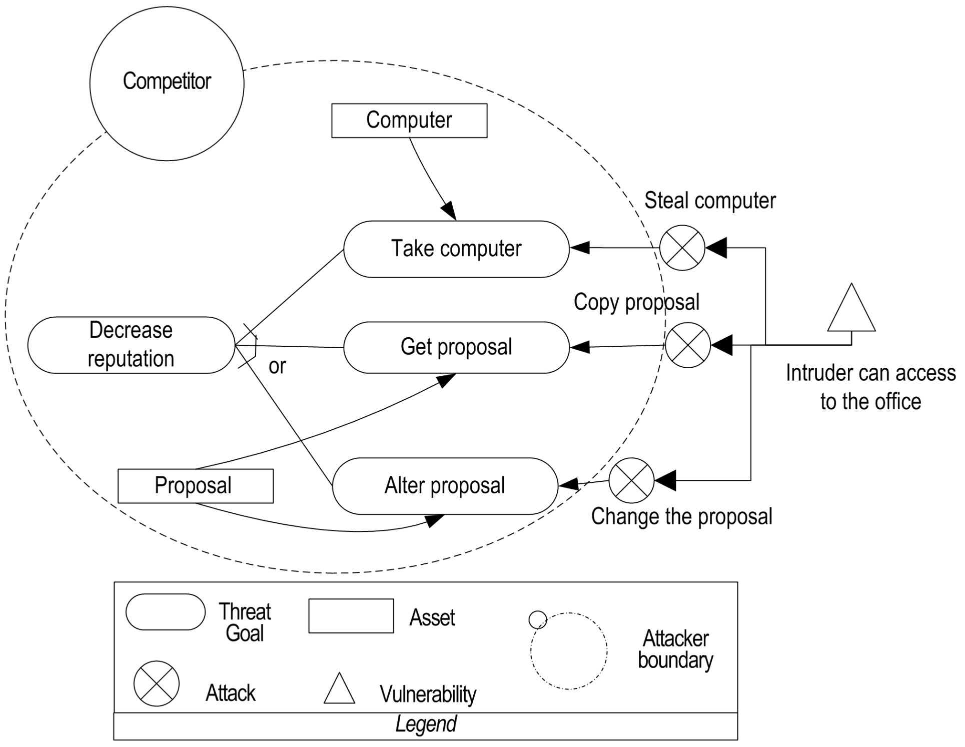


Figure 6.6: ICSRC: Threat Goal Model

- *Decrease reputation*: is the main goal of the attacker. This goal can be satisfied by taking the computer, getting or altering the proposal.
- *Take the computer*: the attacker can get money directly by stealing devices, as for example the computer.
- *Get the proposal*: by getting the proposal the attacker can steal ideas or customers.

- *Alter the proposal*: by altering the proposal the attacker can eliminate competitors from the business.

The threats *get the proposal*, *alter the proposal* and *take the computer* are respectively related to the value of the proposal and computer. The higher the value of *proposal/computer*, the higher the level of the threat *get-alter the proposal/take the computer.* Several attacks can be executed by an attacker to reach such threats:

- *Steal the computer*: if the computer is in the office, the attacker takes the device (computer) out of the office.
- *Copy the proposal*: by using electronic devices, such as USB pen or a phone, confidential information can be copied.
- *Change the proposal*: the attacker modifies the information present in the office to eliminate competitor from the business

These attacks exploit the vulnerability *intruder can access to the office* to succeed.

6.1.2 Fuzzy Causal Network

A fuzzy causal network is used to support the analysis of assets and context relevant changes and facilitate the decision making. The causal network is built from the three discussed models and enable us to analyze the impact of the asset-relevant changes, and to select the set of security controls to apply at run-time to mitigate vulnerabilities. Each security concern is translated to a FCN node. At each node of the FCN is associated an aggregation function used to compute the value of the relative node by aggregating the contributions of its incoming links. Table 6.2 shows the aggregation function used for each node. The term $\{A\} \rightarrow B$ denotes the set of nodes of type A that are causally affecting B. The term $\{A\}, \{B\} \rightarrow C$ indicates that the reasoning mechanism initially aggregates the links of type A and B and then combines the different types.

- $\{As\} \rightarrow$ AS: assets are aggregated together using the average function. This choice makes possible to take the middle level between the maximum and the minimum value of assets.
  In the case study, the value of the office is obtained taking a middle level between the value of the computer and the proposal present in the office.

Causal Link	**Aggregation**
$\{As\} \rightarrow As$	Average
$\{As\} \rightarrow T$	Maximum
$\{T\} \rightarrow At$	Maximum
$\{V\} \rightarrow At$	Weighted Average
$\{T\}, \{V\} \rightarrow At$	Minimum
$\{As\} \rightarrow SG$	Maximum
$\{SC\} \rightarrow SG$	Average
$\{As\}, \{SC\} \rightarrow SG$	Minimum
$\{SC\} \rightarrow V$	1+Weighted Average
$\{SC\} \rightarrow NFR$	1-Weighted Average
$\{SC\} \rightarrow FR$	1-Weighted Average
$\{As\} \rightarrow PR$	Maximum
$At \rightarrow PR$	No aggregation
$\{As\}, At \rightarrow PR$	Minimum
$\{PR\} \rightarrow PR$	Maximum
$TR \rightarrow U$	Weighted Sum
$\{SG\} \rightarrow U$	Weighted Sum
$\{NFR\} \rightarrow U$	Weighted Sum
$\{FR\} \rightarrow U$	Weighted Sum
$TR, \{SG\}, \{NFR\}, \{FR\} \rightarrow U$	Sum

Table 6.2: Aggregating causal effects

- $\{As\} \rightarrow$ T: the level of threat goals depends on the value of the assets that can be harmed. In particular, threat goals aggregate the values of assets by using the maximum function.
  In the case study, each threat is connected with exactly one asset. Therefore, maximum, minimum and average are interchangeable.

- $\{T\} \rightarrow$ At: the probability of an attack is related to the maximum level of its threat goal.
  In the case study each attack is connected with exactly one threat. Therefore, maximum, minimum and average are interchangeable.

- $\{V\} \rightarrow$ At: the probability of attacks is related to the presence of system vulnerabilities that the attacker can exploit. The value of an attack depend on the weighted average of the related vulnerabilities.
  In the case study, the probability of success of the attack "steal computer" depend on the probability of occurrence of the vulnerability "intruder can access to the office" .

- $\{T\}, \{V\} \rightarrow At$: threat and vulnerabilities are aggregated together using the minimum function. This choice gives to the attack a low level when the value of the threat goals is high but no vulnerabilities

are present in the system, or, on the opposite way, when the value of the vulnerabilities is high but threat level is low, which means that the value of the assets present in the system is low. Indeed, to estimate the probability of an attack it is necessary to consider not only the vulnerabilities associated to the attack but also the threat level.
In the case study, the probability of the attack "Steal computer" is calculate as the minimum between the level of the threat "Take the computer" and the probability of occurrence of the vulnerability "Intruder can access to the office".

- $\{As\} \rightarrow SG$: the assets connected to a security goal are aggregated together using the maximum function, since in the case of study protecting assets is extremely important.
  In the case study, the satisfaction level of "accountability" depends on the value of the asset "office".

- $\{SC\} \rightarrow SG$: security controls are connected to security goals using the average function. Therefore the satisfaction level of the different goals depend on the average of the strength of the security controls connected.
  In the case of study, the satisfaction level of "authorization" depends on the average of the strength of "PIN, Finger, Iris".

- $\{As\}, \{SC\} \rightarrow SG$: assets and security controls are aggregated together using the minimum function. The value of security goals depends on their criticality, that is calculated based on the value of the assets, and on their satisfaction, that calculated based on the set of security controls active.
  In the case of study, the satisfaction level of "authentication" depends on the strength of the security controls activated ("Revoke [P , T1-001]", 'Revoke [Pd , T1-001]", "Revoke [S , T1-001]") and on the value of the asset "office"

- $\{SC\} \rightarrow V$: since the weights of the contributions between the security controls and the vulnerabilities are negative the aggregation function selected is 1+WeightedAverage (Note that the WeightedAverage is negative). This means that there is an inverse proportionality between the security controls and the vulnerabilities, as the security controls mitigate the vulnerabilities.
  In the case study, the vulnerability "intruder can access to the office" depends on the strength of the security controls selected by the system, more precisely on the strength of authorization ("Revoke [P ,

T1-001]", 'Revoke [P , T1-001]", "Revoke [S , T1-001]") and authentication ("SIM", "Finger", "PIN") security controls.

- $\{SC\} \rightarrow NFR$ *and* $\{SC\} \rightarrow FR$: since the weights between the security controls and the NFR/FR are negative, the aggregation function selected is 1+WeightedAverage (Note that the WeightedAverage is negative). This means that there is an inverse proportionality between the security controls and the NFR/FR, as the security controls negatively impact on NFR/FR.
  In the case of study, the non functional requirement "usability" depends on the strength of the security controls selected by the system, more precisely on the strength of authentication security controls ("SIM", "Finger", "PIN").

- $\{As\} \rightarrow PR$: assets connected to a partial risk are aggregated together using the maximum function, since the risk of an attack depends on the maximum value of the assets that can be harmed.

- $\{At\} \rightarrow PR$: the value of each attack is connected to its partial risk and no aggregation function is necessary.

- $\{As\}, At \rightarrow PR$: risk is usually computed by multiplying the probability of an attack and the related losses. Therefore, to evaluate partial risk PR, the loss factor (i.e., the contribution coming from the harmed assets) should be multiplied by the probability of an attack, which is translated into the minimum function.

- $\{PR\} \rightarrow TR$: for aggregating partial risk to total risk, the maximum function is selected. This function considers the maximum risk of attacks that may harm assets.

- $TR \rightarrow U$: we use the weighted average function to calculate the impact of the total risk on the utility. This impact is negative as the risk negatively affects the evaluation of the utility of security controls since there is an inverse proportion be represent the link between these two elements.

- $\{SG\} \rightarrow U$: utility aggregate security goals using the weighted average to consider the impact (weight) of each security goals on the utility. The contribution is positive since the satisfaction of security goals represents a benefit for the configuration of security controls selected.
  In the case of study, authentication, authorization and accountability positively impact on the utility of the security control configuration.

- $\{NFR\} \rightarrow U$ and $\{FR\} \rightarrow U$: utility aggregates non-functional requirements using the weighted average to consider the impact (weight) of each non-functional requirements on the utility.
- $TR, \{SG\}, \{NFR\}, \{FR\} \rightarrow U$: the sum function is selected for the utility node, since it is not a fuzzy variable and accumulate the value of risk, FR, NFR, and security goals.

6.2 Analysis conducted

In this section are presented a set of scenarios that explain the advantages of using the approach described in developing an adaptive access control system. This system is compared to an hypothetical non-adaptive access control system. The scenarios presented use three different employees (Claudio, Liliana and Bashar) who have the role of student, post-doc and professor, respectively, an information asset (project/patent/proposal) and a physical asset (computer). The system regulates the access to a particular office (T1-001) of the building.

6.2.1 Adding new assets

Access control policies and lists might need to be updated when a new asset is moved inside the building, offices and areas. In the following, it is presented a motivating scenario that highlights the advantages of using a dynamic access control system when an asset is moved/added inside the building.
Bashar (professor) is working on his proposal (Security level: Confidential) in his office T1-001. Since the value of the information inside the office is "medium", post-docs can enter in the office while the access at the students is revoked (Initial state).
On the 15th of April 2012 a new expensive computer (value: H high) is placed inside the office and configured (Event).

Dynamic approach
When the new device is brought into the office T1-001, the security requirements of the system change and the risk of harm increases. To protect the system and satisfy security requirements, access control policies are reviewed and updated accordingly. The system temporarily tolerate the violation of the functional requirements of some employees, since protecting the asset is more important. The access is granted only to the professors (Final state).

Static approach

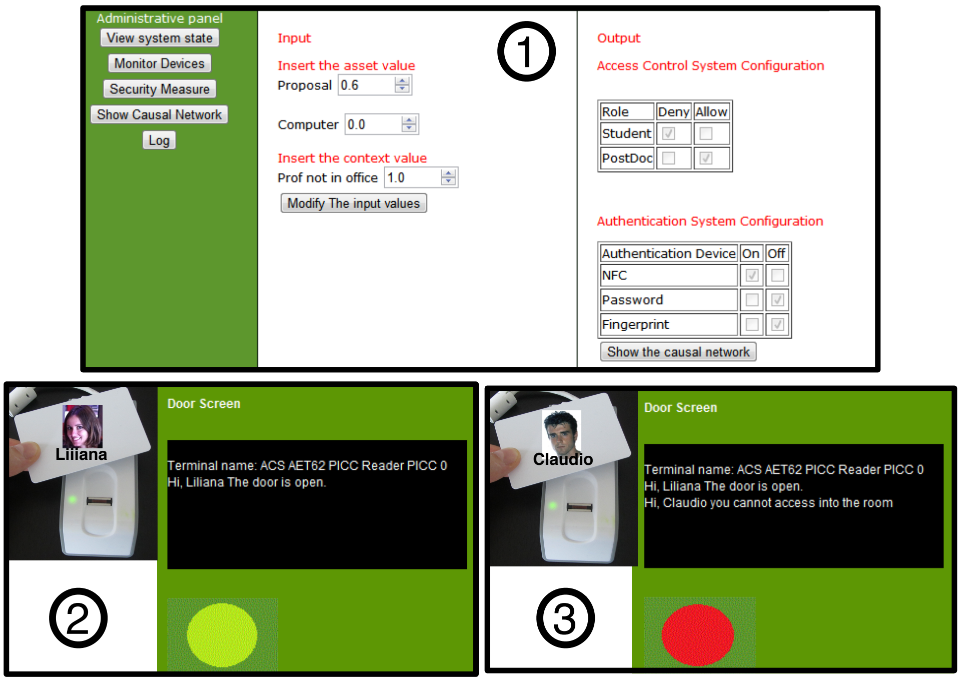
The system does not monitor the value of the assets present in the building. The access control policies related to the office T1-001 are not modified: post-doc can still access to the office. Potential attacker can enter and damage the computer newly added (Final state). The set of security controls to protect the system must be manually changed.

Table 6.3 represents the initial and final values of the assets to be protected, the context in which the application is working and the set of security controls applied to protect the system (with the best utility).

		Initial State	**Final State**
Assets	Proposal	C	C
	Computer	No	*H*
Context	Professor in the office	No	No
Security controls	Professors	Granted	Granted
	Post-Docs	Granted	*Revoked*
	Students	Revoked	Revoked
	NFC	ON	ON
	Password	OFF	OFF
	Fingerprint	OFF	OFF

Table 6.3: Authorization and authentication security controls when a new asset is placed in the office

In Figure 6.7 and 6.8 it is shown the output of the the door screen (managed by the security control manager) and the dashboard of SecuriTAS for this scenario. In this case, when Liliana tries to enter in the office, the access is granted (see door screen in Figure 6.7 (2)). Instead, when Claudio tries to access to the office, the access control system does not allow him to enter (see Figure 6.7 (3)). When the computer is put in the office, since its corresponding smart card is swiped on the NFC reader (see Figure 6.8 (2)). The Adaptation Manager is notified of this change and updates the state of the system, by setting the value of the Computer to 0.6 (see Input view in Figure 6.8 (1)). The Adaptation Manager recomputes the best configuration of security controls that should be applied on the system. In this case, it selects a single factor authentication (via smart card) and it revokes the permission to access to the office to both post-docs and students (see Output view in Figure 6.8 (1)). When both Liliana and Claudio try to access to the office, the access control system does not allow them to enter (see Figure 6.8 (3)).



Input

Insert the asset value

Proposal 0.6

Computer 0.0

Insert the context value

Prof not in office 1.0

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc		✓

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

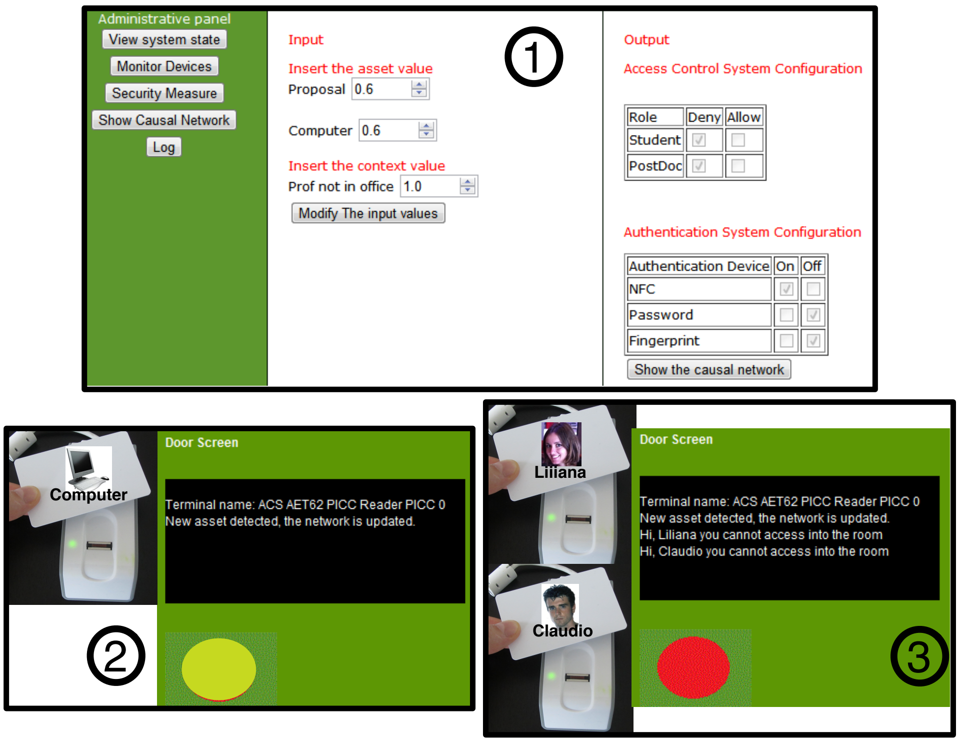
Door Screen

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
```

Door Screen

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
Hi, Claudio you cannot access into the room
```

Figure 6.7: Scenario 1. Initial state



Input

Insert the asset value

Proposal 0.6

Computer 0.6

Insert the context value

Prof not in office 1.0

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc	✓	

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

Door Screen

```
Terminal name: ACS AET62 PICC Reader PICC 0
New asset detected, the network is updated.
```

Door Screen

```
Terminal name: ACS AET62 PICC Reader PICC 0
New asset detected, the network is updated.
Hi, Liliana you cannot access into the room
Hi, Claudio you cannot access into the room
```

Figure 6.8: Scenario 1. Final state

6.2.2 Changing the value of existing assets

The value of the assets in the T1-001 office may change over time. For example, some documents might be only valuable for a time span (e.g., for contract tendering) and later their values might decrease. In the following, it is presented a motivating scenario that highlights the advantages of using an adaptive access control system when the value of an asset changes over time.

Every day, professors have to classify their current research by associating a security code to each paper, proposal, document on which they are working. Bashar (professor) is working on his proposal (Security level: Unclassified) in his office T1-001. Since the value of the information inside the office is low, post-docs and students can enter in the office (Initial state).
On the 30th of April 2012 Bashar finishes his proposal and understands the potential impact of his idea. Using the ICSRC asset management software, he changes the security level of his proposal to Secret (Event).

Dynamic approach
The ICSRC asset management software is connected to SecuriTAS. The Adaptation Manager is notified of this change and updates the state of the system. The system revokes the access to the students (Final state).

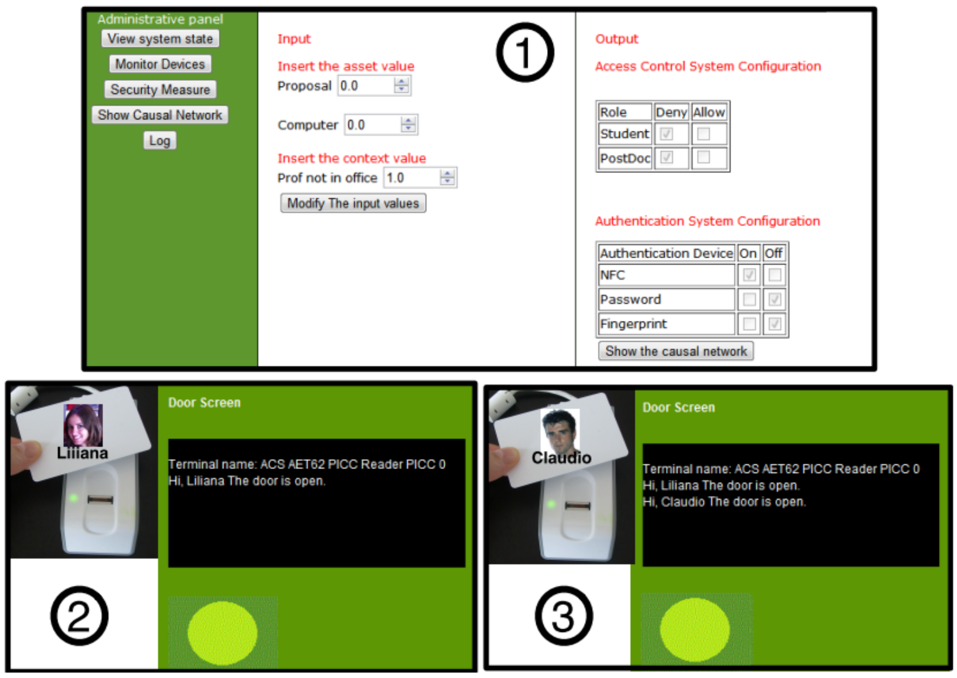
Not dynamic approach
The system does not monitor the assets value. Potential attackers can enter in the office and steal the idea of Bashar. It is necessary to manually revoke the permissions to the students otherwise, the confidentiality of the information is not guaranteed (Final state).

Table 6.4 represents the initial and final values of the assets to be protected, the context in which the application is working and the set of security controls applied to protect the system (with the best utility).

In Figure 6.9 and 6.10 it is shown the output of the the door screen (managed by the security control manager) and the dashboard of SecuriTAS for this scenario. In this case, when Liliana (post-doc) or Claudio (student) tries to enter in the office, the access is granted (see door screen in Figure 6.9 (2-3)). When the value of the proposal changes the Adaptation Manager is notified of this change and updates the state of the system, by setting the value of the Proposal to 0.6 (see Input view in Figure 6.10 (1)). The Adaptation Manager recomputes the best configuration of security controls that should be applied on the system. In this case, it selects a single factor

		Initial State	**Final State**
Assets	Proposal	S	*S*
	Computer	No	No
Context	Professor in the office	No	No
Security controls	Professors	Granted	Granted
	Post-Docs	Granted	Granted
	Students	Granted	*Revoked*
	NFC	OFF	*ON*
	Password	OFF	OFF
	Fingerprint	OFF	OFF

Table 6.4: Authorization and authentication security controls when the values of existing assets changes



Input

Insert the asset value

Proposal 0.0

Computer 0.0

Insert the context value

Prof not in office 1.0

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc	✓	

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

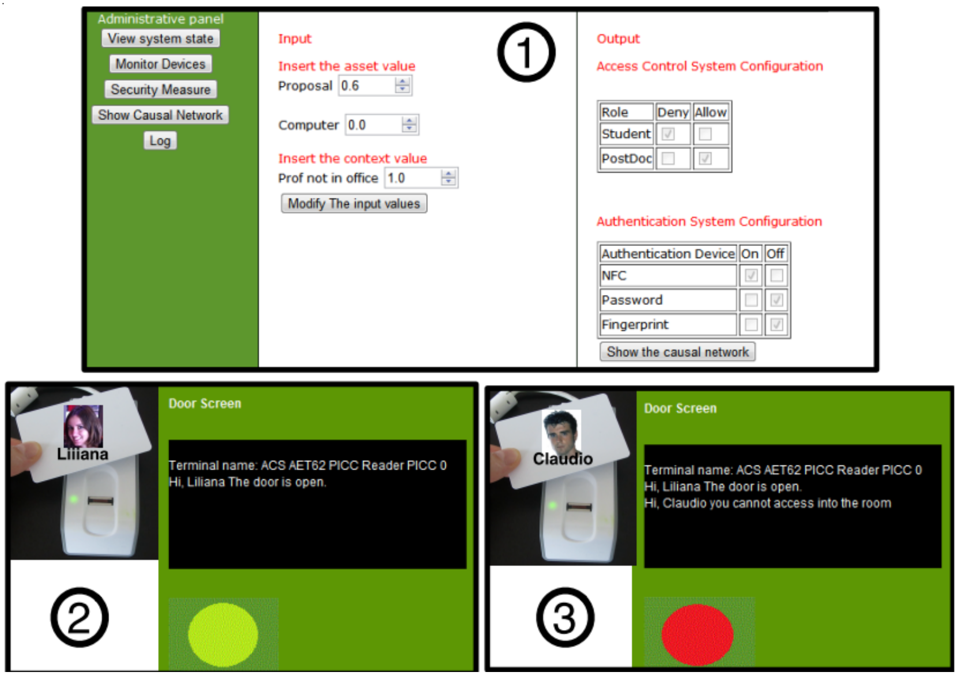
Liliana

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
```

Claudio

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
Hi, Claudio The door is open.
```

Figure 6.9: Scenario 2. Initial state



Input

Insert the asset value

Proposal 0.6

Computer 0.0

Insert the context value

Prof not in office 1.0

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc		✓

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
```

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
Hi, Claudio you cannot access into the room
```

Figure 6.10: Scenario 2. Final state

authentication (via smart card) and it revokes the permission to access to the office students (see Output view in Figure 6.10 (1)). When Claudio tries to access to the office, the access control system does not allow him to enter (see Figure 6.10 (3)).

6.2.3 Protect assets in a changing environment

In this scenario, changes in the environmental conditions can cause a re-configuration of the access controls applied in the system.
In this case, since the value of the assets contained in the office is high (S: Secret, H: High), post-docs and students cannot enter in the office (Initial State).
At 8.00 the professor enter in the the office for lunch (Event).

Dynamic approach
At the beginning the system detects that the professor is not in the office and revokes the access to Liliana (Post-doc) and to Claudio (Student). In this way the security goals confidentiality and integrity are reached and the risks of attack are minimized. When Bashar (Professor) enters in the office, the access to post-docs and students is granted. Therefore, Liliana and Claudio

can enter in the office (Final state). The presence of Bashar guarantees that Liliana and Claudio do not access to any secret document available in the office.

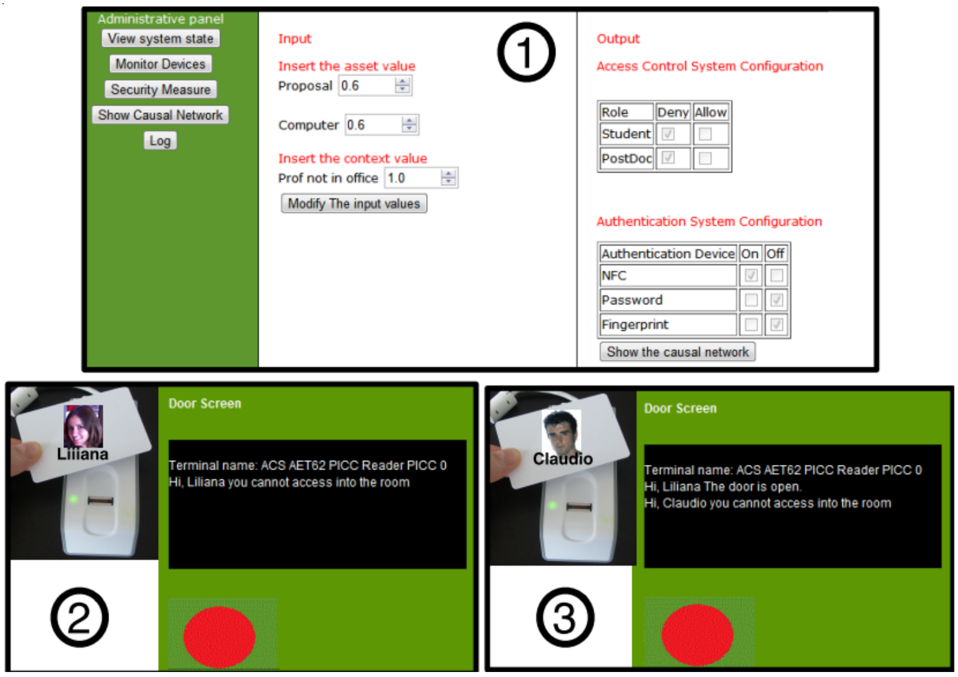
Static approach
Even if Bashar (Professor) is in his office, the system does not grant the access to Liliana (Post-doc). Therefore, Bashar and Liliana cannot work together (Final state).

Table 6.5 represents the initial and final values of the assets to be protected, the context in which the application is working and the set of security controls applied to protect the system (with the best utility).

		Initial State	**Final State**
Asset	Proposal	S	S
	Computer	H	H
Context	Professor in the office	No	Yes
Security controls	Professors	Granted	Granted
	Post-Docs	Revoked	*Granted*
	Master Students	Revoked	*Granted*
	NFC	ON	ON
	Password	OFF	OFF
	Fingerprint	OFF	OFF

Table 6.5: Authorization and authentication security controls when the professor enters in the office.

In Figure 6.11 and 6.12 it is shown the output of the the door screen (managed by the security control manager) and the dashboard of SecuriTAS for this scenario. In this case, when Liliana (post-doc) or Claudio (student) tries to enter in the office, the access is denied (see door screen in Figure 6.11 (2-3)) since the proposal is secret and the value of the computer is high. When Bashar accesses to the office by swiping his card (see Figure 6.12(2)). The Adaptation Manager is notified of this change and updates the state of the system, by setting the value of context condition "Professor is not in the office" to 0.0 (see Input view in Figure 6.12(1)). The Adaptation Manager recomputes the best configuration of security controls that should be applied on the system. In this case, it selects a single factor authentication (via smart card) and grants the permission to access to the office to both post-docs and students (see Output view in Figure 6.12(1)). When both Liliana and Claudio try to access to the office, the access control system



Input

Insert the asset value

Proposal 0.6

Computer 0.6

Insert the context value

Prof not in office 1.0

Output

Access Control System Configuration

Role	Deny	Allow
Student	✓	
PostDoc	✓	

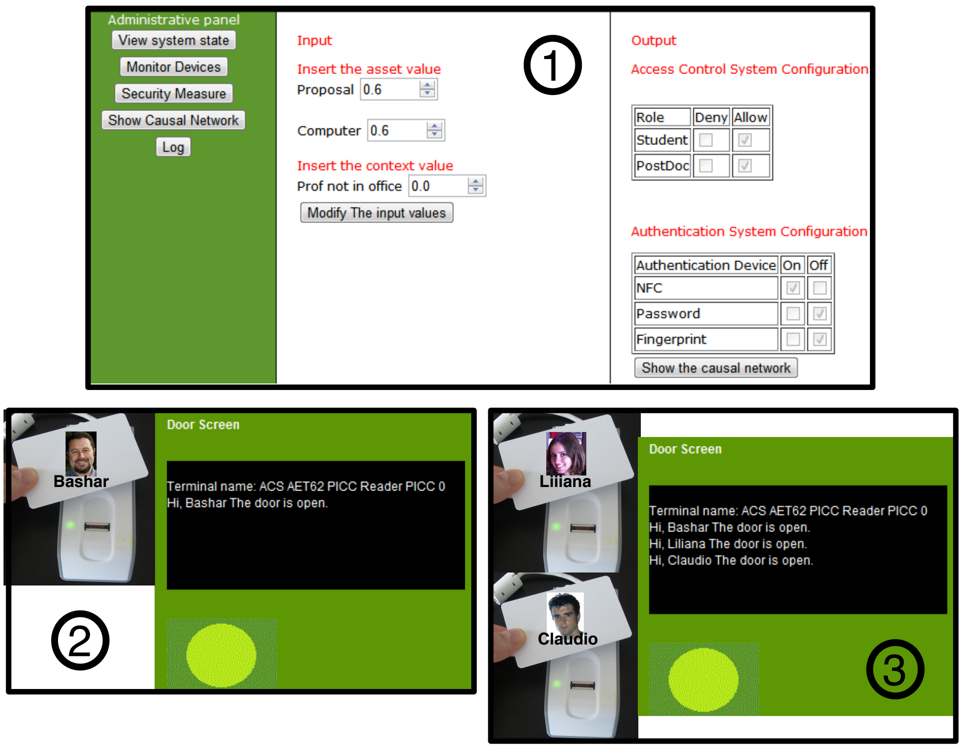
Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana you cannot access into the room
```

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Liliana The door is open.
Hi, Claudio you cannot access into the room
```

Figure 6.11: Scenario 3. Initial state



Input

Insert the asset value

Proposal 0.6

Computer 0.6

Insert the context value

Prof not in office 0.0

Output

Access Control System Configuration

Role	Deny	Allow
Student		✓
PostDoc		✓

Authentication System Configuration

Authentication Device	On	Off
NFC	✓	
Password		✓
Fingerprint		✓

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Bashar The door is open.
```

```
Terminal name: ACS AET62 PICC Reader PICC 0
Hi, Bashar The door is open.
Hi, Liliana The door is open.
Hi, Claudio The door is open.
```

Figure 6.12: Scenario 3. Final state

allows them to enter (see Figure 6.12(3)).

6.3 Discussion

This chapter evaluated the approach described in this thesis using the IC-SRC case study. First of all, it was designed the STrioM model of the case study. Second, this model was translated into a fuzzy causal network. Finally, the behavior of the causal network was evaluated using three different scenarios. For all of them, the behavior of the adaptive access control system designed, and an hypothetical non adaptive one, were compared.

The adaptive access control system was able to modify its behavior when the value of the assets to protect, and the context in which the application is working changed. More precisely, in the different scenarios, this system was able to grant and revoke permissions, finding the best compromise between maximize the satisfaction level of security goals, minimize the risk of attacks and satisfy functional and non-functional requirements. On the other hand, the non adaptive access control system leaved the system vulnerable to attacks (Scenario 1 and 2), while in other, it applied useless security controls (Scenario 3).

The number of security controls configuration tried for each scenario was 2^6, since only binary security controls were considered in this example.In the scenarios the fuzzy causal network converged in maximum 4 iterations, taking less than 250 ms.

Chapter 7

Conclusions

"Certainly it is permitted to anyone to put forward whatever hypotheses he wishes, and to develop the logical consequences contained in those hypotheses."

Giuseppe Peano

Access control systems are used to protect assets from harms. One of their main features is deciding who can interact with the different assets. On the one hand, access control systems have to protect assets against unauthorized modifications, accesses, and disclosures, while, on the other hand, assets availability must be guaranteed to the legitimate users. Existing access control systems are usually designed and implemented as static systems and do not support asset and context awareness.

Context may influence access control systems in different ways. When the context (i.e., time, users location, lighting, temperature) changes, different security concerns, such as requirements, vulnerabilities, tasks, threats, attacks may be activated or deactivated. For example, in a university, if a particular student has to "speak with a professor" (requirement), and "the professor [p] is in the office" (context) the system has to temporarily relax its security controls and grant the access to the student. In other words, considering context should improve the effectiveness of the access control system.

Assets play a central role in adaptive access control. While the system is executing assets may evolve dynamically: new assets can be added to the system, the value of existing assets can change, or existing assets may not be under protection anymore. Assets influence both security requirements and threats. When the value of an asset increases, different security requirements may be affected. Therefore, the security controls necessary to protect the system must be changed accordingly.

Several works have already considered requirements [LYM03, CIN05, FLLD01, MZ08], assets [Fir04, MSLPIORA11] and context [GI97, AS08, MMMA00, CLS$^+$01, ZP04, KBB$^+$03] in the design of access control systems. In this thesis we have presented an approach that combines all of the aforementioned into a unique and coherent framework. The proposed approach, based on a previous work by Salehie et. al. [MSLPIORA11], promotes assets and context as first-class entities in engineering secure software systems and is based on three different steps. During the first step, a model able to capture the interplay between the different concerns involved in a security problem, is designed (STrioM). This is composed of an asset, goal, and threat model. An asset model represents the assets to be protected and is related to the requirements of the system, expressed through a goal model, and the objectives of an attacker, expressed through a threat model. During the second step, the model is translated into a fuzzy causal network, that is used at to drive the adaptation process. In particular, at run-time the fuzzy causal network is used to select the best configuration of security controls to protect the system based on the assets that need to be protected. Even if context plays a central role in secure system, this approach does not consider how context changes influence the security controls that are applied in the system.

7.1 Contributions

This thesis has addressed a range of issues arising from requirements analysis, self-adaptive software and adaptive access control.

- The approach of Salehie et. al. [MSLPIORA11] is weaved with context. To this aim, first, STrioM is extended with an element that represents the context. In particular, is defined how this entity can be connected with the other security concerns. Second, the influence of context on the FCN definition is analyzed. STrioM is used as input to build the Fuzzy Causal Network (FCN). The elements and relationships identified in the asset, goal, and threat models are translated into nodes and links of the FCN. The thesis describes how context impact on the design of the FCN. Finally, it is analyzed how context changes impact on the run-time adaptation of security controls applied to protect assets.

- The approach previously described is customized for analyzing an adaptive access control problem. In particular, the thesis considers three security requirements related to access control problems presented by Samarati et. all. [SS94]: authentication, authorization and auditing. Authentication aims to verify the identity of a user, process, or device, before granting access to some critical assets of the system. Authorization is the process of granting rights to participants to perform an interaction, for instance, to access to an asset. Auditing is used to monitor the system. This thesis analyzes how the satisfaction of these security requirements can be influenced by the assets to protect and the context, through STrioM, and how the FCN can be used at run-time to select the set of security controls necessary to protect the system.

- The thesis presents SecuriTAS[1] a novel tool for engineering adaptive security. This tool guides the system designer in the development of adaptive security systems and, in particular, adaptive access control systems, from requirements modeling to system execution. SecuriTAS can analyze the impact of changes in assets and context on security concerns, and identify an appropriate set of security controls necessary to protect the system. Assets and context states are detected using suitable monitors, while security controls are applied using effectors. It also provides a dashboard to load and visualize the FCN, update the values of assets and context conditions that may affect vulnerabilities, and visualize the current state of the system.

- The approach is evaluated through a case study. The problem presented concerns with the regulation of the access into a particular office of IRSRC[2]. Different users, such as students, professors, post-docs, may want to access to each office. The system has to select at run-time which users can enter based on the value of the assets present inside the office and on the context in which the system is working. To do so, a STrioM model for this case study is designed and the corresponding causal network is inferred. Then the FCN is used in a little demo where SecuriTAS uses the AET 62 NFC Reader to authenticate the different users, to monitor the assets that are added to or removed from the office and the context in which the application is working (people inside the office). Based on these elements SecuriTAS selects best configuration of security controls to be applied on the system.

[1]The tool is available to download from http://securitas.googlecode.com/files/Securitas.

[2]Irish Computer Science Research Centre.

7.2 Future Work

This thesis can be extended in several ways:

- Before using the STrioM model in the runtime adaptation, it is necessary to translate the model into a fuzzy causal network (FCN), that must tuned. Two steps are iteratively executed to tune the network: configuring the aggregation functions, and initialization and tuning. In the first step, the aggregation functions, used to aggregate the contributions of the incoming links of a node and calculate its value, are selected. In the second step, the initial values of the nodes and weights of causal links are initialized. Since, this process is still a manual activity, it is time expensive and also error prone. A learning mechanism that helps the system designer in tuning the network in an automatic/semi-automatic way can be further explored.

- The thesis evaluates the presented approach using a simple case study. Our simulation demonstrates that risk assessment and utility evaluation are plausible in the different operating contexts of the application (risk increases when assets' values increase and, in those cases, stronger security controls are applied). However, a more complex case study can be considered to evaluate the proposed approach. For example, it is interesting to consider cases in which it is necessary to regulate the access on different offices of the same area, to analyze how topological aspects can influence the approach presented.

- This thesis considers that the goal and the threat models are produced at design time and do not undergo changes during the system runtime. In real world scenarios, though, this assumption may reveal to be overly conservative, as both the application requirements and the security requirements may evolve during the system life time. Consequently a possible extension may consider those changes in these models.

- The tool proposed in this thesis is still a prototype. It is still not clear with the size of the case study performed, how this framework will scale-up to industrial size projects. Therefore, since the approach described uses a global search method to find the optimal configuration of security controls, other search algorithms, possibly heuristic based, may be investigated to find a nearly optimal solution.

Bibliography

[924] ISO/IEC 9241-11:1998. Part 11: Guidance on usability.

[ADB$^+$99] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag.

[ADG09] R. Ali, F. Dalpiaz, and P. Giorgini. Goal-based self-contextualization. Amsterdam, The Netherlands, 08/06/2009 2009.

[ADG10a] R. Ali, F. Dalpiaz, and P. Giorgini. A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 15:20, 2010.

[ADG10b] Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini. Contextual goal model. Technical report, Ingegneria e Scienza dell'Informazione, University of Trento DISI-10-020, 2010.

[Ali10] Raian Ali. *Modeling and Reasoning about Contextual Requirements: Goal-based Framework.* PhD thesis, 2010.

[All01] Julia H. Allen. *The CERT Guide to System and Network Security Practices.* Addison-Wesley, first edition, 2001.

[And01] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001.

[Ant96] A.I. Anton. Goal-based requirements analysis. In *Requirements Engineering, 1996., Proceedings of the Second International Conference on*, pages 136 –144, apr 1996.

[AS08] Elisa Bertino Arjmand Samuel, Arif Ghafoor. A framework for specification and verification of generalized spatio-temporal role based access control model in 47907-2086. Technical report, Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette, 2007-08.

[BASD06] D. Byers, S. Ardi, N. Shahmehri, and C. Duma. Modeling software vulnerabilities with vulnerability cause graphs. In *Software Maintenance, 2006. ICSM '06. 22nd IEEE International Conference on*, pages 411 –422, sept. 2006.

[BGR11] Lujo Bauer, Scott Garriss, and Michael K. Reiter. Detecting and resolving policy misconfigurations in access-control systems. *ACM Trans. Inf. Syst. Secur.*, 14:2:1–2:28, June 2011.

[Bis04] Matt Bishop. *Introduction to Computer Security.* Addison-Wesley, first edition, 2004.

[BK86] Bart and Kosko. fuzzy cognitive maps. *International Journal of Man-Machine Studies*, 24(1):65 – 75, 1986.

[B.N08] C. Haley R. Laney J. Moffet B.Nuseibeh. Security requirements engineering: A framework for representation and analysis. Technical report, IEEE Transaction on Software Engineering, vol 34, pp. 133-153, 2008.

[BPG$^+$04] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236, 2004.

[CCB08a] Frédéric Cuppens and Nora Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security*, 7:285–305, 2008. 10.1007/s10207-007-0051-9.

[CCB08b] Frédéric Cuppend Cuppens and Nora Cuppens-Boulahia. Modeling contextual security policies. *Int. J. Inf. Secur.*, 7(4):285–305, July 2008.

[CCGR00] Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. Nusmv: a new symbolic model checker. *International Journal on Software Tools for Technology Transfer (STTT)*, 2:410–425, 2000. 10.1007/s100090050046.

[CF05] Howard Chivers and Martyn Fletcher. Applying security design analysis to a service-based system: Research articles. *Softw. Pract. Exper.*, 35(9):873–897, July 2005.

[CFZA02] M.J. Covington, P. Fogla, Zhiyuan Zhan, and M. Ahamad. A context-aware security architecture for emerging applications. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 249 – 258, 2002.

[CIN05] Robert Crook, Darrel Ince, and Bashar Nuseibeh. On modelling access policies: Relating roles to their organisational context. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, pages 157–166, Washington, DC, USA, 2005. IEEE Computer Society.

[CKM01] Jaelson Castro, Manuel Kolp, and John Mylopoulos. A requirements-driven development methodology. In *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*, CAiSE '01, pages 108–123, London, UK, UK, 2001. Springer-Verlag.

[CLS$^+$01] Michael J. Covington, Wende Long, Srividhya Srinivasan, Anind K. Dev, Mustaque Ahamad, and Gregory D. Abowd. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, SACMAT '01, pages 10–20, New York, NY, USA, 2001. ACM.

[CPP02] Shari Lawrence Pfleeger Charles P. Pfleeger. *Security in Computing.* Prentice Hall, third edition, 2002.

[CS08] Qingtang Lui Chengling Zhao Chaowang Shang, Zongkai Yang. A context based dynamic access control model for web service. Technical report, Research Center for Education Information on Technology Central China Normal University, 2008.

[DBSL02] N. Damianou, A. Bandara, M. Sloman, and E. Lupu. A Survey of Policy Specification Approaches, 2002.

[DCdVPS03] Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Access control: principles and solutions. *Software: Practice and Experience*, 33(5):397–421, 2003.

[Dey01] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5:4–7, 2001. 10.1007/s007790170019.

[DF03] Ramaswamy Chandramouli David Ferraiolo, D. Richard Kuhn. *Role-based access control.* First edition, 2003.

[doi09] Book reviews. *Journal of the American Statistical Association*, 104(485):409–426, 2009.

[DR77] K. E. Shoman D.T. Ross. Structured analysis for requirements definition. Technical report, IEEE Transactions on Software Engineering Vol.3 No.1 P.g 6-15, 1977.

[dV11] S. De Capitani di Vimercati. *Access control policies, models, and mechanisms.* Springer, Berlin, 2011.

[DvLF93] Anne Dardenne, Axel van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. In *SCIENCE OF COMPUTER PROGRAMMING*, pages 3–50, 1993.

[Edw11] Rodger Edwards. Intelligent buildings and building automation. *Construction Management and Economics*, 29(2):216–217, 2011.

[EYZ10] Golnaz Elahi, Eric Yu, and Nicola Zannone. A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. *Requir. Eng.*, 15:41–62, March 2010.

[Fir04] D. Firesmith. Specifying reusable security requirements. *Journal of Object Technology*, 3(1):61–75, 2004.

[FLLD01] Pierre-Jean Fontaine, Axel Van Lamsweerde, Emmanuel Letier, and Robert Darimont. Goal-oriented elaboration of security requirements, 2001.

[FP09] C. Feltus and M. Petit. Building a responsibility model including accountability, capability and commitment. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, pages 412 –419, march 2009.

[FPMT01] A. Fuxman, M. Pistore, J. Mylopoulos, and P. Traverso. Model checking early requirements specifications in tropos. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 174 –181, 2001.

[GD07] Tyrone Grandison and John Davis. The impact of industry constraints on model-driven data disclosure controls, 2007.

[GE07a] Eric Yu Golnaz Elahi. A goal oriented approach for modeling and analyzing security trade-offs. Technical report, University of Toronto, Canada, 2007.

[GE07b] Ravi Sandhu Golnaz Elahi, Eric Yu. A goal oriented approach for modeling and analyzing security tradeoffs. Technical report, 2007.

[GHJP00] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. Mobile ip authentication, authorization, and accounting requirements. Technical report, United States, 2000.

[GI97] Luigi Giuri and Pietro Iglio. Role templates for content-based access control. In *Proceedings of the second ACM workshop on Role-based access control*, RBAC '97, pages 153–159, New York, NY, USA, 1997. ACM.

[Gli07] M. Glinz. On non-functional requirements. In *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International*, pages 21 –26, oct. 2007.

[GS10] A.Feringa G. Stoneburner, A. Goguen. Risk management guide for information technology systems. Technical report, NIST Special pubblication, vol 800, p.30,2002, 2010.

[GSF02] A. Goguen G. Stoneburner and A. Feringa. Risk management guide for information technology systems. *NIST special publication*, 800:30, 2002.

[HA09] Qingfeng He and Annie I. Antón. Requirements-based access control analysis and policy specification (recaps).

Information and Software Technology, 51(6):993 – 1009, 2009.

[HLMN08] Charles B. Haley, Robin Laney, Jonathan D. Moffett, and Bashar Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 34(1):2008, 2008.

[HS06] M.G. Hinchey and R. Sterritt. Self-managing software. *Computer*, 39(2):107 – 109, feb. 2006.

[HSB+05] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma. Context sensitive access control. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, pages 111–119, New York, NY, USA, 2005. ACM.

[Huf90] Anne S. Huff, editor. *Mapping strategic thought.* John Wiley & Sons, New York, 1990.

[IEE01] *American National Standard for Telecommunications: Telecom Glossary 2000.* American National Standard T1.523-2001, American National Standard Institute,, 2001.

[INF] *Information technology Logical Access Control.* Virginia Information Technologies Agency (VITA), 04/18/2007.

[Jac02] Daniel Jackson. Alloy: a lightweight object modelling notation. *ACM Trans. Softw. Eng. Methodol.*, 11(2):256–290, April 2002.

[JT08] M.G. Jaatun and I.A. Tndel. Covering your assets in software engineering. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 1172 –1179, march 2008.

[KBB+03] A.A.E. Kalam, R.E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 120 – 131, june 2003.

[Kos86] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, (24):65–75, 1986.

[Kos88] Bart Kosko. Hidden patterns in combined and adaptive knowledge networks. *International Journal of Approximate Reasoning*, 2(4):377 – 393, 1988.

[Kra99] Micki Krause. *Handbook of Information Security Management 1999.* CRC Press, Inc., Boca Raton, FL, USA, 99th edition, 1999.

[Lam09] A.V. Lamsweerde. Requirements engineering: from system goals to uml models to software specifications. *Change*, 2009.

[LK95] Pericles Loucopoulos and Vassilios Karakostas. *System Requirements Engineering.* McGraw-Hill, Inc., New York, NY, USA, 1995.

[LPN12] Mazeiar Salehie Luca Cavallaro Inah Omoronyia Liliana Pasquale, Claudio Menghi and Bashar Nuseibeh. Securitas: A tool for engineering adaptive security. 2012.

[LYM03] Lin Liu, Eric Yu, and John Mylopoulos. Security and privacy requirements analysis within a social setting. In *Proceedings of the 11th IEEE International Conference on Requirements Engineering*, RE '03, pages 151–, Washington, DC, USA, 2003. IEEE Computer Society.

[Mar] Leo Marcus. Introduction to logical foundations of an adaptive security infrastructure *.

[MC11] R. Muley and M. Chatterjee. Improved dynamic model for policy based access control. In *Proceedings of the International Conference &#38; Workshop on Emerging Trends in Technology*, ICWET '11, pages 781–784, New York, NY, USA, 2011. ACM.

[McA05] Shawna McAlearney. Convergence of physical and logical access control systems @ONLINE. January 2005.

[MJ10] Khaled Ghedira Meriam Jemel, Nadia Ben Azzouna. Towards a dynamic access control model for egoverment web service. Technical report, Higher School of technology and

Computer of Tunis, National School of Computer Science Manouba, Higer Institute of Management of Tunis, 2010.

[MJaR$^+$10] M.V. Martin, K. Jo´ andhannsdottir, G.B. Reynaga, J. Tashiro, and M.A. Garcia-Ruiz. Unconscious mind: Authenticating with something you don't know? or just an infallible liveness test? In *Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference on*, pages 1 –6, may 2010.

[MLY05] Suvda Myagmar, Adam J. Lee, and William Yurcik. Threat modeling as a basis for security requirements. In *In Symposium on Requirements Engineering for Information Security (SREIS*, 2005.

[MMMA00] Michael Covington Matthew, Ý Matthew, J. Moyer, and Mustaque Ahamad. Generalized role-based access control for securing future applications, 2000.

[Mof99] Jonathan D. Moffett. Requirements and policies. HO- Laboratories, Bristol, UK, 1999.

[MS88] J.D. Moffett and M.S. Sloman. The source of authority for commercial access control. *Computer*, 21(2):59 –69, feb. 1988.

[MSLPIORA11] B. Nuseibeh M. Salehie L. Pasquale I. Omoronyia R. Ali. Requirements-driven adaptive security: Protecting variable assets at runtime. Technical report, LERO- Irish Software Engineering Research Centre, Limerick, Ireland, 2011.

[MZ08] Fabio Massacci and Nicola Zannone. A model-driven approach for the specification and analysis of access control policies. In *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems*, OTM '08, pages 1087–1103, Berlin, Heidelberg, 2008. Springer-Verlag.

[NE00] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA, 2000. ACM.

[OGT$^+$99] P. Oreizy, M.M. Gorlick, R.N. Taylor, D. Heimhigner, G. Johnson, N. Medvidovic, A. Quilici, D.S. Rosenblum, and A.L. Wolf. An architecture-based approach to self-adaptive software. *Intelligent Systems and their Applications, IEEE*, 14(3):54 –62, may/jun 1999.

[ora] Oracle: Using the database resource manager.

[PCoPfIS99] 1999 Part 1: Code of Practice for Information Security, London. Information security management. Technical report, British Standards Institution, BS799-1:1999.

[RA10] Paolo Giorgini Raian Ali, Fabiano Dalpiaz. Contextual goal models. Technical report, University of Trento, 2010.

[RSA98] C. Rolland, C. Souveyet, and C.B. Achour. Guiding goal modeling using scenarios. *Software Engineering, IEEE Transactions on*, 24(12):1055 –1071, dec 1998.

[SAW94a] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, pages 85 –90, dec 1994.

[SAW94b] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85 –90, dec. 1994.

[SHF$^+$01] Gary. Stoneburner, Clark. Hayden, Alexis. Feringa, National Institute of Standards, and Technology (U.S.). *Engineering principles for information technology security (a baseline for achieving security) [microform] : recommendations of the National Institute of Standards and Technology / Gary Stoneburner, Clark Hayden, and Alexis Feringa.* U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology ; For sale by the Supt. of Docs., U.S. G.P.O., Gaithersburg, Md. : Washington, D.C. :, 2001.

[SL07] V. Sadana and Xiaoqing Frank Liu. Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. In *Com-*

puter Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, volume 1, pages 215 –218, july 2007.

[spr]

[SR00] Kuhn D.R Sandhu R, Ferraiolo D.F. The nist model for role-based access control: Toward a unified standard. Technical report, National institute of stadard and tecnology, 2000.

[SS75] J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278 – 1308, sept. 1975.

[SS94] R.S. Sandhu and P. Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40 –48, sept. 1994.

[SS97] Ian Sommerville and Pete Sawyer. *Requirements Engineering: A Good Practice Guide.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

[SSS94] Ravi Sandhu, Ravi S. S, and Pierangela Samarati. Access control: Principles and practice. 1994.

[ST94] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5):22 –32, sept.-oct. 1994.

[ST09] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4:14:1–14:42, May 2009.

[SV01] Pierangela Samarati and Sabrina De Capitani di Vimercati. Access control: Policies, models, and mechanisms. In *Revised versions of lectures given during the IFIP WG 1.7 International School on Foundations of Security Analysis and Design on Foundations of Security Analysis and Design: Tutorial Lectures*, FOSAD '00, pages 137–196, London, UK, 2001. Springer-Verlag.

[TAPH05] William Tolone, Gail-Joon Ahn, Tanusree Pai, and Seng-Phil Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37(1):29–41, March 2005.

[TO09] Nobukazu Yoshioka Takao Okubo, Kenji Taguchi. Misuse-cases asset security goals. Technical report, Secure Computing Laboratory, Fujtsu Laboratories Limited, 2009.

[Tom97] David Toman. Memoing evaluation for constraint extensions of datalog. *Constraints*, 2:337–359, 1997. 10.1023/A:1009799613661.

[TS94] R.K. Thomas and R.S. Sandhu. Conceptual foundations for a model of task-based authorizations. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 66 –79, jun 1994.

[Ulf82] JD Ulfman. *Principles o˜ Database Systems.* Computer Science Press, London, 1982.

[Vie05] John Viega. Building security requirements with clasp. *SIGSOFT Softw. Eng. Notes*, 30:1–7, May 2005.

[vL01] A. van Lamsweerde. Goal-oriented requirements engineering: a guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 249 –262, 2001.

[VL04a] A. Van Lamsweerde. Elaborating security requirements by construction of intentional anti-models. In *Proceedings of the 26th International Conference on Software Engineering*, pages 148–157. IEEE Computer Society, 2004.

[vL04b] Axel van Lamsweerde. Elaborating security requirements by construction of intentional anti-models. In *Proceedings of the 26th International Conference on Software Engineering*, ICSE '04, pages 148–157, Washington, DC, USA, 2004. IEEE Computer Society.

[WCSL98] Thomas Y.C. Woo, Thomas Y. C, Woo Simon, and Simon S. Lam. Designing a distributed authorization service, 1998.

[Wie03] Karl Eugene Wiegers. *Software Requirements.* Microsoft Press, Redmond, WA, USA, 2 edition, 2003.

[WJ] Ana Cavalli Willy Jimenez, Amel Mammar. Software vulnerabilities, prevention and detection methods a review.

Telecom SudParis, 9, Rue Charles Fourier 9100 Evry, France.

[WMM08] Christian Wolter, Michael Menzel, and Christoph Meinel. Modelling security goals in business processes. In *Modellierung'08*, pages 197–212, 2008.

[Y.05] Wood D. Norton D. Weschler P. Ferris C. Wilson Y. Single sign-on framework with trust-level mapping to authentication requirements. Technical Report 6892307, May 2005.

[YLM04] Yijun Yu, J.C.S.P. Leite, and J. Mylopoulos. From goals to aspects: discovering aspects from requirements goal models. In *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*, pages 38 – 47, sept. 2004.

[YM93] Eric S. K. Yu and John Mylopoulos. An actor dependency model of organizational work: with application to business process reengineering. In *Proceedings of the conference on Organizational computing systems*, COCS '93, pages 258–268, New York, NY, USA, 1993. ACM.

[YM98] Eric Yu and John Mylopoulos. Why Goal-Oriented requirements engineering. 1998.

[Yu09] Eric Yu. Social modeling and i*. In Alexander Borgida, Vinay Chaudhri, Paolo Giorgini, and Eric Yu, editors, *Conceptual Modeling: Foundations and Applications*, volume 5600 of *Lecture Notes in Computer Science*, pages 99–121. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-02463-4-7.

[Yue87] Yue. What Does It Mean to Say that a Specification is Complete? In *Fourth International Workshop on Software Specification and Design (IWSSD-4), Monterey, USA*, 1987.

[Zav97] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29:315–321, December 1997.

[ZLZ06] Sanming Zhou, Zhi-Qiang Liu, and Jian Ying Zhang. Fuzzy causal networks: general model, inference, and convergence. *Fuzzy Systems, IEEE Transactions on*, 14(3):412 – 420, june 2006.

[ZP04] Guangsen Zhang and Manish Parashar. Context-aware dynamic access control for pervasive applications, 2004.