A trust framework for multi-organization environments

Khalifa Toumi

DOCTORAT EN CO-ACCREDITATION
TELECOM SUDPARIS ET L'UNIVERSITE EVRY VAL D'ESSONNE

Spécialité : Informatique

Ecole doctorale : Sciences et Ingénierie

Présentée par
Khalifa TOUMI

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

A Trust Framework For Multi-Organization Environments

Thèse dirigée par Ana CAVALLI

Soutenue le 01 avril 2014 devant le jury composé de :

Directeur de thèse:	Ana Cavalli	Télécom SudParis- France
Co-encadrant:	César Andrès	Brain Tec- Suisse
Examinateurs :	Frédéric Cuppens	Télécom Bretagne- France
	Nora Cuppens - Boulahia	Télécom Bretagne, France
	Lidia Fuentes	University of Malaga, Spain
	Nina Yevtushenko	Tomsk State University- Russia
	Wissam Mallouli	Montimage-France
	Sammy Haddad	Oppida-France

Thèse N° 2014TELE0004

Contents

List of Figures

Acknowledgments

I would like to express my sincere thanks to Professor Ana Rosa Cavalli, my thesis supervisor at 'Telecom & Management SudParis' institute, who guided my researches and answered my questions. I would like to thank her for all her help, it was absolutely invaluable. I appreciate all the suggestions she made to me as well as the time she devoted to guide and advise me throughout this PhD work.

I also would like to thank Doctor César Andrés, for his precise explanations and criticism. His constructive remarks helped me to build my personal point of view about the security field.

My thanks are also expressed to all those who helped me with their experience and expertise and somehow offered suggestions for my work in particular Professor Nora Cuppens-Boulahia, Professor Frédéric Cuppens, Professor Stephane MAAG, Dr. Anis Laouiti, Mr. Edgardo Montesdeoca, Dr Wissam Mallouli, Dr. Bachar Wehbi and Dr. Mouna Ayari.

More thanks go to my thesis evaluation team composed of Professor Nora Cuppens - Boulahia, Professor Nina Yevtushenko, Professor Fréderic Cuppens, Professor Ana Rosa Cavalli, Mr Wissam Mallouli, Mr Sammy Haddad and Mr Cesar Andres.

Above all, many thanks to all my friends and colleagues who supported me during my PhD. They are, in no particular order, Soua[2], Imen, Khedher, Mehdi, Mazen, Farouk, Faycel, Slim, Haikel, Didi, Hamid, Emad, Soualah, Yassine, Abid, Malek, Amri, Sara, Slah, Brigitte, Xiaoping, Jorge, Jimmy, Mohamed, Marouen, Olga, Vinh, Natalia, Anderson, Pramila, Samiha, Sabir, Fabien, Mariam, Amira and Jeevan.

I hope they find here the expression of my deep gratitude and appreciation.

To my parents Azouz and Aidouda,
I am especially thankful for your love, your understanding and your continuous support. You gave me strengths on weak days and showed me the sun on rainy days. Thanks for always believing in me.

To my brother khaled and my sister Hana,
I am particularly indebted for your unconditional trust and sincere love. Thanks for always standing by my side during difficult times and for the fun moments I have shared with you !

In Honor of my grandfathers, grandmothers, my oncle Saadoun
I dedicate this work as a token of my deep love. May ALLAH forgive You and grant You His Grace and His Mercy.

To all TOUMI family members,
Thanks for your love, kind support and continuous encouragement !

To Hela
Thanks for everything :)

Khalifa Toumi

Abstract

Abstract

The widespread of inexpensive communication technologies, distributed data storage and web services mechanisms currently urge the collaboration among organizations. Partners are participating in this environment motivated by several advantages such as: (1) the ability to use external and professional resources, services and knowledge, (2) the reduction of time-consuming requirements and (3) the benefaction of experts experience.

However, this collaboration is not perfect since several problems can arise such as the misuse of resources, disclosure of data or inadequate services. Therefore, security is an important concern of the participants. In particular trust management and access control are one of the major security issues for an organization. This thesis addresses these two areas in particular. It proposes a novel and comprehensive trust framework for Multi-Organization Environments.

Our approach is organized in four parts. First, we propose a vector based model approach for defining trust vectors. These vectors evaluate a set of requirements, under conditions, and provide a degree of confidence. In our approach, we consider two different types of vectors. On the one hand, a vector that links a user to an organization and, on the other hand, a vector that links two organizations. We also show how these vectors are evaluated and shared among the different organizations, and how we combine the provided trust information in order to enhance the security.

Second, the `TRUST-OrBAC` model was designed to add the previous trust approach to the `OrBAC` model. Moreover, this solution was applied with a real collaboration network between companies.

Third, we present a trust ontology methodology based on access control concepts. This ontology will be used to share the trust beliefs between participants and to make equivalence between their trust objectives. How to define this trust relationship, how to understand the trust objective of a requester, and how to evaluate the recommendation value is addressed in this thesis.

Fourth, we improve our work by designing a passive testing approach in order to evaluate the behavior of a user. This contribution is based on the monitoring tool MMT. Finally the entire architecture of our system is proposed.

Keywords: Trust management, Security Policy, Security Rules, `OrBAC` model, Passive Testing, Trace Collection, Trace Analysis.

Résumé

De nos jours, la propagation rapide des technologies de communication, de stockage de données et des web services encouragent les entreprises à collaborer entre elles formant ainsi un environnement multi-organisationnels. Ces entreprises participent à cet environnement afin de profiter des opportinités offertes tels que: (1) la possibilité d'utilisation des ressources et des services externes et professionnels (2) la réduction du temps de production et (3) les bénéfices résultant des effets de synergie.

Toutefois, cette collaboration n'est pas parfaite. Des nombreux problèmes peuvent apparaître tels que l'utilisation malveillante des ressources, la divulgation des données ou des services inadéquats. Par conséquent, la sécurité est une préoccupation importante des participants. Les principaux défis de sécurité pour un participant sont la gestion de la confiance et le contrôle d'accès. Dans cette thèse, nous avons abordé en particulier ces deux domaines et nous proposons une nouvelle approche de gestion de la confiance pour les systèmes mutli-organisationnels.

Notre approche est divisée en quatre parties. Tout d'abord, nous avons défini un modèle de confiance basé sur la notion des vecteurs. Ces derniers sont composés d'un ensemble de paramètres qui permettent de fournir un degré de confiance sous certaines conditions. Dans notre approche, nous envisageons deux types de vecteurs. D'une part, un vecteur lié à une relation entre un utilisateur et une organisation et d'autre part un vecteur qui relie deux organisations. De plus, nous avons montré comment évaluer et partager ces vecteurs entre les organisations, et comment utiliser les informations évaluées pour améliorer la sécurité.

Concernant notre deuxième contribution, nous avons intégré ce nouveau modèle de confiance dans le modèle de contrôle d'accès `OrBAC` (Organization Based Access Control). Cette intégration a donné naissance à notre modèle `TRUST-OrBAC`. En outre, nous avons appliqué cette solution à un cas d'étude de collaboration entre des entreprises.

Troisièmement, nous avons proposé une nouvelle ontologie de confiance basée sur des concepts de contrôle d'accès. Cette ontologie sera utilisée pour partager les degrés de confiance entre les participants et pour définir l'équivalence entre leurs objectifs. Ainsi, comment définir cette relation de confiance, comment comprendre l'objectif de la confiance d'un demandeur, et comment évaluer la valeur de la recommandation sont toutes des problématiques auxquelles nous avons essayé de répondre dans le cadre de ce travail.

Quatrièmement, nous avons amélioré notre travail par la conception d'une approche de test passif afin d'évaluer le comportement d'un utilisateur. Cette contribution a été basée sur l'outil de test MMT (Montimage Monitoring Tool). Finalement, nous avons conçu une architecture sécurisée d'un système distribué en se basant sur nos contributions.

Mots-clés: Gestion de la confiance, Politique de sécurité, Règles de sécurité, Modèle `OrBAC`, Test passif, Collecte des traces, Analyse des traces.

Thesis Publications

Journal papers

- Khalifa Toumi, Ana Cavalli and Cesar Andres. *Validation of a trust approach in Multi-Organization Environments.* Submitted to the International Journal of Secure Software Engineering, IJSSE'13.
- Khalifa Toumi, Cesar Andres and Ana Cavalli. *A Formal framework for defining trust in Multi-Organization Environment* Accepted by the International Journal of Autonomous and Adaptive Communications Systems, IJAACS'13.

Conference and Workshop papers

- Khalifa Toumi, César Andrés and Ana Cavalli, "*Security Properties in Virtual Organizations*", 15th IEEE International Conference on High Performance Computing and Communication (HPCC'13), IEEE Computer Society Press, November 13-15, 2013, Zhangjiajie, China.
- Khalifa Toumi, César Andrés and Ana Cavalli, "*Trust Ontology Based on Access Control Parameters in Multi-Organization Environments*", 9th International Conference on SIGNAL IMAGE TECHNOLOGY and INTERNET BASED SYSTEMS (SITIS'13), IEEE Computer Society Press, December 2-5, 2013, Kyoto, Japan.
- Khalifa Toumi, César Andrés, Ana Cavalli. "*Trust-OrBAC: A Trust Access Control Model in Multi-Organization Environments*". 8th International Conference on Information Systems Security, (ICISS'12), Lecture Notes in Computer Science, December 15-19, 2012, Guwahati, India.
- Khalifa Toumi, César Andrés, Ana Cavalli, Mazen EL Maarabani. "*A Vector Based Model Approach for Defining Trust in Multi-Organization Environments*" 7th International Conference on Risks and Security of Internet and Systems, (CRISIS'12), IEEE Computer Society Press, October 10-12, 2012, Cork, Ireland.
- Khalifa Toumi, César Andrés, Ana Cavalli. "*Setting trust evaluations with fuzzy logic in MOE*" 9th Workshop on System Testing And Validation, (STV'12), October 24, 2012, Paris, France.
- Khalifa Toumi, Ana R. Cavalli, Mazen El Maarabani., "*Role based interoperability security policies in collaborative systems*". the International Conference on Collaboration Technologies and Systems(CTS'12), IEEE Computer Society Press, May 21-25, 2012, Denver, CO, USA.

Part I

Introduction

"The beginning is the most important part of the work."
Plato, The Republic
"Good seasons start with good beginnings."
Sparky Anderson

General Context

In this thesis, we aim to propose a new framework that guarantees a secure interoperability among entities in Multi-Organization Environments. Different organizations collaborate in order to achieve some objectives. Due to the improvement of the communication protocols and the simplicity of their use, this collaboration is encouraged. Moreover, these kinds of networks have numerous advantages. This interoperability (1) reduces the time-consuming execution, (2) permits the use of professional machines and shared resources, (3) offers the possibility of consulting experts and (4) allows distributing scheduled tasks.

Before participating in a such environment, any entity needs to be sure about the security and the trust level of the whole system. Security issues and trust solutions in distributed systems are addressed by several researchers.

Security in distributed system is very active in the research and the industrial areas [Pre93, Gol10, MCHZ11, TW12]. As illustrated in Figure 1, different concepts are studied by researchers such as:

- Cryptographic solutions [Bla79, Pre93, LV00, KPR11].
- Authentication mechanisms [WL92, No94, PLHCETR06, JAS+11].
- Privacy techniques [BHS02, NWET04, JDF08, PB10].
- Security policy modeling frameworks [CM04, KD06, CCBC06b, FK09].
- Administration [HNAN06, ACCBCB08].
- Deployment [MFBT08, ACBC09].
- Testing [MOC+07, MBCB08, MCHZ11].
- Public key infrastructure [BFK99, MWS04, KKS13].
- Hardware security solutions [Sko05, MKP08, TW12].
- Risk assessment [SM97, ZECC04].
- Intrusion detection systems [SCCC+96, MVD12, HkCHC13].
- Trust management [BFL96, WTS+02, LLYT05, WLWV09a, HCBD09].

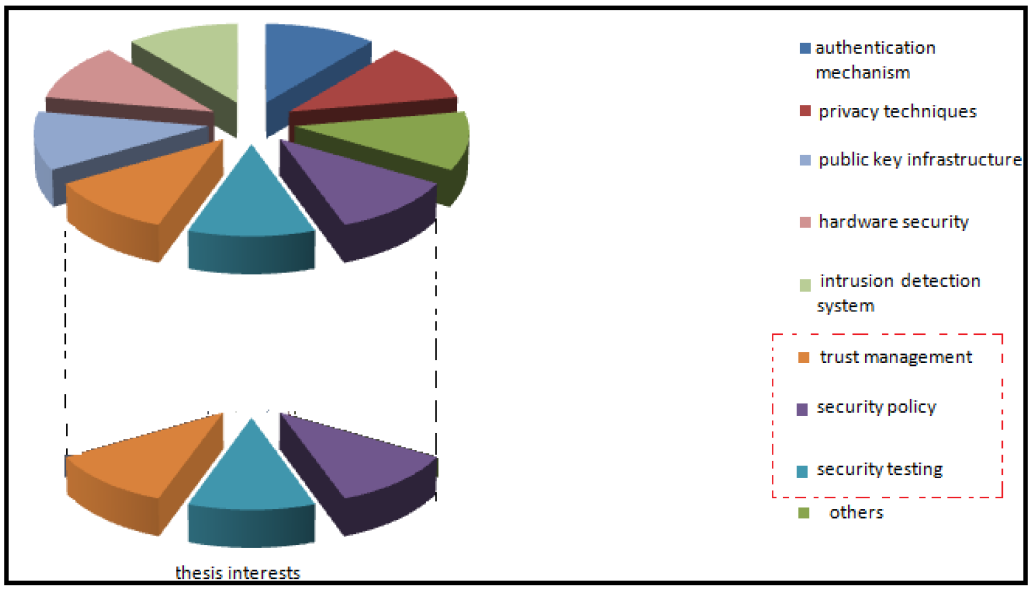


Figure 1: Thesis interests.

The goal of this study is to address trust management, security policy and security testing issues. In this thesis, our first objective is to provide a novel trust framework for Multi-Organization Environments (MOE) and to integrate it in an access control model that is the Organization Role Based Access Control (`OrBAC`). Therefore, a new trust architecture for this kind of distributed system is provided. Our proposal may be used in any distributed environment in which there are collaborations among different entities and, especially, organizations.

In order to achieve our objectives, we studied the existing trust frameworks. This survey has given us the possibility to understand the different challenges and issues to be resolved by our system. One of the most crucial problems in these solutions is the ambiguity of trust definition. Furthermore, the definition of the trust parameters and their evaluations have to be adapted to the MOE requirements. These issues have been the first motivations to study this topic. Additionally, the comparison between the different access control solutions was necessary to understand their advantages and drawbacks in order to choose the most suitable one for our solution. This state of the art helped us to determine a list of objectives.

Thesis Objectives and Contributions

Figure 2 illustrates the list of objectives and contributions. In this thesis:

- We propose a survey of role based interoperability security policies. This assessment is based on some crucial criteria for collaborative environments. This study permits to compare and analyze the advantages of several RBAC and `OrBAC` based models. It entitles us to learn from the diversity of the research based on these models in order to improve the interoperability policies in collaborative environments. This analysis also proposes some perspectives and suggestions.

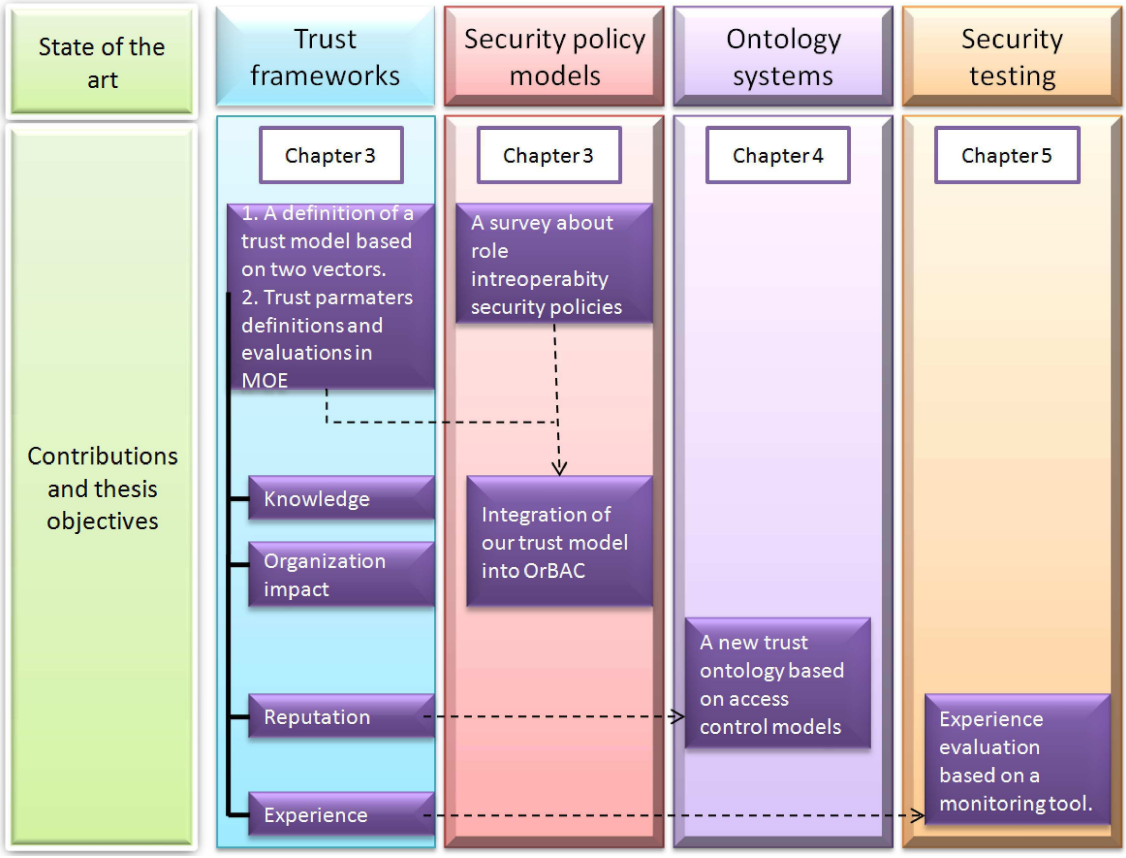


Figure 2: Thesis objectives and contributions.

- We define a new trust model based on two vectors: the first one is applied to users (utv) and the second one is applied to organizations (otv). For instance, the organization trust vector $otv = (e, r, k)$ expresses that the trust relationship between two organizations depends on three parameters. The first one corresponds to the previous interactions between the truster and the organization. The second one represents the reputation of the trustee in the MOE environment. Finally, the last one denotes the knowledge of the organization regarding the truster. An additional contribution of this proposal is to provide an evaluation for each parameter of these vectors. In our model, these evaluations are dynamic, that is, the evaluations depend on time. Therefore, we have that trust is a relation among two entities (the trustee and the truster), related to a specific behavior of the trustee (situation), in a specific slot of time.

- One of the parameters studied in this thesis is the reputation. In a first contribution, we address the reputation definition and evaluation. To achieve this work, we have to resolve other issues such as, for instance, how to share the feedback values, how to understand them and how to reduce the false feedbacks. All these issues are discussed and a new trust ontology based on access control concepts is proposed. We have extended the `OrBAC` ontology with new

trust classes that are presented and detailed with some examples using the Protégé tool [GMF+03]. Furthermore, we have also proposed a mapping algorithm in order to study equivalence between two situations of different participants. Two ontologies, one of the TMSP university [Par14] and another of the MMT company [Mon14], are implemented. This approach details how to use the reputation process with the `TRUST-OrBAC` model to enhance the security of the system. Several subprocesses are proposed to define the trust presentation, to share the trust beliefs, to map among situations and to evaluate the recommendation values.

- The evaluation of interactions is crucial in order to evaluate the experience of an entity. In this thesis, a new monitoring approach to evaluate the trust is proposed. In order to achieve this goal, (1) a new plug-in called 'trust-plug' is developed to analyze the trace and to determine the different elements that will be used to evaluate the trust (2) the formalism permitting the specification of the MMT rule is updated and (3) an extension of the MMT tasks by adding periodical verification and trust level notification is developed.

- An integration of our trust solution into the `OrBAC` model is proposed to form the `TRUST-OrBAC` solution. This integration is based especially on a new trust context that permits to activate or deactivate some rules based on the trust level of a user or his organization.

- We also proposed a whole architecture that includes our trust solution with the `OrBAC` model and the MMT tool in order to secure the interoperability between organizations in MOE.

These contributions have given rise to several publications in international conferences and workshops like [TCM12, TAC12a, TACM12, TAC12b, TAC13c, TAC13b, TCA13, TAC13a].

Thesis Plan

As it is illustrated in Figure 3, the remainder of the thesis is organized as follows.

In the first chapter, we present the basic concepts related to the security in MOE. We first study MOE systems: their advantages, architectures and concepts. Some examples of MOE environments are also presented. We detail the different attacks that may be detected in this environment. The different security services that may be damaged by these attacks are studied. Finally, we focus on the different methodologies that allow to evaluate security and risks of the system.

In the second chapter, we present the background related to security policy models. We define the security policy components. Then, we propose a list of criteria that can be used in order to compare the existent models. The list of models studied in this chapter is based on the two models RBAC and `OrBAC`. For this we provide a comparison between these two models. Then we focus more on the `OrBAC` based solutions. Once the survey is performed, the results clearly highlight that the most

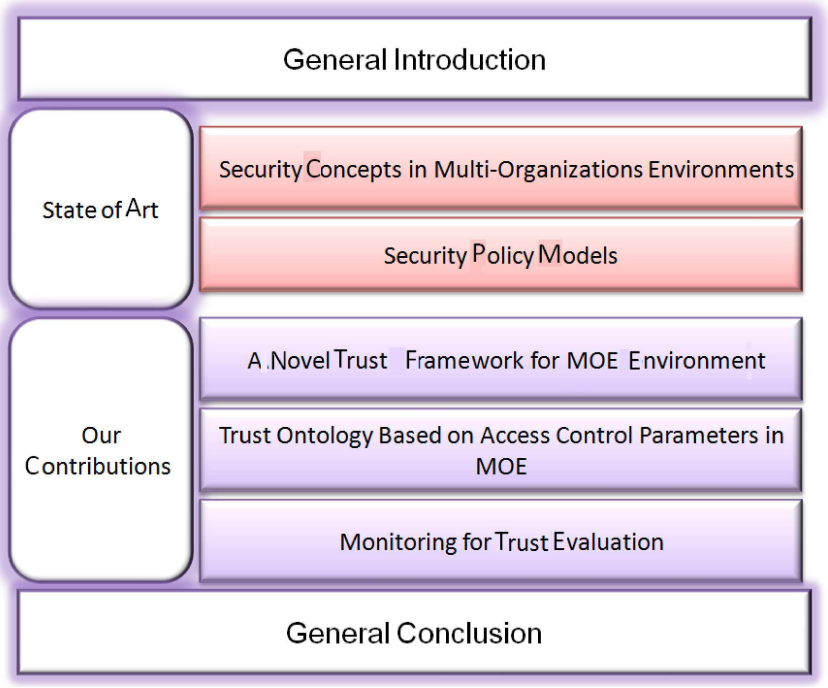


Figure 3: Thesis plan.

adequate model in our case is the OrBAC model for several reasons: The organization concept and its modeling framework is well defined in `OrBAC`. The context notion and its diversity types offer a more dynamic policy. The high abstraction level simplifies the administration task. Moreover, its administration tool MOTOrBAC offers several functionalities as the simulation of the concrete security policy, the edition of the policy and the resolution of conflicts. After this survey, we are convinced that we may design our solution based on the `OrBAC` model.

Chapter three presents our novel trust framework. First, a state of the art of the trust challenges and the existent solutions are presented. In addition, we detail our trust parameters, their definitions and evaluations. Then, some rules to check the integrity of the solution are defined. Our new trust vectors are also illustrated. After that, how to integrate this trust solution into `OrBAC` is explained. This chapter is concluded with a case study and an architecture that integrates our framework.

The fourth chapter extends this work to detail more the reputation parameter. Indeed, some issues related to the reputation are still not resolved in the third chapter. The fourth chapter is structured as follows. The first section presents some related works. Next, our trust ontology and its classes are detailed. In this chapter, we introduce our mapping process and the recommendation evaluations approach that are used within our trust framework.

The fifth chapter also extends chapters three and four by discussing a new assumption related to the behavior evaluation. In different trust solutions, a satisfactory evaluation function behavior is not detailed. In this chapter, we propose to use the MMT tool and to provide some trust properties in order to evaluate an interaction.

Thus, we show how to evaluate the satisfactory function and the possible strategies to evaluate a behavior. Moreover, we illustrate how to integrate the MMT tool with our trust framework. Finally, a case study is detailed.

The final part of this document concludes the thesis by summarizing our main proposals and contributions and presents some perspectives of our work.

Part II

State of the Art: Security Concepts and Policy Models in Multi-Organization Environments

Chapter 1

Security Concepts in Multi-Organizations Environments

"Security is, I would say, our top priority because for all the exciting things you will be able to do with computers - organizing your lives, staying in touch with people, being creative - if we don't solve these security problems, then people will hold back."
Bill Gates

Contents

1.1 Introduction

Distributed systems as network based web services, peer to peer network and grid computing system offer a simple way to collaborate among organizations.

This approach offers several advantages such as: (1) the ability to use extern and professional resources, services and knowledge, (2) the reduction of time-consuming requirements and (3) the contribution of experts experience. However, the participation in a distributed system has a negative side also. This environment has to address more security challenges than a single system. The fact of facilitating the collaboration with external users may encourage malicious nodes to gain illegitimate access or to block a service. Different attacks can be performed in the system that may destroy the data or use it without the authorization of the owner. For instance, a spoofing or a denial of service attack may damage the security of the environment.

In order to protect the system and to encourage organization to participate in this environment, we need to provide a solution that offers *a secure communication* among participants. Following the quote of the author Sun Tzu, in his book "the art of war":

> *"If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle."*

We should know and understand the concept of a distributed system and also to understand the attackers and their strategies in order to protect the system. Therefore, this chapter follows this quotation:

1. We will study at the first step our system: its advantages, architecture and concepts in order to be able to correctly define the environment. Let us note that this knowledge will be enhanced by three examples of distributed systems, those are P2P networks, web services and grid computing.

2. Next, we propose to represent these networks in Multi-Organization Environments, in short MOE. This fact will allow us to understand better the properties of the system.

3. Finally, we will focus on some projects and security threats that will be detected in the system.

Let us remark that the study of the different detected threats in these examples introduces the second base of knowledge that should be addressed in order to protect our system. This base is the knowledge of our enemy (attacker). As a result, we will be able to model different attacks that may happen in MOE. In particular, we will focus on describing:

- The definition of the attack.
- The class of the attack.

- The strategies that are used.

With this information, we are able to study different security services that may be damaged by these attacks. In particular the three basic services those are *confidentiality, availability and integrity* will be detailed and mapped with these attacks. Next step, we will study different methodologies that allow us to evaluate the security level and the different risks that the system can encounter. The understanding of these approaches is recommended, in order to help any researcher or administrator to design its security policy and to establish different targets.

1.2 Multi-Organization Environments

1.2.1 MOE Advantages

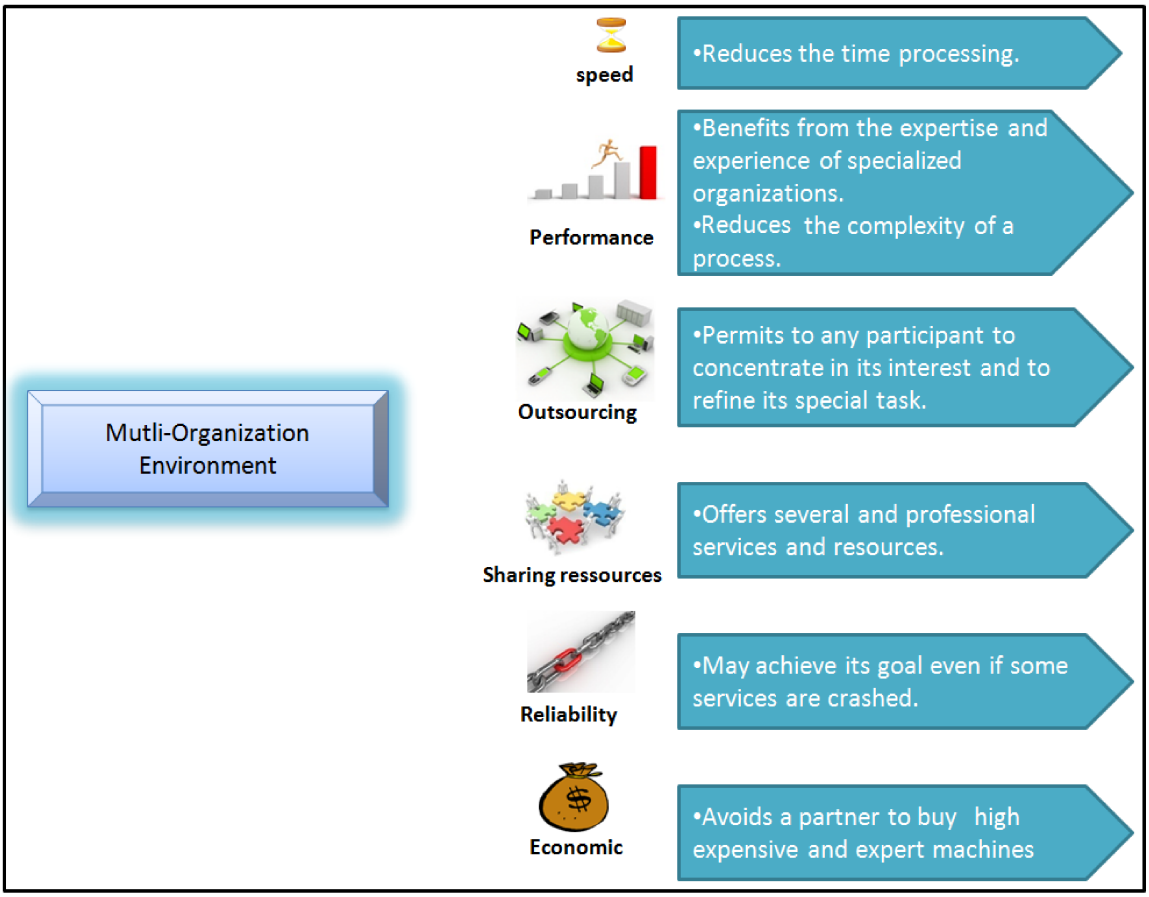


Figure 1.1: MOE advantages

Recently, the widespread of inexpensive communication technologies, distributed data storage and web services mechanisms urge the collaboration among organizations. This interoperability creates the MOE environment that is a distributed

system composed of various organizations. An organization can be defined as a set of shared resources and a set of users working in it. As it is shown in Figure 1.1, MOE offers several advantages for the participants such as:

Speed MOE enables the parallel execution of several processes at the same time. This feature reduces the time processing of tasks.

Performance A distributed system offers a greater service performance than a centralized one. In fact, users in MOE can benefit from the expertise and experience of specialized organizations. Each task will be done by the most qualified partner. Moreover, the segregation of tasks may reduce the complexity of the process: Instead of doing all the sub-processes by one server, they will be divided between several servers.

Outsourcing It is the contracting out of an internal business process to a third-party organization. This allows to the organization to concentrate in its interest and to refine its special tasks. We find that more and more organizations are looking to Information Technology, in short IT, outsourcing through external service providers [LHKP03].

Sharing resources This allows an organization to take advantage from published services and resources. In fact, to share resources is mandatory for several applications, for instance, a transaction between a bank and an university or the use of the patient files between two hospitals.

Reliability The distribution of the different services between several partners permits to continue a task even if one machine or one partner is not available, since, the same service may be provided by other partner or service.

Economics better price A service can be used by several client applications. This environment avoids that a partner acquires costly machines. Moreover, the improvement of the IT permits to reduce the cost of the coordination. Indeed this environment takes advantages of the rapid progress on the IT area. Without this progress, the collaboration will be very difficult and expensive [SLS98].

1.2.2 Definition

In the literature we find several definitions of this distributed system. We cite hereafter some of them:

- It is a temporary network of companies that comes together quickly to exploit fast-changing opportunities [SLS98].
- It is a group of people who interact through interdependent tasks guided by common purpose that works across space, time, and organizational boundaries with links strengthened by web communication technologies [LS00].

- It is a collection of individuals and institutions that is defined according to a set of resource sharing rules [FKT01].
- It is an association of organizations and their related supporting institutions adhering to a base long term cooperation agreement, and adoption of common operating principles and infrastructures, with the main goal of increasing both their chances and their preparedness towards collaboration in potential Virtual Organizations [MA03].

In our approach, we may say that there are several names that are assigned to this paradigm as Virtual Organization (VO), Virtual Enterprise (VE), Multi-Organization Environments (MOE) and Collaborative Environment (CE). In this report, we choose MOE and we define it as the following:

Definition 1 This environment forms a distributed system with several interactions between different organizations in order to achieve a common task. Each organization of the MOE acts as an O-grantee and/or O-grantor. The O-grantor is the participant which offers a resource to be used by another organization called the O-grantee. Figure 1.2 illustrates a simple example of the MOE environment. □

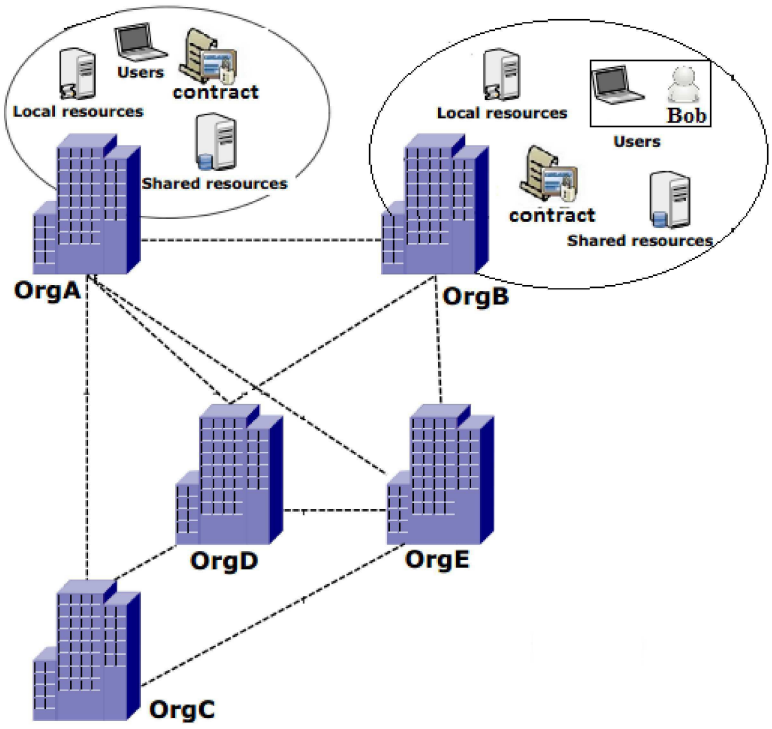


Figure 1.2: MOE Example.

In the following, we present the concepts of contract, interoperability security policy, and architecture in MOE.

Contract In MOE, several control access models [CCBC06b, KLBB08, KDBK09] are based on a contract on which organizations are based to define their rules for collaboration. This contract forms a duality between the maximization of the security and the insurance of the cooperation.

It may specify several information such as duration, financing, shared resources, etc. The signature of this contract means that the O-grantor accepts to share resources with the O-grantee and should respect rules that are specified in this contract. The work in [CCBCC08c] details and studies the creation of contracts in a MOE.

Interoperability Security Policy Each participant has to create its interoperability security policy in order to control the access to its shared resources. The next chapter will investigate in depth this concept by presenting its main concepts and characteristics.

Architecture MOE architecture can be classified into two categories: distributed and centralized architecture. We present on the following the required steps in each topology when a user `u` from an organization `OrgB` needs a service from another organization `OrgA`.

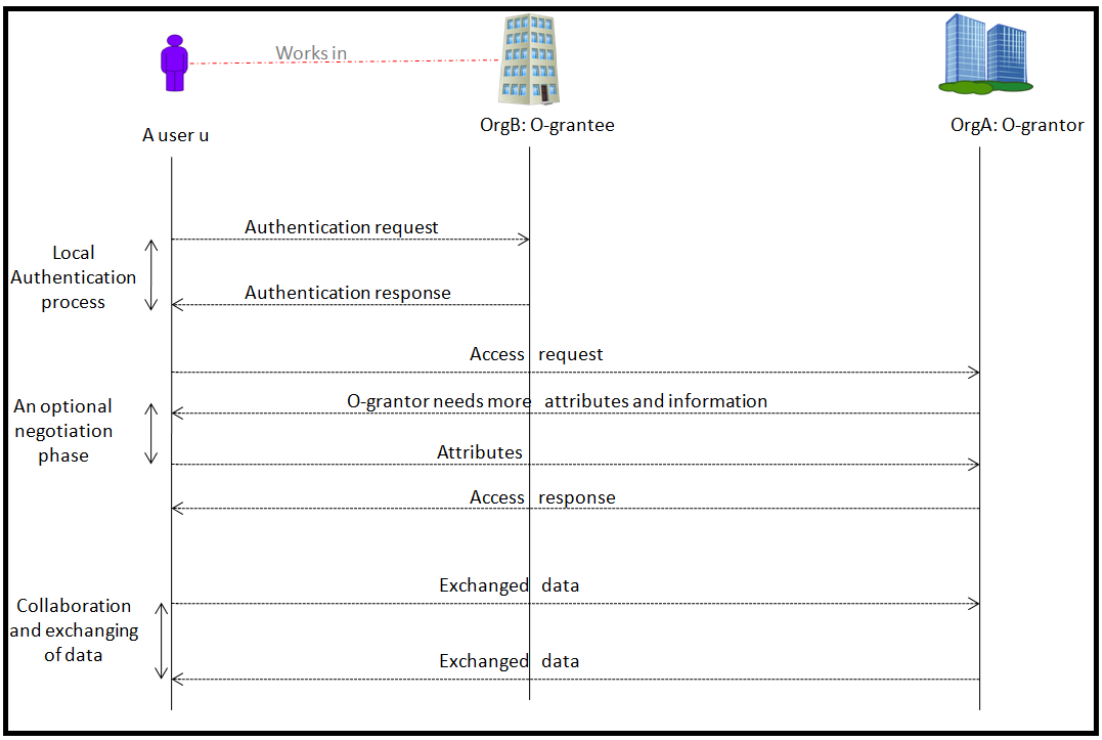


Figure 1.3: Collaboration in a distributed architecture of MOE.

Example 1 In the first case, as it is illustrated in Figure 1.3, the user `u` must first be authenticated in `OrgB`. Next, a request from the `OrgB` with some attributes is sent

to the `OrgA`. This latter will be the responsible of providing a decision to accept or refuse the access based on its interoperability security policy. In this approach, the authentication is done in the user's organization and the decision is realized in the O-grantor.

In the second case, as it is illustrated in Figure 1.4, the architecture requires a third party `OrgC` that must be trusted by the different participants. First, the user `u` sends the request to the organization `OrgC` that authenticates it in order to proof its belonging to the organization `OrgB`. Based on the interoperability security policy, this third part will decide to offer or refuse this access. This step may need an exchange of credential between `OrgC` and `OrgB`.

If accepted, the request will be signed by `OrgC` and sent to `OrgA`. In this approach the authentication and provider decision are managed on a third trusted part.

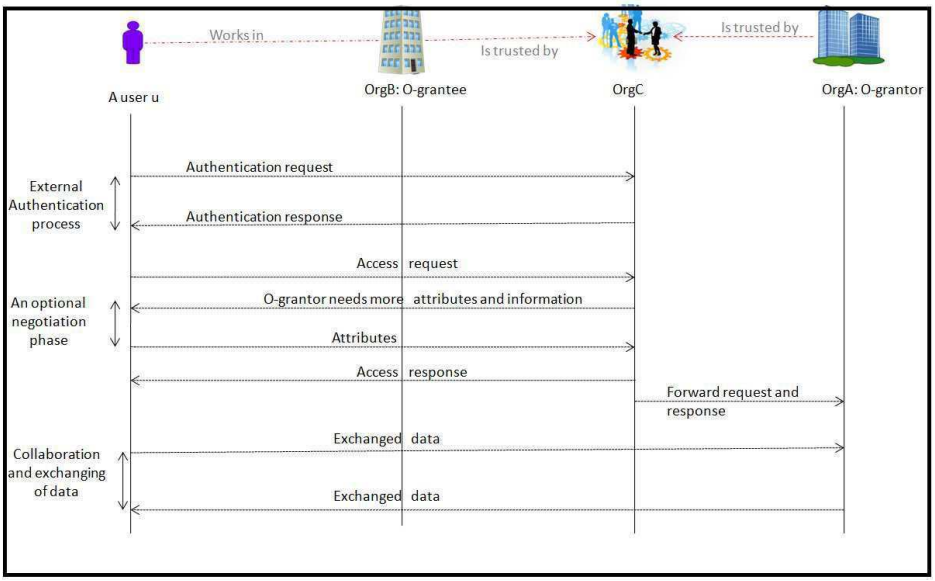


Figure 1.4: Collaboration in a central architecture of MOE

1.3 Communicating Networks (different case studies)

In this section, we give an overview about some pervasive networks (allowing the communication between different partners), their concepts, architecture and some of their security threats.

1.3.1 Peer-to-Peer (P2P) Networks

Several applications as *aMule* , *eDonkey2000* and *Shareaza* have been developed in order to simplify the participation on a P2P system. Based on [EDU09], CNN

used P2P to stream the presidential inauguration to more than 650.000 simultaneous users. Amazon and Google have experimented with some of its consumer services based on this technology. This useful technology is an approach that easily enables to share information between nodes.

It is characterized by a dynamic topology. P2P is not like a classical client/server network with some specific server. Different nodes are equal in P2P and can participate on sharing information or establishing communication with another peer. This system is used for several issues as distributed computing, sharing files, streaming video and interoperability. Nowadays, several issues and challenges are encountering the P2P committee. Following, we present some research projects in P2P:

- P2P-Next project is realized between 2008 and 2011. This project aims to introduce the form of the internet TV. P2P is considered as a distribution mechanism as satellite, cable and terrestrial networks. This project provides a new set-to-Box called NextShareTV in order to design a new delivery mechanism with a social and collaboration between all the nodes of the P2P network. This project aims to gain from the collaboration and the participation of each node.

- P2P_Architect is an European project carried out between 2001 and 2004. Its main goal was to ensure the dependability requirements for the P2P networks. The different challenges studied by this project are related to the availability, reliability, survivability, responsiveness and security of the network. This project developed a methodology and tools for supporting this new architecture.

- PeerTrust aims to design a trust platform for peer to peer entities in order to quantify and evaluate the trustworthiness of another node. In this project a policy languages for trust and security requirements is developed.

The goals that may concern to us are:

1. *How can we adapt the notion MOE to P2P networks?* and,

2. *What are the security risks in P2P system?*

P2P networks can be seen as a MOE environment where a group of peers is considered as an organization. These peers may be students from the same school, employees from the same enterprise. As it is illustrated in Figure 1.5, there are two types of communication: local communication between nodes in the same group and external communication between nodes in different groups (black lines in Figure 1.5). For our scope, we will concentrate only on the communication between these organizations.

P2P networks simplify the distribution of files, however some threats may destroy the collaboration as detailed by the following example:

1. A malicious code can spread easier and faster than a local communication. As a result computers will be more exposed to worms and viruses.

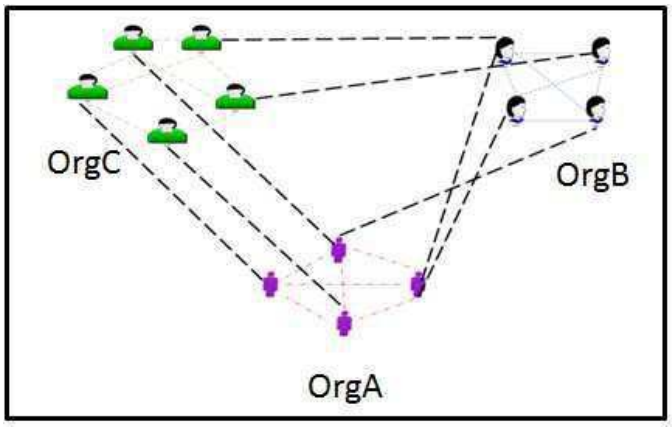


Figure 1.5: MOE structure with P2P networks.

2. The distribution of copyrighted music, movies, software, games, and other materials without authorization can often happen.

3. Unauthorized access to sensitive data: with P2P, users are client and server. Several users share their documents with untrusted nodes unknowingly. In fact, in [JMW08], authors show different experiments searching the Gnutella network for sensitive personal documents. Their resulting files shown in the next table should be surprising to users of P2P networks:

Birth Certificate	45 Results
Passport	42 Results
Tax Return	208 Results
FAFSA	114 Results

4. Malicious P2P software may easily modify the configuration of the P2P client. Therefore, you may give access unknowingly to some data of your hard disk.

1.3.2 Web Services

Web service is a way to collaborate between different partners. It gives a basic infrastructure for distributed web data management. Different concepts, organizations, humans, business and technologies science are merged in the same challenge: How to succeed this collaboration?

The World Wide Web Consortium (W3C) has defined web service as the following:

> *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format. Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*

A web service architecture is based on three components:

- Service provider: this term designates the owner of the service. It publishes, removes and updates its services. In MOE, it corresponds to the O-grantor.
- Service requester: It is the partner that requires a service. In technical view, it will be an application installed in the O-grantee that invokes the service.
- Service broker: It is the repository where users can search for a service description. This entity is a third participant that helps the collaboration between the client and the provider. It is not present in the last definition of the MOE. Any client has to consult the service description directory in order to find and select a service.

Example 2 Figure 1.6 illustrates how to use web services in MOE: The O-grantor will provide a web service that allows the access to a shared resource. A client of the web service have to be installed in the O-grantee in order to offer to any user the possibility to apply for a resource shared by the service provider. The communication between different partners is based on a standardized XML file regardless the platforms used in the client or the server side.

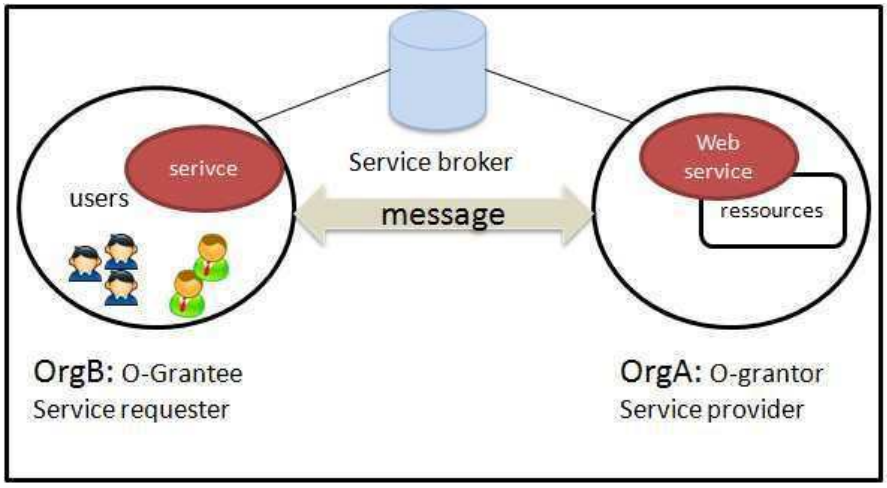


Figure 1.6: MOE structure with web service technology.

Web service brings a lot of advantages as previously detailed. Nevertheless, new threats and risks are emerging. Among them, we may cite:

Malicious code injection in an XML file, such as Xml injection, XPath injection or XQuery attacks, may inject a malicious code into an XML document. For instance: An XPath injection attack, which is similar to the classic SQL injection attack, aims to gain an illegal access. Therefore, the attacker may control, modify or delete data of the XML file without the authorization of the owner.

DOS attack Denial of service attack aims at interrupting the functioning of a server. There are two principal methods of DoS attacks: crashing or flooding methods. The first one aims to exploit the vulnerabilities of a service in order to be crashed. The second is based on sending of several requests or messages to the server causing it to slow down or to stop. One famous DOS attack in web services is the *array href expansion*.

Intermediary attack This attack is based on a web service intermediary that intercepts messages and modifies them without being detected.

Replay Attack Flaws An attacker captures a valid message and replays on it in order to gain unauthorized access to a resource.

1.3.3 Grid computing

Dr Richard Crandall is the first researcher that have experimented the grid computing with its program called Zilla. This latter used machines chained together for complex mathematical treatments. The term grid computing emerged in the midst of 1990s and is an analogy of the grid electricity (power grid) (See Figure 1.7).

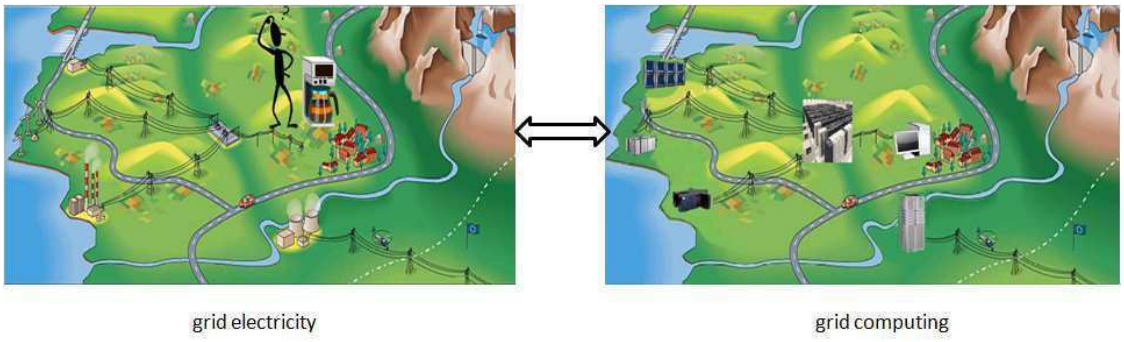


Figure 1.7: Mapping between grid electricity and grid computing [JL06]

Grid computing allows users to perform distributed computing: It leverages the power of computation (processors, memory, etc) of thousands of computers to give the illusion of a powerful virtual machine. This model can solve important computational problems requiring very long time in "classic" environment.

Grid computing works as an electrical distribution network: It provides for its participants all the resources they need through a simplified interface. The complexity of this network is completely hidden. In addition, the user can vary its brutally consumption without prior approach. In grid computing, computing power and storage capacity are virtually unlimited, since all grid resources can be mobilized when needed.

Grid computing components are:

- An administrator for (1) scheduling tasks and managing priorities, (2) association tasks and resources and (3) ensuring normal performance for users.

- A Middleware that allows the execution of a process through the network. Without it all communications between the system are impossible.

Figure 1.8 illustrates how to present a MOE in the Grid computing.

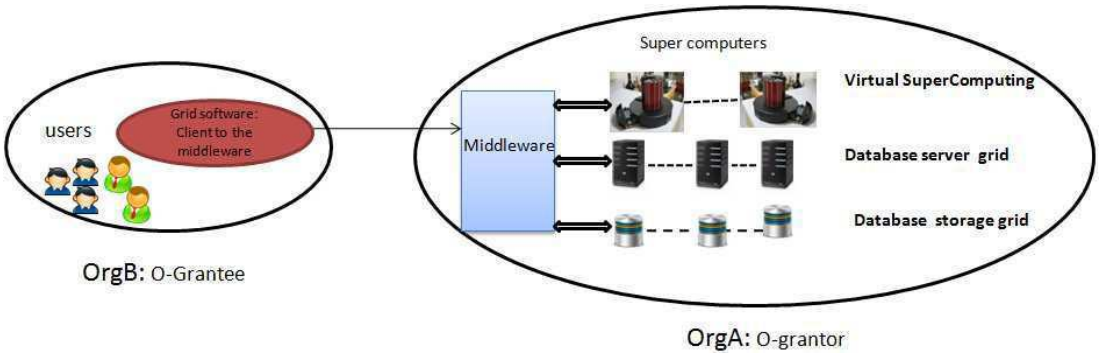


Figure 1.8: MOE structure with grid computing

Mark Teter, chief technical officer of Advanced Systems Group, said

> *"The highly automated manner in which resources are allocated on a grid can be used by a malicious attacker to steal sensitive corporate data. (...) It is crucial to safeguard the grid and the data being distributed. Your whole storage infrastructure can be compromised."*

Different attacks may happen in this network such as

1. Attacks against the client:
   (a) An illegal use of resources rent for other users: A malicious node may use a software client license of another entity in order to consume its CPU cycles rent from the O-grantor.
   (b) An unauthorized use or modification of the client data or metadata saved in the grid.
2. Attacks against the O-grantor:
   (a) An illegal use of the bandwidth, CPU cycles or storage capacity provided by the O-grantor.
   (b) An unauthorized use or modification of the system file, log files and password files of the grid provider.

1.4 Common Threats and Security Requirements in MOE

In the previous section, we presented some specific attacks detected for the P2P, web services and grid computing networks. In order to protect a distributed system from this threats and vulnerabilities, we have to detail:

- The possible threats and vulnerabilities that can disturb the collaboration between the different partners.
- The security requirements that permit to enhance the security level of the network.

In the following, we present different types of attacks that may happen in MOE. The Committee on National Security Systems of United States of America (CNSS) defines an attack as:

> *Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself.*

In this system, two classes of attacks can be defined, passive and active attacks:

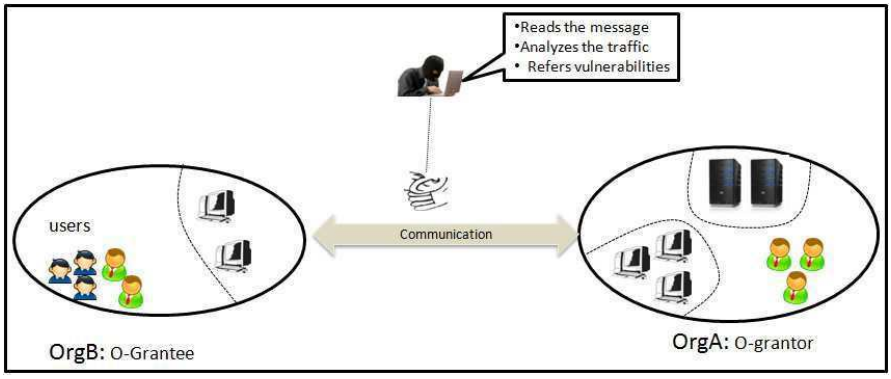


Figure 1.9: Passive attack.

Passive attack The attacker is a malicious node that aims to sniff the message content exchanged between the O-grantor and the O-grantee. In this scenario, the attacker will not modify the content. As it is illustrated in Figure 1.9, the attacker will read and analyze it in order to detect some vulnerabilities, to exploit or steal some sensitive information.

Active attack The attacker aims to modify, disturb and/or interrupt the functioning of the server in the O-grantor. In literature, several attacks belong to this family. Four sub-classes are presented, hereinafter:

Denial of Service (DOS) attack This attack may be based on legal requests that will be sent during a short period by the same attacker (called DOS) or by several attackers (called DDOS). This attack may (1) destruct a server, (2) influence the network bandwidth and (3) largely consume the server memory. Therefore, the O-grantor will lose money and reputation of other participants.

This attack may be based on a simple request or a complex series of packets. The reception of a huge number of packets simultaneously will cause several problems to the organization (see Figure 1.10).

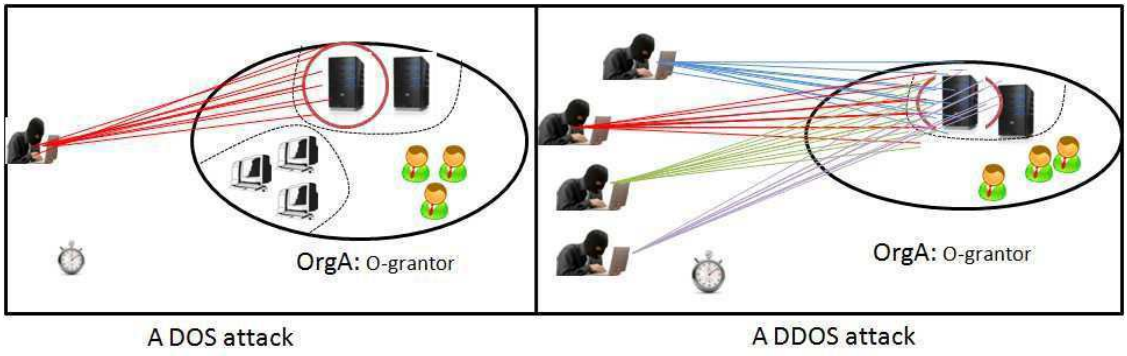


Figure 1.10: Deny of service attack.

Spoofing attack Spoofing attack is a way to hoax or deceive the O-grantor. The attacker aims to trick the victim in order to (1) steal information, (2) hide an attacker and/or (3) gain access to restricted resources. The attacker will send requests using a disguised identity to appear as a legitimate source (see Figure 1.11).

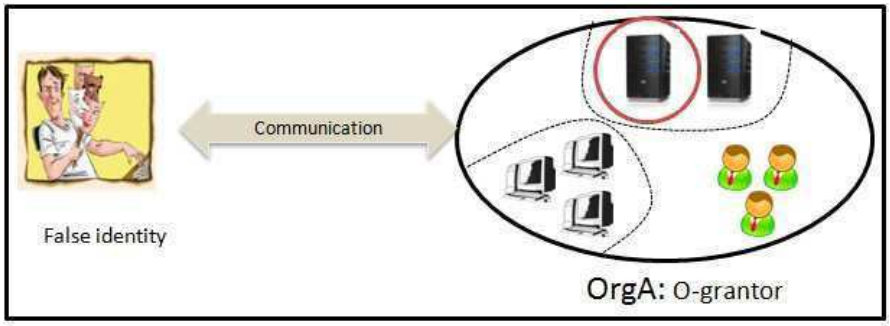


Figure 1.11: Spoofing attack.

Modification attack This attack aims at modifying the content of some messages exchanged with an O-grantor. This modification can be a deletion, an insertion and / or a replay. This modification might be realized to trick the O-grantor and/or to gain of unauthorized access (see Figure 1.12).

The different attacks aim to threaten one or more security services of the distributed system. An administrator or a researcher working in the security of a distributed system has to define and specify the different security services that need to be offered in his system. Next Section will define the different security services to take into account in MOE.

1.4.1 Security Services in Distributed Systems

In Figure1.13, we present the basic security services, those are integrity, availability and confidentiality. The green boxes are the main concepts that will be studied in

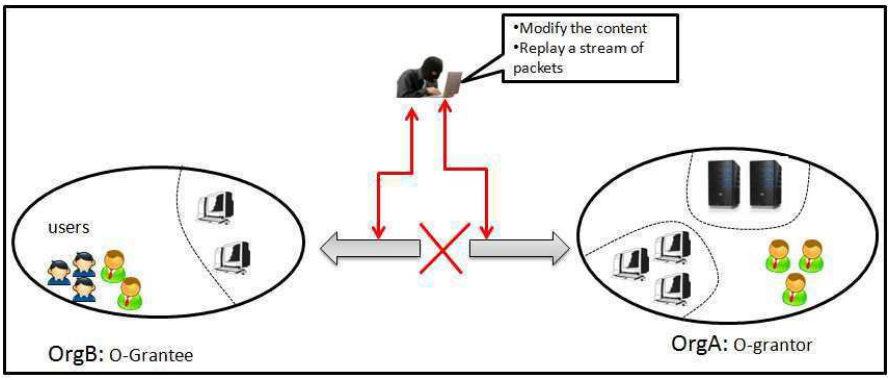


Figure 1.12: Modification attack.

this thesis.

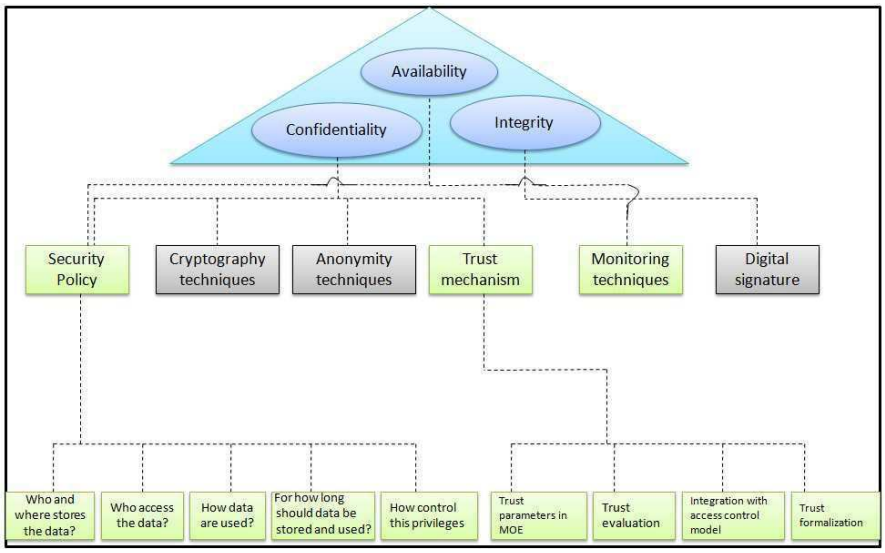


Figure 1.13: Security service and challenges in MOE.

1. Confidentiality. In MOE, servers will have access to different sensitive information of their clients. Providing confidentiality is ensuring that this information will be acceded only by authorized users. Several questions have to be solved in order to offer the confidentiality.

   (a) How to share identities?

   (b) Who stores the data?

   (c) Who controls the data access?

   (d) Who accesses the data?

   (e) How data is used and what are the possible actions that can be realized?

(f) Do we need to encrypt the exchanged data and how to do it?

(g) For how long should data be stored and used?

(h) How should access rights be obtained?

This service depends on different techniques and concepts. Following, we present some of them:

Cryptography is a method that is responsible of transferring data according to an algorithm in order to be unreadable by unauthorized persons. Even if an attacker is able to receive the encrypted data, he will not be able to understand its contents. This data will be only decrypted by a specific receiver. This encryption is based on the symmetric or asymmetric techniques. For the first method, the encryption and the decryption of the data will be based on the same key. For the second one, there is a public key that will be used to encrypt the message and a private key only known by the receiver to decrypt the data.

Anonymity is a technique that enhances the confidentiality of a system. It insures that a user communicating with a server does not share clearly its real identity. Several approaches are proposed to solve this problem. For example: In [UEN+09], an object will create a random identity that will be used only during one session. Only the portal is able to know the real identity of the object after a registration phase.

Security policy and trust techniques Other solutions to improve the security of the system are based on the design of security policies and the use of trust techniques. This two areas are related to our contribution. Therefore, a detailed study is presented in the following chapters.

2. Availability

In 8 may 2009, the Gmail application suffered from a big failure. This problem caused a unavailability of the server more than one hour for several users.

June 1999: An unavailability during 22 hours of the web server eBay site, that has thwarted more than 2.3 million auctions. eBay reimbursed the registration fee lost auctions, between 3 and 5 million.

As it is illustrated in these examples, availability is considered as a security service that aims to ensure the access to the data for the authorized person as it is intended (time and location). DDOS and DOS are the type of attacks that have to be solved in order to offer this service. The use of some monitoring system to prevent the administrator from these attacks is recommended to provide the availability. Different mechanisms help to guarantee this service as: the access control, the duplication of data and the protection of the sensitive information.

3. Integrity

   The third security service is integrity. It ensures that data has not be changed during the transfer. It also includes *source integrity*. This ensures that the real sender is the same extracted from the message. In order to provide integrity, digital certificates and checksums/hash algorithms are used. We give a definition of these two techniques:

   - Digital certificate is a document that aims to identify the author of some data or a message. With a comparison to the real life, we may say that the digital certificate is an identity card in the digital world. This certificate contains several information:
     - Holder name.
     - Start date of the validation of the certificate.
     - Expiration date.
     - Identification of the signature algorithm.
     - Designation of the authority issuing the certificate.
     - Identification of the encryption algorithm and the value of the public key.
     - Identification of the signature algorithm and signature value.

     A certificate is signed by a Certification Authority (CA) with its own private key. The CA is an entity that issues the digital certificate. It is responsible for:
     - Processing of requests, validation/rejection of a request, revocation of a certificate
     - The production of certificates electronic secure environment
     - The distribution of certificates.
   - Hash algorithm is a method based on a cryptographic algorithm. Encrypted data with a hash algorithm cannot be decrypted. It is used in order to obtain a fix and precise quantity of data that permits to identify the sender of a message. It is a way to obtain a digital fingerprint of data. There are different algorithms, famous ones are MD5 and SHA1 [KBC97]. These are used in order to secure passwords, for the digital certificate and also for the data bases.

Example 3 In Figure 1.14, we give a matching between the possible type of attacks and the security service:

Finally, we present on the following different standards that are used to evaluate the security of a system.

	Availability	Confidentiality	Integrity
Passive attack		√	
DOS and DDOS	√		
Spoofing			√
Modification attack		√	√

Figure 1.14: Matching between security services and attacks.

1.5 Security Evaluation

A system is secure if the safety regulations cannot be violated. In order to define the security requirements, the security policy and the mechanism to establish, several organizations propose a methodology to help an administrator to achieve these objectives. This methodology proposes approaches on how to evaluate the system security and how to define the important criteria. In this section, we present some of these methods such as: EBIOS, MEHARI and OCTAVE that help any security administrator and researcher to design their interoperability security policies and to evaluate them.

1.5.1 EBIOS

EBIOS (Expression des Besoins et Identification des Objectifs de Securite) was created in 1995 by ANSSI [ANS13]. It aims to assess and process the different risks for a participant. It proposes to the administrator a security policy based on risk evaluation. This method is composed by different guides, introduction, description of the process, knowledge base and tools. A free software that simplifies the application of the method is offered.

As it is illustrated in Figure 1.15, five steps are proposed by EBIOS in order to determine the security requirements:

1. Study of the context and the environment: It consists in the study of the system to determine the context and the limits of the study (business presentation, the information system architecture, technical and regulatory constraints, etc).

2. Security needs analysis: In this step, the selection of sensitive systems will be done. Then, the different security needs will be determined. These needs will depend on the three criteria confidentiality, availability and integrity.

3. Threats analysis: It consists in the study of the different threats and vulnerabilities based on the technical architecture of the system. Therefore the used equipments, hardware and the network architecture will allow to define possible attacks and vulnerabilities.

4. Security objectives identification: This step aims to determine the security objectives that should be satisfied in order to have a secure system. This is a part of a security specification. These objectives are based on the technical physical procedural and organizational measures.

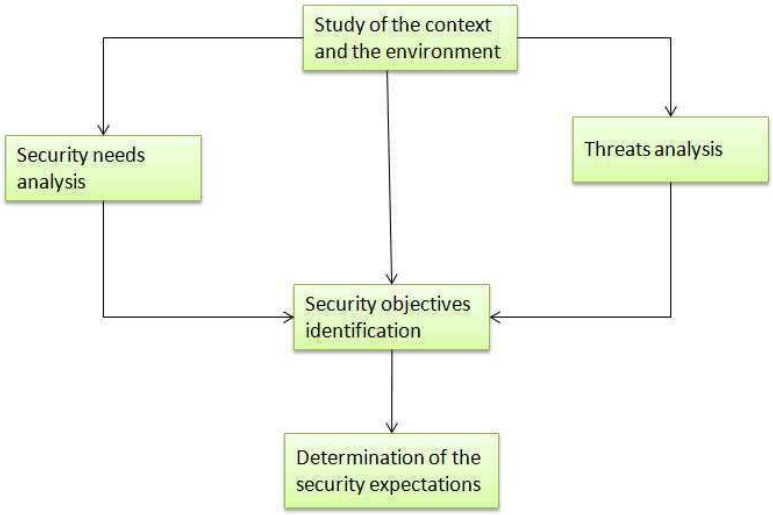


Figure 1.15: EBIOS approach

5. The determination of the security expectations: In this last step, the limits of the security protection have to be fixed. Indeed, in order to reduce the cost of the protection, some risks will be accepted. A plan and a management of risks should be provided. The cost and the probability of the occurrence of the risk will influence this strategy.

1.5.2 Method for Harmonized Analysis of Risk: MEHARI

It is a methodology developed by the French club of information security, called CLUSIF, that replaces an older one, called MARION, as well designed by the club. It offers a tool that helps to define the context, to assess the risks, to plan and monitor the treatment of risks. This methodology conforms to the requirements of ISO / IEC 27005 standard to manage the risks.

The MEHARI approach is based on:

1. Security analysis: In this step, MEHARI approach aims to identify the different problems that might be caused by any security defects then to assess the severity of these problems. Four levels of severity of loss dysfunction may be detected, vital, severe, important and not significant, to obtain the security measures. As EBIOS, these problems may happen due to the lack of confidentiality, integrity and/or availability.

2. Vulnerability analysis: This is based on a quantitative assessment of the quality of security measures that are described in a knowledge base developed by CLUSIF. During this step, the enterprise may correct unacceptable weaknesses by defining a list of action plans and assess the effectiveness of measures implemented. It has also to prepare an analysis of risks posed by weaknesses highlighted and compare with the state of the art or standards.

3. Risk analysis: This step aims to reduce the severity of risk scenarios. It develops a risk management and ensure that all critical situations risks have been identified. We may say that it provides a risk management policy.

1.5.3 Operationally Critical Threat, Asset, and Vulnerability Evaluation: OCTAVE

It is a risk evaluation method published by Federation and Computer Emergency Response Team (CERTS). As it is illustrated in Figure 1.16, Octave is composed of

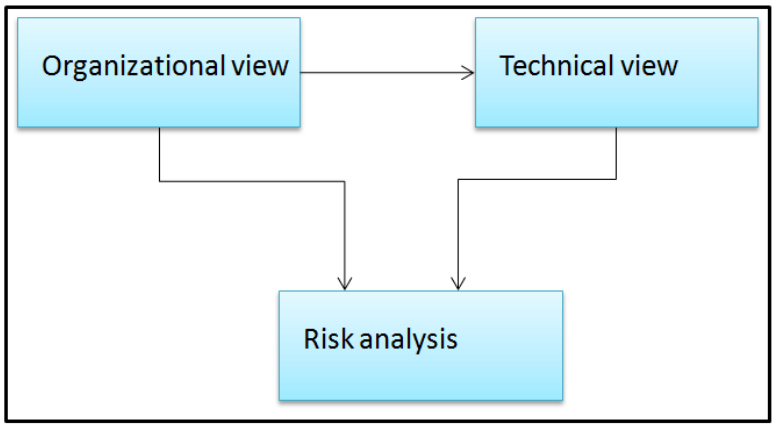


Figure 1.16: OCTAVE approach

three phases:

1. The first phase (Organizational view) identifies important resources, possible threats and security requirements for them.

2. The second phase (Technical view) identifies the vulnerabilities of the infrastructure.

3. The third phase (Risk analysis) of the method helps in the development of a security policy and its planning (protection plan and risk reduction).

1.5.4 Approaches limits

We presented in the previous section three known risk evaluation methodologies. All these approaches do not provide security but they are used as guidelines for the administrators and researchers. In our presentation, we outlined the different phases of each approach. Although the different functionalities offered by each approach, each one of them has some limits. Concerning MEHARI, it does not response to the requirements of the formalization as writing a revision of part of the system. It is a little bit complicate to be used by PMEs that do not have experts in the security area. For EBIOS, the implementation steps as they are designed reduce the flexibility

of this method and complicates its application in the field of service-oriented architectures. Moreover, EBIOS method does not offer any recommendations to simplify the tasks of the administrator as MEHARI. Finally, based on the experience of experts, OCTAVE method suffers from its complexity. We present on the following the opinion of Adam Rice, global CSO and vice president of managed security services at Tata Communications "*When it shipped, we spent hours trying to understand what it was that this package was going to do for us.*" [Vio10].

1.6 Conclusion

In this chapter, we first focused on the MOE paradigm by identifying its characteristics, functionalities, architecture and main concepts.

This study details also the behaviors of an attacker, the possible attacks and the different security services that can be damaged. This section permits to readers to know about attacks in MOE.

Finally and to prepare a helpful document for security concepts, another area more than the environment and the attacker should be known that is the security evaluation methodology that can be a way for any expert to avoid several threats and vulnerabilities. A lot of methodologies are designed in literature. In this chapter EBIOS, MEHARI and OCTAVE are presented.

After this study, we aim in the next chapter to do a survey on the security policy models. We do a comparative work among those which address the collaborative systems. This study is very crucial to choose the basic model on which is based our work.

Chapter 2

Security Policy Models

"Acquire knowledge, and learn tranquility and dignity." Umar ibn al-Khattab

Contents

2.1 Introduction

Access Control (AC) is crucial to enforce security of resources and services shared among organizations in Multi-Organization Environments (MOE). Each organization has to define a security policy that specifies the desired behavior of a user when using its resources. In the literature, there are four basic security models which are Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC) and Organization Based Access Control (`OrBAC`). Many derivatives have been deduced from these models in order to resolve a specific need. Lately, solutions for MOE based on RBAC and `OrBAC` have gained a lot of interests. This is due to the fact that these two models can manage easily larger number of users than the classical models (MAC and DAC). They also have various advantages such as their high level description of rules and their administration models.

AC modeling in MOE presents an interesting area of research. Several studies based on RBAC and `OrBAC` model [CTWS02, JVR05, NLB+05, KD06, KDBK09] have been proposed to address some interoperability security issues. We recall that MOE has its own specificities such as: (1) it is composed of an open and distributed system with various organizations, (2) the number of users contributing in this collaboration can be large, variant and unknown in advance, (3) the administration of the security policy may be done by the organization offering the resources, called O-grantor, by the organization that needs a service from the O-grantor called O-grantee, by both of them or by a specific organization to which administration is delegated, and (4) the communication technologies used in this environment may impose some extensions of models to improve suitability to them.

The aim of this chapter is to perform an assessment of role based interoperability security policies. This assessment is based on some criteria that are crucial in collaborative environments. Moreover, we conduct a complete study that compares and analyzes the advantages of several RBAC and `OrBAC` based models. It also permits to gain from the diversity of the research based on these models in order to improve the interoperability policies in collaborative environments. Finally, our analysis could lead to some proposes, some perspectives and suggestions in this domain.

The remainder of this chapter is organized as follows. Section 2.2 details basic concepts applied to security policies. In section 2.3, we present and define some principal criteria on which we are based to compare and analyze role based interoperability policies. Therefore, we study several access control models as MAC, DAC, RBAC, `OrBAC`, etc. We will close this part by giving a comparative study among RBAC and `OrBAC` models. In the next section, we study three categories of role based interoperability policies. The first category is presented in section 2.7. It deals with the centralized solutions to manage the MOE. In section 2.8, we describe the second category that is based on some technologies. Section 2.9 presents Organization To Organization approach (O2O) that represents the third category. Finally, the section 2.11 concludes this chapter.

2.2 Security Policy

Based on the Cisco systems, *"an organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. A security policy aims to ensure three security services: confidentiality, integrity and availability."*

A security policy is associated to a formal model, administration model and security mechanisms (See Figure 2.1):

-A formal model:

It aims to describe unambiguously the security policy which permits (1) to evaluate it, (2) to detect and resolve possible conflicts, (3)to abstract the security policy, (4) to manage its complexity and (5) to verify that all security objectives are covered. It is also used to test conformity and interoperability of security policies [MHA10].

- An administration model:

The administration model aims to specify who is able to update the security policy. Hence, to administrate is the action to add or remove entities and privileges. In some cases, we may have several administrators. So, the specification of their privileges is required, this task can be provided by the administration model.

- Security mechanisms:

They are the set of actions, tools and machines to implement the security policy. The different mechanisms should be selected, configured and implemented to ensure the required security services.

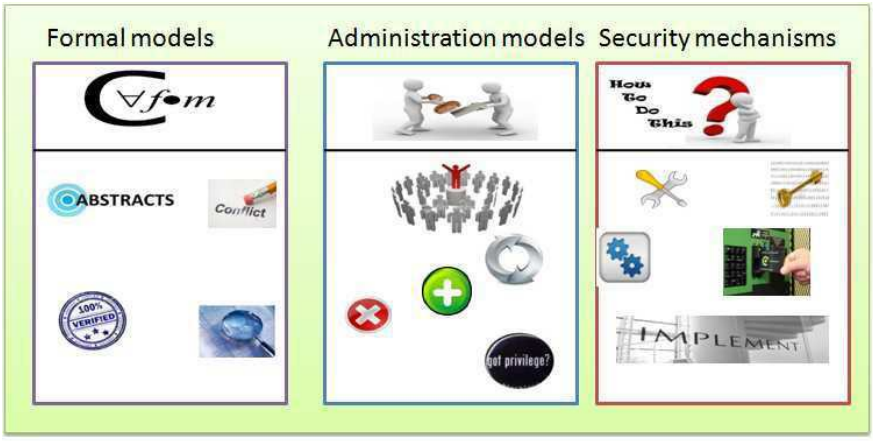


Figure 2.1: Security policy components

In the literature, many security policies proposed are based on authorization policy. This is composed by permission rules that define the legitim actions to realize. This policy is often provided by an access control solutions. The majority of access control solutions define relations applied to these three following components:

- The set of subjects that represents the different entities that require a resource.
- The set of actions that defines the possible tasks to perform in the system.

- The set of objects that contains the different resources, services that may be shared.

Therefore a simple relation in different access control model will follow this rule:

$$\texttt{Relation}(\texttt{Subject}, \texttt{Action}, \texttt{Object})$$

Moreover, a security policy may contain obligation and interdiction rules. This permits to define three types of security policies: an open, closed and mixed policy. In an open policy, we consider that all actions are permitted by default, and we have only to specify the interdiction rules. In a closed policy, all actions are prohibited by default and the administrator may add some permission rules. The third one may contain the two types of rule interdiction and permission. However, this will cause a new problem to solve. Some conflicts may be introduced by the definition of an interdiction and a permission and an approach on how to solve it should be proposed.

Several access control models are designed in the literature. In this thesis, we define three classes: classical solutions, RBAC with its derivatives and `OrBAC` with its derivatives. Before the description of these solutions, we will define in the next section some criteria that permit to compare the AC models.

2.3 Criteria to Compare Access Control Models

In order to compare the different access control models, we present two sets of criteria. The first one, inspired from [KD06, KDBK09, Kam09] is used to compare any two access control models. The second one is defined to compare models for collaborative systems. The first set contains:

Dynamism:

A dynamic model offers a security policy that takes into consideration the environment of a system. The activation of a security rule depends on environmental conditions, called also the context of a security rule. Various types of contexts can be defined in a model to improve dynamicity (e.g. temporal and spatial contexts).

Abstraction:

The model with higher level of abstraction offers a security policy which is more independent from its implementation. It will be also more comprehensive. As a result, the resolution of conflicts will be simpler.

Management complexity:

For any model, we must study the complexity of its management, which is the action to add or remove users, actions and resources from a security policy. It also defines the relationship between these entities. We also note that the existence of a management model that is homogenous with the access control policy facilitates the administration task.

Expressivity:

The diversity and different requirements of the information system has led to the development of a variety of increasingly expressive policy specification languages. For instance, these languages are not restricted to permissions anymore. A security policy

may include various types of security modalities such as permission, prohibition and obligation.
The second set of criteria is defined as follows:

Multi-organization concept:

This criterion shows the impact of the security policy model on the topology of the MOE. For instance, some models require a central entity that administrates the collaboration. This entity may not be initially present in the MOE.

Profiles management:

In a distributed environment, identities/attributes management and authentication are difficult tasks. Many technologies have been designed to solve these problems. This criterion permits to define two groups of security policy models: the first one defines these technologies and proposes extensions of the basic models (RBAC or `OrBAC`). Others ignore this problem and propose an abstract solution.

Heterogeneity:

In a distributed environment, we may have heterogeneous or homogenous partners depending on their structure and their entities (roles, activity, etc). Defining an interoperability security policy between two hospitals (homogeneous organizations) is simpler than between a hospital and a university (heterogeneous organizations). We must verify the ability of a model to create an interoperability security policy between heterogeneous partners.

2.4 Classical Access Control Models

Mandatory Access Control (MAC) and Discretionary Access Control (DAC) [FKC07] are the first models of access control:

- MAC is a centralized solution. Only the administrator controls access to any object by assignment of a label to users and objects. The left part of Figure 2.2 illustrates this process. MAC provides a security level to the user and for the object (classification). Then, authorization is a relation between these two levels. This authorization is based on two rules:

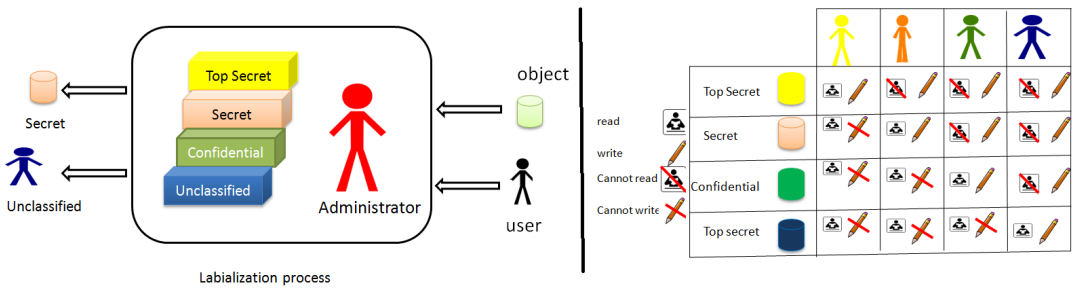


Figure 2.2: Mandatory Access Control (MAC)

  - A subject is able to read an object if its security level is equal to or more than the classification level of this object.
  - A subject is able to write in an object if its security level is less or equal than the classification level of this object.

  The table illustrated in the right part of Figure 2.2 shows the different rights of any user based on its security level and the required object security level.

- DAC is a decentralized solution. Therefore, each object is controlled by its owner. A rule is written as (S, A, O) which means that the subject S can do the action A on an object O.

These two models are static and have a low level of abstraction. As a result, it is difficult to administrate them. Also, they need a lot of memories when we have many users.

2.5 RBAC with its Derivatives

This section will highlight the RBAC model and its derivatives.

2.5.1 Role Based Access Control (RBAC)

As it is illustrated in Figure 2.3, a number of concepts are defined in RBAC [FCK95] such as role, hierarchy, separation of duties and session.

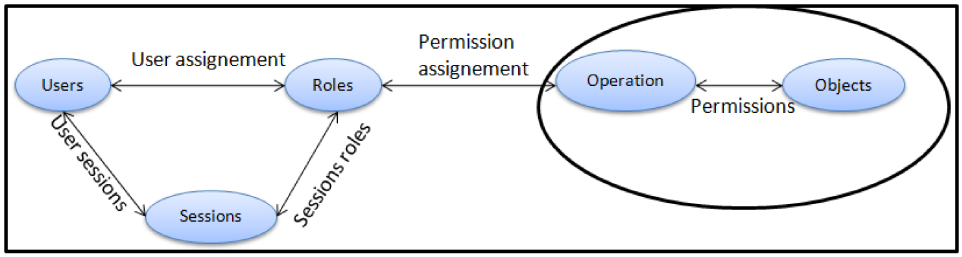


Figure 2.3: Role Based Access Control (RBAC)

Role is a set of users which can be assigned to the same security rules. For example: Frédéric, Robert and Mathieu are doctors in the hospital of EVRY; they will be able to execute same actions on any resources. Consequently, it will be simpler to assign rules to the role doctor and not to each user (see Figure 2.4).

Moreover, it is possible to assign several roles (doctor, head of department) to one subject. This model offers a higher level of abstraction than MAC and DAC. RBAC uses also the concept of hierarchy which is a relation between two roles. It offers the possibility of rules derivation. A role 'A' inherits from role 'B' means that permission given to the role 'B' is automatically assigned to the role 'A'. RBAC defines two types of hierarchy:

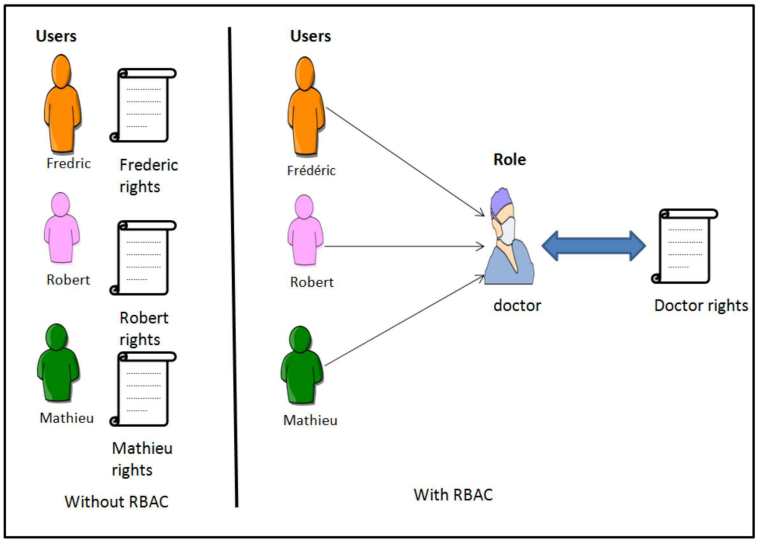


Figure 2.4: Role concept with RBAC

- *Organizational* hierarchy is a relationship between an expert and a super-expert in the same branch. The super expert can control and evaluate the tasks of the expert. RBAC offers the possibility to inherit the expert permissions e.g., head of department and employee.
- *Specification /Generalization* hierarchy is defined between role 'A' and another role 'B' if this latter can do all actions of role 'A'. For instance, role 'A' can be doctor and role 'B' can be a cardiologist. This notion facilitates the design and the management of the security policy.

RBAC also proposes two types of duties separation concept:

- Static Separation of Duties (SSD): It ensures the assignment inability of two opposite roles to the same user.
- Dynamic Separation of Duties (DSD): It offers the possibility to have two contradictory roles for one user. However, they cannot be activated simultaneously.

Moreover, the concept of session is introduced in RBAC. A session is an unique context associated with a user, within which the user activates a subset of assigned roles. Consequently, every activated role belongs to one session, and each session belongs to an unique user. Besides, RBAC has an administration model which is ARBAC. This model proposes two administration approaches. The first one is based on the hierarchy organization that means any role 'A' is able to authorize a role 'B' only if the role 'A' has a higher level of hierarchy than the role 'B'. The second one delegates all the administration tasks to one user.

To summarize, RBAC has a higher level of abstraction than MAC and DAC. Administration also is simpler (it has also an administration model ARBAC). Although, the concept SSD and DSD offer the possibility to create a dynamic policy, RBAC

does not introduce a solution to interact with the environment. With RBAC, there is no possibility to specify interdiction and usage rules. Only permission rules are not sufficient in a complex and distributed system. Besides, the hierarchy concept is still ambiguous with RBAC. Finally, it does not address the problem of collaboration. To ameliorate RBAC, various models are proposed. In the following, we detail some of them.

2.5.2 Temporal Role Based Access Control and Geographic Role Based Access Control

Temporal Role Based Access Control (TRBAC) [BBF01] and Geographic RBAC (Geo-RBAC) [BCDP05] aim to ameliorate the dynamism of RBAC.

For the same user, access to the resource can be authorized or denied in function of time condition in TRBAC. With this model, a role can be activated or deactivated based on a temporal condition.

Geo-RBAC addresses other issues: mobility and location of users. In different networks the requester may change its location, this information may cause a loss or gain of rights. The activation and deactivation of a role based on its position is addressed in Geo-RBAC. Note also that another model is designed based on TRBAC which is GTRBAC in order to solve possible conflicts in the policy. However, these solutions have not an administration model.

2.5.3 Attribute Based Access Control

Attribute Based Access Control (ABAC) [YT05] can be defined as a generalization of RBAC. Access request will be evaluated in function of the attributes which may include the role concept. Attributes are classified into three classes (see Figure 2.5):

- Subject attributes are user information (identity, address, role, etc)
- Environment attributes aim to facilitate interactions with the environment. Time, temperature and location are examples of environment attributes. They provide a dynamic and interactive model.
- Resource attributes are resource characteristics such as owner, status and maintenance.

Attributes definition depends on the application. Therefore, ABAC is a general and a simple solution. It improves dynamism and abstraction of RBAC.

2.6 Organization Based Access Control

Organization Based Access Control (`OrBAC`) [KBB+03] is becoming largely used for modeling access control policies [KI10]. It integrates various concepts defined in the previous work such as role, hierarchy, and context. Also `OrBAC` adds extension to enhance its use in a collaborative system.

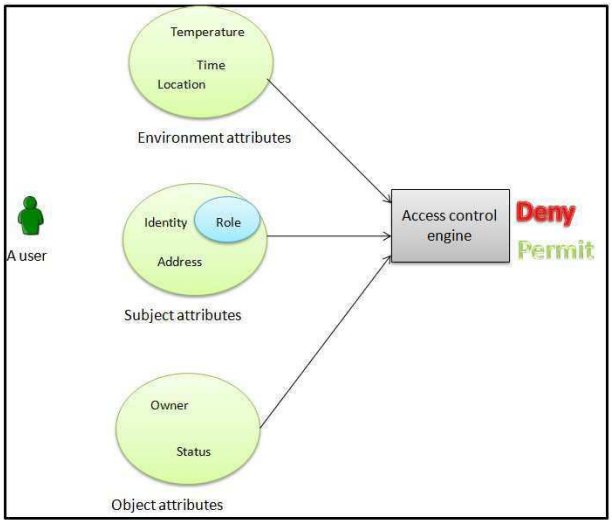


Figure 2.5: Attribute Based Access Control (ABAC)

The main concept of OrBAC is the entity organization. The policy specification is completely parameterized by the organization. This notion encourages researchers to handle simultaneously several security policies associated with different organizations. It is characterized by a high level of abstraction. Instead of modeling the policy by using the concrete and implementation-related concepts of subject, action and object, the OrBAC model suggests reasoning with the roles that subjects, actions or objects are assigned in the organization. Thus, a subject is abstracted into role which is a set of subjects to which the same security rules are applied. Similarly, an activity and a view are respectively a set of actions and objects to which the same security rules apply. Figure 2.6 describes the OrBAC model which introduces two security levels (concrete and abstract).

Security rules in OrBAC have the form:

$$\texttt{Access\_Type}(org, r, a, v, c)$$

where org, r, a, v and c are respectively organization within the rule will be applied, role, activity, view and context.

The context is a condition that must be satisfied to activate a security rule. A mixed policy can be offered in OrBAC which defines four types of access: Permission, prohibition, obligation and recommendation. Rules conflicts can appear in this policy. This problem may be solved by affecting a coefficient to each rule. Several types of contexts can be used as temporal, geographical (physical and logical), pre-request, declared, etc. Also, we may have contexts which depend on the application. The hierarchy notion which facilitates the tasks of the administrator is also used in OrBAC. In the same way as RBAC, two types of hierarchy (specialization / generalization and organizational) are defined. Moreover, this hierarchy can be used between different roles, different views, different activities or different contexts.

OrBAC has its administration model AdOrBAC. It uses the same logical formalism and the same concepts of OrBAC. As a result, OrBAC is a self- administrated model.

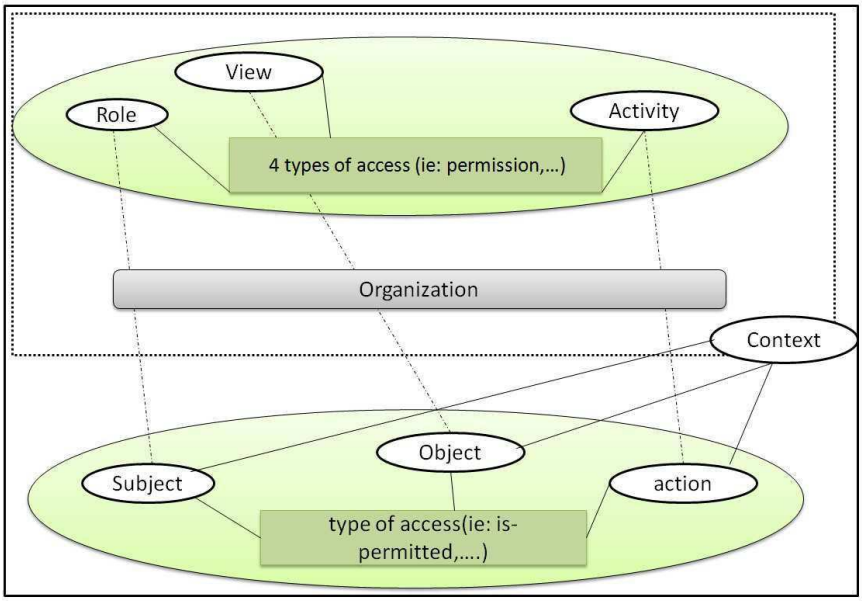


Figure 2.6: Organization Role Based Access Control(OrBAC)

Moreover, a management and policy specification tool MotOrBAC [CCBC06a] was implemented for `OrBAC`. It is designed to allow analyzing and simulating a security policy conforming to `OrBAC`. It also offers a conflict resolution and enables an administration of `OrBAC` (since it implements the AdOrBAC model). Policies created with MotOrBAC can be integrated into an application to secure it. MotOrBAC may easily be extended by adding a specific plugin.

2.6.1 Discussion and Comparison (RBAC VS OrBAC)

`OrBAC` and RBAC have gain increasing interest. `OrBAC` has many advantages over RBAC. Firstly, it has a higher level of abstraction thanks to the use of abstract entities such as: views and activities. It can also describe dynamic security policies since the concept of context is well introduced and defined. `OrBAC` proposes a formal language to express security policies. This language is based on the first order logic that allows expressing various types of security policies such as permission, prohibition and obligation and different types of contexts. This diversity may create conflicts between permission and prohibition rules that are defined in a security policy. Finally, both of them, RBAC and `OrBAC`, have their administration model (ARBAC and AdOrBAC respectively).

To summarize, we can define `OrBAC` as an improvement of RBAC according to the dynamism, abstraction, expressivity and administration criteria. For instance, in the case of an organization with a small number of users and resources, RBAC could be a good candidate. However, when we consider the need of dynamic rules and a huge number of resources, actions and subjects `OrBAC` model is recommended. Another important advantage of `OrBAC` is the MotOrBAC tool which implements

OrBAC and its administration model.

In order to achieve a Multi-Organization Environment, different derivatives of these models have been defined. Solutions based on OrBAC have several advantages on those based on RBAC with respect to different criteria such as dynamism, abstraction, administration, the definition of the organization entity and the expressivity. The comparative study between these solutions permits not only to choose the best one according to specific needs but also to improve the existing solutions. The seniority of RBAC and the experience of the researchers with it in literature may help us to propose solutions for some problems with OrBAC based models. In the remainder, we categorize and discuss the existing solutions according to which collaborative criteria , Multi-Organization concept, profiles management and heterogeneity, they can accomplish.

2.7 Models for Collaborative Systems: Need of a Super-Organization

2.7.1 TeaM based Access Control

TeaM based Access Control (TMAC) [AC04] is based on RBAC. It defines new notions to improve the collaboration criteria of RBAC. It introduces the concept of team work. This model defines a way to identify roles that contribute in a team. A user will be assigned to a team according to its role. Teams interact to accomplish a mission. However, this model suffers from several problems. First, it deals with the problem of a local collaboration; the team definition can be only used in one system. Second, writing rules for a team and a role may create conflicts (permissions assigned to a role which cannot be authorized in its team). How to solve this kind of conflicts is not clear [AC04] in TMAC. In order to improve the dynamism of TMAC, CTMAC model proposes some extensions in order to describe dynamic rules. Time, location and other contexts can be used in CTMAC. These models offer a solution to achieve a mission. They do not address the general problem that is the collaboration between different organizations. Administration in these models will be harder than RBAC since they have not an associated administration model.

2.7.2 Coalition Based Access Control

Coalition Based Access Control (CBAC) [CTWS02] incorporates concepts from RBAC and TMAC: team, task and role. It also defines the notion of organization to address the issue of collaboration between two or more systems. CBAC defines two new notions: Organization Coalition (OC) and mission. OC is a set of organizations. The concept of mission is defined as the task to be performed in an organization. A relationship must be defined between OC and missions. However, these new concepts complicate the management task since it has not an administration model. Figure 2.7 illustrates a request sent from a user of organization OrgB to organization OrgA. This request must contain a coalition level, a local role (role level), the related

credentials (certificates) and the requested service. After receiving this request, the O-grantor extracts the needed certificates related to the role level of this coalition. This model is also improved to simplify the creation of dynamic coalitions. This new function is realized by a specific entity which will be able to advertise a shared service and the coalition level policies. Then, any O-grantee must provide the needed credentials to a specific component called 'Mapper layer' to benefit from this new service. However, this model suffers from some drawbacks. First, it has not an administration model. Next, it needs a third entity which will be able to identify the coalition and the different partners in a mission.

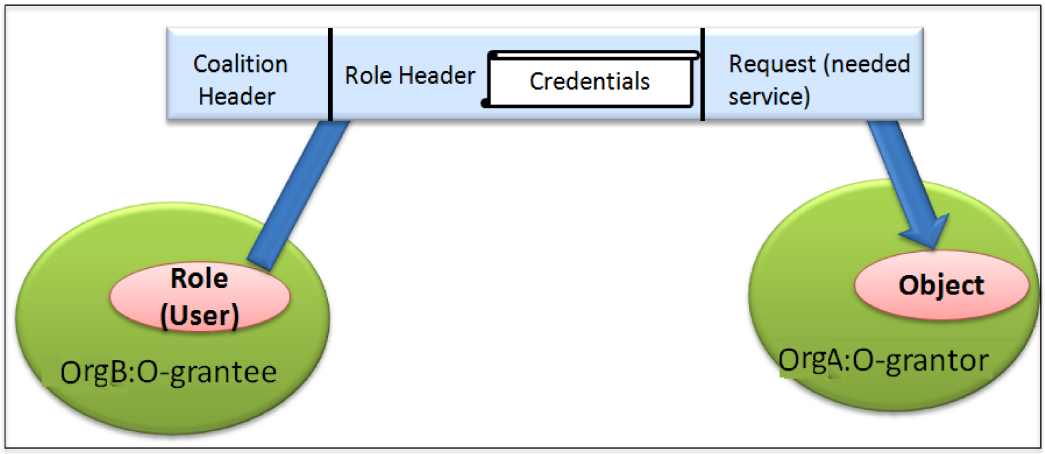


Figure 2.7: Coalition Based Access Control(CBAC)

2.7.3 Multi-OrBAC

Multi-OrBAC [KD06] is based on two main ideas:

- Redefinition of `OrBAC` concepts: role, view, activity and context.
- Introduction of super-organization entity.

Two presentations are proposed to encourage its use: software engineering presentation based on Unified Modeling Language (UML) [DP06] which facilitates the integration of security policies on the different steps of implementation. Also, a presentation on constraint logic programming is involved with Multi-ORBAC to verify security properties such as consistency and completeness.

In Multi-Organizations context, user can have roles from different entities. Also, the user must be able to execute an action on an external resource (which can be defined in different organizations). Multi-OrBAC deals with this problem by the introduction of the following entities: RIO (Role In Organization), AIO (Activity In Organization), VIO (View in organization) and CIO (Context In Organization). The entity RIO defines a relation between one (and only one) role and one (and only one) organization, similarly, for the others AIO, VIO and CIO entities. As it is

illustrated in Figure 2.8, a security rule in Multi-OrBAC corresponds to a relation between a role from org B (R2IOrgB) and an activity (A1IOrgA), a view (V1IOrgA) and a context (C1IOrgA) from OrgA. It is important to mention that view, activity and context must belong to the same organization. Any User can have several roles in one or more organizations, for example: the doctor Philippe can be assigned to the roles doctorIHoP and doctorIHoE simultaneously, where HoP is hospital of Paris and HoE is hospital of Evry.

The definition of these concepts simplifies the collaboration between organizations. Security rules with Multi-ORBAC are defined as a relationship (permission, prohibition or obligation) between an RIO, VIO, AIO and CIO. Accordingly, to define a security rule, the administrator must have a global view on the different parameters from different organizations. Therefore, it will be difficult to assign the administration of security policy to one of the participating organization. Therefore, Multi-OrBAC defines a parent organization (super-organization) which has a global view of the environment. The latter administrates the global security policy that must be consistent with local policies of each organization. Indeed, Multi-ORBAC provides two separable security policies (a global and a local one). Two scenarios may be realized to solve a probable conflict between these two policies. In the first scenario, the super-organization modifies the global policy to resolve the conflict. In the second scenario, it modifies the local policy, if it is allowed by the concerned organization. This model proposes a central solution to the problem of interoperability security policy in a distributed system.

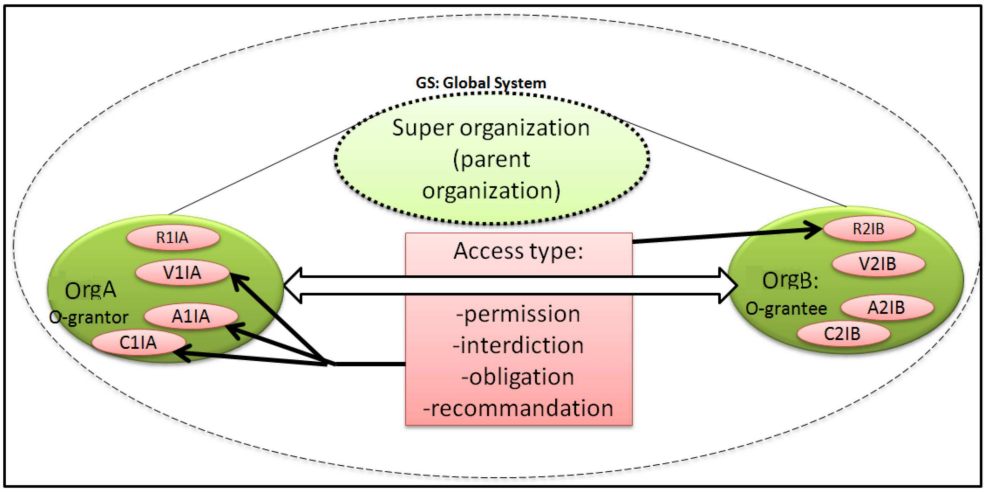


Figure 2.8: Multi-OrBAc model

To summarize, Multi-OrBAC offers many advantages. The O-grantor and O-grantee delegate the creation and the administration of the interoperability security policy to the super-organization which facilitates the elimination of ambiguity and the

resolution of conflicts. However, it suffers from several drawbacks. The overall architecture of the system and the use of a super-organization require a global view which is not always possible. Furthermore, Multi-OrBAC does not detail how to choose the super-organization and what kind of relationship we can have between the super-organization and the other members. Multi-OrBAC will be easily used when the collaboration is between two sub-organizations (or two teams from the same company). Otherwise, it will be difficult to have a common parent organization. So, we conclude that the different organizations must be homogenous. Also, this model will have a huge increasing numbers of roles, activities and views even if there is compatibility between different organizations.

2.7.4 Discussion

The basic idea of these three models(TMAC, CBAC and Multi-OrBAC) is the presence of a third entity that simplifies the creation of a global policy. This proposition provides an efficient solution to resolve probable conflicts when writing the interoperability rules. In Multi-OrBAC and TMAC, we need a super-organization which is in charge of defining the security rules, teams and missions. Thus, this administrator must have a global view on the two partners to define entities in each one of them which may not be allowed by one of the partners. As a result, the use of these solutions can be advised when we deal with local communication between two sub-organizations or teams. In Multi-OrBAC, the administrator may need to define a huge number of roles, views and activities even if we have defined a mapping (compatibility) between the different entities. In CBAC, a third entity is also required to assign identifiers to the different participants and to the coalition.

These solutions transform the distributed environment to a central one and facilitate the creation of the interoperability policy. They, also improve the abstraction level of `OrBAC` and RBAC in a Multi-Organization Environment. On the other hand, they complicate the administration of the security policies (huge number of entities even if we have compatibility roles). Only, CBAC can be used between heterogeneous organizations. Finally, the profile management criterion is not well detailed in all the above models.

2.8 Models for Collaborative Systems: Solutions based on some Technologies

This type of solutions depends on the communication technologies used by the partners. In this section we detail two models which are based on `OrBAC`.

2.8.1 Poly-OrBAC

This proposition addresses distributed system based on web services technologies which is a software system designed to support interoperable machine-to-machine interaction over a network (W3C definition). This is one of the most used technologies

in a distributed system to share services: standards and open protocols are designed and implemented on various platforms. Poly-OrBAC [6] proposes to integrate web services with OrBAC. It defines two main steps before the collaboration:

- The creation of a web-service.
- The integration of new rules based on two new concepts: Virtual User (VU) and Image Web Service (IWS).

During the first step, the O-grantor must create at least one web service to control its resources. Then, a WSDL description (Web Service Description Language based on XML) must be provided and published in UDDI server (Universal Description Discovery and Integration) which acts as a directory to list and to inform users of the services offered by the organizations. The second step explains how OrBAC will be used with web services. This step requires modifications in the local security policies of the two organizations to integrate the concepts VU and IWS. In the O-grantee, we must first define a view V_IWS which contains an object IWS with an activity invocation. Then, a permission rule for (any existing or a new) role 'R' to invocate the V_IWS must be added. As a result, any user assigned to this role 'R' can apply for using the shared resources. In the O-grantor, VU is a virtual user that can execute a web service. So, a role R_VU must be created. This role will be assigned to the VU with an activity to control the web service. Then a permission rule will be given to this role to authorize the execution of the WS.

With Poly-OrBAC, the two organizations contribute in the writing of rules. In fact, each one of them adds rules to collaborate. In the O-grantee, we must have new rules that can invocate the V_IWS. Moreover, new roles and rules will be added to control the access of the web service in the O-grantor.

A contract detailing the functional requirements, funding and quality of service and policy access must be negotiated before the integration of rules. Poly-OrBAC is apt to check in real time the coherence and the compatibility of the policy with requirements. For each organization, security policy will be written with timed automaton and it will be installed in a specific gateway. With these automata, Poly-OrBAC checks interactions and ensures the proper functioning according to the signed contract in real time.

Based on the architecture presented in Figure 2.9, we detail a simple scenario (request/response) to clarify the global process. We assume that the above discussed steps are realized and an existent user C1 from the O-grantee needs the service provided by the WS of the O-grantor. Firstly, the O-grantee verifies the associated access of this user to invocate the IWS. If the request is authorized, an invocation message is sent to the gateway G1 which signs and sends the message to the gateway G2. Then, G2 analyses and checks the contents of this message. The O-grantor evaluates the request to deny or authorize the access. We note that, the organizations use the Simple Object Access Protocol (SOAP) [KDBK09] to exchange messages.

To conclude, this new model offers and details different steps to use OrBAC with web service technologies. Poly-ORBAC benefits of the advantages of web services. This approach offers a simple architecture which can be used with different platforms.

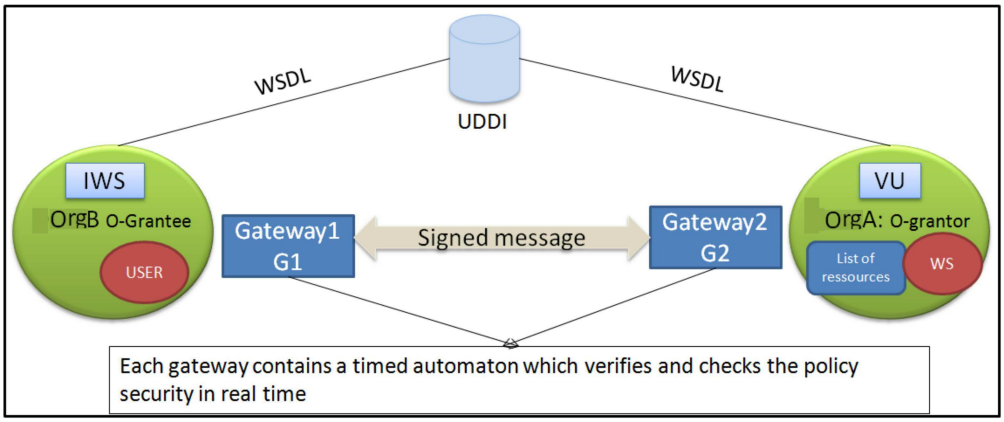


Figure 2.9: Poly-OrBAc model

Besides, the use of standard web services facilitates and encourages the choice of this model. Poly-OrBAC can be used between different heterogeneous organizations. However, it does not detail the problem of privilege management. Moreover, the management of various users and resources will be complexes (invocate a service is equivalent to one action, we will need various web services to manage the different actions).

2.8.2 OrBAC in Virtual Organization

Access control is one of the main issues in VO. In this section we present the proposition detailed in [NLB+05]. In this work, the authors propose an extension of `OrBAC` in order to be used with Federation Identity Management (FIM) and Privilege Management Infrastructure (PMI). We describe in the following these technologies proposed for this extension:

Federation Identity Management

FIM [oi07] is a mean to delegate the authentication to multiple distinct systems. FIM is a way to share identity information between different organizations. To create a FIM, we need the creation of a trust circle between the different participants (see Figure 2.10):

- An IDentity Provider(Idp) manages a set of users: It stores their identities and provides an authentication service.
- A Service Provider (SP) provides services to access shared resources and refers to (Idp) for authentication.
- A circle of trust groups is a set of SPs that accept to consolidate all their users together. It contains at least one identity provider and at least one service.

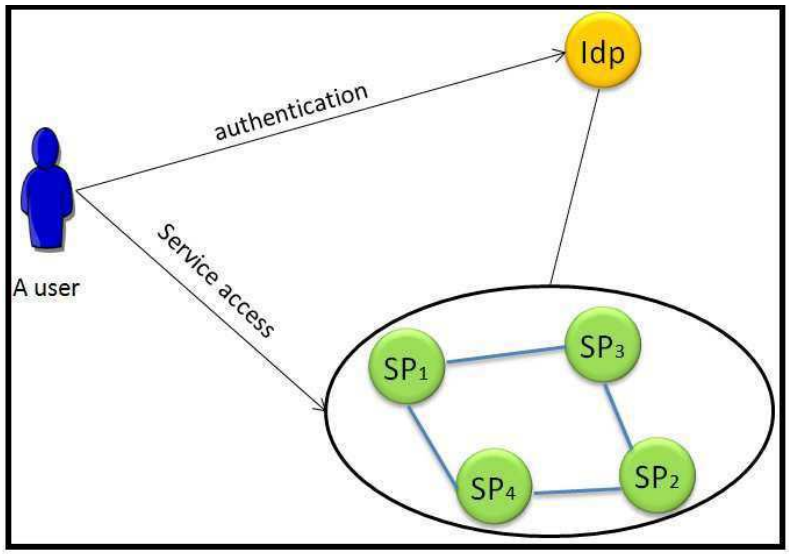


Figure 2.10: Federation Identity Management

The use of FIM does not require an intervention of the administrator to modify the used technologies of authentication in the VO. With this solution, an organization controls only the authentication of its users and not the external ones. Also, FIM ensures the spread of identity with other partners. In [NLB+05], they choose Shibboleth [MCC+04] which is an extension of Security Assertion Markup Language (SAML) [Kam09] to implement federation identity management. The authentication of users is delegated to their home organization.

One of the drawbacks of the FIM is its inability to answer the following question: "Who has the right to do what?". Therefore, in VO, the integration of the FIM with the PMI will be the right solution to manage access control and the identity of participants. In fact, PMI gives authorization in function of the attributes of the user and not only based on the identity. eXtensible Access Control Markup Langue (XACML) [GMA+03] can be presented as a PMI.

PMI: XACML

XACML (eXtensible Access Control Markup Language) [GMA+03] is an OASIS standard for access control. It offers an XML based language to specify access control policies, requests, and responses in distributed computing environments. It allows and simplifies the interoperability between different authorities domain. This framework is based on some important elements (Rule, Policy and PolicySet) and different entities.

A policy is a set of rules. This policy will be enclosed on a PolicySet which contains multiple interrelated policies and solves the expected conflicts by policy combination algorithm.

Figure 2.11 illustrates the basic entities that are:

- Policy Administration Point(PAP) which writes and adds the appropriate pol-

icy, rules or policy sets to the policy library.

- Policy Decision Point(PDP) extracts policies from the policy library to evaluate and return the XACML decision which can be Permit (access allowed), Deny (access refused), Indeterminate (missing value, error occurred, etc) and Not applicable (no existing policies to deal the request).
- Policy Enforcement Point(PEP) that intercepts the user's service access request, and then converts the original message into XACML format request with the abstract and concrete entities after exchanging information. This message will be forwarded to the PDP.

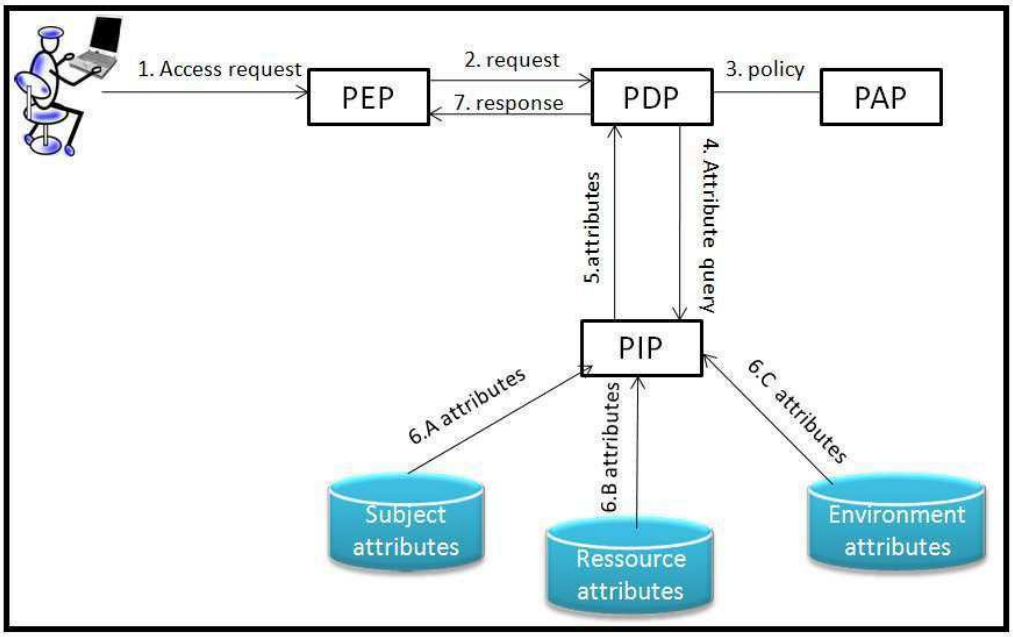


Figure 2.11: XACML architecture

Different profiles of XACML are proposed to offer an easier way to administrate, to abstract, to interact and to be more flexible with the needs of the application and environment. The most important are SAML, RBAC and `OrBAC` profiles [HCBD09].

The integration of a PMI to the identity federation is to complement the functions of identification and accreditation offered by the federation. One of the solutions may be the use of Shibboleth which is based on SAML and allows you to delegate authentication and spread of user attributes with XACML. These technologies are chosen to facilitate the collaboration by avoiding modification of the local authentication/authorization technologies. The use of these two technologies (FIM and PMI) and the definition of Virtual Organization impose extensions on the `OrBAC` model. New notions have to be defined [Kam09], among them:

- MemberOrg(vo1,OrgB): Organization OrgB is a member of the virtual organization vo1.

- ActiveIn(vo1,OrgB): Organization OrgB is active in the virtual organization vo1.
- ExRole(orgA, r1, OrgB, vo1): Organization OrgA exports the role r1 to the organization OrgB in the context of virtual organization vo1.
- InteractionVO (vo1, OrgB, OrgA, r1, a1, v1, c1): An interaction can be done between organizations OrgB and OrgA in the context of the Virtual Organization vo1; the role r1 (it must be exported from OrgA) can execute the activity a1 on the view v1 according to the condition c1.

The principal idea in this solution is the concept of importation and exportation of roles. The O-grantor in VO can export role r1 to other organizations in the same VO by the use of the predicate ExRole. The importation of role r1 is done with the relation interactionVO which defines the specific parameters (action, view and context). This phenomenon is illustrated in Figure 2.12.

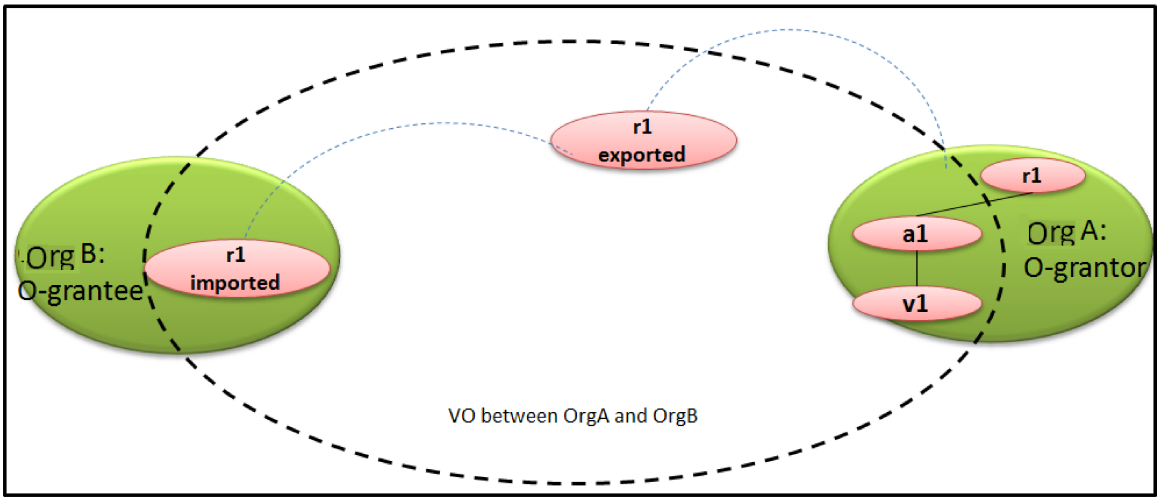


Figure 2.12: Importation and exportation of roles.

To conclude, this solution can be used between heterogeneous organizations. The different partners participate in the writing of rules. VO designer deals with the problem of privileges management (sharing of attributes and identities) and the new concept VO. Consequently, Extensions on `OrBAC` (ExRole, InteractionVO, MemberOrg, ActiveIn) are needed to be integrated in this solution. Similarly as Poly-OrBAC, the abstraction level is lower than `OrBAC` since it depends on some specific technologies.

2.8.3 Discussion

These two models propose steps to extend the `OrBAC` model in order to fit some communication technologies. As a result, these models benefit from the advantages of these technologies (standardization, portability, etc). With these models, the creation of a policy will be realized in a distributed way. Each organization defines

the security policy to manage the access of external users to their resources. However, these models have a lower abstraction level since they depend on these technologies. Thus, they cannot be generalized to any Multi-Organization Environment. Only, the second model deals with the problem of privilege management. Moreover, with Poly-OrBAC, the management of various users and resources will be complex (invocate a service is equivalent to one action, we will need various web services to manage the different actions).

2.9 Organization to Organization

Organization to Organization (O2O) [CCBC06b] is based on two concepts that are:

- Virtual Private Organization (VPO): It is created by the O-grantor to define interoperability rules with only one O-grantee. As indicated in Figure 2.13, the O-grantor OrgA creates two VPOs OrgB2OrgA and OrgC2OrgA to collaborate with organizations OrgB and OrgC respectively. OrgB2OrgA (respectively OrgC2OrgA) controls only the access of users from organization OrgB (respectively OrgC). In O2O, local policy in the O-grantor is used to control the access of local users. A VPO is created when the organizations start the collaboration and it will be deleted in the end of the coalition.

- Role Single Sign On (RSSO): This concept offers the possibility to keep the same role of a user in other organization. The authentication of the role will be done only once in the parent organization. As a result, the collaboration will be simpler.

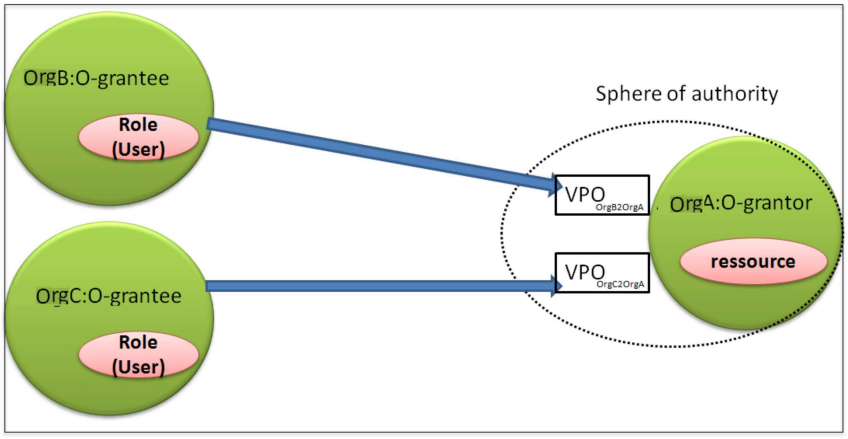


Figure 2.13: VPO in O2O.

O2O defines several steps to automatically create the interoperability security rules. The first step is the creation of the contract, which will be based on the derivation of the interoperability policy from the local policy of the O-grantor. The

O-grantor defines a list of classes (each class identified by Classname which is a set of entities) that can be shared.

Can_be_mapped (O-grantee,O-grantor,Classname) is a rule that will be added in the contract to specify the shared classes. To simplify the class mapping, only important class attributes must be mapped. Indeed, O2O defines three types of attributes: simple attributes, `key` attributes and `key key` attributes. Only the mapping of `Key key` attributes is compulsory. Moreover, O2O defines new predicates to simplify the creation of the VPO, among them:

- restrictionView (V, Sub_V) : if the O-grantor does not accept to share a global view V with the O-grantee, it can define a sub-view sub_V using the predicate restrictionView. Any access affected to the View V will be only assigned to the new view Sub_V in the VPO.
- underivable (O-grantee, Sr) : Sr is a security rule that cannot be derived in the VPO. Sr is designed only to the local users.
- Exception (O-grantee, Sr): the VPO contains new rules which are not applied in the local policy. The predicate exception defines these security rules.

The second step is the definition of compatibility relations. In O2O we have four types of compatibility which are total, partial, symmetric and non-compatible [CCBCC08a]. These relations will be used to derive rules. To establish the compatibility relation, O2O relies on the notion of ontology which is a representation of system knowledge. The ontology ensures a mutual comprehension among organizations.

After this step, O-grantor derives rules and obtains the policy security of VPO according to this equation:

$$Local\ security\ policy\ +\ Compatibility\ +\ Contract\ \Longrightarrow VPO\ Policy$$

A possible conflict among rules of the interoperability and local policy can appear. The resolution of this problem is kept to the administrator. This solution aims to help the O-grantor to write the security policy of the VPO. In addition, it is not intended for a specified group of organizations. However, it can be used with heterogeneous or homogeneous ones. In fact, O2O offers three types of management: decentralized (only by the O-grantors), centralized (by a trusted server) and hybrid (centralized and decentralized) [CCBC06b].

O2O automates and facilitates the creation of interoperability rules. It offers an abstract model which can be used with various architectures. However, O2O will be a bit complex for the administrator, if we cannot have a compatibility relationship between the different organizations.

2.10 Discussion

In literature, we have found four solutions based on `OrBAC` which address the problem of interoperability security policies for access control: Multi-OrBAC, Poly-OrBAC,

OrBAC in VO and O2O. Each approach proposes extensions of OrBAC and addresses one or more challenges that we had discussed in the sections 2.7, 2.8 and 2.9. We remark firstly that interoperability security policies based on OrBAC do not propose enhancement on OrBAC according to dynamism criteria.

Secondly and considering the abstraction level, OrBAC in VO and Poly-OrBAC depend on some specific technologies. As a result, they benefit from the technologies characteristics (standards, implementation, etc). Then, interoperability policy with a specific technology will be simpler than a general one. However, Poly-OrBAC and OrBAC in VO will have a lower abstraction level than OrBAC.

Concerning the administration and creation of interoperability rules, O2O simplifies these tasks. An automatic interoperability security policy will be created but this task will be complex if the different partners are heterogeneous (mapping and deriving rules will be difficult in this case). Moreover, Multi-OrBAC has a simple administration task thanks to the super-organization concept. However, this particular entity converges the problem to a central one.

2.11 Conclusion

In this chapter, we focus on the state of the art of the access control models. Different solutions are designed as the classical solutions (MAC and DAC), RBAC derivatives and OrBAC derivatives. In our study, we tried to present different examples of these models to acquire a clear vision of these models, their advantages and challenges.

The second part of this chapter aims to analyze and present different models developed for collaborative systems. Solutions as Poly-ORBAC, Multi-OrBAC and O2O are detailed.

At the end of this work, we consider that OrBAC may be the best choice in our case to be used in MOE for different raisons: The organization concept and its modelization are well defined in OrBAC. Based on different criteria, OrBAC tries to take advantage from the seniority of RBAC and the experience of the researchers with it in the literature. The context notion and its diversity types offer a more dynamic policy. The high abstraction level simplifies the administration task. Moreover, its administration tool MOTOrBAC offers several functionalities as the simulation of the concrete security policy, the edition of the policy and the resolution of conflicts.

After this survey, we are convinced that we may design our solution based on the OrBAC model. We will study in the next chapters how to adapt this framework to address more security challenges and specially the trust concept.

Part III

Trust Framework in MOE

Chapter 3

A Trust Framework for MOE Environments

"A man who trusts nobody is apt to be the kind of man nobody trusts".
Harold MacMillan
"Trusting is hard. Knowing who to trust, even harder".
Maria V. Snyder, Poison Study

Contents

3.1 Introduction

In previous chapters, we had investigated the different security challenges in MOE and studied the security policy models defined in the literature. This study encouraged us to work more in trust management and access control areas. In this chapter, we will detail our first contribution that can be divided into four parts:

- The definition of a new trust approach that includes (1) the definition of the trust parameters for MOE, (2) the evaluation of these parameters (3) the creation of trust vectors for users and organizations. Figure 3.1 illustrates the basic concepts of our proposal.
- The integration of our trust approach into the `OrBAC` model that forms our new model `TRUST-OrBAC`.
- The proposition of checking integrity rules for any trust model in MOE.
- The design of a possible architecture of MOE integrating our solution.

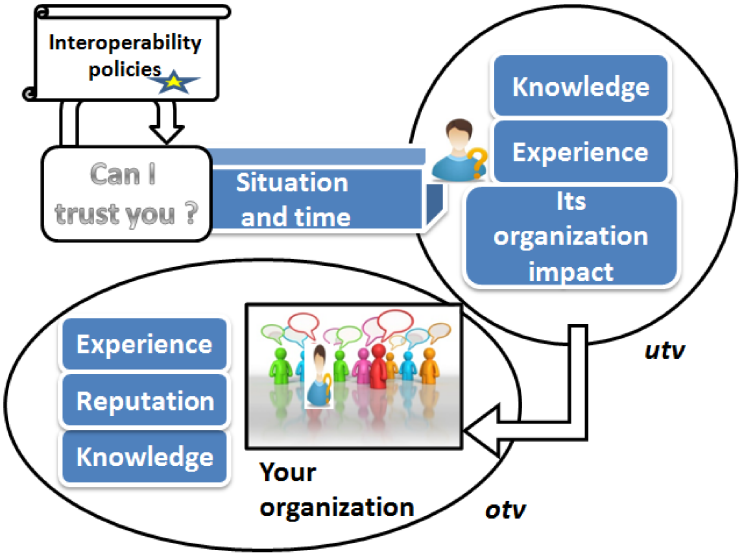


Figure 3.1: Our trust model in MOE.

In our approach we introduce two types of *trust vectors*, the first one is related to users (*utv*) and the second one is related to organizations (*otv*). The different parameters that can contribute to form a trust level of a user and an organization in MOE will be studied and integrated during this chapter. Another main contribution of this solution deals with providing an evaluation method for each parameter of these vectors. In fact our evaluation techniques *are dynamic* which means that the evaluations operations depend on time. Therefore, the trust in our proposal

is a relation among two entities (the trustee and the truster), related to a specific behavior of the trustee (situation), in a specific slot of time.

Within this notation, some new security properties can be represented as follows:

> *If an organization* $\texttt{org}_B$ *is assigned to a low trust level value regarding another organization* $\texttt{org}_A$*, then this fact affects the trust level of the users of the organization* $\texttt{org}_B$*.*
>
> *A user might loose some rights if he or his organization performs bad behaviors, since their trust levels are not static.*

The rest of this chapter is structured as follows. The Section 3.2 studies the different challenges of a trust in MOE and surveys the existent frameworks. In Section 3.3 some preliminaries MOE concepts are introduced. Afterwards, in Section 3.4 our new trust framework is presented and a complete detail of the functioning principles is provided. A formal definition and evaluation of the parameters (experience, reputation and knowledge) are presented. In Section 3.5 some rules to check the integrity of any trust solution in MOE are defined. Afterwards, Section 3.6 and Section 3.7 detail respectively the construction of the trust vectors and the different steps to integrate our trust solution into `OrBAC`. In Section 3.8 we present a case study and finally, a possible architecture that integrates our framework is presented.

3.2 Trust in the Literature

3.2.1 Trust Challenges

In this section we present the different issues addressed by a trust framework in the literature:

Definition of trust - Ambiguity:

Several researchers argue that the definition of trust is not a simple task.

- *Trust definitions have become a "confusing potpourri"* [Sha87].
- *We argue that trust is appropriately difficult to define narrowly* [MC96].
- *Trust is a term with many meanings and perceptions.* [Wil93]
- *These different perceptions make it extremely difficult to navigate through the literature on trust. There are also many theories on trust, some of which diverge from each other only in their identification of the grounds on which it is based.* [IM04]
- *It is widely acknowledged that trust is complex and multidimensional.* [PS05]

Different definitions of trust exist in many area as philosophy, economic, computer science and medical. Moreover, we find several definitions in the same area that do not converge to the same concepts. For example, we cite:

- *Researchers also suggest that trust can be viewed as an attribute of risk-taking behavior.* [MDS95]

- *Trust is defined as "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context (assuming dependability covers reliability and timeliness).* [GS00]

Thus, in this thesis, we firstly focus on proposing a clear definition of the trust concept based on the MOE requirements. This includes the definition of its parameters, their evaluations, the related algorithms and the experiments.

The Trust parameters:

Several trust solutions exist in the literature. However, each one of them evaluates a set of parameters that are needed applied to the addressed environment: some solutions are based on the exchange of attributes [BFL96, WTS+02], others are based on combined parameters in P2P networks [NZ09]. On the other hand, some existing approaches are based on the feedback of the different entities in web services network [CZC09], etc. As a result, a fundamental issue is still open, which is, what are the trust parameters that can be evaluated in MOE environments.

Trust presentation:

How to present trust is another issue to resolve by the different trust frameworks. Some models present the trust level by one value [LLYT05, WLWV09a] that can be a continue [AAJ12] or a discrete variable [ARH97]. Moreover, other approaches may present trust by a vector [RC04] or a matrix [NZ09].

Recommendation:

Different models studied this concept to evaluate trust [JIB07, JGB07, CZC09, HCJ10, HLHV10, SCW12, AAJ12]. Trust is based on the different feedbacks of the participants. Several questions are related to its evaluation:

1. How to avoid the false feedback problem? [XL03, JGB07]

2. How to combine the different values? During the collection of feedback, we may receive different values, and sometimes they are contradictory. For example: the same user may be seen as a trusted entity in an organization $\texttt{org}_\texttt{A}$ and a malicious node with another organization $\texttt{org}_\texttt{B}$. Therefore, we have to find a solution in order to combine them and to determine only one trust level.

3. How to share this value?, how to understand the received feedback? and how to confirm that the received value has the same meaning between the receiver and the sender? (i.e. 0,5 may be defined as a trusted level with $\texttt{Org}_\texttt{B}$ and untrusted one with $\texttt{Org}_\texttt{A}$). To solve these problems, some solutions define new communication protocols, others ignore this problem and others propose to use an ontology approach to solve it.

Forgetting factor:

Based on the wikipedia [wik13], forgetting refers to apparent loss of information already encoded and stored in an individual's long term memory. This factor is used with trust in order to not use the old evaluated interactions. Different approaches use

this factor for their evaluations as [CWZG07,SZL08,ZC13]. In our case, we provide a method that permits to *forget* some old values.

Trust Bootstrapping:

How to choose the initial value is discussed in different trust frameworks. In [HM05,JLK+05,LV10], the trust bootstrapping is based on a constant values. In [AMH+10], an adaptable initial value is proposed. First, an initial trust model categorizes services in different security levels based on their security needs. Based on these security levels, a security factor is generated. The initial trust value will be evaluated depending on the security factor. In other solutions as [BP05], the trust bootstrapping is only based on the digital signature. TRULLO (TRUst bootstrapping by Latently Lifting cOntext) [QHC07] provides the initial trust values based on the past recommendations. However, it cannot bootstrap trust when there is no historical data about the user.

Architecture:

There are two topology types of the trust framework architectures:

- A centralized framework where a central node will be delegated to monitor all the communications, analyzes the historical data and evaluates the trust level of the different systems. Some problems will totally disappear with this architecture such as the sharing of the trust level and the coordination between the different partners for the trust evaluation.

- A decentralized framework, where each node can have the role of a truster and a trustee. Each one may evaluate the trust level of any other entity. A trust module has to be installed in each node. This architecture avoids the congestion of the trust evaluator and simplifies the hard task for this entity. In MOE, we are interested in this kind of architecture.

Time in trust:

Time is a critical parameter in the trust concept. Trust level of an entity may change during the collaboration. Therefore, this concept has to be well defined and considered in order to be integrated in our trust framework.

After studying the different trust challenges, we present on hereafter a comparison study between the different families of trust approaches. In this chapter, we divided the different solutions into three classes that are policy based approach, monitoring based approach and hybrid approach.

3.2.2 Policy Based Approach

This approach is based on two concepts:

- A language that allows to express the different attributes, the actions to perform and the different conditions to establish trust.

- The inference rules that allow us to obtain a trust level between the entities.

Hereafter, some detailed solutions belonging to this class:

RT: A Role-Based Trust-Management Framework

In [NM03], authors propose a new framework that is based on the Attribute Based Access Control (ABAC). This model, called RT, aims to improve previous trust managements (as Keynotes and SPKI/SDSI) by supporting the separation of duties and offering a more flexible delegation mechanism. The concept of trust is constructed by the use of credentials. This latter permits to express the trust with the exchange of the role ownership between the different entities. Moreover, the system provides mechanisms to associate attributes to entities, to delegate attributes to others, to reason about these attributes qualifications and to evaluate the trust level. As it is shown by Figure 3.2, several families of RT framework have appeared:

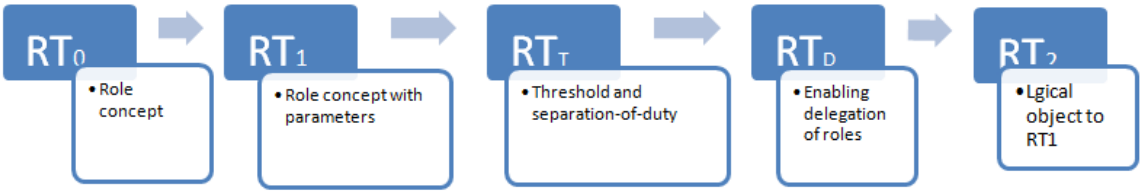


Figure 3.2: RT family.

- RT_0 that is based on the role concept as a parameter of an entity
- RT_1 where different parameters are added to the role of an entity.
- RT_T expresses threshold and separation-of-duty policies by adding multiple roles.
- RT_D provides a mechanism for enabling delegation of roles.
- RT_2 adds the logical object to RT_1 and allows reasoning about the attributes.

TrustBuilder

TrustBuilder [WTS+02] is designed and implemented in the Internet Security Research Lab at Brigham Young University. It is a prototype system for negotiating trust across organizational boundaries. A security agent in each participant will be the responsible of the negotiation management. As RT framework, this approach is also based on ABAC, it offers different negotiation strategies. Its architecture is composed of:

- A credential verification module that performs a validity checking, signature verification, revocation checking, and credential chain discovery.
- A policy compliance checker that translates the credentials from a neutral format, such as XML, into statements in the policy language.

- A negotiation strategy module that builds trust based on a sequential disclosing of credentials. Building trust is related also to the policy disclosure. Since some of them may contain sensitive information that should not be disclosed with untrusted entities.

Finally, this prototype aims at building trust between two entities to protect the requester by negotiating the different attributes to send and to protect the service provider by hiding some policies.

XeNA

XeNA [HCBCD09] is a framework having the same objective as a trustbuilder. Moreover, it provides more services.

XeNA prototype is implemented as an extension of TrustBuilder2 version 0.1. XeNA is based on XACML and `OrBAC`. As a result, it gains from their benefits. XeNa defines a new negotiation methodology based on resource classification within an extended XACML architecture. Three classes are proposed: resource with direct access (without a negotiation process), resource with direct negotiated access (the policy that manages them is not hidden) and resource with indirect negotiated access (the policy that manages these resources should be kept secret). We mention also that a new exception module to treat unresolved negotiation is designed. In the following we present the XeNA architecture.

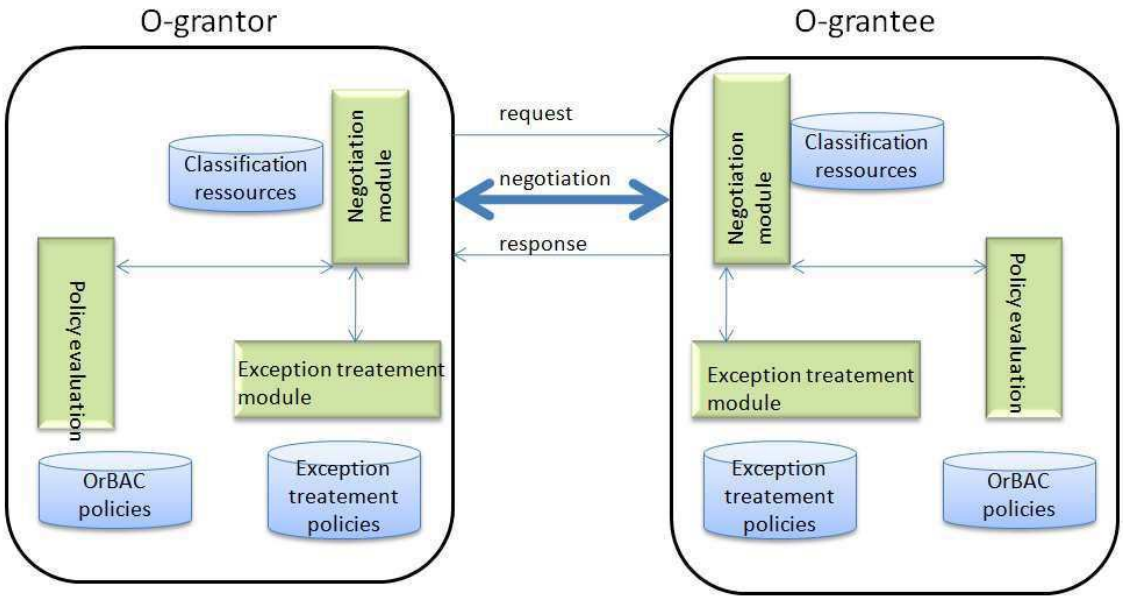


Figure 3.3: XeNA architecture .

3.2.3 Monitoring Approach

Solutions belonging to this class are based on monitoring solutions in order to evaluate the trust level of an entity. We present on the following three solutions of this

class.

A Distributed Trust Model

One of the famous and basic model that focuses on the monitoring based approach is [ARH97]. This solution aims to give a general structure for evaluating trust and reputation in a distributed system. Therefore, different researchers follow their ideas with different updates and modifications related to the addressed environment [CDdV+02, YV03].

This approach provides a distributed recommendation based trust model: A trust management with a recommendation protocol is designed. The authors provide a transitivity trust evaluation based on some conditions. This solution defines two types of trust: direct trust and indirect trust (recommendation). The recommendation protocols aim to send a request for a recommendation value and a response containing the trust evaluation. However, this approach suffers from its unclear formal definition which is considered the main limitation of this method in MOE.

Beta Reputation System

Josang and Ismail [JI02] have designed and developed a beta reputation solution which uses a beta probability density function. Their system defines a method to combine the different feedback from other entities and to provide the resulting trust level. Another concept which is well detailed in this approach is the forgetting factor. In their paper, they prove that reputation values based on a limited duration is better than the one based on the infinite longevity feedbacks since an entity may modify its trust level after a period of time. This system defines a simple and flexible approach for the reputation evaluation system. However, a centralized architecture is needed with this framework.

TRAVOS

TRAVOS [TPJL06] approach is the most similar to the Beta reputation system. It aims to improve some concepts of the Josang approach [JI02]. This solution is based on the past interactions with a node and also on the reputation gathered about it. With this approach, the different evaluations of the past interactions are used when this entity is not a new collaborator. Otherwise a reputation evaluation gathered from other partners about this entity will be done. The Josang approach [JI02] does not consider this separation between the direct and indirect information. Moreover, it assumes that the most of the reputation values are correct it does not use only the different values far from the average. However, TRAVOS reputation is evaluated by filtering out the false opinions. Some challenges are still not resolved with TRAVOS. First, it considers that the behaviour of entities does not change over time. Also, no forgetting factor is defined with TRAVOS.

3.2.4 Hybrid Approaches

Solutions based on hybrid approaches aim to combine the policy and monitoring techniques to evaluate trust. In our study, we present three solutions from this class.

Ray and Chakraborty Approaches

In [RC04], authors aim firstly to provide a clear and general definition of the concept of trust vector, and then they present a mathematical evaluation of this vector. In [CR06], an extension of [RC04] by integrating the trust vector with RBAC model is proposed. This approach permits to define a trust level of a user based on the evaluation of different parameters (direct interactions, indirect interactions and the collected attributes). This trust level allows to define the role of the user. When the trust level of the user is updated, the role assigned to him will change correspondingly. This approach is extended in several new frameworks as [Sar13, RMRH13]. In [Sar13], authors focus on improving it by proposing a generalization and abstraction of trustBAC. In [RMRH13], the authors update his trust evaluation in order to be adequate for mobile clouds. They also give a formalization of the trust with the delegation concept. This approach is based on a very interesting trust evaluation methodology. However it suffers from a critical problem: A user is generally trusted or (distrusted); trust in this proposal is not evaluated related to the action to be performed.

SULTAN

SULTAN is developed by Tyrone Grandion et al. [GS03]. This solution provides a whole architecture with different entities to evaluate trust in the internet system. Its architecture contains:

- State Information that contains all the information about the different entities (experiences, recommendations, etc).
- SULTAN monitor that collects information in order to update the state information content.
- Risk service that is used to identify the possible risks in the system it will be used to evaluate the trust of a system.
- Security policies of the system which are based on the Ponder language [DDLS01].
- Analysis Engine that is the inference system that uses the risk level evaluated by the risk server and the security policy to determine a response.

However, this system is only designed for a centralized architecture.

Framework SECURE

It is a framework developed in the European project SECURE [Jen02]. It aims to provide a trust management system based on a trust evaluator components and a security policy. Like SULTAN framework, this solution has a risk evaluator engine that evaluates the risk related to an entity. It contains also a trust engine and a collaboration tool. This tool aims to save all the interactions in order to be used to evaluate the trust parameters.

However, this framework needs the use of an external language to express policies for the risk interpretation.

3.2.5 Synthesis

In the previous sections, we have described the different trust approaches that summarize the existing three classes of trust framework: policy based approaches, monitoring based approaches and hybrid approaches. This study is realized in order (1) to understand the advantages and the addressed issues of each solution (2) to give us a clear view about the trust challenges that have to be resolved in our framework.

In MOE, the use of a policy based approach is very interesting. It permits to exchange the different attributes of a user that request an external resource in the O-grantor. Trust negotiation is also very important. The O-grantor will not accept to share its policy with all the users. Moreover, we believe that the use of XeNA approach and its extension to address the MOE issues can be a first starting step in our framework. Indeed, we think that the use of a policy based approach is not enough in MOE. Since this kind of solutions does not address the change of a user behavior: No monitoring process is designed and integrated to track the behavior of an entity and to evaluate its trust level.

Regarding the second class of trust framework that is based on a monitoring and analyzing entities behavior, the studied solutions may provide two important values based on the direct and indirect interactions. This study helps to conclude that a central approach is not suitable with the MOE environment. However, with this kind of solutions the policy creation, administration and the expression language of trust are not well addressed. As a result, we believe that a hybrid solution can be a good candidate to address trust in MOE: the use of a policy based approach that provides the negotiation of attributes and their assessment together with a monitoring approach that evaluates the historic behaviors of the user will be necessary to evaluate the trust in MOE. However, we find some issues that are not solved in existent solutions as the definition of a trust level of the user organization, its integration and its use in order to decide about the whole trust level of a user. These issues encourage us to propose a new trust approach based on `OrBAC` to the MOE. The most important issues to be resolved in our work are:

How to define the trust parameters?, How to evaluate them?, How to combine the different evaluations?, How to integrate the trust into `OrBAC`? and How to share these feedback?

Before studying our trust framework, we present on the following some prelimi-

naries related to the use of OrBAC in MOE.

3.3 Preliminaries

3.3.1 OrBAC

We recall that the OrBAC paradigm is frequently used to define the access control [KBB+03, CM04, CCBCC08c, KDBK09, TCM12]. It offers the possibility to create *the interoperability security policy* for external users. This model works with the notion of *role*, that has to be assigned to a set of subjects in an organization. Therefore, the roles will have some specific rights instead of *concrete* users. Another level of abstraction in OrBAC is to group a set of actions as *activities*, and a set of objects as *views*. So, the interoperability security policy will be described in terms of *roles*, *activities* and *views*.

To describe an OrBAC scheme for MOE we use the predicates presented in Figure 3.4. The predicate 3.1 means that in org there exists an employee $user$ that is mapped with $role$. The predicate 3.2 means that in organization org, $action$ is considered as an implementation of $activity$. Next, the predicate 3.3 means that in org the $object$ is used in $view$. Finally, the syntax to define the *security rules* is presented in the two predicates 3.4 and 3.5. We may have two kinds of access types that are permission and interdiction. The later parameter ($context$) is a condition over the environment, and it must be satisfied to activate the security rule. As we will see in Section 3.7, the trust integration into OrBAC will be done based on a new type of context.

$$\texttt{empower}(org, user, role) \tag{3.1}$$

$$\texttt{consider}(org, action, activity) \tag{3.2}$$

$$\texttt{use}(org, object, view) \tag{3.3}$$

$$\texttt{permission}(org, role, activity, view, context) \tag{3.4}$$

$$\texttt{interdiction}(org, role, activity, view, context) \tag{3.5}$$

Figure 3.4: Predicates to define the OrBAC structure in MOE.

In order to use OrBAC in MOE, a first extension named O2O was proposed in [CCBC06b]. In this model, the administration of the different policies may be decentralized. An O-grantor controls the access of its resources based on several VPOs (Virtual Private Organizations). A VPO that is derived from the local policy is created with each O-grantee after the signature of a contract. O2O has several advantages as the decentralization of the administrating task and the automatic creation of the VPOs. However, in several scenarios, this solution suffers from the duplication of policies

(or rules) since several shared resources are the same and these rules are derived from the same local policy.

In the Subsection 3.3.2, we provide our extension of this model that allows to simplify the management task by reducing the number of policies.

3.3.2 GVPO Extension

We define a new concept called *General Virtual Private Organization* (GVPO) as:

For any organization *org* that shares some resources or services, we define the GVPO of *org* as the organization Any2org that administrates the interoperability security policy for a subset of participants. This GVPO will be created when we have at least one O-grantee that can be managed by this policy. This means that its creation and destruction will not be realized after each collaboration with an O-grantee as it is done in O2O. Let us note that with this concept the administrator have three different options: a) To create a GVPO for all the organizations; b) To create a GVPO for a subset of the existent organizations and different VPOs for the rest or; c) To create one VPO for each organization.

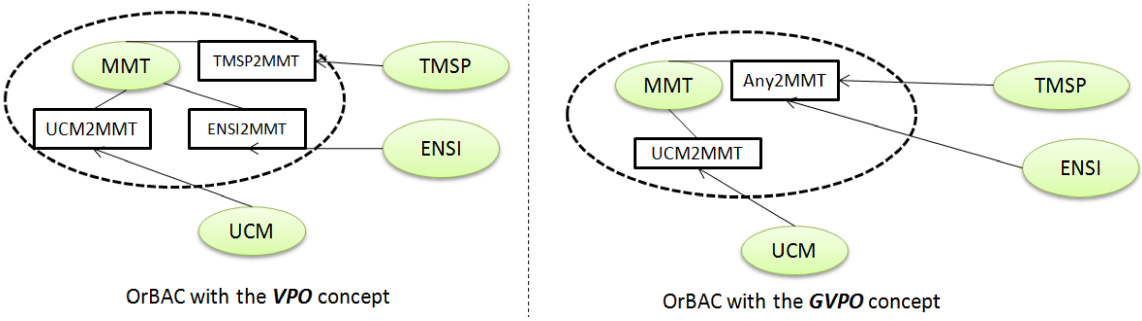


Figure 3.5: GVPO concept.

Example 1 In Figure 3.5, we present an example of MOE. Let us consider four organizations, TMSP laboratory, MMT company, UCM university and ENSI university, that are working together in an international project. Each one of them has their resources and services that can be shared by some users of other participants.

In the left part of Figure 3.5, the creation and management of policies for the organization MMT is realized without the GVPO concept. In this case we will have a VPO for each organization that administrates the policy of one and only one O-grantee. Using the concept of the GVPO (see the right part of Figure 3.5), we are able to have one interoperability security policy with the organizations TMSP and ENSI that will be managed by the virtual organization Any2MMT and another policy with UCM that is managed by the VPO UCM2MMT.

The two tables shown in Figure 3.6 illustrate some activities, views and contexts of MMT and TMSP organizations that will be used in the examples cited in the rest of this chapter.

These elements have to be assigned to the concrete entities. This is done based on rules illustrated in Figure 3.4. An example of this assignment is shown in Figure 3.7.

MMT

activities	views	context
modify	confidential_files	urgent_problem
comment	secret_files	
read	public_files	
consult	files	

TMSP

activities	views
edit	deliverables1
add_remarks	excel_documents

Figure 3.6: Example of activities, views and contexts.

MMT	TMSP
consider(Any2MMT, trace.txt,confidential_files)	consider(Any2TMSP, ISER.V1.0.pdf,deliverables)
consider(Any2MMT, main.c,secret_files)	consider(Any2TMSP, paln.xls,excel_files)
consider(Any2MMT, Install.txt,public_files)	
use(Any2MMT, download,read)	

Figure 3.7: Example of abstract and concrete entities.

$$\texttt{memberOf}(org, user) \tag{3.6}$$

$$\texttt{collaborateWith}(org, org') \tag{3.7}$$

$$\texttt{origin}(\texttt{Any2org}, org) \tag{3.8}$$

$$\begin{aligned}\texttt{empower}^*(\texttt{Any2org}, user, role) \longleftarrow & \exists org' \texttt{ and } org : \texttt{origin}(\texttt{Any2org}, org) \wedge \\ & \texttt{memberOf}(org', user) \wedge \\ & \texttt{collaborateWith}(org', org) \wedge \texttt{cond}(org, user, role)\end{aligned} \tag{3.9}$$

Figure 3.8: Equations to define GVPO.

In order to be able to formally represent GVPO, three new predicates memberof, collaborationWith and origin and an extension of the empower predicate (predicate 3.1) must be included. These predicates are presented in Figure 3.8. The predicate 3.6 determines the organization of the user. Its boolean result is true if org is the organization of $user$. Besides, the equation 3.7 is correct if org' is collaborating with org with respect to a GVPO policy. The equation 3.8 is true if the virtual organization Any2org controls the access of the organization org resources. Finally, based on the last equation, the user will be mapped to the role in the virtual

organization Any2org not only based on some conditions or attributes but also if his organization can be managed by the GVPO. (We mention that cond$(org, user, role)$ is correct if the user provides a set of attributes that permits this assignment).

Rule1 ::= permission(UCM2MMT, $engineer, modify, confidential\_files, default\_ctx$)
Rule2 ::= permission(Any2MMT, $engineer, modify, confidential\_files, default\_ctx$)
Rule3 ::= permission(Any2MMT, $project\_manager, modify, secret\_files, urgent\_problem$)
Rule4 ::= interdiction(Any2MMT, $engineer, read, public\_files, default\_ctx$)

Figure 3.9: MMT interoperability security policies.

Example 2 Next, we show in the following how to represent the MMT interoperability security policies. These policies are managed based on one VPO UCM2MMT and a GVPO Any2MMT. An example of rules are presented in Figure 3.9. The first one permits to manage the access of UCM users and the second one permits to control the access of users that belong to the set of authorized organizations. In this example, we show 1) a rule of the VPO UCM2MMT that assigns a permission to an engineer to modify confidential_files 2) and three rules of the GVPO Any2MMT: two rules(Rule2 and Rule4) for an engineer and one for a project manager (Rule3). Rule2 means that an engineer that belongs to a specified subset of organizations is permitted to modify the confidential_files, however, Rule4 prohibits the same role to read any secret_files. Rule3 is a permission to a project manager to modify the secret files only if they have an urgent problem (deadline of a task exceeded without a response from MMT).

3.4 Our Trust Framework

Some critical scenarios of MOE cannot be completely specified with the classical OrBAC. For instance, let us consider that a user u_1 of UCM is permitted to modify a confidential file based on the interoperability security policies of the company MMT. However, after several interactions, MMT *believes* that his work (or his organization work) related to this *goal* (modification of confidential files) is not correctly done since they detect some failures (some bugs). With the classical OrBAC framework, these beliefs and their influence on the response of the O-grantor (i.e, MMT) are not taken into account. This problem encourages us to study the *trust concept* and its integration in the previous framework. Thus, trust in our framework is based on these two definitions:

- "*Trust is not an objective property of an agent but a subjective degree of belief about agents*" [ARH00];

- "*Trust consists of beliefs. Trust is a mental state, a complex attitude of an agent x towards another agent y about the behavior/action relevant for the result(goal)*" [CF98].

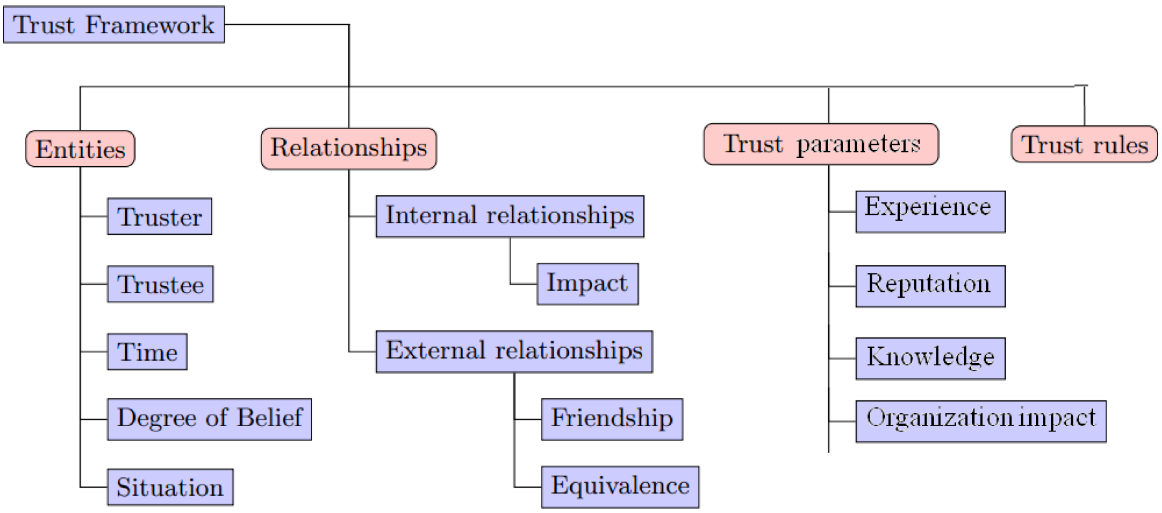


Figure 3.10: Trust framework scheme

In Figure 3.10 the scheme of our solution is depicted. On the left there are the entities that we have defined in our framework. In Subsection 3.4.1 all of them are presented in detail. Next, some internal relationships that relate the different entities of this framework are introduced. In the following, we will detail the concepts behind each of these entities and components that are used to design our framework.

3.4.1 Entities

Definition 1 In `MOE`, the **truster** is any organization that offers access to a specific resource. Any *O-grantor* in `MOE` will be a truster. The **trustee** can be an organization or a user that needs a service. □

Definition 2 The use of **time** is important in our presentation, since trust changes during time. To represent it, we will use *time intervals*. We say that $\hat{T}_i = [t_i, t_{i+1})$ is a *time interval* if t_i, t_{i+1} are time stamps and $t_i < t_{i+1}$ and $t_{i+1} - t_i = dt$, where dt is a constant value. □

Definition 3 **The degree of belief**, also known as *the trust value* is used to measure the belief between two entities. Its value allows us to determine if we can trust or not the trustee. □

Finally, a **trust relationship** cannot be defined globally, it might depend on a new concept called a **situation**. For example, you can trust your friend for driving your car but not to use your credit card.

Following we introduce the **situation** parameter. This concept is defined based on some notions of control access policies: *activities* and *views*. We recall that an activity is a group of actions and a view is a set of objects on which the same security rule is applied.

Let ACTIVITIES and VIEWS be the sets that contain all possible activities and views respectively.

Definition 4 A *situation* is a tuple (a, v) where $a \in$ ACTIVITIES is an activity and $v \in$ VIEWS is a view. The set of all possible situations will be denoted by SIT. □

Example 3 We continue the example 1, Bob is an engineer of the organization UCM, where UCM is the O-grantee. We consider that the organization MMT gives Bob the rights to *modify*, in short a, any *secret file*, in short v of MMT. Thus, we have that a and v are an activity and a view in MMT respectively. Moreover, we are able to define the situation modify▷secret_file as the tuple (a, v).

As it is illustrated in Figure 3.10, different relationships are defined in our trust solutions. In the following we present the local ones.

3.4.2 Local Relationships: Hierarchy and Impact

Syntax

- **Hierarchy Relationship**

In OrBAC framework we can describe a hierarchy relationships among activities and views. Next the syntax to define this relationship is presented.

$$activity\ \mathbf{h_{act}}\ activity'\ \mathbf{in}\ org \mid view\ \mathbf{h_{view}}\ view'\ \mathbf{in}\ org \tag{3.10}$$

These relationships simplify the administrator task by the automation of some privilege assignment.

Example 4 Let us consider the relationship between a subdirectory and the directory in which it is cataloged may be seen as as an organizational heritage between two views. For example: public_files $\mathbf{h_{view}}$ files **in** Any2MMT.

We can also have the following relationship read $\mathbf{h_{act}}$ consult **in** Any2MMT. It means that if a user is permitted to consult a resource then this user will be also permitted to read its content.

- **Impact Relationship**

The second relationship $\mathtt{Impact}_\alpha$ is defined for activities, views and situations that belong to the same organization. The syntax of this relationship is:

$$\begin{aligned} &activity\ \mathbf{I_{act},\alpha}\ activity'\mathbf{in}\ org \mid view\ \mathbf{I_{view},\alpha}\ view'\mathbf{in}\ org \mid \\ &situation\ \mathbf{I_{sit},\alpha}\ situation'\mathbf{in}\ org \end{aligned} \tag{3.11}$$

where $\alpha \in (0, 1]$.

Usually $\mathbf{I_{act,\alpha}}$ and $\mathbf{I_{view,\alpha}}$ might be specified by the administrator in order to study the possible impact of an interaction to the trust belief related to a situation. $\mathbf{I_{sit,\alpha}}$ is a relation between two situations and it is defined when there is an impact relationship between the two activities and views that compose the two situations. α is a value between (0,1] that represents the impact factor.

Example 5 read $\mathbf{I_{act,\alpha}}$ modify **in** `Any2MMT` is an impact rule between the two activities read and modify. This means that any request of a user from TMSP or ENSI to read an object may impact the trust belief of this user with MMT related to a situation that contains the activity modify.

Semantic

- **Impact Relationship**

In order to evaluate the impact between the two situations, we need to introduce two additional functions $\texttt{impV}(v_1, v_2)$ that relates the views v_1 and v_2 and $\texttt{impA}(a_1, a_2)$ that relates the activities a_1 and a_2. The first one means that a trust evaluation related to v_1 impacts on the trust evaluation related to v_2 with a value in $[0, 1]$. The second one represents the same idea regarding two actions a_1 and a_2.

Definition 5 We define $\texttt{impV} : \texttt{VIEWS} \times \texttt{VIEWS} \rightarrow [0, 1]$ and $\texttt{impA} : \texttt{ACTIVITIES} \times \texttt{ACTIVITIES} \rightarrow [0, 1]$ as the functions that compute the *impact factor between two views/activities* respectively in the same organization org. For any $v_1, v_2 \in \texttt{VIEWS}$ and $a_1, a_2 \in \texttt{ACTIVITIES}$, we have that:

$$\texttt{impV}(v_1, v_2) = \begin{cases} 1 & \text{if } v_1 \ \mathbf{h_{view}} \ v_2 \ \mathbf{in} \ org \vee \ v_1 = v_2 \\ \alpha & \text{otherwise} \end{cases}$$

$$\texttt{impA}(a_1, a_2) = \begin{cases} 1 & \text{if } a_1 \ \mathbf{h_{act}} \ a_2 \ \mathbf{in} \ org \vee \ a_1 = a_2 \\ \lambda & \text{otherwise} \end{cases}$$

where α in $[0, 1]$ and λ in $[0, 1]$ □

Note that if the first condition does not hold, then the interoperability security policy administrator can define an *impact value* α. This value allows us to represent the relation level between two views. This process is similar for the λ value.

Example 6 In our running example, the administration can identify an α value closed to 1, between two views of our MOE scenario, the `secret_file` and the `confidential_file` of MMT.

In order to compute the relation between different situations s_1 and s_2, we define the function $\texttt{inf}(s_1, s_2)$. A higher value of $\texttt{inf}(s_1, s_2)$ means that the trust valuation of s_2 will be more influenced (impacted) by the trust valuation related to s_1.

Definition 6 We define the $\texttt{inf} : \texttt{SIT} \times \texttt{SIT} \rightarrow [0, 1]$ function as the *influence* of a situation with respect to another one. For any $s_1 = (a_1, v_1), s_2 = (a_2, v_2) \in \texttt{SIT}$ we have that $\texttt{inf}(s_1, s_2)$ is 0 if $\texttt{impV}(v_1, v_2) = 0$ or $\texttt{impA}(a_1, a_2) = 0$, otherwise:

$$\texttt{inf}(s_1, s_2) = \frac{\texttt{impA}(a_1, a_2) + \texttt{impV}(v_1, v_2)}{2}$$

□

Example 7 Let us continue the Example 1 where the engineer Bob is allowed to modify▷secret_file of the organization MMT. The trust evaluation for Bob related to the situation modify▷secret_file (s_1) will have an influence on the trust evaluation related to the situation modify▷file (s_2).

It happens because both cases share the same activity, that is, modify, and their associated views are related, that is secret_file $\mathbf{h_{view}}$ file **in** MMT. Therefore we have that the influence relation of $\texttt{inf}(s_1, s_2)$ is 1 and:

$$\texttt{modify} \triangleright \texttt{secret\_file}\ \mathbf{I_{sit,1}}\ \texttt{modify} \triangleright \texttt{file}\ \textbf{in}\ \texttt{Any2MMT}$$

3.4.3 Trust Model Parameters

In this section we will define the different parameters of our trust model. We will also present an evaluation function for each one of them. In particular, each parameter will be defined for a *generic trustee* and then we will present the differences between trust for user-organization and for organization-to-organization.

A. Experience

Experience learning is a process which aims to establish wisdom on making decisions. It is based on the evaluation of the previous interactions between the trustee and the truster related on a specific situation at a period of time.

Any person will rely on this parameter to trust another entity and to decide whether to continue or not the collaboration. In our framework, we consider two different types of experience. The experience of the trustee organization that takes into account the historical *behaviors* of all users of this organization. And the direct experience where only the previous *behaviors* between the user and the truster are considered.

In this context, we have proposed through our solution new concept called "behavior" and two related notions, a "request" and a "log". In the following, their definitions are detailed.

Definition 7 A request is a tuple (u,$\hat{T}_i$,s) where u is subject, $\hat{T}_i$ is the time interval, during it the reception of the request is done, and s is the situation that appears in the request.

A *behavior* is a tuple (req, dec, sat) where the first element req is a request, dec is the decision of OrBAC, and $sat \in [-1, 1]$ is the evaluation of this behavior. We

assume that any behavior, denoted b, of this file can be valuated as a *satisfactory* or *unsatisfactory* behavior. If the valuation is unsatisfactory then it is considered as a *bad behavior*, that is, it will decrease the experience evaluation of the trustee. On the contrary, if the valuation is satisfactory it will increase the experience evaluation. This evaluation will be realized by a technique that depends on the application and the requirements of the administrator. We note that the output of this technique is a function $sat(b)$ that associates a value in [-1,1] to the behavior. An example of this function will be given in the case study.

A sequence of behaviors is called a log. The log file for the organization $\texttt{org}_\texttt{A}$ is denoted by $l_{\texttt{org}_\texttt{A}}$, and the set of all logs by $\mathcal{L}$. Given a log file $l \in \mathcal{L}$, we will define three projections functions $\pi_{\hat{T}_i}(l)$, $\pi_{\hat{s}}(l)$ and $\pi_{\hat{u}}(l)$. The first one returns those behaviors that were performed in a time during $\hat{T}_i$. The second one computes all behaviors of the situation s. Finally, the last one correspond to the set of behaviors of the subject u. □

A.1 Experience of a user:
The experience evaluated by a truster $\texttt{org}_\texttt{A}$ for a user u, related to a specific set of behaviors B, a situation s at time interval $\hat{T}_n$ respects the following rules.

1^{st} **Rule**: The evaluation depends on the past evaluation of all the behaviors of the user u related to the same situation.

2^{nd} **Rule**: The influence of an evaluation of any event will decrease with the flow of time. Thus, it needs an attenuation function which decreases the evaluation of an interval depending on the time [RC04].

Example 8 Let us consider the engineer Bob and the situation $s_1 = \texttt{modify} \triangleright \texttt{secret\_file}$ presented in Example 7,

Let $B = \bigcup_{1 \leq j \leq 6} \{b_j\}$ be the set of behaviors done during the first six intervals. The evaluation of the experience of the engineer related to s_1 at $\hat{T}_5$ will depend on all the behaviors done during $\hat{T}_0$ to $\hat{T}_5$. According to the second rule, we have that the evaluation of the behaviors done during $\hat{T}_5$ must be more relevant than those done during $\hat{T}_3$.

Definition 8 For any $u \in \texttt{Subjects}$, $s \in \texttt{SIT}$, $\hat{T}_i \in \mathcal{I}_{\mathbb{R}_+}$, $l \in \mathcal{L}_{\texttt{org}_A}$, we define the experience evaluation function with respect to $\texttt{org}_A$ as:

$$\texttt{eX}_1(u, \texttt{org}_A, \hat{T}_n, s, l) = \frac{\sum_{i=0}^{n} f(i) * \frac{\sum_{b \in l_i} sat(b)}{|l_i|}}{n}$$

where $l_i = \pi_{\hat{T}_i}(\pi_s(\pi_u(l)))$ and f is an attenuation function. □

Let us note that the previous definition computes the experience of any user at a given time and situation as the weighted average of all the evaluations according to an attenuation function.

As an example of attenuation function, we could consider $f(i)$ that is defined as $e^{(-m \cdot (n-i))}$. This function depends on a constant m, that will be chosen by the administrator. It is used to satisfy the 2^{nd} Rule. Following we introduce how this kind of functions works.

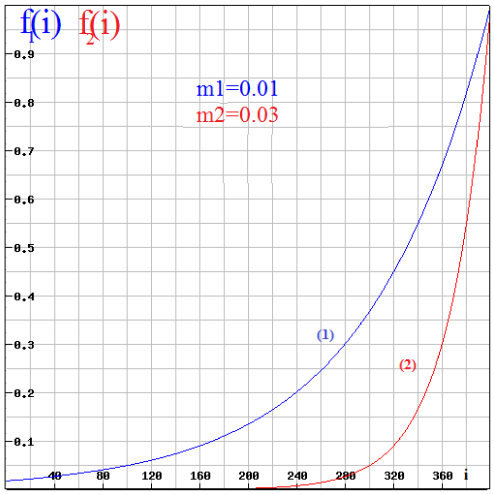


Figure 3.11: Example of attenuation functions.

Example 9 Figure 3.11 presents two different attenuation functions f_1 and f_2 that are respectively associated to m1 (0.01) and m2(0.03). In function f_1 the last 35% of periods have a coefficient bigger than 0.5. However, in function f_2, the 75% first periods are almost ignored, their coefficients are less than 0.25, and only the last 10% periods, have associated a coefficient bigger than 0.5.

Next we focus on the experience evaluation in the organization to organization context.

A.2 Experience of an organization:
The experience of an organization $\texttt{org}_\texttt{B}$ evaluated by an organization $\texttt{org}_\texttt{A}$, with respect to their users $\texttt{employee}(\texttt{org}_B, \texttt{org}_A)$ holds the following rules:

1^{st} **Rule** Similar to the 1^{st} and 2^{nd} rules of the user to organization experience evaluation description.

2^{nd} **Rule** It depends on the experience evaluation of all users of the organization $\texttt{org}_\texttt{A}$ related to the same situation s at the end of interval $\hat{T}_n$.

Definition 9 For any $\texttt{org}_\texttt{B} \in \texttt{Organizations}$, $s \in \texttt{SIT}$, $\hat{T}_i \in \mathcal{I}_{\mathbb{R}_+}$, and for any not empty log $l \in \mathcal{L}_{\texttt{org}_A}$, we define the experience evaluation function of an organization $\texttt{org}_\texttt{B}$ with respect to $\texttt{org}_\texttt{A}$ as:

$$\texttt{eX}_2(\texttt{org}_B, \texttt{org}_A, \hat{T}_n, s, l) = \frac{\sum_{u \in \texttt{employee}(\texttt{org}_B, \texttt{org}_A)} \texttt{eX}_1(u, \texttt{org}_A, \hat{T}_n, s, l)}{|\texttt{employee}(\texttt{org}_B, \texttt{org}_A)|}$$

□

A.3 The impact of influence relation on the evaluation of experience:
Let us note that previous evaluations are related to one situation. Next, we deal with the *influence relation between two situations* and how does it impact on the evaluation of the experience. To do this, we will use the function `inf` presented in Definition 6.

Definition 10 We say that $\texttt{eX} \sim$: $(\texttt{Subjects} \cup \texttt{Organizations}) \times \wp(\mathcal{B}) \times \texttt{Organizations} \times \mathcal{I}_{\mathbb{R}_+} \times \texttt{SIT} \times \mathcal{L}_{\texttt{org}_A} \to [-1, 1]$ relates the evaluation of the experience of a user with respect to an organization taking into account the dependency among different situations. Let $e \in (\texttt{Subjects} \cup \texttt{Organizations})$, $B \subseteq \mathcal{B}$, $\texttt{org}_\texttt{A} \in \texttt{Organizations}$, $s \in \texttt{SIT}$, $\hat{T}_n \in \mathcal{I}_{\mathbb{R}_+}$ and for any not empty log $l \in \mathcal{L}_{\texttt{org}_A}$, we have that

$\texttt{eX} \sim(\texttt{org}_B, \texttt{org}_A, \hat{T}_n, s, l)$ is defined as:

$$\frac{\sum_{s' \in \texttt{SIT}}(\texttt{inf}(s', s).\texttt{eX}_\texttt{i}(\texttt{org}_B, \texttt{org}_A, \hat{T}_n, s, l))}{\sum_{s' \in \texttt{SIT}} \texttt{inf}(s', s)}$$

where the $\texttt{eX}_\texttt{i}$ denotes $\texttt{eX}_1$ if e is a user and $\texttt{eX}_2$ if e is an organization. □

Following we present this idea in our running example.

Example 10 In Example 7, we defined two different situations $\texttt{modify} \triangleright \texttt{secret\_file}(s_1)$ and $\texttt{modify} \triangleright \texttt{confidential\_file}(s_2)$ where $\texttt{inf}(s_2, s_1) = 1$. We suppose that the set of situations is $\{s_1, s_2\}$. Next we present the evaluation of the experience taking into account the previous relationship $\texttt{eX} \sim(\texttt{Bob}, \texttt{org}_A, \hat{T}_5, s_1, l) =$

$$\frac{\texttt{eX}_1(\texttt{Bob}, \texttt{org}_A, \hat{T}_5, s_1, l) + \texttt{eX}_1(\texttt{Bob}, \texttt{org}_A, \hat{T}_5, s_2, l)}{2}$$

where $B = \bigcup_{1 \leq j \leq 6}\{b_j\}$.

A.4 Evaluation:

In the following, we present a comparison between the experience of the engineer Bob regarding the organization $\texttt{org}_\texttt{A}$ with, and without the consideration of the influence relation. These results were simulated during the first eight intervals related to the three situations $\texttt{modify} \triangleright \texttt{secret\_file}(s_1)$, $\texttt{modify} \triangleright \texttt{confidential\_file}(s_2)$ and $\texttt{read} \triangleright \texttt{file}(s_3)$.

The results are presented in Figure 3.12. Two different configurations were considered in these simulations:

On the one hand the input parameters of the first graphic are:

- $\texttt{inf}(s_1, s_2) = 1$
- $\texttt{inf}(s_3, s_2) = 0$

On the other hand, the input parameters of the second graphic are:

- $\texttt{inf}(s_1, s_2) = 0$
- $\texttt{inf}(s_3, s_2) = 1$

Next we compute the experience of Bob without considering the influence relation (blue line (based on the Definition 8)) and with the influence relation (red line (based on the Definition 10)).

- **A.5 Results Analysis:**

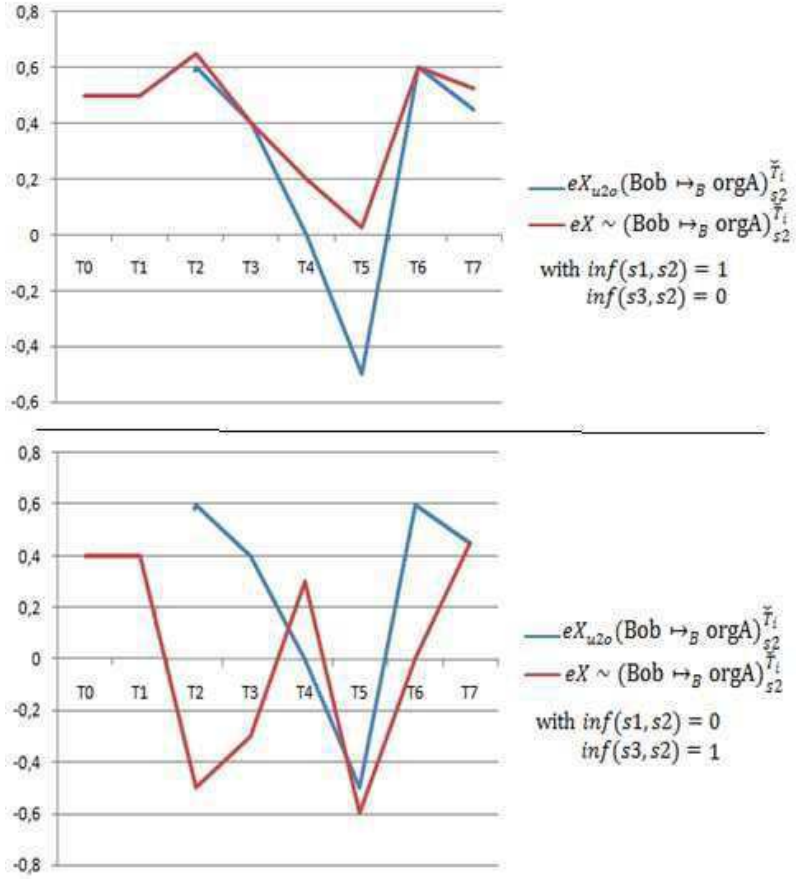


Figure 3.12: Difference between classical experience and influenced one.

The use of the influence relation allows us to have more information about the trustee. For example: during the first two periods, we have not interactions with the engineer Bob during $\hat{T}_0$ and $\hat{T}_1$ related to the situation s_2 (blue curve). However, with the influence relation between s_1 and s_2 we have an experience value related to s_2 during $\hat{T}_0$ and $\hat{T}_1$ that can be used by our system. For these reasons, we remark that the blue curve started at $\hat{T}_2$ where we have the first interaction related to s_2. However, the red curve started at $\hat{T}_0$ since other interactions related to s_1 influence the experience evaluation.

The experience evaluation based on the Definition 10 will depend on the influence value. In order to show this dependence we can compare the two red curves of the first and the second graphics where the influence values are not similar. We remark that the final values (red curves) in the first figure, at the most of periods, are more than those evaluated in the bottom part of Figure 3.12. This means a bad configuration of some concepts (hierarchy relation, impact values) will give a wrong result. For these raisons, these parameters should be specified by experts.

B. Reputation

Reputation aims to gather and aggregate feedback about an entity from another participant. According to [RZFK00], reputation provides an incentive for honest behavior and helps people to make decisions about who to trust.

In MOE, reputation measures the global perception of the trustee based on the trust evaluation of some organizations in this environment. The honesty of the different participants in these statistics should be considered to estimate a right prediction.

B.1 Reputation Syntax

The reputation parameters need the definition of two new relationships: friendship and equivalence.

- Friendship:

The first one concerns two different organizations of `MOE` and it follows this syntax:

$$org \textbf{ is friend of } org' at\ \hat{T}_n \tag{3.12}$$

It means that org consider org' as its friend after the period $\hat{T}_n$ so it can accept its recommendations related to some trustees for the next period. This relation is asymmetric, reflexive and not transitive. The set of organizations that can participate in this evaluation, at a given time interval, is called *Friends Group* of the O-grantor org_A denoted by $\mathtt{fr}(\mathtt{org_A})^{\hat{T}_n}$.

- Equivalence:

The second relationship is used between two *external* situations that are defined between two organizations. The syntax of this relation is:

$$situation \textbf{ is equivalent to } situation' \tag{3.13}$$

where $situation$ and $situation'$ belong to different organizations. The previous relation allows us to share trust beliefs between different organizations. We note that each organization has its local situations. When an organization org' asks org of the trust belief related to its local situation, org may send its trust belief related to an equivalent situation. This function is asymmetric. A more detailed expression of this function will be detailed in the next chapter.

Example 11 Let us present these tow relations:

- ENSI **is friend of** MMT at $\hat{T}_5$
- MMT **is friend of** TMSP at $\hat{T}_3$

In this environment, MMT entreprise (resp. TMSP university) considers the ENSI university (resp. MMT entreprise) as its friend at $\hat{T}_5$ (resp. at $\hat{T}_3$). So, it may trust its recommendations related to other participants.

- `add_remarks`▷`excel_documents` **is equivalent to** `modify`▷`files`

We have two situations: the first one `modify`▷`files` (s_1) defined in MMT and the second one `add_remarks`▷`excel_files`(s_3) defined in TMSP.

This predicate means that MMT considers the s_3 situation as an equivalent one to its situation s_1. Therefore, MMT will send its trust belief related to the modification of files when TMSP requires its recommendation related to s_3. This rule is different from `modify`▷`files` **is equivalent to** `add_remarks`▷`excel_files` that can be defined by TMSP university.

B.2 Reputation Semantic

Definition 11 We say that $\texttt{fr} : \texttt{Organizations} \times \mathcal{I}_{\mathbb{R}_+} \backslash \{\hat{T}_0\} \times \texttt{SIT} \rightarrow \wp(\texttt{Organizations})$ is the function that computes the set of *Friends* of an organization. For any $\texttt{org}_\texttt{A} \in \texttt{Organizations}$ and $\hat{T}_n \in \mathcal{I}_{\mathbb{R}_+} \backslash \{\hat{T}_0\}$ we have that:

$$\texttt{fr}(\texttt{org}_\texttt{A})^{\hat{T}_n} = \left\{ org \;\middle|\; org \in \texttt{Organizations} \wedge \texttt{eval}(R) \geq \delta \right\}$$

where $R = \texttt{trust}(org \mapsto \texttt{org}_\texttt{A})^{\hat{T}_{n-1}}_{recommendation}$ □

We consider that each organization will evaluate the trust level of other organizations related to the recommendation situation during the previous period. Therefore, an entity will belong to the friend group only if the evaluated trust level related to the specific situation "recommendation" is more than the threshold δ.

Definition 12 Let $\texttt{org}_\texttt{B} \in \texttt{Organizations}$, $\hat{T}_n \in \mathcal{I}_{\mathbb{R}_+} \backslash \{\hat{T}_0\}$ and $s \in \texttt{SIT}$. We define for the O-grantor $\texttt{org}_\texttt{A}$ the function that evaluates the reputation of an organization with respect to the MOE environment as:

$$\texttt{rep}(\texttt{org}_\texttt{B}, \texttt{org}_\texttt{A}, \hat{T}_n, s) = \begin{cases} \frac{\sum_{\texttt{org}_\texttt{i} \in FG} Rec(\texttt{org}_\texttt{i}, \texttt{org}_\texttt{B}, s, \hat{T}_{n-1})}{|FG|} & \text{if} |FG| \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $FG = \texttt{fr}(\texttt{org}_\texttt{A})^{\hat{T}_n}$ and $\text{Rec}(\texttt{org}_\texttt{i}, \texttt{org}_\texttt{B}, s, \hat{T}_{n-1})$ is the recommendation value sent by $\texttt{org}_\texttt{i}$ regarding $\texttt{org}_\texttt{B}$. More information about this recommendation value will be given in the next chapter.

□

Note that the reputation parameter is only defined for an organization as a type of trustee. Indeed, it is hard to measure this parameter for users in MOE. According to [RZFK00], the evaluation of the reputation of an entity must respect some properties. One of them is the longevity of agents (e.g. no modification of identity) that is not always offered for users in MOE. Since, the exchange of attributes of the same user with organizations may be different. In addition, the collection process of the trust evaluation for each user from the several organizations requires the exchange of a huge number of messages. Therefore, some problems can appear as the useless consumption of bandwidth.

- **B.3 Evaluation**

In this section, we evaluate our reputation method regarding the different issues related to this parameter defined in the literature. In [Resnick et al., 2000], [Josang et al., 2007], several issues are addressed as:

1. The recommendation value can be modified or used by a malicious node during its propagation.
   $\Longrightarrow$This issue is not studied in our case. In our simulation, the different messages are exchanged in a clear way. As a future work, we aim to use a cryptographic algorithm to secure this communication.

2. An entity may send a false recommendation value.

3. All entities in the system can participate in the evaluation.
   $\Longrightarrow$These two last issues are addressed in our system. First, we do not authorize to all the entities to participate in the evaluation of the reputation value. Only entities in the friend group, initially configured by the administrator, are requested for participation. Moreover, in order to reduce the reception of the false recommendation value, our friend group is dynamic since the number and the participants change during the communication (see Figure 3.13 and its description). In Figure 3.13, we show the number of friends of the $\texttt{org}_{\texttt{A}}$ during the

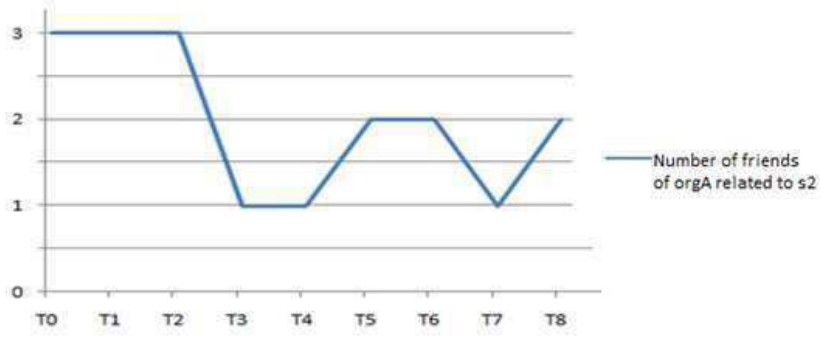


Figure 3.13: A dynamic friends group related to s_2.

   first 8 periods. Initially, three organizations are considered as friends. However, this set will be changed based on the trust value of each one of them regarding the truster that is $\texttt{org}_{\texttt{A}}$. For example it has three friends at $\hat{T}_0$ and only one friend at $\hat{T}_3$. This dynamicity aims to reduce the problem of false recommendations. In our framework, we do not trust the recommendation of an entity related to a situation if its previous recommendations were not correct.

4. The recommender has to understand the needed situation by the requester (trustee). In literature, there are two approaches:

   - First method: Each organization has its local situations. In order to participate to the reputation framework, they have to map them with others defined in the different recommenders.

- Second method: A common set of situations should be created. Actually, we are using this method to simplify the tasks. However, we aim to study the first method and to propose a mapping algorithm to our reputation framework in MOE.

C. Knowledge

Knowledge aims to evaluate the collection of a specific information and attributes about the trustee. The truster seeks to gather the maximum amount of information about the trustee. In MOE, the information gathering process depends on the type of the trustee. In the following, we deal with:

Knowledge about an organization: The source of this knowledge is the contract that is signed with the trustee. It contains relevant information about the trustee such as shared resources, restrictions and type of contract.

Knowledge about a user: It is based on an exchange of credentials or/and attributes. Different models have studied the collect process that can be accompanied with a negotiation protocol to collect the maximum number of attributes and also to protect the policy [HCBCD09]. Some criteria should be defined by the truster for classification of knowledge. Let us remark that this evaluation cannot be defined in a general way, that is, it must be defined by each O-grantor according to its interoperability security policy.

This parameter is evaluated based on the XeNA framework presented in [HCBCD09].

Finally, we note here that initially our model will be based on the knowledge and reputation evaluation when there is no previous interactions between the trustee and the truster.

3.5 Influence Trust Rules Presentation

Next we introduce how to represent the *influence trust rules in this framework*. Basically, an influence trust rules denotes a property that must be checked any time during the collaboration of the different participants. They allow us to ensure that the trust valuation of properties such as, it is *dynamic*, depends on previous interactions of the users, etc. After defining the influence trust rules we present by using examples some of the most useful rules.

$$\begin{aligned} &\texttt{InfluenceRule} ::= \textbf{If } \textit{condition} \textbf{ then } \texttt{TrustBelief} \textbf{ influence } \texttt{TrustBelief} \\ &\texttt{TrustBelief} ::= \texttt{trust}(\texttt{j} \mapsto \texttt{org})^{\hat{T}_n}_{situation} \end{aligned} \tag{3.14}$$

Intuitively, a `InfluenceRule` consists of a boolean condition that activates the rule, and two `TrustBeliefs`. The meaning of any influence rule is that if the boolean condition holds then the first trust belief must influence in someway to the second one. In Figure 3.14 different rule patterns are presented.

- **Rule Patterns:**

$R1$	if $(sit_{org_1,1}\ \mathbf{I_{sit,\alpha_1}}\ sit_{org_1,2}) \wedge (\alpha_1 \neq 0)$then
	$\texttt{trust}(\texttt{trustee}_1 \mapsto \texttt{org}_1)^{\hat{T}_n}_{sit_{org_1,1}}$**influence** $\texttt{trust}(\texttt{trustee}_1 \mapsto \texttt{org}_1)^{\hat{T}_n}_{sit_{org_1,2}}$
$R2$	if true then
	$\texttt{trust}(\texttt{trustee}_1 \mapsto \texttt{org}_1)^{\hat{T}_n}_{sit_{org_1,1}}$ **influence** $\texttt{trust}(\texttt{trustee}_1 \mapsto \texttt{org}_1)^{\hat{T}_{n+i}}_{sit_{org_1,1}}$
$R3$	if $(\texttt{memberOf}(org_i, subject_j))$then
	$\texttt{trust}(\texttt{org}_\texttt{i} \mapsto org_m)^{\hat{T}_n}_{sit_{org_{m_i},1}}$**influence** $\texttt{trust}(\texttt{subject}_\texttt{j} \mapsto org_m)^{\hat{T}_{n+1}}_{sit_{org_m,1}} \wedge$
	$\texttt{trust}(\texttt{subject}_\texttt{j} \mapsto org_m)^{\hat{T}_n}_{sit_{org_m,1}}$**influence** $\texttt{trust}(\texttt{org}_\texttt{i} \mapsto org_m)^{\hat{T}_n}_{sit_{org_m,1}}$
$R4$	if $\texttt{memberOf}(org_i, subject_j) \wedge \texttt{memberOf}(org_i, subject_k)$then
	$\texttt{trust}(\texttt{subject}_\texttt{j} \mapsto org_i)^{\hat{T}_n}_{sit_{org_i,1}}$**influence** $\texttt{trust}(\texttt{subject}_\texttt{k} \mapsto org_i)^{\hat{T}_{n+1}}_{sit_{org_i,1}}$
$R5$	if $\texttt{trust}(\texttt{trustee}_1 \mapsto org_1)^{\hat{T}_n}_{sit_{org_1,1}}$ **influence** $\texttt{trust}(\texttt{trustee}_2 \mapsto org_2)^{\hat{T}_k}_{sit_{org_2,1}} \wedge$
	$\texttt{trust}(\texttt{trustee}_2 \mapsto org_2)^{\hat{T}_k}_{sit_{org_2,1}}$ **influence** $\texttt{trust}(\texttt{trustee}_3 \mapsto org_3)^{\hat{T}_j}_{sit_{org_3,1}} \wedge$
	org_1 **is friend of** org_3 then
	$\texttt{trust}(\texttt{trustee}_1 \mapsto org_1)^{\hat{T}_n}_{sit_{org_1,1}}$ **influence** $\texttt{trust}(\texttt{trustee}_3 \mapsto org_3)^{\hat{T}_j}_{sit_{org_3,1}}$
$R6$	if $(org_1$ **is friend of** $org_2) \wedge (sit_{org_1,1}$**is equivalent** $sit_{org_2,1})$ then
	$\texttt{trust}(\texttt{org}_3 \mapsto \texttt{org}_1)^{\hat{T}_{n-1}}_{sit_{org_1,1}}$ **influence** $\texttt{trust}(\texttt{org}_3 \mapsto \texttt{org}_2)^{\hat{T}_n}_{sit_{org_2,1}}$

Figure 3.14: Influence rules patterns.

In this section, any situation in MOE will be denoted by $sit_{org_m,i}$ where org_m is the organization that defines this situation and i is its identity.

- In Figure 3.14, the $R1$ pattern means that the trust belief related to $sit_{org_1,1}$ at a period $\hat{T}_n$ between two entities will influence their trust relationship related to $sit_{org_2,2}$ at the same period when we have an impact relation between the two situations $sit_{org_1,1}$ and $sit_{org_2,2}$.
  Example:

  $$\texttt{trust}(\texttt{e}_1 \mapsto \texttt{MMT})^{\hat{T}_4}_{\texttt{comment}\triangleright\texttt{public\_files}}\textbf{influence}\ \texttt{trust}(\texttt{e}_1 \mapsto \texttt{MMT})^{\hat{T}_4}_{\texttt{modify}\triangleright\texttt{files}}$$

  Since, we have that $\texttt{comment}\triangleright\texttt{public\_files}\ \mathbf{I_{sit,0.6}}\ \texttt{modify}\triangleright\texttt{files}$.

- The $R2$ pattern means that trust relationship is dynamic and each one will depend on those with the same parameter that happen at a previous periods.
  Example:

  $$\texttt{trust}(\texttt{e}_1 \mapsto MMT)^{\hat{T}_4}_{\texttt{modify}\triangleright\texttt{files}}\textbf{influence}\ \texttt{trust}(\texttt{e}_1 \mapsto MMT)^{\hat{T}_7}_{\texttt{modify}\triangleright\texttt{files}}$$

- The $R3$ pattern means that the trust belief of a user regarding a truster at a period of time will contribute to construct the trust belief of its organization regarding the same truster and the same period. Moreover, the trust belief of an organization regarding a truster will influence the trust belief of their users

related to this truster at the next period.
Example:

$$\texttt{trust}(\texttt{ENSI} \mapsto MMT)^{\hat{T}_4}_{\texttt{modify}\triangleright\texttt{files}} \textbf{ influence } \texttt{trust}(\texttt{e}_1 \mapsto \texttt{MMT})^{\hat{T}_5}_{\texttt{modify}\triangleright\texttt{files}}$$
$$\texttt{trust}(\texttt{e}_1 \mapsto MMT)^{\hat{T}_5}_{\texttt{modify}\triangleright\texttt{files}} \textbf{ influence } \texttt{trust}(\texttt{ENSI} \mapsto \texttt{MMT})^{\hat{T}_5}_{\texttt{modify}\triangleright\texttt{files}}$$

- The $R4$ pattern means that the trust belief of a user regarding a truster related to a situation will influence the trust belief of any user from his organization regarding the same truster at the next period.
Example:

$$\texttt{trust}(\texttt{e}_1 \mapsto \texttt{MMT})^{\hat{T}_5}_{\texttt{modify}\triangleright\texttt{files}} \textbf{influence } \texttt{trust}(\texttt{e}_2 \mapsto \texttt{MMT})^{\hat{T}_5}_{\texttt{modify}\triangleright\texttt{files}}$$

- The $R5$ pattern means that this influence relation will be transitive if the truster of the first trust relation (org_1) is considered as a friend of the truster of the last relation (org_3).
Example: We have ENSI **is friend of** MMT.

$$\texttt{trust}(\texttt{u}_1 \mapsto \texttt{ENSI})^{\hat{T}_4}_{\texttt{write}\triangleright\texttt{report}} \textbf{ influence } \texttt{trust}(\texttt{u}_2 \mapsto \texttt{TMSP})^{\hat{T}_5}_{\texttt{edit}\triangleright\texttt{delivrables}}$$
$$\textit{and}$$
$$\texttt{trust}(\texttt{u}_3 \mapsto \texttt{TMSP})^{\hat{T}_5}_{\texttt{edit}\triangleright\texttt{livrables}} \textbf{ influence } \texttt{trust}(\texttt{UCM} \mapsto \texttt{MMT})^{\hat{T}_7}_{\texttt{modify}\triangleright\texttt{files}}$$
$$\textbf{so } \texttt{trust}(\texttt{IBM} \mapsto \texttt{MMT})^{\hat{T}_4}_{\texttt{write}\triangleright\texttt{report}} \textbf{ influence } \texttt{trust}(\texttt{UCM} \mapsto \texttt{MMT})^{\hat{T}_7}_{\texttt{modify}\triangleright\texttt{files}}$$

- The $R6$ pattern means that an organization will be influenced by the recommendation related to a situation $\text{sit}_{org_1,1}$ of other truster (org_2) only if this truster is considered as its friend and if it exists an equivalent situation $\text{sit}_{org_2,1}$ to $\text{sit}_{org_1,1}$.
Example: we have that:
  - MMT **is friend of** TMSP.
  - `modify▷files` **is equivalent** `add_remarks▷excel_documents`.
  - `modify▷files`(resp. `add_remarks▷excel_files`) is a situation of MMT (resp. TMSP).

$$\texttt{trust}(\texttt{UCM} \mapsto \texttt{MMT})^{\hat{T}_{n-1}}_{\texttt{modify}\triangleright\texttt{files}} \textbf{ influence } \texttt{trust}(\texttt{UCM} \mapsto \texttt{TMSP})^{\hat{T}_n}_{\texttt{add\_remarks}\triangleright\texttt{excel\_documents}}$$

3.6 Trust Vectors Presentation

A trust relationship between a truster and a trustee in our system is associated to a trust vector and an evaluation. In our framework, we define two different types of trust vectors: one for organizations otv and another for users utv.

Definition 13 A *trust vector for an organization* is a tuple $otv = (E_o, R_o, K_o)$ where the first element $E_o \in [-1, 1]$ is the experience evaluation, $R_o \in [-1, 1]$ is the reputation of the trustee value and $K_o \in [0, 1]$ is the valuation concerning the knowledge of this organization.

A *trust vector of a user* is a tuple $utv = (I_u, E_u, K_u)$ where $I_u \in [-1, 1]$ is the evaluation of trust relationship of its own organization, $E_u \in [-1, 1]$ is the evaluation of the experience between u and the truster, and $K_u \in [0, 1]$ is the evaluation of the knowledge concerning this user.

Each vector is related to a situation and a time period. □

For any O-grantor, the evaluation of trust relationship is based on three elements:

1. The type of the trustee (a user or an organization).
2. The associated trust vector.
3. The *trust policy.*

A trust policy aims to define the different weights associated to the parameters of the trust vector. Thus, we define two different trust policies $W_o = (w_{eo}, w_{ro}, w_{ko})$ and $W_u = (w_{otv}, w_{eu}, w_{ku})$ for the otv and the utv respectively. The three elements of each vector belong to the range $[0, 1]$ and their sum is equal to 1. These values are specified by the administrator of the interoperability security policy. As a result, the evaluation of a trust vector otv with respect to W_o will be:

$$w_{eo} * E_o + w_{ro} * R_o + w_{ko} * K_o$$

while the evaluation of the trust vector utv with respect to W_u will be:

$$w_{otv} * I_u + w_{eu} * E_u + w_{ku} * K_u$$

- **Trust Vectors Comparison**

Let us remark that in some situations, the administrator of the interoperability security policy has to *compare* two trust vectors. For instance, if the O-grantor needs the intervention of an external user to accomplish some tasks, several propositions may be available at the same time and thus the administrator may choose the best one among them.

Therefore, a comparison relation must be defined based on the calculated trust vector of each user. This relation can be defined only between two otvs or utvs related to the same situation and at the same time period. Next, we present a comparison method that is based on the following notions:

Trust value This method is based on the comparison of the evaluations of each vector. If there are not equal, the best one is the vector with the highest value.

Trust policy Within this method, we compare the evaluation of each parameter independently. So, we sort the different parameters of the vector according to the decreasing order of their weight. For instance, if $W_o = (w_{eo} = 0.5, w_{ro} = 0.2, w_{ko} = 0.3)$,

the trust vector will be organized as following $otv = (E_o, K_o, R_o)$. The parameter associated to the highest weight will be compared first. If the evaluation of this parameter is the same for the two trustees, we compare the next parameter else the vector with the greatest value related to this parameter is the best. The two vectors are equal, if the evaluations of the different parameters are the same.

In the following, we introduce our algorithm to compare two trust vectors in a MOE scenario.

1. We use the *comparison based trust value*. If the two vectors are not equal, the operation is finished else we continue in 2.

2. The *comparison based trust policy* is applied. In case where we compare two users, if the two vectors are also equal, we will compare the trust vector of their organization.

This last section concludes our first contribution that proposes a general trust model that takes into account several parameters of MOE environment. The first use of our approach was its integration into OrBAC. In the next part, we will detail this second contribution.

3.7 TRUST-OrBAC: Trust Integration into OrBAC

We recall that a context is a condition over the environment, and it must be satisfied to activate the security rule. In [CC08], several contexts, temporal, spatial, user-declared, prerequisite and provisional contexts for OrBAC were described. A context in OrBAC is defined with this rule $\texttt{Hold}(org, subject_i, action_j, object_k, ctx_m)$ where the context ctx_m is specified in an organization org (VPO or GVPO in our proposal) among a subject $subject_i$, an action $action_j$ and an object $object_k$. To control the activation context, the system has to give some information to check if the condition is satisfied or not [CC08]. The establishment of this system needs the implementation of different components. The Figure 3.15 shows the architecture of this system for an O-grantor. The gray boxes are new components specified for the trust purpose.

Trust context permits to check if the trust levels of the requesters (the organization and the user) respect the administrator beliefs or not.

3.7.1 Trust Classes Presentation

There are three concepts that are incorporated in order to manage this context. These are *trust subject class*, *trust organization class*, and *trust class*. Basically, theses concepts propose a classification of the behavior of the abstract entities in MOE. A *trust subject* and a *trust organization classes* are intervals $[c_1, c_2] : -1 \leq c_1 \leq c_2 \leq 1$. Each organization might define several trust subject and trust organization classes. The set of all trust subject/organization classes are denoted by TSC and TOC respectively. Based on these two elements, we define a *trust class* as a tuple (tsc, toc) where $tsc \in$ TSC and $toc \in$ TOC, and the set of all trust classes will be denoted by TC.

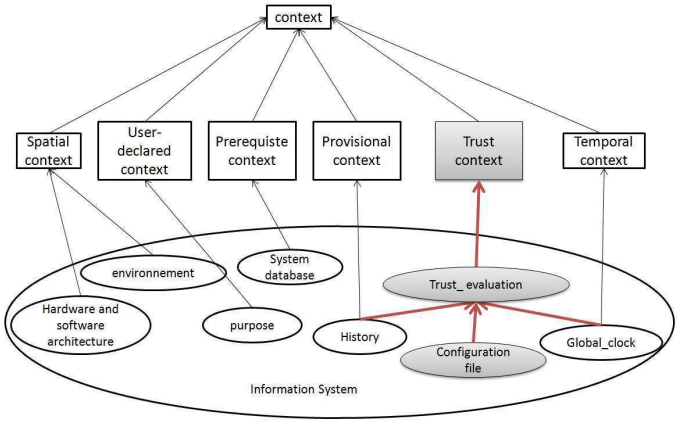


Figure 3.15: System Information.

Example 12 We show an example of a possible trust classes

- A class toc1=[-0.1,0.3]: The O-grantor trusts neutrally any organization which trust evaluation belongs to [-0.1,0.3].
- A class toc2=[0.2,0.5]: The O-grantor trusts slightly any organization which trust evaluation belongs to [0.2,0.5].
- A class tsc1=[-0.4,0.3]: The O-grantor trusts neutrally a user which trust evaluation belongs to [-0.4,0.3].
- A class tsc2=[0.4,1]: The O-grantor trusts moderately a user which trust evaluation belongs to [0.4,1].

An example of a rule taking into account the trust context may be illustrated in this example:

An engineer can modify confidential_files only if

1. he is trusted moderately this means that the degree of belief assigned to its trust relationship regarding this truster related to the situation `modify▷confidential_files` related to the previous period must belong to tsc2 [0.4,1].
   or
2. he is trusted neutrally and his organization is trusted slightly.

After each period, the system evaluation mechanism evaluates the trust value for the possible trustees related to the different situations. These evaluations are determined by a function system that is based on our trust vectors and policy. These evaluations are based on the history log that contains the different interactions.

After each period, these evaluations will permit to assign a user or an organization to a set of trust classes for each situations. This information (the trustee, the

situation, the assigned classes) will be saved on the configuration file that will be updated after each period. To validate a given query for an access, we have to retrieve the assigned classes to the requester (user and his organization) related to the needed situation from the configuration file and to check if these classes respect the context or not.

3.7.2 Representation of Different Trust Contexts

In order to represent this context, we update the existent object system information(Global_clock) and define a new predicate (getSituation).

1. We will use the Global_clock defined for the temporal context [CC08]. Four attributes are associated to it: Time, Day, Week, Month and Year. We add the Period attribute that makes possible to obtain the current period for the evaluation of the trust.

2. getSituation(a,o,s) is a new predicate that determines the situation related to the request. It is correct if the action a and the abject o can be mapped to the situation s.

Finally, we present four trust contexts using the OrBAC notation in Figure 3.16. Based on them, the administrator is able to define a composed context in order to specify the maximum (and\or minimum) trust level of the user and organization in any OrBAC rule.

For example: $\texttt{Hold}(\texttt{Any2MMT}, e_1, \texttt{download}, \texttt{trace.txt}, \texttt{trustorgmin}(0.5))$ permits to check if the trust level of the ENSI (the organization of the subject e_1) related to the situation $\texttt{read} \triangleright \texttt{confidential\_files}$ (see the assignment table in Figure 3.7) is more than 0.5.

3.7.3 Composed Trust Context

Basic trust contexts will be used with the context algebra in order to define the trust classes.

Example 13 To continue the Example 12, we have to define these contexts:

- Slightly_trusted_organization: $\texttt{trustorgmin}(0.2)$ & $\texttt{trustorgmax}(0.5)$
- Neutrally_trusted_user: $\texttt{trustusermin}(-0.4)$ & $\texttt{trustusermax}(0.3)$
- Moderately_trusted_user: $\texttt{trustusermin}(0.4)$ & $\texttt{trustusermax}(1)$

As a result, the trust context of the example 12 will be defined as:
Moderately_trusted_user $\bigoplus$ (Slightly_trusted_organization & Neutrally_trusted_user)
and the rule2 in the example 2 will be defined as:

$$\texttt{Rule2} ::= \texttt{permission}(\texttt{Any2}MMT, engineer, modify, \texttt{confidential\_files}, \texttt{Moderately\_trusted\_user} \bigoplus (\texttt{Slightly\_trusted\_organization} \& \texttt{Neutrally\_trusted\_user}))$$

trustorgmax :	$\forall org_s \in$ Organizations, $\forall sit_{org_i,1} \in$ SIT, $\forall action_j \in$ Actions, $\forall object_k \in$ Objects and $v \in [-1,1]$ Hold$(org, subject_l, action_j, object_k,$ trustorgmax$(v)) \leftarrow$ getSituation$(action_j, object_k, sit_{org_i,1}) \wedge$ memberof$(org_s, subject_l) \wedge$ Period$(Global_clock, period) \wedge$ trust_evaluation$(org_s, sit_{org_i,1}, period) < v$
trustorgmin :	$\forall org_s \in$ Organizations, $\forall sit_{org_i,1} \in$ SIT, $\forall action_j \in$ Actions, $\forall object_k \in$ Objects and $v \in [-1,1]$ Hold$(org, subject_l, action_j, object_k,$ trustorgmin$(v)) \leftarrow$ getSituation$(action_j, object_k, sit_{org_i,1}) \wedge$ memberof$(org_s, subject_l) \wedge$ Period$(Global_clock, period) \wedge$ trust_evaluation$(org_s, sit_{org_i,1}, period) > v$
trustusermin :	$\forall org_s \in$ Organizations, $\forall sit_{org_i,1} \in$ SIT, $\forall action_j \in$ Actions, $\forall object_k \in$ Objects and $v \in [-1,1]$ Hold$(org, s, action_j, object_k,$ trustusermin$(v)) \leftarrow$ getSituation$(action_j, object_k, sit_{org_i,1}) \wedge$ Period$(Global_clock, period) \wedge$ trust_evaluation$(subject_l, sit_{org_i,1}, period) > v$
trustusermax :	$\forall org_s \in$ Organizations, $\forall sit_{org_i,1} \in$ SIT, $\forall action_j \in$ Actions, $\forall object_k \in$ Objects and $v \in [-1,1]$ Hold$(org, s, action_j, object_k,$ trustusermax$(v)) \leftarrow$ getSituation$(action_j, object_k, sit_{org_i,1}) \wedge$ Period$(Global_clock, period) \wedge$ trust_evaluation$(subject_l, sit_{org_i,1}, period) < v$

org is a GVPO or a VPO related to the O-grantor org_i

Figure 3.16: Trust contexts.

3.8 Case Study of TRUST-OrBAC

Our case study aims to highlight the new functionalities provided by our work with respect to similar models like XeNA [HCBCD09] and TrustBAC [CR06, RRC09].

In this case study four organizations, denoted by $lab1$, $lab2$, $indis1$ and $indis2$, are working in a French project called ISER. The first lab aims to share several versions of the ISER documentation (views): file_code, driver_code, interface_code.

The actions that can be performed in these views are: edited, validated, commented, copied, and deleted.

We denote by s_1 and s_2 the following situations edit▷interface_code and edit▷driver_code, being the possible roles in this project research_engineer and engineer denoted respectively r_1 and r_2.

In this case study, the choice of the different trust classes (See Figure 3.17) and their use with the rules are defined based on some experiments and they are fixed by the administrator.

The reputation parameter will not be used in this case study ($w_{ro} = 0$) since we do not have the right to access to the trust evaluation of the other partners. The different participants would hide their collaboration statistics for confidentiality reasons. In addition, we use as *evaluation function* the approach developed in [MCJ+11]. This function allows us to detect some vulnerabilities of the source, after any modification of a C program.

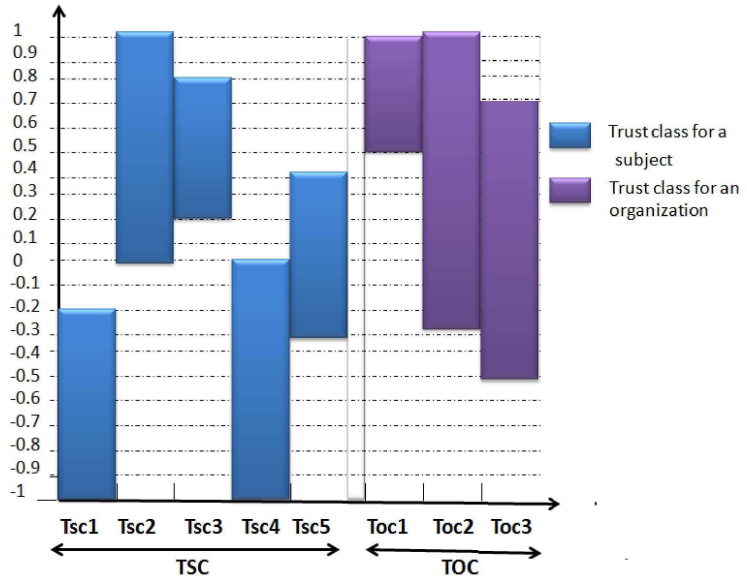


Figure 3.17: Trust classes definition for our case study.

Configuration file at $\hat{T}_3$

Trustee type	Trustee	Organization	sit	Value	Trust Classes
org	$indis1$	_	s_1	0.8	toc_1, toc_2
org	$indis1$	_	s_2	0.5	toc_1, toc_2
org	$indis2$	_	s_1	1	toc_1, toc_2
org	$indis2$	_	s_2	0.5	toc_1, toc_2
org	$lab1$	_	s_1	- 0.2	toc_3
sub	u_1	$indis1$	s_1	0.4	tsc_2,tsc_3
sub	u_3	$indis1$	s_1	0.8	tsc_3
sub	u_2	$indis2$	s_1	0.33	tsc_2,tsc_3,tsc_5
sub	u_2	$indis2$	s_2	0.1	tsc_2,tsc_5

Behaviour trace at $\hat{T}_4$

Req_id	Subject	Organization	sit	sat(b)
1527	u_1	$indis1$	s_1	0.85
2110	u_3	$indis1$	s_1	0.3
2116	u_5	$lab1$	s_1	-0.2
3150	u_4	$lab1$	s_2	0.3
7145	u_2	$indis2$	s_1	0.5
3189	u_4	$lab1$	s_2	0.8
7355	u_4	$lab1$	s_1	0.3
...	...	...	...	...

Figure 3.18: Configuration and logs files of the system.

Basically, in TRUST-OrBAC after any time period we provide a new *configuration table* that contains the list of the different users and organizations, their trust values related to the previous period, the associated trust classes, and the situation. In the left part of the Figure 3.18 is presented a part of this file from ISER project. For instance, the $6-th$ row represents the trust information of the user u_1, belonging to the organization $indis1$ related to the situation s_1 at the end of the period $\hat{T}_3$. This valuation was computed using the Definition 8, and according to Figure 3.17, the valuation 0.4 can be mapped in the trust user classes tsc_2 and tsc_3.

The update of the configuration file will be done after the end of each period. For instance to create the configuration file of $\hat{T}_i$ TRUST-OrBAC will take as input parameters the configuration file $\hat{T}_{i-1}$ and the behavior *logs* of the period $\hat{T}_i$. A behavior log of this system is presented on the right part of Figure 3.18. Each line contains the request identity (req_id), the subject, his organization, the situation and the evaluation of the behavior ($sat(b)$) for a permitted request.

Next, we detail in Figure 3.19 how the reception process of a request in TRUST-OrBAC is done. We consider that the user u_1 of the organization $indis_1$ applies to edit a file

called app.c. TRUST-OrBAC works as follows:

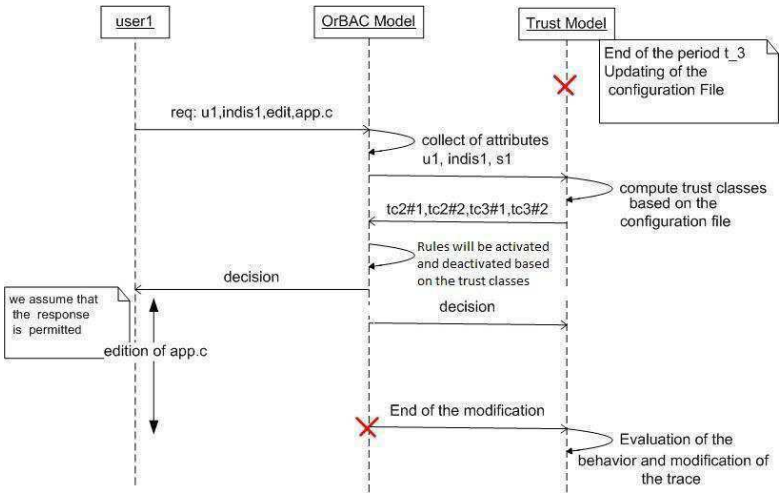


Figure 3.19: Reception process of a request.

1. First of all, TRUST-OrBAC determines the situation, that is edit▷interface_code[1].
2. TRUST-OrBAC collects the different attributes of u_1 by using different negotiation strategies [HCBCD09].
3. The situation, the user and the organization will be forwarded to the trust model. Next, it retrieves the information related to the user u_1 and the organization $indis1$ from the configuration file.
4. According to Figure 3.17, the trust subject and object classes associated to this request are toc_1, toc_2 and tsc_2, tsc_3.
5. These trust classes will permit to activate or deactivate some rules. Actually, the abstract entities are ready, based on the activated rules and the inference engine a decision will be taken and sent to the requester and to the trust module. If this request is permitted, the user will be authorized to modify this file.

3.8.1 Discussion

We discuss in this part the relevant properties of TRUST-OrBAC with previous approaches as: XeNA, and TrustBAC. Figure 3.20(1) presents one diagram that illustrates the dynamic trust value of some users and organizations. It shows the trust level of the user u_1, u_2 and the organization $indis1$ related to the situation s_1 during the different periods $\hat{T}_0$ to $\hat{T}_{14}$. Figure 3.20(2) represents the dynamic response of the same request sent by the user u_1 in order to perform the situation s_1 with the three models.

[1]since the file app.c belongs to the view interface_code

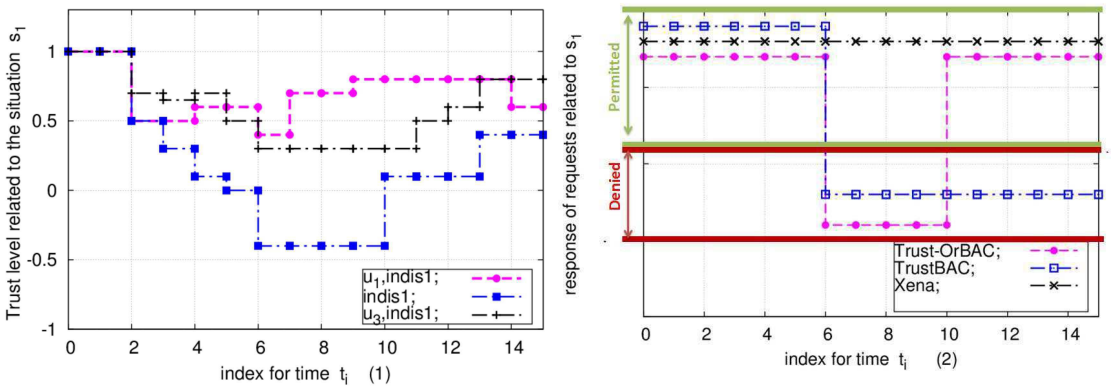


Figure 3.20: Dynamic trust level and its influence on the response of the same request.

1. Time influence: We studied here the responses of the three trust models regarding the same request with the same environment conditions and the same interoperability policy during several periods. For TrustBAC, the response may be changed based on the recent behaviors of the users. With our approach (TRUST-OrBAC), this may be caused due the behaviors of the users or his organization. Indeed, in MOE, the collaboration is defined with the organization. Sometimes, we do not know the real user [KDBK09]. For this reason, the use of trust value only with the user cannot satisfy the needs of this environment. Finally for XeNA, the response is the same during these periods since this framework does not address the behaviour evaluation and its influence on the request decision.

2. Punishment: The behavior of the organization may have a bad influence. With our solution, we may have two users from two organizations providing the same attributes and having the same trust level and they will have different responses regarding the same request. This situation happens because the two O-grantees that need the service do not belong to the same trust classes.

3. Rewards: In TRUST-OrBAC the influence of the good behavior of the organization in trust is taken into account. For instance, In the Figure 3.20(2) after $\hat{T}_6$, the response of any request sent by u_1 related to the situation s_1 is always denied with TrustBAC. However, this response may be changed with TRUST-OrBAC, i.e, at $\hat{T}_{10}$. The trust level of the $indis1$ increases based on some good interactions of other users. This offers the possibility to u_1 to regain some rights that cannot be obtained with TrustBAC.

4. We report the important role of the situation for the same subject, in the same period. This is similar for the three models, the trust model depends on the situation. However, the definition of this concept is only detailed in our framework.

Finally, we summarize the three approaches in the following table:

	XeNA	TrustBAC	Our proposal: TRUST-OrBAC
Time influence	✓	X	X
Influence of the organization behavior	X	X	✓ (punishment and rewards)
Access Control Model	Extended RBAC	RBAC	OrBAC
Influence of the user behavior	X	✓	✓
Situation definition	Ambiguous	Ambiguous	Detailed
Knowledge	Collection of attributes with negotiation strategies	Ambiguous	Inspired from XeNA

Table 3.1: Trust model in Role Based Access Control.

3.9 The Trust Framework Architecture

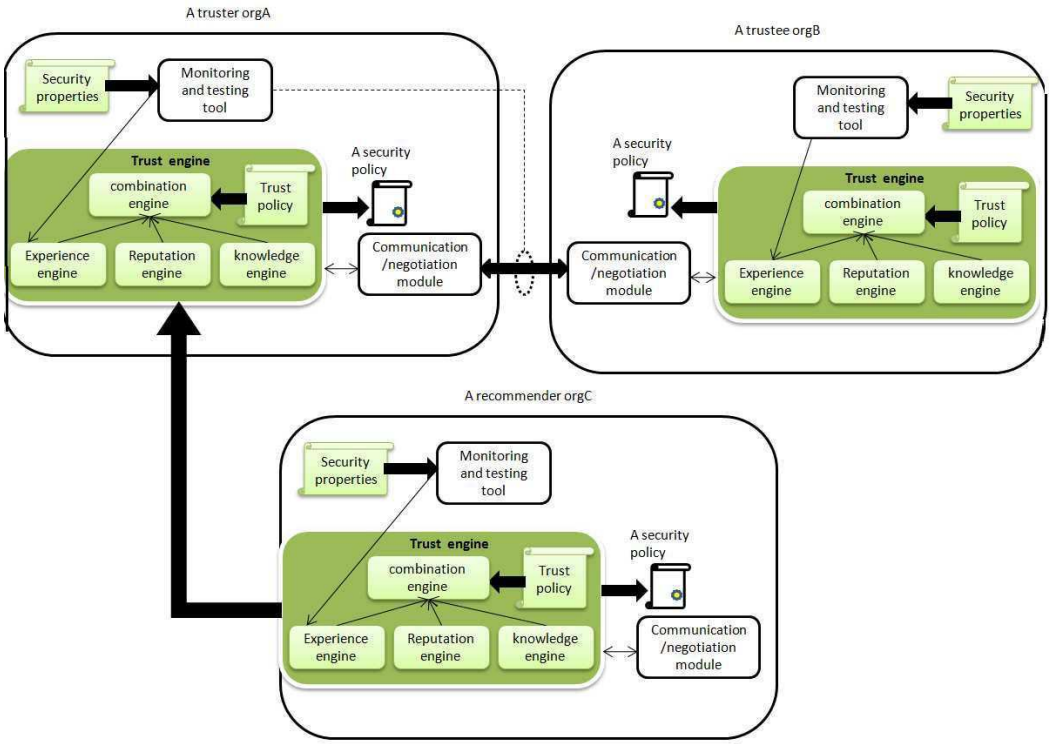


Figure 3.21: A possible architecture of our trust framework in MOE.

In Figure 3.21, we present the architecture of our solution that may be used to

enhance the security level and the dynamism of a security policy model. In the following, we define the role of each module.

Trust engine. It contains our new sub-engines, those that permit to evaluate the reputation, the experience and the knowledge. The last sub-engine called 'a combination engine' evaluates the trust level based on the trust policy (trust vectors and weights). After each period the trust values related to the different situations will be saved in a configuration file that will be used by the security policy.

Security properties and monitoring tool. For each situation, we will have some security properties to respect. Based on them the monitoring tool will first collect and analyze the trace in order to evaluate each interaction. This value will be used by the experience engine in order to evaluate the experience parameter for the organizations and the users. In Chapter 5, this module will be more covered.

Security policy. It contains several rules that specify how to control the access of the shared resources. In our proposal, these rules will be based on the different trust values in order to activate or not a rule. In our framework, we integrate a trust context with the `OrBAC` model. This permits to have a policy:

- that takes into account the new interactions,
- with higher security level since it monitors the different events and any participants may loss or gain some rights if his trust level changes.

We note here that our method is easy to use. It is true that we combine a big set of parameters. However, this framework can be configured based on the need of the administrator. The configuration of the trust policy permits to ignore some parameters by modifying their weights.

3.10 Conclusion

In this chapter, we presented our new trust framework for MOE environments.

First of all, we have identified the different trust challenges and the existent solutions. Next we have defined the different entities forming the trust relationships. Different relations between these entities as the `equivalence` and `Impact` relations are also detailed. Moreover, we have defined the possible trust parameters in MOE. An evaluation of these parameters with some experiments are also detailed. We have to mention that we have also studied the influence of the organization trust level regarding the rights of their users.

Moreover, trust rules to check the integrity of our proposal have been detailed. Next, we have presented our new model `TRUST-OrBAC` that is based on the integration of our new trust model into the `OrBAC` model. A case study with some experiments is also introduced and discussed. Finally, a possible architecture of our solution is presented.

Chapter 4

Trust Ontology Based on Access Control Parameters in MOE

"To be trusted is a greater compliment than being loved."
George MacDonald

Contents

4.1 Introduction

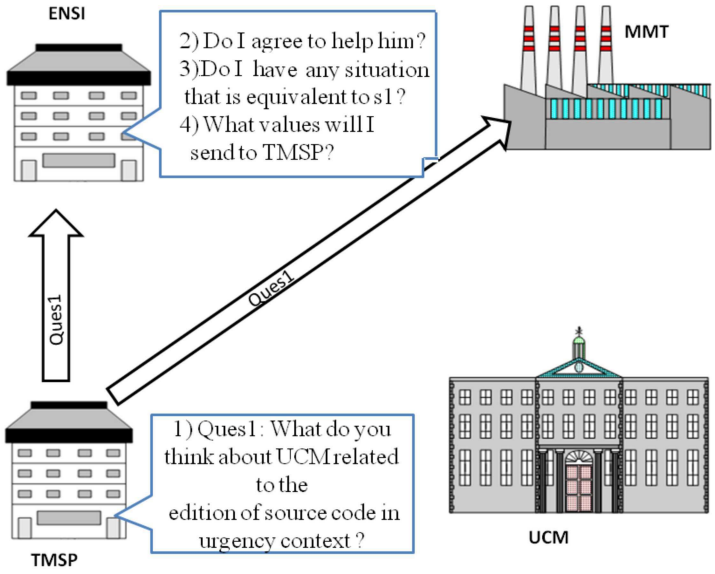


Figure 4.1: The reputation process.

In this chapter, we detail the reputation parameter that aims to gather and aggregate feedback about an entity from other participants. An important issue to solve related to this parameter is how to understand and use the received feedback.

Let us introduce this need in the scenario presented in Figure 4.1. The O-grantor and the O-grantee are the organizations TMSP and UCM respectively. According to the previous chapter, only a subset of organizations can participate in the evaluation of the reputation of UCM. This set contains the friends of TMSP, denoted $\text{fr(TMSP)}^{\hat{T}_n}$, those are ENSI, MMT[1]. The reputation evaluated by TMSP regarding UCM related to the situation s_1 "`edit▷source_urgency`" is based on recommendation values sent by ENSI and MMT. In fact, TMSP sends a request **Ques1** to their friends:

> **Ques1**: "what do you think about the organization UCM regarding the situation `edit▷source_urgency`"?

When a recommender receives this request, it may refuse to give any response for security reasons (question 2 in Figure 4.1). Otherwise, it has to send its recommendation value. To do this we have to address these two issues:

- The first one is **how the recommender makes a decision about an unknown situation** (that is composed of local parameters of the requester). Thus, we will study how to define this parameter and how to study the equivalence between the required situation and a local one (question 3 in Figure 4.1).

[1]We have to thank the Montimage company (MMT) for their collaboration. Montimage is an innovative company created in 2004 and located in Paris. It is specialized in software development and monitoring services.

- The second issue is about **how to define the recommendation value**, called also feedback(question 4) in Figure 4.1. This value will allow us to evaluate the reputation of an organization applied to a situation.

In particular, in this chapter we will detail the reputation parameter in order to present:

1. How organizations share the feedback?
2. How to define equivalence among situations?
3. How to provide the recommendation value?

Let us note that the way used to share feedback among the different organizations is critical and different strategies based on ontologies have been proposed [GPH03, DM08]. Indeed, an ontology lays the ground rules for modeling a domain by defining the basic terms and relations to make up the vocabulary of this topic field [NFF+91]. These works define different *trust ontologies* that can be used in distributed systems like `MOE`. However, the previous ontologies are not designed to be used with access control models. In our proposal, we will provide a generic trust presentation related to `OrBAC` model with the definition of new trust parameters. Next, we will implement an algorithm that allows us to determine a set of equivalent situations. This algorithm will be based on the similarity function concept [CCBCC08b]. Besides, we will detail different approaches that can be used in order to define the recommendation value. A case study described along the chapter is detailed and discussed to illustrate our contributions.

The rest of this chapter is structured as follows. In Section 4.2 some related works are presented. In Section 4.5 our trust ontology and its classes are detailed. Next, in Sections 4.6 and 4.7 we present the mapping process and the recommendation evaluation approach. Finally, in Section 5.9 the conclusion is presented.

4.2 Related Works

As it is presented in Figure 4.2, our new contribution is related to different issues those are access control models in `MOE`, trust ontology research works in distributed systems and mapping systems for ontologies. We will not detail the first challenge that is well presented in the second chapter.

Different works have been proposed in the definition of trust ontology in distributed system [GPH03,TD04,DM08]. In [GPH03], the authors propose a solution that is an extension of FOAF (Friend Of A Friend) schema [Bri14]. They allow to express trust in people, statements, other contents of information sources. [GPH03] has been extended in [TD04] fusing on how the messages should be exchanged in the context of a communication environment. According to [DM08], all trust ontologies convey as the same objective which is the representation of trust relationships and

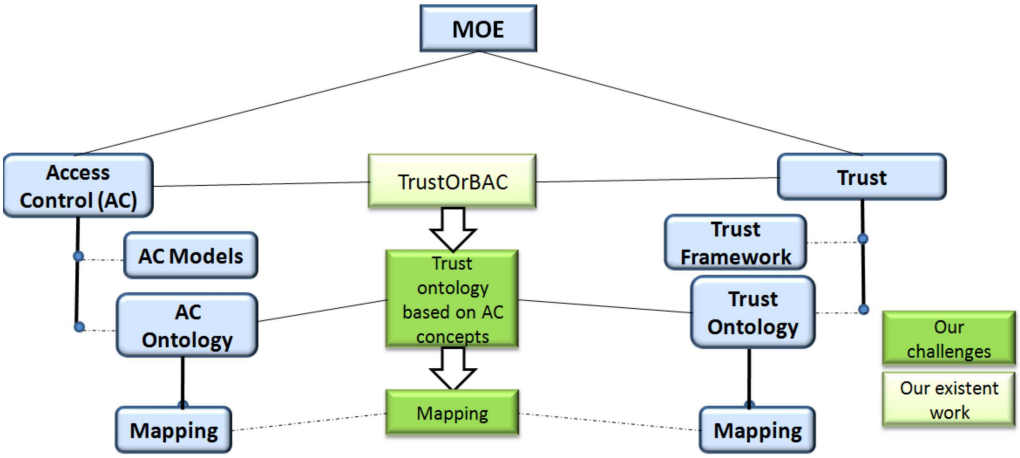


Figure 4.2: Our challenges.

situation. In this chapter, we focus on how to define the situation ontology based on the different parameters of the OrBAC model.

The last challenge is related to the ontological mapping [FN11]. Several mapping methodologies exist in the literature, and these are the closest to ours [JPE01, DMD+03,CCBCC08b]. In [JPE01] an approach called Cupid developed by Microsoft is presented. In this work authors present mappings between schema elements based on their names, data types, constraints, and schema structure. In [DMD+03], the author proposes a system that permits a semantic mapping between ontologies based on a machine learning approach. However, these mapping approaches are not enough in the case of interoperability security policies. In order to affine this mapping and to adapt them to this area, a solution proposed in [CCBCC08b] provides a mapping between two OrBAC ontologies. This proposal will be more detailed in Section 4.6 since our mapping solution will be an extension of it in order to take into consideration the trust entities.

4.3 Ontology Concept

An ontology is *"an explicit specification of a conceptualization where a conceptualization is an abstract, simplified view of the world that we wish to represent"* [Gru93]. This concept specifies a vocabulary that has to be understandable by the machine and people which represents a common knowledge of a domain. It is used for several reasons as the sharing of knowledge and reuse, the separation between the domain knowledge and the operational knowledge and the analyze of the domain knowledge [ASL07]. It has been used in several applications as software engineering, biomedical systems, the semantic Web, etc.

An ontology is composed of classes, their attributes with their possible values and the relationships between classes. The ontology and the different instances of classes compose the base knowledge of a domain. Next, we summarize the different

rules related to the creation of an ontology [ASL07]:

- There is no single correct way to model a domain there are always viable alternatives. The best solution depends on the application and the extensions that the designer aims to do.
- Ontology development is necessarily an iterative process.
- Concepts in the ontology should be close to objects (physical or logical) and relationships in the domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe the domain.

4.4 Ontology for Interoperability Use

In the literature, the use of an ontology in a distributed system for semantic interoperatbility is based on two approaches:

- Integrated approach: The different participants have to create a global ontology for the whole system. This is done by merging the local ones.
- Distributed approach[13]. This is realized where each organization used its local ontology. Each ontology is created without consideration of other entreprises. This approach is related to an interesting research area that is mapping between ontologies. This topic is critical in order to interoperate without the modification of local system or the conception of a new mechanism (i.e: global ontology).

4.5 Trust Ontology for Access Control

In order to define a trust ontology, we need:

- To design a global vocabulary about trust to ensure a comprehensible way to share feedback.
- To define the situation concept (its classes and properties) that facilitates the mapping process among the situations.

To the best of our knowledge, the proposed ontology is the first generic presentation of trust based on access control concepts in `MOE`. This ontology is depicted in Figure 4.3, its main ontology classes are: **trustRelationship** (presented in Section 4.5.1), **Situation** (presented in Section 4.5.2), and **TrustLevel** (presented in Section 4.5.3).

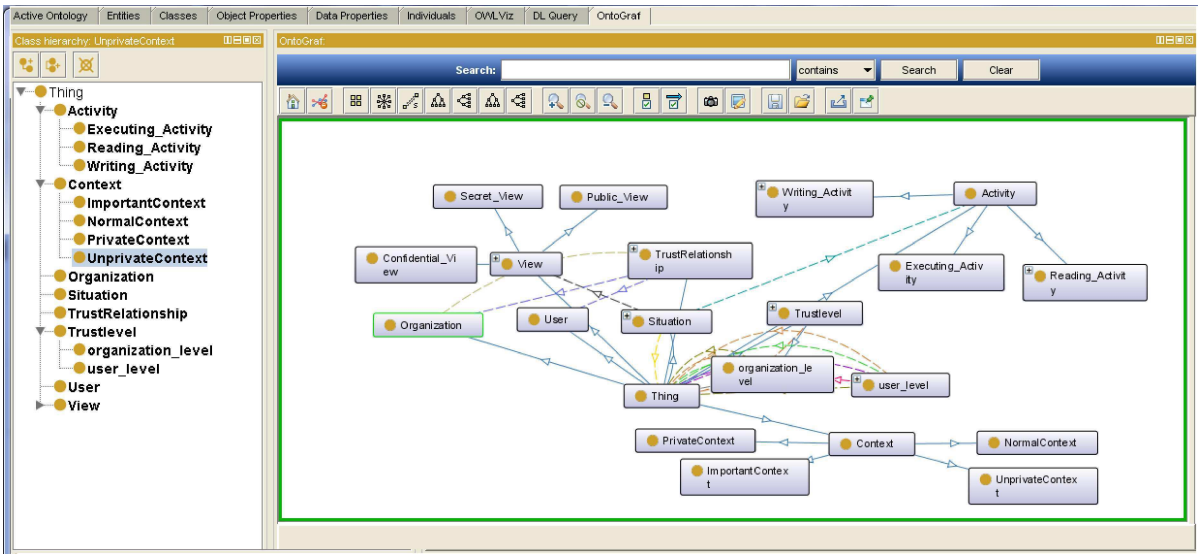


Figure 4.3: Trust ontology with the Protege Ontology Editor tool.

4.5.1 TrustRelationship class

In order to define our trust relationship class, the following five concepts: **Truster, Trustee, TimeStamp, Situation,** and **TrustLevel** are needed.

The **truster** in our system is an O-grantor. It is the organization that will provide the service or offer the access. Each O-grantor has to define its trust policy and to periodically evaluate the trust values of the different participants. Let us note that any organization may be a truster.

The **trustee** in MOE can be the user or the organization that applies for a service. The trust relationship aims to evaluate its trust level.

In our framework, the trust level of an entity is evaluated after each period with respect to the **TimeStamp**. The concept of time permits to illustrate the dynamism of our framework. For this reason, each relationship will be characterized by its evaluated time.

Finally, the **Situation** and **TrustLevel** are defined as two ontological classes.

4.5.2 Situation Class

In this chapter, we extend the definition presented in the chapter 3. With this extension, the situation class is defined as a tuple (a, v, ctx) where a is an activity (a set of actions), v is a view (a set of objects) and ctx is a context (a specific condition that activates or deactivates a rule).

OrBAC ontology gives a presentation of the three classes: views, activities and contexts [CCBCC08b]. In this work, we extend this definition for trust proposal; Figure 4.4 shows our situation taxonomy. Following the new trust classes related to the tuple (a, v, ctx) presented in the Figure 4.4 are introduced.

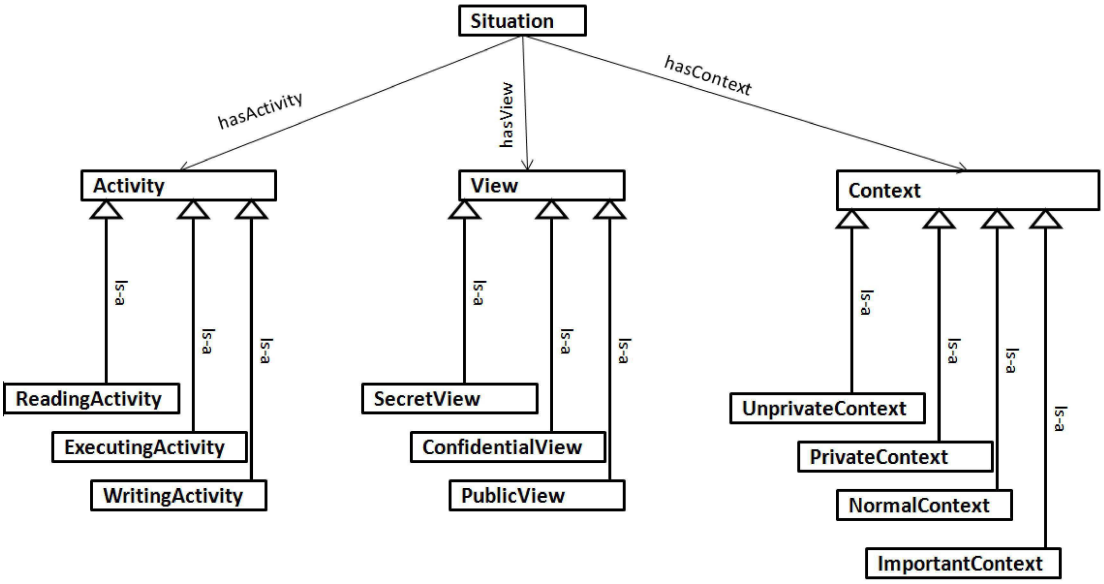


Figure 4.4: Situation Taxonomy.

a For any activity entity, we have three new subclasses that are related to the type of the activities. Indeed, any activity in an access control model [TCM12] belongs to at least one type of these classical classes *ReadingActivity*, *WritingActivity* or *ExecutingActivity*. The table 4.1 presents an example of an assignment of some `OrBAC` activities to these new ontology classes of TMSP university.

	ReadingActivity	WritingActivity	ExecutingActivity
declare			✓
manage	✓	✓	
add_note		✓	
submit			✓
consult	✓		
notify			✓

Table 4.1: Mapping between `OrBAC` activities and the new classes.

v The new subclasses of the view concept are the *PublicView*, *ConfidentialView* and *SecretView*. This conception is inspired from the information classification of the European Commission and the OCCAR (Organisation for Joint Armament Cooperation) organizations [wik14]: **(1)**Resources that contain information that may be used with any extern employees of another partner will be in *PublicView*. Any unclassified resource belongs to this type. The modification, the consultation of these data does not influence any financial, operational or personal problem. **(2)** A resource that contains a sensitive data belongs to the *ConfidentialView*. This information has an impact on the service level and performance of the enterprise. They may cause some financial loss, penalty,

loss of confidence, etc. **(3)** Finally, *SecretView* contains very serious personal and enterprise data. The malicious use of this document may cause a major economic impact, a fire of an employee, an interruption of relationship with another enterprise. They are available to some users that belong to a particular mission (project, task, etc).

ctx For the context, we have four new subclasses that are related to the *privacy* and *importance* of the context. On the one hand we have that a private context is defined for application proposals that means it depends on the local parameters as auditing results designed by the administrator, provisional context, logic context [CCB08]. Any context that does not belong to the private class will be an element of the unprivate class. Mapping between two private contexts is highly difficult. We say that a context is unprivate or general, when it can be understandable or used in several organizations as temporal and geographical contexts.

On the other hand, with respect to the concept of importance, we have that this property is determined by the administrator in order to highlight the importance of a context in a situation. We deal only with two classes important and normal context. The first one is defined for the context that must be considered in order to share trust knowledge between the different organizations.

In our running example, any context is defined as a normal one. That means it will be ignored in the reputation process. In this case the different situations `modify▷files▷at_night`, `modify▷files▷at_night` and `modify▷files` are similar. Otherwise, the administrator has to change this type.

4.5.3 Trustlevel Class

Figure 4.5 illustrates the components of the **TrustLevel** class. This class is defined in order to present a trust level of a user or an organization. For this, it has two subclasses that are *OrganizationLevel* and *UserLevel*. Both of them are related to the two classes *EvaluationMethod* and *TrustClassification*. The first one specifies the method used to evaluate the trust value that can be based on combining different parameters as it is defined in [ZL09,TACM12] or based on one parameter as the work in [WLWV09b].

The second one is based on Golbeck classification of trust [GPH03]. In this classification, there are 9 types: absolute distrust, high slightly distrust, moderate distrust, slightly distrust, neutral, slightly trust, moderate trust, high trust and absolute trust. This classification is used to have a more comprehensible trust evaluation. Since, the same trust value may be considered differently in two organizations.

Other properties are designed for trust level classes. They are the experience, the trust values related to the trustee and the *OrganizationLevel* that is only defined for the *UserLevel* class. The latter contains an instance of the *OrganizationLevel* class.

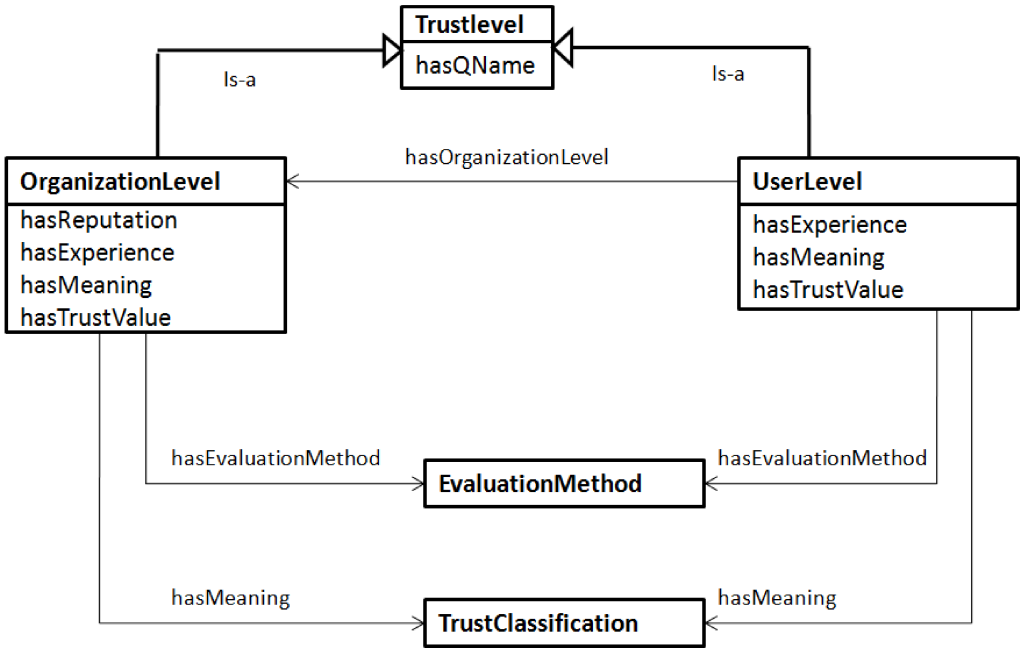


Figure 4.5: TrustLevel ontology- TrustLevel components.

4.6 Mapping Process

The second contribution in this chapter aims to provide a mapping process between an external situation and the local ones. We will present in the following, the mapping solution that is based on the new trust properties.

Different steps have to be realized during this process:

1. **Before the collaboration:**
   (a) For each participant in the MOE, we extend his `OrBAC` ontology based on the new trust parameters.
   (b) We install a trust web service in each organization. This web service will be responsible for providing the list of equivalent situations to an external one. This service will use two inputs, the trust ontology and a new table called Server Situation Mapping (SSM). In this table, we save the equivalent situations and equivalence rate with the external situations. Initially, this table is empty, and it will be filled during the communication.
2. **During the collaboration (Figure 4.6):**
   (a) Initially this web service is waiting for any request from an O-grantor.
   (b) When a request is received, the recommender will extract the requested situation and will consult its SSM table.

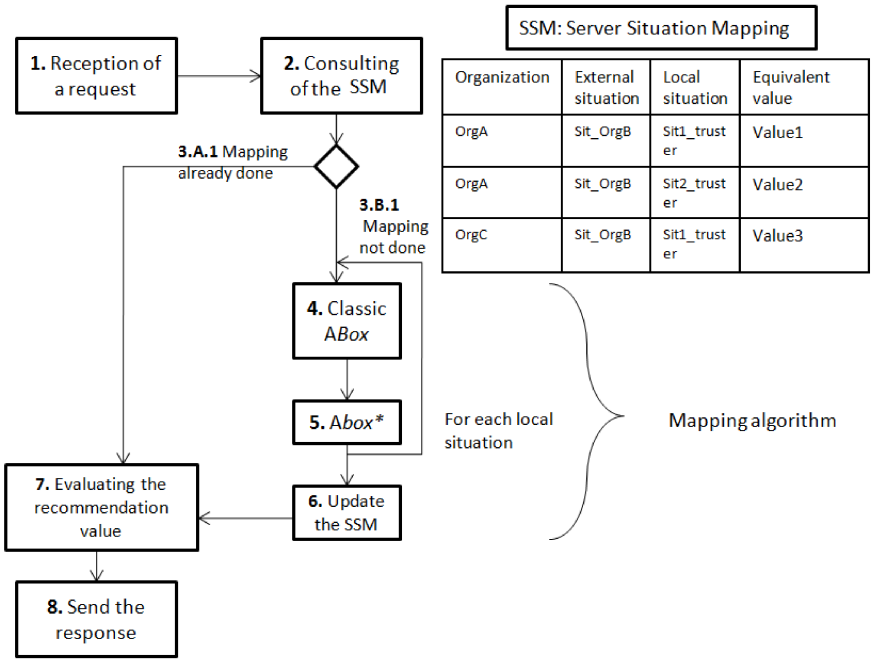


Figure 4.6: Different tasks of the trust web service.

(c) If the mapping is already done (that means the recommender has received a previous request related to the same situation), we go to the task 7 in order to determine the recommendation value.

(d) Otherwise, the recommender has to do the mapping algorithm that will check the equivalence between the requested situation and all the local situations of the recommender. Then it will update the SSM table.

4.6.1 Mapping Algorithm

During this process, each local situation will be compared with the requested situation based on the two algorithms: `ABox` and `ABox`*. The first one takes the ontology presentations of the:

- Two activities (a local activity with the activity to perform).
- Two views (a local view with the needed view).
- Two contexts (a local context with the actual context) as inputs.

The mapping is based on a structure and correspondence between attributes and their values. This relation defines a similarity function, denoted sim, between two entities. This function gives a value between [0,1], where 0 means that there is no similarity between the two entities and 1 means that the two entities are highly similar. This method is developed in [CCBCC08b] and does not take into consideration

the new trust classes and properties. Based on [CCBCC08b], we can say that two entities are similar if their similarity value given by the ABox algorithm is more than a threeshold_entities fixed by the administrator.

The second new algorithm takes as inputs the presentation ontology of the two situations and the mapping results of the ABox method. It permits to check the equivalence between two situations s_1 and s'_2 from different organizations by taking into account the new trust elements. It permits to determine the equivalence rate, denoted $P_{s'_2 \to s_1}$, between them. This value will belong to [0,1]. The closest the value to 1, the more the situations are equivalent. This new algorithm is presented in Figure 4.7.

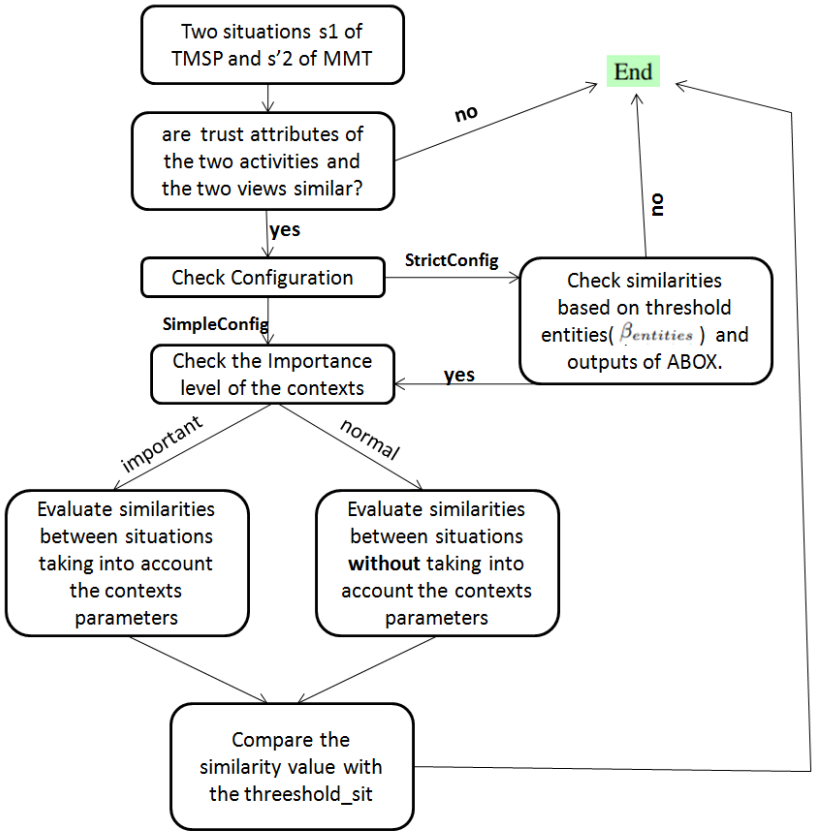


Figure 4.7: Equivalence between situations.

In the following, we introduce the ABox* algorithm in detail:

In our approach, we consider that an equivalence between two situations cannot happen if the trust parameters of their activities or views are not similar.

Example 1:

For instance: Presentation (v1) and Results(v'2) are two views that belong respectively to TMSP and MMT. These two views are similar regarding the mapping ABox. However, they are not equivalent based on their trust attributes (e.g. v1 is a public view and v'2 is a confidential one). Therefore, any situation s_1 and s'_2 that contain

respectively v1 and v'2 as views cannot be equivalent.

In our approach we provide two types of configuration , a *simpleconfig* and *strictconfig*. The first one is only based on the new trust parameters.

Example 2:
With *simpleconfig* configuration, if MMT asked for a reputation value related to 'add_comments▷results' from TMSP. The latter will try to find the set of situations that contains an activity where the type is writing and the view class is confidential (since add_comments belongs to the WritingActivity and results belongs to the ConfidentialView).

With respect to the equivalence rate, it might depend on the type of the situation contexts. If the two contexts are normal then the equivalence rate is the average of the similarity function of the two activities and views. Otherwise, it will depend also on the similarity function of the two contexts. With this configuration, we do not care about the similarities between the contexts. When we have heterogeneous systems, we can only use this configuration. Finally, we will have an equivalence between two situations only if the rate is more than the threshold_sit.

For the second configuration type (*strictconfig*), the equivalence depends on the ABox results and the trust parameters. We will have an equivalence:

- If the similarity values between the two activities and the two views are more than the threshold $\beta_{entities}$, the two contexts are not important and the equivalence rate between the two situations is more than threshold_sit.
- or if the two contexts are important, the similarities between the two activities, two views and two contexts are more than the threshold $\beta_{entities}$ and the equivalence rate between the two situations is more than threshold_sit.

Based on these results, the SSM will be updated and the process of looking for equivalent situations will be repeated until we compare the requested situation with all the local ones.

Evaluation: We have implemented two trust ontologies for TMSP and MMT with their instances detailed in Figure 4.8. We have applied several requests from MMT to TMSP to test the mapping algorithm. As an example, we show here the mapping algorithm behavior with the strict and the simple configuration for the situation s_1 'download▷logs▷during_the_implementation'. The curves of the left part of Figure 4.9 describe the percentage of the equivalent situation to s_1 after the modification of the $\beta_{entities}$ and the second one is done by the modification of the threshold_sit.

Based, on the left part of Figure 4.9, we remark that the number of equivalent situations with the simple configuration is more or equal to the strict configuration. Indeed with the second configuration, we are based on more conditions than the first one that compares only the trust concepts. This one is based on a generic mapping that is recommended for heterogenous organizations where it is highly difficult to find

	Activities number			Views number			Contexts number				Situation number
TMSP	6			9			3				162
MMT	9			8			3				216
	E.A	R.A	W.A	P.V	C.V	S.V	I.C	N.C	P.C	U.C	
TMSP	3	2	2	4	4	1	1	2	0	2	
MMT	5	2	3	2	5	1	2	1	1	2	

E.A: **Executing Activity** || R.A:**Reading Activity**|| W.A: **Writing Activity**|| P.V:**PublicView**|| C.V: **ConfidentialView**|| S.V:**Secret View** I.C:**ImportantContext**|| N.C: **Normal Context**|| P.C:**Private Context**|| U.C:**UnprivateContext**

Figure 4.8: TMSP and MMT ontologies composition.

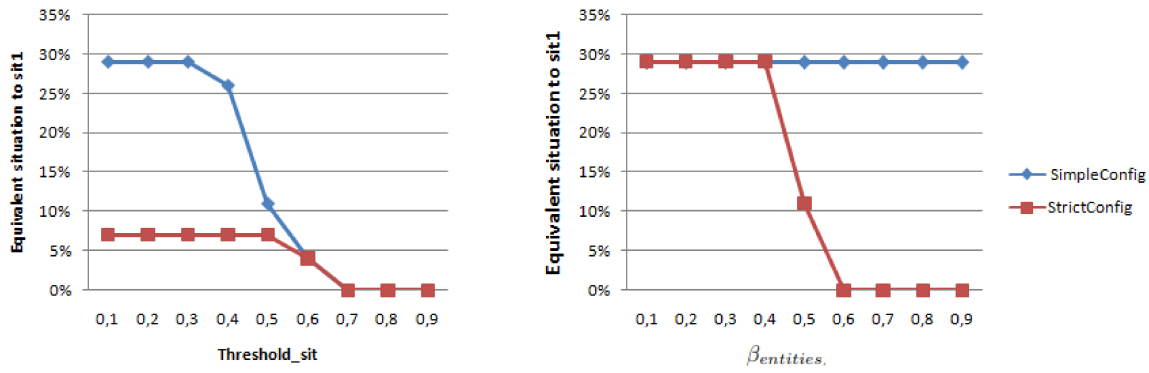


Figure 4.9: Comparison between simple and strict configuration.

similarity between the different entities. In the right part of Figure 4.9, we remark that with a small threshold for entities, the number of the equivalent situations with the two configurations will be the same. However, the simple configuration is not influenced by the modification of this threshold unlike the second one that will converge to zero.

4.7 The recommendation Value

In this step (task 7 of Figure 4.6), the recommender has a set of equivalent situations to the requested one from the O-grantor. It now has to resolve the third issue '*how to provide the recommendation value?*'. In the next example, we detail this problem.

Example 3:

We assume first that the equivalent situation set of s_3 (a situation of TMSP): `submit`▷`Livrables` according to MMT is { `update`▷`desciption_files` (s'_1), `update`▷`contract_templates`(s'_2), `download`▷`contract_templates`(s'_3), `download`▷`desciption_files`(s'_4)} where the equivalence rates between s_3 and these situations are detailed in the following table:

We note that based on [TCM12] any O-grantor (e.g. MMT) will evaluate the trust level of all the trustees related to its local situations after each period. Therefore,

Equivalence rates	s'_1	s'_2	s'_3	s'_4
s_3	0.46	0.5	0.5	0.75

Table 4.2: equivalence rates for s_3

the trust levels related to s'_1 , s'_2 , s'_3 and s'_4 are also evaluated. When MMT has to send its recommendation related to s_3, it will be based on the trust levels related to s'_1 , s'_2 , s'_3 and s'_4. So it has to choose between sending the trust level related to one of them or a combination of them. In the following, we detail this process.

In our approach, different possibilities to send a recommendation value to the truster are proposed. For each recommender, we collect the trust values regarding the different trustees related to the local situations. After this, we will keep only the trust beliefs related to the trustee and regarding any situations that belong to the equivalent set. We will denote by $\texttt{SIT}_{eq}(s_1)$ the set of the equivalent situations of s_1.

- **case1-** $\texttt{SIT}_{eq}(s_1) = \emptyset$ or no information about the trustee in the filtered file: Unavailable information will be sent as a response.
- **case2-** $\texttt{SIT}_{eq}(s_1) \neq \emptyset$:
  - Possibility1- nearest approach: We choose the nearest situation $s_{nearest}$ to s_1. Then, the trust value regarding $s_{nearest}$ related to the previous period will be sent to the O-grantor. For any $s_1 \in \texttt{SIT}_{org_B}$, we say that s'_i is the nearest situation to s_1 if its equivalence value is the highest one.

    Example 4:
    We say that s'_4 is the nearest situation to s_1 since $\mathrm{P}_{s'_4 \to s_1} > \mathrm{P}_{s'_1 \to s_1}$, $\mathrm{P}_{s'_4 \to s_1} > \mathrm{P}_{s'_2 \to s_1}$ and $\mathrm{P}_{s'_4 \to s_1} > \mathrm{P}_{s'_3 \to s_1}$.
  - Possibility2- optimistic or pessimistic approach: For these approaches, we first of all have to compare the different trust values stored in the filtered file. For the optimistic approach, the highest trust value will be sent to the O-grantor. For the second approach, the lowest one will be chosen as the recommendation value.
  - Possibility3- mathematical expectation approach: In this case, we will use all the values in the filtered file. In fact, the recommendation value of org_B regarding org_C at T_k that will be sent to org_A is the weighted average of all the trust values in the file (the weight of each situation s'_i will be $P_{s'_i \to s_1}$):
    $\sum_{i=1}^{i=n} P_{s'_i \to s_1} * trust(org_C \to org_B)_{s'_i}^{T_{k-1}}$ in which n is the number of equivalent situations to s1.

4.8 Conclusions and Future Work

In this chapter we have presented a new trust ontology based on access control concepts. We have extended the `OrBAC` ontology with new trust classes that are

presented and detailed with some examples using the protege tool. Moreover, we have also proposed a mapping algorithm in order to study equivalence between two situations of different participants. Two ontologies, one of the TMSP university and another of the MMT company, are implemented. The mapping algorithm is realized between these two ontologies and some results are presented and discussed. This approach details how to use the reputation process with the `TRUST-OrBAC` model to enhance the security of the system. Several subprocesses are proposed to define the trust presentation, to share the trust beliefs, to map between situations and to evaluate the recommendation values.

Chapter 5

Monitoring for Trust Evaluation

"If you want to succeed in your life, remember this phrase: The past does not equal the future. Because you failed yesterday, or all day today, or a moment ago; or for the last six months, the last sixteen years; or the last fifty years of life, doesn t mean anything All that matters is: What are you going to do, right now? "
Anthony Robbins

Contents

5.1 Introduction

"Experience is the teacher of all things." (Julius Caesar)

The importance of the experience parameter encourages several researchers to define and evaluate it. This parameter evaluation depends on the historic interactions of the trustee. However, different challenges are still open. For instance, some assumptions related to the monitoring task of the behavior have to be discussed: "How to monitor the behavior" of the trustee is not addressed or detailed in [CR06, RRC09, TACM12, TAC12b]. These works assume that the behavior evaluation can be used as an input. In order to address this issue, we propose in this chapter:

- A methodology on how to evaluate the different interactions.
- An extension of a monitoring tool for the evaluation of a behavior based on the trust needs.

Our approach will be used as an extension of our previous works [TACM12, TAC12b] in order to have a prototype of the whole trust framework system. To achieve this objective, we have used the Deep Packet Inspection (DPI) technique that is a form of computer network packet filtering. It is the process of capturing network traffic, analyzing and inspecting what is happening in the network. This technique is very useful to design some tools as MMT. This latter is a monitoring tool based on the DPI technique. This tool allows providing a real-time visibility of network traffic. It provides network, application, flow and user level visibility. It facilitates network security, performance monitoring and operation troubleshooting. MMT rules engine can correlate network and application events in order to detect operational, security and performance incidents. Moreover, this tool incorporates a plug-in architecture that simplifies its extension to be used in several applications. For these reasons, we have chosen to extend the MMT tool in order to evaluate the trust of the different entities.

In order to achieve this goal, (1) a new plug-in called 'trust-plug' is developed to analyze the trace and to determine the different elements that will be used to define trust (2) the formalism permitting the specification of the MMT rule is updated and (3) an extension of the MMT tasks by adding periodical verification and trust level notification is developed.

The rest of the chapter is structured as follows. In Section 5.2 we recall the experience definition. Next, we give an overview of our passive testing approach in Section 5.3. Section 5.4 details the MMT tool that will be used in our proposal. The next section will show how to evaluate the satisfactory function and the possible strategies to evaluate a behavior. Section 5.6 introduces our trust traces. Section 5.7 illustrates how to integrate the MMT tool with our trust framework. Moreover, a case study is detailed in Section 5.8. Finally, we conclude the chapter and we propose some relevant future works in Section 5.9.

5.2 Experience Parameter

We recall in this section the experience definition.

- **Experience learning**

Definition 1 Experience learning aims to establish wisdom in making decision. It is based on the evaluation of the previous interactions between the **trustee** and the **truster** related on a specific **situation** at a **period of time.** □

There are two types of experiences:

- The experience of the trustee organization that takes into consideration the previous *behaviors* of all users of this organization,
- The direct experience where only the previous *behaviors* between this user and the truster are considered.

5.3 Passive Testing Approach

In our contribution, we focus on using a passive testing approach in order to not disturb the natural function of the System Under Test (SUT). In this case, we will need an observation point that sniffs the communication for each O-grantor. As it is illustrated in Figure 5.1 an Observation Point (OP) will sniff the data exchanged between $\texttt{org}_\texttt{A}$ and the different entities in MOE.

Figure 5.1 gives an overview of our approach. During the communication, our solution will first collect the exchanged messages between the O-grantor and other entities in MOE. Next, the monitoring tool will check the conformity of the implementation with the specification. This specification is based on some trust properties. Then, the monitoring tool will provide verdicts related to these properties. These verdicts will be the inputs of the satisfactory evaluation that will provide finally a configuration file. These processes will be detailed on the rest of the chapter.

This passive testing approach will not inject or influence the function of the SUT. A monitoring tool is needed to design a passive approach of our system. In our case, we will not start from scratch, we will use an existing and famous tool in security testing, that is MMT. Moreover, we will update it to achieve our objectives.

5.4 Montimage Monitoring Tool

5.4.1 Tool Overview

MMT is a monitoring solution that allows combining: data capture; filtering and storage; events extraction and statistics collection; and, traffic analysis and reporting. It provides network, application, flow and user level visibility. An open source prototype variant particularly adapted for security monitoring (called MMT_Security) is used in the work presented here and is referred in the following as MMT.

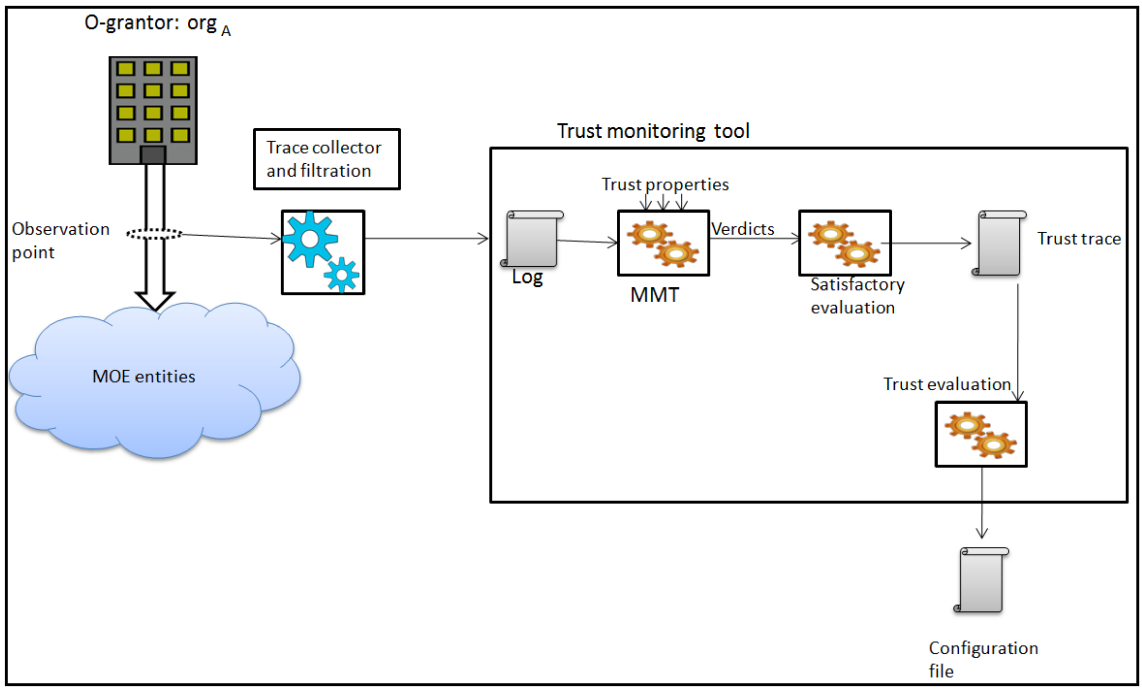


Figure 5.1: Overview of our approach.

MMT allows to verify application or protocol network traffic traces against a set of MMT_Security properties. MMT_Security properties can be either “Security rules" or “Attacks" as described by the following:

- A “Security rule" describes the expected functional or security behaviour of the application or protocol under-test. The non-respect of the MMT_Security property indicates an abnormal behaviour.
- An “Attack" describes a malicious behaviour whether it is an attack model, a vulnerability or a misbehaviour. Here, the respect of the MMT_Security property indicates the detection of an abnormal behaviour that might imply the occurrence of an attack.

5.4.2 Tool Architecture

MMT is composed of three complementary, but independent modules as shown in Figure 5.2:

- MMT_Extract is the core packet processing module. It is a C library that analyzes network traffic using Deep Packet/Flow Inspection (DPI/DFI) techniques in order to identify network and application based events by analyzing: protocols' fields values; network and application Quality of Service (QoS) parameters; and, Key Performance Indicators (KPI). In a similar way, it also allows analyzing any structured information generated by applications (e.g.,

traces, logged messages). MMT_Extract incorporates a plug-in architecture for the addition of new protocols or messages, and a public API for integration into third party probes.

- MMT_Security is a security analysis engine based on MMT_Security properties. It analyzes and correlates network and application events to detect operational and security incidents. For each occurrence of a security property, MMT_Security allows detecting whether it was respected or violated.

- MMT_Operator is a visualization application for MMT_Security currently under development. It allows collecting and aggregating security incidents to present them via a graphical user interface. MMT_Operator is conceived to be customizable, i.e., the user will be able to define new views or customize one from a large list of predefined views. With its generic connector, MMT_Operator can be integrated with third party traffic probes. At the time of writing this chapter, a web based representation of the analysis results is provided.

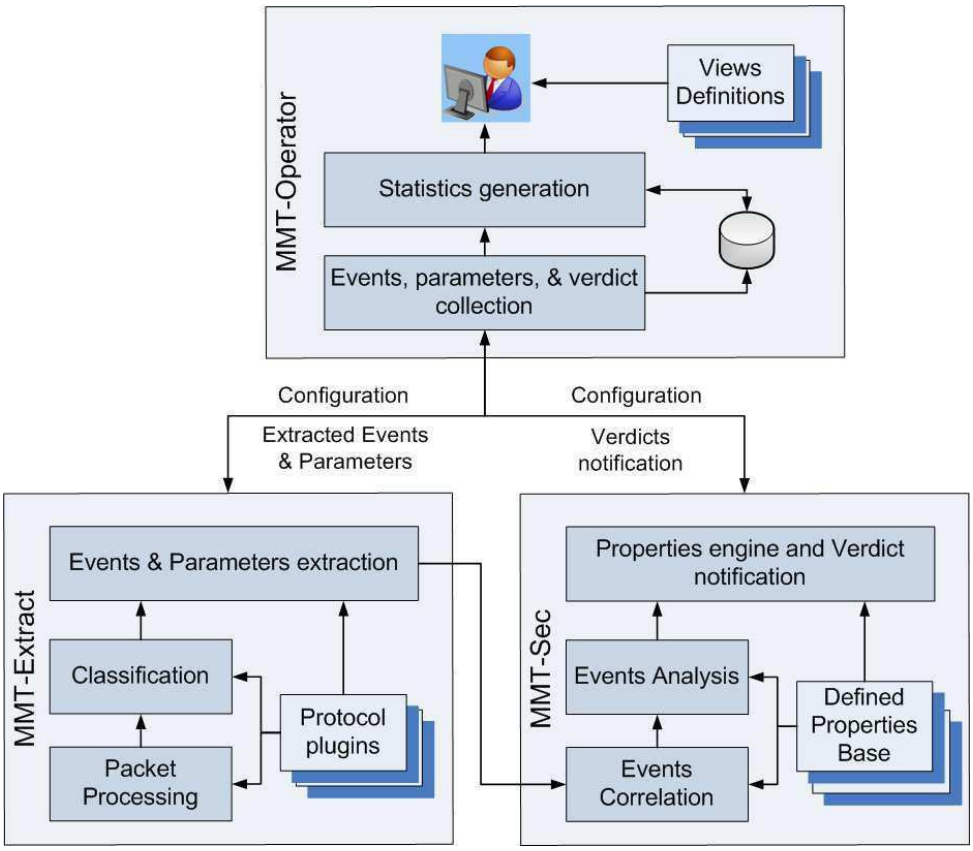


Figure 5.2: MMT Global Architecture.

5.5 Satisfactory Evaluation

The satisfactory evaluation method aims to assess each behavior b. This assessment associates a value between [-1,1] to this behavior, it will be denoted $\texttt{Sat(b)}$. If $\texttt{Sat(b)} \in [-1, 0]$ then this means that the previous behavior does not respect some requirements and it is considered as a bad one that have to decrease the experience evaluation of the user. Otherwise b is considered as a good behavior that will increase the experience evaluation of the user.

5.5.1 Satisfactory Evaluation Strategies

In MOE, we aim to propose some techniques that can be used to evaluate a behavior.

1. **Recommendation Strategy** Access control based on OrBAC [CM04, KD06, KLBB08, KDBK09] permits to write some recommendation rules with the following predicate:

   $$\texttt{Recomx}(\texttt{Any2Org}_\texttt{A}, \texttt{role}, \texttt{activity}, \texttt{view}, \texttt{context})$$

   This rule is defined as an authorization. It includes some desirable or advisable events. If they are not done, they abstract the responsibility of the user. Therefore, we propose to use this predicate in order to define a list of rules that will be named *recommendation policy*. For each situation some recommendation policies with new parameters (views, activities,etc) are defined and their verification is done during the execution of the behavior.

   Example 1 In an MOE scenario, we may have a situation Edit▷Source. If an employee will be permitted to perform this situation, he will be able to write code and description or copy several files of a project. A recommendation policy may contain some rules to check, for instance a) some files are edited before others, b) critical bugs are corrected before the normal one, or c) the time needed to execute some actions does not exceed some limits.

2. **Modification Assessment Strategy**. This strategy aims to evaluate the modification of the resource. It is done after the execution of the behavior, i.e, in the context of programming, we have to evaluate the syntax of the source code after each modification. This strategy cannot evaluate the behaviors that do not change any data as validation or reading. As a result, the administrator has to define some resource reference to be compared with the modified one.

   Example 2 For the same situation Edit▷Source, we suppose that any modification of a file will contain firstly the name of the author, the date and the title of the modification. An evaluation system have to check these information to evaluate a behavior related to this situation.

3. **Hybrid Strategy.** This strategy is based on the two previous solutions. with this strategy, we have three cases related to one situation. These are a) those that can be evaluated with recommendation and modification strategies, b) other that can be associated to only one method c) the third set could not be evaluated.

Figure 5.3 illustrates these previous strategies.

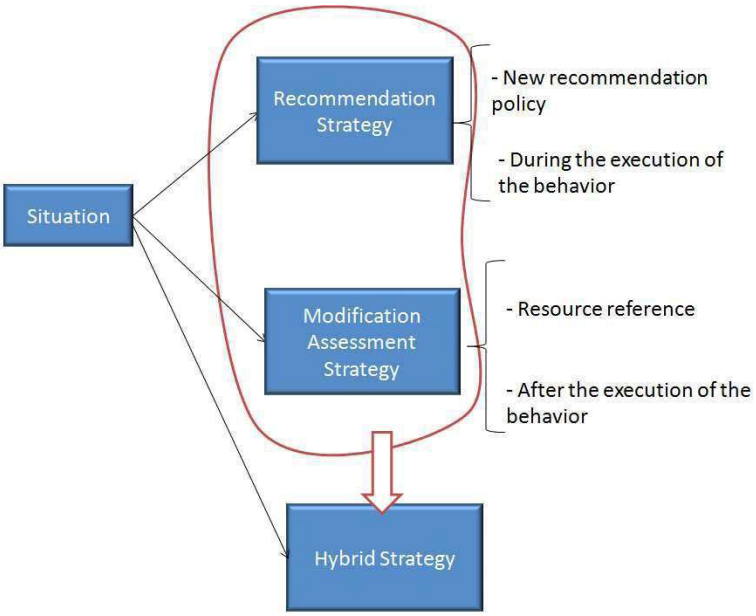


Figure 5.3: Satisfactory evaluation.

5.5.2 Evaluation of the Satisfactory Function

Regarding the definition of satisfactory function, in this section we will present how to apply the definition and how to evaluate it. We will introduce it using the example presented in Figure 5.4. In this Figure are depicted:

- The user that interacts with the system.
- The O-grantor that shares resources.
- A black line between them that represents different interactions of the user with the O-grantor.
- MMT Tool, the monitoring tool, that can record the previous black line in log files.
- A set of rules of different situations (S1,S2,...) used in MMT to assign a verdict about the correctness of the logs.

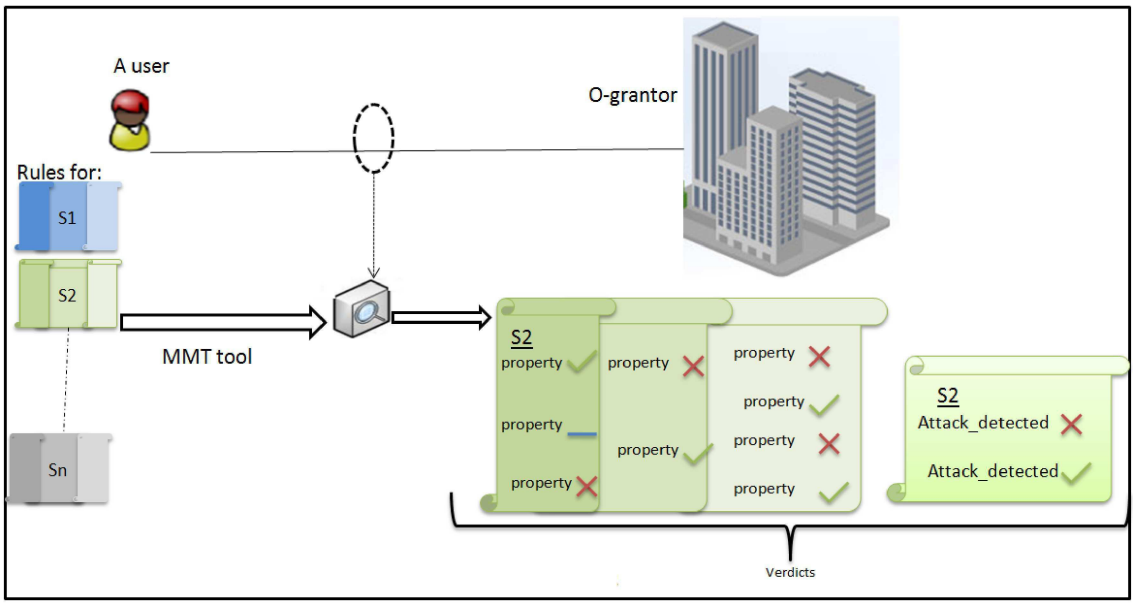


Figure 5.4: Evaluation of an interaction.

- On the right part the verdict of the logs with respect to the rules

As it is shown in Figure 5.4, for each *situation* we will have a list of *rules* that can be a security property to respect or an attack to detect. These *rules* will be written based on an extension of MMT language. Any *interaction* of a user will belong to one situation.

To evaluate it, we have:

1. To select the list of rules related to a fixed situation. In the running example, in Figure 5.4, we consider that the interaction of the user is related to S2.

2. To apply these rules as inputs of MMT tool. Then, it automatically compute which rules are respected, which rules are disrespected, and which are the different violations during the interaction.

In this proposal, **the influence of the different properties is not the same.** As it is shown in Figure 5.5, we provide three partitions of the different properties (high , medium and low) in order to differ between the list of properties. We will say that *in the case of a "security property", a high (resp. medium) security rule is more important than a medium (resp. low) rule.*

Based on the MMT tool and a new plug-in "Trust-plug", that analyzes the trace for trust proposals we are allowed to check that:

- If the rule is respected, then a value `+1` is assigned to it.

- If the rule is not respected, then a value `-1` is assigned to it.

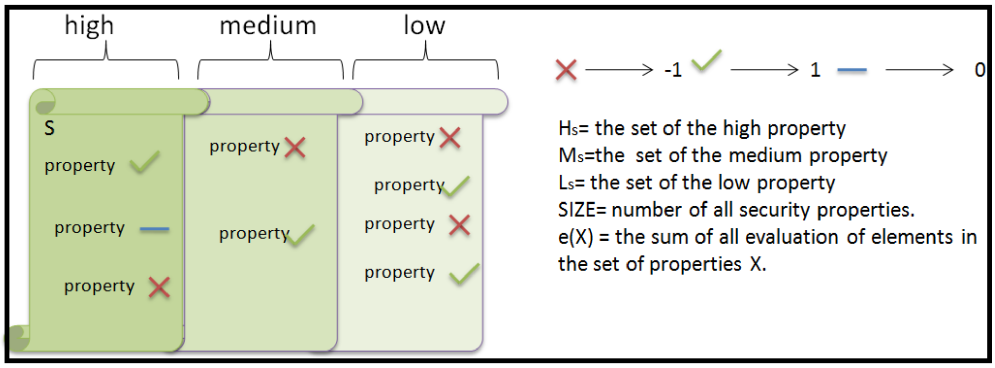


Figure 5.5: Properties partition.

- If we cannot have a decision about this rule during the interaction, then a value 0 is assigned to it.

Example 3 Let us focus on Figure 5.5. With respect to the previous definition, we have that the value assigned to each rule in this Figure are presented in the following table:

$+1$	-1	-1
0	$+1$	$+1$
-1		-1
		$+1$

In order to have a global valuation of the previous matrix, next we define the satisfactory function.

Evaluation1: We define the satisfactory function for a situation s as:

$$\begin{cases} -1 & if\ an\ attack\ is\ detected \\ \frac{e(H_s)+e(M_s)/2+e(L_s)/4}{SIZE} & otherwise \end{cases}$$

where H_s, M_s and L_s are the set of high, medium and low security properties defined for a situation s, $SIZE$ is the total number of rules for this situation, and e is a function that takes as inputs a set of properties and give as output the sum of their verdict.

Example 4 According to the previous function, we have that the satisfactory function of our running example is

$$\frac{(+1+0-1)+\frac{-1+1}{2}+\frac{-1+1-1+1}{4}}{9}=0$$

	All the rules are high security properties	All the rules are low security properties
all the properties are respected	1	0.25
all the properties are not respected	-1	-0.25

Figure 5.6: Satisfactory evaluation for high and low security properties.

In the table presented in Figure 5.6, we are showing how the satisfactory evaluation works when all the properties are respected (or not respected) and no attacks are detected. This comparison aims to highlight the differences among the different partitions. For the second row, we assume that all the rules during the interaction are respected. In the first case where all the properties are high, the satisfactory evaluation will be equal to 1. However, for a set of a low properties, it will be equal to 0.25. As a result, we remark that the respect of the high properties has more influence than the low properties.

Moreover, this partition permits to be more vigilant regarding the non respect of a high property. For example as it is shown in the third row in the table, if all the rules are not respected then the satisfactory evaluation in the first case (high security rules) will be equal to -1. However, it will be -0.25 for the second case.

5.6 Trust Traces

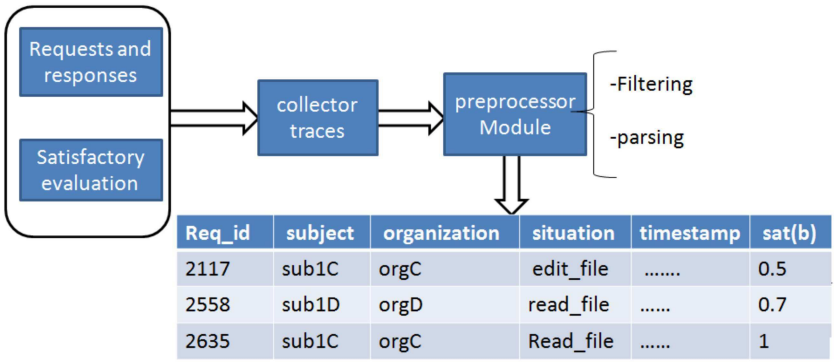
As it is described in Figure 5.7, this satisfactory evaluation will be combined with the related request and some basic information to define a trust trace that can be used to determine the experience evaluation of the user and the organization. Indeed, this trace will contain all the information that we need to evaluate the experience. In our model, a line in a trace is formed by the request and the subject identities (Req_id,subject), the organization, the related situation, the timestamp and the satisfactory evaluation of the behavior. The analysis of this trace based on the experience algorithm defined in the Chapter 3 will be used to determine the experience values.

5.7 Integration

In this section, we present the whole architecture of our proposal with the MMT tool (see Figure 5.8).

We define on the following the different modules that were not discussed in the previous sections.

Security policy: It contains different rules to specify *how to control the access of the shared resources*. In our proposal, these rules will be based on a trust policy to be activated.



Req_id	subject	organization	situation	timestamp	sat(b)
2117	sub1C	orgC	edit_file	.....	0.5
2558	sub1D	orgD	read_file	.....	0.7
2635	sub1C	orgC	Read_file	.....	1

Figure 5.7: Creation of trace file.

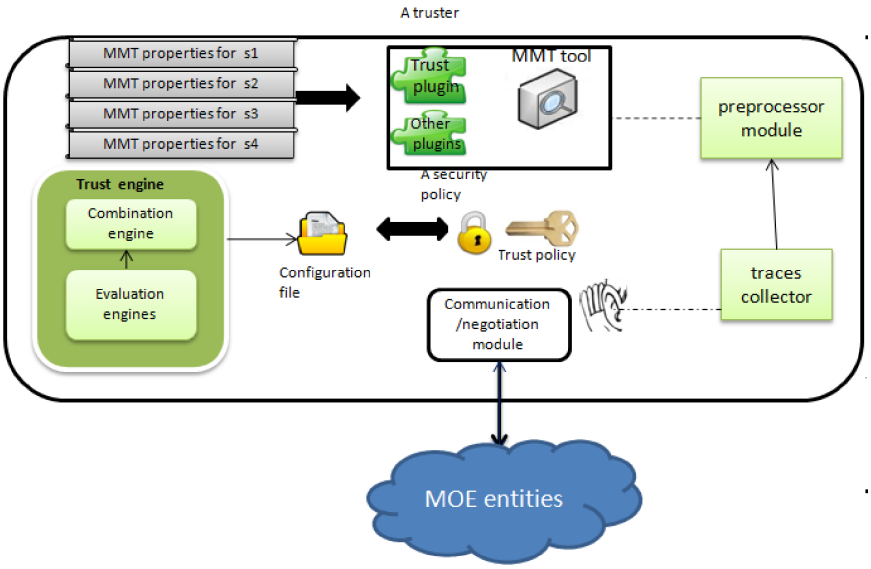


Figure 5.8: MMT integration in our trust framework.

Trust policy Each rule in the security policy will be only activated if the *trust level* of a user and\or its organization belong to trust threshold defined for this rule.

A combination engine It aims to define the different weight associated to the parameters of the trust vector. Thus, in this chapter we define two different trust policies Wo= (w_{eo},w_{ko}) and Wu= (w_{eu}, w_{ku}) for the organization trust vector (otv) and the user trust vector (utv) respectively. The two elements of each vector belong to the range [0,1] and their sum is equal to 1.

These values are specified by the administrator of the interoperability security policy. As a result, the evaluation of any trust vector will depends on these coefficients.

Trace collector: This module is responsible for two tasks, the first is to facilitate the integration of observation points. The second task is to extract the messages

exchanged which will be grouped in a single file named initial trace. This trace is then sent to the preprocessor module to be treated. This collector is offered by the MMT tool.

Preprocessor module: The initial trace will be handled by a preprocessor module that performs the following tasks:
(1) Filtering: this task is to filter the initial trace to extract only key messages needed for the test. This filtering can be based on the protocol type , the type of message, the destination address, etc. This task allows to improve the monitoring performance.
(2) Parsing: this task is to organize the trace in an XML file with well defined elements.

Trust-plug: It is a new plug-in developed for MMT in order to parse the trace and to extract the required parameters to evaluate the trust of an entity.

Given an interaction, the MMT tool automatically provides the detected attacks evaluation, the respected evaluation and the disrespected evaluation.

The output of this process is a trust trace (An example will be shown in the next section) that contains relevant information about any interaction. In particular it contains the identity of the request related to this interaction, the subject who needs the service, its organization, the related situation, the period, and the satisfactory evaluation. Let us note that, based on this trust trace, the experience evaluation of a user will be evaluated and saved in a configuration file.

Finally, a request will be accepted only if the user is permitted to perform the action into the needed view based on the security policy and only if a permission rule for this user related to this situation is activated by the trust policy.

5.8 Case Study

In this section we will present a case study, where we can show the usability of our approach.

5.8.1 Scenario

We will consider the following MOE scenario:

- Four organizations are participating in the same development of a project.
- The first organization $\texttt{org}_\texttt{A}$ has a server where several virtual machines are offered.
- The considered **activities** are: configure, modify, execute, test, and manage.
- The organization $\texttt{org}_\texttt{A}$ also offers the following **views**: source_code, application, testing_script, OS_System and resources.

- In this scenario four different **external** roles are defined: engineer, researcher, tester, and project manager.

5.8.2 Specification of the Interoperability Security Policy

The first phase is the specification and the deployment of an interoperability security policy. This policy is the result of a negotiation process between the O-grantor and the O-grantee. An example of how to do it is detailed in [CCBC06b]. We show on the following a part of the $\texttt{org}_{\texttt{A}}$ interoperability security policy.

- R1: An engineer is permitted to manage OS_System.
- R2: A researcher is prohibited to manage resources.
- R3: An engineer is permitted to modify the source_code
- R4: An engineer is permitted to execute an application.
- These rules are only applied for the **external** engineers and researchers that do not belong to $\texttt{org}_{\texttt{A}}$.

Related to this rule we have this trust policy:

- R1 is activated only if the trust evaluation of:
  - the user is more than 0.4 and
  - the organization is more than 0.
- R2 is activated only if the trust evaluation of:
  - the user is less than 0
    or
  - 1. the organization is between -0.3 and 0.2 and
    2. the user is between 0 and 0.4

    or
  - the organization is less than -0.3.
- R3 is activated only if the trust evaluation of:
  - the user is more than 0.5 and
  - the organization is more than 0.7.
- R4 is activated only if the trust evaluation of:
  - the user is more than 0.5.

5.8.3 Trust Properties Definition

The second step is to define a list of properties and threats that permits to evaluate the different interaction with $\texttt{org}_A$. For each situation, we have to write a list of properties.

Example 5 Figure 5.9 shows the trust properties for the situation manage▷OS_System.

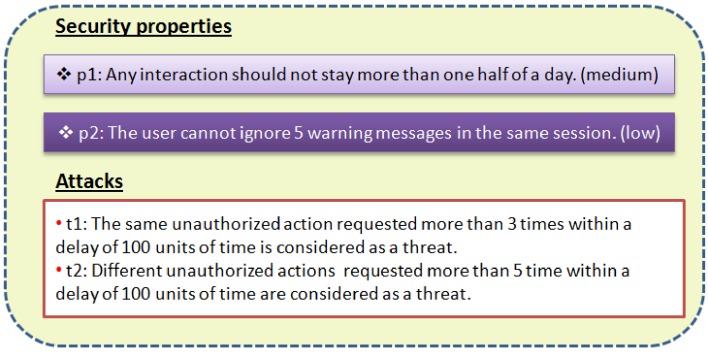


Figure 5.9: Trust properties for manage▷OS_System

The Figure 5.10 shows how to write the the property p1 in MMT language:

```
<property value="THEN" property_id="1"  delay_max="4"  type_property="SECURITY RULE" partition="medium"
          description=" An engineer should not stay more than one half of a day.">
        <event value="COMPUTE" event_id="1"
          description="First request received from an engineer to manage the OS."
          boolean_expression="((Trust.ROLE == 'engineer')&amp;&amp;(Trust.ROLE_ORG == 'VPO_Any_to_OrgA')
          &amp;&amp;(Trust.name == 'Begin')&amp;&amp;(Trust.ACTIVITY == 'manage')&amp;&amp;(Trust.VIEW == 'OS_System'))" />
        <event value="COMPUTE" event_id="2"
          description="the opened session is closed."
          boolean_expression="((Trust.session == Trust.session.1)&amp;&amp;(Trust.name == 'End'))"/>
</property>
```

The new parameter

the new plugin

Figure 5.10: A security property for the situation manage OS_System

We add first a new parameter 'partition' for the tag <property> in order to precise the importance of the rule. Moreover, as it is shown in Figure 5.10, a trust plug-in is developed that has to analyze the xml trace and to extract several elements as the **external** role of the user, its organization, the type of the request and etc.

Therefore, the satisfactory function of any interaction related to the situation manage▷OS_System will be:

$$\begin{cases} -1 & if\ t1\ or\ t2\ are\ detected \\ \frac{e(\varnothing)+\frac{e(\{p1\})}{2}+e(\{p2\})/4}{4} & otherwise \end{cases}$$

5.8.4 Executing MMT with the Previous Rules

After each period, MMT provides a result file as it is shown in the Figure 5.11. This file contains three tables:

Security rules summary results (period 4)

Id	Description	High	medium	low		
p1	SECURITY RULE: Any interaction should not stay more than one half of a day		medium		1	0
p2	SECURITY RULE:The user cannot ignore 5 warning messages in the same session		low		1	1

Attack summary results (period 4)

Id	Description		
t1	ATTACK: Unauthorized action requested more than 3 times within a delay of 100 units of time.	0	0
t2	ATTACK: Different unauthorized actions requested more than 5 time within a delay of 100 units of time .	0	0

Satisfactory evaluation (period 4)

Req_Id	user	Organization	Situation	TimeStamp	Sat
2117	a1	OrgA	manage OS_System	2013-06-04 16:34:12.000000	0.25

Figure 5.11: Result file from MMT.

- The first one cites the different properties, how many times that are respected or disrespected and the partition of the property.
- The second table is about the detected attacks.
- The last one provides a table that shows the interactions (request identity, the user, the organization, the situation and the timestamps) with its assigned satisfactory evaluation.

Based on these results, the trust level of the user and the organization $\texttt{org}_A$ will be updated in the configuration file. We show a part of this file in Figure 5.12.

Subject	Organization	Situation	Period	Trust evaluation
a1	orgA	Manage_OS_System	T4	0.5
-	orgA	Manage_OS_System	T4	0.1

Figure 5.12: A part of the configuration file during the period 5.

These results with the trust and the security policies will permit to response to any request. For example, a request that will be received during the period 5 to manage OS_System will be accepted since:

- A permission rule (R1) is provided for this user (see subsection 5.8.2).

- R1 is activated since the trust evaluation of the user is equal to 0.5 more than 0.4 and the trust evaluation of the organization is equal to 0.1 more than 0 (as shown in Figure 5.12).

This approach offers to an access control system to take into consideration the new interactions between the trustee and the truster. It permits to react by giving new permissions to unauthorized employee, to refuse an access for an authorized user in the next periods and to have a dynamic policy based on the analysis of the requester behaviors.

5.9 Conclusions and Future Work

In this chapter, we presented a methodology that permits to evaluate an interaction between a trustee and a truster. An extension of the monitoring tool MMT was proposed. Moreover, the basic function 'satisfactory evaluation' that permits to assess an interaction was well detailed. Finally, the different steps to implement our contribution with a case study were presented.

As future work, we are planning to use our approach in other distributed system as the VANET networks and e-Voting system for the European project INTER-TRUST.

Part IV

Conclusions and Perspectives

"I have come to the conclusion, after many years of sometimes sad experience, that you cannot come to any conclusion at all."
Vita Sackville-West, In Your Garden Again

Conclusions

In this thesis, we proposed a a novel and comprehensive framework in MOE environment. We have firstly done an interesting state of the art related to three security concepts, those are:

- Trust management frameworks: We studied the trust challenges and the existent approaches. This entitles us to understand the important concepts to define for the MOE environment.
- Security policy solutions: It was very important to do a large research related to security policy solutions in order to understand their components and to have a background about the security policy models. Then, we addressed the access control solutions used in collaborative systems in more depth. Most of the studied solutions are based on RBAC and `OrBAC`. This work helps us to choose `OrBAC` model as a security policy model in our framework. Then, we focus on its adaptation with our new trust approach.
- Security testing: A comparison between the active and passive testing techniques and a study of the testing tools was necessary to design an approach that can evaluate the behavior in MOE. Therefore, this approach will be used to evaluate the trust level of the user and the organization. Our solution is based on the MMT tool. This latter is a monitoring tool of capturing network traffic, analyzing and inspecting network packages to conclude about the network behavior.

Our framework is based on two new trust vectors for MOE; those are a user trust vector and an organization trust vector. In this framework we combined the possible parameters that can be used in MOE. Moreover, we proposed an evaluation of the different parameters and, above all, how to evaluate the impact of the trust organization level into the user rights. In this proposal, the situation concept is detailed and defined based on access control notions. Indeed, a trust relationship always depends on the action to be performed and the needed resource. We cannot have a global trust level. Different relationships between the trust parameters and rules are defined. Despite of the richness of concept definitions, the use of our solution is not complex since the administration can ignore some parameters based on its trust policy.

Different extensions are realized to improve our approach. The first one focuses on the integration of our trust approach to the OrBAC model. The second one proposes a trust ontology based on access control concepts in order to share feedback between organizations. The third one updates the MMT monitoring tool in order to evaluate the trust level of an entity. In this extension, we also proposed some evaluation strategies that can be used.

The proposed approach is generic. Its main advantage is that it can be easily extendable to any kind of applications: Cloud computing, VANET system, real-time networks, etc.

Perspectives

"In every end, there is also a beginning."
Libba Bray, A Great and Terrible Beauty

Currently, we are looking to extend our new framework to be used in other areas and to use the thesis results to improve the interoperability security between entities in other networks. Hereafter, we detail our planning and objectives to ameliorate our work. Moreover, we will also present two related works of external researchers that cited our work and proposed some perspectives:

Our perspectives

- At present, we are working in the INTER-TRUST project. As an objective, we aim to adapt our results to be used with the new INTER-TRUST framework. The INTER-TRUST project aims to develop a dynamic and scalable framework to support trustworthy services and applications in heterogeneous networks and devices, based on the enforcement of interoperable and changing security policies, addressing the needs of developers, integrators and operators. The project will be tested with two case studies: VANET network and e-voting system. The open issues related to this work are:
  - How to evaluate the vehicle behaviour and how to collect the trace in this scenario knowing that the mobility in this network is very high.
  - How to design the trust policy in this network? Is it needed to use all the defined parameters or new parameters need to be defined?
- Trust in cloud computing: In this thesis we have the opportunity to present several papers in international conferences. One of the key questions that has

been asked several times is why you do not use your trust approach in cloud computing. This scenario is very interesting and several trust solutions are designed for this architecture. We plan to study this architecture and to adapt our work based on its needs.

- Currently, we are implementing an administration tool for our approach. This tool will help the administrator (1) to define the trust policy, (2) to modify the forgetting factor (3) and to specify the trust classes. Figure 5.13 presents the current version of our tool.

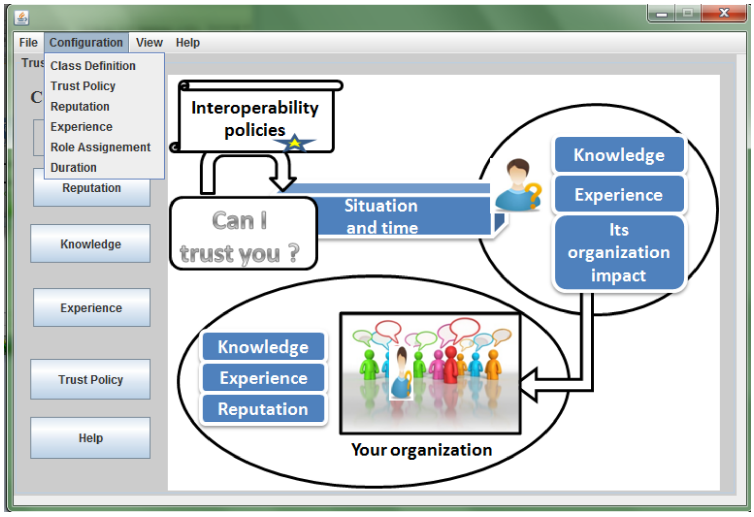


Figure 5.13: Current version of our administration tool.

Other perspectives

In 2013, we found two research teams [LMM13,FDFBJ13] that cited our work and try to improve and provide more functionalities related to trust management solutions.

- In [LMM13], the authors share the same idea about our trust vector definitions and evaluations and propose using a distributed network monitoring in order to provide trust information. A formal approach is designed to express trust properties and to evaluate them on real execution traces. For future works, the authors plan to apply their approach to other digital ecosystems to test trust properties targeted to diverse agents.

- In [FDFBJ13], authors cited our solution as a trust management technique. They propose to study the privacy preserving mechanism for trust management in more detail. Their method suggests to use some credentials based on the privacy preferences of the different parties. Moreover, the privacy of the two participants is protected by a protocol for secure multiparty computation.

Bibliography

[AAJ12] M. Arasteh, M. Amini, and R. Jalili. A trust and reputation-based access control model for virtual organizations. In *9th International ISC Conference on Information Security and Cryptology, ISCISC'12*, pages 121–127, 2012.

[AC04] F.T Alotaiby and J.X Chen. A model for team-based access control. In *Information Technology: Coding and Computing*, 2004.

[ACBC09] S. Ayed, N. Cuppens-Boulahia, and F. Cuppens. Deploying security policy in intra and inter workflow management systems. In *International Conference on Availability Reliability and Security, ARES'09*, pages 58–65. IEEE, 2009.

[ACCBCB08] F. Autrel, F. Cuppens, N. Cuppens-Boulahia, and C. Coma-Brebel. MotOrBAC 2: a security policy tool. In *Conference on Security of Network Architectures and Information Systems, SARSSI'08*, 2008.

[AMH+10] I. Sheikh Ahamed, M. H. Munirul, Md. E. Hoque, F. Rahman, and N. Talukder. Design, analysis, and deployment of omnipresent formal trust model (ftm) with trust bootstrapping for pervasive environments. *Journal of Systems and Software*, 83(2):253 – 270, 2010.

[ANS13] ANSSI. (2013):l anssi [online]. available. In *http://www.ssi.gouv.fr/fr/anssi/*, 2013.

[ARH97] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proceedings of the workshop on New security paradigms, NSPW'97*, pages 48–60. ACM, 1997.

[ARH00] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *33rd annual Hawaii International Conference on System Sciences, HICSS'00.*, volume 11, janaury 2000.

[ASL07] ASLab.org. (2007 may): A simplified guide to create an ontology [online]. available. In *http://tierra.aslab.upm.es/documents/controlled/ASLAB-R-2007-004.pdf*, 2007.

[BBF01] E. Bertino, P.A. Bonatti, and E. Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):191–233, 2001.

[BCDP05] E. Bertino, B. Catania, M.L. Damiani, and P. Perlasca. GEO-RBAC: a spatially aware rbac. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 29–37. ACM, 2005.

[BFK99] M. Blaze, J. Feigenbaum, and A.D. Keromytis. Keynote: Trust management for public-key infrastructures. In *Security Protocols*, pages 59–63. Springer, 1999.

[BFL96] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *The Symposium on Security and Privacy, S& P'96*. IEEE Computer Society, 1996.

[BHS02] F. Belanger, J.S. Hiller, and W.J. Smith. Trustworthiness in electronic commerce: the role of privacy, security, and site attributes. *The Journal of Strategic Information Systems*, 11(3):245–270, 2002.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. *International Workshop on Managing Requirements Knowledge*, 1979.

[BP05] L. Bin and U.W. Pooch. A lightweight authentication protocol for mobile ad hoc networks. In *International Conference on Information Technology: Coding and Computing,. ITCC'05.*, pages 546–551, 2005.

[Bri14] D. Brickley. (2014 january) rdfweb:the friend of a friend vocabulary. In *http://rdfweb.org/foaf/*, 2014.

[CC08] F. Cuppens and N. Cuppens. Modeling contextual security policies. *International Journal of Information Security (IJIS)*, 7(4):285 – 305, 2008.

[CCB08] F. Cuppens and N. Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security (IJIS)*, 7(4):285–305, august 2008.

[CCBC06a] F. Cuppens, N. Cuppens-Boulahia, and C. Coma. Motorbac : un outil d'administration et de simulation de politiques de sécurité. In *Int. conference on security in network architectures and security of Information systems(SAR-SS'06)*, 2006.

[CCBC06b] F. Cuppens, N. Cuppens-Boulahia, and C. Coma. O2O: Virtual private organizations to manage security policy interoperability. In *2nd Int. Conf. on Information Systems Security, ICISS'06, LNCS 4332*, pages 101–115. Springer, 2006.

[CCBCC08a] C. Coma, N. Cuppens-Boulahia, F. Cuppens, and A. Cavalli. Secure interoperation with O2O contracts. In *First workshop on security of spontaneous networks, SETOP'08*, 2008.

[CCBCC08b] C. Coma, N. Cuppens-Boulahia, F. Cuppens, and A.R. Cavalli. Context ontology for secure interoperability. In *International Conference on Availability, Reliability and Security, ARES'08*, pages 821 –827, 2008.

[CCBCC08c] C. Coma, N. Cuppens-Boulahia, F. Cuppens, and A.R. Cavalli. Interoperability using O2O contract. In *International Conference signal-image technology and Internet-based systems, SITIS'08*. IEEE Computer Society, 2008.

[CDdV+02] F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Implementing a reputation-aware gnutella servent. In *NETWORKING Workshops*, pages 321–334. Springer, 2002.

[CF98] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *International Conference on Multi Agent Systems, 1998*, pages 72–79. IEEE, 1998.

[CM04] F. Cuppens and A. Miège. AdOrBAC: an administration model for Or-BAC. *Comput. Syst. Sci. Eng.*, 19(3), 2004.

[CR06] S. Chakraborty and I. Ray. TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In *ACM Symposium on Access Control Models And Technologies, SACMAT'06*, pages 49 – 58. ACM, 2006.

[CTWS02] E. Cohen, R. Thomas, W. Winsborough, and D. Shands. Models for coalition-based access control (CBAC). In *Proceedings of the seventh ACM symposium on Access control models and technologies(SACMAT'02)*, pages 97–106. ACM, 2002.

[CWZG07] H. Chen, H. Wu, X. Zhou, and C. Gao. Agent-based trust model in wireless sensor networks. In *International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD'07*, volume 3, pages 119–124, 2007.

[CZC09] H. Chen, S. Zhao, and G. Chen. A reputation model for evaluating web services. In *International Conference on Communications and Networking. ChinaCOM'09*, pages 1–10, 2009.

[DDLS01] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder policy specification language. In *Policies for Distributed Systems and Networks*, volume 1995 of *Lecture Notes in Computer Science*, pages 18–38. Springer Berlin Heidelberg, 2001.

[DM08] N. Dokoohaki and M. Matskin. Effective design of trust ontologies for improvement in the structure of socio-semantic trust networks. *International Journal On Advances in Intelligent Systems*, 1(1942-2679):23–42, 2008.

[DMD+03] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Y. Halevy. Learning to match ontologies on the semantic web. *VLDB Journal*, 12(4):303–319, 2003.

[DP06] B. Dobing and J. Parsons. How UML is used. *Commun. ACM*, 49(5):109–113, 2006.

[EDU09] EDUCAUSE. (2009 march): Things you should know about P2P [online]. available. In *http://net.educause.edu/ir/library/pdf/est0901.pdf*, 2009.

[FCK95] D. Ferraiolo, J. Cugini, and D.R. Kuhn. Role-based access control RBAC: Features and motivations. In *Proceedings of 11th Annual Computer Security Application Conference*, 1995.

[FDFBJ13] O. Farràs, J. Domingo-Ferrer, and A. Blanco-Justicia. Privacy-preserving trust management mechanisms from private matching schemes. In *8th DPM International Workshop on Data Privacy Management*, 2013.

[FK09] D.F. Ferraiolo and D.R. Kuhn. Role-based access controls. *arXiv preprint arXiv:0903.2171*, 2009.

[FKC07] D F. Ferraiolo, D. Richard Kuhn, and R. Chandramouli. Role based access control. In *Second Edition, Artech House,2007*, 2007.

[FKT01] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, aug 2001.

[FN11] S. M. Falconer and N. F. Noy. Interactive techniques to support ontology matching. In *Schema Matching and Mapping*, pages 29–51. Springer, 2011.

[GMA+03] S. Godik, T. Moses, A. Anderson, B. Parducci, C. Adams, D. Flinn, G. Brose, H. Lockhart, K. Beznosov, M. Kudo, et al. extensible access control markup language (XACML) version 1.0, 2003.

[GMF+03] J.H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubezy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of protege: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

[Gol10] D. Gollmann. Computer security. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):544–554, 2010.

[GPH03] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. In *In Proceedings of Cooperative Intelligent Agents, LNCS*, pages 238–249. Springer, 2003.

[Gru93] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[GS00] T. Grandison and M. Sloman. A survey of trust in internet applications. *Communications Surveys Tutorials IEEE*, 3(4):2–16, 2000.

[GS03] T. Grandison and M. Sloman. Trust management tools for internet applications. In *Trust Management*, volume 2692 of *Lecture Notes in Computer Science*, pages 91–107. Springer Berlin Heidelberg, 2003.

[HCBCD09] D. Abi Haidar, N. Cuppens-Boulahia, F. Cuppens, and H. Debar. XeNA: an access negotiation framework using XACML. *Annals of telecommunications*, 64(1-2):155– 169, 2009.

[HCBD09] D. Abi Haidar, N. Cuppens-Boulahia, and F. Cuppensand H. Debar. Xena: an access negotiation framework using xacml. *annals of telecommunications - annales des telecommunications*, 64(1–2):155–169, 2009.

[HCJ10] W. T. Harwood, J. A. Clark, and J. L. Jacob. Networks of trust and distrust: Towards logical reputation systems, 2010.

[HkCHC13] C.F. Hsieh, k.f. Cheng, Y.F. Huang, and R.C. Chen. An intrusion detection system for ad hoc networks with multi-attacks based on a support vector machine and rough set theory. *Journal of Convergence Information Technology*, 8(2), 2013.

[HLHV10] A. Herzig, E. Lorini, J.F. Hubner, and L. Vercouter. A logic of trust and reputation. *Logic Journal of the IGPL, Normative Multiagent Systems*, 18(1):214–244, 2010.

[HM05] M.C. Huebscher and J.A. McCann. A learning model for trustworthiness of context-awareness services. In *International Conference on Pervasive Computing and Communications Workshops. PerCom'05*, pages 120–124, 2005.

[HNAN06] J.R. High, Nadalin, J. Anthony, and N. Nagaratnam. Role-permission model for security policy administration and enforcement, oct 2006.

[IM04] T. Ishaya and D. P. Mundy. Trust development and management in virtual communities. In *In the Second International Conference of Trust Management, iTrust04*, pages 266–276, 2004.

[JAS+11] W. Jie, J. Arshad, R. Sinnott, P. Townend, and Z. Lei. A review of grid authentication and authorization technologies and support for federated access control. *ACM Comput. Surv.*, 43(2):1–26, 2011.

[JDF08] Josep J. Domingo-Ferrer. A survey of inference control methods for privacy-preserving data mining. In *Privacy-preserving data mining*, pages 53–80. Springer, 2008.

[Jen02] C. D. Jensen. Secure environments for collaboration among ubiquitous roaming entities. In *First Internal iTrust Workshop on Trust Management in Dynamic Open Systems*, 2002.

[JGB07] Y. Jin, Z. Gu, and Z. Ban. Restraining false feedbacks in peer-to-peer reputation systems. In *International Conference on Semantic Computing, ICSC'07.*, pages 304–312, 2007.

[JI02] A. Jøsang and R. Ismail. The beta reputation system. In *Bled Electronic Commerce Conference*, 2002.

[JIB07] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[JL06] A. Jolly and S. Laskri. (2006): Grid computing [online]. available. In *http://www-igm.univ-mlv.fr/ dr/XPOSE2006/Jolly_ Laskri/historique.html*, 2006.

[JLK+05] H. Jameel, X. H. Le, U. Kalim, A. Sajjad, L. Sungyoung, and L. Young-Koo. A trust model for ubiquitous systems based on vectors of trust values. In *International Symposium on Multimedia*, 2005.

[JMW08] M.E. Johnson, D. Dan McGuire, and N.D. Willey. The evolution of the peer-to-peer file sharing industry and the security risks for users. In *41st Hawaii International Conference on Systems Science (HICSS'08)*, 2008.

[JPE01] M. Jayant, B. A. Philip, and R. Erhard. Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases,VLDB '01*, pages 49–58. Morgan Kaufmann Publishers Inc., 2001.

[JVR05] W. Janice, A. Vijayalakshmi, and M. Ravi. A credential-based approach for facilitating automatic resource sharing among ad-hoc dynamic coalitions. In *DBSec*, pages 252–266, 2005.

[Kam09] M. Kamel. Patrons organisationnels et techniques pour la sécurisation des organisationsvirtuelles. In *Ph.D. dissertation, University of Toulouse, September 2009*, 2009.

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

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. *Internet Request for Comment RFC 2104*, 1997.

[KD06] A.A. El Kalam and Y. Dewsarte. Multi-OrBAC: a new access control model for distributed, heterogeneous and collaborative systems. In *International Symposium on Systems and Information Security, SSI'06*. IEEE Computer Society, 2006.

[KDBK09] A.A EL Kalam, Y. Deswarte, A. Baina, and M. Kaaniche. PolyOrBAC: A security framework for critical infrastructures. *International Journal on Critical Infrastructure Protection*, 2(4):154–169, 2009.

[KI10] A.A. El Kalam and N. Idboufker. Specification and verification of security properties of e-contracts. In *International Conference on Communications, (COMM'10)*, pages 427–430. IEEE Computer Society, 2010.

[KKS13] C. Kościelny, M. Kurkowski, and M. Srebrny. Public key infrastructure. In *Modern Cryptography Primer*, pages 175–191. Springer, 2013.

[KLBB08] M. Kamel, R. Laborde, A. Benzekri, and F. Barrere. A best practices-oriented approach for establishing trust chains within virtual organisations. In *Enterprise Distributed Object Computing Conference Workshops, EDOCW'08*. IEEE Computer Society, 2008.

[KPR11] S. Kamara, C. Papamanthou, and T. Roeder. Cs2: A searchable cryptographic cloud storage system. *Microsoft Research TechReport MSR-TR-2011-58*, 2011.

[LHKP03] Jae-Nam Lee, Minh Q. Huynh, Ron Chi-Wai Kwok, and Shih-Ming Pi. It outsourcing evolution: past, present, and future. *Commun. ACM*, 46:84–89, 2003.

[LLYT05] K. Lin, H. Lu, T. Yu, and C. Tai. A reputation and trust management broker framework for web applications. In *International Conference on e-Technology, e-Commerce and e-Service, EEE'05*, pages 262–269, 2005.

[LMM13] J. Lopez, S. Maag, and G. Morales. A formal distributed network monitoring approach for enhancing trust management systems. In *In*

the Proc. of the 5th International ACM/IFIP Conference on Management of Emergent Digital EcoSystems, MEDES'13, 2013.

[LS00] J. Lipnack and J. Stamps. *Virtual Teams: People Working Across Boundaries with Technology, Second Edition*. John Wiley & Sons, Inc., New York, NY, USA, 2000.

[LV00] A. K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. In *Public Key Cryptography*, pages 446–465. Springer, 2000.

[LV10] C. Lin and V. Varadharajan. Mobiletrust: a trust enhanced security architecture for mobile agent systems. *International Journal of Information Security*, 9(3):153–178, 2010.

[MA03] L. Camarinha-Matos M and H. Afsarmanesh. Elements of a base ve infrastructure. *Computers in industry*, 51(2):139–163, 2003.

[MBCB08] W. Mallouli, F. Bessayah, A. Cavalli, and A. Benameur. Security rules specification and analysis based on passive testing. In *Global Telecommunications Conference, GLOBECOM'08.*, pages 1–6. IEEE, 2008.

[MC96] D. Harrison Mcknight and N. L. Chervany. The meanings of trust. Technical report, 1996.

[MCC+04] R. Morgan, S. Cantor, S. Carmody, W. Hoehn, and K. Klingenstein. Federated security: The shibboleth approach. *Educause Quarterly*, 27(4):12–17, 2004.

[MCHZ11] M. El Maarabani, A. Cavalli, I. Hwang, and F. Zaïdi. Verification of interoperability security policies by model checking. In *2011 IEEE 13th International Symposium on High-Assurance Systems Engineering, HASE'13*, pages 376–381. IEEE, 2011.

[MCJ+11] A. Mammar, A. Cavalli, W. Jimenez, W. Mallouli, and E. de Oca. Using testing techniques for vulnerability detection in c programs. In *Int. Conf. on Testing software and systems, ICTSS'11*, pages 80–96. Springer-Verlag, 2011.

[MDS95] R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *The Academy of Management Review*, 20(3):709–734, 1995.

[MFBT08] T. Mouelhi, F. Fleurey, Benoit, and Y. Le Traon. A model-based framework for security policy specification, deployment and testing. In *Model Driven Engineering Languages and Systems*, pages 537–552. Springer, 2008.

[MHA10] M. El Maarabani, H. Iksoon Hwang, and A.Cavalli. A formal approach for interoperability testing of security rules. In *International Conference on Signal-Image Technology and Internet-Based Systems (SITIS'10)*, pages 277–284. IEEE Computer Society, 2010.

[MKP08] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *IEEE International on Test Conference, ITC 2008*, pages 1–10. IEEE, 2008.

[MOC+07] W. Mallouli, J.M. Orset, A. Cavalli, N. Cuppens, and F. Cuppens. A formal approach for testing security rules. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 127–132. ACM, 2007.

[Mon14] Montimage. (2014 january):Qui sommes nous [online]. available. In *http://http://www.montimage.com/*, 2014.

[MVD12] T. Morris, R. Vaughn, and Y. Dandass. A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems. In *45th Hawaii International Conference on System Science, HICSS'12*, pages 2338–2345. IEEE, 2012.

[MWS04] D.J. Malan, M. Welsh, and M.D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, SECON'2004.*, pages 71–80. IEEE, 2004.

[NFF+91] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Magazine 12(3)*, pages 36–56, 1991.

[NLB+05] B. Nasser, R. Laborde, A. Benzekri, F. Barrere, and M. Kamel. Access control model for inter-organizational grid virtual organizations. In *On the Move to Meaningful Internet Systems (OTMWorkshops'05)*, volume 3762, pages 537–551. Springer Berlin Heidelberg, 2005.

[NM03] L. Ninghui and J.C. Mitchell. RT: a role-based trust-management framework. In *DARPA Information Survivability Conference and Exposition*, pages 201–212, 2003.

[No94] B. C. Neuman and T. Ts o. Kerberos: An authentication service for computer networks. *Communications Magazine IEEE*, 32(9):33–38, 1994.

[NWET04] P. Nixon, W. Wagealla, C. English, and S. Terzis. Privacy security and trust issues in smart environments, 2004.

[NZ09] M. Novotny and F. Zavoral. Matrix model of trust management in p2p networks. In *International Conference on Research Challenges in Information Science. RCIS'2009.*, 2009.

[oi07] Federation of identities. Gestion des identites. In *Technical paper, Team « Identity Management»*, 2007.

[Par14] Telecom Management Sud Paris. (2014 january) Télécom sudparis: le choix de l'excellence [online]. available. In *http://www.telecom-sudparis.eu/p_fr_ecole_presentation_8238.html*, 2014.

[PB10] S. Pearson and A. Benameur. Privacy, security and trust issues arising from cloud computing. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science, CloudCom'10*, pages 693–702. IEEE, 2010.

[PLHCETR06] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estévez-Tapiador, and A. Ribagorda. LMAP: A real lightweight mutual authentication protocol for low-cost rfid tags. In *Proc. of 2nd Workshop on RFID Security*, page 6, 2006.

[Pre93] B. Preneel. *Analysis and design of cryptographic hash functions*. PhD thesis, Katholieke Universiteit te Leuven, 1993.

[PS05] V. Patil and R.K. Shyamasundar. Trust management for e-transactions. *Sadhana*, 30(2-3):141–158, 2005.

[QHC07] D. Quercia, S. Hailes, and L. Capra. TRULLO - local trust bootstrapping for ubiquitous devices. In *International Conference on Mobile and Ubiquitous Systems: Networking&Services, MobiQuitous'07*, pages 1–9. IEEE Computer Society, 2007.

[RC04] I. Ray and S. Chakraborty. A vector model of trust for developing trustworthy systems. In *European Symposium on Research in Computer Security , ESORICS'04*. Springer, 2004.

[RMRH13] I. Ray, D. Mulamba, Indrakshi Ray, and K. J. Han. A model for trust-based access control and delegation in mobile clouds. In *Annual IFIP Conference on Data and Applications Security and Privacy, DBSec'13*, pages 242–257, 2013.

[RRC09] Indrakshi Ray, Indrajit Ray, and S. Chakraborty. An interoperable context sensitive model of trust. *Journal of Intelligent Information Systems*, 32(1):75–104, 2009.

[RZFK00] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12), 2000.

[Sar13] N. Sarrouh. Formal modeling of trust-based access control in dynamic coalitions. In *Computer Software and Applications Conference Workshops, COMPSACW'13*, pages 224–229, 2013.

[SCCC+96] Stuart Staniford-Chen, Steven Cheung, Richard Crawford, Mark Dilger, Jeremy Frank, James Hoagland, Karl Levitt, Christopher Wee, Raymond Yip, and Dan Zerkle. Grids-a graph based intrusion detection system for large networks. In *Proceedings of the 19th national information systems security conference*, volume 1, pages 361–370. Baltimore, 1996.

[SCW12] W. Susilo, Y. Chow, and R. Wiangsripanawan. Towards formalizing a reputation system for cheating detection in peer-to-peer-based massively multiplayer online games. In *international conferen on Network and System Security, NSS'12*, volume 7645 of *Lecture Notes in Computer Science*, pages 291–304. Springer Berlin Heidelberg, 2012.

[Sha87] S. P. Shapiro. The social control of impersonal trust. *American journal of sociology*, 93(3):623–658, 1987.

[Sko05] S.P. Skorobogatov. Semi-invasive attacks-a new approach to hardware security analysis. *Technical report, University of Cambridge, Computer Laboratory*, 2005.

[SLS98] T.J. Strader, F. Lin, and M.J. Shaw. Information infrastructure for electronic virtual organization management. *Decis. Support Syst.*, 23(1):75–94, may 1998.

[SM97] M. G. Stewart and R. E. Melchers. *Probabilistic risk assessment of engineering systems*. Chapman & Hall London, 1997.

[SZL08] Y.L. Sun, H. Zhu, and K.J.R. Liu. Defense of trust management vulnerabilities in distributed networks. *Communications Magazine, IEEE*, 46(2):112–119, 2008.

[TAC12a] K. Toumi, C. Andrés, and A. Cavalli. Setting trust evaluations with fuzzy logic in MOE. In *9th Workshop on System Testing And Validation, STV'12*, page in press, 2012.

[TAC12b] K. Toumi, Cesar Andres, and A. Cavalli. Trust-orbac: A trust access control model in multi-organization environments. In *International Conference on Information Systems Security, ICISS'12, LNCS*. Springer, 2012.

[TAC13a] K. Toumi, C. Andres, and A. Cavalli. Formal framework for defining trust in multi-organization environment. *Accepted in the International Journal of Autonomous and Adaptive Communications Systems, IJAACS'13.*, 2013.

[TAC13b] K. Toumi, C. Andres, and A. Cavalli. Security properties in virtual organizations. In *15th IEEE International Conference on High Performance Computing and Communication (HPCC'13)*. IEEE Computer Society Press, 2013.

[TAC13c] K. Toumi, C. Andres, and A. Cavalli. Trust ontology based on access control parameters in multi-organization environments. In *9th International Conference on SIGNAL IMAGE TECHNOLOGY and INTERNET BASED SYSTEMS (SITIS'13)*. IEEE Computer Society Press, 2013.

[TACM12] K. Toumi, Cesar Andres, A. Cavalli, and Mazen EL Maarabani. A vector based model approach for defining trust in multi-organization environments. In *International Conference on Risks and Security of Internet and Systems, CRISIS'12*. IEEE Computer Society Press, 2012.

[TCA13] K. Toumi, A. Cavalli, and C. Andres. Validation of a trust approach in multi-organization environments. *Submitted to the International Journal of Secure Software Engineering, IJSSE'13*, 2013.

[TCM12] K. Toumi, A. Cavalli, and M. El Maarabani. Role based interoperability security policies in collaborative systems. In *Int.Symposium on Security in Collaboration Technologies and Systems*. IEEE Computer Society, 2012.

[TD04] S. Toivonen and G. Denker. The impact of context on the trustworthiness of communication: An ontological approach. In *SWC Workshop on Trust, Security, and Reputation on the Semantic Web*, 2004.

[TPJL06] W.T.L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[TW12] M. H. Tehranipoor and C. Wang. *Introduction to Hardware Security and Trust*. Springer, 2012.

[UEN+09] P. Urien, S. Elrharbi, D. Nyamy, H. Chabanne, T. Icart, F. Lecocq, C. Pepin, K. Toumi, M. Bouet, G. Pujolle, P. Krzanik, and J-F. Susini. Hip-tags architecture implementation for the internet of things. In *Int. Conf. on Internet Asian Himalayas, AH-ICI 2009*, pages 1–5. IEEE Computer Society Press, 2009.

[Vio10] Bob Violino. (2010 may): It risk assessment frameworks: real-world experience [online]. available. In *http://www.csoonline.com/article/592525/it-risk-assessment-frameworks-real-world-experience*, 2010.

[wik13] wikipedia. Forgetting[online]. available http://en.wikipedia.org/wiki/forgetting. 2013.

[wik14] wikipedia. Classified information[online]. http://en.wikipedia.org/wiki/classified_information. 2014.

[Wil93] O. E. Williamson. Calculativeness, trust, and economic organization. *Journal of Law and Economics*, 1993.

[WL92] T.Y. Woo and S. Lam. Authentication for distributed systems. *Computer*, 25(1):39–52, 1992.

[WLWV09a] Y. Wang, K. J. Lin, D. Wong, and V. Varadharajan. Trust management towards service-oriented applications. *Service Oriented Computing and Applications*, 3(2):129–146, 2009.

[WLWV09b] Y. Wang, K.-J. Lin, D.S. Wong, and V. Varadharajan. Trust management towards service-oriented applications. *journal of Service Oriented Computing and Applications*, 3(2):129–146, 2009.

[WTS+02] M. Winslett, Y. Ting, K.E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and Y. Lina. Negotiating trust in the web. *Internet Computing, IEEE*, 6(6):30–37, 2002.

[XL03] Li Xiong and Ling Liu. A reputation-based trust model for peer-to-peer e-commerce communities. In *International Conference on E-Commerce, CEC'03.*, pages 275–284, 2003.

[YT05] E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. In *Proceedings. 2005 IEEE International Conference on Web Services, 2005. ICWS 2005.* IEEE, 2005.

[YV03] W. Yao and J. Vassileva. Bayesian network-based trust model. In *International Conference on Web Intelligence, WI'03.*, pages 372–378, 2003.

[ZC13] J. Zhang and R. Cohen. A framework for trust modeling in multiagent electronic marketplaces with buying advisors to consider varying seller behavior and the limiting of seller bids. *ACM Trans. Intell. Syst. Technol.*, 4(2):1–24, 2013.

[ZECC04] G. Zsidisin, L. Ellram, J. Carter, and J.L. Cavinato. An analysis of supply risk assessment techniques. *International Journal of Physical Distribution & Logistics Management*, 34(5):397–413, 2004.

[ZL09] H. Zhao and X. Li. Vectortrust: Trust vector aggregation scheme for trust management in peer-to-peer networks. In *18th Int. Conf. on Computer Communications and Networks, ICCCN'09*, pages 1–6. IEEE Computer Society, 2009.