THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Informatique*

Présentée et soutenue par :
Dana AL KUKHUN
Le Mardi 2 Octobre 2012

Title: *Steps towards adaptive situation and context-aware access: A contribution to the extension of access control mechanisms within Pervasive Information Systems*

Titre : *Vers un accès adaptatif sensible à la situation et au contexte : Une contribution à l'extension des mécanismes de contrôle d'accès aux Systèmes d'Information Pervasifs.*

JURY

Elisa BERTINO	Professeur, Purdue University	*Examinatrice*
Claude CHRISMENT	Professeur, Université Toulouse III	*Président du jury*
Bruno DEFUDE	Professeur, Télécom Sud Paris	*Rapporteur*
Thierry DELOT	HDR, Université de Valenciennes	*Rapporteur*
AbdelKader HAMEURLAIN	Professeur, Université Toulouse III	*Examinateur*
Elisabeth MURISASCO	Professeur, Université du Sud Toulon-Var	*Examinatrice*
Florence SÈDES	Professeur, Université Toulouse III	*Directrice de Thèse*

École doctorale : *École Doctorale Mathématique Informatique Télécommunications de Toulouse*
Unité de recherche : Institut de Recherche en Informatique de Toulouse – IRIT UMR 5505 CNRS
Équipe d'accueil : *Systèmes d'Informations Généralisés - Données et Documents Semi-Structurés*

Dana AL KUKHUN

Steps towards adaptive situation and context-aware access: A contribution to the extension of access control mechanisms within Pervasive Information Systems

Supervisor:

Florence Sèdes, Professor at Université Toulouse III - Paul Sabatier

Abstract

The evolution of pervasive computing has opened new horizons to classical information systems by integrating new technologies and services that enable seamless access to information sources at anytime, anyhow and anywhere. Meanwhile this evolution has opened new threats to information security and new challenges to access control modeling.

In order to meet these challenges, many research works went towards extending traditional access control models (especially the RBAC model) in order to add context awareness within the decision-making process. Meanwhile, tying access decisions to the dynamic contextual constraints of mobile users would not only add more complexity to decision-making but could also increase the possibilities of access denial. Knowing that accessibility is a key feature for pervasive systems and taking into account the importance of providing access within real-time situations, many research works have proposed applying flexible access control mechanisms with sometimes extreme solutions that depass security boundaries such as the Break-Glass option.

In this thesis, we introduce a moderate solution that stands between the rigidity of access control models and the riskful flexibility applied during real-time situations. Our contribution is twofold: on the design phase, we propose PS-RBAC – a Pervasive Situation-aware RBAC model that realizes adaptive permission assignments and alternative-based decision-making based on similarity when facing an important situation. On the implementation phase, we introduce PSQRS – a Pervasive Situation-aware Query Rewriting System architecture that confronts access denials by reformulating the user's XACML access request and proposing to him a list of alternative similar solutions that he can access. The objective is to provide a level of adaptive security that would meet the user needs while taking into consideration his role, contextual constraints (location, network, device, etc.) and his situation.

Our proposal has been validated in three application domains that are rich in pervasive contexts and real-time scenarios: (i) Mobile Geriatric Teams, (ii) Avionic Systems and (iii) Video Surveillance Systems.

Keywords: Access Control, Pervasive Information Systems, Context-awareness, Situation-awareness, Adaptation, RBAC, XACML.

Institut de Recherche en Informatique de Toulouse – UMR 5505 CNRS

Université Toulouse 3 – Paul Sabatier, 118 route de Narbonne, F-31062 Toulouse cedex 9

Dana AL KUKHUN

Vers un accès adaptatif sensible à la situation et au contexte : Une contribution à l'extension des mécanismes de contrôle d'accès aux Systèmes d'Information Pervasifs

Directrice de thèse :

Florence Sèdes, Professeur à l'Université Toulouse III - Paul Sabatier

Résumé

L'évolution des systèmes pervasives a ouvert de nouveaux horizons aux systèmes d'information classiques qui ont intégré des nouvelles technologies et des services qui assurent la transparence d'accès aux resources d'information à n'importe quand, n'importe où et n'importe comment. En même temps, cette évolution a relevé des nouveaux défis à la sécurité de données et à la modélisation du contrôle d'accès.

Afin de confronter ces challenges, differents travaux de recherche se sont dirigés vers l'extension des modèles de contrôles d'accès (en particulier le modèle RBAC) afin de prendre en compte la sensibilité au contexte dans le processus de prise de décision. Mais la liaison d'une décision d'accès aux contraintes contextuelles dynamiques d'un utilisateur mobile va non seulement ajouter plus de complexité au processus de prise de décision mais pourra aussi augmenter les possibilités de refus d'accès. Sachant que l'accessibilité est un élément clé dans les systèmes pervasifs et prenant en compte l'importance d'assurer l'accéssibilité en situations du temps réel, nombreux travaux de recherche ont proposé d'appliquer des mécanismes flexibles de contrôle d'accès avec des solutions parfois extrêmes qui depassent les frontières de sécurité telleque l'option de "Bris-de-Glace".

Dans cette thèse, nous introduisons une solution modérée qui se positionne entre la rigidité des modèles de contrôle d'accès et la flexibilité qui expose des risques appliquées pendant des situations du temps réel. Notre contribution comprend deux volets : au niveau de conception, nous proposons PS-RBAC - un modèle RBAC sensible au contexte et à la situation. Le modèle réalise des attributions des permissions adaptatives et de solution de rechange à base de prise de décision basée sur la similarité face à une situation importanteÀ la phase d'exécution, nous introduisons PSQRS - un système de réécriture des requêtes sensible au contexte et à la situation et qui confronte les refus d'accès en reformulant la requête XACML de l'utilisateur et en lui proposant une liste des resources alternatives similaires qu'il peut accéder. L'objectif est de fournir un niveau de sécurité adaptative qui répond aux besoins de l'utilisateur tout en prenant en compte son rôle, ses contraintes contextuelles (localisation, réseau, dispositif, etc.) et sa situation.

Notre proposition a été validé dans trois domaines d'application qui sont riches des contextes pervasifs et des scénarii du temps réel: (i) les Équipes Mobiles Gériatriques, (ii) les systèmes avioniques et (iii) les systèmes de vidéo surveillance.

Mots Clés: Contrôle d'Accès, Systèmes d'Information Pervasifs, Contexte, Situation, Adaptation.

Université Toulouse 3 – Paul Sabatier, 118 route de Narbonne, F-31062 Toulouse cedex 9

To the soul of my grandmother

To mom & dad

To all the angels who where behind this work

Pour l'âme de ma grand-mère

À ma mère et mon père

À tous les anges qui étaient derrière ce travail

Acknowledgments

This thesis was the fruit of a constant effort made through many years and also of many beneficial exchanges and collaborations. This work would not have been accomplished without the valuable and helpful people who shared the same passion for scientific research. It is with great pleasure that I thank today all those who supported me during these years of hard work and enabled me to succeed in this thesis.

Sincere thanks to Pr. Claude Chrisment - Université Paul Sabatier - Toulouse III, who welcomed me in the SIG team and honoured me in chairing this committee.

I would like to thank Pr. Bruno Defude from Télécom Sud Paris, and Dr. Thierry Delot from Université de Valenciennes, for accepting to be the reporters of this thesis, for the time taken to review my dissertation and for their valuable comments that helped me to improve the quality of this document.

I would like to thank Pr. Elisa Bertino for welcoming me in her department at Purdue University for 3 months, for her interest in my research works, for the critical scientific vision that she has always offered, for her enriching remarks and for honouring me by participating in this jury and examining my thesis.

I would also like to thank Pr. Elisabeth Murisasco - Université du Sud Toulon-Var and Pr. AbdelKader Hameurlain - Université Paul Sabatier - Toulouse III, for their interest in my research works and for the honour they have accorded to me while participating in this committee and examining my dissertation.

I would like to express my deepest gratitude to my supervisor Pr. Florence Sedes, Université Paul Sabatier - Toulouse III, for her continuous encouragement and her valuable advices that helped me to grow and improve the quality of my works during these years. I thank her for her availability, patience and for motivating me and guiding me towards the exploration of the interesting world of research.

A big thanks goes to Dr. Marie-Françoise Canut and Dr André Peninou from IUT de Blagnac for their collaboration and support with a big heart, the great interest and attention that they have accorded to my works, their availability and patience till the last minute. Their reviews, constructive comments and our discussions have helped me to improve the content of the manuscript.

I also thank Dr. Beatrice Vincent and Dr. Marie-Annick Montalan from Laboratoire de Gestion et Cognition (LGC) at Université de Toulouse III, for their cooperation, for being scientifically open and for giving me the opportunity to accompany them while visiting the Mobiles Geriatric Teams.

Furthermore, I would like to thank the French Embassy of Jordan and Mr. Yann PRADEAU, the Counsel of Cultural Action and Cooperation for giving me the desire and the opportunity to pursue my graduate studies in France.

My thanks also go to the industrial projects that have participated in financing a part of my thesis: the GEODESIE project - AirBus and the LINDO project.

My thanks go as well to all members of the SIG department at IRIT (Information Technology Institute of Toulouse) for their help and support. Thank you to the staff of the laboratory for their kindness and for their help.

I also extend my thanks to all PhD students who are present (Ana Maria Manzat, Dana Codreanu, Dieudonné Tchuente, Laure Soulier, Eya Znaidi, Faten Atigui and Damien Dudognon) with whom I have shared good times at the office, the lab, coffee breaks, university restaurant, ...

I do not forget to thank the doctors who were collegues and former PhD students (Bouchra Soukkarieh, Lobna Hlaoua, Ilhème Ghalamallah, Corine Zayani, Mihaela Brut, Sébastian Laborie, Guillaume Cabanac, Karim Djemal and Desire Kompaore).

A thank you also goes to Justin Guthrie, the talented intern from The Robert Gordon University who helped me in achieving the implementation part.

To Bouchra Soukkarieh and Sireen Najdi thank you for all the unforgettable moments we spent together. A big thanks to the friends I met here in Toulouse (Manal, Wissam, Zaher, …) and to all my friends in Jordan and other countries for their encouragement and endless support despite the distance that separated us.

Finally, I would like to express my deepest gratitude to my parents who never stopped to support me. Great thanks to my sisters Reem and Mays and my brothers Ali and Ala' who never stopped believing in me during all these years of study and who have always encouraged me to achieve my ambitions. To all my family members (uncles, antes and cousins) who were always close to me despite the distance between us.

Remerciements

Cette thèse est le fruit des années d'effort incessants, mais aussi d'échanges bénéfiques et de collaborations fructueuses. Ce travail n'aurait pas pu aboutir sans le concours précieux et généreux de personnes qui partagent la même passion pour la recherche scientifique. C'est avec un énorme plaisir que je remercie aujourd'hui toutes les personnes qui m'ont soutenue durant ces années de travail pour faire réussir cette thèse.

J'adresse mes sincères remerciements à Monsieur Claude Chrisment, Professeur à l'Université Paul Sabatier –Toulouse III, qui m'a accueilli au sein de l'équipe SIG et qui me fait l'honneur de présider ce jury.

Je tiens à remercier Monsieur Bruno Defude, professeur à Télécom Sud Paris et Monsieur Thierry Delot, maître de conferences HDR à l'Université de Valenciennes, d'avoir accepté d'être rapporteurs de ce mémoire. Leurs lectures précises de ce manuscrit et leurs judicieuses remarques m'ont permis d'améliorer la qualité du présent document.

Je tiens à remercier Madame Elisa Bertino, professeur à Purdue University pour m'avoir accueilli dans son département pendant 3 mois, pour l'intérêt et le regard critique qu'elle a toujours portés à mes recherches, pour ses remarques enrichissantes et pour l'honneur qu'elle m'a fait en participant à ce jury et examinant ce mémoire.

Je tiens également à remercier Madame Elisabeth Murisasco, Professeur à l'Université du Sud Toulon-Var et Monsieur AbdelKader Hameurlain, Professeur à l'Université Paul Sabatier – Toulouse III, pour l'intérêt qu'ils ont porté à mes travaux et et pour l'honneur qu'ils m'ont fait en participant à ce jury et en examinant ce mémoire.

Je tiens à exprimer mes plus vifs remerciements à ma directrice de thèse Madame Florence Sèdes, Professeur à l'Université Paul Sabatier – Toulouse III, pour ses encouragements et ses conseils précieux qui m'ont permis de progresser et d'améliorer la qualité de mes travaux durant ces années de thèse. Je la remercie pour sa disponibilité, sa patience et pour m'avoir guidée vers l'exploration des pistes intéressantes dans le monde de recherche.

Un grand merci à Madame Marie-Françoise Canut, maître de conférences à l'IUT Blagnac et à Monsieur André Péninou, maître de conférences à l'IUT Blagnac, pour leur collaboration et soutien avec un grand cœur, le grand intérêt qu'ils ont porté à mon travail, leur disponibilité et patience jusqu'à la dernière minute. Leur lecture, remarques constructives et nos discussions m'ont permis d'améliorer le contenu du manuscrit.

Je tiens à remercier également Madame Béatrice Vincent et à Madame Marie-Annick Montalan, maîtres de conférences au Laboratoire Gestion et Cognition (LGC) à l'Université Toulouse III, pour leur coopération et ouverture scientifique et pour m'avoir donné l'opportunité de les accompagner aux visites des Equipes Mobiles Gériatriques.

Par ailleurs, je voudrais remercier l'ambassade de France en Jordanie et Monsieur Yann PRADEAU, le conseilleur de Coopération et d'Action Culturelle pour m'avoir donné l'envie et la chance de poursuivre mes études supérieures en France.

Mes remerciements vont aussi aux projets industriels qui ont participé à financer une partie de ma thèse : le projet GEODESIE - AirBus et le projet LINDO.

Mes remerciements vont de même à tous les membres de l'équipe SIG de l'IRIT pour leur aide et leur gentillesse. Merci aux personnels du laboratoire pour leur gentillesse ainsi que pour leur aide.

J'adresse également mes remerciements à tous les thésards qui sont présents (Ana Maria Manzat, Dana Codreanu, Dieudonné Tchuente, Laure Soulier, Eya Znaidi, Faten Atigui et Damien Dudognon) avec qui j'ai partagé de bons moments au bureau, à la salle machine, aux pauses café, au RU, …

Je n'oublie pas non plus de remercier les docteurs qui ont été des anciens thésards (Bouchra Soukkarieh, Lobna Hlaoua, Ilhème Ghalamallah, Corine Zayani, Mihaela Brut, Guillaume Cabanac, Karim Djemal et Désiré Kompaore).

Un merci va également au stagiaire doué Justin Guthrie qui m'a aidé dans la réalisation de la partie implémentation.

A Bouchra Soukkarieh et Sirène Najdi merci pour tous les moments inoubliables qu'on a passé ensemble. Un grand merci va aux amis que j'ai rencontré ici à Toulouse (Manal, Wissam, Zaher,) et à mes amis en Jordanie et aux autres pays pour leur encouragement et leur fidélité malgré la distance qui nous sépare.

Je remercie du fond du cœur mes parents qui n'ont pas cessé de me soutenir. Merci à mes sœurs Mays et Reem et mes frères Ali et Ala' qui n'ont jamais cessé de croire en moi pendant toutes mes années d'études et qui m'ont toujours encouragée à atteindre mes ambitions. A tous les membres de ma famille (mes oncles, antes et cousin(e)s) qui étaient toujours proches de moi malgré la distance qui nous séparait.

Table of Contents

List of Figures

List of Tables

General Introduction

With the increasing development in telecommunication systems, networking capabilities, hardware and software, traditional information systems have evolved towards ensuring more transparence, interoperability and better information sharing between the different subcomponents of an information system, which usually belong to heterogeneous levels of confidentiality.

The technological evolution and the integration of new technologies in daily life applications have enabled better connectivity and enhanced accessibility to information sources. This development has enabled the user to freely interact and access different information sources at “anytime, anywhere and anyhow” and that’s where the system becomes pervasive or ubiquitous [Park et al., 2004]. The notion of ubiquity was initially introduced by Weiser [Weiser, 1991] who predicted the future of information systems in the 21st century where computing elements would disappear while functioning homogeneously in total transparence.

Transparency is a strongly required quality that can ensure better access to information sources at all the levels of a system. However, applying this quality without constraints may make resources vulnerable to threats and to security attacks. That’s what forms a scientific and technological barrier standing in the way of the evolution of these systems.

Analyzing the accessibility challenge within Pervasive Information Systems (PIS), we find that attaining a balance between data discretion and data transparency is highly needed especially when considering access requests to information sources that are located within multi-distributive environments and managed by different authorities. The evolution of PIS has introduced a new challenge to the management of data access especially for mobile users. These systems should, at once, allow users to achieve high availability and also protect the system by applying rigid access control policies that can guarantee it’s safety and protect it from being vulnerable to intruder attacks.

The challenges of access management within traditional information systems have been resolved through the proposal of several access control models such as DAC, MAC and RBAC. But with the technological evolution, new challenges have been introduced such as the complexity of gouverning access within dynamic contexts.

As a result, several research studies were conducted to expand the RBAC model as a basis for context-aware access control models, but the proposed models did not take into account the importance of usability, situation-awarenesss and improving access opportunities. The unability to respond to a user’s access request usually ends in a refusal, which could be critical in important situations especially that in a pervasive context the reason behind an access denial might often be related to the dynamicity of the user’s context or to the existence of conflicting access policies.

The management of distributed data has always been a challenging problem to resolve due to the heterogeneity of machines, operating systems, data models and the languages used for applications. However, the web and its standards have changed the situation; where it headed towards providing a communications protocol between machines, called "web services" which consists of a service-oriented architecture that

enables easy transformation and distribution of data sources that became available everywhere.

In the age of service-oriented architectures and taking into consideration the multi-distributive nature of the access control policies managing information sources within pervasive environments, efforts to access control went towards decision-making from multiple distributed access policies, managed by different services and sometimes generated in real-time. The quality assurance of a service in this context is ensured by guaranteeing interoperable security and effective decision-making, it is for this reason that different XML-based security standards [XML, 1998] were proposed such as XML-Signature [XML-Signature, 2002], XML Encryption [XML Encryption, 2002] and XACML [OASIS, 2003, 2005].

Through our research, we were interested in the XACML standard due to its ability to produce centralized access decisions based on decentralized access policies expressed in XML. The power of XACML is its ability to take into account the contextual constraints from a pervasive environment.

Considering the advances of access control modeling and management, a challenge remains present in resolving the paradox of “usability vs. safety” for access control management during critical situations by implementing access mechanisms that can balance between these two conflicting goals and avoid the usage of extreme situations such as the “Break-Glass” solution.

It is within this motive that we have modeled and implemented PS-RBAC (Pervasive Situation-aware RBAC Model), an access control model that extends the RBAC model by adding to it context and situation awareness. The proposed model is designed to perform adaptive decisions that react to access denials by proposing a list of alternative authorized resources, that are "similar" to those requested by the user.

The PS-RBAC model aims to increase access opportunities during critical situations without depassing the security limits. This was achieved through PSQRS (Pervasive Situation-aware Query Rewriting System), which reacts in the case of an access denial by rewriting the launched XACML access requests in order to adapt and find alternative accessible resources that fit with the contextual constraints of the user.

This model was implemented and validated in different application domains and in different contexts (medical, aerospace, video surveillance).

The thesis is organized as follows: The first part consists of a state of the art that deals with three main aspects: (i) the conventional management of access control, (ii) the development of access modeling to meet with the needs of PIS and (iii) the orientation of access control towards service-oriented architectures.

In the second part, we detail our contribution where we present PS-RBAC - a context and situation-aware access control model that is dedicated to meet the needs of PIS users and the PSQRS architecture that employs an XACML query rewriting method to perform this adaptation.

The third section describes different validation scenarios that demonstrate the appliquability of our contribution within diverse application domains and includes an implemented example.

Finally, we conclude by stating the main contributions and presenting the short- and long-term perspectives of our work.

Part I:

Background: The Evolution of Access Control Modelling

Chapter 1:
Classical Access Control Management

1.1 Introduction

Providing adequate security to information sources and information systems is a fundamental management responsibility. Nearly all applications that deal with security, privacy, safety, or defense, include some form of access control.

As organizations are moving their structures towards decentralization, flexibility and dynamicity, their information system architectures are becoming, more and more, distributed and dynamic. Thus, a critical need arises to ensure that subjects are only allowed to access authorized resources.

In this chapter, we introduce the basic concepts of access control and we follow the evolution of access control management starting from models offering simple administration to classical information systems reaching to models designed to meet the needs of managing data access and privilege distribution within distributed systems.

1.2 Access Control

1.2.1 Definition

Access control is the traditional center of gravity of computer security. It is where security engineering meets computer science. Its main function is to control which principals (persons, processes, machines, . . .) have access to which resources in the system. For example: which files they can read, which programs they can execute, how they share data with other principals, and so on [Anderson, 2001].

Access control is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system. In some systems, complete access is granted after successful authentication of the user, but most systems require more sophisticated and complex control.

Authentication is any process by which the system verifies that someone is the one who he claims to be. This usually involves a ***username*** and a ***password***, but can include any other method of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints.

In addition to the authentication mechanism, access control is concerned with how authorizations are structured. ***Authorization*** is finding out if the person, once authenticated, is permitted to get access to the resource. It is realized by evaluating requests against applicable policies ascertaining that requested actions can be granted.

In some cases, authorizations may mirror the structure of the organization, while in other cases it may be based on the sensitivity level of various documents and the clearance level of the user accessing those documents.

1.2.2 Basic Concepts – Access Life Cycle

The life cycle of access control starts by an access demand (***request***) issued by a ***subject*** who asks the system to perform a certain ***action*** on a certain ***object***, see Fig. 1.1. According to the user's characteristics and context, the system will study the authorizations granted to this entity and will reply to his request with binary ***responses***: a "**Yes**" would allow the subject to perform the desired action on the object or a "**No**" which would revoke this demand. In order to clearly understand this process, we provide next, the definitions of the main entities that exist in an authorization system.

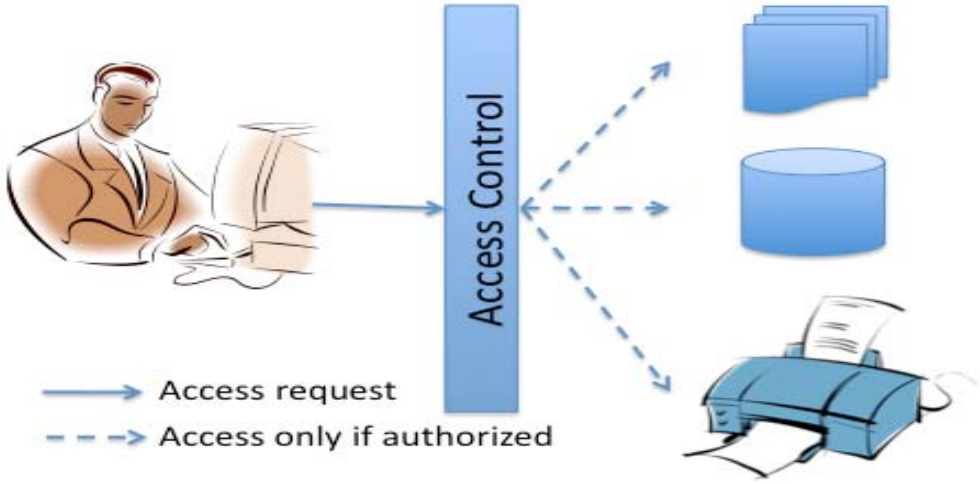


Figure 1.1: Access Control life cycle

- ***Subject***: An active entity, generally in the form of a person, process, or device that causes information to flow among objects or can perform changes to the system state [NCSC, 1988].
- ***Object***: An entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are records, fields (in a database record), blocks, pages, segments, files, directories, directory trees, process, and programs, as well as processors, video displays, keyboards, clocks, printers, and network nodes. Devices such as electrical switches, disc drives, relays, and mechanical components connected to a computer system may also be included in the category of objects [NCSC, 1988].
- ***Action/ Operation***: An active process invoked by a subject on objects; for example, different actions can be performed on a certain record (read, write, execute, delete, etc).
- ***Permission (privilege)***: An authorization assigned to a subject to perform some action on the system. In most computer security literature, the term permission refers to some combination of object and operation. A particular operation used on two different objects represents two distinct permissions, and similarly, two different operations applied to a single object represent two distinct permissions [Ferraiolo et al., 1992].

1.2.3 Applying Access Control

Dealing with confidential information sources within a decentralized system requires access control mechanisms that can protect these sources from any threat. In order to guarantee full protection, accessibility should be controlled through all the communication channels starting from the application level, the middleware level, the operating system level and finally through the network, the characteristics of adequate access control is described in each of these levels [Anderson, 2001]:

- The access control mechanisms, which the user sees at the ***application level***, may express a very rich and complex security policy. This can be realized by using authentication and authorization controls appropriately robust for the risk of the application, monitoring access rights to ensure they are the minimum required for the user's current business needs, using time-of-day limitations on access as appropriate, logging access and security events, and finally by using a software that enables rapid analysis of user activities.
- The applications may be written on top of ***middleware***, such as a database management system or book-keeping package, which enforces a number of protection properties. For example, a book-keeping software may need to make sure that a transaction that debits one ledger for a certain amount must credit another ledger for the same amount.
- The middleware will use facilities provided by the underlying ***operating system***. As this constructs resources such as files and communications ports from lower-level components, it acquires the responsibility for providing ways to control access to them.
  Realizing efficient access control should be done through: (i) securing access to system utilities, (ii) restricting and monitoring privileged access, (iii) logging and monitoring user or program access to sensitive resources and alerting on security events, (iv) updating the operating systems with security patches and (v) securing the devices that can access the operating system through physical and logical means.
- Access control should be highly ensured at the ***network level***: it works through multiple layers of access controls in order to provide protection against unauthorized access. Institutions should group network servers, applications, data, and users into security domains (e.g., untrusted external networks, external service providers, or various internal user systems), establish appropriate access requirements within and between each security domain, implement appropriate technological controls to meet those access requirements consistently, and monitor cross-domain access for security policy violations and anomalous activity.

Finally, the operating system access controls will usually rely on hardware features provided by the processor or by associated memory management hardware. These controls which memory addresses a given process can access.

1.2.4 Managing Access Control

As we will show in this section, the implementation of access control is based on three main concepts: access control policies, models, and mechanisms [De Capitani di Vimercati et al., 2007]. We demonstrate in Fig. 1.2 the relation between these concepts.

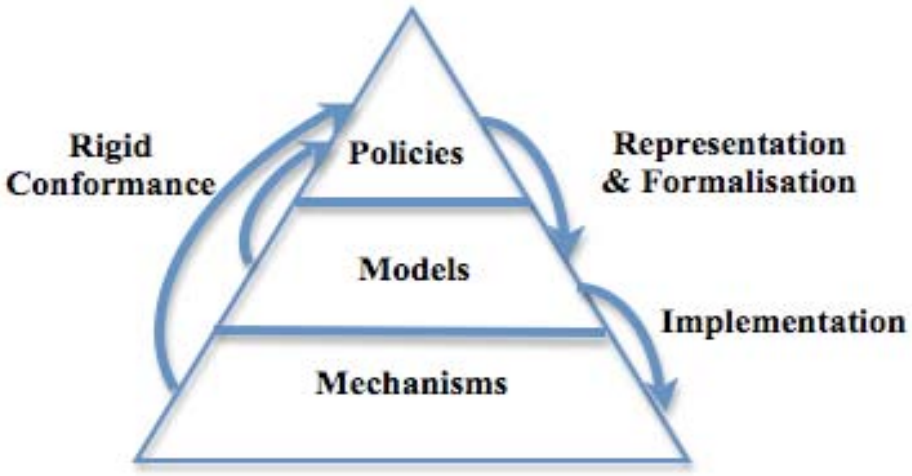


Figure 1.2: Important Concepts for Implementing Access Control

- ***Access control policies*** define the rules according to which access control must be regulated. In general, access control policies are dynamic in nature as they have to reflect evolving business factors, government regulations, and environmental conditions.
  Policies are high-level requirements that specify how access is managed and who may access information under what circumstances. For instance, policies may pertain to resource usage within or across organizational units or may be based on need-to-know, competence, authority, obligation, or conflict-of-interest factors.
  Chapter 3 presents XACML – an access control policy language and a policy example is provided in Fig. 3.3.
- ***Access control models*** provide formal representation of access control security policies. The formalization allows the proof of the security properties that are provided by the designed access control system.
  Being formal presentations of the security policy enforced by the system, security models are useful for proving theoretical limitations of a system. Section 1.3 presents the basic access control models (DAC, MAC and RBAC).
- ***Access control mechanisms*** usually come at the low abstraction level where they enforce these high-level access control policies and translate a user's access request in terms of a specific structure that the system provides.

We highlight the importance of access control models due to their ability to bridge the abstraction gap between access policies and mechanisms. Rather than attempting to evaluate and analyze access control systems exclusively at the

mechanism level, access control models are usually written to describe the security properties of an access control system.

1.3 Basic Access Control Models

The complexity of managing access to information sources has increased due to the constant evolution of computer information systems. In order to guarantee the quality of providing secure accessibility to information system ressources, several access control models were proposed.

1.3.1 Discretionary Access Control (DAC)

In Discretionary Access Control (DAC) models, access rights restrict access on protected objects based on the identity of subjects or, in order to improve scalability, to groups they belong to.

1.3.1.1 Basic concepts of DAC

According to the TCSEC[1] (Trusted Computer System Evaluation Criteria), a DAC Policy is: « A means of restricting access to objects based on the identity of the subjects or groups, or both, to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject ».

The DAC model is based on an access matrix, which was proposed by [Lampson, 1974] and formalized by [Harrison et al., 1976]. It declares for each combination of a subject and an object the set of allowed actions. The access rights can be stored in a matrix with the columns representing objects, rows representing subjects, and the entries being the granted privileges. Such policies are typically employed for operating systems.

In other words, a DAC policy is simply based on the definition of a set of rules, called authorizations, explicitly stating which user can perform which action on which resource. These rules can be represented as triples of the form (s, o, a) stating that subjects s can execute action a on object o. When a user makes an access request, the policy is enforced on the basis of the identity of the requester and on the rules involving itself.

The DAC model leaves a certain amount of access control to the discretion of the object's owner or anyone else who is authorized to control the object's access [NCSC, 1987]. For example, it is generally used to limit a user's access to a file [NIST, 1994]; it is the owner of the file who controls other users' accesses to that file. Only those users specified by the owner may have some combination of read, write, execute, and other permissions to the file. DAC policy tends to be very flexible and is widely used in the commercial and government sectors.

1.3.1.2 drawbacks of DAC

In DAC models, permissions are assigned to subjects directly. The disadvantage of such an approach is that, in a very large system, the granting of permissions to

[1] DoD, Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD

operate on individual data items to individual users is very time consuming and difficult to manage. It is also difficult to remember which permissions should be revoked from users when they leave the company or change jobs [Osborn, 2007].

Also, DAC is known to be inherently weak for two reasons [NIST, 2006]:

1. Granting read access is transitive so as the ressource owner grants access to a certain user, the latter (granted user) could access the ressource and allow other users to read it without referring to the owner.
2. A DAC policy is vulnerable to Trojan horse attacks because programs inherit the identity of the invoking user.

Thus, formally, the main drawbacks of DAC are as follows:

- Authorized information can be copied from one object to another; therefore, there is no real assurance on the flow of information in a system.
- No restrictions apply to the usage of information when the user has received it.
- The privileges for accessing objects are decided by the owner of the object, rather than through a system-wide policy that reflects the organization's security requirements.

1.3.2 Mandatory Access Control (MAC)

To solve the drawbacks of the DAC model, the Mandatory Access Control (MAC) model have distinguished between users and subjects. Mandatory polices were initially introduced in the operating system context, where objects to be protected are essentially files containing the data. Later studies such as [Jajodia et al., 1991] investigated the extension of mandatory policies to the database context.

1.3.2.1 Basic concepts of MAC

Accoding to TCSEC, the MAC model is defined as « A mean of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to information of such sensitivity ».

The MAC model can offer a highly secure level of administration to information sources. In MAC, access control decisions are applied by a central authority, not by the individual owner of an object (resource), and the owner cannot change access rights.

An example of MAC occurs in military security, where an individual data owner does not decide who has a Top Secret clearance, nor can the owner change the classification of an object from Top Secret to Secret [Pfleeger, 1997]. MAC is the most mentioned NDAC Policy [NIST, 2006].

The need for MAC arises when the security policy of a system dictates that:

1. Protection decisions must not be decided by the objects owners.
2. The system must enforce the protection decisions (i.e., the system enforces the security policy over the wishes or intentions of the object owner).

Usually a ***labeling mechanism*** and a set of interfaces are used to determine access based on the MAC policy. Considering a system managing access to sources according to their security level and labeling them in three incremental security classes: (confidential, secret, top secret), a user who is running a process at the ***Secret*** classification level should not be allowed to read a file with a label of ***Top Secret***. This is known as the "simple security rule," or "no read up."

Conversely, a user who is running a process with a label of ***Secret*** should not be allowed to write to a file with a label of ***Confidential.*** This rule is called the "*-property" (pronounced "star property") or "no write down." The *-property is required to maintain system security in an automated environment. A variation on this rule called the "strict *-property" requires that information can be written at, but not above, the subject's clearance level.

Multilevel security models such as the Bell-La Padula Confidentiality [Bell et al., 1973] and Biba Integrity models [Biba, 1977] are used to formally specify this kind of MAC policy. However, information can pass through a covert channel in MAC, where information of a higher security class is deduced by inference such as assembling and intelligently combining information of a lower security class.

1.3.2.2 Drawbacks of MAC

Although the mandatory policy protects data better than the discretionary policy, it has some problems [De Capitani di Vimercati et al., 2007]:

- The main problem is that MAC models are designed for applications where the keeping of secrets and the control of information flow are the primary requirements. Thus, MAC controls only flows of information in the system that happen through ***overt channels***, that is, channels operating in a legitimate way. But MAC is, at the other hand, vulnerable with respect to ***covert channels***, which are channels not intended for normal communication but can still be exploited to infer information.
- Another drawback of MAC is that subjects and objects have to be classified and this may not always be feasible (a lattice-based set of labels have be applied on all objects and subjects and that constraints concerning reading and writing of objects must be satisfied [Sandhu, 1993]). Moreover, access is evaluated only on the basis of this classification, consequently the system may be too rigid.

1.3.3 Role Based Access Control (RBAC)

The principal motivation behind the proposal of the RBAC (Role Based Access Control) systems [Ferraiolo et al., 1992] was to provide a model and tools to manage access control in a complex environment with a very large number of users and an even larger number of data items.

RBAC has emerged as a viable alternative to traditional DAC and MAC access control policies because it is based on an enterprise's organizational structure. As such, systems, data, and applications administrators and owners can more effectively manage and maintain information resources in a manner that is consistent with enterprise-wide security policies. RBAC has the further benefit of facilitating systems

administration by assigning roles to manage users as opposed to using each individual user's identity to manage users.

The role-based access control (RBAC) is an access control model that ties the concepts of users, roles, sessions, and permissions. The power of RBAC relies in it's central privilege management which offers greater administrative efficiency as well as the ability to intuitively administer and enforce a wide range of access control policies.

1.3.3.1 Basic Concepts of Core RBAC

The figure 1.3 illustrates the core RBAC model and its components. The basic elements are summarized as follows [Ferraiolo et al., 1992]:

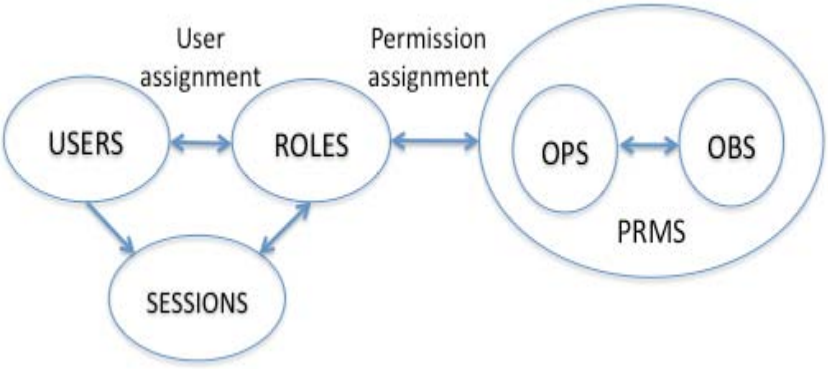


Figure 1.3: The Core RBAC Model

- ***Users (USERS)*** are human beings. It is important to know that it may be extended to represent processes, or any active entity of the system acting behalf of the user.
- ***Roles (ROLES)*** are job functions within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.
- ***Objects (OBS)*** are data or resources to be accessed.
- ***Operations (OPS)*** are processes that execute some function on behalf of the user.
- ***Permissions (PRMS)*** are defined as an approval of a particular mode of access to one or more objects in the system. Permissions establish a relation between operations and objects. That is, they define the operations to be performed over objects. Thus, the permissions are expressed as the following:

$$\mathbf{PRMS = 2^{OPS \times OBS}}$$

That is, $2^{OPS \times OBS}$ denotes the powerset of the set of ***OPS × OBS***

OPS × OBS denotes the cartesian product of operations and objects.

The core RBAC model defines relations between the following objects:

- ***User Assignment (UA)*** defines the relations among users and roles. Note that a user may be assigned to more than one role, and several users can be assigned to the same role. It is a many-to-many mapping, where:

$$\mathbf{UA \quad USERS \times ROLES}$$

- ***Permission Assignment (PA)*** defines the relation between roles and permissions. Here, a role may be assigned to multiple permissions. Similar to UA, it defines a many-to-many mapping:

PA PRMS × ROLES

- In addition, RBAC defines a mapping between users and roles as a set of ***sessions (SESSIONS)***. The idea is that when a user logs into the system, the user may be mapped to one or several roles through a session. This allows a user to activate only the minimum subset of roles they need in order to perform a specific task, and hence by using sessions, RBAC supports the least privilege principle.

The mapping is established by two functions:

- ***Session_User (SU)*** defines the one-to-many mapping relation between the user and the number of sessions that can be assigned to him.

SU SESSIONS × USERS

- ***Session_Role (SR)*** defines the many-to-many mapping relation between the number of assigned sessions and the roles connected to them.

SR SESSIONS × ROLES

The core RBAC model is characterized by applying 2 main concepts:

1. ***Least Privilege*** is an administrative action of avoiding the assignment of permissions to users that are unnecessary to accomplish a task. Least privilege means that once access requirements are determined, that role should only be given permissions to accomplish the required tasks; no additional permissions should be given. This prevents users from being given no more permissions than needed.

2. ***Separation of Duties*** The concept applies constraints on roles coexisting in a single session and that might have conflicting permissions, thus the model reduces the number of potential permissions available to the user. The procedure is used to prevent frauds.

1.3.3.2 Evolution of RBAC Models

In order to meet the changing needs and requirements of access management not only at the system level but also at the network level, the initial RBAC model has been subject to several evolutions aiming to enhance administrative and security benefits.

As a result, the spectrum of RBAC definitions includes models that range from the simple to sophisticated analysis of various definitions of RBAC [Sandhu et al., 1997], [Sandhu et al.,1998]. They define the basic RBAC model, referred to as $RBAC_0$, as including least privileges and separation of duties.

Subsequent RBAC models were introduced and built on this basic model, introducing new concepts of hierarchies and constraints, see Table 1.1 [NIST, 2002].

The evolution started by applying various principles such as roles hierarchy in $RBAC_1$ where a user can inherit the access rights of another user by the inclusion of constraints in roles assignment. Next, in $RBAC_2$, the dynamic separation of duties was

introduced to solve cases where a single user should perform a mission while having various roles that might contradict. Finally, $RBAC_3$ covered all the previous concepts.

Model	Hierarchies	Canstraints
$RBAC_0$	No	No
$RBAC_1$	Yes	No
$RBAC_2$	No	Yes
$RBAC_3$	Yes	Yes

Table 1.1: Evolution of RBAC Models

$RBAC_1$: $RBAC_1$ has introduced the concept of role hierarchies, see Figure 1.3. Role hierarchies are a natural extension of the authority and responsibility roles that exist within an organization [Sandhu, 1996].

The role hierarchy (roles containing other roles) extends the role layer to multiple layers, reducing the number of relationships that are managed. It helps to manage role complexity through structure to exploit commonality not only between users but among roles. Role hierarchies allow a policy implementer to write generic access rules just once, rather than for every role to which the rule applies.

When reflected at the network level, this approach can increase the administrative efficiency of the network. Rather than re-specifying all of the permissions of the junior role for the senior role, the junior role is specified as a permission of the senior role. As the organization levels or the number of permissions increase, the greater the system benefits from establishing role hierarchies.

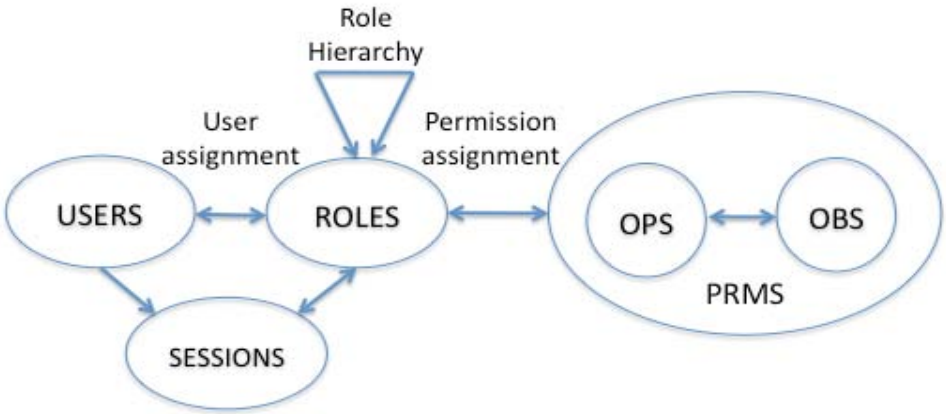


Figure 1.3: $RBAC_1$ - The Hierarchical RBAC Model

$RBAC_2$: $RBAC_2$ is also based on the original $RBAC_0$ model but introduces the concept of constraints. The most frequent use of constraints is to achieve separation of duties within an organization. For example, a constraint can state that if a user has a particular role, that user cannot be assigned a separate role, see Fig. 1.4.

However, constraints can also be used in many other situations. Constraints can be used to establish membership to a particular role. If an organization wants to have only one department head, then it can impose a particular constraint stating that if someone is in a particular role, then no one else can be admitted to that role. This concept has been referred to as cardinality. Constraints can also be used as prerequisites for entry into roles. For example, the only way that role x can be assigned to a user is if the user is already in role y.

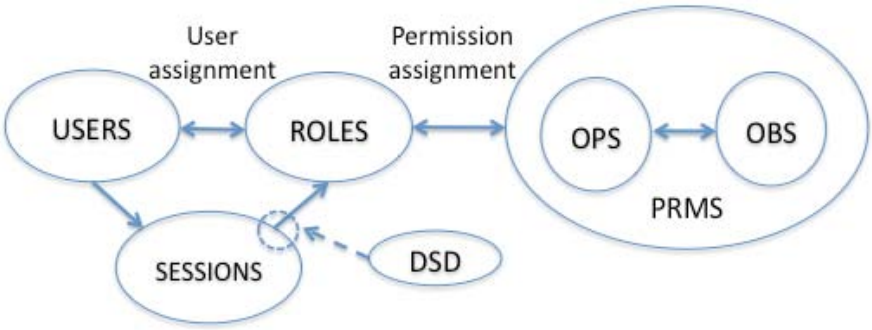


Figure 1.4: The RBAC2 Model with Dynamic Separation of Duties

RBAC$_3$: The NIST RBAC model is RBAC$_3$. It is the most complex RBAC model, including both role hierarchies and constraints. In RBAC$_3$, constraints can be imposed on the hierarchical roles within an organization [Kuhn et al., 1997]. For example, junior roles can be constrained to have a maximum number of senior roles, multiple junior roles can be constrained to have different senior roles, or constraints can be imposed on users to limit the number of senior roles to which they can be assigned. The sensitive interactions that occur between role hierarchies and constraints in RBAC$_3$ make it the most sophisticated and complex RBAC model, see fig. 1.5.

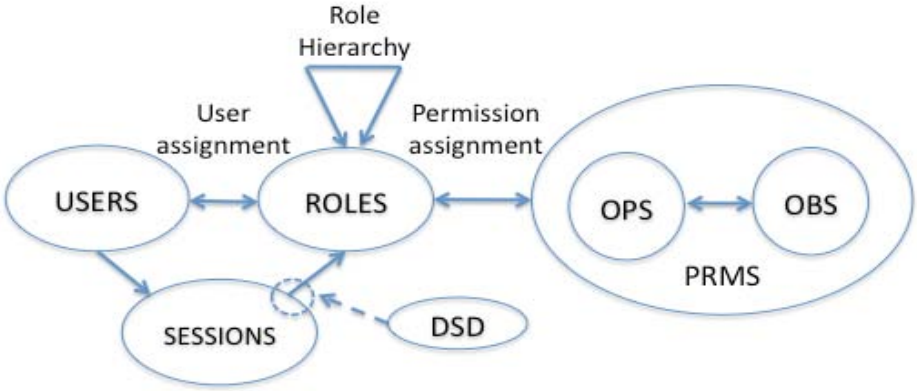


Figure 1.5: The RBAC3 Model with Role Hierarchies & Dynamic Separation of Duties

1.4 Conclusion

In this chapter, we have followed the evolution of the access control modeling needs and presented the main access control models introduced to meet the challenges of access management in modern information systems, which are the DAC, MAC and RBAC models.

At the end of this chapter, we state the main reasons for the widespread employment and adoption of the RBAC model, which succeeded in employing the

organizational roles to efficiently manage access permissions within modern distributed information systems. As discussed, RBAC exists in many forms, but even its simplest form is considered as an improvement over other alternative methods.

"RBAC features such as policy neutrality, principle of least privilege, and ease of management make [RBAC models] especially suitable candidates…Such models can express both DAC and MAC policies, as well as user-specific policies. In essence, RBAC models can provide a generic framework for expressing diverse security requirements" [Joshi et al., 2001].

RBAC provides the capability to visualize and manage user privileges across heterogonous platforms and applications. By centrally storing and managing roles as both collections of users and collections of privileges. RBAC is able to define, constrain, review, and enforce access control policies as user/role, role/role, or role/privilege relations.

RBAC is considered to be policy-neutral in the sense that, by using role hierarchies and constraints, a wide range of security policies can be expressed to include traditional DAC as well as a variety of nondiscretionary separation of duty (SOD) policies through the definition of constraints.

Throughout this thesis, we will be interested in applying adaptive accessibility within pervasive environments through the RBAC model. The main benefits associated with using RBAC rather than any another model are:

- RBAC provides simplified systems administration and privilege management.
- RBAC enhances systems security and integrity (improved security and audit trails).
- RBAC policies can express both DAC and MAC policies, as well as user-specific policies.
- RBAC is a generic model that can be adopted within any application domain and any environment.

Chapter 2:

Modelling Access Control for Pervasive Information Systems

2.1 Introduction

Due to the revolution of Information Technology, a new computing era is taking place. Many challenges need to be met, especially in a mobile and dynamic environment where users are interacting with different devices and constructing adhoc networks, while systems are supposed to provide them with proactive value added services.

This chapter aims at presenting a general idea about the evolution of pervasive computing environments, their caracteristics and the challenges that face applying access control to these evolutionary systems. The objective is to make a survey about the research works that tried to meet the main quality requirements: allowing a transparent user accessibility and interaction in a secure manner.

The Chapter is organized as follows: we start by presenting several definitions and features of pervasive information systems. Next, we'll explore the objectives of pervasive computing, the sub components of a pervasive environment showing the main challenges and the importance of adaptation to face them in order to achieve a homogeneous interaction. Finally, we will focus on the main requirements for ensuring access control for pervasive environments: context and situation awareness.

2.2 Pervasive Information Systems

2.2.1 Introduction

With the increasing development in telecommunication systems, networking capabilities, processors and hardware devices, a pervasive computing environment should be easily realized but still it's a challenging mission. The implementation of a pervasive system requests combining the efforts of hardware engineers, software engineers and human machine interaction engineers in order to satisfy user needs and to ensure transparency, usability and adaptation.

The desired transparency makes the system vulnerable to security threats and attacks, especially in pervasive environments where it becomes difficult to separate physical security from logical security. ***Digital security*** in our context is the protection of information within a system, this protection includes:

- ***Physical security*** - where a group of measurements are taken in order to prevent attackers from accessing a facility, resource, or information stored on physical media (hardware)

- ***Logical security*** - which aims to provide some measures that would ensure that only authorized users are able to perform actions or access information in a network or a workstation.

2.2.2 Evolution & Definition

The growing need of transparency in the user's interaction with digital services has promoted the notion of Pervasive Computing. ***Pervasive or Ubiquitous Computing*** was first introduced by Weiser as his vision for the computing future in the 21st century [Weiser, 1991].

Weiser believed that the exponential evolution of data, software, hardware and connectivity would generate new environments rich of computing elements that lack proper interaction. As a result, he presented a paradigm where computing elements would disappear from the user's consciousness while functioning homogeneously in the background of his environment. The final objective is to provide users with omnipresent and seamless services available whenever, however and wherever needed [Park et al., 2004].

Pervasive computing allows the coupling of the physical world to the information world, and provides a wealth of ubiquitous services and applications that allow users, machines, data, applications and physical spaces to interact seamlessly with one another [Ranganathan et al., 2005].

Pervasive computing merges physical and computational infrastructures in an integrated environment, where different computer devices and sensors are gathered to provide new functionalities, offer specialized services and boost productivity [Campbel et al., 2002].

While analyzing pervasive computing and studying it's progression, it was found that for hardware and computing elements to disappear, software needs to disappear and the spatial temporal relationships between people and objects (human-machine interaction) has to be well defined in the early design phase within dynamic ubiquitous computing environments [Want et al, 2002].

2.2.3 The Objective of Pervasive Computing

As digital services enter many day life applications and can sometimes replace human beings, technology becomes a leading factor that could affect and enhance life quality and business productivity by assuring higher interoperability between different business partners and the surrounding dynamic environments. With all these dynamic elements, a growing demand for easier, mobile, transparent and seamless interaction is sought in order to adapt with the user's situation, abilities and needs.

In business applications, both services and profit are essential and in order to achieve the aimed profit, user satisfaction and confidance has to be gained by ensuring value-added services that will raise the usability and efficiency of present systems. Current technological applications have many advantages such as connectivity, wireless networking, improved machine performance, excellent processing, storage capabilities and also high quality displays. But these features are not being noticed or appreciated because of the lack of important elements like: usability, security and adaptation. This lack is reducing the trust-worthiness of these systems.

Nowadays, pervasive computing is adopted in many day-life applications where we find it at home, in the office and even in our way embedded in our automobiles [Want et al., 2002] so a vital objective while designing and developing such systems would be aiming to gain the user trust and satisfaction in a way that would promote these systems and assure their success.

2.2.4 Pervasive Computing Environment Sub Components

While studying pervasive systems, we found that these systems are interactive systems that aim to facilitate the interaction of users with unfamiliar systems. Thus, we look at pervasive systems as environments that contain four principal components and that aim to provide them with homogeneous interaction [Al Kukhun and Sèdes, 2006]. These components are: users, data, software and hardware.

The interaction between these components should be as transparent as possible. Transparency becomes a key requirement for ensuring quality and user satisfaction and this transparency can be applied using quality metrics along with run-time, automatic adaptation for both content and context starting from the early stage of design till late testing and execution of the system.

This pervasive environment should adapt with the dynamic, evolutive and distributed systems that extend the boundaries of physical spaces, the building infrastructures and even the devices contained within these environments.

Next, we'll start by characterizing and structuring each sub component of the pervasive computing environments. Dividing the mentioned environment to sub components and defining each component allows us to find its features and the required enhancements that are needed to perfect it's functionality in order to attain a homogeneous environment.

2.2.4.1 Users

Pervasive computing environments are caracterised by being ***user-centered*** applications that aim to satisfy the users of pervasive computing environments who might have different levels of familiarity with the system (from novice users to professionals).

In pervasive environments, users are not interacting to one machine anymore; they are interacting with multiple technologies, moving around non-familiar environments that they find not trust worthy. Users try to stay focused while manipulating and relocating data across devices while their access rights might be changing over time [Duan and Canny, 2004].

A pervasive computing environment should be as mobile as its users in the sense that it should be able to adapt according to the availability of its resources.

With the growing complication of technology and multimodality, novice and disabled users are facing difficulties and are being highly perturbed. At the same time, even professionals are facing problems in their interaction with pervasive systems and are demanding for more adaptive and powerful interaction that would increase the reliability of the system and would enable them to work efficiently.

Multimodal interaction aims to break the barriers between users and technology and to enable smooth, spontaneous adaptive interaction so that users would forget the fact that they are using computers.

We find that users are a changing and dynamic element that has multi-dimensional evolutionary needs and limited capabilities. So if the mission of pervasive computing is to gain user satisfaction, some quality metrics such as usability, security and adaptability should be taken into considertion in order to optimize user interaction in pervasive environments.

2.2.4.2 Data

The data consulted within pervasive environments come in different forms and formats. In the age of hypermedia it can be a text, an image, an audio, or a video stream. So data is not only heterogeneous in kind but also in source where it exists in decentralized systems and comes from different sources that's why we think that the most important aspect in pervasive environments is accessibility which would offer transparent usage of data and have a great influence on the system's efficiency.

Easier access to data and information is the final objective of using pervasive computing environments. In pervasive environments, data is often generated dynamically, in different formats, is streaming at high rates over heterogeneous networks or devices and is dealt in real time. As multimedia and multimodal interaction is advancing, data is becoming of central importance.

In pervasive environments, we need to unify and structure the content of various data formats in order to manipulate information in an easier way and adapt with their heterogeneity.

2.2.4.3 Hardware

Hardware devices are the ***physical components*** of the pervasive environments and they are often different COTS "Commercial of The Shelf" products that are equipped with advanced networking capabilities such as Bluetooth and Wi-Fi. Invisible embedded devices and sensors are turning physical spaces into active, smart surroundings making the space interactive and adaptable [Munoz et al., 2005], [Campbel et al., 2002].

Nowadays, a huge evolution in the computing machinery performance has been achieved; the progression of hardware devices is being beneficial and of a great effect on ubiquitous computing where resources are accessed and shared by multiple users.

Many improvements took place and helped to adapt with the mobility and dynamicity of pervasive environments, specially the shrinking size and weight of hardware devices; where users are moving freely while handling their small, light devices that are provided with high connectivity and wireless networking capabilities such as Bluetooth and Wi-Fi [Campbel et al., 2002]. Processing capabilities are increasing over time and offering higher efficiency levels especially with the extensible storage capacities and high quality displays that help the user forget the barriers and allow him to acquire the underlying information unconsciously without effort.

Hardware devices have entered a huge development life cycle but still have many limiting factors such as the growing interaction complexity caused by the

shrinkage of machine size and the increasing cognitive overload due to the inadaptable interface design, especially that a user could be a novice user who needs a user-friendly environment or a professional who needs a highly developed environment.

Pervasive environments require devices that can be used and integrated easily, that can balance between providing high security levels, total privacy protection and high interoperability.

Hardware devices are advancing exponentially in several aspects such as: storage capacities, shrinking size and weight but meanwhile these advanced functions are affecting negatively the usability and the HCI.

With ad-hoc networking, connectivity has become an easy mission but at the other side, it has become a risky, dangerous and unreliable channel.

2.2.4.4 Software

Software is the ***logical component*** in a pervasive computing environment and it's of central importance; it enables the connection between different heterogeneous devices and different users within a dynamic environment and it performs suitable mappings between each task and the desired services that users require [Chen et al., 2004].

In order to deal with the dynamicity of devices used, situations encountered, users accessing the system and the environments from which the users are connected, the software should be highly adaptable and flexible. The software of a pervasive system should be able to integrate many devices and external software systems in order to provide services that meet user needs and assure the homogeneity and collaboration between its components.

The use of XML based technologies enhances the flexibility of systems and enables adaptable mappings between a task and it's provided services [Chen et al., 2004].

Aiming for a better specification and implementation of productive pervasive systems with no complexity, a domain specific language ***Perv-ML*** Pervasive Modeling Language was proposed in [Munoz et al., 2005] along with the combination of two software engineering trends: ***software factories*** and ***MDA model-driven architecture***. MDA was introduced to deal with the low abstraction level that is caused by the heterogeneity of used technologies and software factories were used to enhance the programmability and reduce the amount of programming code.

Being in a real time streaming environment, a system requires multi-channel streaming over heterogeneous networks and devices and must support multi channel protocols in order to ensure interoperability with the existing multimedia systems [Park et al., 2004].

In such open dynamic pervasive environments, a demanding need arises for security, privacy, authentication and access control as it's important to prevent unauthorized access attempts. This has motivated the emergence of XACML ***eXtensible Access Control Markup Language*** [OASIS, 2003, 2005a]; a new XML-based policy language that automates the several managerial required tasks and enables interoperable interactions between several applications along with the reuse of access control policies [Almenárez et al., 2005].

In order to increase the productivity, quality and interoperability, the implementation of a pervasive computing environment can follow one of two different programming models as follows:

1) Context - Driven Model where several contexts can be defined in advance using descriptive logic. On runtime, the system will explicitly know the behavior that it should follow according to its current state; this model assures the interoperability, extensibility and scalability of the system and is ideal for discovering contradictory system behaviors, to detect conflicts and to adapt with multimodal environments.

2) Service - Oriented Model which is considered as a more expressive, proactive and procedural model that allows higher programming control levels and concentrates on the services that should be provided using several decision making techniques that memorize past actions along with the environment's history during a service lifecycle [Yang et al., 2006].

2.2.5 Conclusion: Interaction challenges & the importance of adaptation

As we have shown through this section, pervasive systems are user-centered systems. A pervasive environment is composed of several sub components interacting with eachothers in order to provide users with transparent accessibility to desired resources or services at anywhere, anytime and anyhow.

As a conclusion, we present, in Fig. 1, the main challenges facing the interaction of the different sub-component within pervasive environments knowing that ensuring some qualities might contradict with the existence of some others. A balance between contradictory quality metrics is highly needed and that's why we highlight the importance of applying an adaptive layer that would help achieve this mission [Al Kukhun and Sèdes, 2006].

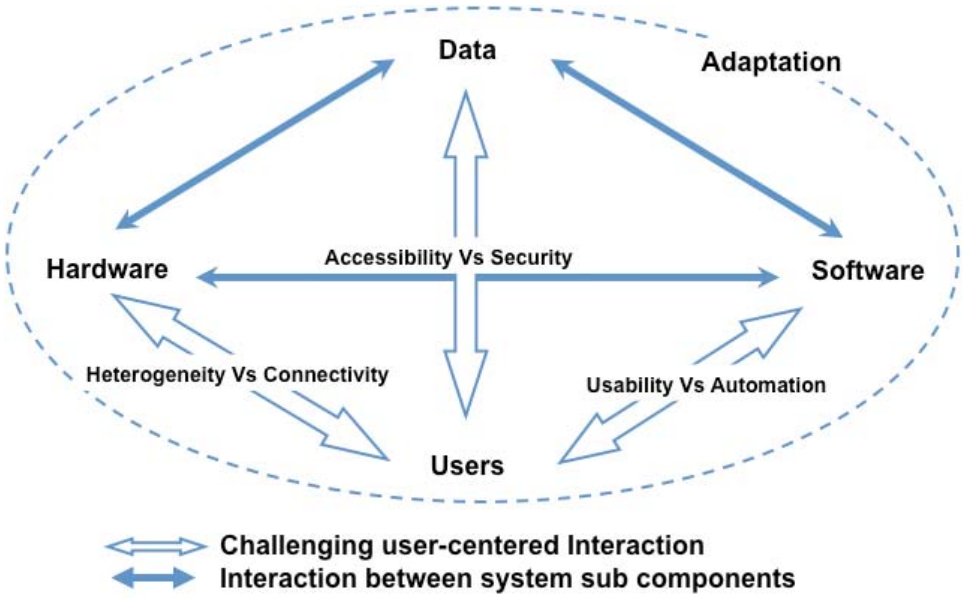


Figure 2.1: The effect of adaptation in guaranteeing a homogeneous interaction between the sub-components pervasive environments

In the ***user -- Data*** relationship, the user needs to obtain easy and transparent accessibility to the desired system ressources. Successful data access and retrieval can

be attained by ensuring better data representation, integration and management but in a pervasive context, users are mobile, they log in from different places and get connected through open and dynamic environments. Thus, the system should pay great attention to ensuring being safe and secure enough to protect it's resources in order to make sure that only authorized users are allowed to obtain access.

The challenge is to achieve a balance between data transparency and data discretion [Gschwind et al., 2002]; ***data transparency*** states that a pervasive system should be open enough to allow seamless accessibility to information sources. At the other hand, ***data discretion*** makes sure that the only users granted access to information are the ones authorized to do so. We highlight the importance of applying multi-layered adaptation in order to balance between the user requirements and the system security constraints.

As we will show through the rest of this thesis, our research works are directed to find solutions to achieve this balance. We highlight the importance of using proactive and interoperable access control policies that would ensure fine-grained expression for access conditions. Then we apply adaptive (context and situation aware) decision making that would gouvern access requests issued at anytime, anywhere and anyhow.

Similar challenges are present in the ***user -- hardware*** relationship, where hardware engineers want to provide users with small-sized, light-weighted, small-screened devices that have high storage and processing capacities. At the other side, users need to interact easily with their machines and other sensors embedded in the surrounding environment without facing human-machine interaction difficulties or being influenced by the networking or connectivity problems that could deprive them of smooth and efficient usage.

Thus, applying multimodal interaction and content adaptation to meet the user's needs seems to be unevitable. The adaptation will take into consideration the user preferences, device capabilities and the network conditions. It can be applied based on the client profile, network characteristics and the content profile (meta-data).

Finally, the ***user -- Software*** relationship is also a confusing one; the user demands for flexible, usable and interactive systems that would enable him to control the system behavior and as a result obtain a system that can evolve to meet his changing needs. However, software engineers tend to provide users with automated services that react towards predicted situations and this way, the system behavior and reliability will be guaranteed without worrying about undesired user control.

In order to assure a homogeneous interaction within pervasive systems, adaptation should be taken into consideration not only to the content; where the systems consider the exchanged file formats, the storage space or the available software but also to the context, where the system is sensitive according to the situation of the environment as a whole, the identity of its users and their location, the efficiency of it's machines and interfaces and finally the effectiveness of it's networking capabilities.

The adaptation should be taken into consideration at all the phases of development and should be as dynamic as its environment; especially that user needs are changing along the time, according to his location, state or situation and to the availability of its resources.

Multi-layered adaptation means that the adaptation should be applied on all the layers of the system starting with the first step of system conceptualisation till the late run-time execution phase.

Adaptation guarantees the homogeneity of pervasive environment components and assures better interaction between sub-systems. With multi-layered adaptation, systems could accomplish the first and most important target that is user satisfaction.

2.3 Access Control Challenges within Pervasive Systems

2.3.1 Introduction

Providing quality metrics like security, privacy and access control is very important within pervasive systems as thay could serve as a protection shield when considering the risks that accompany other qualities like transparency, sharability and interoperabile accessibility within decentralized systems and unfamiliar environments where sources generate dynamic data in a real-time basis.

Controlling access within pervasive systems is considered as a challenging mission since it can be viewed from different angles [Al Kukhun and Sèdes, 2007]. Looking at it from a user's point of view, access control can be considered as a barrier to accessibility since it might not provide users with access to a desired resource at anytime, anyhow and anyway. Meanwhile, if we take the system administrator's point of view, access control is considered extremely important to ensure secure interactions within open environments especially if a system could tie contextual constraints (time, location, network bandwith, machine characteristics, etc.) to access decisions and provide dynamic access to resources according to the user's context.

Another challenge in ensuring access control within pervasive environments concerns the interpretation of the context itself since it can be differently expressed and defined according to the application domain, to the designer's objective or to many existing techniques employed for context acquisition, modeling and interpretation.

Furthermore, knowing that in pervasive environments, access to information sources takes place in real-time conditions, an access control model should be flexible and responsive enough to deal with any type of situation confronting the user while demanding access (emergency, un-expected event, etc.).

The aim of our research was directed towards ensuring these both qualities within an access control model as many research works have highlighted the importance of both context and situation awareness for pervasive systems: according to [Zimmermann et al., 2005], pervasive computing should not only be aware of the context but should also be able to capture situational information.

We have chosen to concentrate our research works around the RBAC (Role Based Access Control) model due to it's wide spread usage and it's efficiency in managing access privileges within distributed environments.

The emergence of the new pervasive paradigm has revealed that classical RBAC model can't satisfy the access control requirements in a rapidly changing dynamic pervasive environment. Thus, there is a need for expressing context and situation within this model.

Accordingly, as we will show next, different research works took place aiming for developing new RBAC models considering contextual attributes and situation-awareness within the decision-making process.

2.3.2 Access Control and Context-Awareness

Integrating context information as part of an access control system is a challenging task due to several reasons [Kulkarni et al., 2008]:

1) Acquiring appropriate context information requires interfacing the access control system with various kinds of ambient sensors. Integrity and authenticity of this information is paramount because it may be used in making access control decisions.
2) Certain aspects of the context information may be inherently dynamic in nature. During the course of execution of a context-dependent task, it is possible for the related context condition to become false. For certain applications, it may be important that a role member's permissions to execute that task are revoked when such context changes occur.
3) Context-based constraints may restrict the resources and services that may be dynamically interfaced with a pervasive computing application.

2.3.2.1 Introduction: Definition of Context

Various definitions of context have been proposed in the literature. Broadly, the notion of context in pervasive computing applications relates to the characterization of ambient conditions and physical world situations that are relevant for performing appropriate actions in the computing domain for its correct or desired behavior.

A person's context can be defined in terms of his/her current physical location, devices being used, network on which the devices are connected, and the activities in which the user is currently engaged. Additionally, there can be other conditions and characteristics that may be relevant in defining a context.

For example, in some situations the temporal attributes associated with an activity, such as its duration and time of occurrence, may be important. Other factors such as device capabilities, physical proximity of devices, and available bandwidth can also be important in some situations [Kulkrani et al, 2008].

The term "Context Aware" was first introduced to the mobile computing community by Schilit and Theimer [Schilit et al., 1994]. In their definition, context is defined as "the location and identities of nearby people and objects and changes to those objects". While this definition is useful for mobile computing, it defines context by example, and thus is difficult to generalize and apply to other domains.

Pascoe in [Pascoe, 1998] defines context to be a subset of physical and conceptual states of interest to a particular entity. This definition has sufficient generality to apply to a recognition system.

Cheverest [Cheverest, 2000] describes context in anecdotal form using scenarios from a context aware tourist guide. His system is considered one of the early models for a context aware application.

Finally, Dey has reviewed the definitions of context, and defined it as: "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and application themselves" [Dey, 2001]. He

considered a system to be context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

2.3.2.2 Some Solutions for Context-Aware Access Control

In ubiquitous computing environments, users are mobile and typically accessing resources using mobile devices. As a result the user's context (e.g., location, time, network state, etc.) becomes highly dynamic, and thus, granting a user access without taking the user's current context into account can compromise security as the user's access privileges not only depend on "Who the user is" but also on "Where the user is" and "What is the user's state and the state of the user's environment".

As a result, even an authorized user can damage the system's integrity as the system may have different security requirements within different contexts. Thus, access control mechanisms for ubiquitous applications require that changes of the privileges of a user dynamically based on contextual information [Lim and Shin., 2007].

With the development of pervasive information systems, role and permission assignment for a user has become more complex and dependent on his context. That is why many research works have proposed to extend the RBAC models in order to take into account the evolving definition of context (time, location, system characteristics, network connection, machine, etc.).

1. Temporal RBAC

Time was the first contextual element that was taken into consideration within the model RBAC. [Bertino et al., 2001] have extended the RBAC model to present Temporal RBAC (TRBAC), which considered time as an important constraint that can determine the activation and deactivation of a role.

The TRBAC supports periodic role enabling and disabling and temporal dependencies among permissions by introducing time into the access control infrastructure.

A role is enabled if assumed by a user. Priorities are associated with role events, which in conjunction with a set of precedence rules, are used to resolve conflicts. TRBAC also allows an administrator to issue runtime requests for enabling and disabling a role.

Integrating the temporal aspect has given more flexibility to create exceptions for individuals and to specify time dependencies between the different actions that can be performed by a user.

The model, however, cannot handle several other important temporal constraints, which are elaborated as follows: First, the model does not include temporal constraints for the user-role and role-permission assignments. It assumes that only roles are enabled and disabled at different time intervals.

2. Generalized Temporal RBAC

The GTRBAC model was presented as an extension to the TRBAC model in order to allow better specification of a comprehensive set of temporal constraints [Joshi et al., 2005]. In particular, constraints on role enabling and activation and various temporal restrictions on user-role and role-permission assignments can be specified through the GTRBAC model.

The model has also presented time-based semantics of hierarchies and SoD (Seperation of Duty) constraints. A notion of safeness has been introduced to generate a safe execution model for a GTRBAC system.

In a temporal context, it is essential to establish unambiguous semantics of permission-inheritance and role-activation within a hierarchy when enabling and/ or activation times of hierarchically related roles are considered.

In a role hierarchy, permission-inheritance semantics identify the permissions that a role can inherit from its junior roles. Similarly, once a user is assigned a role, the role-activation semantics identify the set of junior roles that can be activated by that user.

Prior to presenting the temporal hierarchies and time-based SoDs, the GTRBAC model introduced four status predicates that model the acquisition and role-activation semantics beyond the explicit assignments through the hierarchical relations among roles. Three categories of hierarchies were defined:

1. ***unrestricted hierarchies***, in which permission-inheritance and role-activation semantics are not affected by the presence of any timing constraints on the hierarchically related roles,
2. ***enabling time restricted hierarchies***, in which the permission-inheritance and role-activation semantics depend on the enabling times of the hierarchically related roles, and
3. ***activation time restricted hierarchies***, in which the permission-inheritance and role-activation semantics depend on the active states of the hierarchically related roles.

The unrestricted and enabling-time restricted hierarchies may be of three types: inheritance-only hierarchy related roles (I-hierarchy), activation-only hierarchy (A-hierarchy), or inheritance-activation hierarchy (IA-hierarchy).

3. Spatial RBAC

The Spatial RBAC (SRBAC) model was proposed by [Hansen et al., 2003]. It extends the RBAC model to incorporate location information associated with roles in order to permit location-based security policies. In the SRBAC model, permissions are dynamically assigned to the role dependent on location, thus a user assigned to a role may have different permissions depending on his location.

Users are considered to be mobile units that can establish (wireless) communication with system resources to perform some activities.

Locations are represented by means of symbolic expressions called location expressions that describe location domains identifiable by the system.

A location domain Z is divided on the physical layer into subareas, called primary location cells denoted as πi, i = 1, .., k, which reflect the ability of the underlying architecture to uniquely map user location into cells. However, using primary location cells in SRBAC can be unpractical therefore the authors introduced logical location domains that reflect organizational location infrastructure and organizational security policy.

A location l from LOC is called homogeneous with respect to role r from Roles if r has the same permissions available in any position inside l. Location l from LOC is

called homogeneous (with respect to Roles), if it is homogeneous with respect all r from Roles.

Definition 1. Set of locations L = {l_1, l_2, ..., l_k} from LOC are called normalized with respect to set of roles R from Roles if it is :

- a partition of LOC : $LOC = \bigcup_{i=1}^{k} l_i, li \cap lj = \varnothing \; for \; i \neq j$
- any location li from LOC is homogeneous with respect to R.

Definition 2. SRBAC model consists of the following components.

- USERS, ROLES, PRMS, SESSIONS and LOC, represent the finite set of users, roles, permissions, sessions and locations respectively,
- U A USERS × ROLES, the relation that associates users with roles,
- assigned_users(r : ROLES) → 2^{USERS} , the mapping of a role onto a set of users. Formally: assigned_users(r) = {u USERS | (u, r) UA},
- PA ROLES × LOC × P RMS, the relation that assigns a permission to a role available in location,
- assigned_permissions(r : ROLES, l : LOC) → 2^{PRMS} , the mapping of a role r onto a set of permissions based on location. Formally: assigned_permissions(r, l) = {p P RMS| (r, l, p) P A},
- user_sessions(u : USERS) → $2^{SESSIONS}$, assigns a user onto a set of sessions,
- session_roles(s : SESSIONS) → 2^{ROLES}, the mapping of each session to a set of roles,
- avail_session_permissions(s : SESSIONS, l : LOC) → 2^{PRMS} , the permissions available in a session for a location, r session_roles(s) assigned_permissions(r, l).

The authors have also defined a hierarchical SRBAC and constrained SRBAC.

4. Geo-RBAC

The GeoRBAC was presented in [Bertino et al., 2005] as a comprehensive framework, general and flexible enough to deal with spatial aspects in real mobile applications, see figure 2.3. The objective was to secure data access within location-based services and mobile applications.

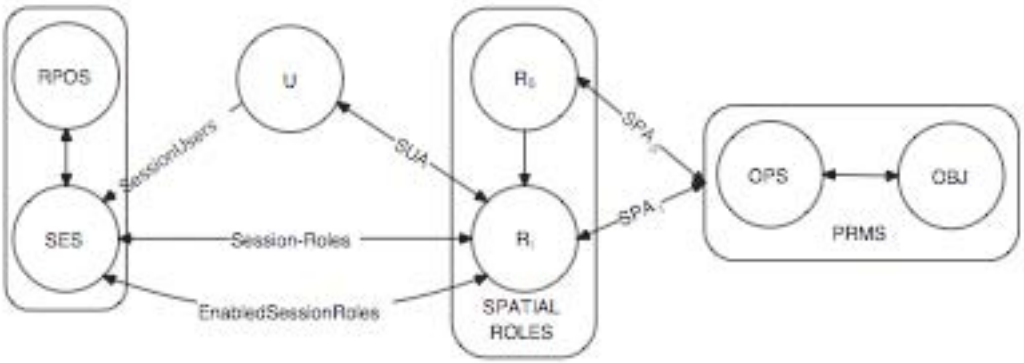


Figure 2.3: The Core Geo-RBAC Model

A ***spatial role*** in GEO-RBAC represents a geographically bounded organizational function. The boundary is defined as a feature, such as a road, a city or a building. The boundary specifies the spatial extent in which the user is to be located for being enabled to play such a role.

The model distinguished between a ***physical position***, obtained from a given mobile terminal such as a GPS based vehicle tracking device or a cellular phone, and a ***user's logical position*** representing the feature in which the user is located (device independent). Logical positions can be computed from real positions by using specific mapping functions.

To enhance the flexibility of the model, it was assumed that logical positions can be represented at different granularities, depending on the spatial role played by the user. If the user is located inside the spatial boundary of the role that has been selected (activated) during the session he/ she has logged in, the role is said to be enabled. To specify the type of the spatial boundary of the role and the granularity of the logical position, the concept of spatial role schema is introduced.

Spatial roles are thus specified as instances of role schemas. The usage of role schemas and instances makes the model quite flexible since the type of role extents and logical positions can be customized (and the definition re-used), depending on the function the role represents.

GEO-RBAC is a comprehensive model, which like RBAC, consists of three components referred to as Core, Hierarchical and Constrained GEO-RBAC:

Core GEO-RBAC specifies the basic concepts of the model, thus the notion of spatial role, role schema, real/logical position, activated/enabled role, which are used by the subsequent components.

Hierarchical GEO-RBAC extends the conventional concept of hierarchy by introducing two major novelties:

- First, two distinct hierarchies are provided, one over role schemas and one over role instances. The role schema hierarchy supports the inheritance of permissions and user memberships among sets of homogeneous roles and thus further simplifies role definition.
- The second extension concerns the formal definition of role activation and enabling in the presence of hierarchies. To this purpose, a model is presented in which the role instance hierarchy is used to derive the roles which not only are activated but also enabled in a session.

Constrained GEO-RBAC supports the specification of separation of duty (SoD) constraints for spatial roles and role schemas. Since exclusive role constraints are important to support the definition and maintenance of access control policies in mobile contexts, SoD constraints are extended to account for different granularities (schema/instance level), dimension (spatial/non-spatial), and different verification time (static, dynamic at activation time, dynamic at enabling time). The resulting set of constraints represents the first comprehensive class of constraints for spatially-aware applications.

5. Dynamic RBAC

Dynamic Role Based Access Control (DRBAC) model [Zhang and Parshar, 2003] addresses the dynamic access control requirement of applications in pervasive environments. It extends the traditional Role Base Access Control (RBAC) model to use dynamic context information while making access control decision.

Specifically, DRBAC addresses two key requirements:

(1) A user's access privileges must change when the user's context changes.

(2) A resource must adjust its access permission when its system information (e.g., network bandwidth, CPU usage, memory usage) changes.

The model, demonstrated in figure 2.2, dynamically adjusts Role Assignments and Permission Assignments based on context information. In the proposed approach, each user is assigned a role subset (by the authority service) from the entire role set. Similarly the resource has permission subsets for each role that will access the resource.

During a secure interaction, state machines are maintained by delegated access control agents at the subject (Role State Machine) to navigate the role subset, and the object (Permission State Machine) to navigate the permission subset for each active role.

The state machine consists of state variables (role, permission), which encode its state, and commands, which transform its state. These state machines define the currently active role and its assigned permissions and navigate the role/permission subsets to react to changes in the context.

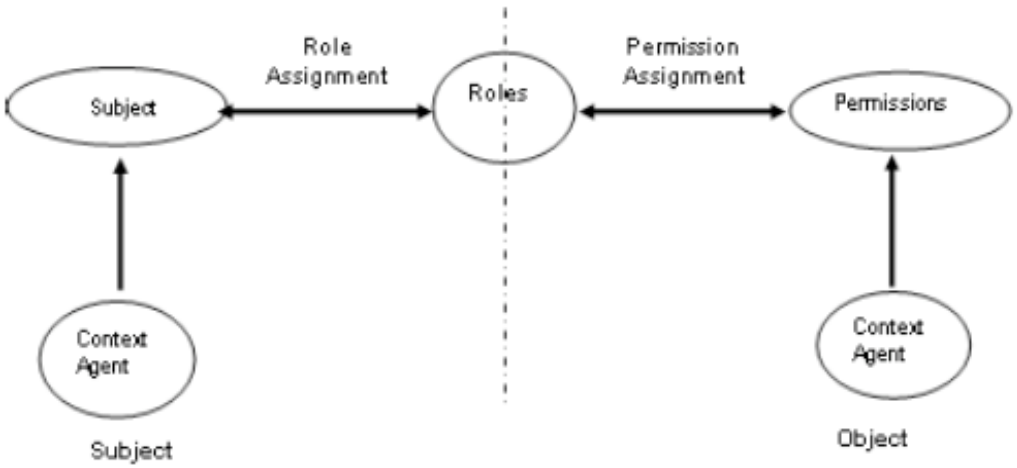


Figure 2.2: The Dynamic Context Aware RBAC Model

The main components of the DRBAC model are:

- ***USERS***. A user is an entity whose access is being controlled.
  USERS represents a set of users.
- ***ROLES***. A role is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role. ROLES represents a set of roles.
- ***PERMS***. A permission is an approval to access one or more RBAC protected resources. PERMS represents a set of permissions.

- ***ENVS***. ENVS represents the set of context information in the system. An authorized "Context Agent" is employed to collect context information.
- ***SESSIONS***. A session is a set of interactions between subjects and objects. A user is assigned a set of roles during each session. The active role changes dynamically among the assigned roles for each interaction. SESSIONS represents a set of sessions.
- ***UA***. UA is the mapping that assigns a role to a user. In the session, each user is assigned a set of roles, the context information is used to decide which role is active. The user will access the resource with the active role.
- ***PA***. PA is the mapping that assigns permissions to a role. Every role that has privilege to access the resource is assigned a set of permissions, and the context information is used to decide which permission is active for that role.

In the approach, a Central Authority (CA) maintains the overall role hierarchy. When the user logs on the system, based on the user's capability, a subset of the role hierarchy is assigned to the user for each session. Then the CA sets up an agent for that user and delegates the user's right to that agent. The agent will monitor the environment status of the user and dynamically change the active role of the user.

Every resource maintains a set of permission hierarchies for each potential role that will access the resource. The resource maintains its environment and dynamically adjusts the permissions for each role.

6. uT-RBAC

The UbiquiTous Role-Based Access Control model was introduced by [Chae et al. 2006]. The model considers the time and the location of the user as important elements for the activation and disabling of a role.

The model proposes a simple method for representing situational information for ubiquitous computing environments. The role states of the uT-RBAC are Assign, Disable and Enable. By using the role states, the authors reduced the representation complexity.

The state of user's role is changeable during a session. When a user logs into system, the system assigns roles to user. At this point in time, the role state is Assign. The system checks situation information, after which the role state is changed to Enable or Disable whether it is satisfied or not.

The situation information regards to the information of user's time and location. In uT-RBAC model, this is represented as new constraints for time and location varying environment.

The constraint C is a product of LC × TC.

- ***LC*** is ***location constraint***, which is a set of {l1, l2, l3, . . . , lj} or Anywhere. l is represented by symbolic location name (e.g. Room202 and Second Floor) and includes hierachical expression.
- ***TC*** is ***time constraint,*** which is a set of {t1, t2, t3, . . . , t j} or {ti : tk}.
  t is represented as {Time|Day|Year|Anytime}.
  ':' means continuous operation. For example, Monday:Friday means continous time duriation from Monday to Friday.

The access control policy is represented as a form of (C, RoleS tate, Rolenames). It is same as (LC, TC, Rolestate, Rolenames).

7. Context-Role Based Access Control Model

The context-role based access control model adds a notion of context-role to a traditional role based access control. Proposed by [Park et al., 2006], the context-role represents environment state of the system by a mapping context-roles and context information. The model was built using the uniform notion of a role to capture both user and context attribute and illustrated by a simple intelligent home example.

CRBAC has three relations UA, PA, and CA that define the associations between user roles, user permission assignments and context roles.

– UA: UA is the mapping that assigns a user role to a user.
– CA: CA is the mapping that assigns a context role to a context.
– PA: PA is the mapping that assigns permissions to a role.

The CRBAC model is formally defined as follows :

- U, C, R,UR,CR, P, S (users, contexts, roles, user roles, context roles, permissions, sessions, respectively).
- UA U × UR, a many-to-many mapping user-to-user role assignment relation.
- assigned users (ur : UR) → 2^U, the mapping of user role ur onto a set of users. Formally: assigned users (ur) = u ◘ U|(u, ur) ◘ UA
- R 2(UR × CR), the set of roles.
- PA P × R, a many-to-many mapping permission-to-role assignment relation
- assigned permissions (r : R) → 2^P , the mapping of role r onto a set of permissions.
- user sessions (u : U) → 2^S, the mapping of user u onto a set of sessions.
- session roles (s : S) → 2^R, the mapping of session s onto a set of roles. Formally: session roles(si) r ◘ R|(session users(si), r) ◘ UA

The context role is used to capture security-relevant context information in CRBAC policies. Context means situational information. Almost any information available at the time of an interaction can be seen as context information.

The context role shares many characteristics with user roles. So, context role has role activation, role revocation, and role hierarchies. Fig. 2.4 show an example of role hierarchies of context role.

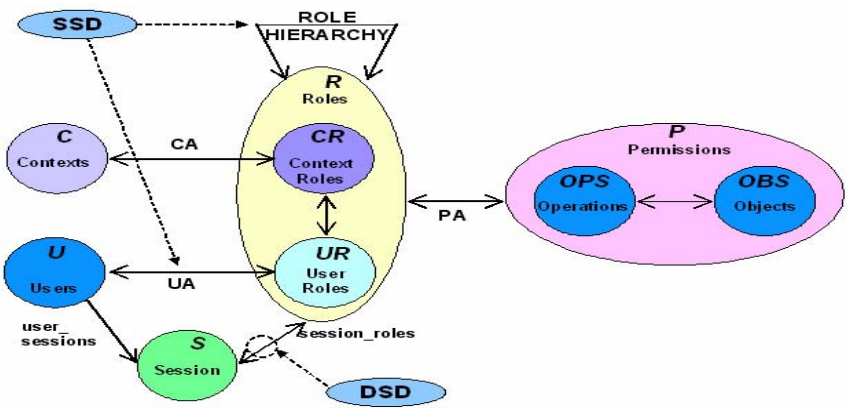


Figure 2.4: example of role hierarchies of context role

8. Context-aware Access Control Model for Pervasive Environments (CAP)

Other research studies as [Emami et al., 2007] have presented a Context-Aware Access Control Model for Pervasive Computing Environments, the model is based on the RBAC model and emphasizes the fact that the Contextual attributes are very dynamic which may risk destabilizing the permissions.

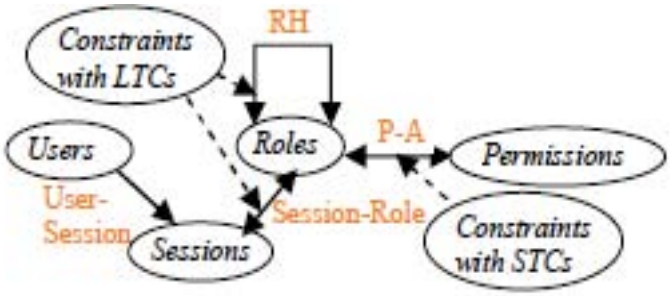


Figure 2.5: The CAP Model

Accordingly, the authors distinguished between 2 types of contextual factors that can be connected to the user or the environment, see fig.2.5:

(i) **Long-term** contextual elements: connected to ***role assignment***.

> ***Long-Term Context (LTC)*** is the one that its value does not change in a time period, named μ times of average session lifetime, such as age, weight and system capabilities. LTC-Set contains two sets of LTCs: ***Environmental LTC*** (E-LTC-Set) and ***User LTC*** (ULTC-Set).
>
> ***Role Assignment Condition (RAC)*** maps a subset of Environmental LTC-Set and User LTC-Set to each role
>
> $$RAC: R \rightarrow P(U\text{-}LTC\text{-}Set) \times P(E\text{-}LTC\text{-}Set)$$

(ii) **Short-term** contextual elements: attached to ***permission assignment***.

> ***Short-Term Contexts (STC)*** are elements that may be changed during a session, such as time, location and CPU load. STC-Set also includes ***Environmental STCs*** (E-STC-Set) and ***User STCs*** (USTC-Set).
>
> ***Role-Permission Condition (RPC)*** is a mapping function that assigns a subset of E-STC-Set and USTC-Set to each role with a specific permission.
>
> $$RPC: R \times Prm \rightarrow P(U\text{-}STC\text{-}Set) \times P(E\text{-}STC\text{-}Set)$$

Finally, a ***Request-Authorization*** mapping function assigns a "Grant" or "Deny" response to the session and requested permission (according to permission conditions and current Short-Term contexts).

9. Intelligent Access Control Mechanism for Ubiquitous Applications

The proposed scheme extends the RBAC model by adding dynamic role and permission assignments based on context information. The authors [Lim and Shin, 2007] have confronted the dynamicity of contextual constraints by using intelligent access decision-making through a back propagation neural networks algorithm.

An access policy specifies a set of contextual information as follows:

<UserID, Object, Authorization type, Place, Time, System Information, Priority>

- ***UserID***: who sent the request.
- ***Object***: Data object is being accessed.
- ***Authorization type***: Type = {+, -},
  – means negative permission,
  + means positive permission.
- ***Location***: Where the access request was issued (e.g., IP address, physical location, category of location such as "mobile")
- ***Time***: When this access request was issued.
- ***System Information***: When this access request was issued at that time system information such as network bandwidth.

An access policy specifies which role has what kind of permissions under some contextual constraints. Although access policies are mainly used by the authorization engine to make access control decisions, these policies also need to be exchanged among trust situations. In order to attain a more suitable category classification for permissions assignment, a neural network algorithm was applied for making access control decisions.

Fig. 2.6 illustrates the proposed neural network model. It has input layer, hidden layers and output layer. Its elements are as follow:

- ***Input layer***: The data inputs without distortion to hidden layer.
- ***X1, …, Xn***: Input data (e.g., active role(X1), userID(X2), time(X3), location(X4), authorization type(X5),…, other context information(Xn))
- ***Output layer***: The output layer where the classified information is retrieved.
- ***Pm***: Output data (permission(Pm))

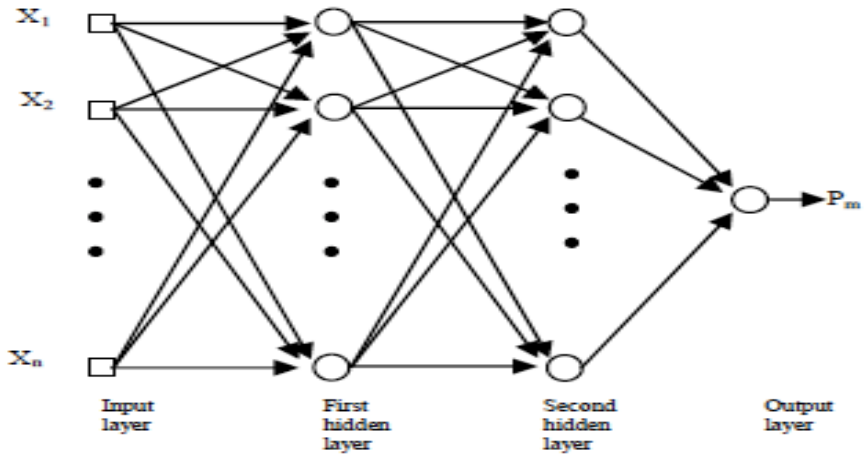


Figure 2.6: Architecture of The Back Propagation Neural Network

The proposed scheme insures the following characteristics:

- It prevents an unauthorized user from using system ressources.
- It provides dynamic control over user privileges whenever a user's context changes.

- It offers more suitable permission assignments as it intelligently grants and adapts permissions to users according to current context with user roles.

10. Context-Aware RBAC in pervasive computing systems

Another extension was proposed by [Kulkarni et al., 2008] who provided a context-aware RBAC model to meet the requirements of pervasive systems. The model, illustrated in fig. 2.7, separated context management from the access control model in order to facilitate decision-making in cases where an authorization decision is connected to several contextual constraints. This was accomplished by a special service dedicated for context and resource management.

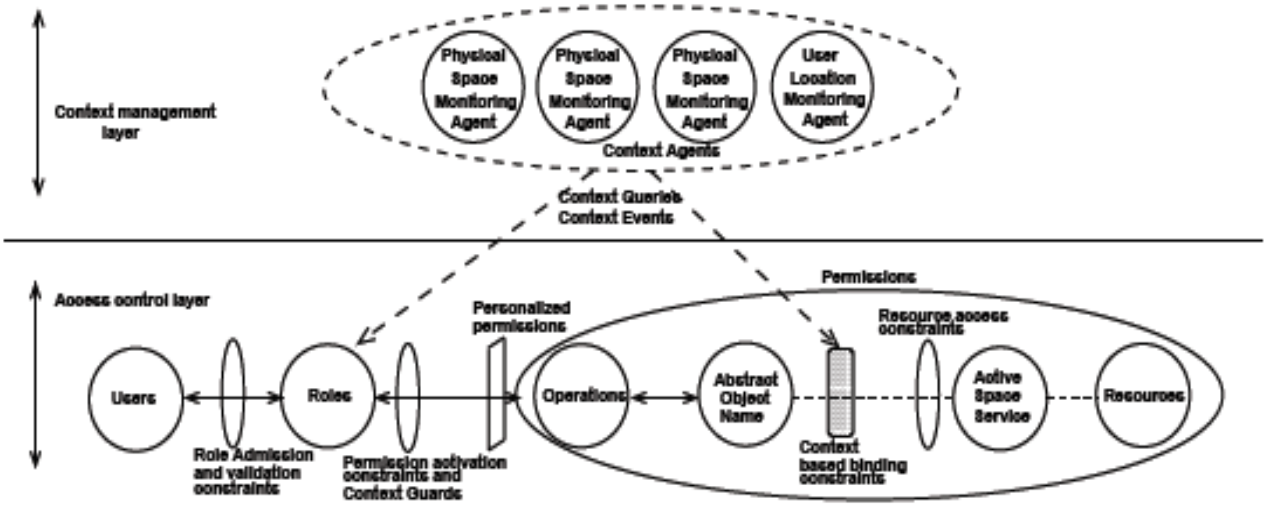


Figure 2.7: The Context-Aware RBAC Model (CA-RBAC)

The context-aware RBAC (CA-RBAC) model was accompanied by a role-based programming framework for designing secure context-aware applications. The framework contains high-level abstractions for specifying context-based access control requirements, specifically addressing the following aspects:

- ***Role admission and validation constraints:*** These constraints specify context-based conditions that need to be satisfied before admitting a user to a role, and also for continuing a user's membership in a role.
- ***Context-based role permissions:*** Dynamic object binding causes role operations to interface with different services under different context conditions.
- ***Personalized role permissions:*** Such permissions allow different role members to access different active space services based on their individual context.
- ***Context-based permission activation constraints:*** These constraints are associated with specific role permissions, and specify context-based conditions that need to hold for a role member to execute such permissions.
- ***Context-based resource access constraints:*** These constraints restrict a role member's access to a subset of resources that are managed by an active space service.

The model also supports revocation of a user's membership in a role when context conditions fail to hold. The authers have also identified the context invalidation problem and addressed this problem by providing a context guard mechanism in the CA-RBAC model.

2.3.2.3 Survey

In analyzing the different models that we summarized in the previous section, we highlight the importance of taking context into consideration while modeling access control for pervasive environments. As we show in our conluding table 2.1, the models have evolved with the evolution of the context definition.

As we can see, the main objective of the surveyed models is to provide access decisions that meet the dynamicity of contextual constraints within pervasive environments.

No	Contextual Constraints		Time	Location	Date	Other
	Model	Reference				
1	T-RBAC	[Bertino et al., 2001]	x			
2	GTRBAC	[Joshi et al., 2005]	x			
3	Spatial RBAC	[Hansen et al., 2003]	x	x		
4	Geo-RBAC	[Bertino et al., 2005]	x	x		
5	Dynamic RBAC	[Zhang et al., 2003]	x	x	x	Role State Machine
6	uT-RBAC	[Chae et al., 2006]	x	x		Logic
7	Context Role-Based Access Control (CRBAC)	[Park et al., 2006]	x	x	X	
8	Context-Aware Access Control Model for Pervasive Computing Environments	[Emami et al., 2007]	x	x	x	
9	Intelligent Access Control Mechanism for Ubiquitous Applications	[Lim and Shin, 2007]	x	x		neural networks
10	Context-aware RBAC in pervasive computing systems	[Kulkarni et al., 2008]	x	x	x	

Table 2.1: The evolution of context-aware RBAC models

After completing this study on the various extensions proposed to adapt the RBAC model in order to meet the evolving needs of pervasive and real-time systems, we notice that these extensions did not take into account the user's situation and the complexity that might take place in depending on contextual based authorizations within critical situations that he might face a user (emergency case, fire, etc.).

In the next section, we'll explore the different research works that were interested in adapting access control decisions to meet the necessity imposed by the situation at which the user demands access.

2.3.3 Access Control and Situation-Awareness

As we have shown in the previous section, the decision-making process in RBAC has evolved to suit the dynamicity of contextual elements of pervasive environments. As a result, the access control mechanism does not only tie permission assignment to the user's identity and role but also to his contextual attributes (that evolve over time).

Meanwhile, pervasive computing should not only be aware of the context but should also be able to capture situational information [Zimmermann et al., 2005]. Taking into consideration that in pervasive environments, users demand access to information sources in real-time conditions, we highlight the importance of providing an access control model that is flexible and responsive enough to deal with any type of situation confronting the user (emergency, un-expected event, etc.).

In this section, we'll start by citing some situation definitions included in the literature then, we'll expose the different research works that were interested in adapting access modelling to deal with real-time situations and finally, we'll conclude with a survey describing the main directions that were followed.

2.3.3.1 Introduction / Definition of Situation

The term « situation » can be mixed with the « context » as many research works have interpreted it differently according to the research area and application domain (e.g. the works of [Bouidghaghen et al., 2010] have treated situation-aware user profiles depending on the user's environmental data).

The works within the context-aware computing area have tied the definition of the context as being the attributes characterizing the user's situation [Dey, 2001]. As contextual elements are acquired and interpreted, an entity's state is determined. A situation is described to be a collection of states. Finally, the situation is employed as an input to enhance the quality of services provided by the adaptive system.

The works of [Weißenberg, 2006] highlighted the importance of qualifying the information resulting from context acquisition (location, time, temperature, etc.) in a more ***high-level*** and ***time-invarient*** way called the situation of the user (e.g. eating at home, driving the car, working in the office). High-level transformations were achieved through logical labels annotating a list of context dimension values. For example for context dimension (Location: Home, Time: Night), the associated high-level qualifications would be: sleeping.

The works of [Kawagoe et al., 2011] defined the situation and tied it to access control. According to them, a situation is an abstract of conditions which is composed of user contexts and related objects contexts. For example, a user A can obtain access permission privileges on the blood type of a patient of user A if the user A is under consultation and the patient is also hospitalized.

In some research works interested in task-oriented recommendation systems and based on ontologies, contextual attributes classification and ontologies were used as an input to extract situations [Luther et al., 2008]. The situation was employed as an enrichiching element for the recommendation system and served to increase the precision of services offered to the mobile user.

Some research works interested in the quality assurance of ad-hoc services provided within ubiquitous environments have been interested in situation-awareness and tied it to the network connectivity and status withins a sensor network. The works of [Kim et al. 2009] proposed a Situation-Aware Technology and distinguished between situation-awareness and context-sensitivity as follows:

- ***Situation awareness:*** is the capability of a device to capture and analyze the relationship among multiple contexts and actions over a period of time in order to be able to take actions automatically and timely depending on situations.
- ***Context-sensitivity:*** Context sensitivity is the capability of a device to detect its current context and changes in contextual data.

In access control, a situation is viewed as an "un-expected exceptional event or an urgent event taking place". It is considered as an important factor that can influence and alter decision-making and access priviliges [Povey, 2000], [Rissanen et al., 2004].

2.3.3.2 Introducing Flexible Access Control & « Break-Glass »

Respecting the rigidity of access control decisions is strongly required to ensure the security and integrity of an information system. However, it is almost impossible to specify and predict all the situations and scenarios of required access.

Thus, providing a flexiblity of access is essential especially in critical situations (medical emergency, fire, technical problems on a plane, etc.) where an access denial may not only risk the quality of the system but also the lives of system users.

The need of situation-awareness in access control was first expressed by [Povey, 2000], who highlighted the importance of providing a sort of ***optimistic security*** scheme that would relax access rules in order to enable users to meet exceptional circumstances (disasters, medical emergencies or time-critical events).

The works of [Rissanen et al., 2004] have also highlighted the importance of providing a ***flexible security*** system that would offer more than yes or no answers and that would not rely on predefined solutions in meeting un-anticipated access demands. The proposed system offered the possibility to override security policies and to search for an authority resolution that is an automatic process that finds an access policy authority that can approuve an override.

The previously mentioned works have tried to provide means to maintain the system's integrity and ensure tracability like audit, administrative checks and by making sure that users employ this flexibility based on a conscious decision and know that they will be accountable for any illegitimate access.

This flexible security was next called ***« Break-Glass »*** solution and was adopted as a standard within healthcare systems [Break-Glass, 2004] and applied in many other application domains. The solution helps users to confront emergent situations by granting them access to unauthorized needed resources.

Typically, a strategy for implementing the «Break-Glass» is comprised of the following steps:

1. ***Pre-staging «Break-Glass» accounts:*** Emergency accounts are created in advance to allow careful thought about the access control policies and audit trails associated with them.

2. ***Distributing pre-staged accounts*** needs to be carefully managed to provide timely access when needed: Breakglass requires the emergency accounts be made available in an appropriate and reasonable manner. The account details may be provided on media such as a printed page, a magnetic-stripe card, a smart card or a token.
3. ***Monitoring the use of «Break-Glass» accounts***: The use of emergency accounts needs to be carefully monitored. Audit mechanisms should be used and a procedure defined to examine the security audit trails on a regular basis to identify any use of the emergency accounts. In addition, systems can alert the security administrator in the event an emergency account is activated.
4. ***Cleaning up after «Break-Glass»:*** A procedure should be established to clean up after an emergency account has been used.

Finally the traditional «Break-Glass» solutions store such emergency accounts either completely electronically or printed on paper and, e.g., stored in a glass cabinet.

Despite its importance and all the protective steps accompanying it, the «Break-Glass» is considered as extreme solution that enables users to perform illegitimate intrusions and unjustified access attempts. Therefore and as we will show next, various research works have focused on either controlling the usage of «Break-Glass» by improving its modeling and on facilitating it's integration within the conventional access control models in order to confront the privacy and integrity threats or on proposing other less risky situation-aware access control solutions.

2.3.3.3 Proposals to situation-aware access control model with minimal risks

An analytical study of the different research works conducted in the field of situation-aware access control modeling has led us to distinguish three main directives.

The first group has applied situation-awareness by using the «Break-Glass» solution while trying to minimize the risk margine. Those proposals depend on ***applying access flexibility based on predefined rule-based solutions***. The implementation was either based on a requirement engineering analysis that helps to predict the different possibilities that trigger an emergency or y trying to integrate mechanisms to fortify the auditing that pursues the «Break-Glass».

The works of [Ferreira et al., 2006] proposed an implementation that aims to control the risks of unjustified illegitimate access when using the « Break-Glass » solution in healthcare systems. The objective of the proposed solution was to ensure that users, benefiting from the « Break-Glass » solution, take ***maximum freedom*** in accessing ressorces to confront urgent situations and ***maximum responsibility*** for their actions. This was achieved through adding a non-repudiation mechanism that ensures the accountability of the person using the «Break-Glass» solution and to guarantee that he'll take responsibility for any unjustified action.

A more advanced step followed in [Ferreira et al., 2009] where the «Break-Glass» solution was included within the RBAC model in a secure and transparent way. The introduced model, called BTG-RBAC (Break The Glass - Role Based Access Control), aims to: (i) surpass the rigidity of access decisions provided by the RBAC model, (ii) provide users with more flexible access, (iii) ensuring non-repudiation by increasing the user's awareness while using the «Break-Glass» solution.

The works of [Brucker and Petritsch, 2009] went towards integrating the «Break-Glass» solution in a generic model-driven mechanism that applies access control flexibility on a fine-grained policy-based manner. Thus, the flexibility specification is performed at the permission level instead of the role or subject level. To minimize the risks of security violation, the model proposes to restrict the use of «Break-Glass» solution at certain emergency levels.

The second group of research works has proposed to minimize the risks of violation by employing ***monitored or assisted access control flexibility*** where the user confronting an urgent situation will obtain access to the needed (unauthorized) resources from a higher authority. This authority might provide him with the needed services/information or can help him in finding another subject that has access to the needed resources and that can delegate access permissions to him.

The works of [Keppler et al., 2006] presented a situation aware general sharing control framework for expressing access policies that react in the case of access denials returned in mission-critical situations. Thus, before denying an unauthorized request, the system determines other possible actions to take, these actions include: (i) redirecting the request to other authorized principals that are related to the original requestor, (ii) sending the request to an overriding authority that can grant him access, and (iii) sending both the request and the requested data to a trusted intermediary. The solution is implemented using a formal logic based sharing policy language.

The works of [Catarci et al., 2008] proposed a pervasive platform that meets and manages the complexity of access denials to requests launched during a crisis or an emergency. The authors provide accessibility through a collaborative system that connects the various subjects dealing with the situation. The collaboration can engage subjects located on site and external authorities that may interfere to provide flexible access to external resources that are not authorized.

The research works of [maymi et al., 2008] proposed Ancile: a system that proposes to create a pervasive environment composed of the decentralized system resources and an ad-hoc location-based peer-to-peer network for sharing real-time information to handle a critical situation within different applications (fire-fighting, military operations, etc.). The system proposes using the pervasive environment to find authorized subjects that can help a user in accessing an unauthorized access demand.

The Works of [Kawagoe et al., 2011] highlighted the importance of providing an efficient situation-aware emergency system that can share and exchange patient's personal information when demanded in medical treatments and disaster situations. The authors presented the STRAC (Situation, Team and Role based Access Control), the model realizes flexible access control management in emergency cases through predefined authorizations and enables access control of user personal information with consideration of context changes.

Finally, the third direction was interested in ***applying more autonomy in decision-making and accompanied it with risk management***. Providing autonomy enables producing ad-hoc access control flexibility.

The works of [Cheng et al., 2007] have proposed a multi-level access control model based on fuzzy logic. The model aims to provide flexible access control solutions in a system that is dedicated to manage access permissions while sharing and disseminating large amounts of sensitive information that are collected on a real time

basis from different distributed data sources. The application domain is related to national security where the rigidity of access decisions is not acceptable and where violating security shields or data privacy may be less risky than receiving a denial of access at a critical mission.

Instead of relying on providing static binary "allow/deny" responses, the authors offer a dynamic, multi-decision access control model that relies on quantified risk estimates and risk tolerance. The use of fuzzy logic provided an intermediate zone (between access permission and access denial) where access permissions are made and accompanied by a calculation of risk.

Quantified risk = (probability of damage) x (value of damage)

2.3.3.4 Analysis/ Survey

In this section, we have performed a review of the various research works that were interested in integrating situation-awareness within access control modeling in order to provide flexible access decisions in critical situations.

The «Break-Glass» option was one of the main proposals to confront urgent situations thus, many research works were directed towards improving it's integration within access solutions with minimal amount of risk. Other solutions proposed attaining flexible access permissions through assisted help from other authorized subjects or by employing ad hoc decision-making.

Despite the security threats or the risks reveiled when applying flexible access control solutions, ensuring real-time accessibility to data sources at emergent and unexpected situations is an important characteristic that is critical to ensure the quality of service of an access control system dedicated for any vital application.

As we illustrate in fig. 2.8, the flexibility level offered by the access control model might be directly proportional to the risk of violating the system's security:

- The more the access control flexibility is performed on a rule-based, predefined or assisted manner, the less the violation risks are introduced.
- The more the flexibility is provided in an automatic and ad hoc manner, the more the risk level is elevated.

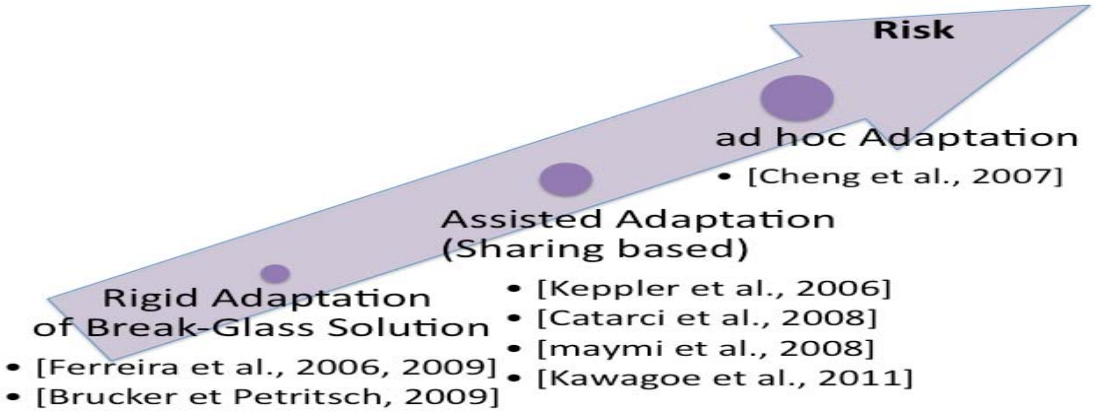


Figure 2.8: The directly proportional relation between access control flexibility and violation risks

Finally, the risk has been usually tolerated by the audit and accountability methods and other calculation means provided by the previously cited solutions.

2.4 Conclusion

In this chapter, we have conducted a study about modeling access control within pervasive information systems. In order to cover the needs of these systems we have started by introducing the pervasive computing paradigm and presented its definition, main characteristics, objectives, components and features. Next, we have identified the main security challenges ans the different characteristics needed for modeling access control within Pervasive information systems.

Our study has revealed that a pervasive access control model should provide adaptive decision-making that would meet the users accessibility needs and would offer him access at anytime, anywhere and anyhow.

After studying the different research works conducted within the field in order to achieve a pervasive access control model, we have noticed that the presented proposals were directed towards adapting the classical decision making procedure within the basic access control models in order to take two main factors into consideration which are: context-awareness and situation-awareness.

Finally, this chapter has provided a state of the art about the different works interested in modeling access control, this step stands in the systems design phase within software engineering lifecycle, the next chapter will be dedicated to introduce the main technologies that can elevate and insure the implementation of such models at the application domain where organizations are achieving interoperable accessibility through service-oriented architectures.

Chapter 3:
Implementing Access Control for Service-Oriented Architectures

3.1 Introduction

The introduction of Service-Oriented Architectures has provided high levels of interoperability and high means of adaptability. Nevertheless, distributed service-based systems should rely on robust, flexible and scalable access control mechanisms in order to achieve efficient security.

Following the evolution of access control management, we find that the RBAC model has succeeded in resolving the problem of administrating access permissions within distributed systems by providing centralized permission assignment through roles (see chapter 1).

Next, with the evolution of service-oriented architectures and web services, the problem of access management became more complicated since the distribution does not only touch the resources to be accessed but also as the policies managing these resources which are distributed, dynamic and issued by different authorities (administrating services). To resolve this problem, the XACML standard was introduced by [OASIS, 2003].

Finally, the advantages of the RBAC model, it's power to map permissions assignment in a way that matches the organizational structure and it's common usage within the different services and application domains have motivated OASIS to adapt the RBAC scheme and generate an XACML profile in order to provide a compatible role based and service oriented access control management

In this chapter, we'll introduce the main requirements for managing access control within service-oriented architectures then we'll introduce the notion of Identity Based Access Control and present the XACML standard and the XACML RBAC profile.

3.2 Access Control Requirements within Service-oriented Systems

In order to design appropriate authorization infrastructures within Service-Oriented computing paradigms, the following characteristics have to be taken into consideration [Wimmer, 2007]:

- ***Autonomy of Authorization***:
  Web services, as the predominant technology for realizing service-oriented architectures, are fine-grained, modular software components that independently enforce access control. Thus, in contrast to monolithic architectures, no single point of administration is given in service-oriented IT infrastructures.

- ***Multi-layered Authorization***:
  Service-oriented architectures can be used to integrate existing enterprise applications and legacy systems by use of standardized service interfaces. Services can on their part be combined to realize higher order services, thus, leading to composite applications. Regarding service compositions, access control is enforced in a multilayered manner.

- ***Coalition-based Access Control***:
  Via service compositions, intra- and inter-organizational value creation chains can be realized. In order to enable ***inter-organizational cooperations***, the authorization framework needs to support the delegation of access rights across administrative boundaries and the evaluation of authorizations within collaboration networks. Through service invocations, users (e.g. customers, suppliers, and partners) can directly access business relevant data from outside the organization. This is what [Lord, 2002] refers to as "***dis-inter-mediation***" as requests are directly passed to the company's information system instead of being mediated by employees who supervise the execution.

- ***Demand for Scalability***:
  The employed access control models and mechanisms need to be scalable. In particular, identity-based authorization is not meaningful in dynamic coalition environments. Instead, authorizations should be inferred based on the requesters' attributes.

3.3 Attribute-based Access Control

The evolution of service-oriented computing have imposed new constraints to access management. The resources and services of these open systems are complex and heterogeneous, the users are changeable and members of multiple organizations.

In order of meet the needs of distributed systems, the traditional closed and inflexible Identity Based Access Control (IBAC) models [Harrison et al., 1976] were substituted by new models like Role-based Access Control Models (RBAC) [Ferraiolo et al., 1992].

Next, a more appropriate approach called Attribute Based Access Control models (ABAC) was introduced to fit the requirements of service-oriented computing [OASIS, 2003],[Wang et al., 2004a]. Unlike IBAC and RBAC, the model builds access decisions on properties (attributes) of the requester and of the resource. Basing authorization on attributes of the resource/service requester provides flexibility and scalability that is essential in the context of large distributed and service-oriented systems, where subjects are identified by their characteristics.

XACML (eXtensible Access Control Markup Language) is a standard designed for access control within service-oriented architectures presented by [OASIS, 2003]. XACML policies are ABAC-Based; they allow decisions making and permission assignment based upon any security relevant characteristics (known as attributes) of requestors, actions, resources, and environment.

In the following we will first take a look at this standard and it's components in more detail then, we'll expose the XACML RBAC profile: an extended version of the standard that adopted the RBAC model.

3.3.1 The eXtensible Access Control Markup Language XACML

XACML (extensible Access Control Markup Language) [OASIS, 2003], [OASIS, 2005a] is an XML based standard that describes access control policies to allow the attribution of user privileges on system sources. The standard provides a system for authentication and authorization taking into account various factors related to the user's context.

XACML is a simple protocol that enables centralized management for distributed data sources in a way that crosses the organisation's boundaries and provides fine-grained authorizations.

XACML describes both a policy language and an access control decision request/response language:

- The ***policy language*** is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc.
- The ***request/response language*** enables users to form a ***request*** to ask whether or not a given action is allowed to be performed on a given resource at a given context, and to interpret the result. The ***response*** includes one of four values: ***Permit, Deny, Indeterminate*** (an error occurred or some required value was missing, so a decision cannot be made) ***or Not Applicable*** (the request can't be answered by this service).

There are many existing proprietary and application-specific languages for doing this kind of thing but XACML has several points in its favor:

- ***It's standard***. By using a standard language, you're using something that has been reviewed by a large community of experts and users, you don't need to roll your own system each time, and you don't need to think about all the tricky issues involved in designing a new language. Plus, as XACML becomes more widely deployed, it will be easier to interoperate with other applications using the same standard language.
- ***It's generic***. This means that rather than trying to provide access control for a particular environment or a specific kind of resource, it can be used in any environment. One policy can be written which can then be used by many different kinds of applications, and when one common language is used, policy management becomes much easier.
- ***It's distributed***. This means that a policy can be written which in turn refers to other policies kept in arbitrary locations. The result is that rather than having to manage a single monolithic policy, different people or groups can manage sub-pieces of policies as appropriate, and XACML knows how to correctly combine the results from these different policies into one decision.
- ***It's powerful***. While there are many ways the base language can be extended, many environments will not need to do so. The standard language already supports a wide variety of data types, functions, and rules about combining the results of different policies. In addition to this, there are already standards groups working on extensions and profiles that will hook XACML into other standards like SAML (Security Assertion Markup

Language) and LDAP (Lightweight Directory Access Protocol), which will increase the number of ways that XACML can be used.

To give a better idea of how all these aspects fit together, what follows is a discussion of XACML policy, which will demonstrate many of the standard features of the language. Note that XACML is a rich language, so only some of its features are shown here and for more information about it's features, the OASIS specification can be consulted.

3.3.1.1 XACML Architecture

The XACML specification provides an architecture that performs access management in a structured way where a group of dedicated components interact with each others to acquire an access decision, see fig. 3.1.

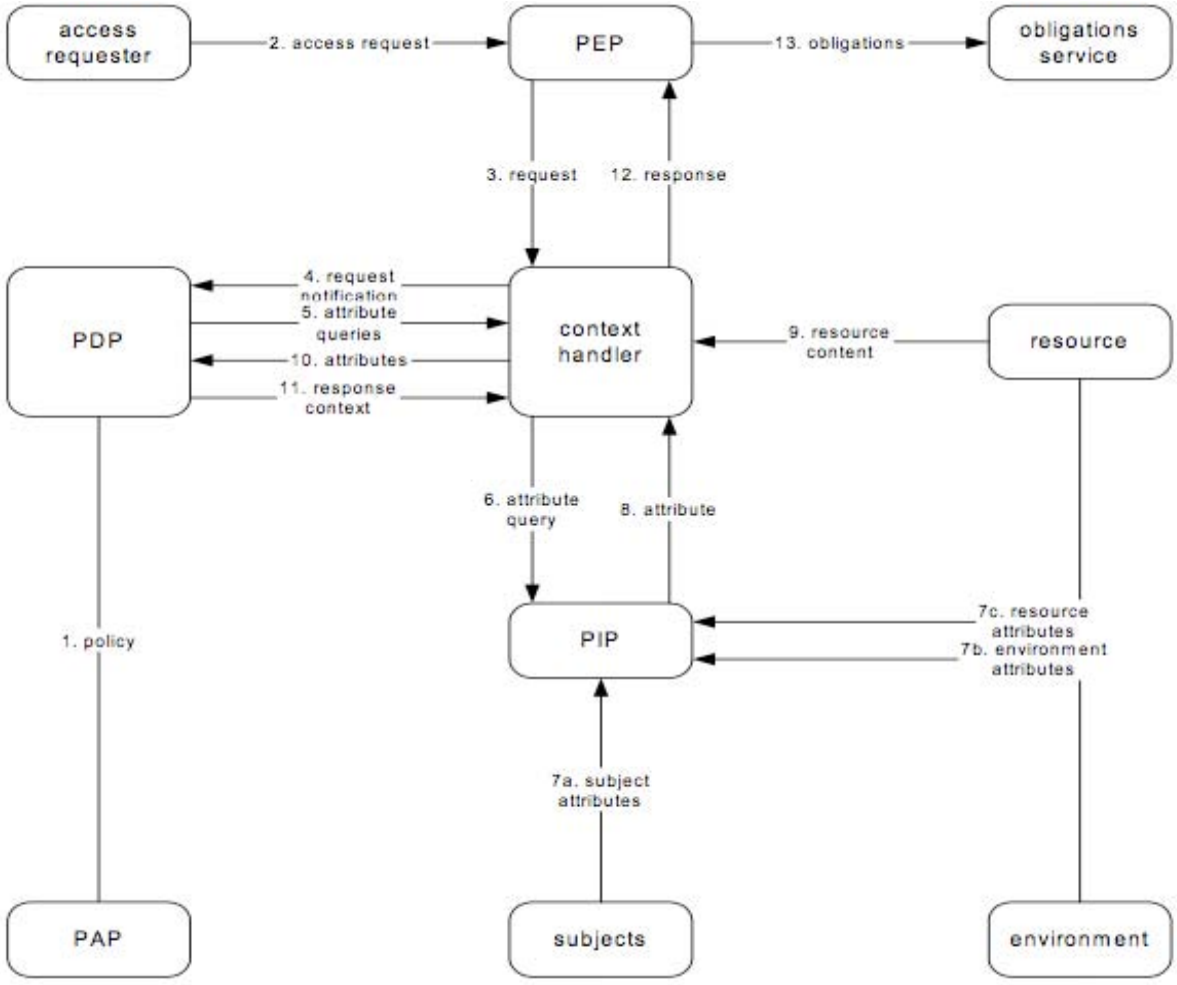


Figure 3.1: XACML Data Flow Chart, OASIS

(1) The Policies are sent and communicated from the ***Policy Administration Point*** (PAP) to the ***Policy Decision Point*** (PDP).

(2) A user sends a request to a ***Policy Enforcement Point*** (PEP).

(3) The PEP forms a simple request that contains subject, resource and action attributes.

(4) The ***Context Handler*** sends this request to a ***Policy Decision Point*** (PDP).

(5) The PDP looks at the request and finds some policy or policy sets that apply to the request from a policy store and asks for the more attributes in order to make a decision.

(6 - 9) The PDP queries the ***Policy Information Point*** (PIP) service to retrieve the attribute values related to the subject, the resource, or the environment and gets a response with desired attribute values.

(10) The attributes are sent back to the PDP.

(11) The PDP compares the attributes in the request against the attributes contained in the policy rules and returns an answer about whether access should be granted or not.

(12) That answer is returned to the PEP, which can then allow or deny access to the requester.

(13) The answer can be accompanied by an ***obligation*** to be imposed while applying the decision.

3.3.1.2 XACML Policy Language

After describing the sequence of the communication that takes place between the different subcomponents of the XACML standard, we'll illustrate the relation between the different XACML documents created within the previously mentioned components and we'll include a detailed description for the structure and content, see fig. 3.2.

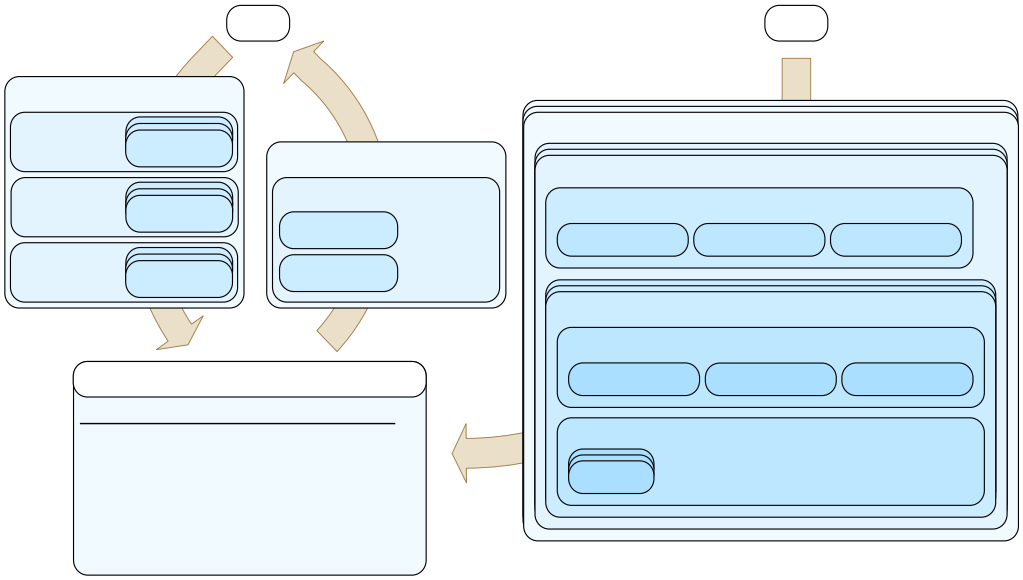


Figure 3.2: XACML Documents and their Relationships

1. **Top-Level Constructs: Policy and PolicySet**

At the root of all XACML policies is a Policy or a PolicySet. A **PolicySet** is a container that can hold other Policies or PolicySets, as well as references to policies found in remote locations. A **Policy** represents a single access control policy, expressed through a set of Rules. Each XACML policy document contains exactly one Policy or PolicySet root XML tag.

Because a Policy or PolicySet may contain ***multiple policies or Rules***, each of which may evaluate to different access control decisions, XACML needs some way of reconciling the decisions each makes. This is done through a collection of **Combining**

Algorithms. Each algorithm represents a different way of combining multiple decisions into a single decision.

There are ***Policy Combining Algorithms*** (used by PolicySet) and ***Rule Combining Algorithms*** (used by Policy). An example of these is the Deny Overrides Algorithm, which says that no matter what, if any evaluation returns Deny, or no evaluation permits, then the final result is also Deny. These Combining Algorithms are used to build up increasingly complex policies, and while there are seven standard algorithms, you can build your own to suit your needs.

2. Targets and Rules

Part of what an XACML PDP needs to do is find a policy that applies to a given request. To do this, XACML provides another feature called a ***Target***. A Target is basically a set of simplified conditions for the Subject, Resource and Action that must be met for a PolicySet, Policy or Rule to apply to a given request. These use boolean functions (explained more in the next section) to compare values found in a request with those included in the Target.

If all the conditions of a Target are met, then its associated PolicySet, Policy, or Rule applies to the request. In addition to being a way to check applicability, Target information also provides a way to index policies, which is useful if you need to store many policies and then quickly sift through them to find which ones apply. For instance, a Policy might contain a Target that only applies to requests on a specific service. When a request to access that service arrives, the PDP will know where to look for policies that might apply to this request because the policies are indexed based on their Target constraints. Note that a Target may also specify that it applies to any request.

Once a Policy has been found and verified to apply to a request, its Rules are evaluated. A policy can have any number of ***Rules*** that contain the core logic of an XACML policy. The heart of most Rules is a ***Condition***, which is a boolean function. If the Condition evaluates to true, then the ***Rule's Effect*** (a value of **Permit** or **Deny** that is associated with successful evaluation of the Rule) is returned. Evaluation of a Condition can also result in an error (**Indeterminate**) or discovery that the Condition doesn't apply to the request (**NotApplicable**). A Condition can be quite complex, built from an arbitrary nesting of non-boolean functions and attributes.

3. Attributes, Attribute Values, and Functions

The currency that XACML deals in is attributes. ***Attributes*** are named values of known types that may include an issuer identifier or an issue date and time. Specifically, attributes are characteristics of the Subject, Resource, Action, or Environment in which the access request is made. A user's name, their security clearance, the file they want to access, and the time of day are all attribute values. When a request is sent from a PEP to a PDP, that request is formed almost exclusively of attributes, and they will be compared to attribute values in a policy to make the access decisions.

A Policy resolves attribute values from a request or from some other source through two mechanisms: the ***AttributeDesignator*** and the ***AttributeSelector***. An ***AttributeDesignator*** lets the policy specify an attribute with a given name and type, and optionally an issuer as well, and then the PDP will look for that value in the request, or elsewhere if no matching values can be found in the request.

There are four kinds of designators, one for each of the types of attributes in a request: ***Subject, Resource, Action,*** and ***Environment***. Because Subject attributes can be broken into different categories, SubjectAttributeDesignators can also specify a category to look in. ***AttributeSelectors*** allow a policy to look for attribute values through an XPath query. A data type and an XPath expression are provided, and these can be used to resolve some set of values either in the request document or elsewhere.

Both the ***AttributeDesignator*** and the ***AttributeSelector*** can return multiple values (since there might be multiple matches in a request or elsewhere), so XACML provides a special attribute type called a Bag. ***Bags*** are unordered collections that allow duplicates, and are always what designators and selectors return, even if only one value was matched. In the case that no matches were made, an empty bag is returned, although a designator or selector may set a flag that causes an error instead in this case.

Once some Bag of attribute values has been retrieved, they need to be compared in some way to expected values to make access decisions. This is done though a powerful system of functions. ***Functions*** can work on any combination of attribute values, and can return any kind of attribute value supported in the system. Functions can also be nested, so you can have functions that operate on the output of other functions, and this hierarchy can be arbitrarily complex. Custom functions can be written to provide an ever richer language for expressing access conditions.

One thing to note when building these hierarchies of functions is that most functions are defined as working on specific types (like strings or integers) while designators and selectors always return Bags of values. To handle this, XACML defines a collection of standard functions of the form [type]-***one-and-only***, which accept a bag of values of the specified type and return the single value if there is exactly one item in the bag, or an error if there are zero or multiple values in the bag. This is one of the most common functions that you will see in a Condition. [type]-one-and-only functions are not needed in Targets, however, since the PDP automatically applies the matching function to each element of a bag.

3.3.1.3 Policy Example

In this section, we present a simple example Policy that uses the features discussed above. Its Target says that the Policy only applies to requests for the server called "SampleServer". The Policy has a Rule with a Target that requires an action of "login" and a Condition that applies only if the Subject is trying to log in between 8am and 6pm. Note that this example can be extended to include other Rules for different actions. If the first Rule provided here doesn't apply, then a default Rule is used that always returns Deny (Rules are evaluated in order).

```
<Policy PolicyId="SamplePolicy"
   RuleCombiningAlgId = "urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-
overrides">
   <Target>
      <Subjects>          <AnySubject/>       </Subjects>
      <Resources>
         <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
             SampleServer              </AttributeValue>
            <ResourceAttributeDesignator
                DataType="http://www.w3.org/2001/XMLSchema#string"
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
</ResourceMatch>  </Resources>
      <Actions>          <AnyAction/>       </Actions>
   </Target>
   <Rule RuleId="LoginRule" Effect="Permit">
      <Target>
         <Subjects>             <AnySubject/>          </Subjects>
         <Resources>              <AnyResource/>          </Resources>
         <Actions>
            <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                  login</AttributeValue>
               <ActionAttributeDesignator
                   DataType="http://www.w3.org/2001/XMLSchema#string"
                  AttributeId="ServerAction"/>
            </ActionMatch>          </Actions>       </Target>
      <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-
equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector
                  DataType="http://www.w3.org/2001/XMLSchema#time"
                  AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
            </Apply>
            <AttributeValue ataType="http://www.w3.org/2001/XMLSchema#time">
             08:00:00 </AttributeValue>          </Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-
equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
               <EnvironmentAttributeSelector
                  DataType="http://www.w3.org/2001/XMLSchema#time"
                  AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
</Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">
             18:00:00</AttributeValue>          </Apply>       </Condition>    </Rule>
   <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

Figure 3.3: An example of an XACML Policy

3.3.2 RBAC Profile of XACML

The widespread usage of the RBAC model within the different organizations and services has motivated the proposition of an XACML profile that adapts the RBAC sheme. The XACML RBAC profile was presented by [OASIS, 2005b], it presents a method of assigning role attributes, specifying core and hierarchical roles and their permissions to enable creating access authorizations.

This profile uses four XACML policy/policy set types to implement RBAC:

- ***Permission Policy Set or PPS***: As shown in Fig. 3.4 [Jin, 2006], PPS is a collection of permissions associated with a role. It is a <PolicySet > that contains a set of polices which defines permissions associated with the given role directly, and a set of PPS references which associate with other roles that are junior to the given role. In order to support the hierarchy model, it required that the <Target> element of a PPS if present, must not limit the subjects to which the <PolicySet> is applicable.

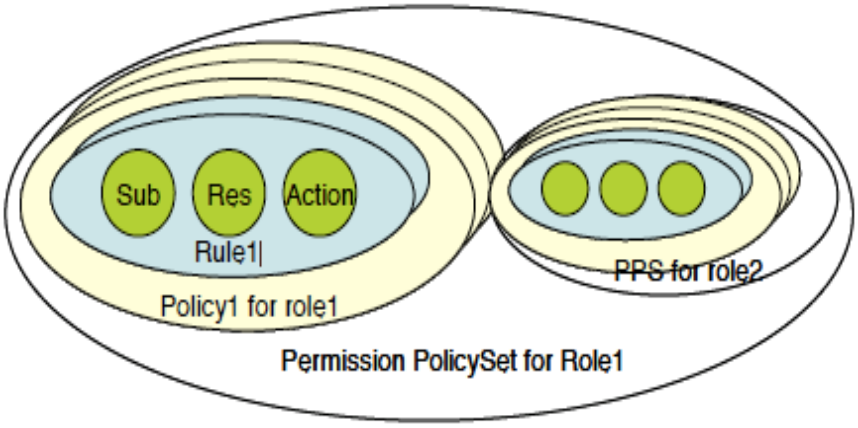


Figure 3.4: Permission Policy Set (PPS) - XACML RBAC

- ***Role Policy Set or RPS***: As shown in Figure 3.5 [Jin, 2006], an RPS connects a role to the corresponding PPS that contains actual permissions associated to the given role. Each RPS contains a <Target> element indicating the applicable role and a reference to PPS.

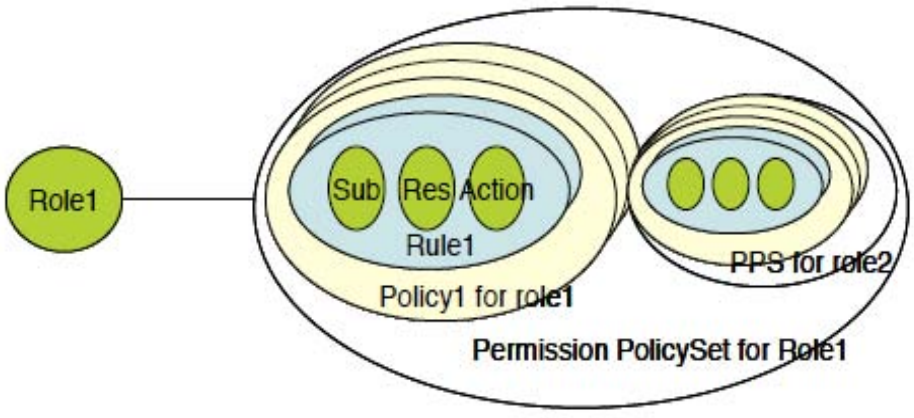


Figure 3.5: Role Policy Set (PPS) - XACML RBAC

- ***Role Assignment Policy:*** defines which roles are assigned to which subjects.
- ***HasPrivilegesOfRole Policy:*** supports queries to determine if a user has a certain role assigned.

Finally, In order to express a role hierarchy, the PPS of one role references the PPS of another role, thereby inheriting all the permissions of that referenced role.

3.4 Conclusion

In this chapter, we made an overview of the technologies introduced to implement and model access control within service-oriented computing architectures. We have concentrated our study over the XACML standard due to its ability to perform centralized decision making within multi-distributed environments, where the distribution envolves the requested resources and the policies administrating these resources (that might be generated by different authorities from different services).

The main functionalities of XACML can be summarized as follows [Ardagna et al., 2007]:

- ***Policy combination:*** XACML provides a method for combining policies that are independently specified. Different entities can then define their policies on the same resource. When an access request on that resource is submitted, the system has to take into consideration all of these policies.
- ***Combining algorithms:*** Since XACML supports the definition of policies independently specified, there is the need for a method for reconciling such policies when their evaluation is contradictory. XACML supports different combining algorithms, each representing a way of combining multiple decisions into a single decision.
- ***Attribute-based restrictions:*** XACML supports the definition of policies based on properties (attributes) associated with subjects and resources other than their identities. This allows the definition of powerful policies based on generic properties associated with subjects (e.g., name, address, occupation) and resources. XACML includes some built-in operators for comparing attribute values and provides a method of adding nonstandard functions.
- ***Multiple subjects:*** XACML allows the definition of more than one subject relevant to a decision request.
- ***Policy distribution:*** Policies can be defined by different parties and enforced at different enforcement points. Also, XACML allows one policy to contain or refer to another.
- ***Implementation independence:*** XACML provides an abstraction layer that isolates the policy-writer from the implementation details. This means that different implementations should operate in a consistent way, regardless of the implementation itself.
- ***Obligations:*** XACML provides a method for specifying some actions, called obligations, that must be fulfilled in conjunction with the policy enforcement.

Finally, we have also exposed the XACML RBAC profile that resulted from the need of a service-oriented standard that maps the organizational permission assignment structure that is usually performed through RBAC.

Part II:

A Contribution to the Extension of Access Control Models for Pervasive Information Systems

Chapter 4:

A Pervasive Situation-Aware Role-Based Access Control Model

4.1 Introduction

Accessibility is considered as a key feature for pervasive systems, consequently modeling access control within pervasive information systems is considered as a challenging mission since it can be viewed from different angles. Looking at it from a user's point of view, access control can be considered as a barrier to accessibility since it might not provide users with access to a desired resource at anytime, anyhow and anyway. Meanwhile, if we take the system administrator's point of view, access control is considered extremely important to ensure secure interactions and maintain the system's integrity while offering accessibility within open environments.

In order to meet the pervasive vision, transparent but secure access needs to be provided within mobile, intelligent and dynamic environments. As a result, pervasive access control modelling went towards tying contextual constraints (time, location, user machine, connexion channel, etc.) to the decision-making process. This feature does not only enable better access management to system sources at anytime, anywhere, anyhow but it also meets the dynamic accessibility needs of mobile users.

Taking a close look to access control needs within different application domains in the pervasive age, we find that permission assignment at the enterprise level has to follow the evolution. Knowing that the distributed resource management is usually performed using the **RBAC** (Role Based Access Control) model, the emergence of the new pervasive paradigm has opened new needs for expressing context-awareness within this model. Accordingly, and as we have shown in chapter 2, different proposals have introduced new RBAC models that take the contextual constraints while performing access decisions.

Despite the richness of models proposed, context-aware decision-making remains to form a challenging research problem since the interpretation of the user's context can be differently expressed and defined according to the application domain, to the designer's objective or to many existing techniques employed for context acquisition, modelling and interpretation.

Furthermore, in analyzing the related works, we notice that the dynamicity of the pervasive environment will eventually be reflected on permission assignments. Thus, there's a need to consider the consequences of having volatile and highly changing assignments and eventually some permission denials on the stability, reliability and usability of the system especially within real-time systems.

The dynamicity of permission assignment within pervasive real-time systems is not only related to changes within the contextual dimension but to many other aspects since the accessibility that is traditionally ruled by strict policies to protect data privacy may:

1. face exceptions in particular situations (e.g. in an emergency case),

2. evolve over time (e.g. depending on the patient's treatment) and,
3. be subject to provisional authorizations.

That's why we highlight the importance of supporting adaptive behaviors while modeling access control for pervasive systems especially when dealing with critical real-time systems.

As a result, we highlight the fact that modeling pervasive access needs to consider adopting adaptive behaviour not only to deal with the contextual dynamicity but also to consider the user's situation when demanding access, which might be a critical or life-threatening case (fire, toxic gaz leakage, medical emergency, etc.).

An access denial taking place at an urgent situation could be considered as a barrier to the mission accomplishment and an obstacle standing in the way of a transparent interaction with the system. Accordingly many reseach works were interested in adapting access control modeling to take into consideration situation-aware decision-making and to provide flexible solutions that would enable users to surpass security shields when needed in order to confront a situation (see chapter 2).

Our interest in modeling access control for pervasive systems and the results that we attained from studying the pros and cons of the different works conducted in the field has directed us to look at the problem from another perspective and enabled us to discover a new un-treated zone.

We argue that in some situations, there's a need for confronting access denials by proposing ad hoc and innovative alternative-based solutions that already exist rather than trying to depass the system's security. This need arises in scenarios where even employing a «Break-Glass» solution wont be of any help to confront a situation (e.g. enabling a user to a access an area that is contaminated in a toxic gaz or nuclear leakage won't be an effective solution during an accident). Thus, we aim to provide users with an adaptive model that offers alternative solutions that might already be authorized to them and that are beneficial to confront the situation.

The adaptation that we conceive aims to offer alternative solutions rather than binary (yes/ no) solutions. The adaptive solutions provided by our proposal are not only dedicated for extreme situations but also for various situation sensitivity levels where the denial of access might be related to simpler causes like the dynamicity of the user's context within a pervasive environment or to difficulties in obtaining the right credentials to access a certain service or network or due to misfunctionalities of the server, machine or network providing access to the needed resources, etc.

In this chapter, we will present our vision through PS-RBAC: a Pervasive Situation-aware Role Based Access Control model that aims to meet the access control requirements within pervasive environments. Our model enables:

- Context-aware role assignment,
- Context-aware permission assignment,
- Situation-aware permission assignment that offers alternative resources without violating the security shields.

The situation-awareness solution provided within the proposed model forms an intermediate solution that offers a moderate flexibility level between offering access denials and the extreme solutions that are overheaded with risks. At the long run, our

model can also be flexible enough to integrate some extreme solutions when needed (if the situation's sensitivity level is high).

4.2 Model Overview

Our proposed extension for the RBAC model aims at constructing a sort of flexible authorizations and adaptive permissions that meet the needs of pervasive users. Flexibility is introduced through the proposal of adaptive permissions that enable access to alternative similar resources. The adaptation is performed according to the user's context and the sensitivity of his situation [Al Kukhun and Sèdes, 2009a], [Al Kukhun et al., 2012a].

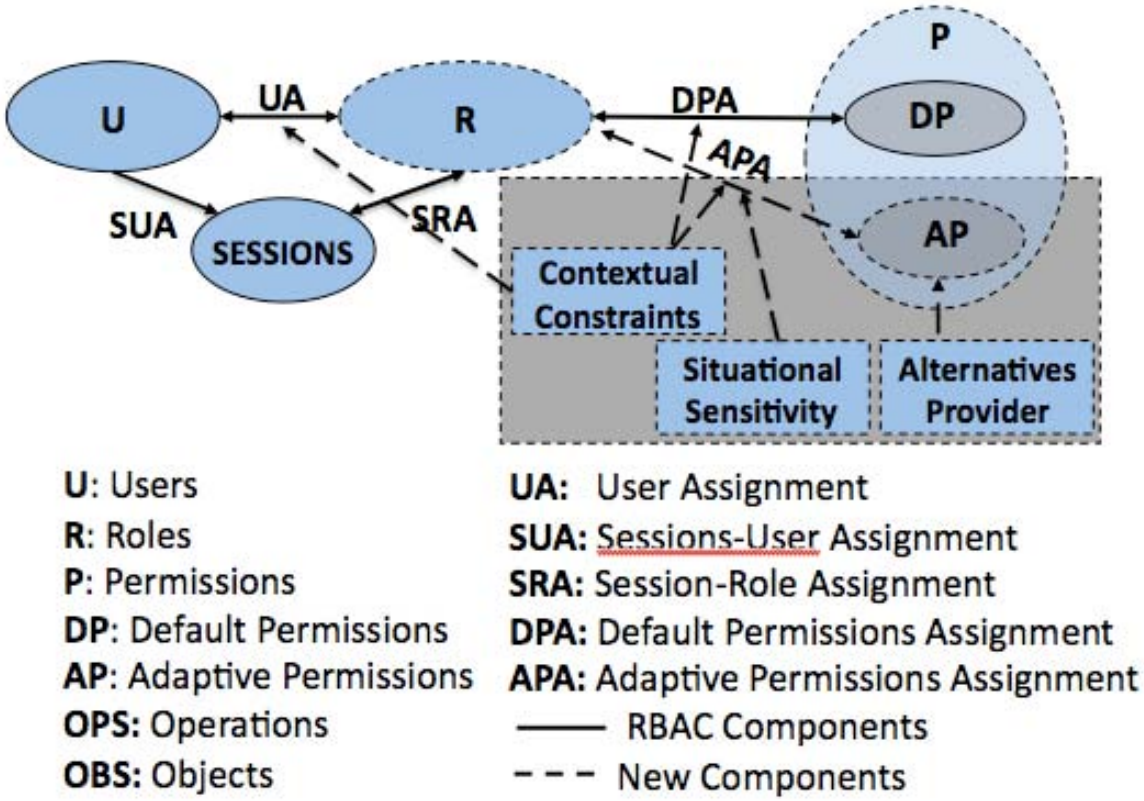


Figure 4.1: PS-RBAC Pervasive Situation-aware Role Based Access Control

In our model, ***Users*** are assigned to ***Roles*** in order to acquire different permissions. These permissions are divided into 2 types: ***Default and Adaptive Permissions***. The type of permission assignment depends on the contextual and situational attributes of the user's access request.

The Adaptive Permissions AP proposed by our model are permissions that grant access to alternative ressources in cases where users are confronted with an access denial during an important situation. The proposed ressources are provided by an ***Alternative Ressource Provider Component*** that can employ several techniques to provide users with adaptive and authorized solutions.

In the following, we describe the basic components of the PS-RBAC model and we provide the formal definitions that explain the relations between them, see fig. 4.1.

Users U: Users in our model are the people who interact with the system. A user could also be extended to represent a computer process, a machine, a web service (in a service-oriented application) or any active entity of the system acting behalf of the user.

Roles R: A role is a job function or a position within the context of an organization that empowers a user to obtain some authorities and execute operations. Roles are assigned to users according to some predefined semantics and contexts.

The PS-RBAC defines a mapping between users and roles as a set of ***sessions (SESSIONS)***. When a user logs into the system, the user may be mapped to one or several roles through a session. This allows a user to activate only the minimum subset of roles they need in order to perform a specific task, and hence by using sessions, PS-RBAC supports the least privilege principle.

The mapping is established by two functions:

Session_User Assignment SUA: defines the one-to-many mapping relation between the user and the number of sessions that can be assigned to him.

SUA SESSIONS × USERS

Session_Role Assignment SRA: defines the many-to-many mapping relation between the number of assigned sessions and the roles connected to them.

SRA SESSIONS × ROLES

Contextual Constraints CC: In order to adapt with the mobility of pervasive users and the dynamicity of the user's context within pervasive environments, the contextual attributes of the user are taken into consideration within the proposed model.

As we will show next, the Contextual Constraints influence the assignment of: Roles, Default Permissions and Adaptive Permissions.

The ***User Assignment UA*** in the PS-RBAC model follows a many-to-many relationship where a user (person, computer process, machine, etc.) can be assigned to one or several roles during a session and a single role can be assigned to one or multiple users at the same time.

In a pervasive environment, this assignment is connected to the user's Contextual Constraints CC that are extracted at the system sign-in time.

UA ⊆ Users × Roles x CC

Permissions P: Permissions are the approvals that enable a type of access to objects or system sources. Permissions establish a relation between operations and objects; where they specify the operation(s) OPS (read, write, update, etc.) that can be performed on the objects OBS or system sources (documents, computer processes, machines, etc).

Objects OBS are data or resources to be accessed.

Operations OPS are processes that execute some function on behalf of the user.

The relationship between these objects and those operations is mutual; an operation may be performed on one or several objects and an object can be assigned different permissions.

$$P = 2^{OPS \times OBS}$$

$2^{OPS \times OBS}$ denotes the powerset of the set of ***OPS × OBS***

OPS × OBS denotes the cartesian product of operations and objects.

In our model, we provide 2 types of permissions: ***Default Permissions DP*** and ***Adaptive Permissions AP***.

$$P = \{DP \cup AP\}$$

1) ***Default Permissions DP:*** represent the regular basic permissions that are defined explicitly by system administrators. DPs translate the authorizations that approve performing operations on objects or system sources.

Default Permission Assignment DPA describes the many-to-many relationship that connects Roles with DPs; where the assignment of a role guarantees a user one or several Default Permissions DP and a single Default Permission may be assigned to one or several roles.

This assignment is predetermined and provided by the different system policies. They might depend on the user's Contextual Constraints that are rechecked at every request time.

$$DPA \subseteq Roles \times DP \times CC$$

2) ***Adaptive Permissions AP:*** In the classical RBAC System, users are assigned to roles and roles are assigned to pre-defined permissions, our model adds to these assignments new responsive ad-hoc assignments that can be offered in case of an access denial. The proposed AP are based on providing alternative permissions to access authorized resources provided by an ***Alternatives Provider*** component.

   Different examples alternative providers are presented at the validation part (e.g. a Semantic Similarity Matching component in Chapter 6, decision making flow chart in Chapter 7 and an alternative solutions database in Chapter 8).

The objective is to suit special contexts and situations confronting users (e.g. important consultation, critical mission, crisis, emergency, etc.). The decision-making takes the user's contextual attributes into consideration.

Adaptive Permission Assignment APA: is a many-to-many mapping relationship that describes the assignment of Roles to Adaptive Permissions AP. These permissions are generated in access denial cases, where our model tries to search for similar authorized resources that can be offered as an alternative solution. The solution is determined and gouverned by:

1. the user's Contextual Constraints CC,
2. the Situation Sensitivity level of an access request SS,
3. the existence of alternative resources that are authorized.

$$APA \subseteq Roles \times CC \times SS \times AP \cup Alt$$

4.3 The Interpretation of Context

Applying context-aware decision-making is an important feature for modeling access control within pervasive systems.

The technological evolution of computing machinery has enabled pervasive environments to extract the user's contextual information and reach to his surrounding environment. Nowadays, modern devices are equipped with advanced networking capabilities (such as Bluetooth and Wi-Fi, etc.), location-aware services (GPS) and many other options. At the environment level, invisible embedded devices and sensors are turning physical spaces into active, smart surroundings making the space more interactive and adaptive to meet the user needs.

As we will show in this section, our model highlights the importance of not only taking the user's identity into consideration for role and permission assignments but also his contextual attributes that are provided by the ***Context Provider CP*** that is an application dependent component.

A non-exclusive application dependant list of contextual attributes can be taken into consideration within our model, such as the list proposed by [Dey, 1999]:

- Spatial information (e.g. location, orientation, speed, and acceleration)
- Temporal information (e.g. time of the day, date, and season of the year)
- Environmental information (e.g. temperature, air quality, light or noise level)
- Social situation (e.g. who you are with, and people that are nearby)
- Resources that are nearby (e.g. accessible devices, and hosts)
- Availability of resources (e.g. battery, display, network, and bandwidth)
- Physiological measurements (e.g. blood pressure, heart rate, respiration rate, muscle activity, and tone of voice)
- Activity (e.g. talking, reading, walking, and running)
- Schedules and agendas

The presented model enables two types of context-aware assignments and decision-making, which are: context-aware role assignments and context-aware permission assignments.

4.3.1 Context-aware Role Assignment

Context-aware role assignments can facilitate and enrich the role assignment process. The inclusion of contextual attributes is performed and specified by the application domain in a way that serves the users. Following our model, the contextual attributes needed for role assignments can be efficiently defined and updated by resource administrators according to the emerging system needs.

Eventually, every contextual attribute can be personalized according to the needs of the application domain and the assignment of a role could be tied to the existence of a combination of several contextual attributes.

Taking an example in the healthcare domain, the proposed model would enable scenarios like detecting the location of a doctor as he enters the emergency section and automatically assigning him the "emergency_Dr" role.

The location can be interpreted in different levels and different types according to the application needs. A ***General location***, for example, can be defined within the hospital bound (detected by the user's GPS coordinates and double checked using his RFID card as he checks in at the entrance), ***Specific location*** can be related to a specific department (detected using RFID card or any other authentication mechanism employed), ***Proportional proximity*** from a certain object can also help to determine a user's location (e.g. Dr approaching from a patient bed or an examination machine, etc.). An example of context-based role assignment using proximate location attributes can be the attribution of "Treating_Dr" role as the Dr enters a patient room.

In a more advanced system, the contextual constraints of a user and their usage in role assignments can help the system to perform intelligent interactions related to permission and task assignment that would enrich the medical procedure. Applications might be like loading automatically the patient's case details to the Dr's machine as he approaches from the patient bed (due to his new "Treating_Dr" role assignment), other applications could notify the doctor of critical changes to the patient condition and provide him with the procedure to follow while treating a case, etc.

A similar hierarchy can be also applied to the temporal aspect; a ***General time*** scheme could define the current time at which the user is gaining access to the system. A ***Specific time*** scheme could be defined to specify, for example, the interval of working hours at which a specialized Dr is granted his "Clinical_Examinator" role and is authorized to access the records related to the patients included in the system's agenda, a ***Proportional timing*** could be related to another event or action taking place such as granting a Nurse the permission to obtain "Medication_Service" role in order to obtain access to the medication cabinet 15 minutes after the meal serving time.

We would like to note that the contextual-based role attribution is not only meant to be employed for facilitating access to data sources but it can also be used as a protective security step to cancel a role attribution. For example, the same system that may offer more accessibility as the Dr enters the emergency department section, can employ the users contextual attributes to allocate the Dr getting out of the emergency section and entering the hospital cafeteria in order to revoke all his access rights to the patient information.

Furthermore, the contextual attributes of a user can be used by the system for mission assignments that take place according to a user's profile and to his location (for example, if a patient case faces some complications and a specialized Dr is needed to interfere as soon as possible, the system could allocate the group of Drs that might be pertinent to fulfill the mission and assign the task to the nearest Dr).

4.3.2 Context-aware Permission Assignment

At a finer-grained level, a pemission assignment within our model can be also performed depending on the existence of some contextual constraints that might be related to or different from the ones needed for role attribution.

The importance of verifying the contextual constraints at the permission attribution level can be due to:

1) the need for more specific qualities in the user's context in order to perform the permission,

2) the dynamicity of the user's context and the fact that they evolve over time in pervasive environments imposes a high level of detection and update whenever the context changes, which might be coastly to the system so an alternative solution would be performing the verification step to test wheather the contextual constraints detected at the role attribution level still correspond to the role and are applicable for the permission at the permission attribution time.

In general, our vision favors the distinction between the type of contextual attributes related to each assignment; the contextual attributes related to role assignments can be more generic and might define the type of authorizations permitted while the permission-assignment contextual constraints might be more specific and related to the performance and functionality of the granted permission (e.g. the contextual attributes of a user's machine can ensure access to the acquired resources while providing the most appropriate data presentation).

Continuing the previously mentioned scenario, context-aware permission assignments would enable the previously mentioned Emergency_Dr to automatically obtain access to the patient record as he approaches from his bed. In a pervasive context, the patient data and any other machine-extracted information could be directly loaded on his personal tablet, other permissions and task-based authorizations may follow, such as: ability to execute further operations, order treatements, etc.

4.3.3 Benefits of Context-aware Assignments

The proposed PS-RBAC model allows making authorization decisions based on contextual attributes of the user and his surrounding pervasive environment. This can improve the decision-making process in many ways.

The benefits of our model are summarized as follows:

- ***Flexibility.*** Context-awareness is applied dynamically within our model, at the design level, administrators can have the freedom to specify complex context-aware authorizations that can be applied and enforced easily at the implementation level and at run-time.
- ***Explicitness***. The biggest advantage of context-driven modeling is its explicitness. By describing possible contexts for role and permission attribution, the application domain and resource owners can unambiguously identify which contexts should be active at a specific resource usage/ information consultation process, which facilitates the management process and guarantees constraint accessibility.
- ***Interoperability and Extensibility***. The explicitness greatly promotes extensibility and interoperability in an open and constantly changing system such as a pervasive space. The separation of the knowledge definition and the availability of the entities greatly enhance its capability to adapt to changing configurations.
- ***Conflict Detection***. Because there is no ambiguity in identifying active contexts, and the explicitly defined associated actions to take, context-driven access control modelling can be powerful enough to detect impermissible contexts and contradictory behaviors and to reflect the results on the permissions granted. This can be easily performed since

the contextual attributes of the subjet are sometimes checked twice (role attribution level and at the permission attribution level).

- ***Capture Environmental Effect***. Our proposed context-driven model is reactive; instead of trying to proactively control and coordinate assignments, systems following this model passively react to the active contexts observed by the Context Provider and thus, executing predefined actions associated with active contexts. This reactive nature also contributes to help in capturing impermissible contexts and environmental effects.
- ***Scalability***. The scalability offered by the PS-RBAC model arises in its ability to help access management systems to handling the growing complexity of role and permission assignments while taking context awareness into consideration.

4.4 The Interpretation of The Situation

The situation is an important component in our model; our basic objective for enabling situation-aware decision-making is to help users in confronting access denials during an urgent or critical case. Knowing that the classical solutions proposed for situation-awareness tend to use the "Break-Glass" concept or provide flexible accessibility through assisted help, we introduce a relatively low risk solution based on finding similar alternative ressources that might serve in confronting situation scenarios and in preventing some unnecessary access right violations.

In order to provide a generic solution that can also consider applying any other method for situation-aware decision-making, we have proposed to employ different adaptive procedures according to different situation levels included in a ***situation sensitivity scale***. The sensitivity levels can eventually be precised and personalized by the application and ressource administrators along with the different adaptive solutions and permission assignments related to them. Finally, for more freedom, an additional user-based triggering mechanism can be integrated at the application level, where the user facing an un-expected case or un-planned situation could choose the adaptive flexibility of access needed through inserting a sensitivity level.

Our model leaves the detailed situation specifications to the application domain but in general, a conventional situation determination mechanism employs deductive methods to attain situations where they logically tie contextual values to define a situation. For example, the simple data measurements of a heat or smoke detecting sensor can only turn into situational information to compose a fire alarm or a critical chemical leackage situation if they reached to a certain limit.

Knowing that the process of finding alternative solutions is expensive in terms of processing cost and time, we have proposed to tie the process of alternative permissions search and retrieval to a certain situation sensitivity level or threshold (that varies from an application to another).

The usage of sensitivity levels can also serve in deciding the depth or granularity of the search process conducted to find alternatives. Examples can be in cases where the application employs ontologies for providing semantic similarities, or in cases where the search process could attain more results when expanded to cover wider

location zones or when using information sources existing in external processing servers.

Finally, we note that In order to perform efficient retrieval for alternative solutions, the adaptive permission assignment proposed within our model takes into consideration searching for permissions that matches the user's contextual constraints.

4.5 RBAC Vs. PS-RBAC

Our choice of upgrading the RBAC model into the PS-RBAC model returns to the importance of adapting this widespread used model with the evolving needs of the modern pervasive computing environments where context and situation awareness are key quality requirements.

In addition to it's wide spread usage in modeling access rights in a way that maps with the organizational structure, our choice is also due to the significant transparence and suitability of the RBAC representation and its application within various modern day IT infrastructures. Today, RBAC feautures are included at all the levels of enterprise computing, including the operating system, the database management system, the network and the different enterprise management levels.

The use of RBAC is also being incorporated and integrated within infrastructure technologies such as Public Key Infrastructure (PKI), workflow management systems, and directory and Web Services.

Our proposed PS-RBAC model has added some characteristics on the RBAC model in order to enable:

- Dynamic and fine-grained authorizations

  On both role assignment and permission assignment levels.

- Context-aware decision making

  Our model enables dynamic context-aware access control. At design time, administrators have the flexibility to specify any needed context-aware permissions. At run-time, an authorization can enforce any context-aware decision making automatically because it is not statically bound to a specific application.
  Context implementation is separated from the main model and tied to target applications. Since every context type definition and context implementation is independent of the specification of the access rules, any change to them has no effect on other parts of the model. Thus the security infrastructure is flexible and permits easy extensibility.

- Situation-aware decision making

  Situation-awareness has been designed to be personalized and adjusted at the application domaine. This enriches the flexibility of the decision-making process and makes it more pertinent to meet the application accessibility needs.
  The integration of sensitivity levels has enriched the application of situation aware decision-making and the granularity of solutions Low risk solutions proposed. They have also enabled applying extensible

solutions that can have some risk such as “Break-Glass” or assisted access solutions.

4.6 Conclusion

In this chapter, we have presented PS-RBAC: a generic access control model that extends the RBAC model to provide context and situation aware decision-making. Our proposal aims to offer a responsive access control model that guarantees flexible and dynamic decision-making in order to meet the evolving accessibility needs within pervasive environments without influencing the system’s integrity.

The proposed solution provides a balance between the rigidity of the security constraints specified by system administrators and the “anytime, anywhere, anyhow” accessibility required to ensure the quality of service to the users of pervasive systems.

PS-RBAC has confronted the access denials taking place during critical situations by providing alternative permissions that allow users to access similar resources retrieved by an application specific component dedicated for providing similarity.

Finally, we conclude this chapter with a comparative table to comparing our model with the different research works that were conducted to extend access control modeling (in particular the RBAC model) in order to meet the needs of pervasive computing systems. This objective was basicly achieved by taking two directions into consideration: context-awareness and situation-awareness, see table 4.1.

The table shows how the PS-RBAC model has succeeded to provide adaptive context and situation-aware decision-making and has also provided means that enable adopting the different previously proposed situation-aware access solutions.

In the next chapter, the system architecture that performs our proposed adaptive decision-making will be presented in detail.

Model	Contextual awareness				Situation awareness		
	Time	Location	Date	Other	Break Glass	Assisted Accessibility	Ad hoc Adaptation
T-RBAC [Bertino et al., 2001]	x						
Spatial RBAC [Hansen et al., 2003]	x	x					
Dynamic RBAC [Zhang et al., 2003]	x	x	x				
Geo-RBAC [Bertino et al., 2005]	x	x					
uT-RBAC [Chae et al., 2006]	x	x		Logic			
Context-Aware Access Control Model for Pervasive Computing Environments [Emami et al., 2007]	x	x	x				
Intelligent Access Control Mechanism for Ubiquitous Applications [Lim et al., 2007]	x	x		Neural networks			
Context-aware RBAC in pervasive computing systems [Kulkarni et al., 2008]	x	x	x				
Quality-Aware Context-Based Access Control model for ubiquitous applications [Filho et al., 2008]	x	x	x	Quality of Context			
BTG RBAC [Ferreira et al., 2009]					x		
Extending access control models with break-glass [Brucker et al., 2009]					x		
Redirection policies for mission-based information sharing [Keppler et al., 2006]						x	
Pervasive Software Environments for Supporting Disaster Responses [Catarci et al., 2008]						x	
Pervasively Shared Situational Awareness [mayuni et al., 2008]						x	
Team and Role based Access Control [Kawagoe et al., 2011]	x	x	x			x	
Risk-Adaptive Access Control [Cheng et al., 2007]							x
PS-RBAC Pervasive Situation aware RBAC [Al Kukhun et al., 2009, 2012]	x	x	x	Similarity	x	x	x

Table 4.1: The evolution of situation and context-aware access control modelling

Chapter 5:

A Pervasive Situation-aware Query Rewriting System

5.1 Introduction

The evolution of Pervasive Information Systems PIS has introduced new challenges to access management. Especially when dealing with the need to provide secure access to users at anytime, anywhere, anyhow.

After realizing an analytic study of the different works conducted for modeling a pervasive access control system, we have identified the the main characteristics needed for achieving efficient decision-making within pervasive environments, which are: (i) the dynamicity of context-awareness, (ii) situation-awareness and (iii) the multi-distributive nature of pervasive resources, access policies and managing authorities.

A primary interest of our research is to achieve a balance between the security required to protect pervasive resources and the seamless accessibility needed to attain a transparent interaction and exchange for information flows between different services, see fig. 5.1. In order to achieve this vision, we have applied multi-layered adaptation on the life cycle of designing effective access control management system.

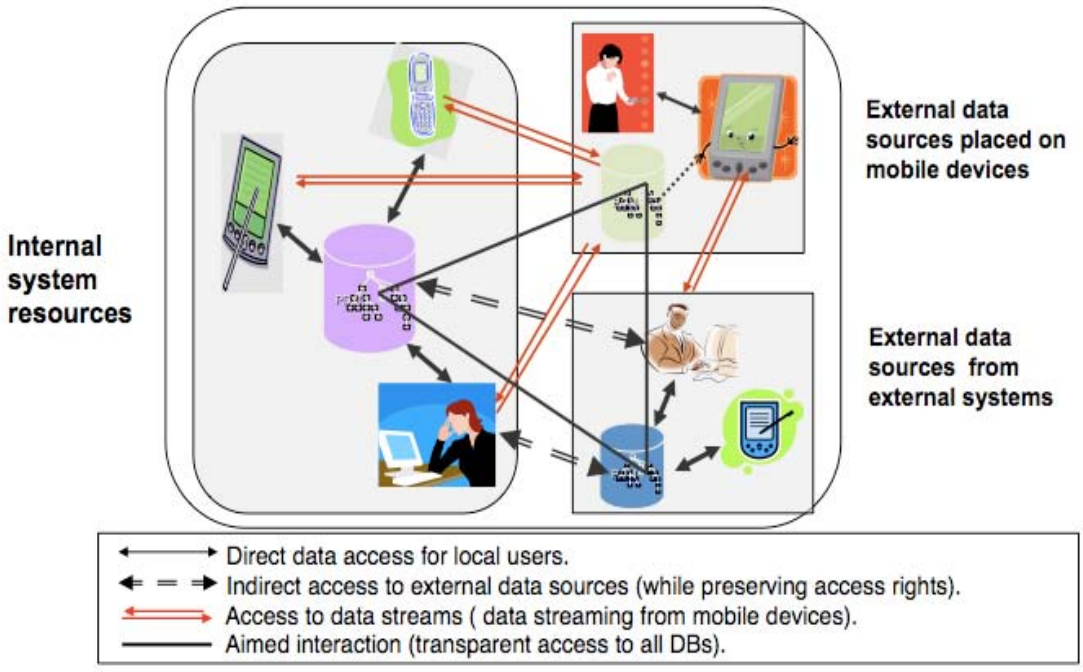


Figure 5.1: Access modalities within Pervasive Information Systems

At a first step, we have considered the access control modeling level; since PIS resources are protected by organizational policies and practices that are usually expressed using the RBAC (Role based Access Control) model, we have proposed to

apply adaptive organizational modeling through the PS-RBAC (Pervasive Situation-aware Role Based Access Control) model.

The propsed PS-RBAC model has taken into account the need for generating dynamic permission assignments that meet the volatility of the contextual attributes of pervasive users. In addition, the system has proposed to confront access denials within critical situations by generating alternative permissions that allow access to similar authorized ressources (see chapter 4).

At a second step, we're interested in reflecting and elevating the adaptive vision presented in the PS-RBAC model to the implementation level. Thus, we need to repose on a technology that can:

1) enable the distribution of access control rights in a service-oriented architecture,
2) perform efficient decision-making despite the dynamicity of pervasive users context and,
3) meet the multi-distributive nature of pervasive resources, access policies and managing authorities.

The choice was drawn on XACML (eXtensible Access Control Markup Language): an interoperable service-oriented standard that enforces access control through fine-grained, context-aware security policies and realizes centralized decision-making while managing the distribution of resources and access control policies.

In this chapter, we will apply the adaptive vision of the situation-aware pervasive permission assignments, proposed within the PS-RBAC model at the application level by presenting PSQRS: a Pervasive Situation-aware Query Rewriting System. As we will illustrate in the rest of this chapter, the different system subcomponents interact to offer adaptive context and situation-aware decision-making in a service-oriented approach. The alternative access solutions are achieved through rewriting XACML queries.

5.2 PSQRS System Architecture

The system architecture we're introducing in detail in this section aims to provide an adaptive querying process that can: (i) generate context-aware permission assignments and (ii) compensate the lack of situational reactivity that exists within traditional access control systems especially in cases where pervasive users face realtime situations and get confronted with access denials.

The objective is to provide users with transparent accessibility, interactive context and situation-aware querying that meets their needs and satisfies the security requirements imposed by the system. This is accomplished through a query rewriting mechanism, which interprets access denials and rewrites XACML queries - based on similarities and semantics - in order to offer accessibility to alternative authorized ressources.

In traditional situation-aware access control systems, administrators either anticipate a group of extreme situations where users may request access to non-authorized resources and design some predefined responsive measures to be taken into consideration or allow users to attain unlimited flexibility through «Break-Glass» option.

The challenge of ensuring situation-awareness within pervasive systems is that an access denial might be diagnosed due to challenges in attaining access not only at the application content level but also might be resulting from the contextual attributes of the user or from difficulties to obtain the right credentials to use a certain service, access a certain network or due to misfunctionalities of the server providing the needed information. In addition, the dynamicity of the pervasive context could offer a non-extensive group of unexpected situations that have a varying importance level.

Our research highlights the importance of providing a level of adaptive access control that can react in situations where even using extreme solutions like «Break-Glass» wont serve in saving the situation (e.g. providing access to a contaminated area during a nuclear crisis or a toxic gaz leackage accident won't serve users) or when the consultation faced with an access denial is not taking place in an extreme situation that justifies depassing the system's security shields but in a case that is important for the accomplishment of the user's mission.

The presented solution comes as a modular step between the strict query processing procedures and other flexible accessibility solutions like BTG & delegation-based solutions that can be also integrated within our model in the future.

Next, we present the detailed functionality performed by the different sub components of the proposed system (PSQRS) illustrated in fig. 5.2.

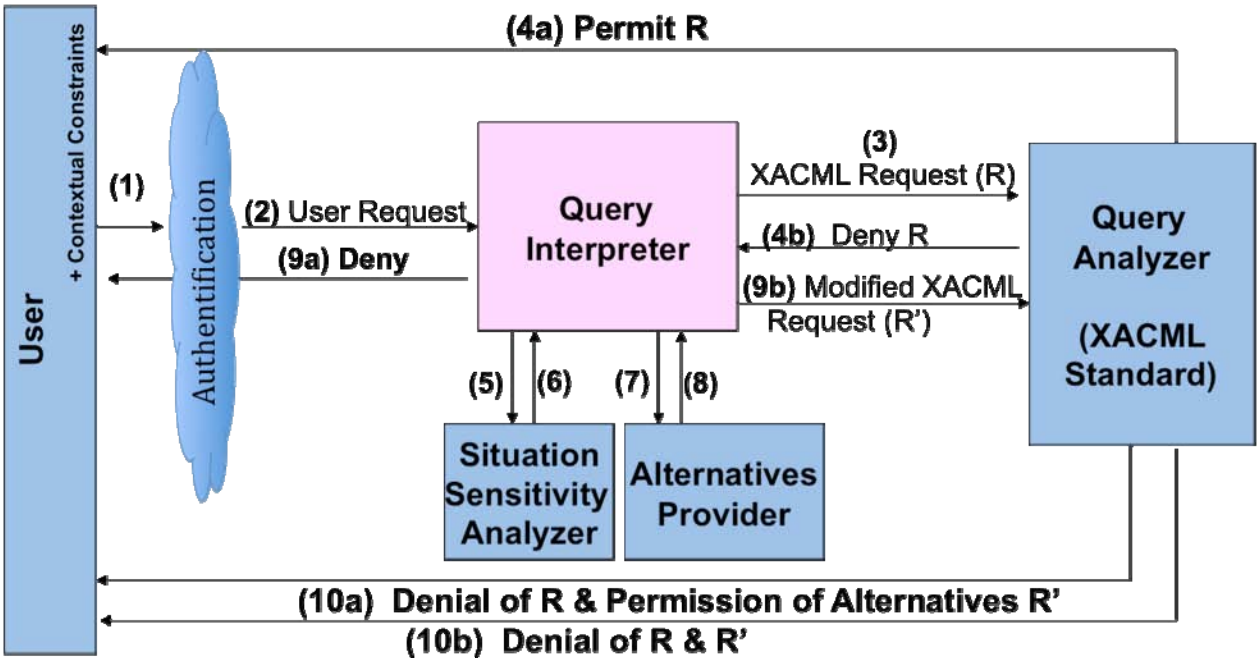


Figure 5.2: PSQRS Pervasive Situation-aware Query Rewriting System

5.2.1 User Authentication

Authentication is an important step within our system; it does not only allow the system to verify the user's identity but it also contributes in providing the system with the basic contextual attributes obtained at login time.

Authentication is achieved based on a validation test performed upon the presentation of any method that demonstrates the user's identity and his unique credentials to that system. These credentials can be in the form of:

- ***Something the user knows*** (simple username & password),
- ***Something the user has*** (smart RFID card),
- ***Something the user is*** (e.g. biometric authentication, which has many forms such as: finger print, eye iris, voice, etc.) or,
- ***Something the user is near from*** (proximity-based authentication is achieved using credentials that are gathered from the surrounding environment).

Reflecting the results of a successful authentication step to our PS-RBAC scheme, we'll find that at the moment the user is authenticated, his contextual attributes will be retreived and saved within the assigned session's log file. This combination of identity and contextual information will contribute to perform role assignments that would enable the user to execute several operations on system objects.

5.2.2 Context Representation

Our system extracts the user's contextual attributes at system login time and at each querying step as they influence role assignments and permission assignments. The acquition of contextual attributes is application-dependant and is generally easily achieved within pervasive environments due to their richness with sensing objects and with advanced off-the-shelf products.

It's important to state that we distinguish between ***context management*** that is an application-dependant feature generally achieved by an internal Context Handler and ***context-aware decision-making*** that is one of the main objectives of our system and is performed by using data extracted and modelled within this application-based Context Handler component.

According to application needs, the context management layer can be designed in order to:

1. ***Model context to define the granularity of detail needed in context acquisition.*** It is the responsibility of the application designer to define the appropriate context models based on the application requirements. Design of such models also depends on the available sensing technologies [Kulkarni et al., 2008]. For example, a nurse's location may be modeled at the granularity of a ward or based on the proximity of the nurse to a specific patient in a ward. The available location tracking sensors would determine this granularity.
2. ***Personalize the time-rate at which the user's contextual attributes can be constantly checked in order to detect changes***. knowing that a contextual attribute change should be mapped within context models as domains or intervals (location change granularity can be decided in a way that maps and serves the application need, e.g. location change can be from a room to another, department to another, floor to another, building to outside, etc.).
3. ***Decide the influence of detecting a contextual change***: events to be triggered and actions to be taken when contextual changes take place (e.g. enabling a new role or revoking a present role).

The purpose of context modelling is to drive the design of the context management layer for aggregating the sensored or acquired data in order to generate contextual information in a way that suits the application needs. In the literature, different approaches have been used for context modeling [Strang et al., 2004]:

- ***Attribute-value pairs*** represent contextual elements and their values [Schilit, 1994]. They are easy to handle despite their limited functionality in comparison with other advanced methods.
- ***Domain specific ontologies***, such as RDF and OWL [Wang et al., 1994b]. Basic XML schemas support interoperability and provide a shared understanding of a context model in a way that allows its usage by different consumers. Ontologies provide mechanisms to define relationships between different contextual elements and domain concepts.
- ***Graphical approach*** such as the Object-Role Modeling (ORM) framework [Halpin, 2001], [Henricksen et al., 2004]. The ORM approach provides graphical mechanisms that help in the development of context models for the domain of interest.

Our system interprets contextual elements that are modelled in attribute-value pairs and pass them by to be considered within the decision-making process within XACML requests. See fig 5.3 for simple attribute-value context representation.

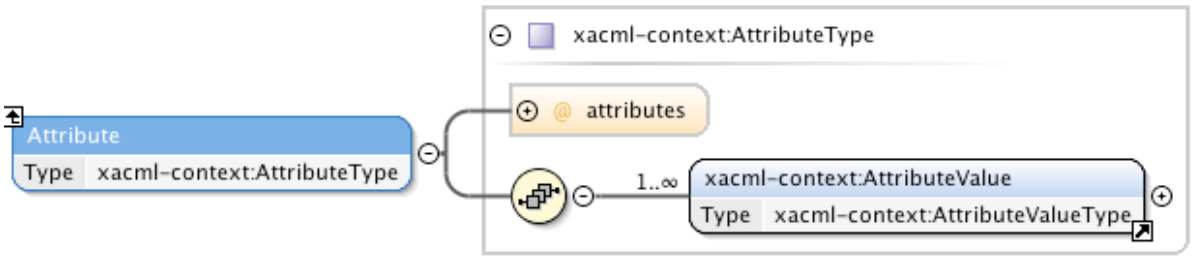


Figure 5.3: XACML context representation through attribute-value pairs

Finally, we cite some examples of contextual information that can be taken into consideration within our system and can influence the decision making for role and permission assignments:

- Spatial information (e.g. location, orientation, speed, and acceleration)
- Temporal information (e.g. time of the day, date, and season of the year)
- Environmental information (e.g. temperature, air quality, and light or noise level)
- Social situation (e.g. who you are with, and people that are nearby)
- Resources that are nearby (e.g. accessible devices, and hosts)
- Availability of resources (e.g. battery, display, network, and bandwidth)
- Physiologicalmeasurements (e.g. blood pressure, heart rate, respiration rate, muscle activity, and tone of voice)
- Activity (e.g. talking, reading, walking, and running)
- Schedules and agendas

5.2.3 Query Interpretor

The Query Interpretor is the main coordinator in our system and the only component that is not application based (but can be also adjusted by the application domain if the need arises). It's in charge of managing the interactions that take place between the different subcomponents and the user, such as:

- Receiving the ***User Query*** and his ***Contextual Attributes***,
- Reformulating the Query into an ***XACML Request*** and,
- Sending the reformulated ***Query*** to the ***Query Analyzer*** to be checked by the methodology provided within the XACML standard.
- In the case of receiving an access denial result when analyzing the XACML query, the ***Query Interpretor*** will redirect the initial request in order to perform a failure recovery step that aims to find alternative options through the communication with 2 main components:
  - Send the request details to the ***Situation Sensitivity Provider*** in order to retrieve information about the situation level (and any related information that can be linked to the solution to be performed next).
  - Send the requested ressource to the ***Alternatives Provider*** that is an application specific component that performs a search process in order to retreive a list of alternative ressources (if available) that can be proposed as a solution to face the unauthorized access requests.
- In the case of finding similar ressources, the ***Query Interpretor*** will rewrite the XACML request and resend it to the ***Query Analyzer*** to verify if the user can obtain access to the proposed ressource(s).
- Finally, in the case of not finding any similar ressources that can help in reformulating the query to find an adaptive solution, the access denial returned from the Query Analyzer will be passed back to the user.

5.2.4 Situation Sensitivity Analyzer

The definition of a user's situation was generally achieved in the literature by qualifying the information resulting from the acquisition of the user's contextual attributes (location, time, temperature, etc.) in a more ***high-level*** and ***time-invarient*** way [Weißenberg, 2006].

Our research is interested in the situation of the consultation, which might extend the user's contextual domain and his environment. Thus, a situation in our perception is concerned of describing the state, condition or circumstances facing the user or causing a system consultation and an access demand.

The objective of the Situation Sensitivity Analyzer component is to judge if the sytem could perform the adaptive search in order to help a user faced with an access denial when asking for an access permission.

The functionality of the situation sensitivity analyzer component is application based since most applications build an internal knowledge base personalized to manage

the situations that might take place. In general, we define the main steps that need to be performed for situation detection, modeling and analysis at the application level:

- ***The personalization of sensitivity levels***:
  Taking the example of a building alarm management system: an alarm notifying a broken glass problem at the building entrance door can trigger a situation of level 1. While an elevator problem alarm can trigger a situation of level 2 and a fire alarm can trigger a situation of level 3.
- ***The interpretation of the significance of situation levels elevation:*** For instance, the previously mentioned alarm management system considers the increase of the situation level as an increase of the situation's importance.
- ***The definition of mechanisms to identify and model situations:*** The extraction of a situation is usually performed by applying some reasoning logic on a combination of some contextual attributes. Continuing our building alarm management system example, the notification of a broken door might be triggered by a sound recognition process performed by the sound detection sensors, by the wind detection sensors or by temperature sensors placed at the building entrance. Eventually, an additional consultation using the video surveillance camera is usually added for verification. The elevator problem can be triggered by the internal elevator program or explicitly notified by a user trapped within. Finally, the fire alarm is triggered by smoke detection sensors.
- ***Specifying the results built on the occurrence of each situation level or the reactions to take place***: Continuing our alarm management system example, a broken door situation can be tied to a phone call or an automatic notification performed by the system in order to deliver the problem to the maintenance department. In the case of an elevator alarm situation, the notification will be targeted to the maintenance department of the elevator's company, which would send a specialist in a certain delay that also depends on their own scale of situational interpretation (light problem, technical problem, someone trapped in, etc.). Finally, a fire alarm situation triggers a notification delivered directly to the fire fightenning department for immediate response.

5.2.5 Alternatives Provider

Our research and the adaptive alternative-based solution is mainly concerned in content-based access requests. The process of retreiving alternative documents, elements or resources is performed and customized at the application level where the structure and content are clearly modeled, several examples are demonstrated in the validation part (e.g. a Semantic Similarity Matching component in Chapter 6, a decision making flow chart in Chapter 7 and an alternative solutions database in Chapter 8).

In the case of employing semantic similarity for finding alternatives, the choice of the type and granularity of the similarity search performed depends on the type of resources existing within the system and the type of information requested in the user access permission query.

For example, taking a medium-level emergency case where a patient faints down and a nurse launches an access request to access his latest analysis lab result in order to check if he's diabetic. Knowing that the current contextual attributes don't authorize the nurse to access this particular file, an alternative similarity search process could redirect the query to provide him/ her with the needed information from other resources such as the patient's nutrition folder.

The objective of employing the adaptive similarity-based retrieval is to prevent using extreme flexible access control solutions such as the «Break-Glass» option when other less risky solutions exist.

At more profound level, the realization of such efficient results of ***keyword based similarity-search*** can only be attained through specific queries e.g. the nurse explicitly asking for the latest sugar level test within the latest lab analysis file and there, the retrieval system would be able to perform a more profound search process to retrieve any information related to blood sugar results within any other authorized folder.

Knowing that the patient records are xml-based documents, the retrieval of a certain result existing within other authorized documents would be easily achieved. Overmore, our implemented system (illustrated in ch 7) has included a special functional component that enables filling the prototype's datatbase in xml-based documents so that our system would be able to perform (automatic document parsing, keyword extraction and eventually similarity comparison - based on xml tags).

More advanced semantic similarity search can also be applied through ***ontology-based semantic expansion techniques***. Where the element or resource requested within the user's access request could be replaced by similar domain-specific concepts (e.g. queries related to sugar level tests can be reformulated to diabetics, blood pressure can be replced with hypertention or hypotention, etc.).

The ontology-based semantic expansion can provide a variety of options according to the granularity and the level of detail chosen to conduct the search for alternatives (a term or concept could be replaced with other neighboring terms that have the same father concept or with its own child elements or with both). For example, the choice of replacing "blood pressure" into "hypertention" or "hypotention" follows a parent-child relationship.

Knowing that the increasing number of alternatives to be checked can be costy in terms of query processing time and power, the level of detail and the topological relationships can be customized and tied to the importance of the sitation, which can be expressed within our system through the situation sensitivity level.

The alternative solutions we're looking to provide when meeting unexpected or undesired situations might not only be offered on the resource level through similarity search (keyword based search or semantic similarity based search) but might also exist in the form of ***predefined alternative solutions*** saved in a simple database in the system. The solutions included in this database can be built and filled gradually by system administrators and by end users as they confront situations in real-time basis.

To illustrate this form of solutions, we return to the building alarm management system example. Taking the case where the broken door problem takes place at weekend or vacation time where the maintenance department personnel are not available, the following alternative solution can be predefined within the system; the

notification will be redirected to the reception clerc or security gardian and would propose to him to place a hard carton panel to cover the damaged area. Eventually, a security and access control step would accompany such procedure, e.g. giving the contacted person the authorization to access the buildig material warehouse.

This predefined alternative solutions option can also contribute in solving other types of access denials related to technical issues that might face a pervasive user. Examples of such situations can be: the demanded resource format is not supported by the user's machine, the network bandwith is not sufficient for delivering the desired resource, the user's machine storage can't handle the needed resource, etc. The alternative solutions could be provided through integrating a data format conversion application, using cloud spaces for previewing the needed content, etc.

Finally, in some application domains, the alternative predefined solutions could be extracted from the Frequently Asked Questions FAQ database.

5.2.6 Query Analyzer

The ***Query Analyzer*** is the decision-making component of our system. In order to suit the complexity of access control needs within pervasive environments, we have chosen to realize decision-making using the XACML standard [OASIS, 2003, 2005a].

One of the main advantages of XACML is its capacity to realize centralized decision-making within multi-distributive environments where the resources are decentralized and the access control policies are distributed and managed by different authorities.

Another reason behind choosing XACML is it's ability to support context-awareness in an expressive manner that does not only cover the subject's contextual constraints but that can also characterize the contextual attributes related to the requested resource, the action that the subject needs to perform and his environment, see fig. 5.4.

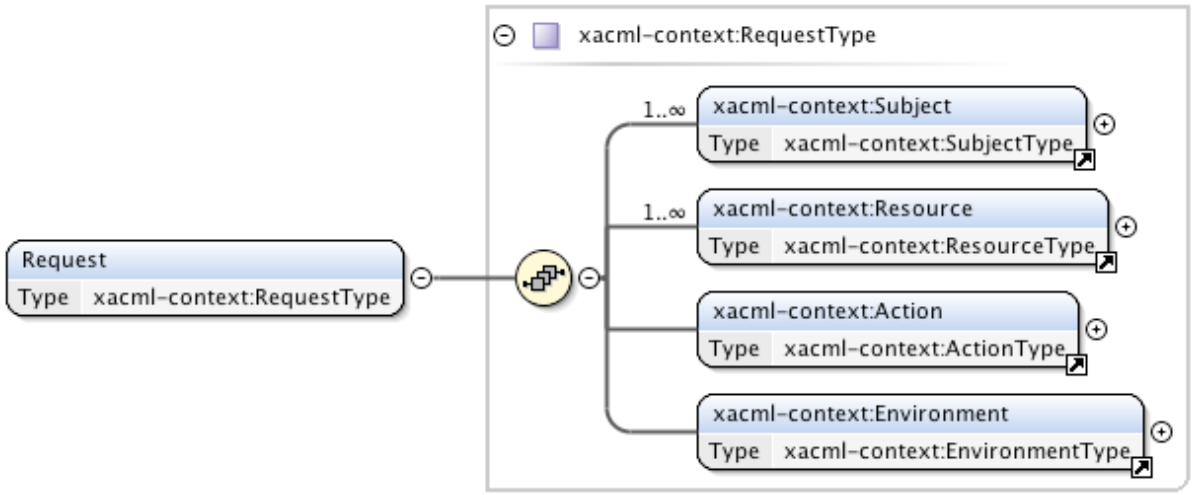


Figure 5.4: The context-awareness of XACML requests

The XACML standard has provided an implementation scheme that performs decision-making through an organized interaction procedure that takes place between its different subcomponents, see fig. 5.5.

In general, in order to check the validity of an access request, the access control system must check if there is a security policy that corresponds to this request. The verification procedure in XACML proceeds as follows:

- An access request launched to perform an action on a particular resource is usually sent to the ***Policy Enforcement Point*** PEP component that interferes - in a service oriented manner - to verify whether access is authorized or not.
- The PEP will form an XACML request with the help of the ***Context Handler***, which passes the contextual attributes related to: the requester, the resource in question, the action, and the environment along with other information pertaining to the request to the ***Policy Decision Point*** PDP that is in charge of decision-making.

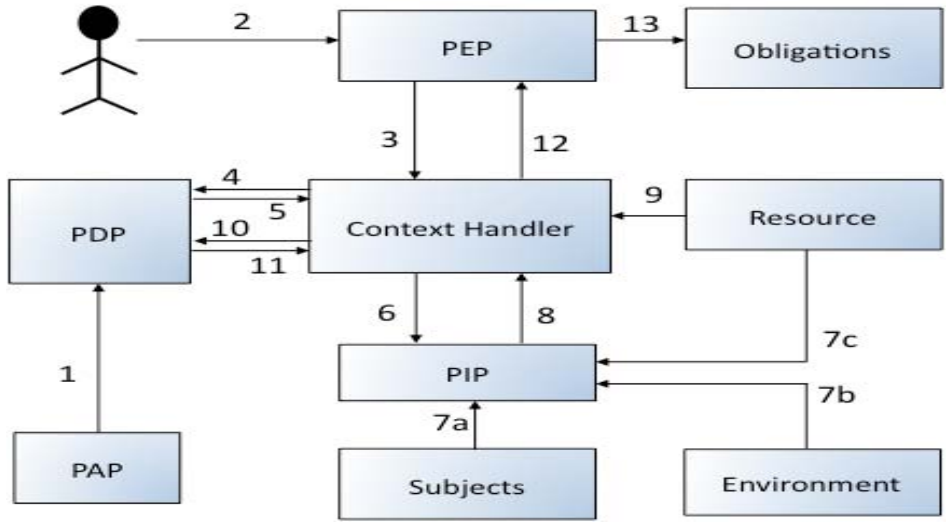


Figure 5.5: XACML data flow chart

- The PEP ***Policy Enforcement Point*** represents the access control management system that is responsible for protecting the resource. In our system, it is represented by the ***Query Interpretor,*** which receives the user query along with the user's contextual constraints and forms an ***XACML request (R)*** to be passed by and judged through the PDP that is placed within our ***Query Analyzer*** component.
- A decision is attained by consulting the list of access policies in charge of managing access to system resources, which are located in PAPs (***Policy Administration Points***). A relevant security policy is located and chosen by the PDP who will verify if the attributes included within the XACML request meet the required attributes described within the policy. Accordingly, a decision is made about whether or not the access should be granted.
- That answer in the typical XACML sequence will be returned to the PEP, which returns the appropriate response to the customer and ensures that this decision is respected and that the client can only access authorized resources, which can then allow or deny access to the requester.

- In PSQRS, if the access is permitted, the result will be sent directly to the user and in the case of an access denial, the result will be sent back to the ***Query Interpretor*** that will check the sensitivity of the situation in order to launch the adaptive search for alternative similar resources.
- As similar ressorces are found, the initial user query is rewritten to a new ***alternative query (R')*** that is sent again to be verified by the ***Query Analyzer.***
- Finally, in the case of of attaining an access permission, the result will be directed to the user to propose to him the allowed ressources as alternative solutions.

5.3 Conclusion

In this chapter, we have presented PSQRS: a Pervasive Situation-aware Query Rewriting System architecture that realizes context-aware access decisions and responds to access denials taking place in real-time situations by searching for alternative ressources based on semantic similarity.

The system realizes access decisions based on the XACML standard and is designed to ensure a generic solution that can be applied within any application domain through the usage of interactive, flexible and portable components.

In addition to the main characteristics provided within the default settings of our system layers, we have provided a customization ability in order to increase the efficiency of decision making in a way that suits the conceptual characteristics and different user needs according to the application domain, the flexibility influences the following aspects:

- Context acquisition, modeling, interpretation and context-aware permission assignment.
- Situation acquisition, situation levels definition, situation-aware access control and the amount of flexibility to be applied.
- Similarity search that can vary according to system content and semantic structure.

Part III:
Implementation & Validation Examples

Chapter 6 :

Ensuring Pervasive Accessibility to Mobile Geriatrics Teams

6.1 Introduction

In this chapter, we present a first application example where we apply our proposed PS-RBAC model and the PSQRS architecture in managing the accessibility challenges that face the different members of the Mobile Geriatric Team MGT when trying to access patient records during their mobile missions.

The presented works are based on the results of a research project that was financed by the French Ministry of Health[2] and realized by a joint collaboration between the Faculty of Medecin and the Laboratory of Management and Cognition at the University of Toulouse III. The project's objective was to evaluate the performance of the MGT Service at the Midi Pyrénées region and their efficiency in taking elder patients in charge [Arthus et al., 2009].

As we will show in the corse of this chapter, the success of a MGT mission and its tasks is highly related to the ability of the team members to access information sources from anywhere, at anytime and anyhow. Nevertheless, the transparency required for distant access and for facing emergent cases and critical real-time scenarios might contradict with the rigid security constraints that manage access to healthcare systems. Therefore, using an adaptive access control model that enables context and situation aware decision-making seems to be unavoidable.

The chapter is organized as follows: first, we'll expose the main characteristics that motivated the pervasive vision of healthcare systems. Next, we'll present a brief state of the art about the importance of security in the healthcare domain. Finally, we'll present a detailed usecase study that tackles the accessibility challenges that meet the MGT members during their missions and presents an implemented solution that reflects our adaptive context and situation aware decision-making vision.

6.2 The Pervasive Characteristics of Healthcare Systems

Nowadays, healthcare systems are heading towards the adoption of new intelligent technologies to guarantee better generation, processing, archiving and consultation of medical data. The objective behind this adoption is to improve the quality of medical services, system's efficiency and to provide real-time accessibility in different contexts.

As information systems are becoming open systems and aiming to be more responsive and adaptive to user mobility, new medical and healthcare systems have promoted a collaborative usage of patient's medical information through a distributed

[2] Contrat HAS/CNSA, n°07/0008, INSERM U558 – Département de Santé publique faculté de médecine de Toulouse et Laboratoire Gestion et Cognition (EA 2048) - Université Toulouse III - Paul Sabatier.

network; where the patient record has become a virtual digital record that can be treated by different users in different physical locations.

The evolution of a central ***Electronic Health Record*** that can be accessed from different systems has helped to avoid record redundancy and helped to acquire online, up-to-date patient information whenever needed. This has highly affected the quality of patient treatment and the time consumed to retrieve the patient record.

The continuous evolution of healthcare systems towards the integration of intelligent technologies and mobile services has highlighted the need for transforming classical healthcare information systems into ***pervasive healthcare information systems*** where users would acquire seamless accessibility to medical resources at anytime, anywhere and anyhow [Al Kukhun and Sèdes, 2008].

Classical healthcare systems have always considered the patient as the main entity of their services but with the continuous evolution and decentralization of healthcare resources and the different services and users interacting with the patient record, pervasive healthcare systems are directing towards providing transparent interactions in a user-centered manner through service-oriented technologies, where the access to the medical record can be any healthcare agent, the patient himself or any member of his family. In fig. 6.1 we illustrate the importance of ensuring a homogeneous interoperable interaction between the various healthcare subsystems dedicated to serve the patient (including the MGT).

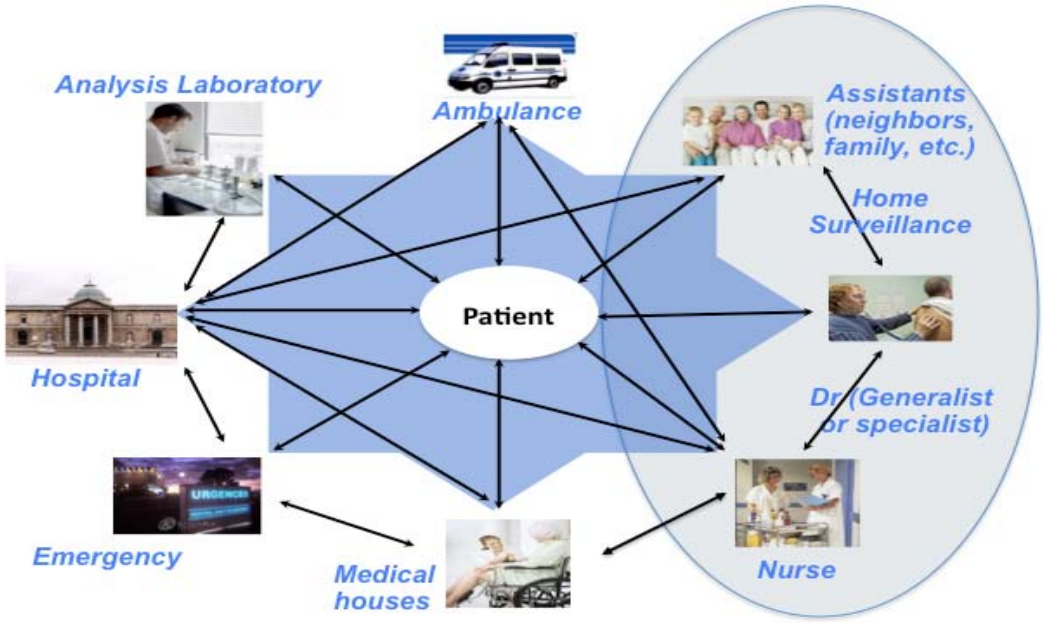


Figure 6.1: The pervasive interaction between healthcare subcomponents

The evolutionary nature of medical data is considered as an important feature since it doesn't only reflect the evolution of a patient's situation through time but it also shows the different interpretations (interventions) that were done by each member of the medical team. Knowing that a great part of the patient's medical data is generated and processed in real-time, the administration of access rights should be centralized and managed in a way that would ensure the system's integrity and provide reliable decision-making.

In order to share patient information, medical centers tend to perform record transactions or to access medical records of external resources. Ensuring the patient privacy in such operations is very important and therefore any record transfer should be justified and any record access should only be allowed in particular purposes and to particular agent.

The confidentiality and sensitivity of medical data has justified the imposition of rigid security procedures on their access and the fact that they should be conserved in their original resources within the different healthcare sub systems (hospitals, analysis laboratories, medical cabinets, etc.). This decentralization of resources influences the management of access rights that is generally implemented through the RBAC model [Ferraiolo et al., 1992] due to it's ability to adapt access rights according to a user's role or position within the system.

Since medical records contain several types of multimedia content (text, image, video, etc.), they are usually represented in semi-structured XML documents. The eXtensible Markup Language is considered as a standard language that facilitates data sharing and exchange. XML's power lies in its ability to describe the content and structure of a healthcare record in a simple textual format.

The simplicity, expressiveness and Interoperability of XML have facilitated its deployment in the exchange of medical data and have motivated the creation of a specialized standard for exchanging medical information [HL7, 1994].

Considering the sensitivity of data in such exchange, access control becomes a highly needed characteristic especially in the case of accessing distributed resources with different gouverning authorities. For this mission, we employ XACML eXtensible Access Control Markup Language [OASIS, 2003], [OASIS, 2005a] - a service-oriented data exchange standard that supports centralized decision-making within distributed environments and allows the expression of contextual attributes. Using this standard, access rights are translated in the form of XML policies where medical laws are transformed into semi-structured rules.

XACML that is an XML-based standard that provides an efficient decision making mechanism dedicated to manage distributed access policies and within service-oriented environments. Next, we'll introduce the main laws that gouvern medical data access and exchange.

6.3 The importance of security in healthcare systems

Medical data are considered by law as sensitive, private and confidential information. Thus, they need to be highly protected and their access needs to be strictly managed. Various international and national laws have been proposed for the protection of medical data such as: the declaration of Helsinki [Helsinki, 1964], the Privacy Act [Privacy Act, 1974], the HIPAA (Health Insurance Portability and Accountability) Act [HIPPA, 1996] and the act of patient rights and healthcare systems quality in France [Loi 2003-303, 2003].

The privacy of patient data justifies their storage in their original systems and imposes more accessibility constraints particularly in the case of a mobile or distant consultation. Consequently, accessing data in any healthcare system must respect the principles of the patient's personal data. Therefore, these data are not equally available

to all members of the medical team and are often limited to the needs of the task being performed by the user.

In pervasive healthcare systems, managing data access and defining security constraints becomes more demanding since the access does not only depend on the user's role or the consultation time but it's also attached to his contextual constraints such as his location, the device he's using, the network bandwidth with which he's connected to the system, etc.

Finally, as healthcare systems provide precious services that deal with people lives, we emphasize the fact that they are considered as ***critical systems*** and that the definition of quality assurance in such systems is different than other classical profit-based systems; in critical systems, user satisfaction becomes of minor importance if service delivery is endangered especially in real-time life-threatening situations (where extreme solutions that threaten the patient privacy are used such as the «Break-Glass» solution), while in classical systems; user satisfaction has equal importance to service delivery and precision.

Next, we'll present the accessibility challenges that meet the MGT members as an application in the healthcare domain, where access to healthcare resources has to be managed in a way that takes different aspects into consideration such as the user's role, his contextual attributes and the situation at which he's consulting the system.

6.4 Application on Mobile Geriatric Teams

6.4.1 The Nature of a MGT Mission

A typical MGT Intervention procedure takes place after receiving an "action request" fax sent by a specialized service. By analyzing the types of requests forming the MGT missions, we distinguish two activity modes:

- ***Intra-hospital Activity:*** where the MGT members mobilize from the geriatrics department to other departments in the same hospital in order to perform an intervention.
- ***External Activity***: where the MGT members mobilize to perform a service outside the hospital such as in retirement homes, nursing homes and in ambulatory care of frail elderly patients, etc.

The activity is normally required to offer a comprehensive treatment to elder patients. Therefore, the deplacement involves the whole team – composed of a secretary, nurse, nursing assistant and physician (generalist or specialist) – that will need to obtain access to the different types of medical resources in order to build a geriatric events record.

Considering the mobile characteristic of the team, we highlight the importance of ensuring accessibility to healthcare resources from any department/ location, at any time, using any machine and any connectivity channel as an important criterion for the functionality of this mobile team.

6.4.2 Accessibility and security challenges in retrieving and transforming patient information within the MGT

In this section, we highlight the importance of providing an effective information system that could serve the MGT members in consulting the patient record during an intervention.

The collaborative nature of a MGT intervention involves an evolutive creation of the patient record where each team member can consult and access patient data from the information system (according to his/ her access rights) in order to interact by adding observations, annotations, comments and other data.

In the current system, the information gathered and produced during a MGT intervention are transmitted from a member to another at the end of their task either by hand through standard predesigned half-filled forms (in the case of external activity) or with forms printed from the computer system (in the case of an intra-hospital activity).

To perform better consultations and achieve a smoother information flow, the geriatric event file must be completed in a cooperative and interactive manner. The specialist should be able to consult the patient's situation and access real-time datastreams that are extracted from monitoring machines and many more techniques to update his record after examination.

For this reason, we stress the importance of providing a pervasive system that would improve the consultation of patient data and enable better information handling in a collaborative and interactive manner.

A pervasive evolution of the current healthcare information system would influence the quality of service provided by the MGT when being able to treat elder patients from anywhere, anyhow and at any time.

Having a pervasive healthcare system in the disposal of the MGT would resolve the complexity of accessing patient records during mobile missions. Enabling seamless access to data sources would be beneficial to ensure an efficient interaction and data flow within the team members.

Meanwhile, the seameless accessibility that we aim to provide should respect the sensitive nature of the medical resources and personal data consulted during a MGT mission which should conform to the policies defined for protecting the privacy and confidentiality of patient data.

Our research has covered the pervasive vision at the access control level where authorizations should take into consideration the mobile nature of the MGT missions and the need to provide a level of flexibility for dealing with real-time consultation needs within external missions and emergency situations.

The objective is to balance between the seameless accessibility desired by the MGT members and the rigid security constraints imposed by the healthcare system. Usually, when an access denial takes place during an important consultation in a mission, the members go towards employing the «Break-Glass» option.

In a mobile context and a pervasive environment, the probability of attaining access denials increases due to the connectivity of access decisions to the user’s dynamic contextual constraints. In order to prevent unjustified usage of the extreme

«Break-Glass» option, we have envisioned a pervasive access management system that can offer flexible alternative-based decision-making with minimal violation risks.

Next, we'll present the implementation of the proposed solution, which employs the PS-RBAC model for realizing context and situation-aware permission assignments and the PSQRS architecture to perform adaptive decision-making and to find alternative similarity-based solutions through XACML query rewriting.

6.4.3 The Implementation of an adaptive access control system

The prototype presented throughout this section highlights the importance of providing an effective information system in the service of the MGT. Our implementation provides an access control management system that performs decision-making to access healthcare ressources according to the following factors: (i) the user's role, (ii) his contextual information (location, time, machine, connection, etc.) and (iii) the situation at which he's consulting the system (an emergency, distant access, access to non-familiar databases, etc.).

6.4.3.1 Policy Administration Panel

To facilitate the adoption of the XACML standard, we have provided an "Administrator panel" component dedicated for managing access rights within the MGT service. This management involves a simple user-friendly interface that enables the administrator to define context-aware access permissions to healthcare resources and automatically translates them into XACML policies, see fig. 6.2.

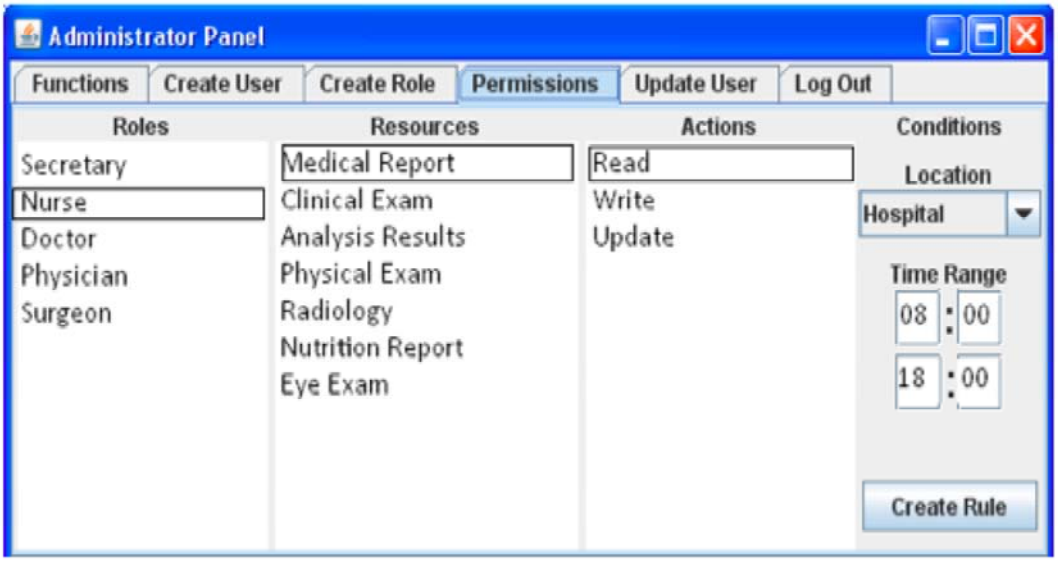


Figure 6.2: Context-aware permission assignment using the administrator panel

As demonstrated and following the PS-RBAC model, a user is represented by a role that allows him to perform different actions (e.g., read, write, update) on different resources according to certain contextual constraints (here: location and duration). The result of this assignment would generate an XML file containing the XACML access policy, see Fig. 6.3.

```
<Policy PolicyId="GeneratedPolicy"  RuleCombiningAlgId="urn:oasis:
names:tc:xacml:1.0:rule-combining-algorithm:ordered-permit-overrides">
   <Target>
      <Subjects>      <AnySubject/>     </Subjects>
      <Resources>   <AnyResource/>  </Resources>
      <Actions>        <AnyAction/>     </Actions>
   </Target>
   <Rule RuleId="urn:oasis:names:tc:xacml:1.0:hospital-system:rule" Effect="Permit">
      <Target>
         <Subjects>
            <Subject>
               <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-
                  Role" DataType="http://www.w3.org/2001/XMLSchema#string”>
                  <AttributeValue> Nurse </AttributeValue></Attribute>
            </Subject>  </Subjects>
         <Resources>
          <Resource>
            <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#string">
               <AttributeValue>Medical_Report.xml</AttributeValue> </Attribute>
         </Resource>      </Resources>
         <Actions>
          <Action>
            <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string">
               <AttributeValue>Read</AttributeValue> </Attribute>
         </Action>   </Actions>   </Target>
      <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
         <Apply FunctionId="urn:oasis:names:tc:xacml:
                              1.0:function:time-greater-than-or-equal">
            <Attribute> <AttributeValue>08.00.00.495000000+02:00 </AttributeValue>
            </Attribute></Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-
                              equal">
            <Attribute><AttributeValue>18.00.00.495000000+02:00 </AttributeValue>
            </Attribute>     </Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <Attribute  AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-loc"
               DataType="http://www.w3.org/2001/XMLSchema#string">
               <AttributeValue>Hospital</AttributeValue> </Attribute>
         </Apply>
      </Condition>
   </Rule>
   <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

Figure 6.3: XACML policy resulting from translating the permission assignment within the administrator panel

6.4.3.2 Consultation Example

The prototype has a dedicated portal to system users (doctors, nurses, etc.) who consult the system by launching different queries within different contexts via a special interface; an example of a user access demand is illustrated in fig. 6.4.

The proposed system operates in an intelligent pervasive environment where the patient number can be extracted from the RFID chip of his insurance card and the contextual constraints of the MGT member consulting the system are extracted explicitly or implicitly:

- Time is always defined implicitly.
- In the case of a mobile mission, the user's location is detected through the GPS of his machine and then verified against the patient's address that is mentioned in his record.
- In the case of an intra-hospital activity, the verification is done through the hospital's address or by the usage of the internal network.

However, in the case of a lack of contextual information during a consultation, these constraints can also be specified explicitly by the user.

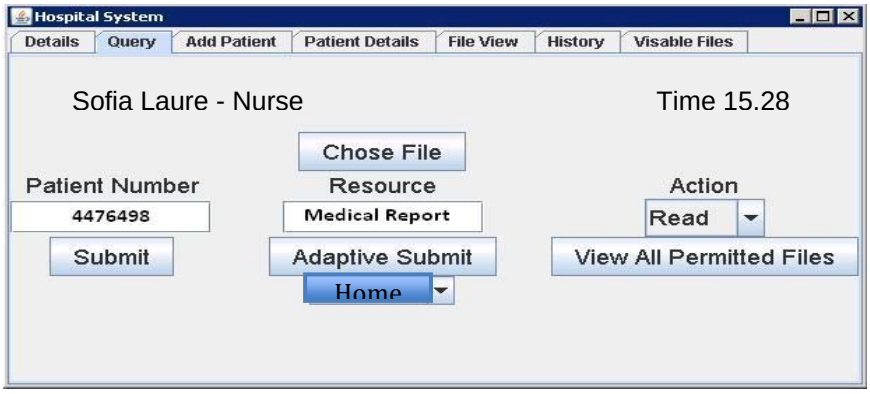


Figure 6.4: Example of an access request performed using our system

A user's access demand is interpreted following the PSQRS architecture and translated into an XACML request. Fig. 6.5 shows a simplified version of the XACML request resulting from the translation of the access demand launched in fig 6.4 above. The subject (nurse at the MGT – Toulouse Hospital) requests access to a resource (medical report of a patient) in the following contextual attributes:

- Time: 3:28 p.m.
- Location: the patient's home.

The result of this consultation during this mobile mission situation will be an access denial due to the current context of the user (located at the patient's home), which is considered as a privacy threat in the system. In such contexts, employing the adaptive query rewriting mechanism offerefd by the PSQRS would be beneficial to find alternative resources that could be authorized within the same context and who would offer the user relevant information to his consultation such as the lab analysis and radiology files. These similar resources are attained using the Similarity Provider component of the architecture.

```
<Request>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               DataType="http://www.w3.org/2001/XMLSchema#string”>
      <AttributeValue> Sonia Laure </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-Role"
               DataType="http://www.w3.org/2001/XMLSchema#string”>
      <AttributeValue> Nurse </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-loc"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> Patient House </AttributeValue>
    </Attribute>
    <Attribute  AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-time"
                DataType="http://www.w3.org/2001/XMLSchema#time">
      <AttributeValue> 15.28.49.495000000+02:00 </AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> Medical Report </AttributeValue>  </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> read </AttributeValue>
    </Attribute>
  </Action>
  <Environment/>
</Request>
```

Figure 6.5: The XACML translation of the nurse request

6.5 Conclusion

A growing need arises for a pervasive evolution within modern healthcare systems where access to medical sources can be attained at anytime, anywhere and anyhow. Meanwhile, this accessibility needs to be governed in a fine-grained basis as healthcare sources are owned and administrated by different authorities.

The dynamic nature of the MGT missions and the richness of important real-time consulting situations have formed a challenge to preserving access control. Providing the MGT members with seamless accessibility is vital especially during their mobile missions and in critical situations. Meanwhile, respecting access control constraints is an essential building block in ensuring a secure management for healthcare sources. The need for providing a balanced flexibility has formed an ideal case study for the validation of our context and situation-aware access control model.

In this chapter, we have implemented an adaptive access management system that is based on the PS-RBAC model and the PSQRS architecture. The implementation provides user-friendly interfaces that enable administrators to define access permissions then translates them into XACML access policies and enable users to launch several queries to access healthcare sources then translates these demands into XACML requests in order to enable effective XACML based decision making.

The system enables context and situation-aware access decisions and confronts the access denials that meet the MGT members during their mobile missions by proposing flexible accessibility to alternatives or similar authorized resources that don't threaten the system's security or integrity.

Finally, we highlight that the success of our alternative-based methodology depends on the availability of similar sources that are obtained through a keyword similarity search. Future works are envisioned to increase the chances of finding more options through a more profound similarity search based on medical ontologies.

Chapter 7:

Situation–Aware Accessibility for Crisis Management within Avionic Systems

7.1 Introduction

Through this chapter, we present a second usecase for the validation of our proposal where we expose the challenges that face access management to information sources within Avionic Information Systems. The works presented are based on a research conducted for the GEODESIE[3] project and that provided an analytic study about the evolution of access control services [Al Kukhun and Sèdes, 2009b]. The project aimed for ensuring optimized and secure data management within Embedded Systems.

An Avionic Information System is composed of a group of resources that are distributed within different sub-components, managed by different authorities, administrated according to different constraints with different access levels that are defined according to different levels of confidentiality.

In this chapter, we aim to enhance accessibility to avionic sources during critical real-time situations. We start by introducing the characteristics and components of avionic IS. Next, we point out the accessibility challenges that face resource management within the classical avionic IS and we propose to confront them using an adaptive pervasive vision that applies our proposed PS-RBAC model and the PSQRS architecture.

7.2 The characteristics of Avionic Information Systems

Ensuring transparent and efficient accessibility to avionic sources is an indispensable criterion especially in real-time situations and in emergent cases. Meanwhile, attaining this accessibility is a challenging mission due to: (i) the distribution of avionic data sources that are usually managed in a decentralized manner through different DataBase Management Systems DBMSs (the avionics system, flight information system, open-world system and the cabin system), (ii) the heterogeneity of the security levels administrating these sources and (iii) the decentralization of the authorities managing these sources.

The distribution of access control policies is critical as they are defined by different aircraft certification authorities and sub-components within Avionic Systems. These policies become more complex in a pervasive context as they don't only tie access to the user's identity and role but also to his contextual attributes.

Thus, a growing need arises for ensuring access when needed while providing a homogeneous, logically centralized data management that could ensure better data distribution and communication (exchange) between the diiferent avionic sub systems.

[3] A joint industrial project led by Airbus, with ONERA Centre de Toulouse and IRIT treating modeling aspects for GEODESIE system engineering.

The challenge of avionic data management lies also in the distribution of the resources within incompatible worlds from an access control level perspective and in the reconciliation of the different worlds vision.

As illustrated in fig. 7.1, an avionic IS is composed of three main sub-components that acquire heterogeneous levels of security:

1. ***Cabin***: Although open to the world, via digital radio links, the whole onboard system is designed to be highly secure and totally locked. In order to ensure the operational functionality of the system at any time and to avoid the risk of failure, the system relies on a redundant architecture where the IS is duplicated. This criterion ensures the safety and the security of the system. The IS is based on an embedded system designed to be extremely safe, both in terms of computer security and operational availability.

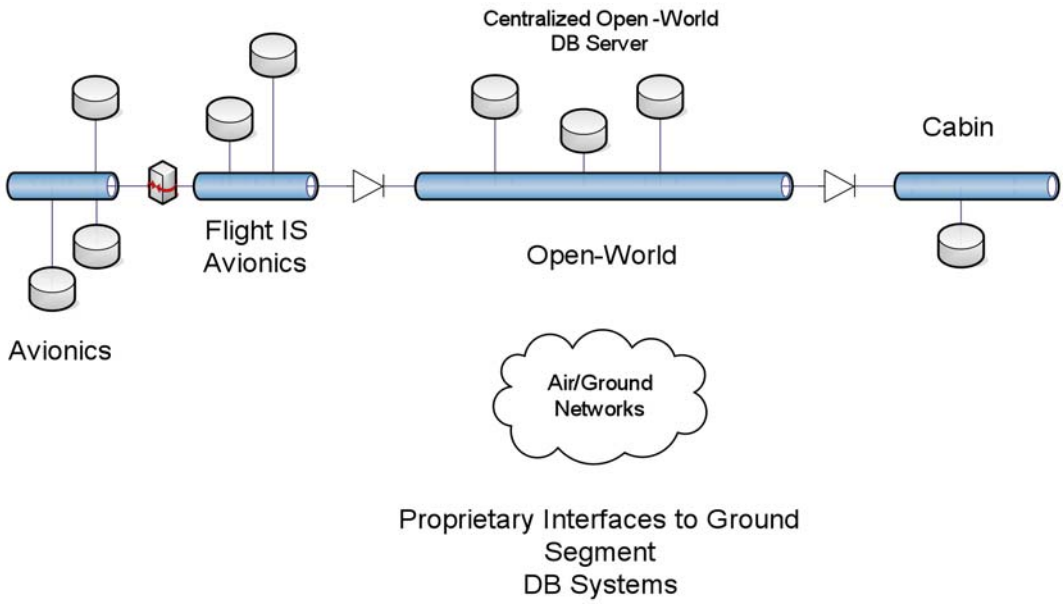


Figure 7.1: A350 Database Architecture

2. ***Open World***: This component contains an IS that is completely open. It collects, centralizes and compiles all the data related to the flight on a single system and provides external communication means, data calculation and storage capacities. This modular central system also hosts applications unique to Airbus and airline companies, that deal with the actual operation of the plane all the way through to the services offered to passengers (e.g., onboard electronic documentation, navigation diagrams, performance calculations, flight logs, etc.).
3. ***Avionics & Flight Avionics IS***: This component is composed of two parts:

   The ***Avionics*** part has a dedicated IS for managing data that are strictly in the avionics domain. It is highly secure and has a high level of confidence. This section must be connected to the ground (to make exchanges, updates, etc.). Meanwhile, this connectivity must respect the access constraints and security levels that vary depending on the service requesting access (mechanical service, thermal service, etc.).

The ***Flight Avionics IS*** part contains the information and documents relating to flight operations. It is freely and openly connected to the outside world (in-flight entertainment system, wireless internet connections, etc.).

7.2 The classical ressource management of avionic IS

Taking a close look at the architecture of the A350 information system, we can distinguish many challenges standing in the face of transparency and efficient real-time accessibility to information sources.

Access to information sources should be highly efficient in a way that users won't feel the distribution of data and the fact that they are at different physical locations with variant security and accessibility levels (according to their confidentiality and importance level).

As the current architecture lacks a standard storage protocol, the system employs a data replication technique in order to facilitate access to needed resources without saturating the system. But the drawback of this option can result in overloading the system with redundant data sources that might not always be up-to-date.

This multi-factored distribution raises the complexity of data sharing and highlights the importance of ensuring secure interaction through the various system components and via the input/output services that are dedicated for better access management.

7.3 A pervasive vision for ensuring adaptive accessibility to information sources within avionic IS

An analytical study of the features required to ensure proper access to the users of avionic IS has led us to consider them as critical systems where users need to obtain access to information sources from anywhere, anytime and anyhow, which rhymes with the accessibility needs of pervasive information systems PIS.

Moreover, Avionic IS have many pervasive characteristics, such as:

- The distribution of their data sources.
- The need to manage real-time generated data streams.
- The evolutive nature of avionic data.
- Manage access to highly secure data.
- Data is gouverned with multi-distributive access privileges (policies are distributed within different sub components and managed by different certifying authorities).
- The need for context-aware access management.
- The need for situation-aware access management.

The absence of a dedicated standard or protocol for data storage justifies the use of the existing solution in which data sources are replicated on multiple subsystems in order to facilitate their access.

In the lack of a generic coherent data management scheme, this data replication solution has resulted in increasing the complexity, which is related to: (i) data storage

overhead, (ii) the difficulty of tracing updates and (iii) a data consistency problem resulting from these updates especially with the real-time evolution (typical problem associated with data duplication and redundancy).

Our contribution aims to solve the accessibility challenge within real-time consultation situations. Thus, we apply an adaptive access control scheme using the PS-RBAC model. Based on the RBAC model, the PS-RBAC model provides context and situation-aware access decisions that can manage access to distributed data sources and help in differentiating and balancing between the different security levels (defined by the various sub systems managing avionic sources).

At the application level, we propose to achieve this adaptive accessibility by employing the PSQRS architecture. Based on the XACML standard, PSQRS can cope with the challenging distribution of access privileges and their volatility. This architecture can ensure efficient data sharing and access management along with centralized decision-making that assigns access permissions based on distributed access privileges that belong to different authorities.

The sensitivity of avionic data sources justifies the use of a rigid security approach. Meanwhile, these systems are also designed to cope with emergency situations where the denial of an access request may risk the lives of passengers.

We present in fig. 7.2 a proposed decision-making procedure that our system can follow when encountered with an access request. The flowchart describes the various options that can be taken into consideration to face the emergencies that may occur in avionic IS. The choice of the flexibility of access is related to the sensitivity of the situation encountered in real time.

Level 1: Rigid adaptation based on predefined solutions

The solutions offered at this level are defined following the pre-designed protocols existing to face the critical cases of avionic IS. In this kind of adaptation, the occurrence of a certain type of critical scenario triggers a method that (i) allows the user to access certain procedures/ unauthorized resources in order to react and face the situation and (ii) allows the system to apply predefined automatic solutions.

Level 2: Assisted adaptation

This type of adaptation can take place when the user demanding access to avionic sources during an urgent situation is faced with an access denial. The refusal may be due to a technical problem occurring to the system or due to the security constraints gouverning access according to the user's role or to his location (outside the contextual area defined by the access privileges managing the avionic sources).

Depending on the importance of the situation, the system interferes and applies an assisted adaptive procedure that transfers the refused access request to a more authorized user who can help in managing the situation by accessing the necessary sources.

Level 3: Adaptive ad-hoc search based on alternatives

We offer this type of adaptation to address situations where the existing pre-designed solutions cannot help the user faced with an access denial during an emergency situation (e.g., sub system breakdown, connectivity issues, etc.).

Using the PSQRS architecture an access denial would be faced by an access request reformulation that would enable users to access alternative resources.

Level 4: Break-Glass Option

The avionic system applies this solution on the occurrence of an unpredicted crisis that couldn't be managed by the different means mentioned above. The «Break-Glass» solution consists of applying a temporal flexibility that enables depassing the imposed access control constraints in order to confront the situation. The risk of using this flexible security lies in revealing the system's security but in an extreme situation, the risk of violation is uncomparable when compared to the risk of loosing the lives of passengers.

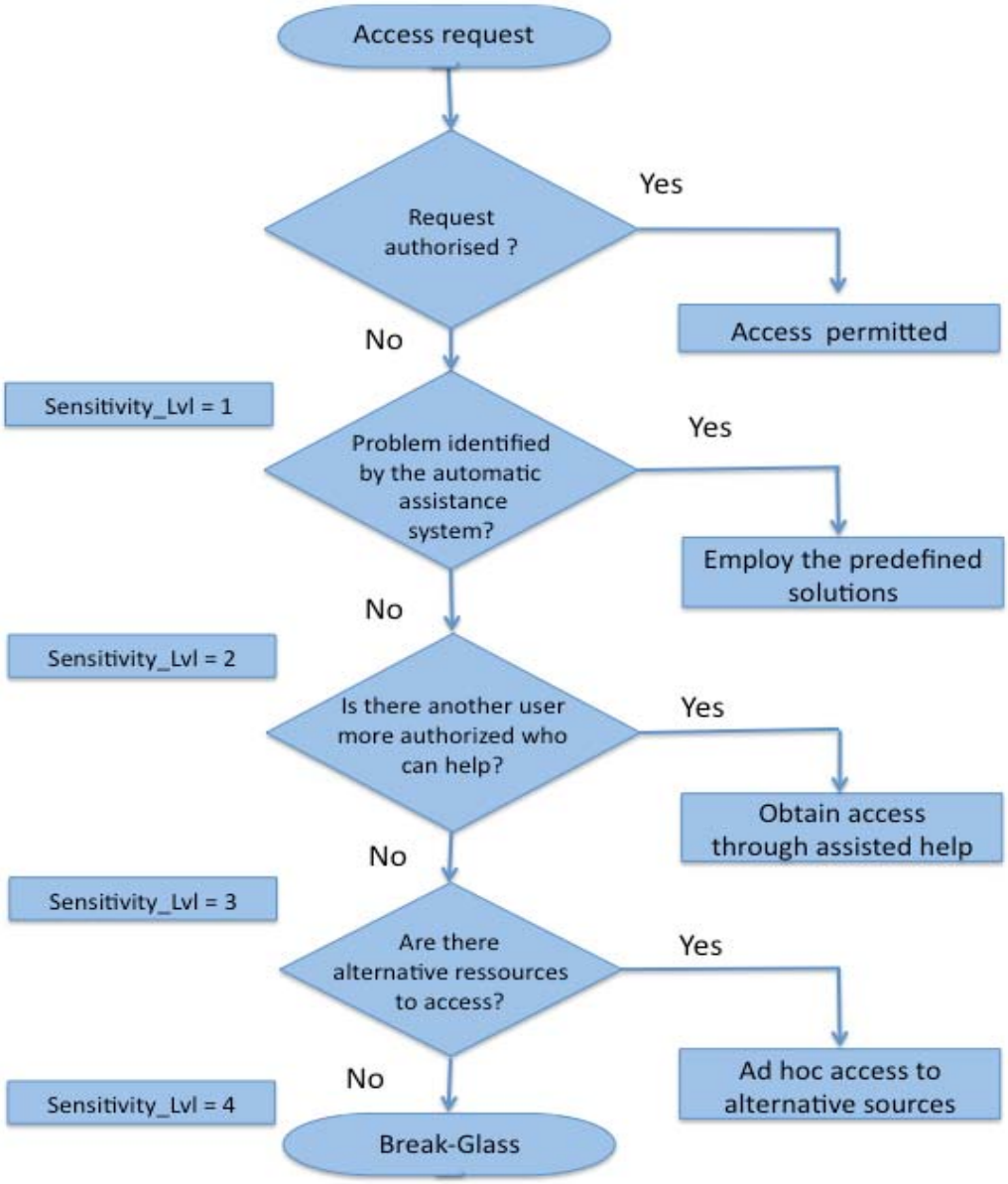


Figure 7.2:The processing of an access request in a situation

7.4 Conclusion

Through this chapter, we have employed our proposal in facing the problems of resource management and data access needed that are highly needed within avionic systems. The proposed solution highlights the importance of providing transparent connectivity between the various avionic sub-components and of enhancing data storage, sharing and update taking into consideration the evolutive nature of avionic sources.

Managing Accessibility is a challenging mission in avionic IS. On one hand, it should respect the access control constraints and ensure a high security level; on the other hand, it should provide efficient solutions to confront emergent situations where the denial of an access request may risk the lives of passengers.

After analyzing the different solutions that avionic systems offer upon the occurrence of an urgent situation, we have identified some cases that can be faced by applying our adaptive and alternative-based decision-making procedure. Thus, we have joined our proposal along with the existing solutions (through the use of the PS-RBAC model and the associated PSQRS query rewriting system) in order to increase the chances of facing access denials during emergent situations by proposing alternative pertinent sources.

Chapter 8:

Providing Adaptive Secure Querying to a Video Surveillance Management System

8.1 Introduction

In this chapter, we present a third case study to validate our proposal. The application domain is a video surveillance data management system. Our goal is to resolve access denial problems that encounter users requesting access to resources. This study is based on works realized in the LINDO[4] project (Large Scale Distributed Indexing of Multimedia Objects).

The chapter starts by introducing the LINDO project and it's query processing procedure in section 8.3. In section 8.4, we apply an access control layer on top of the LINDO system and explore the access challenges that take place during different querying contexts. Next, in section 8.5, we apply an adaptive pervasive vision to face access control challenges in a video surveillance system. Finally, a detailed scenario of employing our adaptive situation-aware query rewriting is presented in section 8.6.

8.2 Background of the LINDO project

The European project LINDO aims to develop a generic architecture, in which not only the multimedia data storage is distributed, but also the indexing process is performed on different remote servers that are heterogeneous in type, geographically dispersed and have varying capacities.

Instead of moving the multimedia contents and their associated metadata (attained by the indexation process) to be processed within central servers when replying to a user query, an alternative solution has been followed where the indexing routines are directly executed on distant servers where the contents are acquired. Consequently, only the necessary information meeting the request will be transferred to the user, see Fig. 8.1.

[4] http://www.lindo-itea.eu/

Figure 8.1: Les challenges du traitement d'une requête d'un utilisateur

This strategy of distributed indexing and storage of multimedia contents and their associated metadata is advantageous as it aims to avoid the various disadvantages of centralized processing such as:

- ***The slow query processing***: the processing of a query against the whole metadata existing within the system would most likely overload the central server, especially when processing complex queries or multiple requests simultaneously.
- ***Overloading the network bandwidth***: when transferring all multimedia content and their metadata to the central server.
- ***The system centralization***: this could raise problems like fault resistance, if the central server is no longer available the metadata collection needs to be recomputed and resent to the server. Moreover, in the case of a dynamic information system, the central server update would be very expensive.
- ***The violation of the access rights defined on the contents***: some metadata shouldn't be stored on the central server for privacy issues.

8.3 Query processing in LINDO system

To ensure the efficiency of the retrieval process, the research works of the team were interested in the indexing process. The objective is to improve query processing by applying filtering algorithms and reformulating queries based on metadata semantic links [Laborie et al., 2009], [Brut et al., 2011].

The main goal is to attain a quick response to a request thus; the proposed solution was to transfer only a concise version of metadata (a summary) describing the multimedia contents to the central server. This summary, extracted in XML format, is periodically updated when new metadata is generated by indexing algorithms on remote servers. Therefore, the central server can be used to respond directly to general queries.

When the need expressed in the user query is not satisfied with the metadata generated by the generic algorithms (placed at the central server), the system can respond and offer the user the ability to execute explicit algorithms that can perform a more detailed treatment and recover more relevant results. These specific algorithms are deployed on remote servers where the contents are acquired.

We emphasize the importance of the effect of the choice of an indexing algorithm on the results retrieved by the system. In the following, we provide two examples that show the difference in level of detail offered by each indexing type (implicit or explicit).

The Fig. 8.2 shows an example of results obtained from the real time indexing performed during the acquisition of a video surveillance fragment. This processing uses implicit indexing algorithms that store the results on the central server. These indexing algorithms distinguish two acquisition contexts: indoor and outdoor.

	Indoor	**Outdoor**
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of	- Number of people, number of vehicles - Main color of the people upper part.

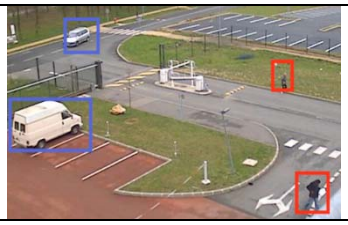
	the people	- Main color of vehicles
		

Figure 8.2: Example of information extracted by implicit algorithms

The Fig. 8.3 shows the results produced after using explicit indexing algorithms. These algorithms can provide more specific details regarding the content. A more profound processing is performed, generating different metadata describing the same segment of the video. Such algorithms are deployed on remote servers and are selected based on user queries seeking for more detailed answers.

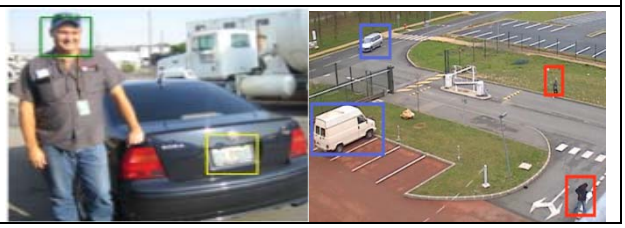
	Indoor	**Outdoor**
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the person - Face recognition - voice recognition & speech-to-text	- Number of people, number of vehicles - Main color of the person's upper part. - Main color of vehicles - Car plate number - Face recognition
		

Figure 8.3: Exemple d'informations récupérées par le traitement d'algorithmes explicites

Choosing an explicit indexing algorithm can provide more accurate results to users, but this can eventually have a very high processing cost (depending on the level of granularity provided). For this reason, the system often restricts the possibility of allowing a user to customize the query using more explicit specific algorithms.

In the following, we emphasize that the sensitivity of the content released is related to the level of detail provided in explicit algorithms (located on remote servers). This sensitivity is strongly related to data protection and is considered as another reason that can prevent a user to explicitly select an algorithm that can customize its search.

8.3 The accessibility challenges to be considered when adding an access control layer

Our objective is to ensure an efficient information retrieval process despite the security challenges, see Fig. 8.4. The sensitivity of the video surveillance contents and the privacy protection law (anonymity requirments) imposed on these contents justify the need for applying an access control scheme on top of the LINDO architecture. This manages and customizes access depending on the user's role. The security layer is responsible for managing:

1. The access rights granted to users or services demanding access to the video surveillance data sources that vary not only according to their role but also in terms of their querying context (time, location, etc.).
2. The access rights for using the explicit indexing algorithms: the risk of disclosing personal or confidential information arises with the granularity of the level of detail sought and provided by the algorithm.

We highlight the fact that the lack of responses returned to user's queries might not be due to the lack of results existing within the system but due to access restrictions imposed by the security layer. This is the point where the search for adaptive responses and alternative solutions seems to be important.

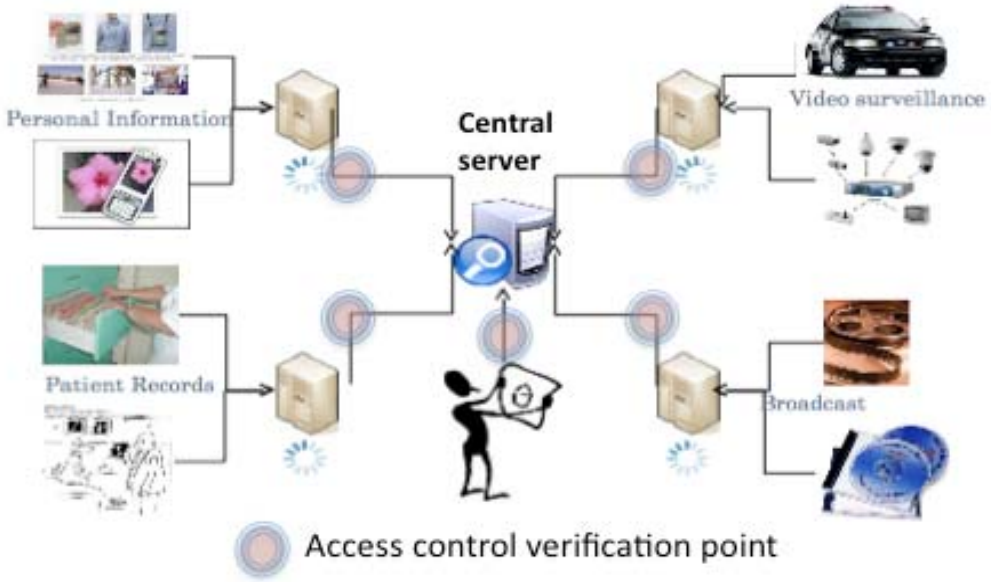


Figure 8.4: The challenges of query processing with the access control layer

The lack of access to the needed resources may also be due to causes that are far from the security domain and related to system performance and retrieved content presentation problems. These causes can come from various problems such as:

- Technical problems affecting the capturing devices (hardware dysfunctionality) or remote servers;
- Problems Related to contextual constraints indexing algorithms using context on remote servers. The level of accuracy of the algorithm influences the processing time and cost and may overload the system (Processor, OS);
- Problems concerning the volume of the document returned as a result on the user's device (mobile device);
- Problems related to user's context (machine, connectivity, storage space, etc.) preventing the good reception of the content.
- Problems regarding the format compatibility associated with the device (not supported by the machine), etc..

Next, we propose a solution that aims to prevent denial of access due to reasons mentioned above and to provide alternative access way.

8.4 The LINDO system seen as a Pervasive Information System PIS

As we have already mentioned, the lack of relevant answers returned to the user may not be caused by the lack of relevant content but is rather related to access restrictions imposed on some resources and on the indexing algorithms or to the difficulty of displaying the result due to the current contextual constraints of the user.

Therefore, our goal is to find solutions that can ensure greater accessibility to requested resources at any time, from anywhere and anyhow.

In the LINDO system, the complexity of query processing is often due to constraints related to access control. Hence, this system represents an ideal case of application for validating our proposal, particularly if one takes into account the pervasive characteristics of LINDO such as:

- the distribution of data sources
- the variation of the authorities managing these resources
- the evolving nature of these resources (generated in real time)
- the sensitivity and confidentiality of the content of these resources
- the diversity of contextual information
- the distribution of the indexing process performed by a variety of indexing algorithms
- the execution of access requests in real time
- the importance level of obtaining reactive solutions in critical situations

8.5 The adaptation of access control in a video surveillance system

The sensitivity of the video surveillance data and the fact that they are often gouverned by privacy laws that require the respect of persons anonymity justify the application of restrictions that filter the access privileges based on the user's role, when consulting the resources (depending on access rights specified by the RBAC model).

Since video surveillance systems are used to handle the situations in real time, a «Break-Glass» option is often included to surpass emergencies and important situations. The «Break-Glass» mechanism gives the user the ability to overcome the complexity caused by security constraints, and thus to react in a better way. This flexibility helps to overcome the multimedia content access restrictions and facilitates the explicit utilization of indexing algorithms.

Between the respect of the rigid access decisions adopted, by default, by the system and the extreme flexibility of the «Break-Glass» option, there exists situations where users need moderate solutions and flexible access decisions. This is where our proposal may intervene to provide access alternatives.

The access control relaxation that we propose to carry out [Al Kukhun et al., 2012b] does not affect the access rights to the video contents, but it ensures their respect and their support. It applies the flexibility and the adaptation of the decision making at two main functionalities:

(i) The choice of using explicit indexing algorithms (located on remote servers).

(ii) The presentation of the video contents (the identity of filmed persons is protected by privacy laws that assure their anonymity).

The success of the proposed adaptive procedure is tied to the richness of functionalities offered by the explicit indexing algorithms and by the adaptation of the presentation that can help the user to overcome the complexity of the encountered situations.

Therefore, in the case of confronting an access denial, a lack of response or the retrieval of unsatisfactory results, our solution can provide the user with the ability to customize the research methodology in order to receive solutions adapted to its context and needs.

The PS-RBAC model and the PSQRS architecture realize the adaptation of access decisions by taking into account not only the user's role, but also his contextual attributes and the importance level of the situation at which he consults the system. Such a solution will improve the service quality on the user's side without threatening the security or the integrity of the system.

8.6 Application Scenario

We present in this section an example where the implementation of our proposal is used to overcome the lack of answers provided by the system. The scenario treats the case of a forgotten item in a metro station. As we will illustrate next, in the case of facing difficulties to obtain access to needed video surveillance resources, the system will modify the query processing procedure and will adapt access decisions according to the level of importance of the querying situation.

Taking the metro from the « Trocadéro » station to « Place d'Italie » at 14:15, Helen has forgotten her red bag on a bench at the waiting line. As soon as she realized, she went out to report the problem at the information counter.

A typical treatment of such situations goes through the customer service agent who would open a lost object file, take the descriptions and transmit them to the security officer on site. The security agent will follow different steps in order to find the object; he will check if the object has already been found or returned to the lost and found office by someone. Otherwise, he will try to see the video surveillance system to check if the object is still in the same location.

8.6.1 Typical query processing performed by the LINDO system

Fig. 8.5 shows the typical interpretation performed by the information retrieval system provided by LINDO. The launched request will be processed and parsed to extract the main keywords that are then reformulated in the form of an XML query.

The distributive nature of resource management and query processing in the LINDO system justifies the use of a filtering-based retrieval mechanism. The objective is to find the results that strictly meet the expressed needs in the application and minimize the subset of metadata that the system has to scan in real-time while processing the request.

Query 1: Find all videos containing a *red bag*, forgotten in *Trocadéro, Paris* metro station, on *Thursday, 2 Febuary*, between *2:00pm and now (3:00pm)*.

Figure 8.5: Structure of an XML query launched to the LINDO IR system

After keyword extraction, the query processing proceeds by locating the servers responsible for managing the datastreams captured by the cameras located in the Trocadéro station waiting line. Next, a filtering step is performed to restrict the search within the segments captured between 14:00 and 15:00.

The system will then, determine a list of indexing algorithms that meet the needs, properties and context expressed within the query. This step will retrieve the subset of metadata describing the segments corresponding to the query.

In this scenario, the requested information are generic thus, the query processing will perform the search on the metadata generated by the implicit indexing algorithms and placed at the central server. The system will continue the search to find a red object in the retrieved list of metadata describing the chosen segments.

A filtering process is applied to take into account access control rules. Analyzing the access rights assigned to the security agent, we find that he is not authorized to access the videos containing passenger faces nor to use the personalized search options that employ the explicit indexing algorithms existing at remote servers.

Therefore, considering these access restrictions, the system will perform another filtering step to eliminate the segments that contain people faces and finally return to the user the list of segments that contain a red object (if available).

8.6.2 Adaptive situation-aware query processing with PSQRS

The search results returned to the security agent in this case might be insufficient especially that the red bag might be present within unauthorized segments containing passenger faces. Our proposal can take place at this level as a step towards ensuring a better quality of service by offering a wider subset of resources to the user while respecting the access rights defined on the consultation of the video surveillance data sources.

Through the usage of our proposed PS-RBAC model, the system would be able to offer more accessibility and adapt the permissions assigned to the security agent according to his contextual attributes and to the importance level of the situation of the consultation.

This adaptive solution can be employed when the system identifies access challenges related to the user's context or situation. In this scenario, the « lost object » situation identification can be obtained from the file number.

The implementation of the adaptive solutions is performed by PSQRS, which adapts decision-making by rewriting XACML queries. The solution proves its effectiveness due to its ability to achieve decision-making to access video surveillance sources that are distributed and administrated by different authorities.

Therefore, this simple request launched by the security agent (composed of keywords describing the content he needs, see Fig. 8.5) will be incorporated in an XACML access request that is more structured and rich of elements characterizing the context of an access request (describing the contextual constraints of the subject, his role, the importance level of the situation at which he's consulting the system, etc.), see Fig. 8.6, 8.7.

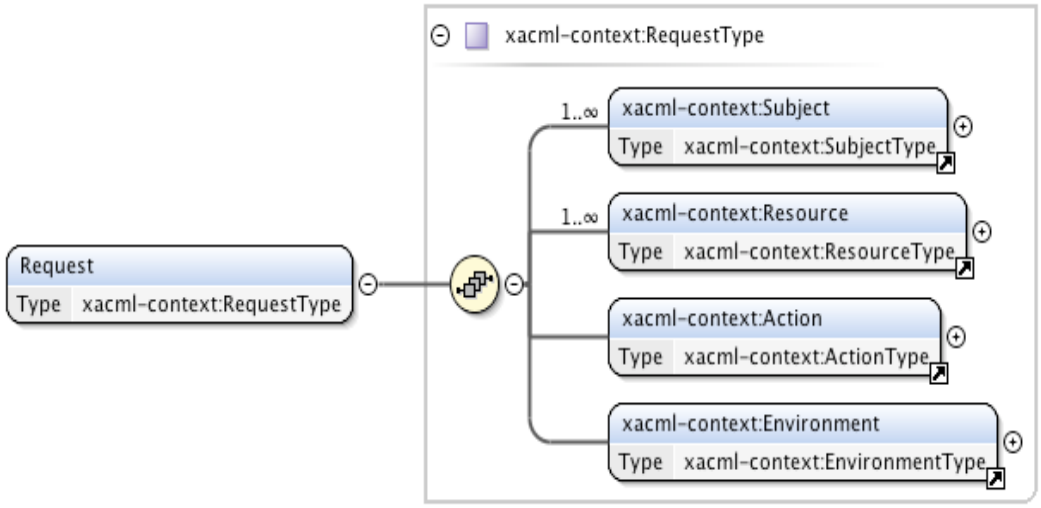


Figure 8.6: A generic schema of an XACML query

As shown in fig. 8.7, the richness of the elements embedded within an XACML query enables it to describe the contextual attributes characterizing:

(i) the requested source in the « **resource** » tag
(ii) the user launching the request in the « **subject** » tag
(iii) the situation at which the user has launched the access request in the « **environment** » tag

The contextual information of the query can then be extracted and interpreted by the system. The contextual attributes describing the requested video segments in our example are:

- ***The period*** estimated by the user, which can be also extracted from the information recorded within the RFID travel card. In the example, it's from 14:00 to 15:00.
- ***The GPS coordinates*** describing the location at which the video segments sought were acquired (extracted from the location indicated in the request).

GPS Coordinates of Trocadéro metro station: 2°17'59"E, 48°53'59"N (2.3, 48.9)

In the case of receiving negative or unsatisfactory results, the adaptive system will diagnose the consultation situation from the situation level attribute that is included in the **« environment »** tag:

Sit_Lvl = 0 → normal consultation
Sit_Lvl = 1 → search of a lost object
Sit_Lvl = 2 → search for a lost child

```
<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
  http://docs.oasisopen.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>John Smith</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
      DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>Security Agent</AttributeValue> </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:securityAgent-id"
      DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>sa2023</AttributeValue>  </Attribute>
  </Subject>
  <Resource>
    <ResourceContent>  <UserQuery>  <QueryInText> find all videos
          containing a red bag, forgotten in Trocadéro, Paris metro station,
          on Thursday, 2 Febuary, between 2:00pm and 3:00pm).</QueryInText>
        <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
        <MediaFormat>Video</MediaFormat>
          <TimeSpan>  <From>2012-02-02T14:00:00</From>
                      <To> 2012-02-02T15:00:00</To>  </TimeSpan>
      </UserQuery> </ResourceContent>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Read</AttributeValue> </Attribute>
  </Action>
  <Environment>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:environment-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
    <AttributeValue>Situation</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:situation-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Forgotten Object</AttributeValue> </Attribute>
     <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:sitLevel-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>1</AttributeValue> </Attribute>
  </Environment> </Request>
```

Figure 8.7: XACML request embedding the user's query

The importance level of the situation will determine the level of adaptation to be realized. The activation of the adaptive search mode will be communicated from the XACML response in the form of an « **obligation** » that accompanies the resulting access decision, see Fig. 8.8.

```
<Response>
  <Result>
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:2.0:status:ok"/>
    </Status>
    <Obligation FulfillOn="Deny" ObligationId="ApplyAdaptiveQueryingMode">
      <AttributeAssignment AttributeId="AQM"
            DataType="http://www.w3.org/2001/XMLSchema#string">
            On
      </AttributeAssignment>
    </Obligation>
  </Result>
</Response>
```

Figure 8.8: XACML response containing the obligation to be considered

As the adaptive querying mode is triggered, the query processing mechanism will change to ensure the success of the search by providing a variety of adaptive solutions in correspondence with the situation's sensitivity level.

This adaptive search solution is realized by the PSQRS that detects the situation sensitivity through the ***Situation Analyzer*** component and turns to the ***Similarity Provider*** component to find similar resources that will guide the query rewriting process (see Fig. 5.2).

In the case where the search didn't retrieve satisfactory results to the user and the consultation is taking place in a normal situation (Sit_Lvl = 0), the system will perform the adaptive query rewriting step through semantic similarity. The keywords of the user query will be reformulated using similar words or more generic concepts offered by the ***Similarity Provider***. Similar works have been introduced in [Al Kukhun and Sèdes, 2008], the objective is to maximize accessibility chances without crossing the security boundaries.

The semantic reformulation options can be achieved with the help of a standard lexical dictionary such as WordNet. For example, the word "bag" can be replaced by various synonyms {backpack, lugguage, purse, etc.}. The use of semantic reformulation was also suggested within the works of [Brut et al., 2011] but not in the context of finding alternative solutions when facing obstacles from access control.

At the other hand, the adaptation process in the mentioned scenario will follow another scheme since the lost object situation is judged to be of higher importance (Sit_Lvl = 1). Hence, the ***Similarity Provider*** component will be replaced by an ***Adaptive Solutions Provider***. This component will provide some predefined solutions that could bypass the access control challenge or would assist the user in adapting and reformulating his query by pointing out the access challenge and offering him adaptive solutions that would suit his context, the solutions are often saved in a predefined database. Table 8.1 shows examples of the solutions that the system can offer.

New solutions can also be inserted to the ***Adaptive Solutions*** database through a learning mechanism that detects the solutions that users employ when encountered with access challenges in real time.

The success of the adaptive solutions suggested by the users would eventually be more efficient if they knew the reason behind the access denial. The error messages that often accompany returned negative responses can serve as indicators to help the users in finding alternative solutions.

Problem	The adaptive Solution
The privacy law imposing the protection of anonymity of audiovisual content	
Passenger faces are not authorized	Display the content after the execution of an algorithm that applies a blur face function
Voice is not-authorized	Use an algorithm for speech-to-text transcription
Volume of the video	
Lack of storage capacity on the user's machine	Use a compression algorithm in order to obtain a smaller file
Format not supported by the user's machine	Use a conversion algorithm into a compatible format.
Download problems due to low bandwidth	Use a summarization algorithm in order to obtain a concise version of the content.

Table 8.1: Examples of adaptive solutions proposed by our system

Therefore, the adaptive solution for this example will modify the treatment process and will: (i) neglect the filtering step responsible for imposing the access control constraints and (ii) replace it with an adaptive step-related to the presentation of resources with unauthorized content.

By applying this process to the scenario described above, the system will return the video segments taken from the Trocadéro station between 14:00 and 15:00 and containing a red object.

These results will be filtered in order to detect the unauthorized segments (containing passenger faces). This is where the system will apply the adaptation process that would filter the display to conform with the access restrictions imposed by the system.

The adaptation will be performed through a face detection step and the use of an algorithm that applies a "blur function" to protect the privacy of passengers appearing in these segments in order to return to the user a list of pertinent results that respect the access rules.

8.7 Conclusion

The necessity of handling a huge quantity of high resolution multimedia content created by multiple sources in a distributed environment emerges and raises new challenges concerning the indexing and access to the multimedia content, such as: distributed storage with decentralized processing, choice of the indexing algorithms, real time information retrieval and location-aware retrieval within mobile or pervasive contexts.

The LINDO architecture has been proposed to tackle the above-mentioned challenges. The objective of the project was to build a distributed system for multimedia content management, and to ensure effective indexing and storage of data acquired in real time. The project didn't address the issues linked to data privacy and security.

Knowing that ensuring the protection of multimedia content is a key issue in certain application domains (e.g., video surveillance, medical domain, etc.), we have highlighted the importance of taking access control management into consideration at the different levels of data processing and should take into account the user's mobility. Meanwhile, these security constraints should not affect the user's accessibility needs especially in important situations.

The objective is to achieve a pervasive accessibility where the user can access data sources at anytime, from anywhere and anyhow. To achieve this goal, we have employed PSQRS - a Pervasive Situation-aware Query Rewriting System that offers adaptive context and situation - aware access solutions. The decision making within the system is based on the RBAC model and employs the XACML standard. These technologies are adapted to the distributed access management needs within the LINDO framework.

The proposed solution overcomes the access denials taking place in real time situations by modifying the query processing mechanism of the LINDO framework and by providing adaptive solutions that can bypass the access control constraints. A validation example was included in the context of video surveillance usecase.

The solution lies in an intermediate zone between respect for the rigidity of access decisions and the extreme flexibility of the option «Break-Glass» is commonly used in critical situations.

The adaptive and alternative based situation-aware solution could increase the complexity of request processing, but if we consider the usefulness of the results provided in real time and the fact they do not violate the access rights defined by the privacy law, this complexity seems quite acceptable.

Conclusion and Future Works

The evolution of Pervasive Information Systems PIS has introduced significant challenges related to security and access management. These systems should allow users to obtain transparent access to information sources at “anywhere, anytime and anyhow” while protecting integrity and confidentiality through the emposition of rigid security policies.

To meet this accessibility challenge, various research works were conducted directing towards enhancing access control modeling in three main directions: the first was interested in providing context-aware access control models that adapt decision-making according to the user’s context. The second went towards situation-awareness and meeting critical situations by integrating solutions for flexible decision-making such as the "Break-Glass" option. The third went towards enabling efficient service-oriented decision-making that meets the multi-distributive nature of pervasive sources, access policies and managing authorities.

In this context, the objective of our research works was to provide an efficient access control model that adapts decision-making according to the user’s context and situation. Our proposal presents a moderate flexibility between the rigid respect of the security policies gouverning information sources and other extreme access offered by the Break-Glass solution. The solution aims to ensure a better Quality of Service QoS and a transparent access without compromising the system’s security or integrity.

To attain this objective, we have introduced PS-RBAC – a Pervasive Situation-aware Role Based Access Control model that enables adaptive permission assignments to meet the volatility of the contextual attributes of pervasive users and the dynamic nature of the real-time consulting situations.

This adaptive decision-making model was translated within PSQRS – a Pervasive Situation-aware Query Rewriting System that employs a service-oriented architecture, which reacts in the case of an access denial, to provide alternative authorized sources through the use of an adaptive XACML query rewriting mechanism.

The presented model and system architecture have highlighted the importance of providing flexible decision-making and access alternatives especially in dynamic environments and real-time systems as in PIS. Our proposal attempts to meet this objective and has been validated in three application areas that are rich in real-time scenarios: (i) the Mobile Geriatric Teams, (ii) Avionics Systems, (iii) Video Surveillance Systems.

During our study, we were interested in confronting the access denial problems that encounter the users of pervasive systems while consulting information sources in real-time situations. Assuming that the reason behind an access denial was the security constraints imposed on the requested content, our proposed solution would interfere to search for alternative sources.

However, in a pervasive context, an access denial may not only be linked to the security constraints gouverning the content but might also be related to technical problems taking place (e.g. malfunctionality of the network/ server/ service providing

access to the needed content) or to difficulties in obtaining the right credentials or in aligning the access control policies managing data resources.

The variety of the reasons causing access denials has led us to identify many future works that could extend the works presented in this thesis.

In a short-term perspective, we aim to focus on improving the implemented system and the process of finding alternative solutions. An interesting method for providing similar sources would through the use of specialized domain ontologies that are related to the application areas that we have treated for the validation of our proposal. Considering the medical domain, many specialized domain ontologies exist such as: MESH[5], UMLS[6], GALEN[7], DICOM[8]. For the avionic systems case, an internal predefined database of anomalies can serve for finding alternative solutions and resolving situations. Finally, for the video surveillance application, ontologies related to audio-visual sources can be used.

Mid-term future works are also planned where we aim to upgrade the implemented prototype to enable it to perform adaptive decision-making in a multi distributive environment. Using a SOA, we can apply the adaptive process and search not only for alternative content but also for alternative services / servers that provide/ store the requested sources or the collection of access policies that manage them.

Finally, a long-term perspective is envisioned where we aspire to validate our proposal and perform a complete evaluation on a system developed on a large scale.

[5] MEdical Subject Headings http://www.nlm.nih.gov/mesh/
[6] Unified Medical Language System http://www.nlm.nih.gov/research/umls/
[7] http://www.opengalen.org/index.html
[8] Standard for Digital Imaging and Communications in Medecine http://medical.nema.org/dicom/2001.html

Plan de Résumé Français

Résumé Français

1. Introduction

L'évolution des Systèmes d'Information est liée au développement des télécommunications, de la connectivité, des matériels et des logiciels. Ces systèmes visent de plus en plus à proposer des environnements transparents et interopérables afin d'assurer un meilleur partage d'informations entre différents (sous-)systèmes d'information, pouvant présenter des niveaux de confidentialité hétérogènes.

Grâce au développement technologique et à l'intégration des nouvelles technologies dans toutes les applications de la vie quotidienne, la connectivité a réussi à renforcer l'accessibilité aux ressources. Cette évolution a donné une grande liberté d'interaction à l'utilisateur pour accéder à différentes ressources de n'importe où, n'importe comment et à n'importe quel moment : c'est ainsi que les systèmes sont devenus pervasifs ou ubiquitaires [Park et al., 2004]. La notion d'ubiquité a été initialement présentée par Weiser [Weiser, 1991] qui a prévu que dans les systèmes d'information du futur "au 21ème siècle", les éléments de calcul allaient "disparaître" en fonctionnant d'une manière homogène et en transparence totale.

La transparence devient une qualité de plus en plus fortement requise pour assurer une meilleure accessibilité aux ressources à tous les niveaux d'un système ou d'une entreprise. Toutefois, cette transparence risque de rendre les ressources vulnérables aux menaces et aux attaques de sécurité. C'est ici que se situe un des verrous scientifiques (et technologiques) constituant un frein à l'évolution de ces systèmes.

En effet, si l'on considère le problème de l'accès aux données dans les Systèmes d'Information Pervasifs (SIP), on se trouve face à un challenge qui est de trouver un équilibre entre la protection des données et la transparence d'accès aux ressources existant dans des environnements ouverts et gérées par différentes parties. Ainsi, l'évolution des SIP a introduit un nouveau défi lié à la gestion de l'accès aux données pour des utilisateurs mobiles. Ces systèmes doivent, à la fois, permettre aux utilisateurs d'obtenir une grande accessibilité mais aussi protéger le système en appliquant des politiques d'accès qui assurent sa sécurité pour le rendre invulnérable aux attaques d'intrus.

Les défis de gestion d'accès aux Systèmes d'Information classiques ont été résolus grâce à la proposition de plusieurs modèles de contrôle d'accès tels que DAC, MAC et RBAC. Mais l'évolution technologique a introduit de nouvelles contraintes liées au contexte et à la décentralisation des politiques de contrôle d'accès chargées de gérer l'accès aux ressources distribuées.

Avec le développement des architectures orientées-service et la répartition quasi-systématique des ressources, les efforts en matière de contrôle d'accès se sont donc plus orientés vers la prise d'une décision d'accès à partir de plusieurs politiques distribuées, gérées par différents services et générées parfois en temps réel. L'assurance de la qualité de service dans ce contexte a été réalisée par la garantie de l'efficacité de la prise d'une décision d'accès déterministe. C'est pour cette raison que différents standards ont été proposés tels que XML-Signature [XML Signature, 2002], XML Encryption [XML Encryption, 2002] et XACML [OASIS, 2003, 2005]. Ce dernier est

un standard qui vise à produire des décisions d'accès en se basant sur des politiques d'accès exprimées en XML [XML, 1998]. Le pouvoir de XACML réside dans sa capacité à prendre en compte les contraintes contextuelles issues d'un environnement pervasif.

Plusieurs travaux de recherche ont été menés pour étendre les modèles d'accès de base à des modèles sensibles au contexte mais ces travaux n'ont pas pris en compte l'importance de l'utilisabilité du système ni l'amélioration des possibilités d'accès : l'impossibilité de répondre à une demande utilisateur se solde toujours par un refus d'accès, souvent lié à la non prise en compte de la dynamique du contexte et de l'existence de politiques d'accès contradictoires dans une situation critique.

Le défi consiste donc à résoudre ce paradoxe (utilisabilité vs. sécurité) en mettant en œuvre des politiques d'accès conciliant ces deux objectifs parfois contradictoires.

C'est dans ce cadre que nous avons modélisé et mis en œuvre PS-RBAC (Pervasive Situation-aware RBAC Model), un modèle de contrôle d'accès qui étend le modèle RBAC en ajoutant la sensibilité au contexte et à la situation de l'utilisateur. En s'appuyant sur ce modèle, les mécanismes mis en œuvre ont pour objectif de réagir, dans les cas de refus d'accès, en réalisant une prise de décision adaptative qui permet de proposer une liste de ressources alternatives, "similaires" à celles demandées par l'utilisateur et qui sont, elles, autorisées.

Le modèle PS-RBAC vise à augmenter les possibilités d'accès, cet objectif a été atteint à travers PSQRS (Pervasive Situation-aware Query Rewriting System) : un système réécriture de requêtes XACML. L'objectif de PSRQS est de trouver, à partir des contraintes contextuelles du demandeur, les politiques qui s'appliquent et les ressources auxquelles il peut accéder puis, en cas de refus d'accès, de rechercher s'il existe des ressources qui contiennent des informations pertinentes ou similaires à celles demandées initialement et de les proposer comme solution adaptative ou alternative.

Ce modèle a été implanté et validé dans différentes applications et prototypes afin d'en évaluer les points forts et points faibles dans divers contextes (médical, aéronautique, vidéo surveillance).

Afin de présenter cette contribution, le mémoire est structuré en 3 parties présentant respectivement l'état de l'art, notre contribution et enfin l'implémentation et les résultats.

L'état de l'art traite trois points principaux : (i) la gestion classique du contrôle d'accès, (ii) l'évolution de la modélisation de la gestion d'accès pour répondre aux besoins des SIP et (iii) l'orientation des mécanismes d'accès vers les architectures orientées-service.

Dans la deuxième partie, nous détaillons notre contribution qui vise à proposer, d'une part, un modèle de contrôle d'accès sensible au contexte et à la situation des utilisateurs des SIP et d'autre part, une méthode de réécriture de requêtes XACML pour effectuer cette adaptation.

La troisième partie décrit des scénarii qui illustrent notre apport puis montre un exemple d'implémentation et les résultats qui valident notre proposition.

Enfin, nous concluons en précisant les contributions principales et en présentant les perspectives à court et long terme de nos travaux.

2. Etat de l'art : la gestion d'accès aux Systèmes d'Information Pervasifs

Un processus de gestion d'accès à l'ère des SIP est réalisé en trois étapes principales :

- La première comprend une modélisation des droits d'accès réalisée à travers l'adoption d'un modèle générique qui répond aux besoins des administrateurs, des utilisateurs et à la nature de la distribution des ressources ;
- La deuxième consiste à adapter le modèle d'accès choisi selon les contraintes d'utilisation d'un SIP (contexte, situation, etc.) ;
- La troisième est chargée de choisir un mécanisme orienté-service qui peut réaliser une gestion décentralisée pour assurer la prise de décision à partir des privilèges d'accès distribués.

En fait, l'évolution des SIP a conduit la plupart des contributions du domaine à étendre et personnaliser le modèle d'accès pour assurer la gestion des contraintes contextuelles de l'utilisateur lors de l'attribution d'une permission.

Dans cette section, nous citons une partie représentative des travaux de recherche qui ont contribué à faire évoluer la gestion d'accès dans ces trois étapes précédentes. Nous commençons par l'évolution des modèles d'accès classiques, puis nous citons des travaux qui traitent de l'adaptation de la prise de décision pour répondre à la dynamique du contexte pervasif et à l'importance de la situation dans laquelle l'utilisateur demande d'accéder au système (urgence, criticité, priorité, …) et finalement nous citons un standard de gestion d'accès dans les architectures orientées-service.

2.1 Les Modèles de base du contrôle d'accès

2.1.1 Le Modèle DAC

Le modèle de Contrôle d'Accès Discrétionnaire (DAC) a été défini par TCSEC (Trusted Computer System Evaluation Criteria) comme : "un moyen de restriction d'accès aux objets basé sur l'identité des sujets et/ou du groupe auquel ils appartiennent. Les contrôles sont discrétionnaires dans le sens où le sujet est capable de transférer les permissions d'accès à d'autres sujets" [NCSC, 1987].

Proposé par [Lampson, 1974] et formalisé par [Harrison et al., 1976], le modèle DAC a représenté les politiques de contrôle d'accès sous forme d'un triplet <utilisateur, objet, action> qui exprime que l'utilisateur peut effectuer une certaine opération identifiée par l'action (e.g. lire) sur l'objet spécifié. Le triplet est appelé une règle d'autorisation.

Dans certains systèmes, l'accès se fait uniquement en spécifiant, d'une façon explicite, un ensemble de règles d'autorisation. En d'autres termes, si aucune règle d'autorisation n'est définie pour un utilisateur, l'accès lui sera interdit. Ce type de politique est appelé une politique fermée [Samarati & al, 2000]. Inversement, dans les systèmes adoptant une politique ouverte, l'accès aux objets sera interdit, uniquement, en présence de règles d'autorisation (négatives), c'est-à-dire que l'utilisateur a le droit

d'accéder à tous les objets du système sauf si une règle d'autorisation a été définie explicitement lui interdisant l'accès aux objets.

Le modèle DAC limite l'accès aux objets uniquement en se basant sur l'identité de l'utilisateur et ne distingue pas les utilisateurs des sujets. Le fait que les permissions peuvent être transférées d'un utilisateur vers un autre, rend le processus de contrôle d'accès moins gérable et plus vulnérable aux risques de fuite d'information et d'attaques comme les "chevaux de Troie" [NCSC, 1987].

2.1.2 Le modèle MAC

Afin de remédier au problème de fuites d'information des modèles de contrôle d'accès discrétionnaires, les modèles obligatoires (Mandatory Access Control, MAC) fixent des règles incontournables destinées à forcer le respect des exigences de contrôle d'accès.

Le modèle de contrôle d'accès obligatoire distingue utilisateurs et sujets ; les ***utilisateurs*** sont des entités passives qui peuvent se connecter au système alors que les ***sujets*** sont des processus qui s'exécutent pour le compte des utilisateurs. Un utilisateur se connectant au système avec une classe d'accès donnée génère un sujet de cette classe d'accès. En fonction du niveau de sécurité (confiance) attribué à l'utilisateur, différents sujets peuvent être générés par cet utilisateur. Ainsi, le modèle MAC attribue aux sujets et aux objets des niveaux de sécurité non modifiables par les utilisateurs et, par conséquent, limite leurs pouvoirs dans la gestion des accès à leurs données. Par contre, la rigidité du modèle MAC ne lui permet pas de gérer le fait qu'il peut y avoir des exceptions entre les différents niveaux de sécurité.

Le modèle MAC s'inscrit dans les modèles de sécurité multi-niveaux, des exemples formalisés ont été proposés par [Bell et al., 1973]. On peut citer aussi le modèle d'intégrité de [Biba, 1977].

2.1.3 Le modèle RBAC

La motivation principale autour de la proposition d'un modèle de Contrôle d'Accès à Base de Rôles (RBAC) était de faciliter l'administration des privilèges d'accès pour un grand nombre d'utilisateurs accédant à des ressources distribuées. La solution présentée par [Ferraiolo et al., 1992] était de regrouper les utilisateurs dans des rôles reflétant la structure organisationnelle de l'entreprise puis de distribuer les permissions à ces rôles au lieu de les attribuer à chaque individu.

La notion de rôle forme le cœur du modèle RBAC où il intervient comme une entité intermédiaire entre les utilisateurs et les permissions. Un rôle regroupe un ensemble de privilèges et les attribue, ensuite, aux utilisateurs par rapport à leur fonction dans l'entreprise.

Comme le montre la fig. 1, l'attribution des rôles aux utilisateurs (AU) suit une relation mutuelle dans le modèle RBAC ; un utilisateur (personne, processus informatique, machine, etc.) peut jouer plusieurs rôles dans une seule session et un rôle peut être attribué à plusieurs utilisateurs.

AU Utilisateurs x Rôles

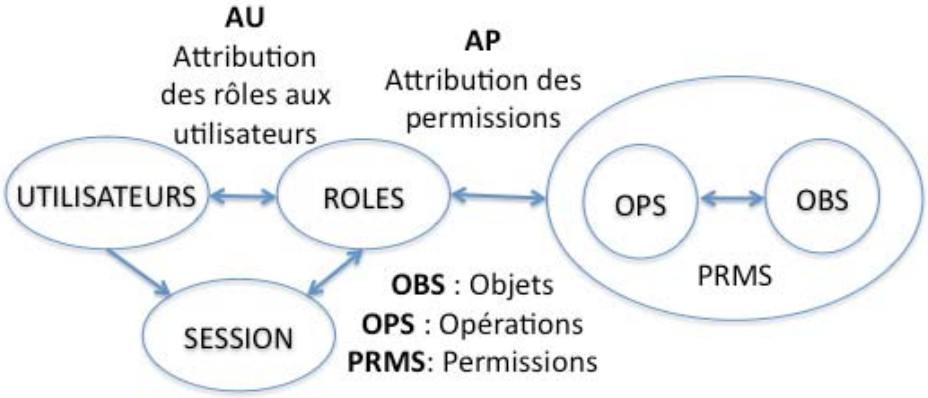


Figure 1 : Le Modèle RBAC

L'attribution d'un rôle à un utilisateur lui garantira une ou plusieurs permissions (PRMS). Dans le même temps, une permission peut être attribuée à un ou plusieurs rôles.

AP Rôles x PRMS

La nature d'une permission (PRM) décrit l'ensemble d'opérations (OPS) (e.g. lire, écrire, mettre à jour, etc.) autorisé sur les objets (OBS) qui représentent des ressources de données du système (documents, processus informatique, machines, etc.).

La relation entre ces objets et les opérations attribuées est aussi mutuelle ; une opération peut être autorisée sur plusieurs objets et à un objet peuvent être attribuées différentes permissions.

PRM OPS x OBS

Afin de répondre aux besoins évolutifs de la gestion d'accès au sein de l'entreprise, le modèle RBAC a été étendu à différents profils afin de combler les lacunes et atteindre une meilleure performance à travers différents principes tels que :

- La hiérarchie des rôles est présentée dans le profil RBAC-1 [Sandhu, 1996] où un utilisateur peut hériter les droits d'accès d'un autre utilisateur ;
- La possibilité de préciser des contraintes a priori lors de l'attribution d'un rôle est introduite dans le profil RBAC-2. Ce profil a également introduit la possibilité de sélectionner les permissions à hériter d'un rôle vers un autre, ce qui a aidé à établir des sous-hiérarchies personnalisées et privées pour limiter les risques de fuites d'information ;
- La combinaison des caractéristiques des deux profils précédents a formé le profil RBAC-3 [Kuhn et al., 1997]. Cette combinaison a permis de :
  - définir des contraintes pour établir des hiérarchies ;
  - caractériser l'interaction entre les différents rôles et appliquer la séparation des tâches entre les différents intervenants d'une mission. De telles séparations peuvent par exemple, prévenir un cas de fraude;
  - présenter le concept de rôles privés.

2.1.4 Synthèse

Dans cette section, nous avons suivi l'évolution des modèles de contrôle d'accès en présentant les trois modèles de base : DAC, MAC et RBAC. Après l'analyse de leurs caractéristiques, nous avons centré nos travaux de recherche autour du modèle RBAC qui est le modèle le plus adapté aux systèmes d'information de par :

(i) son efficacité d'administration et de distribution des permissions ;
(ii) le fait qu'il reflète la structure organisationnelle du système ;
(iii) sa capacité à englober les politiques DAC et MAC ;
(iv) son utilisation possible dans des environnements différents.

2.2 Le Contrôle d'Accès aux Systèmes d'Information Pervasifs

L'évolution des SIP a introduit des nouveaux défis à la gestion du contrôle d'accès. Un modèle du contrôle d'accès dédié à la gestion d'accès dans de tels systèmes doit :

- prendre en compte la dynamique des éléments décrivant le contexte de l'utilisateur qui est caractérisé par sa nature mobile (temps, localisation, caractéristiques du système, modalité de connexion réseaux, type de dispositif de l'utilisateur, etc.) et ;
- pouvoir réagir en temps réel si l'utilisateur rencontre un cas exceptionnel ou une situation critique où l'accès aux ressources est essentiel.

Nous exposons, dans la suite, les travaux sur l'adaptation des modèles d'accès pour atteindre ces deux éléments importants.

2.2.1 Le contrôle d'accès sensible au contexte

L'attribution d'une permission à un utilisateur, au sein des SIP, est devenue plus complexe à cause de sa relation et de sa dépendance au contexte. C'est pour cette raison que plusieurs travaux de recherche ont essayé de traiter ce problème par la proposition d'extensions des modèles d'accès existants prenant en compte l'évolution de la définition du contexte.

Dans la suite, nous introduisons l'ensemble des travaux qui se sont intéressés à l'adaptation du modèle RBAC.

2.2.1.1 Les modèles RBAC sensibles au contexte

L'évolution des modèles d'accès contextuels a commencé avec la prise en compte de l'axe temporel au sein du modèle RBAC. Les travaux de [Bertino et al., 2001] ont étendu le modèle RBAC vers un ***modèle T-RBAC*** (Temporal RBAC) qui considère le temps comme une contrainte qui peut déterminer l'activation ou la désactivation d'un rôle. L'intégration de l'aspect temporel a donné plus de flexibilité pour créer des exceptions pour les individus et pour spécifier des dépendances temporelles entre les actions réalisées par un utilisateur.

Ensuite, avec l'évolution des dispositifs mobiles, l'intégration de la localisation de l'utilisateur a fait l'objet de plusieurs travaux de recherche comme ceux de [Hansen

et al., 2003] qui ont proposé un ***modèle RBAC Spatial*** où l'attribution d'un rôle dépend de la position de l'utilisateur.

Le caractère dynamique du contexte a été aussi traité par [Zhang et Parshar, 2003] en proposant une extension du modèle RBAC vers un ***modèle RBAC dynamique*** (Dynamic-RBAC). Ce dernier adapte l'attribution des rôles et des permissions en prenant en compte le changement des informations contextuelles. La dynamique du modèle a été réalisée à travers l'utilisation d'automates.

Les travaux de [Bertino et al., 2005] proposent un ***modèle Géo-RBAC*** (Geographical RBAC) qui déterminent la localisation d'un utilisateur, soit par son positionnement physique exact (à l'aide d'un GPS), soit à travers son positionnement logique calculé implicitement (dans la région dans laquelle il se déplace, cette région pouvant être définie à différents niveaux de granularité).

Avec l'évolution des systèmes ubiquitaires, différents modèles sensibles au contexte ont été proposés tels que le ***modèle uT-RBAC*** [Chae et al., 2006] qui emploie un automate pour décrire et prendre en compte les changements du contexte représentés par le temps et la localisation de l'utilisateur. Ces éléments sont considérés comme des éléments importants pour l'activation et la désactivation d'un rôle.

L'exploitation des mécanismes d'Intelligence Artificielle a été aussi introduite par [Lim et Shin, 2007] qui ont proposé un ***modèle RBAC basé sur les réseaux neuronaux*** pour fournir plus de flexibilité lors d'une prise de décision d'accès dans un contexte ubiquitaire.

D'autres travaux ont présenté un ***modèle RBAC sensible au contexte pour les systèmes pervasifs*** [Emami et al., 2007] et ont souligné le fait que les attributs contextuels sont très dynamiques ce qui peut risquer de déstabiliser les autorisations. En conséquence, les auteurs ont distingué deux types d'éléments contextuels : (i) des éléments de durée longue – chargés de l'attribution des rôles, et (ii) des éléments de durée courte – chargés de l'attribution des permissions. Ces éléments peuvent être reliés soit à l'utilisateur, soit à l'environnement.

Une autre extension a été proposée pour fournir un ***modèle RBAC adapté aux besoins des systèmes pervasifs*** par [Kulkarni et al., 2008]. Ce modèle sépare la gestion du contexte du contrôle d'accès pour faciliter la prise de décision dans le cas où une autorisation est liée à plusieurs contraintes contextuelles. Les auteurs fournissent un service dédié à la gestion des ressources pervasives. A partir de l'ensemble des contraintes contextuelles, une décision d'accès à une ressource est prise.

2.2.1.2 Synthèse

Après avoir réalisé cette étude analytique sur les différentes extensions du modèle RBAC (voir table 1), nous soulignons l'importance de la prise en compte du contexte lors de la prise d'une décision d'accès. L'objectif est de rendre l'application plus adaptée aux contraintes du temps réel et au caractère dynamique des systèmes pervasifs.

Contraintes contextuelles Modèle	Référence	Temps	Localisation	Autres	Technologie
T-RBAC	Bertino *et al.* 2001	x			
Spatial RBAC	Hansen *et al.* 2003	x	x		
Dynamic RBAC	Zhang *et al.* 2003	x	x	x	
Geo-RBAC	Bertino *et al.* 2005	x	x		
uT-RBAC	Chae *et al.* 2006	x	x	x	logique
Context-Aware Access Control Model for Pervasive Computing Environments	Emami *et al.* 2007	x	x	x	
Intelligent Access Control Mechanism for Ubiquitous Applications	Lim *et al.* 2007	x	x	x	réseaux neuronaux
Context-aware RBAC in pervasive computing systems	Kulkarni *et al.* 2008	x	x	x	

Tableau 1 : L'évolution des modèles d'accès sensibles au contexte

2.2.2 Le contrôle d'accès sensible à la situation

Comme nous l'avons montré dans la section précédente, l'attribution d'une permission à un utilisateur au sein des SIP est très liée au contexte. Le fait qu'un attribut contextuel dynamique soit devenu un élément décisif dans le processus de prise de décision d'accès, augmente la complexité de la prise de décision ayant pour conséquence de générer des refus d'accès plus nombreux. De telles réponses négatives peuvent influencer la qualité de service des SIP qui réclament plus de transparence et plus d'accessibilité aux ressources de données pour les utilisateurs depuis n'importe où, n'importe quand et n'importe comment.

Le respect de la rigidité des décisions d'accès est fortement exigé pour assurer la sécurité et l'intégrité des systèmes d'information. Cependant, il est quasiment impossible de prédire et spécifier toutes les situations et scénarii d'accès requis. Or, une flexibilité des accès s'avère indispensable spécialement dans des situations critiques (urgence médicale, incendie, problème technique dans un avion, etc.) où un refus d'accès peut non seulement risquer la qualité de service du système mais aussi la vie des bénéficiaires du système.

Afin de résoudre ce problème de refus d'accès se produisant dans des situations critiques, une vision optimiste de la sécurité a été présentée par [Povey, 1999], [Rissanen et al., 2004] où une solution de politiques relaxées a été proposée. Dans cette solution, appelée ensuite « bris-de-glace » [Break-Glass, 2004], le système aide les utilisateurs à se confronter aux situations d'urgence en appliquant une relaxation temporelle aux restrictions d'accès afin de débloquer la situation.

Malgré son importance, la solution « bris-de-glace » est considérée comme une solution extrême présentant des risques pour la confidentialité des données. Pour cette raison, une étape d'audit et de vérification administrative doit être ensuite mise en

œuvre afin d'assurer la traçabilité et maintenir l'intégrité du système [Povey, 1999]. Cette étape a prouvé l'utilité du « bris-de-glace » dans les cas d'urgence mais elle a aussi révélé son danger lorsque les utilisateurs l'exploitaient pour effectuer des intrusions illégitimes et non justifiables.

Par conséquent, et comme nous allons le montrer, différents travaux de recherche se sont intéressés à contrôler le processus du « bris-de-glace », à améliorer sa modélisation et à faciliter l'intégration des droits d'accès flexibles au sein des modèles d'accès classiques.

2.2.2.1 Propositions pour contrôler les risques du « bris-de-glace »

Une étude analytique des travaux traitant la modélisation du contrôle d'accès sensible à la situation nous permet de dégager trois directions principales.

La première direction a étudié ***l'adaptation à la situation avec des règles ou solutions prédéfinies intégrées dans le modèle d'accès.*** La réalisation se base sur une planification et une modélisation a priori qui prédit les différentes possibilités qui peuvent générer des cas urgents et les considère comme des conditions pour autoriser le déclenchement d'une situation d'urgence dans laquelle l'utilisateur bénéficie de plus d'accessibilité aux ressources du système grâce à l'option du « bris-de-glace ».

Les travaux de [Ferreira et al., 2006] ont proposé une implémentation qui vise à contrôler les risques des intrusions illégitimes lors de l'utilisation du « bris-de-glace » au sein des systèmes de santé. L'objectif de la solution proposée est de s'assurer que les utilisateurs, bénéficiant de l'option "bris-de-glace", prennent la responsabilité de leurs actions. Le système effectue la *non-répudiation* en utilisant une procédure de "bris-de-glace" bien surveillée.

Un travail de modélisation plus approfondi a suivi dans [Ferreira et al., 2009] où l'option "bris-de-glace" a été intégrée dans le modèle RBAC d'une manière sécurisée et transparente. Le modèle introduit, appelé BTG-RBAC, vise à (i) dépasser la rigidité des décisions d'accès fournies par le modèle RBAC, (ii) offrir un accès plus flexible aux utilisateurs, (iii) assurer la *non-répudiation* par la sensibilisation des utilisateurs bénéficiant de l'option "bris-de-glace".

Les travaux de [Brucker et al., 2009] se sont dirigés vers l'amélioration de l'intégration des solutions "bris-de-glace" par la proposition d'un modèle générique de "bris-de-glace". La solution peut intégrer différents modèles de contrôle d'accès ; elle fournit des outils qui assurent une spécification avec une granularité très fine où la flexibilité d'accès est réalisée au niveau de la permission d'accès au lieu de celui du rôle. Afin de minimiser les risques de violation de sécurité du système, le modèle fournit la possibilité d'adapter le "bris-de-glace" selon le niveau d'urgence.

La deuxième direction a proposé de réaliser une ***flexibilité d'accès surveillée ou assistée*** où l'utilisateur confronté à un cas d'urgence obtiendra plus d'accessibilité aux ressources du système à partir d'une autorité supérieure. Cette dernière pourra ensuite lui fournir les services/données dont il a besoin ou trouver un sujet qui pourra lui déléguer des autorisations afin de pouvoir accéder aux ressources demandées.

Les travaux de [Keppler et al., 2006] ont présenté une plateforme générique de gestion d'accès qui gère les refus d'accès aux ressources de données en utilisant une fonction de partage basée sur la logique. Le système vérifie si les ressources

demandées sont critiques pour la réalisation de la tâche courante et en conséquence réagit de 2 façons :

(i) Soit il cherche un autre sujet qui peut être lié à la tâche et autorisé à accéder ces ressources afin de fournir les informations demandées.

(ii) Soit il essaie de trouver la raison du refus d'accès (manque d'information ou de privilèges d'accès) et s'il y a une possibilité de négocier avec des dispositifs (machine, utilisateur) ayant une autorité supérieure pour lui autoriser l'accès.

Les travaux de [Catarci et al., 2008] ont proposé une plateforme pervasive qui répond et gère la complexité d'accès lors des crises et situations urgentes. L'accessibilité est assurée à travers un système collaboratif qui assure la liaison entre les différents sujets traitant la crise ou la situation urgente sur place et des autorités externes qui peuvent intervenir pour fournir un accès flexible à des ressources externes qui ne sont pas autorisées a priori.

La troisième direction s'est orientée vers plus ***d'autonomie de prise de décision accompagnée d'une gestion des risques***. L'autonomie a permis de produire des permissions flexibles de manière ad hoc.

Les travaux de [Cheng et al., 2007] ont proposé un modèle de sécurité multi-niveaux basé sur la logique floue. Le système fournit une gestion autonome aux ressources des données rassemblées à partir des ressources distribuées. Le domaine d'application est lié à la sécurité nationale[9] où la rigidité des décisions d'accès n'est pas acceptable et où le fait de dévoiler les ressources de données peut être moins risqué qu'un refus d'accès. Par conséquent, le modèle assure la flexibilité de prise de décision en utilisant une fonction de logique floue qui offre une zone intermédiaire (entre les permissions et les refus d'accès) où des autorisations d'accès sont réalisées et accompagnées d'un calcul de risque.

2.2.2.2 Synthèse

Dans cette section, nous avons réalisé une étude analytique qui montre les différents travaux traitant de la modélisation du contrôle d'accès sensible à la situation et de la recherche d'une intégration standardisée de l'option de "bris-de-glace" avec le moindre coût de risque.

Malgré le fait que l'utilisation du "bris-de-glace" a souvent révélé des violations d'accès, l'assurance d'un accès flexible aux ressources de données pour les utilisateurs des SIP est primordiale pour garantir la qualité de service lors d'une situation d'urgence.

Comme nous le résumons en fig. 2, l'adoption de la flexibilité d'accès au sein du modèle de contrôle d'accès suit une relation linéaire avec le risque :

- ❖ plus l'adaptation est définie de manière rigide en se basant sur des conditions et des scénarios prédéfinis, moins le risque des violations d'accès est élevé ;

[9] Un projet en collaboration entre le laboratoire de recherche de l'armée américaine et la Ministère de la défense de l'Angleterre.

- plus le processus d'adaptation s'oriente vers une prise de décision flexible en se basant sur des fonctions ad hoc, plus le risque des violations d'accès et des intrusions illégales peut se poser.

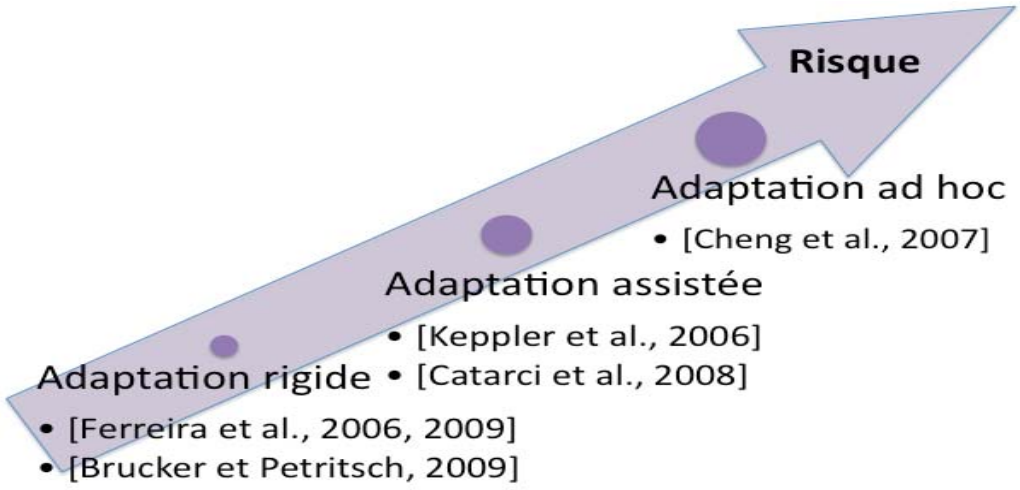


Figure 2 : La relation entre l'adaptation à la situation et les risques des violations d'accès

2.3 Le Contrôle d'Accès Orienté Service - Le standard XACML

L'évolution des SIP s'est donnée pour objectif, un accès transparent aux ressources de données, à n'importe quel moment, depuis n'importe où et n'importe comment. Afin de réaliser cet objectif, une distribution des systèmes d'information a été nécessaire avec la notion de l'exploitation et du partage des ressources ou services par différents sujets de différents domaines. De tels partages doivent être gouvernés par plusieurs parties sous forme de politiques d'accès distribuées, dynamiques et évolutives en temps réel.

Cette nouvelle conception des architectures orientées-service a facilité l'exploitation des services dans différentes applications et systèmes d'information. Elle a aussi introduit un nouveau challenge pour la gestion d'accès où un mécanisme de contrôle d'accès doit non-seulement pouvoir gérer la distribution d'un grand nombre de privilèges destinés à plusieurs sujets demandant l'accès aux ressources distribuées mais, doit aussi pouvoir prendre des décisions d'accès à partir de politiques d'accès distribuées, évolutives et parfois contradictoires. Pour réaliser de telles prise de décision, le standard XACML a été développé.

XACML (***eXtensible Access Control Markup Language***) est un langage destiné au contrôle d'accès, basé sur XML et proposé par [OASIS, 2003]. Dédié à l'administration des politiques d'accès dans les domaines orientés services, XACML décrit des politiques de contrôle d'accès permettant de définir les privilèges des utilisateurs sur un service. Ce standard permet à la fois, d'authentifier et de sécuriser l'accès en prenant en compte différents éléments reliés au contexte de l'utilisateur.

La spécification XACML fournit une architecture qui décrit le processus et les étapes permettant de fournir une décision d'accès lors d'une demande d'autorisation (cf. fig. 3).

(1) La gestion d'accès au niveau du système commence par la distribution et la publication des politiques d'accès dans les points d'administration des privilèges ***PAPs***

(***Policy Administration Points***) qui les mettent à la disposition des points de prise de decision ***PDP*** (***Policy Decision Point***).

(2) Quand une demande d'accès est effectuée, une requête est formulée et envoyée au point d'imposition des droits d'accès ***PEP*** (***Policy Enforcement Point***).

(3) Le PEP récupère cette requête et sera chargé de vérifier sa validité à l'aide du Gestionnaire du Contexte (***Context Handler***).

(4,5) Le ***Context Handler*** identifie la ressource demandée et récupère - du ***PDP*** - les attributs contextuels exigés pour réaliser une décision d'accès.

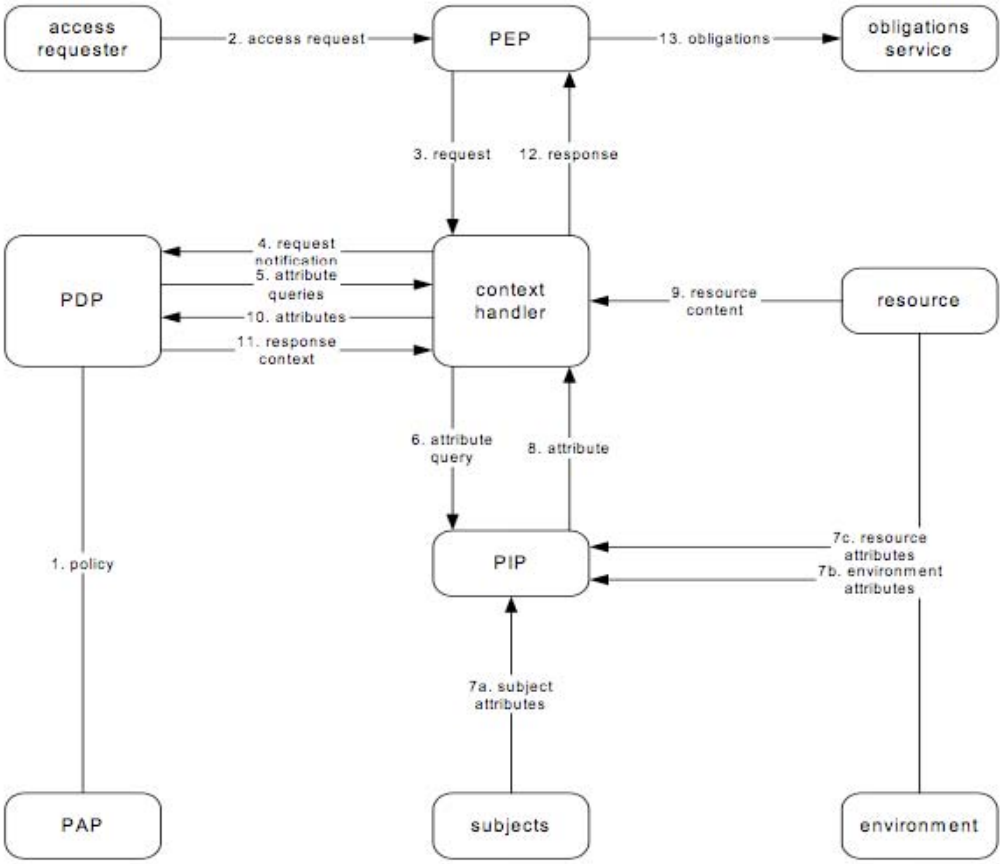


Figure 3 : Modèle simplifié de la gestion d'accès à un flot de données en XACML

(6-10) A partir des attributs retenus, le ***Context Handler*** récupère les valeurs contextuelles liées à la demande d'accès et formule une requête XACML qui sera, ensuite, envoyée au ***PDP*** pour qu'elle soit analysée.

(11-12) Le ***PDP*** analyse la requête XACML en comparant les valeurs actuelles avec les valeurs définies par les politiques d'accès puis, il envoie la réponse (***Permit / Deny***) au ***Context Handler*** qui la transmet au ***PEP*** afin de l'imposer et assurer qu'elle sera appliquée.

(13) Un ***Service d'Obligation*** est finalement responsable de l'application des actions que les politiques d'accès ont précisées. De telles obligations peuvent être réalisées (avant/pendant/après) l'imposition d'une réponse d'accès.

L'architecture de XACML permet de représenter les stratégies de contrôle d'accès sous forme de règles et repose sur les notions de ***Rule***, ***Policy*** et ***Policy Set***. Ces

notions permettent de préciser les conditions d'application des éléments ***Subject***, ***Resource***, ***Action***, ***Environment*** qui sont prisent en charge par ***le Context Handler***. Une ***Rule*** (loi) comprend des ***Conditions*** et des ***Effects*** (effets) et une ***Policy*** (Politique) comprend des ***Rules*** (lois) et ***Obligations***.

XACML est considéré comme un standard efficace pour sa capacité à gérer les droits d'accès d'une manière distribuée en prenant en compte les différents attributs contextuels de l'utilisateur ou du service.

Un profil XACML RBAC a été introduit par [OASIS, 2005] pour favoriser la portabilité du standard vers des services à grande échelle.

2.4. Discussion

Dans cet état de l'art, nous nous sommes intéressés à l'évolution de la modélisation du contrôle d'accès dans l'objectif de mieux répondre aux besoins des SIP.

Parmi les modèles classiques du contrôle d'accès, le modèle RBAC a souvent été choisi pour sa capacité à gérer la distribution des ressources de données et à prendre en compte la relation entre les utilisateurs selon la hiérarchie organisationnelle du système.

Afin d'intégrer la caractéristique dynamique du contexte des SIP, différents travaux se sont dirigés vers l'extension du modèle RBAC afin d'assurer la prise en compte du contexte de l'utilisateur lors d'une demande d'accès.

Etant donnés que ces systèmes doivent permettre l'accès aux ressources de données en temps réel, le modèle d'accès doit prendre en compte la situation dans laquelle l'utilisateur consulte le système. Différents travaux ont proposé d'étendre le modèle du contrôle d'accès afin de réaliser une prise de décision sensible à la situation. En conséquence, un utilisateur doit pouvoir réagir et parfois dépasser les droits d'accès en utilisant la fonction "bris-de-glace" afin d'affronter une situation critique.

L'introduction du standard XACML a assuré l'interopérabilité de la prise de décision d'accès où les ressources de données sont distribuées et gérées par différents services. La complexité de la prise de décision dans les SIP est due à la distribution qui touche non seulement les ressources de données mais aussi les politiques gérant ces ressources.

Le fonctionnement de la prise de décision du standard XACML offre des solutions plutôt rigides. Par conséquent, nous soulignons l'importance d'offrir des réponses assez flexibles qui s'adaptent pour répondre aux besoins évolutifs des SIP où les utilisateurs se trouvent souvent dans des contextes dynamiques et confrontés parfois à des situations critiques [Al Kukhun et Sèdes, 2007].

En réponse à ces challenges d'accès, nous proposons dans la suite un modèle de contrôle d'accès adaptatif, sensible au contexte et à la situation qui offre des solutions alternatives permettant de suppléer aux demandes d'accès non-autorisées.

3. Contribution

Dans ce mémoire, nous présentons donc, une solution adaptative en montrant qu'une consultation d'un système exige parfois un accès «intelligent et proactif» où le système cherchera, dans le cas d'une demande rejetée, s'il existe des ressources autorisées qui sont pertinentes (alternatives) dans ce contexte. Pour réaliser une telle recherche, nous présentons une extension du modèle RBAC qui s'adapte aux besoins des utilisateurs et/ ou à la nature du service attendu par le système.

3.1 PS-RBAC : un modèle RBAC pervasif et sensible à la situation

Le modèle PS-RBAC (Pervasive Situation-aware RBAC) [Al Kukhun et Sèdes, 2009a], [Al Kukhun et al., 2012a] que nous proposons est une extension du modèle RBAC. L'objectif est de permettre la construction d'autorisations flexibles qui s'adaptent au changement de droits d'accès causé par la mobilité de l'utilisateur. Notre modèle prend en compte les attributs contextuels de l'utilisateur et la situation dans laquelle il consulte le système afin de lui fournir des propositions d'accès à des ressources alternatives.

Nous avons choisi d'utiliser le modèle RBAC pour ses nombreuses caractéristiques telles que sa capacité à faciliter l'administration des ressources décentralisées, à prendre en compte la relation entre les utilisateurs selon la hiérarchie organisationnelle du système, à passer à l'échelle mais aussi pour sa popularité dans les différents systèmes de gestion d'accès.

Comme nous le montrons en fig. 4, nous utilisons le principe d'attribution des rôles du modèle RBAC. Puis, nous étendons les permissions attribuées à ces rôles pour acquérir deux types de permissions : (i) des permissions prédéfinies par les politiques existantes dans le système, (ii) des permissions adaptatives définies en temps réel par le processus adaptatif que nous avons défini.

Afin de réaliser cette adaptation et de construire ces nouvelles permissions, le mécanisme d'attribution des permissions dans le modèle PS-RBAC s'appuie sur un composant qui étudie le contexte de l'utilisateur puis la sensibilité de sa situation qui, en fonction du résultat, réalisera un processus de recherche vers des ressources similaires autorisées et les proposera comme solutions alternatives.

Dans ce qui suit, nous présentons la modélisation du processus adaptatif que nous proposons en intégrant des composants supplémentaires dans le modèle RBAC. Nous allons définir formellement ce nouveau modèle afin d'expliquer les relations entre les composants de ce modèle.

En principe, au moment de l'accès au système, l'utilisateur s'identifie pour ouvrir une session dans laquelle il lui sera attribué un ou plusieurs rôles. Cette attribution lui permet d'accéder à une partie des ressources du système afin de réaliser différentes tâches. Cette attribution prend en compte les attributs de l'utilisateur tels que son identité, son profil, le contexte, etc.

Le modèle reflète la mobilité et la variété du contexte : l'utilisateur sera aussi caractérisé par des attributs contextuels dynamiques (localisation, type de connexion, heure, etc.). Ces caractéristiques dynamiques influencent les permissions qui seront données.

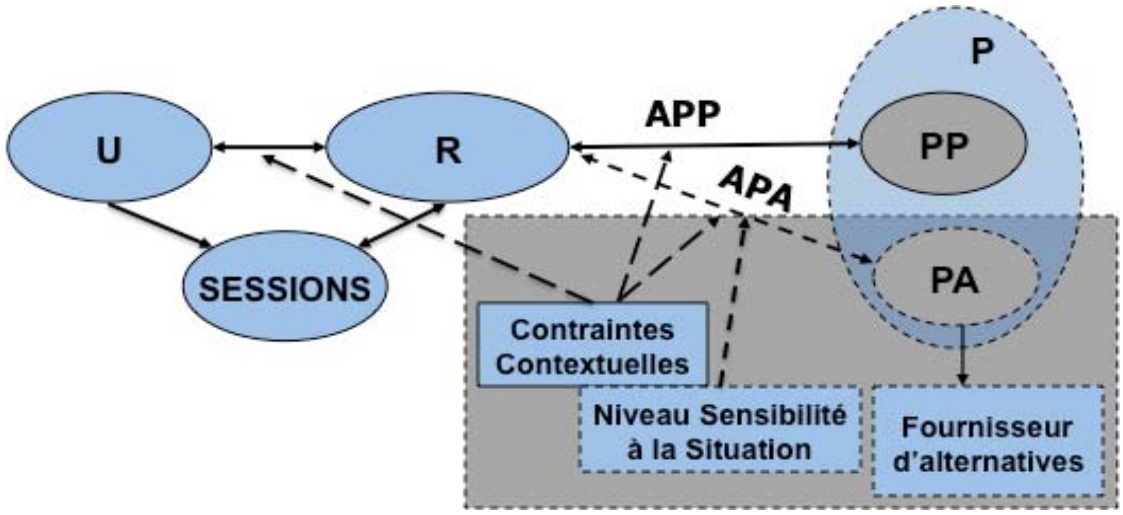


Figure 4 : Le Modèle PS-RBAC « Pervasive Situation-aware RBAC »

Un Rôle R reflète le positionnement d'un utilisateur dans la hiérarchie organisationnelle de l'entreprise. Le regroupement de plusieurs personnes (effectuant une tâche similaire) dans un rôle a pour objectif de faciliter la distribution et la mise-à-jour des droits d'accès aux ressources de données.

Le rôle du modèle proposé est placé au cœur du système : il fait la liaison entre les utilisateurs et les permissions et assure une gestion décentralisée des droits d'accès (qui est souvent réalisée par une troisième partie (e.g. un administrateur)).

L'attribution des rôles aux utilisateurs UA est représentée par une relation «many-to-many» où un utilisateur peut se voir attribuer plusieurs rôles et un rôle peut être relié à plusieurs utilisateurs en même temps. Dans un contexte pervasif, cette attribution est gérée par un certain nombre des ***Contraintes Contextuelles CC***.

AU Utilisateurs x Rôles x CC

Par exemple, dans le cas d'un système de santé, un utilisateur peut occuper plusieurs rôles en même temps : il peut être patient (sous traitement) et médecin traitant en même temps ou médecin et chef de service, etc. Les contraintes contextuelles gérant de telles attributions peuvent être l'heure et la localisation.

Une Permission P est une autorisation qui donne à l'utilisateur le droit d'accéder aux ressources du système, cette autorisation passe à travers le rôle. Notre modèle génère deux sortes de permissions selon la situation et le contexte de consultation : des permissions prédéfinies et des permissions adaptatives.

P = {PP U PA}

Les Permissions Prédéfinies PP sont des autorisations définies explicitement a priori par les gestionnaires du système. Dans le cas d'un système distribué qui utilise XACML pour la gestion des droits d'accès, ces permissions sont sauvegardées dans des politiques d'accès puis distribuées dans plusieurs PAPs (Policy Administration Points).

L'attribution des Permissions Prédéfinies APP est représentée par une relation «many-to-many» où un rôle peut être attribué à plusieurs permissions et une permission peut être associée à plusieurs rôles. Dans un contexte pervasif, cette attribution est aussi gouvernée par les Contraintes Contextuelles CC de l'utilisateur.

APP Roles x PP x CC

Les Permissions Adaptatives PA sont des autorisations alternatives proposées d'une manière ad hoc par notre modèle. La génération de telles permissions aura lieu dans le cas d'un rejet d'une demande d'accès établie dans une situation importante (consultation extra hospitalière, urgence, etc.).

Ces autorisations sont liées au contexte courant de l'utilisateur. Le contexte, dans le modèle proposé, est représenté par un ensemble de Contraintes Contextuelles CC qui sont caractérisées par leur nature dynamique due à la volatilité des environnements pervasifs. Le modèle proposé prend en compte la détection des contraintes, l'identification de l'ensemble des ressources accessibles dans ce contexte et la recherche de ressources alternatives qui peuvent répondre à la demande de l'utilisateur.

Les ressources alternatives sont obtenues à l'aid d'un fournisseur des ressources alternatives qui est personnalisé selon le besoin du domaine d'application. La similarité peut être utilisée afin de proposer une ressource pertinente. Cette similarité (calculée entre les ressources non autorisées demandées et l'ensemble des ressources authorisées) peut être liée au contenu d'un document (contenu textuel), à sa structure (document XML) ou à différentes relations spatio-temporelles (dans le cas d'une recherche vers une ressource localisée à proximité).

L'attribution des Permissions Adaptatives APA forme une relation « many-to-many » entre les rôles et les permissions Adaptatives qui sont fortement influencés par le contexte courant présenté par les contraintes contextuelles, le niveau de sensibilité de la situation et l'existance des ressources alternatives.

APA Rôles x PA x CC x SS U Alt

Dans cette section nous avons présenté comment nous avons modélisé la prise de décision et dans ce qui suit, nous allons montrer l'architecture associée du système proposé qui applique la prise de décision adaptative et sensible à la situation. Cette architecture met en œuvre la prise de décision du standard XACML et l'adapte selon la situation en utilisant des notions de similarité. Dans la suite, nous allons présenter le système qui applique cette procédure adaptative en utilisant la réécriture des requêtes.

3.2 PSQRS : Un système adaptatif sensible au contexte et à la situation basé à la réécriture des requêtes XACML

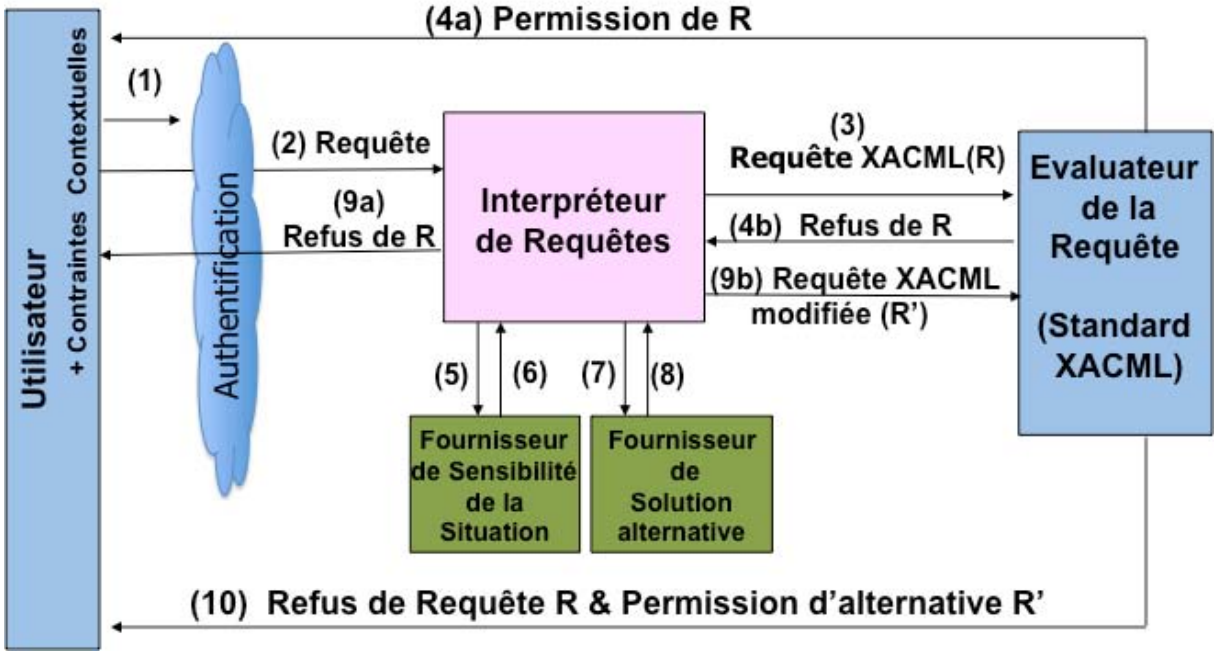
Nos travaux appliquent une procédure adaptative qui prend en compte le rôle de l'utilisateur, son contexte et sa situation afin de lui fournir des moyens pour accéder à des ressources alternatives dans le cas où il effectue une demande non autorisée.

Un refus d'accès peut être causé par un changement du contexte de l'utilisateur ou par la variété de contenu des systèmes interrogés. Pour résoudre ce challenge, nous proposons un mécanisme de réécriture des requêtes de l'utilisateur [Al Kukhun et Sèdes, 2007] en utilisant la recherche de degrés de similarité entre un élément non autorisé (demandé par l'utilisateur) et des documents ou des services existants dans le système et autorisés à l'utilisateur.

L'architecture PSQRS ***Pervasive Situation-aware Query Rewriting System*** que nous proposons vise à étendre le modèle de prise de décision dans XACML pour le rendre plus adaptable, voir fig. 5. La flexibilité de prise de décision est réalisée à partir d'une couche adaptative mise en oeuvre par un mécanisme de réécriture de la requête

de l'utilisateur dans le cas où elle est refusée par le PDP [Al Kukhun et al., 2008a, 2008b].

La réécriture de requêtes XACML est aussi considérée comme un élément clé pour assurer un accès sécurisé aux ressources, en permettant de modifier la visualisation de la structure arborescente d'un document selon les privilèges d'accès



accordés à un utilisateur.

Figure 5 : PSQRS – Un système de réécriture des requêtes XACML

Le système PSQRS récupère les contraintes contextuelles de l'utilisateur et les reçoit avec une étape d'authentification (1). Puis, quand l'utilisateur lance sa requête, le système la communique au générateur de requête (2) qui la traduit vers une requête R de format XACML. Celui-ci prend en compte cette demande et la combine avec les contraintes contextuelles puis l'envoie vers l'Evaluateur de requêtes (3) qui joue le rôle d'un PDP et suit le processus normal de XACML.

Selon les droits d'accès de l'utilisateur (précisés par les politiques d'accès sauvegardées dans les PAPs (voir le schéma XACML en section 2.2.1), le système répond à cette demande soit en permettant à l'utilisateur d'accéder à la ressource demandée (4a), soit en lui répondant avec un refus d'accès (4b). C'est dans ce dernier cas que le mécanisme adaptatif proposé intervient pour étudier la situation dans laquelle l'utilisateur a consulté le système. Cette situation est définie par le Fournisseur de Sensibilité de la Situation (5 et 6) qui autorise la régénération de la requête R' dans le cas d'une situation d'urgence par exemple.

Cette régénération ou réécriture de requête est réalisée grâce au Fournisseur de des solutions alternatives (7 et 8) qui prend les contraintes contextuelles de l'utilisateur comme un point de départ pour la recherche des documents ou des services alternatifs autorisés ayant des similarités de contenu ou de fonctionnement avec la ressource jugée comme non autorisée initialement demandée par l'utilisateur.

Cette étape est destinée à restituer à l'interpréteur de requêtes des ressources alternatives similaires. Dans le cas où le système n'a pas de propositions, l'interpréteur des requêtes va envoyer à l'utilisateur un refus d'accès (9a) (cas classique). En revanche, dans le cas où le système trouve des ressources similaires alternatives, le Générateur de la Requête réécrit la requête initiale en remplaçant la ressource demandée par la nouvelle ressource jugée similaire (par le Fournisseur de Similarité) puis l'envoie vers l'Evaluateur de Requêtes qui réévalue la nouvelle requête R' (9b) et répond à l'utilisateur avec un refus d'accès pour sa demande initiale et une permission pour accéder aux ressources alternatives (10).

3.3 Bilan

Dans cette partie, nous avons détaillé le modèle PS-RBAC et l'architecture PSQRS que nous avons proposés pour réaliser une prise de décision sensible au contexte et à la situation de l'utilisateur. Dans ce qui suit, nous allons montrer des exemples d'application dans lesquels nous avons validé notre proposition.

4. Des Scénarii d'Applications

La proposition du modèle PS-RBAC et de la réécriture de requêtes XACML constitue une solution générique qui a été validée dans différents contextes applicatifs.

4.1 Application au service des Equipes Mobiles Gériatriques EMG

Dans cette section, nous présentons un exemple d'application au sein d'un service d'Equipes Mobiles Gériatriques EMG. Notre travail s'appuie sur les résultats d'un projet réalisé par l'Université de Toulouse (Faculté de Médecine et Laboratoire de Gestion et Cognition) pour le ministère de Santé[10] dont l'objectif est d'évaluer la performance des services EMG de la région Midi Pyrénées et leur efficacité dans la prise en charge des personnes âgées fragiles [Arthus et al., 2009].

Comme nous allons le montrer, l'assurance de l'accessibilité aux ressources des systèmes de santé depuis n'importe où est un critère très important pour le fonctionnement de cette équipe mobile. Par contre, la transparence d'accès exigée (dans des cas d'urgence et dans des scénarii critiques temps réel) se contredit avec la rigidité des contraintes de sécurité des systèmes de santé. Dès lors, l'utilisation d'un modèle du contrôle d'accès sensible au contexte et à la situation nous apparaît incontournable.

Une intervention d'une EMG prend lieu après la réception d'un fax d'un «bon de demande d'intervention» adressé par un service. En analysant les types de demande d'une mission, nous distinguons deux modes d'activité :

- ***Activité intra-hospitalière*** où l'EMG se mobilise pour effectuer une intervention dans un autre service dans le même hôpital ;
- ***Activité extra-hospitalière*** où l'EMG se déplace vers un service hors l'hôpital tel que le centre de soins de suite et de réadaptation SSR, l'établissement d'hébergement

[10] Contrat HAS/CNSA, n°07/0008, INSERM U558 – Département de Santé publique faculté de médecine de Toulouse et Laboratoire Gestion et Cognition (EA 2048) - Université Toulouse III - Paul Sabatier.

pour les personnes âgées dépendantes EHPAD, le centre local d'information et de coordination (CLICS).

L'activité demandée vise normalement à traiter un patient et pour cela, l'équipe – composée d'une secrétaire, d'une infirmière, d'une aide-soignante et d'un médecin spécialiste – aura besoin d'accéder aux différentes ressources médicales puis, de construire un dossier d'événements gériatriques.

En considérant la caractéristique mobile de l'équipe, nous soulignons l'importance de pouvoir réaliser ces missions en accédant aux ressources d'information depuis n'importe quel service/localisation, à n'importe quel moment et en utilisant n'importe quelle machine et système d'information.

4.1.1 Les caractéristiques pervasives des systèmes de santé

En analysant les caractéristiques des systèmes de santé, nous pouvons dire qu'ils sont de plus en plus centrés sur l'utilisateur (médecin, infirmière, patient, etc.) et qu'ils utilisent des technologies orientées service pour garantir une certaine qualité. Dans ces systèmes, la qualité de services est critique car elle touche la vie du patient. La fig. 6, montre l'importance de l'interaction entre les différents sous-systèmes qui interagissent pour le service du patient (y compris les EMG).

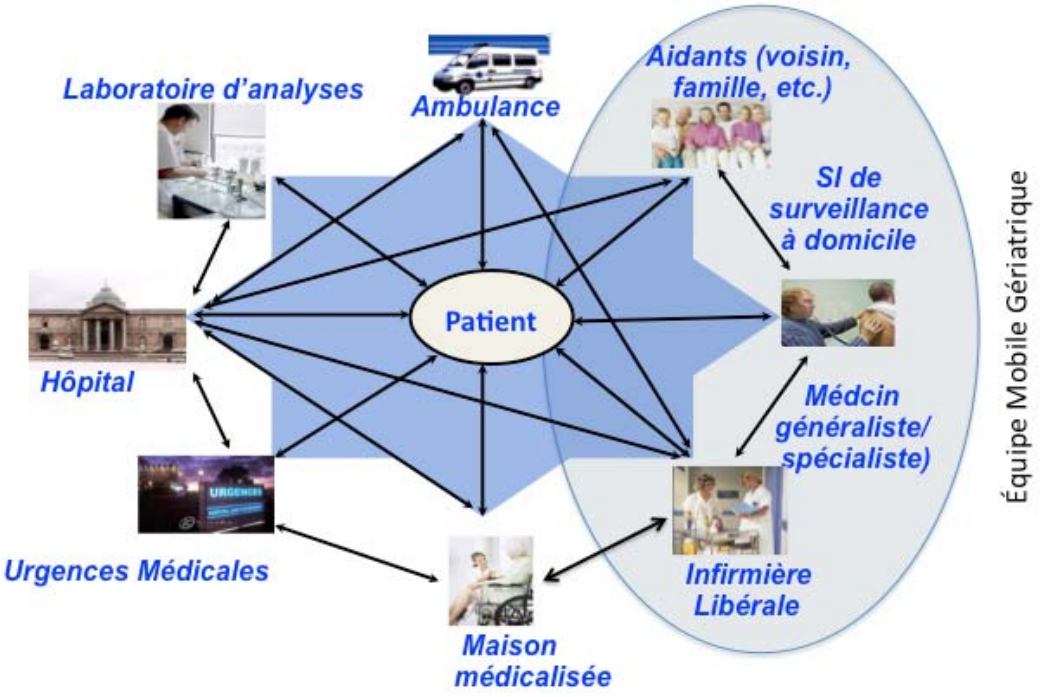


Figure 6 : L'interaction des sous composants d'un Système de Santé Pervasif

La sensibilité et la confidentialité des données médicales justifient le fait qu'elles soient conservées dans leurs ressources d'origine et distribuées dans différents sous-systèmes (hôpitaux, laboratoires d'analyse, cabinets de médecin, etc.). Cette décentralisation influence la gestion d'accès aux ressources médicales qui passe par la distribution des privilèges d'accès selon le rôle de l'utilisateur dans la hiérarchie du système en utilisant le standard RBAC [Ferraiolo et al., 1992].

La nature évolutive des données médicales est très intéressante car elle reflète non seulement l'avancement d'une situation d'un patient en prenant en compte l'axe

temporel mais aussi, les différentes interventions des membres de l'équipe médicale qui ont eu lieu. Sachant qu'une grande partie des données médicales est générée et traitée en temps réel, l'administration des privilèges d'accès doit être centralisée pour assurer l'intégrité du système et une prise de décision fiable.

La gestion des droits d'accès est soigneusement appliquée aux ressources médicales qui sont souvent regroupées et classées par patient dans un dossier médical personnel. Ce dossier contient plusieurs types de données (texte, image, vidéo, etc.) qui décrivent l'évolution de la situation d'un patient.

Afin de faciliter le partage de données, elles sont souvent regroupées et sauvegardées dans des documents «semi-structurés» représentés en XML – eXtensible Markup Language – un format textuel qui décrit le contenu et la structure d'un document. La simplicité, l'expressivité et l'interopérabilité de XML ont favorisé son déploiement dans l'échange des données médicales.

Le fait que XML devienne, de plus en plus, un standard d'échange d'information médicale [HL7, 1994] rend nécessaire l'utilisation d'un standard de contrôle d'accès pour gérer la prise de décision dans cet échange. Ainsi, XACML [OASIS, 2003] permet d'appliquer les principes donnés par les législations médicales.

4.1.2. L'importance de la sécurité dans les systèmes de santé

Les données médicales étant considérées par la loi comme des données privées, sensibles et confidentielles, différentes législations internationales et nationales ont été proposées pour assurer la protection des données médicales, en particulier la déclaration de Helsinki [Helsinki, 1964], l'acte de confidentialité « Privacy Act » [Privacy Act, 1974], la loi HIPPA [HIPPA, 1996] et la loi des droits des malades et la qualité de systèmes de santé en France [Loi 2003-303, 2003].

Cette confidentialité justifie le stockage des données médicales dans leurs ressources d'origine et impose plus de contraintes d'accès, particulièrement dans le cas d'une consultation mobile. En conséquence, l'accès aux données dans un système de santé doit respecter les principes de la protection des données personnelles du patient. Ces données ne sont pas accessibles de la même façon pour tous les membres de l'équipe médicale mais sont souvent restreintes aux besoins de la tâche à réaliser par l'utilisateur.

Dans les systèmes de santé pervasifs, l'accès aux données devient plus exigeant car il dépend non seulement du rôle de l'utilisateur ou de l'horaire de sa demande mais aussi de différentes contraintes contextuelles telle que sa localisation, le dispositif qu'il utilise et le réseau avec lequel il se connecte.

La section suivante présente un scénario qui illustre les challenges d'accès lors d'une consultation réalisée par un membre de l'EMG.

4.1.3. Scénario actuel d'une intervention d'une EMG

En fig. 7, nous présentons une version simplifiée de la traduction d'une demande d'accès sous forme d'une requête XACML où un sujet (ici une infirmière à l'EMG de l'hôpital de Toulouse) demande l'accès à une ressource (le dossier médical d'un patient) dans un certain contexte :

- à une certaine heure : 15h28,

- à partir d’une localisation précise : la maison du patient.

```
<Request>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               DataType="http://www.w3.org/2001/XMLSchema#string”>
      <AttributeValue> Sonia Laure </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-Role"
               DataType="http://www.w3.org/2001/XMLSchema#string”>
      <AttributeValue> Nurse </AttributeValue>
    </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-loc"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> Patient House </AttributeValue>
    </Attribute>
    <Attribute  AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-time"
                DataType="http://www.w3.org/2001/XMLSchema#time">
      <AttributeValue> 15.28.49.495000000+02:00 </AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> Medical Report </AttributeValue>  </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue> read </AttributeValue>   </Attribute>
  </Action>
  <Environment/>
</Request>
```

Figure 7 : Une version simplifiée d'une requête XACML

Le résultat de cette la demande de consultation dans cette situation, sera rejetée à cause du contexte courant de l'utilisateur (maison du patient) qui est considéré comme une menace de sécurité pour le système. Dans de tels contextes, il serait intéressant que le système puisse autoriser à cet utilisateur l'accès à d'autres ressources moins sensibles et qui contiennent des informations pertinentes pour la consultation telles que les fichiers contenant les analyses et les radiologies du patient.

4.1.4. Les challenges d'accès et de sécurité dans le processus du recueil et du passage d'information au sein de l'équipe

Dans cette section, nous soulignons l'importance de fournir un système d'information efficace au service de l'EMG lors d'une consultation.

La nature d'une consultation exige la création du dossier patient d'une manière évolutive où chaque membre de l'équipe peut récupérer et consulter les données du patient à partir du système d'information (selon ses droits d'accès) et ensuite, peut insérer ses remarques, annotations, commentaires et autres données au fur et à mesure.

Dans les systèmes actuels, le passage d'information lors de la fin d'une tâche est réalisé directement par chaque membre soit en utilisant des formulaires à moitié remplis à la main (dans le cas d'une intervention extra-hospitalière) soit avec des formulaires imprimés à partir du système informatique (dans le cas d'une intervention intra-hospitalière).

Pour réaliser une meilleure consultation et un passage d'information, le dossier d'événement gériatrique doit être rempli de manière coopérative et interactive. Le spécialiste pourra consulter la situation du patient et la mettre à jour selon plusieurs techniques, en accédant aux flux des données générés en temps réel et extraits à partir des machines dédiées pour la surveillance du patient.

C'est pour cette raison, que nous soulignons l'importance de fournir un système pervasif qui améliore la consultation des données du patient et assure le passage d'information de manière collaborative et interactive.

Grâce à cette évolution des systèmes d'information de santé vers des systèmes pervasifs, la qualité de service sera assurée par une EMG qui pourra traiter le patient depuis n'importe où, n'importe comment et à n'importe quel moment.

Le fait d'avoir un système pervasif résoud la complexité d'accès au dossier du patient et le traitement de données dans un cas de consultation mobile. L'interaction au sein de l'EMG, plus efficace, fournit un échange transparent avec les ressources de données.

4.1.5. L'implantation d'un système de contrôle d'accès adaptatif

Dans cette section, nous soulignons l'importance de fournir un système d'information efficace au service de l'EMG lors d'un processus de consultation.

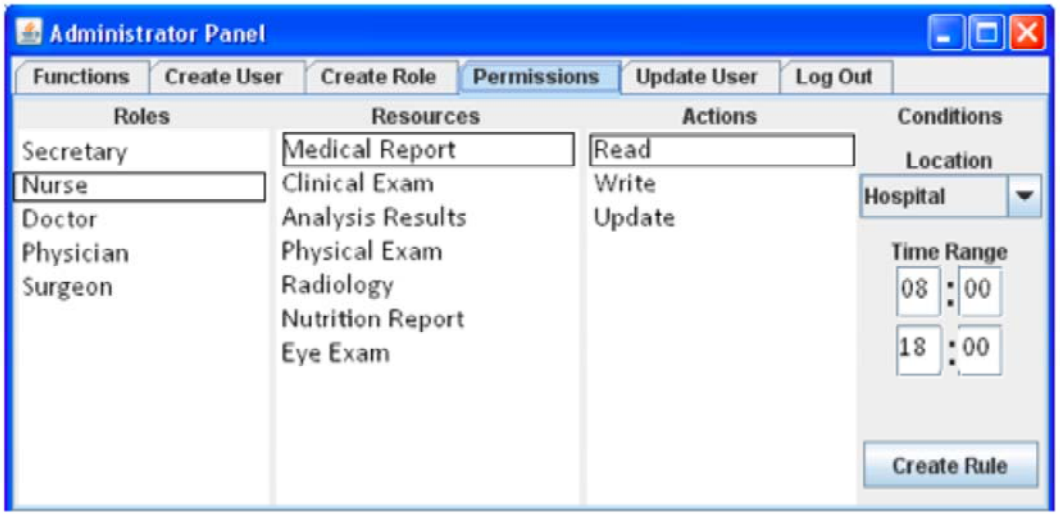


Figure 8 : Prototype chargé de la création des politiques d'accès XACML

Afin de faciliter l'adoption du standard XACML, nous avons fourni un prototype dédié aux administrateurs chargés de la gestion de contrôle d'accès au sein des EMG. Cette gestion passe par la création des politiques de contrôle d'accès sous forme de politiques XACML, cf. fig. 8.

Comme nous le montrons, un utilisateur est représenté par un rôle auquel on peut attribuer différentes permissions pour accéder à différentes ressources selon certaines contraintes contextuelles (ici : localisation et durée). Le résultat d'une telle opération sera la génération d'un fichier XML contenant la politique d'accès, voir fig. 9.

Le prototype est dédié également aux utilisateurs du système (médecins, infirmières, etc.) qui lancent des requêtes à partir de différents contextes. Un exemple d'une requête XACML a été présenté en fig. 9, cette requête est formulée via une interface spécialisée, voir fig. 10. Cette interface interroge la base de données des politiques XACML et dans le cas d'un refus d'accès lors d'une situation critique, le Fournisseur de Similarité cherche s'il y existe des ressources similaires à proposer à l'utilisateur.

```
<Policy PolicyId="GeneratedPolicy"  RuleCombiningAlgId="urn:oasis:
names:tc:xacml:1.0:rule-combining-algorithm:ordered-permit-overrides">
  <Target>
    <Subjects>      <AnySubject/>     </Subjects>
    <Resources>   <AnyResource/>  </Resources>
    <Actions>        <AnyAction/>     </Actions>
  </Target>
  <Rule RuleId="urn:oasis:names:tc:xacml:1.0:hospital-system:rule" Effect="Permit">
    <Target>
      <Subjects>    <Subject>
          <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-
            Role" DataType="http://www.w3.org/2001/XMLSchema#string”>
            <AttributeValue> Nurse </AttributeValue></Attribute>
        </Subject>
      </Subjects>
      <Resources>        <Resource>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
          DataType="http://www.w3.org/2001/XMLSchema#string">
          <AttributeValue>Medical_Report.xml</AttributeValue> </Attribute>
      </Resource>      </Resources>
      <Actions>        <Action>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema#string">
          <AttributeValue>Read</AttributeValue> </Attribute>
      </Action>      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:
                          1.0:function:time-greater-than-or-equal">
        <Attribute> <AttributeValue>08.00.00.495000000+02:00 </AttributeValue>
        </Attribute></Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-
                          equal">
        <Attribute><AttributeValue>18.00.00.495000000+02:00 </AttributeValue>
        </Attribute>     </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <Attribute  AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-loc"
          DataType="http://www.w3.org/2001/XMLSchema#string">
          <AttributeValue>Hospital</AttributeValue> </Attribute>
      </Apply>
    </Condition>
  </Rule>
  <Rule RuleId="FinalRule" Effect="Deny"/>
</Policy>
```

Figure 9 : Exemple d'une politique XACML réalisée avec notre prototype

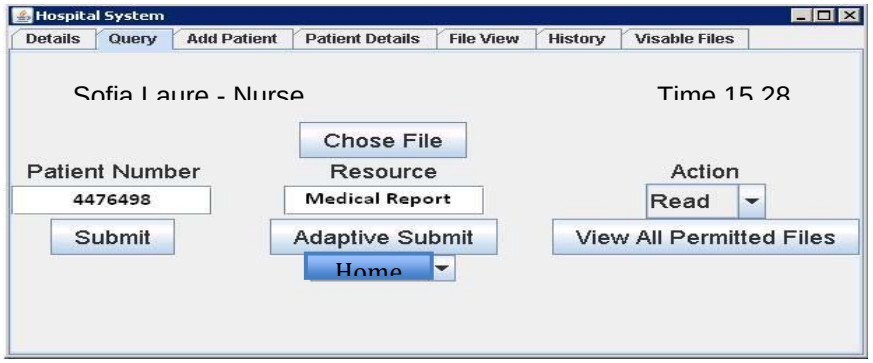


Figure 10 : Exemple d'une requête réalisée avec notre prototype

Le système proposé fonctionne dans un environnement pervasif où la détection des contraintes contextuelles de l'utilisateur sera réalisée implicitement (extraction du numéro du patient à partir de la puce RFID de sa carte d'assurance ou précision de sa localisation en faisant la liaison entre l'adresse actuelle détectée par un GPS intégré dans la machine de l'utilisateur et l'adresse mentionnée dans le dossier du patient ou l'adresse de l'hôpital, etc.). Toutefois, en cas de manque d'informations, ces contraintes peuvent être aussi précisées explicitement par l'utilisateur lors d'une interrogation.

4.1.6. Bilan

La nature pervasive des systèmes de santé, le caractère dynamique du contexte d'une EMG et la richesse des situations d'urgence auxquelles l'équipe est confrontée en temps réel ont formé des éléments d'un cas d'étude idéal pour la validation de notre proposition.

L'accessibilité aux ressources d'information dans les systèmes de santé est un aspect très important en particulier dans une consultation en temps réel ou dans des situations critiques. De plus, le contrôle de cet accès forme une brique de base essentielle pour gérer cette accessibilité.

Dans cette section, nous avons confronté les refus d'accès retournés aux membres des EMG lors d'une consultation effectué en déplacement par la proposition d'un accès flexible et des solutions alternatives.

Pour réaliser cette flexibilité, nous avons utilisé le système PSQRS qui prend en compte le contexte de l'utilisateur et la sensibilité de la situation d'accès. Les solutions proposées sont fournies en utilisant la similarité entre les ressources de données autorisées.

L'utilisation d'un modèle qui fournit des solutions d'accès flexibles ou alternatives lors d'un déplacement de l'EMG permet à l'utilisateur d'avoir un pourcentage de chance plus élevé pour dépanner la situation et d'atteindre une transparence d'accès aux ressources sans menacer la sécurité ou l'intégrité du système.

La flexibilité offerte par le système de réécriture de requêtes PSQRS proposé est liée à la recherche de ressources similaires autorisées (si elles existent) et c'est à ce niveau que nous atteignons les limites de notre proposition lorsque la recherche ne retourne aucune ressource.

L'utilisation d'ontologies médicales est envisagée dans la suite pour enrichir la recherche et avoir plus d'alternatives à offrir à l'utilisateur.

4.2 Application pour l'accès aux ressources d'un SI avionique

Dans cette section, nous exposons les challenges de gestion d'accès à des ressources de SI avioniques qui forment un deuxième cas d'étude pour la validation de notre proposition. Ce travail est basé sur une étude analytique de l'évolution des services d'accès aux ressources de données réalisé au sein du projet GEODESIE[11] [Al Kukhun et Sèdes, 2009b].

Un SI avionique est composé de ressources distribuées dans différents sous-composants, gérées par différents univers, administrées selon différentes contraintes, avec des niveaux d'accès définis sur plusieurs niveaux de confidentialité.

4.2.1 Les caractéristiques des SI avioniques

L'assurance de la transparence et de l'efficacité d'accès en temps réel aux ressources avioniques est un critère indispensable, ce qui peut donner lieu à de nombreux défis face aux difficultés dues à la distribution des données à des niveaux de sécurité hétérogènes. Ces données sont gérées d'une manière décentralisée par l'intermédiaire de différentes bases de données (système avionique, système de vol, système ouvert, système cabine, etc.).

Le besoin est, ici, d'offrir une gestion des données homogénéisée ou logiquement centralisée pour assurer un meilleur partage et une meilleure communication (échange) entre ces systèmes.

Un tel système repose sur différents sous-composants qui définissent des politiques d'accès spécifiques aux ressources qui se complexifient selon la localisation des utilisateurs. La répartition des politiques et l'accès aux données sont donc très critiques en raison des niveaux de sécurité imposés par la certification aéronautique.

Le défi est de gérer les données distribuées, dans des univers incompatibles à priori du point de vue des niveaux d'accès et de concilier la vision de différents univers.

Comme illustré dans la fig. 11, un SI avionique est composé de 3 sous-composants principaux avec des niveaux de sécurité hétérogènes :

1. ***Cabine*** : le SI conçu doit être totalement sécurisé et fermé. Pour assurer le fonctionnement opérationnel du système à tout moment et pour éviter les risques de panne, le système repose sur une architecture redondante où le SI est dupliqué. Ces critères garantissent la sûreté et la sécurité du système. Le SI repose sur un système embarqué conçu pour être extrêmement sûr, tant du point de vue de la sécurité informatique que de la disponibilité opérationnelle.

[11] Projet industriel réalisé par le LAAS, l'ONERA et l'IRIT pour AIRBUS Toulouse.

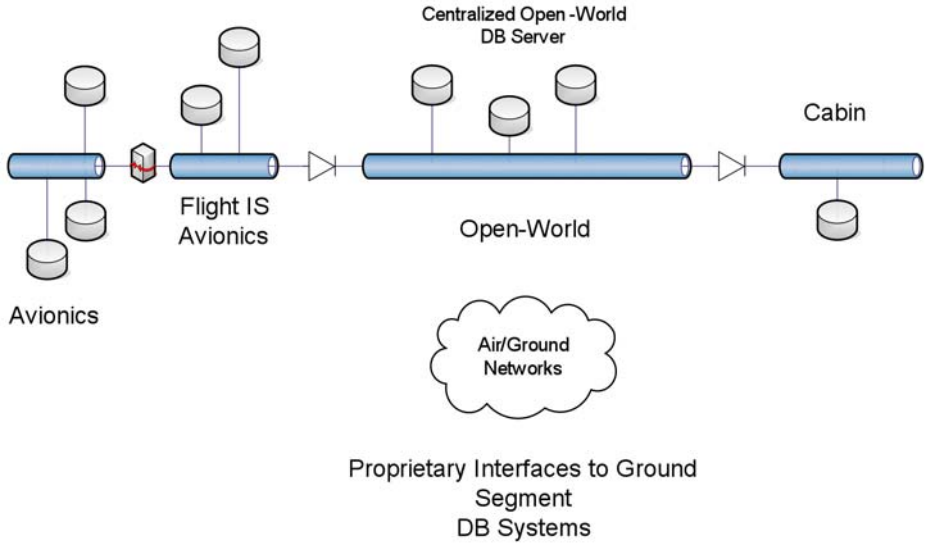


Figure 11 : Les sous-composants d'un SI avionique

2. ***Open-World*** : Ce composant contient un SI qui est totalement ouvert. Il recueille, centralise et compile toutes les données inhérentes au vol et fournit des moyens de communication externes, des capacités de stockage et de calcul des données. Ce système central et modulaire héberge aussi des applications propres à Airbus et aux compagnies aériennes qui portent sur le fonctionnement même de l'avion jusqu'aux services offerts aux passagers (e.g. la documentation électronique de bord, les diagrammes de navigation, des calculs de performance, le journal de bord, etc).

3. ***Avionics & Flight IS Avionics*** : composé de deux parties : La partie ***Avionics*** est un SI dédié à la gestion des données du domaine strictement avionique. Elle est extrêmement sécurisée et possède un haut niveau de confiance. Cette partie doit être en lien avec le sol (pour réaliser des échanges, mises-à-jour, etc.). Par contre, cette connectivité doit respecter les contraintes d'accès et les niveaux de sécurité qui varient selon le service demandeur (service mécanique, thermique, etc.).

La partie ***Flight IS Avionics*** contient les informations et documents relatifs aux opérations de vol. Elle est connectée librement au monde extérieur (système de divertissement de bord IFE et connexions sans fil).

4.2.2 La gestion classique des ressources des SI avioniques

L'accès aux ressources d'information doit être efficace pour acquérir la transparence malgré la localisation des différents SI à des endroits physiques différents avec des niveaux d'accès différents (en fonction de leur confidentialité et de leur niveau d'importance).

L'architecture actuelle ne dispose pas de protocole de stockage. Par conséquent, le système utilise une technique de réplication des données afin de faciliter l'accès aux ressources nécessaires sans saturer le système global. L'inconvénient de cette option est la surcharge du système avec des ressources redondantes qui pourraient ne pas être toujours à jour.

Cette distribution soulève la complexité du partage des données et met en lumière l'importance d'assurer une meilleure gestion d'accès et de sécuriser l'interaction à travers les différents composants du système et via l'entrée/sortie.

4.2.3 Une vision pervasive pour assurer un accès adaptable aux ressources des SI avioniques

Une étude analytique des caractéristiques d'accès requises par les utilisateurs des SI avioniques nous a conduit à envisager ces derniers comme des SI dans lesquels les utilisateurs ont besoin d'accéder aux ressources d'information depuis n'importe où, n'importe quand et n'importe comment, ce qui rejoint les caractéristiques d'un SI pervasif.

Les caractéristiques pervasifs des ressources avioniques sont :

- des données distribuées ;
- générées en temps réel ;
- évolutives ;
- hautement sécurisées ;
- gérées par des privilèges d'accès distribués dans différents univers.

L'absence d'un protocole de stockage des données a justifié la solution existante qui réalise la réplication des ressources de données sur les multiples sous-systèmes afin d'en assurer l'accessibilité.

En l'absence d'un schéma générique cohérant de gestion de données, cette solution a augmenté la complexité qui est non seulement liée : (i) à la surcharge de stockage ; (ii) à la difficulté de la traçabilité des mise-à-jour et (iii) à la consistance des données qui se génèrent et évoluent en temps réel (problème classique lié à la duplication et à la redondance).

Notre contribution tente de résoudre les différents niveaux de la problématique par la définition d'un contrôle d'accès adaptatif utilisant le modèle ***PS-RBAC***. Ce dernier, basé sur le ***modèle RBAC***, offre une gestion d'accès aux ressources distribuées et permet de différencier les différents niveaux de confidentialité définis par les (parties) gérant des ressources avioniques.

Au niveau de l'application, nous proposons d'utiliser le système ***PSQRS***. Ce dernier est basé sur le ***standard XACML*** qui peut faire face au challenge de la distribution et de la volatilité des privilèges d'accès. Cette architecture de partage et de gestion d'accès offre à l'utilisateur une décision d'accès centralisée à partir des privilèges d'accès distribués appartenant aux différents univers.

La sensibilité des ressources de données que les SI avioniques possèdent nécessite l'utilisation d'une démarche de sécurité rigide. En même temps, ces systèmes sont aussi conçus pour affronter des situations d'urgence où le refus d'une demande d'accès risque la vie des passagers.

Nous présentons en fig. 12, une classification des solutions du contrôle d'accès que notre système peut adapter pour résoudre les problèmes rencontrés dans les différentes situations d'urgence qui peuvent avoir lieu dans les SI avioniques. Le choix de la flexibilité d'accès est lié au niveau de sensibilité de la situation rencontrée en temps réel.

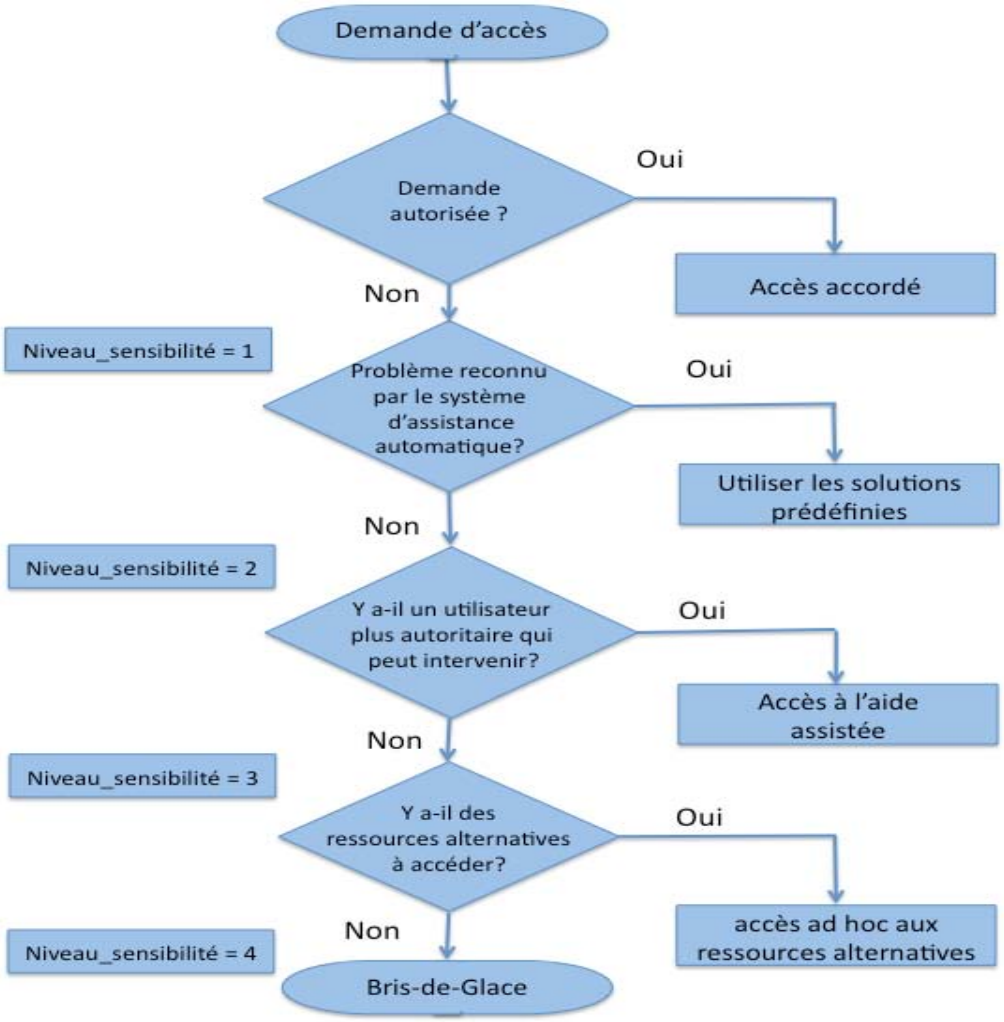


Figure 12 : Exemple du traitement d'une demande d'accès lors d'une situation d'urgence

Niveau 1 : Adaptation rigide avec des solutions prédéfinies

Les solutions offertes à ce niveau sont prédéfinies par les protocoles pré-conçus pour les cas critiques des SI avioniques. Dans ce genre d'adaptation, l'occurrence d'un certain type de scénario critique déclenche une méthode qui autorise à l'utilisateur à accéder à certaines procédures / ressources non autorisées afin de pouvoir réagir et dépanner la situation ou qui permet au système d'appliquer des solutions automatiques prédéfinies.

Niveau 2 : Adaptation assistée

Ce type d'adaptation peut avoir lieu lors de l'occurrence d'une situation urgente dans laquelle l'utilisateur est confronté à un refus d'accès aux ressources avioniques. Ce refus peut être dû à un problème technique touchant le système ou aux contraintes de sécurité liées au rôle de l'utilisateur ou à sa localisation (hors de la zone contextuelle définie par les privilèges d'accès gérant les ressources).

Selon l'importance de la tâche, le système peut intervenir et appliquer un processus adaptatif assisté en transférant cette demande d'accès vers un utilisateur plus autoritaire qui pourra l'aider à accéder aux ressources nécessaires pour dépanner la situation.

Niveau 3 : Adaptation ad hoc basée sur la recherche de solutions alternatives

Nous proposons ce type d'adaptation pour traiter des situations où les solutions existantes ne peuvent pas aider l'utilisateur confronté à un refus d'accès lors d'une situation urgente (e.g. dépannage d'une partie du système, problème de connectivité, etc.).

Le modèle adaptatif proposé basé sur la réécriture des requêtes peut alors intervenir et reformuler les demandes d'accès des utilisateurs en fonction des ressources accessibles afin de trouver des ressources alternatives.

Niveau 4 : Bris-de-Glace

Cette solution est mise en œuvre par le système avionique lors de l'occurrence d'une situation de crise non prédite à l'avance et non gérable par les moyens mentionnés précédemment. La solution comprend une relaxation temporelle des contraintes d'accès posées par défaut. Le risque d'utiliser cette sécurité flexible et de dévoiler les contraintes de sécurité dans un tel contexte s'avère beaucoup moins couteux que de les respecter. Il en va en effet de la vie des passagers.

4.2.4 Bilan

Dans cette section, nous avons montré comment nos travaux peuvent résoudre le problème de la gestion de ressources avioniques. La solution proposée permet d'assurer la connectivité entre les différents sous-composants et un meilleur stockage, partage et mise-à-jour de ressources évolutives.

La gestion d'accès est un défi très important dans les SI avionique. Elle doit, à la fois, respecter des contraintes hautement sécurisées et assurer des solutions efficaces lors de l'occurrence de situations d'urgence où un refus d'accès peut risquer la vie des passagers.

Nous avons étudié les différentes solutions que les SI avioniques offrent lors de l'occurrence d'une situation urgente et nous avons montré que notre proposition (modèle PS-RBAC et système de réécriture de requêtes associé PSQRS) permet d'intégrer les solutions existantes et d'ajouter une solution d'accès adapté permettant en cas de refus d'accès de proposer des solutions alternatives.

4.3 L'accès aux ressources d'un système de vidéo surveillance

Dans cette section, nous présentons un troisième cas d'étude permettant de valider notre proposition. Il s'agit d'un système dédié pour gérer l'indexation et l'accès aux ressources de vidéo surveillance. Notre objectif est de résoudre les problèmes de refus d'accès rencontrés lors des demandes d'obtention de ressources. Cette étude se base sur les travaux réalisés dans le projet LINDO[12] (Large scale distributed INDexation of multimedia Objects).

4.3.1 Contexte général du projet LINDO

Le projet européen LINDO a pour but de développer une architecture générique, dans laquelle non seulement le stockage des données multimédia est distribué, mais également l'indexation qui est répartie sur différentes unités de stockage, éventuellement hétérogènes, éloignées géographiquement et de capacités diverses.

Plutôt que de déplacer les contenus et les métadonnées vers les serveurs centraux de traitement, une solution alternative a été considérée où les routines d'indexation pertinentes sont exécutées sur les sites distants. En conséquence, seule la stricte information nécessaire répondant à une requête sera transférée à l'utilisateur, voir fig. 13.

Figure 13 : Les challenges du traitement d'une requête d'un utilisateur

Cette stratégie distribuée de l'indexation et du stockage des contenus multimédias et de leurs métadonnées est avantageuse car elle vise à éviter de nombreux inconvénients du traitement centralisé tels que :

– ***La lenteur de traitement d'une requête*** : le traitement d'une requête sur la totalité des métadonnées du système a de fortes chances de surcharger le serveur central, surtout lors du traitement de requêtes complexes et lorsque plusieurs requêtes sont traitées simultanément.

– ***La surcharge de la bande passante*** : tous les contenus multimédias ou toutes les métadonnées doivent être transférés par le réseau au serveur central.

[12] http://www.lindo-itea.eu/

– ***La centralisation du système*** : si le serveur central ne répond plus, l'ensemble des métadonnées doit de nouveau être recalculé et renvoyé sur un serveur central. De plus, dans le cas d'un système d'information dynamique, la mise-à-jour du serveur central serait très couteuse.

– ***Le non-respect des droits d'accès sur les données*** : certaines métadonnées ne doivent pas être stockées sur le serveur central pour des raisons, par exemple, de respect de la vie privée.

4.3.2 Le traitement des requêtes dans le système LINDO

Afin d'assurer l'efficacité du système de recherche, les travaux de l'équipe se sont intéressés au processus d'indexation pour améliorer le traitement des requêtes de l'utilisateur en appliquant des algorithmes de filtrage et des reformulations basées sur les liaisons sémantiques des métadonnées [Laborie et al., 2009], [Brut et al., 2011].

Le but principal étant de répondre rapidement à une requête, la solution proposée est de transférer uniquement une version concise des métadonnées (résumé) décrivant le contenu des ressources au serveur central. Ce résumé, au format XML, est extrait lors de la génération des ressources de données en utilisant des algorithmes d'indexation implicites. Par conséquent, le serveur central peut s'en servir pour répondre directement à des requêtes générales.

Quand le besoin exprimé dans la requête de l'utilisateur n'est pas satisfait par les résultats générés par les algorithmes génériques (placés au niveau du serveur central), le système peut réagir et offrir à l'utilisateur la possibilité de sélectionner explicitement des algorithmes qui peuvent réaliser un traitement plus détaillé et récupérer des résultats plus pertinents. Ces algorithmes spécifiques sont localisés sur les serveurs distants.

Nous soulignons l'importance de l'effet du choix d'un algorithme d'indexation sur les résultats restitués par le système. Dans la suite, nous fournissons deux exemples qui montrent la différence de niveau de détail offert par chaque type d'indexation (implicite ou explicite).

La fig. 14 présente un exemple de résultats fournis à partir du processus de traitement automatique réalisé lors de l'acquisition d'un extrait de vidéo surveillance. Ce traitement emploie des ***algorithmes d'indexation implicites*** qui stockent les résultats sur le serveur central. Ces algorithmes d'indexation distinguent deux contextes d'acquisition : intérieur et extérieur.

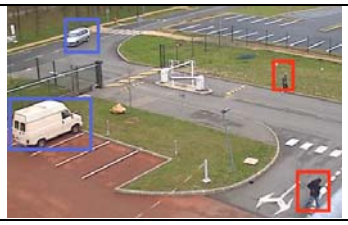
	Indoor	**Outdoor**
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles
		

Figure 14 : Exemple d'informations récupérées par les indexeurs implicites

La fig. 15 présente les résultats produits après l'utilisation d'algorithmes d'indexation explicites, ceux-ci peuvent fournir des détails plus spécifiques sur le contenu. Un traitement plus profond est réalisé, ce qui génère une richesse des métadonnées décrivant le même extrait de la vidéo. De tels algorithmes sont placés sur les serveurs distants et sont choisis à partir des requêtes des utilisateurs cherchant des réponses plus précises.

	Indoor	**Outdoor**
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people - Face recognition - voice recognition & speech-to-text	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles - Car plate number - Face recognition
		

Figure 15 : Exemple d'informations récupérées par le traitement d'algorithmes explicites

Le choix explicite d'un algorithme d'indexation peut rendre des résultats plus précis aux utilisateurs mais cette opération peut avoir un coût de traitement très élevé (selon le niveau de granularité fourni). Pour cette raison, le système restreint, souvent, la possibilité de permettre à un utilisateur de personnaliser sa requête en utilisant des algorithmes explicites plus spécifiques.

Dans la suite, nous insistons sur le fait que la sensibilité du contenu dévoilé est liée au niveau de détail fourni par les algorithmes explicites (placés sur les serveurs distants). Cette sensibilité est fortement liée à la protection des données et est considérée comme une autre raison qui peut empêcher un utilisateur de choisir explicitement un algorithme qui peut personnaliser sa recherche.

4.3.3 Les challenges d'accès lors de l'application d'une couche de sécurité

Notre objectif est ici, d'assurer l'efficacité de la recherche d'information en prenant en compte les challenges de sécurité, voir fig. 16. La sensibilité du contenu des ressources de vidéo surveillance et la loi d'anonymat définie sur le contenu justifie l'application d'un contrôle d'accès qui gère et personnalise l'accès selon le rôle de l'utilisateur. Cette couche est chargée de gérer :

(i) Les droits d'accès des utilisateurs/services aux ressources de données qui varient non seulement en fonction de leur rôle mais aussi en fonction de leur contexte (temps, localisation, etc.) ;

(ii) Les droits d'accès pour l'utilisation des algorithmes explicites : le risque de dévoiler des informations personnelles ou confidentielles s'élève avec le niveau de granularité de détail recherché et fourni par l'algorithme.

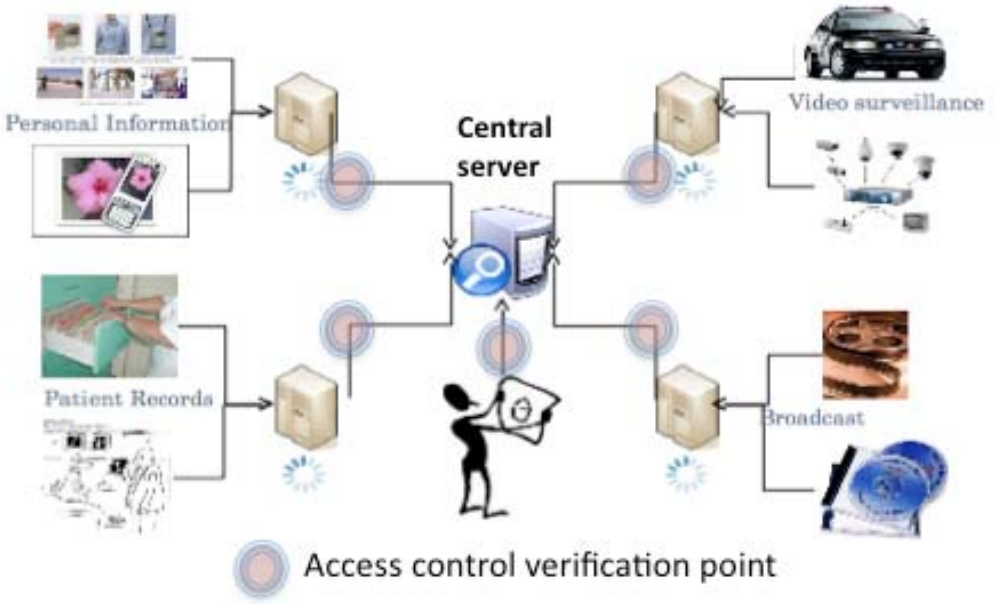


Figure 16 : Les challenges du traitement d'une requête avec la prise en compte d'une couche de contrôle d'accès

Nous soulignons le fait que le manque de retour d'une réponse à la requête de l'utilisateur peut être dû à une restriction d'accès selon les droits d'accès imposés par le système et non à l'absence de résultats. C'est là que l'adaptation des réponses et la recherche de solutions alternatives autorisées nous apparaissent comme des solutions importantes.

L'impossibilité d'accès aux ressources requises peut être également due à des causes éloignées du domaine de la sécurité et plus liées à la performance du système et à la présentation des réponses retrouvées. Ces causes peuvent provenir de problèmes :

- techniques touchant les dispositifs de capture (dépannage de matériels) ou de serveur distant ;
- liés au contexte d'usage des algorithmes d'indexation par les serveurs distants. Le niveau de précision de l'algorithme influence la durée et le coût de traitement et peut surcharger le système (Processor, OS) ;
- de volume du document restitué au niveau de la machine de l'utilisateur (dispositif mobile) ;
- liés au contexte de l'utilisateur (machine, connectivité, espace de stockage, etc.) empêchant la bonne réception du contenu.
- de compatibilité de format associé au dispositif (non supporté par la machine), etc.

Dans ce qui suit, nous proposons une solution qui vise à éviter les refus d'accès dus aux raisons mentionnées précédemment et à fournir des solutions alternatives d'accès.

4.3.4 Le système LINDO vu comme un SIP

Comme nous l'avons déjà mentionné, le manque des réponses pertinentes retournées à l'utilisateur peut ne pas provenir de la non existence des résultats mais est plutôt lié aux restrictions d'accès imposées aux ressources et aux algorithmes d'indexation utilisés ou à la difficulté d'affichage du résultat due aux contraintes contextuelles de l'utilisateur.

En conséquence, notre objectif est de trouver des solutions qui peuvent assurer plus d'accessibilité aux ressources demandées à n'importe quel moment, depuis n'importe où et n'importe comment.

Dans le système LINDO, la complexité d'obtention des résultats est souvent due à des contraintes liées au contrôle d'accès. Ce qui forme un cas d'application idéal pour valider notre proposition en particulier si l'on prend en compte les caractéristiques pervasives de LINDO telles que :

- la distribution des ressources de données ;
- la variation des entités gérant ces ressources ;
- la nature évolutive de ces ressources (générées en temps réel) ;
- la sensibilité et la confidentialité du contenu de ces ressources ;
- la richesse des informations contextuelles ;
- la distribution des algorithmes d'indexation ;
- l'exécution des demandes d'accès en temps réel ;
- le degré d'importance d'obtention de solutions réactives lors de situations critiques.

4.3.5 L'adaptation des décisions d'accès lors de la consultation du système de vidéo surveillance

La sensibilité du contenu des données de vidéo surveillance et le fait qu'elles sont souvent gérées par des lois qui respectent l'anonymat des personnes filmées justifie l'application de restrictions qui filtrent les privilèges d'accès en fonction du rôle de l'utilisateur consultant les ressources (suivant les droits d'accès précisés par le modèle RBAC).

Etant donné que les systèmes de vidéo surveillance sont utilisés pour gérer des situations en temps réel, une option « bris-de-glace » est souvent inclue pour dépanner les cas d'urgence. Le déclenchement du « bris-de-glace » offre à l'utilisateur la possibilité de dépasser les contraintes de sécurité pour mieux réagir. Cette flexibilité aide à dépasser les restrictions imposées sur l'accès au contenu multimédia et facilite l'usage explicite des algorithmes d'indexation.

Entre le respect de la rigidité des décisions d'accès suivies par défaut dans le système et la flexibilité extrême de l'option «bris-de-glace », il existe des situations où les utilisateurs du système ont besoin d'un contrôle d'accès flexible ou modéré. C'est là que notre proposition pourra intervenir pour donner des solutions alternatives d'accès.

La relaxation du contrôle d'accès que nous proposons de mettre en œuvre ne touche pas les droits d'accès gérant le contenu des ressources vidéo, mais elle assure leur respect et leur maintien [Al Kukhun et al., 2012b]. Elle applique la flexibilité et l'adaptation de prise de décision au niveau de deux fonctionnalités principales :

(i) ***Le choix d'utilisation d'algorithmes d'indexation explicites*** (placés sur des serveurs distants).
(ii) ***La présentation du contenu des ressources vidéo*** (l'identité des personnes filmées est souvent protégée et régie par des lois d'anonymat).

La réussite de la procédure adaptative proposée est liée à la richesse des fonctionnalités offertes par les algorithmes d'indexation explicites et par les solutions d'adaptation de présentation qui peuvent aider l'utilisateur à dépasser la complexité des situations rencontrées.

Par conséquent, dans le cas d'un refus d'accès, d'un manque de réponses ou de la réception de résultats non satisfaisants, notre solution peut intervenir pour offrir à l'utilisateur la possibilité de personnaliser la méthodologie de recherche afin de recevoir des solutions adaptées.

Le modèle PS-RBAC et le système PSQRS réalisent l'adaptation des décisions d'accès en prenant en compte non seulement le rôle de l'utilisateur mais aussi ses paramètres contextuels et l'importance de la situation dans laquelle il consulte le système. Une telle solution améliorera la qualité de service côté utilisateur sans menacer la sécurité ou l'intégrité du système.

4.3.6 Scénario d'application

Nous donnons ici, un exemple où la mise en œuvre de notre proposition est utilisée pour pallier le manque des réponses restituées par le système. Comme nous allons l'illustrer, le système va modifier le traitement de la requête et l'adaptation de prise de décision d'accès selon le niveau d'importance de la situation.

4.3.6.1 Cas d'un objet oublié dans une station métro

En prenant le métro de la station Trocadéro vers la place d'Italie à 14h15, Hélène a oublié son sac rouge sur un banc d'attente sur un quai. Dès qu'elle s'en est rendue compte, elle est sortie et s'est rendue au guichet de la station pour signaler le problème.

Le traitement d'une telle situation passe par l'agent de service clientèle qui ouvre un dossier et prend les descriptifs de l'objet perdu et les transmet à l'agent de sécurité sur place. Ce dernier va suivre différentes étapes pour retrouver l'objet : il va vérifier si l'objet a été déjà retrouvé ou remis au service par quelqu'un. Sinon, il va essayer de consulter le système de vidéo surveillance pour vérifier si l'objet est toujours au même endroit.

- *Traitement typique effectué par le système LINDO*

La fig. 17 montre l'interprétation typique réalisée par le système de recherche d'information fourni par le système LINDO. La requête lancée sera traitée et parcourue afin d'extraire les mots-clés qui ensuite seront reformulés sous forme d'une requête XML.

Query 1: Find all videos containing a *red bag*, forgotten in *Trocadéro, Paris* metro station, on *Thursday, 2 Febuary*, between *2:00pm and now (3:00pm)*.

Figure 17 : Structure XML d'une requête envoyée au système de RI LINDO

La nature distributive de la gestion des ressources de données et du traitement des requêtes dans le système LINDO justifie l'utilisation d'une méthode de recherche basée sur le filtrage. L'objectif est de retrouver des résultats qui répondent strictement aux besoins exprimés dans la requête et de minimiser le sous-ensemble de métadonnées que le système doit parcourir en temps réel lors du traitement de la requête.

Après l'extraction des mots-clés de la requête, le traitement de la requête va procéder à la localisation des serveurs gérant les différents flux capturés par les caméras situées dans les quais d'attente de la station Trocadéro. Puis, une étape de filtrage sera effectuée pour restreindre la recherche dans les parties acquises entre 14h00 et 15h00.

Le système va déterminer, ensuite, une liste d'algorithmes d'indexation appropriée à l'ensemble des besoins, des propriétés et des contextes exprimés dans la requête. Cette étape va générer les métadonnées liées à la requête.

Dans ce scénario, les informations demandées sont basiques, la requête sera traitée à l'aide des résultats d'indexation réalisés par les algorithmes implicites placés au niveau du serveur central. Le système va poursuivre la recherche pour trouver un objet rouge dans les métadonnées décrivant les segments choisis.

Un processus de filtrage additionnel est appliqué pour la prise en compte des règles de contrôle d'accès. En examinant les droits d'accès de l'agent de sécurité, nous trouvons qu'il n'a pas l'autorisation de consulter des vidéos qui affichent les visages des passagers, ni d'utiliser les algorithmes d'indexation explicites existants au niveau des serveurs distants.

Par conséquent, le système filtre les ressources en éliminant les parties qui contiennent des visages de personnes et enfin, renvoie à l'utilisateur des segments qui contiennent un objet rouge (s'il en existe).

- *Traitement adaptatif sensible à la situation proposée*

L'analyse des résultats restitués à l'agent de sécurité dans ce cas, montre que ces derniers sont insuffisants. Notre proposition peut intervenir à ce niveau afin

d'améliorer la qualité de service et d'offrir à l'utilisateur plus de ressources accessibles sans dépasser les droits d'accès imposés sur la consultation des ressources de données.

L'utilisation du modèle de prise de décision PS-RBAC proposé permettra au système de modifier le niveau d'accessibilité et d'adapter les permissions offertes à l'agent de sécurité selon son contexte et l'importance de la situation de consultation.

L'utilisation de cette solution est liée au déclanchement de la reconnaissance d'une situation ou d'un contexte par le système. Dans ce scénario, la situation sera reconnue depuis l'identifiant du dossier « objet perdu ».

L'implémentation de notre proposition est réalisée par le système PSQRS qui adapte la prise de décision par la réécriture des requêtes XACML. Cette solution a prouvé son efficacité par sa capacité à fournir une prise de décision d'accès à partir de politiques distribuées et à prendre en compte des éléments contextuels liés à la requête.

Par conséquent, cette simple requête lancée par l'agent de sécurité (composée par des mots-clés décrivant le contenu recherché, voir fig. 17) sera incluse dans une demande d'accès sous forme d'une requête XACML plus structurée et enrichie des métadonnées (décrivant les contraintes contextuelles de l'utilisateur, son rôle, le niveau d'importance de la situation dans laquelle il consulte le système, etc. voir fig. 18, fig. 19).

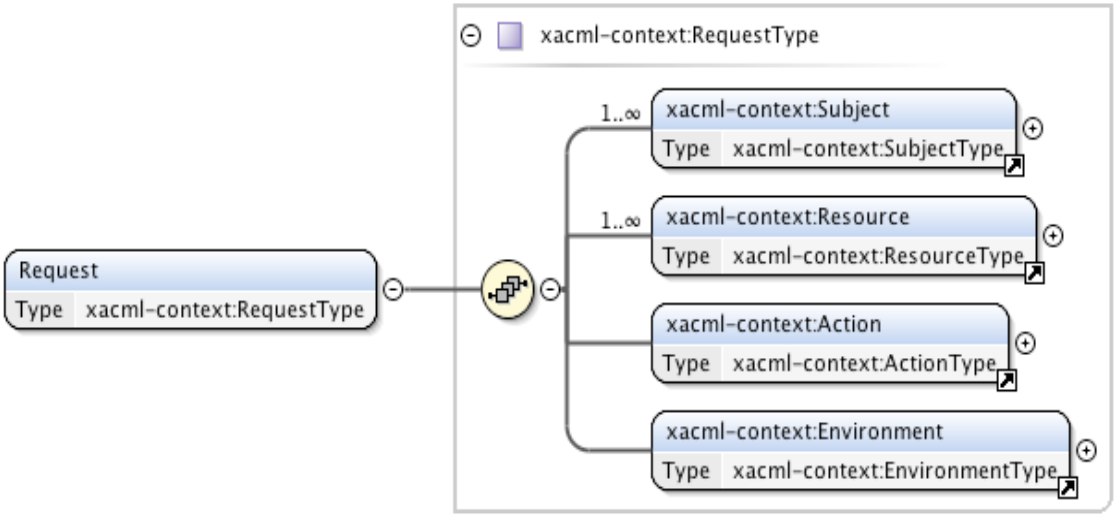


Figure 18 : Schéma générique d'une requête XACML

Comme le montre la fig. 19, la richesse des métadonnées des requêtes XACML permet de représenter les caractéristiques contextuelles liées :

(i) à la requête dans la balise « **resource** » ;
(ii) à l'utilisateur lançant la requête dans la balise « **subject** » et ;
(iii) à la situation de consultation des ressources demandées dans la balise « **environment** ».

Les informations contextuelles de la requête peuvent être vérifiées et réinterprétées au sein du système ensuite. Il s'agit de :

❖ ***La période estimée*** par l'utilisateur qui peut être vérifiée à partir des informations enregistrées dans sa carte de transport.

❖ ***Les coordonnées GPS*** liées aux segments vidéo recherchés qui seront extraites à partir de la localisation indiquée à la requête.

Coordonnées GPS de Trocadéro : 2°17'59"E, 48°53'59"N (2.3, 48.9)

Dans le cas de la réception d'une décision négative ou non satisfaisante, le système adaptatif va diagnostiquer la situation de recherche à partir du niveau de la situation de recherche incluse dans la balise « environment » :

Sit_Lvl = 0 consultation normale
Sit_Lvl = 1 recherche d'un objet perdu
Sit_ Lvl = 2 recherche d'un enfant perdu

```
<Request ……>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>John Smith</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
      DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>Security Agent</AttributeValue> </Attribute>
    <Attribute
      AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:securityAgent-id"
      DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>sa2023</AttributeValue>  </Attribute>
  </Subject>
  <Resource>
    <ResourceContent>  <UserQuery>  <QueryInText> find all videos
          containing a red bag, forgotten in Trocadéro, Paris metro station,
          on Thursday, 2 Febuary, between 2:00pm and 3:00pm).</QueryInText>
        <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
        <MediaFormat>Video</MediaFormat>
          <TimeSpan>  <From>2012-02-02T14:00:00</From>
                      <To> 2012-02-02T15:00:00</To>  </TimeSpan>
      </UserQuery> </ResourceContent>  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Read</AttributeValue> </Attribute>
  </Action>
  <Environment>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:environment-id"
    DataType="http://www.w3.org/2001/XMLSchema#string">
    <AttributeValue>Situation</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:situation-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Forgotten Object</AttributeValue> </Attribute>
     <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:sitLevel-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>1</AttributeValue> </Attribute>
  </Environment> </Request>
```

Figure 19 : Requête XACML englobant la requête de l'utilisateur

Le niveau d'importance de la situation va déterminer le niveau d'adaptation qui sera réalisé ensuite. L'activation du mode de recherche adaptatif sera communiquée à partir de la réponse XACML sous la forme d'une « **obligation** » qui accompagne la réponse, voir fig. 20.

```
Response>
   <Result>
      <Decision>Deny</Decision>
      <Status>
         <StatusCode Value="urn:oasis:names:tc:xacml:2.0:status:ok"/>
      </Status>
      <Obligation FulfillOn="Deny" ObligationId="ApplyAdaptiveQueryingMode">
          <AttributeAssignment AttributeId="AQM"
                DataType="http://www.w3.org/2001/XMLSchema#string">
                On
          </AttributeAssignment>
      </Obligation>
   </Result>
</Response>
```

Figure 20 : Réponse XACML avec les obligations à suivre

Le déclanchement du mode adaptatif (Adaptive Querying Mode) va changer le processus du traitement de la requête afin d'assurer la réussite de recherche en proposant des solutions adaptatives.

Cette solution est réalisée dans le système PSQRS au niveau du ***Fournisseur de Sensibilité de la Situation*** qui détecte la situation puis, s'oriente vers le ***Fournisseur de Similarité*** pour réaliser la réécriture de la requête (c.f fig. 5).

Dans le cas où la situation de consultation est normale (Sit_Lvl = 0), le système réalisera une reformulation sémantique des mots-clés de la requête en utilisant des mots similaires ou des concepts plus génériques au niveau de ***Fournisseur de similarité***. Un travail similaire a été introduit dans [Al Kukhun et Sèdes, 2008], l'objectif étant d'augmenter les chances de restitution des résultats aux utilisateurs malgré les challenges de sécurité.

La reformulation sémantique peut être réalisée avec l'aide d'un dictionnaire lexical standard telque WordNet[13]. Par exemple, le mot « bag » peut être remplacé par différents synonymes {backpack, lugguage, purse, etc.}. L'emploi de la reformulation a été également proposé par les travaux de l'équipe [Brut et al., 2011] mais pas dans le cadre d'un processus de requêtage sensible aux droits d'accès.

Au niveau du traitement du scénario courant, le niveau d'importance de la situation est plus élevé (Sit_Lvl = 1). De ce fait, le ***Fournisseur de Similarité*** sera remplacé par un ***Fournisseur de Solutions Adaptatives.*** Ce composant va réaliser une adaptation automatique ou assister l'utilisateur pour adapter sa requête en lui fournissant des propositions de solutions adaptatives sauvegardées dans une base de données prédéfinie. Le tableau 2 montre des exemples de solutions proposées par le système.

[13] http://wordnet.princeton.edu/

L'alimentation de la base de données peut aussi être effectuée par une méthode d'apprentissage automatique à partir des solutions proposées par les utilisateurs en fonction des situations rencontrées en temps réel.

La réussite de telles solutions adaptatives ou alternatives (proposées par les utilisateurs) sera plus probable si on connaît la cause d'un refus d'accès. Les messages d'erreur qui accompagnent souvent les réponses négatives retournées peuvent servir comme des indicateurs pour trouver des solutions alternatives.

Problème	Solution Adaptative
Loi d'anonymat imposée au contenu des ressources vidéo capturées	
Visage non-autorisé	Afficher le contenu après l'emploi d'un algorithme qui applique la fonction « blur » sur les visages.
Voix non-autorisée	Utiliser un algorithme de transcription textuelle « speech-to-text ».
Volume de vidéo	
Manque de capacité de stockage sur la machine de l'utilisateur.	Utiliser un algorithme de compression ou de conversion vers un format plus léger.
Format non supporté par la machine.	Utiliser un algorithme de conversion vers un format compatible.
Difficulté de téléchargement due à la faiblesse de la bande passante du réseau.	Utiliser un algorithme de synthèse du contenu de vidéo ou héberger les ressources et les consulter à partir d'un espace externe « Cloud computing ».

Tableau 2 : Exemples des solutions adaptatives proposées par notre système

Par conséquent, la solution adaptative pour cet exemple va modifier le processus du traitement et va : (i) négliger l'étape de filtrage chargée d'imposer les contraintes du contrôle d'accès et (ii) la remplacer par une étape adaptative liée à la présentation de ressources ayant du contenu non-autorisé.

En appliquant ce processus au scénario décrit précédemment, le système va restituer - s'ils existent - les segments vidéo capturés dans la station Trocadéro entre 14h00 et 15h00 et qui contiennent un objet rouge.

Ces résultats seront classifiés de façon à détecter les parties non autorisées (contenant des visages de personnes) et c'est là que le système appliquera un processus de filtrage qui adapte leur affichage pour qu'ils soient conformes aux restrictions d'accès imposées par le système.

L'adaptation de présentation consistera à la détection des visages puis à l'utilisation d'un algorithme qui applique la fonction de floutage « blur » sur les visages apparaissant dans ces segments afin de les présenter à l'utilisateur en respectant les règles d'accès.

4.3.7 Bilan

La nécessité de manipuler des contenus multimédias de haute résolution, créés à partir de plusieurs sources dans des environnements distribués, s'impose et pose de nouvelles problématiques d'indexation et d'accès à l'information : stockage réparti, traitement distribué, génération de descripteurs, formats de métadonnées, recherche d'information temps réel, géolocalisation dans des contextes mobiles ou pervasifs, etc. Dans ce contexte, le transfert de contenus multimédias pertinents, depuis leurs sources de stockage, est devenu un réel verrou, notamment en ce qui concerne la performance d'accès à l'information.

L'objectif du projet LINDO était de construire une architecture générique de gestion des données multimédia distribuées. Cette étape, destinée à améliorer le fonctionnement du système, avait pour objectif d'assurer l'efficacité de l'indexation et du stockage de données acquises en temps réel.

Les travaux de l'équipe se sont intéressés à améliorer la pertinence des requêtes posées au système par l'utilisation de métadonnées et par la reformulation des requêtes utilisateurs par des relations sémantiques.

Nos travaux ont pour objectif d'assurer la qualité de service coté utilisateur en prenant en compte le problème de l'accès aux données multimédia dû aux règles imposées du contrôle d'accès. L'objectif est d'atteindre une accessibilité pervasive où l'utilisateur pourra accéder aux ressources de données à n'importe quel moment, depuis n'importe où et n'importe comment.

Afin d'atteindre cet objectif, nous avons employé une solution adaptative du contrôle d'accès sensible au contexte et à la situation de recherche. La solution surmonte les refus d'accès restitués en temps réel en modifiant le processus de traitement des requêtes et en proposant des solutions adaptatives pour contourner l'effet des politiques de contrôle d'accès.

La solution proposée se situe dans une zone intermédiaire entre le respect de la rigidité des décisions d'accès et la flexibilité extrême de l'option «bris-de-glace » qui est souvent employée dans les situations critiques.

La solution proposée peut augmenter la complexité du traitement de la requête mais si l'on considère l'utilité des résultats fournis en temps réel et le fait qu'ils ne violent pas les droits d'accès, cette complexité nous semble tout à fait acceptable.

5. Conclusions et Perspectives

L'évolution des systèmes d'information pervasifs a introduit des challenges très importants au niveau de la sécurité et de la gestion d'accès aux ressources de données. Ces systèmes doivent, à la fois, permettre aux utilisateurs d'obtenir une accessibilité transparente (depuis n'importe où, n'importe quand et à n'importe quel moment) et protéger l'intégrité du système et respecter la confidentialité des données en posant des politiques de sécurité rigides.

Pour relever ce challenge, différents travaux de recherche se sont dirigés, d'un côté, vers la proposition de modèles du contrôle d'accès qui adaptent les décisions d'accès selon le contexte de l'utilisateur. D'un autre côté, d'autres travaux se sont orientés vers la proposition de modèles sensibles à la situation et de solutions parfois extrêmes telles que le « bris-de-glace » applicables à des situations temps réel.

Dans ce cadre, l'objectif de nos travaux était de proposer un modèle qui puisse mettre en œuvre une prise de décision adaptative sensible au contexte et à la situation et un système qui offre une flexibilité modérée sans dépasser les règles d'accès. Une telle proposition peut fournir une meilleure qualité de service et une transparence d'accès sans menacer la sécurité ou l'intégrité du système.

Pour réaliser cet objectif, nous avons présenté PS-RBAC : un modèle du contrôle d'accès adaptatif qui étend le modèle RBAC en prenant en compte la sensibilité au contexte et à la situation de l'utilisateur lors d'une prise de décision.

Ensuite, nous avons mis en œuvre cette prise de décision par PSQRS : un système basé sur la réécriture des demandes d'accès (exprimé sous forme des requêtes XACML) et qui dispose d'un mécanisme de prise de décision adaptatif qui réagit, dans le cas d'un refus d'accès, en cherchant des ressources alternatives ou similaires à celles demandées par l'utilisateur et qui lui sont autorisées.

Nous avons mis en évidence l'importance de fournir une flexibilité de prise de décision d'accès et des solutions alternatives en particulier dans des environnements dynamiques tels que les SI pervasifs. Notre proposition tente de répondre à cet objectif et a été validé dans trois domaines d'application riches en scénarii temps réel : (i) les Equipes Mobiles Gériatriques; (ii) les systèmes avioniques; (iii) un système de vidéo surveillance.

Au cours de notre travail, nous nous sommes intéressés à résoudre les problèmes de refus d'accès rencontrés lors de la consultation d'un SIP dans des situations temps réel. En supposant qu'un refus d'accès est dû aux restrictions imposées au contenu des ressources du système, la solution présentée met en œuvre la recherche de ressources alternatives accessibles. Toutefois, un refus d'accès peut être également lié aux problèmes de non disponibilité des serveurs fournissant cet accès ou aux problèmes d'alignement des politiques d'accès gérant les ressources de données.

Par conséquent, nous avons identifié de nombreuses perspectives aux travaux présentés dans ce mémoire.

Un premier travail est envisagé à court terme où nous allons nous concentrer sur l'amélioration du système implémenté et du processus de recherche de solutions alternatives. Une piste intéressante pour assurer la proposition de solutions similaires sera de les extraire à partir d'ontologies de domaine des bases de données spécialisées.

Si l'on considère les domaines de validation de nos travaux, nous pouvons citer quelques exemples : les ontologies médicales telles que MESH[14], UMLS[15], GALEN[16], DICOM[17], une base de données des anomalies des systèmes avioniques, les ontologies liées aux ressources audio-visuelles.

Des perspectives à moyen terme sont également envisagées pour faire évoluer le système afin d'offrir une prise de décisions d'accès adaptative aux ressources distribuées dans différents services. Dans une architecture orientée service, nous pouvons appliquer le processus d'adaptation et de recherche d'alternatives non seulement au contenu des ressources recherchées mais aussi aux services/serveurs qui fournissent/stockent ces ressources ou à la collection des politiques d'accès qui les gère.

Finalement, une perspective à long terme consistera à valider nos travaux par une évaluation globale du système développé à grande échelle.

[14] MEdical Subject Headings http://www.nlm.nih.gov/mesh/
[15] Unified Medical Language System http://www.nlm.nih.gov/research/umls/
[16] http://www.opengalen.org/index.html
[17] Standard for Digital Imaging and Communications in Medecine http://medical.nema.org/dicom/2001.html

REFERENCES

[Al Kukhun and Sèdes, 2006] Al Kukhun D. and Sedes F., "A Taxonomy for Evaluating Pervasive Computing Environments", Mulitmodal and Pervasive Services Workshop, MAPS 06, IEEE Conference on Pervasive Services, 29 June 2006, Lyon, France.

[Al Kukhun and Sèdes, 2007] Al Kukhun D., Sedes F., "Interoperability in Pervasive Enterprise Information Systems - A Double-Faced Coin Between Security And Accessability. " ICEIS (3) 2007, p. 237-242.

[Al Kukhun and Sèdes, 2008] Al Kukhun D. and Sedes F., "Adaptive Solutions for Access Control within Pervasive Healthcare Systems". Dans : International Conference On Smart homes and health Telematics (ICOST 2008), Ames, IA, USA, 28/06/2008 - 02/07/2008, vol. 5120, Sumi Helal, Simanta Mitra, JonnyWong, K. Chang, Mokhtari Mounir (Eds.), Springer, LNCS, p. 42-53, juin 2008.

[Al Kukhun and Sèdes, 2009a] Al Kukhun D., Sèdes F., "La mise en oeuvre d'un modèle de contrôle d'accès adapté aux systèmes pervasifs". Application aux équipes mobiles gériatriques. Dans : Document numérique, Hermès, Vol. 12, N. 3, p. 59-78, décembre 2009.

[Al Kukhun and Sèdes, 2009b] Al Kukhun D. and Sedes F., "Etude des services d'accès aux ressources de données", Contribution au WP5 Modélisation du projet GEODESIE, Rapport interne, Juin 2009.

[Al Kukhun et al., 2012a] Al Kukhun D., Sèdes F., Sun Y., Bertino E., "Modélisation du Contrôle d'Accès aux Systèmes Pervasifs: Sensibilité à la Situation et au Contexte", UBIMOB 2012,Cépaduès, p. 158-166.

[Al Kukhun et al., 2012b] Al Kukhun D., Codreanu D., Manzat A.M., Sèdes F. "Applying Pervasive and Flexible Access Control to Distributed Multimedia Retrieval". In The 2nd International Workshop on Information Management for Mobile Applications IMMOA 2012 in conjunction with VLDB 2012, Istanbul, Turkey, 31/08/2012.

[Almenárez et al., 2005] Almenárez F., Marín A., Campo C. et García C., "TrustAC: Trust-Based Access Control for Pervasive Devices", 2nd International Conference on Security in Pervasive Computing, Boppard, Germany, April 2005. P. 225-238.

[Anderson, 2001] Anderson R., "Security Engineering: A Guide to Building Dependable Distributed Systems", Chapter 4, p.51-71, Wiley, 2001.

[Ardagna et al., 2007] C.A. Ardagna, E. Damiani, D. De Capitani di Vimercati, P. Samarati, "XML Security", Security, Privacy, and Trust in Modern Data Management, p. 71-86, Springer 2007.

[Arthus et al., 2009] Arthus I., Montalan M.A., Vincent B., "Quels outils pour piloter la performance d'une Equipe Mobile de Gériatrie", Journal d'Economie Médicale, vol. 27. n° 1-2. 2009, p. 43-59.

[Bell et al., 1973] D. E. Bell and L. J. La Padula, "Secure Computer Systems: Mathematical Foundations". MITRE Corporation, Technical Report 2547, Vol. I, 1973.

[Bertino et al., 2001] Bertino E., Bonatti P., Ferrari E., "TRBAC: A temporal role-based access control model", ACM Trans. Inf. Syst. Secur, vol. 4, n° 3, 2001, p. 191-233.

[Bertino et al., 2005] Bertino E., Catania B., Damiani M.L., Perlasca P., "GEO-RBAC: a spatially aware RBAC", Proc. of the Tenth ACM Symposium on Access Control Models and Technologies (SACMAT 2005), Stockholm, Sweden, p. 29-37.

[Biba, 1977] Biba, K. J. "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.

[Break-Glass, 2004] Break-Glass – An Approach to Granting Emergency Access to Healthcare Systems. White paper, Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC), International Medical Informatics, 2004.

[Brucker and Petritsch, 2009] Brucker A., Petritsch H. 2009. Extending access control models with break-glass. SACMAT 2009 : 197-206.

[Brut et al., 2011] Brut M., Codreanu D., Dumitrescu S. D., Manzat A.M., Sedes F.: A Distributed Architecture for Flexible Multimedia Management and Retrieval. DEXA (2) 2011, p. 249-263.

[Bouidghaghen et al., 2010] Bouidghaghen O., Tamine L., Daoud M., Laffaire C.: Contextual evaluation of mobile search. Workshop on Contextual Information Access, Seeking and Retrieval Evaluation, Milton Keynes, 28/03/2010-28/03/2010, Vol. 569, Bich-Liên Doan, Joemon Jose, Massimo Melucci, Lynda Tamine (Eds.), CEUR Workshop Proceedings, 2010.

[Campbel et al., 2002] Campbel R., Al-Muhtadi J., Naldurg P., Sampemane G. and Mickunas M. D., "Towards Security and Privacy for Pervasive Computing", Proceedings of International Symposium on Software Security, Tokyo, Japan, 2002.

[Catarci et al., 2008] Catarci T., de Leoni M., Marrella A., Mecella M., Salvatore B., Vetere G., Dustdar S., Juszczyk L., Manzoor A., Truong H.L.: Pervasive Software Environments for Supporting Disaster Responses. IEEE Internet Computing 12(1): 26-37 (2008).

[Chae et al., 2006] Chae S.H., Kim W., Kim D.K., "uT-RBAC : Ubiquitous role-based access control model", IEICE transactions, 2006, vol. 89, n° 1, p. 238-239.

[Chen et al., 2004] Chen E., Zhang D., Shi Y., and Xu G., "Seamless Mobile Service for Pervasive Multimedia", Advances in Multimedia Information Processing, PCM 2004: 5th Pacific Rim Conference on Multimedia, Tokyo, Japan, 30 Nov – 3 Dec, 2004. Proceedings, Part II.

[Cheng et al., 2007] Cheng P.C., Rohatgi P., Keser C., Karger P., Wagner G., Reninger A.S. "Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control". IEEE Symposium on Security and Privacy 2007, p. 222-230.

[Cheverest et al., 2000] Cheverest K., Davies N. and Mitchel K., "Developing a context aware electronic tourist guide: Some issues and experiences", in Proceedings of ACM CHI '00, pp 17-24, ACM Press, New York, 2000.

[De Capitani di Vimercati et al., 2007] De Capitani di Vimercati S., Foresti S., and Samarati P., Authorization and Access Control, in Security, Privacy and Trust in Modern Data Management, M. Petkovic, and W. Jonker (eds.), p. 39-53, Springer-Verlag, 2007

[Dey, 1999] Dey, A.K. and Abowd, G.D. (1999). Toward a better understanding of context and context-awareness. GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology.

[Dey, 2001] Dey A.K., "Understanding and Using Context", Personal and Ubiquitous Computing Journal, vol. 5, n° 1, 2001, p. 4-7.

[Duan and Canny, 2004] Duan Y. and Canny J., "Protecting User Data in UbiComp: Towards trustworthy environments", Privacy Enhancing Technologies (PET 2004). Selected Papers, p.

167.

[Emami et al., 2007] Emami S. S., Amini M., and Zokaei S., "A Context-Aware Access Control Model for Pervasive Computing Environments", Proceedings of the international Conference on intelligent Pervasive Computing (October 11-13, 2007), IPC. IEEE Computer Society, Washington, DC, p. 51-56.

[Ferraiolo et al., 1992] David F. Ferraiolo and D. Richard Kuhn, Role-Based Access Controls, proceedings of the 15th National Computer Security Conference, p. 554-563, 1992.

[Ferreira et al., 2006] Ferreira A., Cruz-correia R., Antunes L., Farinha P., Oliveira-palhares E., Chadwick D. W., Costa-pereira A., "How to Break Access Control in a Controlled Manner", CBMS 2006, p. 847-854.

[Ferreira et al., 2009] Ferreira A., Chadwick D.W., Farinha P., Correia, R. J. C., Zhao G., Chilro R, Antunes L. F. C. "How to securly break into RBAC: The BTG-RBAC Model". ACSAC 2009, p. 23-31.

[Gschwind et al., 2002] Gschwind T., Jazayeri M. and Oberleitner J., "Pervasive Challenges for Software Components", 9th International Workshop of Radical Innovations of Software and Systems Engineering in the Future, RISSEF 2002, Venice, Italy, October 7-11, LNCS, p.152-166.

[Halpin, 2001] Halpin T. "Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design". Morgan Kaufmann Publishers Inc., 2001.

[Hansen et al., 2003] Hansen F., Oleshchuk V., "SRBAC: A Spatial Role-Based Access Control model for mobile systems", Proceedings of the 7th Nordic Workshop on Secure IT Systems, 2003, Gjvik, Norway.

[Harrison et al., 1976] Harrison M., Ruzzo W. and Ullman J. "Protection in Operating Systems". Communications of the ACM (1976) volume 19, no. 8, p. 461–471.

[Helsinki, 1964] Déclaration d'Helsinki de l'association médicale en 1964. http://www.genethique. org/carrefour_infos/textes_officiels/ titres_textes/declaration_helsinki_2000.htm.

[Henricksen et al., 2004] Henricksen K. and Indulska J. "A Software Engineering Framework for Context-Aware Pervasive Computing". In PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04), p. 77, 2004.

[HIPPA, 1996] Health Insurance Portability and accountability act of 1996, http://aspe. hhs. Gov / admnsimp /pl104191.htm#261

[HL7, 1994] HL7, Health Level Seven, accredited by ANSI in 1994. http://www.hl7.org/

[Jajodia et al., 1991] Jajodia S., Sandhu R. "Toward a multilevel secure relational data model". In Proc. of the ACM SIGMOD Conference on Management of Data, Denver, CO, USA. 1991

[Jin, 2006] Jin X., "Applying Model Driven Architecture approach to Model Role Based Access Control System", Master thesis, University of Ottawa, 2006

[Joshi et al., 2001] Joshi J., Ghafoor A., Aref W. G., Spafford E. H., "Digital Government Security Infrastructure Design Challenges". IEEE Computer 34(2): 66-72 (2001)

[Joshi et al., 2005] J. Joshi, E. Bertino, U. Latif, A. Ghafoor: A Generalized Temporal Role-Based Access Control Model. IEEE Trans. Knowl. Data Eng. 17(1): 4-23 (2005)

[Kawagoe et al., 2011] Kawagoe K. and Kasai K., Situation, "Team and Role based Access Control", Journal of Computer Science, Vol 7, No. 5, p. 629-637, 2011.

[Keppler et al., 2006] Keppler D., Swarup V., Jajodia S. "Redirection policies for mission-based information sharing", SACMAT 2006, p. 210-218.

[Kim et al. 2009] Kim J. D., Jeong D. and Baik D. K., "Extending the UbiMDR Supporting Situation-Aware Access Control", International Journal of Advanced Science and Technology, Vol. 3, p. 33-40, Feb. 2009.

[Kuhn et al., 1997] Kuhn D.R. (1997). "Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems". 2nd ACM Workshop Role-Based Access Control. P. 23–30.

[Kulkarni et al., 2008] Kulkarni D., Tripathi A., "Context-aware role-based access control in pervasive computing systems", SACMAT 2008, p. 113-122.

[Laborie et al., 2009] Laborie S., Manzat A.M., Sedes F.: Création et utilisation d'un résumé de métadonnées pour interroger efficacement des collections multimédias distribuées. INFORSID 2009: p. 227-242.

[Lampson, 1974] Lampson, B. (1974). Protection. In ACM SIGOPS Operating Systems Review, volume 8, no. 1, p. 18–24.

[Lim and Shin, 2007] Lim T., Shin S., "Intelligent Access Control Mechanism for Ubiquitous Applications", ICIS 2007, 11-13 July 2007, p. 955-960.

[Loi 203-303, 2003] Loi n°2002-303 du 4 mars 2002 relative aux droits des malades et à la qualité du système de santé. http://www.legifrance.gouv.fr/WAspad/ UnTexteDeJorf?numjo=MESX0100092L

[Lord, 2002] P. Lord, "Managing E-Business Security Challenges. Technical report", Oracle Corporation, Redwood Shores, CA, USA. 2002.

[Luther et al., 2008] M. Luther, Y. Fukazawa,M. Wagner and S. Kurakake, "Situational reasoning for task-oriented mobile service recommendation", The Knowledge Engineering Review, Vol. 23:1, p. 7–19. 2008

[maymi et al., 2008] F. Maymi, M. Rodriguez-Martinez, Y. Qian, P. C. Manz: Ancile: Pervasively Shared Situational Awareness. IEEE Internet Computing 12(1): p. 48-50 (2008)

[Munoz et al., 2005] J. Munoz and V Pelechano, "Building a Software Factory for Pervasive Systems Development", Advanced Information Systems Engineering: 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005.

[NCSC, 1987] National Computer Security Center, "A Guide to Understanding discretionary Access Control in Trusted systems", 1987.

[NCSC, 1988] National Computer Security Center (NCSC), "Glossary of Computer Terms," Report NSCD-TG-004, Fort Meade, Md.: NCSC, 1988.

[NIST, 1994] NIST Special Publication 800-7, National Institute of Standards and Technology, October 1994.

[NIST, 2002] NIST, "The Economic Impact of Role-based Access Control", 2002, http://www.nist.gov/director/prog-ofc/report02-1.pdf .

[NIST, 2006] Assessment of Access Control Systems, Interagency Report 7316, National Institute of Standards and Technology, September 2006.

[OASIS, 2003] OASIS, “A brief Introduction to XACML”, 14 mars 2003, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

[OASIS, 2005a] OASIS, “eXtensible Access Control Markup Language 3 (XACML) Version 2.0“, February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[OASIS, 2005b] OASIS, “Core and hierarchical role based access control (RBAC) profile of XACML v2.0“. February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf

[Osborn, 2007] S. L. Osborn, Role-Based Access Control, in Security, Privacy and Trust in Modern Data Management, M. Petkovic, and W. Jonker (eds.), p. 55-70, Springer-Verlag, 2007.

[Park et al., 2004] Park I., Kim W. and Park Y., "A Ubiquitous Streaming Framework for Multimedia Broadcasting Service with QoS based mobility Support" LNCS 3090 in Springer-Verlag (SCI-E), June 2004, pp.65-74.

[Park et al. 2006] Park S, Han Y., Chung T., “Context-role based access control for context-aware application”. High Performance Computing and Communications, vol. 4208, September 2006, Springer Berlin/Heidelberg, pp. 572-580.

[Pascoe, 1998] J. Pascoe, “Adding generic contextual capabilities to wearable computers”, in Proceedings of the 2nd International Symposium on Wearable Computers, pp 92-99, 1998.

[Pfleeger, 1997] Pfleeger C. P., “Security In Computing,” Second Edition, Prentice-Hall PTR, 1997.

[Povey, 2000] Povey D. "Optimistic security : a new access control paradigm", in Proceedings of the 1999 workshop on New security paradigms. ACM Press, 2000, p. 40-45.

[Privacy Act, 1974] Privacy Act of 1974. http://www.house.gov/matheson/the_privacy_act_of_1974.html

[Ranganathan et al., 2005] Ranganathan A., Al-Muhtadi, J., Biehl, J., Ziebart, B., Campbell R. and Bailey R. Towards a Pervasive Computing Benchmark. In PerWare '05 (Workshop on Middleware Support for Pervasive Computing) at the IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), Kauai Island, Hawaii, March 8-12, 2005.

[Rissanen et al., 2004] Rissanen E, Firozabadi S, Sergot M. “Towards a Mechanism for Discretionary Overriding of Access Control”. Proceedings of the 12th International Workshop on Security Protocols, Cambridge. 2004.

[Sandhu, 1993] R. Sandhu. Lattice-based access control models. IEEE Computer, 26:9–19, Nov. 1993.

[Sandhu et al., 1997] Sandhu, R., E. Covne, H. Feinstein, and C. Youman. “Role-Based Access Control Models.” 1997, IEEE Computer vol. 29(2), p. 38-47.

[Sandhu et al., 1998] R. Sandhu, “Role-Based Access Control.” Advances in Computers vol. 46, p. 237-286, 1998.

[Sandhu, 1996] Sandhu R., “Role Hierarchies and Constraints for Lattice-Based Access Controls”, ESORICS 1996, p. 65-79.

[Schilit et al., 1994] B. Schilit, N. Adams, and R. Want. “Context-Aware Computing Applications“. In

IEEE Workshop on Mobile Computing Systems and Applications, p. 85–90, Santa Cruz, CA, US, 1994.

[Strang et al., 2004] T. Strang and C. Linnhoff-Popien. A Context Modeling Survey. In Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, September 2004.

[Wang et al., 2004a] L. Wang, D. Wijesekera, and S. Jajodia. “A Logic-based Framework for Attribute based Access Control”, in 2nd ACM Workshop on FMSE, 2004, p. 45-55.

[Wang et al., 2004b] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung. “Ontology Based Context Modeling and Reasoning Using OWL“. In PERCOMW ’04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, p. 18.

[Want et al., 2002] Want R., Pering T., Borriello G. and Farkas K., “Disappearing hardware”, Pervasive Computing, IEEE , Vol. 1, Issue 1, Jan.-March 2002 p. 36 – 47.

[Want et al., 2005] R. Want, T. Pering, G. Borriello and K. Farkas, “Disappearing hardware”, Pervasive Computing, IEEE ,Vol 1, Issue 1, Jan.-March 2002 pp:36 – 47.

[Weiser, 1991] Weiser M., "The Computer for the Twenty-First Century*", Scientific American*, p. 94-10, September 1991.

[Weißenberg, 2006] N. Weißenberg, R. Gartmann, A. Voisard: “An Ontology-Based Approach to Personalized Situation-Aware Mobile Service Supply“. GeoInformatica 10(1), 2006, p. 55-90.

[Wimmer, 2007] M. R. Wimmer, “Efficient Access Control for Service-oriented IT Infrastructures“, http://mediatum2.ub.tum.de/node?id=622739, 2007.

[Yang et al., 2006] H. Yang, E. Jansen and S. Helal, “A Comparison of Two Programming Models for Pervasive Computing”, Applications and the Internet Workshops, 2006. SAINT Workshops 2006. International Symposium, 23-27 Jan. 2006, pp134 – 137.

[Zhang and Parshar, 2003] Zhang, G. and Parashar, M. 2003. “Dynamic Context-aware Access Control for Grid Applications“. In Proceedings of the 4th international Workshop on Grid Computing 2003. International Conference on Grid Computing. IEEE Computer Society, Washington, DC, p. 101.

[Zimmermann, 2005] Zimmermann A., Lorenz A., and Specht M., “Applications of a Context-Management System”, Modeling and Using Context, LNCS, 2005, Volume 3554/2005, 31-48.