Contrôle d'accès logique au dossier patient informatisé

Romuald Thion, Stéphane Coulondre, André Flory

Contrôle D'Accès Logique Au Dossier Patient Informatisé

Romuald Thion, Stéphane Coulondre, André Flory

LIRIS - Laboratoire d'InfoRmatique en Images et Systèmes d'information
UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1
Université Lumière Lyon 2/Ecole Centrale de Lyon,

Bâtiment Blaise Pascal (501), 20, avenue Albert Einstein,
69621 Villeurbanne Cedex, France
prenom.nom@liris.cnrs.fr

RÉSUMÉ. *Le respect de la confidentialité est une problématique majeure de la sécurité de l'information médicale. Définir les règlements, les implémenter dans un dispositif de contrôle d'accès et les vérifier sont des défis organisationnels, techniques et scientifiques auxquels doivent faire face les établissements de santé. Cette problématique est d'autant plus exacerbée lorsque l'information est partagée et répartie. Cet article propose un cadre logique apportant une réponse aux problèmes d'expression, de stockage, d'interrogation et de validation des politiques d'autorisation complexes pour les systèmes d'information médicale. Nous illustrons notre approche sur une application des recommandations du Groupement pour la Modernisation du Système d'Information Hospitalier et la plateforme d'information médicale de la région Rhône-Alpes.*

ABSTRACT. *Confidentiality is major pitfall of eletronic health information security. Definition, implementation and conformance checking of security policies are organizational, technical and scientific issues wich have to be addressed by healthcare facilities. These difficulties are enhanced in shared distributed eletronic records. This paper proposes a logic-based framework for complex access control policies. It addresses storing, complex security statements expression, querying and validation issues. Our approach is based upon the Rhône-Alpes health platform and a french health-dedicated access control model.*

MOTS-CLÉS : *Politique de sécurité, autorisation, RBAC, GMSIH, dossier patient partagé réparti.*

KEYWORDS: *Security policy, authorization, Role-Based Access Control, electronic medical record, health platform.*

1. Introduction

1.1. *Contexte*

Les technologies de l'information et de la communication actuelles permettent la mise en place de réseaux de soins améliorant la prise en charge des patients et la coopération des acteurs de la santé, à l'échelle régionale, voire nationale. Le déploiement de tels réseaux d'information médicale pose de véritables défis organisationnels et techniques comme l'intéropérabilité des Systèmes d'Information Hospitaliers (SIH) hétérogènes, la conformation aux standards actuels, l'intégration de données de types sensibles et variés ou la coopération d'acteurs aux profils différents. Un de ces défis est la sécurité des systèmes : les obligations éthiques imposent une protection particulière des données médicales (Deswarte *et al.*, 2004) qui ne peut être assurée sans une prise en compte réfléchie et mesurée de la sécurité. La sécurité de l'information est traditionnellement déclinée en quatre propriétés :

– la *confidentialité*, l'information ne doit être accessible qu'aux ayants droits,

– l'*intégrité*, l'information doit être cohérente et valide,

– la *disponibilité*, l'information doit être accessible,

– la *traçabilité*, les accès et les opérations doivent être authentifiés et enregistrés.

L'importance de la prise en compte de la *confidentialité* dans le domaine de l'informatique médicale ne cesse de croître, à cause des enjeux importants qu'elle engage, du caractère sensible de l'information et des dispositions légales en vigueur. Une *politique de sécurité* est (Rihaczek, 1991) :

> Les lois, règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique.

La mise en place d'une *politique de sécurité* est un dispositif organisationnel fondamental pour garantir la sécurité de l'information. La *politique d'autorisation* est une déclinaison de la politique de sécurité qui définit des modèles permettant de gérer les autorisations d'accès au système, elle renforce particulièrement la *confidentialité* et l'*intégrité* de l'information, *a fortiori* sa disponibilité. Informellement, une politique d'autorisation définit "qui a droit à quoi".

1.2. *Motivations*

Les risques liés à l'information médicale sont d'une importance capitale (Deswarte *et al.*, 2004). Dans le cas d'un dossier patient informatisé, le viol des quatre propriétés de sécurité peut par exemple conduire à :

– une prise de décision médicale pouvant porter préjudice au patient,

– une information médicale amoindrie ou inutilisable,

– l'impossibilité d'hexiber un dossier informatisé comme preuve,

– l'utilisation illégitime de données confidentielles,

– une rupture du suivi médical.

Des solutions techniques pour garantir les propriétés de sécurités existent, comme les dispositifs de *protection réseaux* (par exemple : pare-feu, détection d'intrusions), les dispositifs de *cryptage* de l'information ou les architectures d'*authentification* [1] des utilisateurs. Si ces outils sont nécessaires pour renforcer la politique de sécurité globale, ils ne remplacent pas une politique d'autorisation.

Les problèmes auxquels sont confrontées les organisations lors de l'établissement d'une politique d'autorisation sont :

– comment *organiser* les droits, selon quels principes ?

– comment intégrer dans la politique toutes les *spécificités* de leurs activité ?

– comment effectivement *stocker* la politique dans leur système ?

– comment s'assurer que la politique *satisfait* bien aux règlements ?

– comment *administrer* la politique de sécurité ?

– comment *minimiser les coûts* de mise en place et de gestion des autorisations ?

Actuellement, les dispositions existantes concernant la sécurité dans des dossiers patients informatisés ont un caractère binaire : les personnels médicaux disposent de quasiment tous les droits sans adaptation en fonction du profil de l'acteur, les autres n'ont aucun accès. Nous pensons qu'il s'agit d'une conséquence au fait que les modèles de contrôle d'accès "traditionnels" sont inadéquats, bien trop rigides pour de telles applications.

Quant à proposer des modèles de contrôle d'accès de plus en plus flexibles, si l'idée est conceptuellement séduisante, il faut éviter de tomber dans l'écueil inverse où les mécanismes deviennent trop complexes pour être administrables efficacement. Pour éviter cette situation, l'intégration du contrôle d'accès doit se faire de la manière la plus homogène possible, accompagnées d'outils de vérification et de validation.

1.3. *Contribution*

Les Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) sont les pierres angulaires des SIH : ils sont utilisés par tout type d'application et sont la technologie incontournable de stockage, gestion et interrogation des données. Basé sur cette constatation, cet article propose d'utiliser un cadre théorique adéquat issu des bases de données pour répondre aux problèmes actuels d'*expression*, de *stockage*, d'*interrogation* et de *validation* des politiques d'autorisation complexes pour les systèmes d'information médicale.

1. l'authentification se définirait informellement par "prouver que l'on est bien celui que l'on prétend être". Le contrôle d'accès suppose l'existence de mécanismes d'authentification préalables.

Cette approche originale nous permet de proposer un cadre logique pour les politiques d'autorisations qui (Thion *et al.*, 2006b) :

– est formellement bien défini,

– est capable d'exprimer des politiques de données complexes,

– s'intègre parfaitement au sein du modèle relationnel,

– peut être ré-utilisé pour des usages plus traditionnels de base de données,

– intègre harmonieusement les notions d'intégrité et de confidentialité,

Suite à cette introduction, la section suivante présente la politique d'autorisation du Groupement pour la Modernisation du Système d'Information Hospitalier sur lequel nous appuierons notre exposé. Les cadres applicatif et théoriques sont présentés en section 3, préambule à la description détaillée de notre approche proposée en section 4. Les sections 5 et 6 détaillent la méthodologie de mise en place du modèle que nous décrivons ainsi que les aspects d'administration de la sécurité. Enfin, la section 7 conclue cet article et ouvre des perspectives de recherche.

2. Contrôle d'Accès à l'Information

2.1. *Contrôle d'Accès Basé sur les Rôles*

Le *contrôle d'accès*, ou *autorisation*, est le fait de déterminer si un *sujet* (ex : un processus, un utilisateur) peut effectuer une *opération* (ex : supprimer, modifier, créer) demandée sur une *ressource* (ex : une table, un fichier, un document). Une permission est un droit d'opération sur une ressource. Une *politique de contrôle d'accès*, ou *Politique d'Autorisation* (PA), spécifie les permissions des sujets dans un système. Ainsi, lorsqu'un utilisateur désire effectuer une opération, le système vérifie si cet accès est autorisé par ses droits.

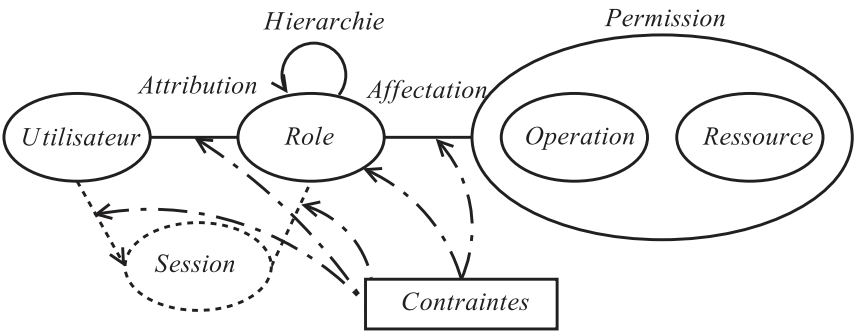

Figure 1. *Le modèle RBAC, pas de formalisme spécifique (Ferraiolo* et al.*, 2003)*

Les modèles de politiques basés sur les rôles, dits “modèle(s) RBAC” pour Role-Based Access Control, ont été introduits pour combler les déficits des modèles *mandataires* (basés sur une hiérarchie de labels comme “confidentiel”, “secret”, “très secret”) et *discrétionnaires* (l'utilisateur propriétaire d'une ressource définit lui-même ses autorisations) qui s'avèrent trop rigides, trop complexes à administrer ou vulnérables (Harrison *et al.*, 1976).

Les modèles basés sur les rôles sont fondés sur la constatation que la majeure partie des décisions de contrôle d'accès sont déterminées par l'autorité hiérarchique ou la fonction du sujet dans son organisation (Sandhu *et al.*, 1996) : cela forme le concept central de *rôle*. L'introduction de ce concept dans les PA comme “intermédiaire” entre les sujets et les permissions simplifie l'administration et en réduit les coûts. Lorsqu'un utilisateur a besoin d'effectuer une tâche, il suffit de lui *attribuer le ou les rôles correspondants*. Ainsi, quand un utilisateur change de fonction dans son organisation, il suffit de changer les rôles qui lui sont attribués sans introduire directement de notion de révocation de permissions.

La figure 1 traduite de (Ferraiolo *et al.*, 2003) illustre l'organisation du modèle RBAC. Les concepts présents dans ce schéma sont :

– les *utilisateurs* : un ensemble de personnes physiques qui vont accéder au système et auxquelles vont être attribués des rôles. Par exemple $\{Albert, Brice, Charles\}$,

– les *rôles* : un ensemble de rôles par lesquels les utilisateurs vont disposer de permissions. Par exemple $\{Infirmier, Médecin\ généraliste, Chirurgien, Spécialiste\}$,

– la *hiérarchie* des rôles : elle permet de réduire le nombre d'affectations de permissions en introduisant une relation d'héritage entre les rôles, cette notion est détaillée en section 4.3. Par exemple un $Chirurgien$ *est-un* $Spécialiste$,

– les *attributions* de rôles aux utilisateurs : les rôles du système que les utilisateurs sont autorisés à endosser, par exemple les rôles $Chirurgien$ et $Responsable\ de\ service$ sont attribués à $Albert$.

– les *opérations* : un ensemble d'actions qui peuvent être effectuées sur les ressources du systèmes. Par exemple $\{Lire, Ajouter, Modifier\}$,

– les *ressources* : un ensemble de documents sur lesquels les utilisateurs vont effectuer des opérations. Par exemple $\{Image\ médicale, Compte\text{-}rendu\ opératoire, Analyses\}$,

– les *permissions* : un sous-ensemble d'opérations sur les ressources qui vont être affectées aux rôles. Ce sont les droits accordés dans le système. Par exemple $Ajouter$ des $Compte\text{-}rendu\ opératoire$ est une permission.

– les *affectation* de permissions aux rôles : les privilèges conférés aux rôles selon l'autorité hiérarchique et la fonction qu'ils représentent, par exemple on confère au rôle $Chirurgien$ le droit d'$Ajouter$ des $Compte\text{-}rendu\ opératoire$ à un dossier patient,

– les *sessions* : un ensemble de sujets qui représentent les utilisateurs actifs dans le système. Une session est attribuée à *exactement un seul* utilisateur et indique quels sont les rôles qu'il endosse. Il s'agit de l'élément dynamique du modèle RBAC. Par exemple, dans la session s, $Albert$ endosse le rôle de $Chirurgien$,

– les *contraintes* : un ensemble de restrictions concernant les concepts du modèle RBAC, comme par exemple "aucun utilisateur ne peut se voir attribuer les rôles $Médecin\ généraliste$ et $Spécialiste$". Cette notion sera détaillée en section 5.

2.2. *Un Modèle de Politique d'Autorisation pour l'Informatique Médicale*

Le Groupement pour la Modernisation du Système d'Information Hospitalier (GMSIH) a, dans le but de faciliter la mise en place de politiques de sécurité, proposé une Politique d'Autorisation (GMSIH, 2003). Un modèle principal est proposé, décliné en trois versions : *simplifié*, *intermédiaire* et *complet*. Il s'agit d'une aide pragmatique à l'élaboration de politique de sécurité pour la santé. Les modèles proposés par le GMSIH sont des extensions du modèle RBAC adaptés au contraintes spécifiques des milieux hospitaliers :

– l'ajout de la notion de *Structure* et de la hiérarchie associée qui modélise les établissements et leurs services,

– la classe d'association *Affectation* qui est une association ternaire entre *Utilisateurs*, *Rôles* et *Affectations*, ainsi un utilisateur peut disposer de plusieurs rôles dans des structures différente. Cela permet de représenter le fait qu'une personne soit *Surveillant* dans un service, et *Infirmier* dans un autre,

– les liens entre *Dossier*, *Patient* et *Professionnel de Santé*, en effet, c'est le propriétaire du dossier médical qui définit qui a accès aux données le concernant,

– le contenu sémantique du dossier médical (urgence, détail, synthèse) : tout le contenu du dossier ne doit pas être accessible à tous les PS, en effet certaines informations (exemple : anatomie, pathologie, virologie) sont particulièrement confidentielles. De même les attributs urgence, détail et synthèse sont des méta-informations qui précisent la nature du document,

La sécurité, et plus particulièrement les modèles de politique d'autorisation, ont déjà été étudiés (Deswarte *et al.*, 2004). Un modèle, Organization-Based Access Control - OrBAC -, a été proposé pour répondre à la problématique de l'instanciation d'une même politique dans plusieurs établissements (Kalam, 2003). Nous avons choisi d'appuyer notre approche sur le modèle du GMSIH car c'est à notre avis, la proposition la plus concrète et pragmatique.

Par soucis de clarté de l'exposé, nous avons choisi d'utiliser le modèle intermédiaire (figure 2) du GMSIH et de ne pas prendre en compte les sessions. Nous nommerons par la suite PA-GMSIH le modèle intermédiaire du GMSIH que nous avons adapté pour mieux correspondre à la plateforme d'information médicale de la région Rhône-Alpes.

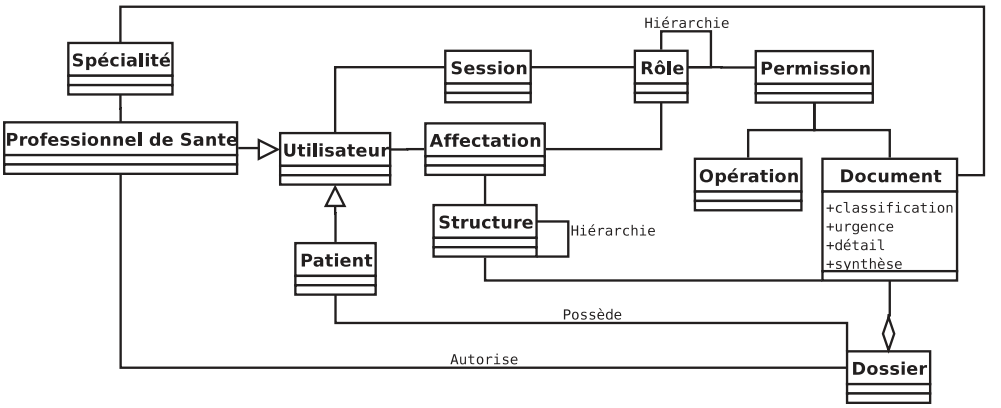

Figure 2. *Le modèle UML de la politique d'autorisation intermédiaire du GMSIH*

3. Cadres Applicatif et Théorique

Pour illustrer notre approche nous nous sommes appuyés sur la plateforme de santé de la région Rhône-Alpes. Cette section décrit cette réalisation et présente le cadre théorique sur lequel nous fondons notre proposition.

3.1. *Plateforme de Santé de la Région Rhône-Alpes*

La région Rhône-Alpes a mis en place une plateforme d'information médicale (SIS-RA), connectant la plupart des établissement de santé de la région. Cette plateforme est composée de trois projets principaux : le Serveur Télématique d'Identité Communautaire (STIC), la Plateforme d'Echange entre les Professionnels de Santé (PEPS) et le Dossier Patient Partagé et Réparti (DPPR).

Le Dossier Patient Partagé et Réparti (DPPR) est un projet dont l'objectif est de donner l'accès à des professionnels de la santé aux informations médicales relatives à un patient et localisées dans le SIH de la structure. Chaque SIH connecté envoie à la plate-forme centrale une référence des informations médicales qu'il souhaite partager. Le DPPR se base sur l'identifiant régional du patient. Le DPPR est un dossier (Durand *et al.*, 2007) : *réparti* : ce n'est pas une centralisation des informations, *maximum* : toutes les informations concernant la personne malade sont référencées : Ville, Hôpital, EFS, SECU, . . . , *multipathologique* essentiel à une prise en charge coordonnée du patient et destiné à *faire communiquer* 250 hôpitaux / cliniques, 12000 médecins de ville pour 6 millions d'habitants, et des milliers de soignants.

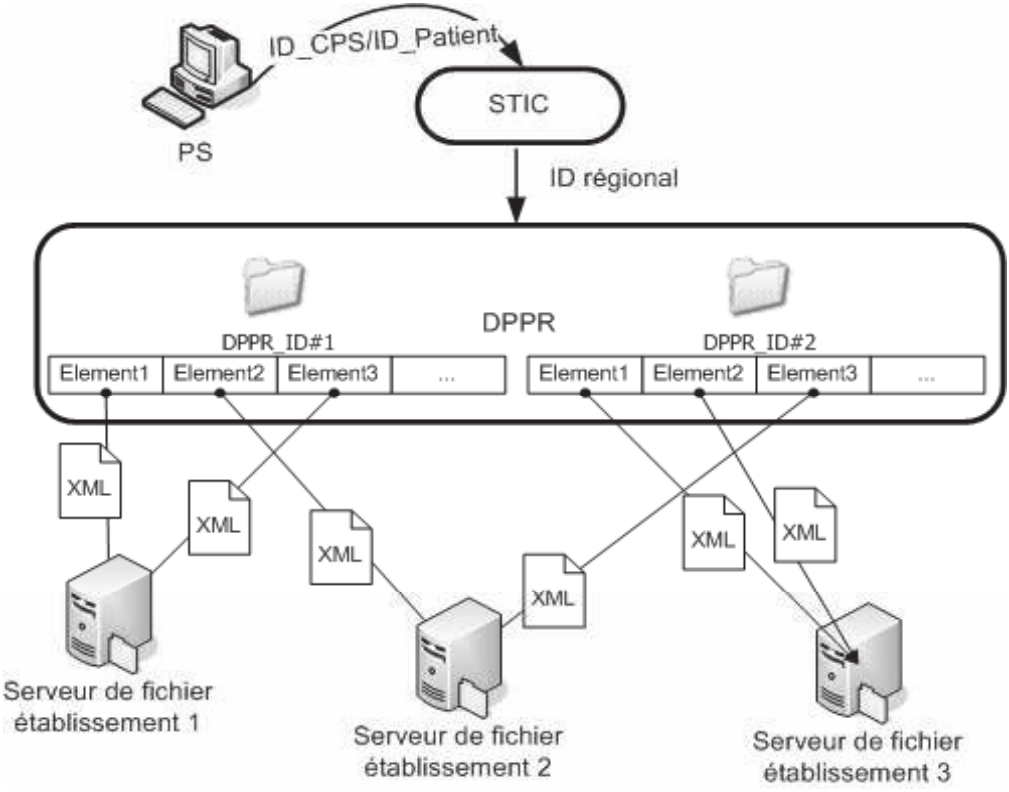

Figure 3. *Schéma structurel de base du DPPR, pas de formalisme spécifique*

Les besoins fonctionnels du DPPR sont :

– la souplesse dans le format de l'information accepté par le DPPR : allant de la non structuration (txt, pdf, doc, jpg) à une complète structuration relationnelle,

– la mise en forme faite par le DPPR pour permettre une adaptation de cette mise en forme selon le média connecté,

– les futures utilisations du DPPR comme point d'accès aux gisements pour des bases de données thématiques, épidémiologiques, . . . ,

– la possibilité d'utilisation par les établissements de santé quel que soit leur niveau d'informatisation,

L'architecture retenue pour répondre à ces besoins est d'organiser le dossier patient comme une liste de pointeurs (ex : Uniform Resource Locator - URL) sur des documents de santé stockés dans les établissements où ils ont été produits, comme l'illustre la figure 3. Les besoins fonctionnels du projet de la région Rhône-Alpes illustrent l'importance du besoin de flexibilité de l'informatique médicale. Nous pensons que ce critère de souplesse est également incontournable pour le contrôle d'accès.

(1) $\forall ID, Nom_1, Prenom_1, Nom_2, Prenom_2$
$employe(ID, Nom_1, Prenom_1) \wedge employe(ID, Nom_2, Prenom_2)$
$\rightarrow Nom_1 = Nom_2 \wedge Prenom_1 = Prenom_2$

(2) $\forall ID, IDEquipe, Fonction$
$responsable(ID, IDEquipe, Fonction)$
$\rightarrow \exists Nom, Prenom\ employe(ID, Nom, Prenom)$

Tableau 1. *Exemple de dépendances écrites en logique*

3.2. *Un Cadre Théorique Issu des Bases de Données*

Les besoins de vérifiabilité et d'intégrité des politiques d'autorisation nécessitent un cadre théorique capable de capturer la complexité du modèle PA-GMSIH. Une formalisation des concepts de politique permet une meilleure compréhensions des mécanismes de sécurité ainsi qu'une intégration homogène avec des systèmes existants.

Les contraintes d'intégrité (ou dépendances de données) est un des piliers de la théorie des Base de Données (BdD) relationnelles, ce sont des propriétés devant être satisfaites *par toutes les instances d'un schéma de BdD*. Ces dépendances sont regroupées en classes, c'est à dire en sous-ensemble restreints de la logique du premier ordre (sans négation, sans fonctions, sans disjonctions) (Abiteboul *et al.*, 1995). Les classes de dépendances les plus étudiées et utilisées en pratique dans les systèmes de gestion des base de données sont :

– les dépendances fonctionnelles qui *généralisent la notion de clé primaire*, elles imposent l'égalité de certaines données en fonction d'autres. Par exemple la dépendance (1) du tableau 1 indique que l'attribut ID de la relation $employe$ définit fonctionnellement les attributs Nom et $Prenom$: deux tuples "employés" ayant le même identifiant doivent avoir le même nom et le même prénom. Cette dépendance exprime le fait que l'attribut ID est un identifiant unique à partir duquel ont peut déduire toutes les informations concernant l'employé,

– les dépendances d'inclusion *qui généralisent la notion de clé étrangère*, elles imposent la présence de tuples en fonction d'autres, par exemple la dépendance (2) du tableau 1 indique que l'attribut ID de la relation $responsable$ est inclus dans l'attribut ID de la relation $employe$: elle signifie qu'un "responsable" est un "employé".

La littérature propose une grande variété de dépendances, répondant au besoin d'expressivité nécessaire pour modéliser des situations complexes. La théorie des dépendances a trouvé par exemple des applications avec la conception de schémas, l'optimisation de requêtes ou plus récemment la correspondance de schémas (Fagin, 2006). Une des classes la plus générique, englobant la majorité des classes existantes, est celle des *dépendances génératrices de tuples contraintes* (Maher *et al.*, 1996). Son expressivité est suffisamment grande pour permettre de représenter tout la palette de concepts des modèles RBAC (Thion *et al.*, 2006b) et du modèle PA-GMSIH. Notre hypothèse de travail pourrait-être exprimée de façon informelle

par “les autorisations d’un système sont des données structurées selon un modèle de politique, utilisons des outils de bases de données pour les exprimer et les vérifier”.

L’aspect théorique relatifs aux dépendances le plus étudié est certainement celui de l’*implication logique* : étant donnée un ensemble de dépendances F et une dépendance g, g est-elle satisfaite pour toute instance de schéma qui satisfait F ? Ce problème de l’implication s’avère être un problème incontournable dans tous les cas d’utilisation des dépendances. Le problème de l’implication est dit *décidable* pour une classe de dépendance s’il existe un algorithme donnant un réponse exacte à ce problème en un *temps fini*. Nous utilisons les algorithmes de preuve de l’implication pour répondre à des problèmes d’administration concernant “toute instance de politique”.

Ce cadre nous permet de proposer une architecture logique de contrôle d’accès pour les dossiers patient, où la politique est administrée de façon *déclarative*, sans ce soucis des mécanismes sous-jacents.

4. Contrôle d’Accès Logique au Dossier Patient

Afin de garantir un contrôle d’accès fiable et de faciliter la conception des politiques, nous pensons qu’il faut se rapprocher le plus possible des paradigmes existants dans le domaine des BdDs et d’utiliser les outils développés en la matière. Cette approche nous permet de concilier la gestion des données médicale ainsi que celle des autorisations dans un même cadre.

4.1. *Architecture du Contrôle d’Accès*

Basés sur le cadre des bases de données, nous avons identifié une structuration en quatre modules pour la réalisation de politiques d’autorisation (cf. figure 4). Ces modules font partie de la “base données système” (appellée aussi méta-base) du SGBD :

– le module $F_{politique}$ traduit les *principes* de la politique d’autorisation de l’organisation : c’est le coeur du modèle de contrôle d’accès qui définit comment dériver les autorisations à partir de la politique.

– le module $F_{intégrité}$ est l’ensemble des *contraintes d’intégrité* qui permettent d’assurer qu’une politique est intègre, cohérente. Le but est d’interdire des opérations d’administration qui conduiraient à un état inconsistant de la politique,

– le module $F_{données}$ représente *l’instance* de la politique d’autorisation, c’est à dire les données que vont manipuler les administrateurs de sécurité et qui seront stockées dans une BdD relationnelle,

– le module $F_{contraintes}$ est l’ensemble des *contraintes métier* qui permettent de limiter les permissions accordées aux utilisateurs.

Les ensembles $F_{politique}$ et $F_{intégrité}$ définissent le modèle d’autorisation et les règles qui permettent de s’assurer de l’intégrité de *toute* politique. $F_{politique}$ et

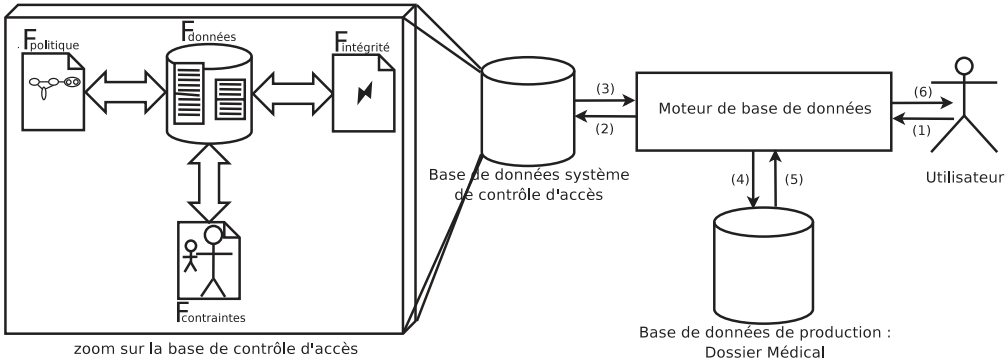

Figure 4. *Architecture du Contrôle d'accès, pas de formalisme spécifique*

$F_{intégrité}$ sont fixes une fois le choix du modèle de CA fait (par exemple entre les trois déclinaisons simplifiée, intermédiaire et complète du modèle du GMSIH). Les tâches des administrateurs sont, dans l'architecture proposée, de gérer la politique $F_{données}$ et les contraintes métiers $F_{contraintes}$.

Cette structuration permet de faciliter l'administration et l'ingénierie des rôles en définissant des étapes dans la conception de politiques. L'idée de séparer le contenu de la politique des principes du modèle a été proposée dans (Halpern *et al.*, 2003), pour renforcer la séparation entre le contenu de la politique et des règles qui la gouverne.

Notre proposition rapproche la gestion relationnelle de l'information médicale de la gestion relationnelle des politiques d'autorisation. Notre architecture suppose l'existence d'une base de *données de production* dans laquelle sont stockées les dossiers DPPR (cf. figure 4). Cette base contient la liste des patients, des professionnel de santé, leurs relations et les pointeurs du dossier vers des documents. La base de données que nous décrivons en section suivante stocke quant à elle la politique d'autorisation, elle est stockée dans la base de *données système* du SGBD.

Lorsqu'un utilisateur désire accéder à un dossier patient, un mécanisme de décision va décider si cet accès est autorisé ou non. Cette procédure est décomposable en 6 étapes (cf. figure 4) :

1) envoi de la requête de l'utilisateur au moteur de la base de données,

2) interrogation de la politique d'autorisation,

3) réponse de la politique,

4) le moteur accède à la base de production si l'accès est autorisé,

5) lecture du contenu du dossier patient,

6) retour de la requête ou exception en cas d'accès non-autorisé.

4.2. *Base de Données*

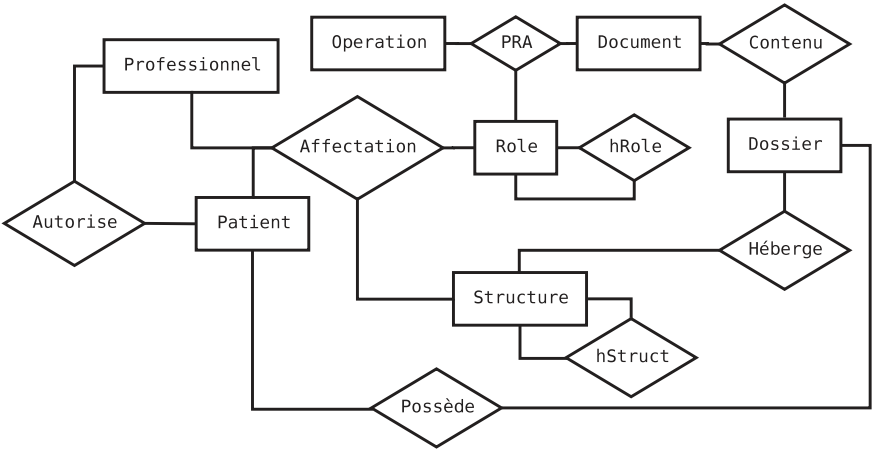

Figure 5. *Schéma entité-relation de la base de données de la politique*

La notion de BdD d'une politique a été étudiée dans les travaux attenant à RBAC : elle représente l'ensemble des faits connus dans la politique, son contenu. C'est cette BdD qui va être saisie et modifiée par les opérations d'administration concernant :

– l'ensemble des utilisateurs : PS et patients,

– les rôles et leurs relations hiérarchiques,

– les structures et leurs relations hiérarchiques,

– les permissions accordées aux rôles,

– les affectations de rôles aux utilisateurs dans les structures,

– les documents qui composent les dossiers patients

Nous n'avons pas inclus d'attributs dans le schéma Entité-Relation de la figure 5 pour des raisons de lisibilité mais les méta-informations concernant les documents et les dossiers (spécialité, classification, urgence, détail et synthèse pour les documents, PS autorisés et propriétaire pour les dossiers) sont stockés dans cette base. C'est à partir de la base de données, $F_{données}$, et du modèle d'autorisation, $F_{politique}$, que seront prises les décisions d'autorisations. Il s'agit là véritablement d'un schéma auquel seront ajoutées des dépendances qui assureront l'intégrité de la politique, $F_{intégrité}$, et les contraintes métiers $F_{contraintes}$.

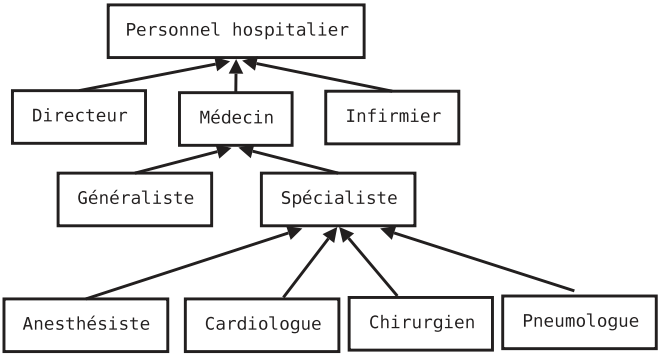

Figure 6. *Exemple de hiérarchie de rôles, flèche du spécialisé au générique*

4.3. *Modélisation des Principes de la Politique*

A partir des travaux sur la formalisation en programmation logique des modèles RBAC (Barker *et al.*, 2003), nous proposons un ensemble de dépendances capturant les principes du CA du GMSIH. Nous avons étendu les travaux de Barker et Stuckey avec la notion de contraintes d'intégrité des politiques, les extensions spécifiques aux SIH et la structuration en quatre parties.

La *hiérarchie* des rôles est une des innovations proposées dans le modèle RBAC. Elle permet de réduire le nombre de regroupement de permissions et d'affectation de rôles aux utilisateurs en introduisant une notion d'héritage *est-un* entre rôles : un rôle hérite des permissions accordées à ses parents. On peut de façon équivalente dire que un a *est-un* b quand le rôle a hérite du rôle b. La hiérarchisation permet de réduire le nombre de rôles en présence dans la politique, simplifiant de ce fait son administration. Cette relation d'héritage est également utilisée pour la hiérarchisation des structures et pourrait être utilisée pour raffiner la notion de spécialité de médecine.

Notre approche consiste à modéliser avec des dépendances le mécanisme de décision d'accès. Supposons qu'un utilisateur demande d'effectuer une opération sur une ressource (par exemple un médecin qui désire consulter un résultat de diagnostic) à partir du contenu de la politique et des règles qui régissent le modèle, un algorithme d'inférence va décider si cet accès est autorisé (dans l'exemple le médecin verrait s'afficher le diagnostic demandé) ou non (le médecin recevrait un message d'erreur).

La *règle principale* de dérivation d'une autorisation à partir de la politique est :

> Une opération sur un document est autorisée si les rôles endossés par l'utilisateur dans la structure lui en donnent le droit (grâce à l'héritage de rôles), si cet utilisateur fait partie de la liste des personnes autorisées et que le document est hébergé dans la structure en question (grâce à la hiérarchie de structures).

Données de la politiques contenues dans $F_{données}$:

```
dRole(medecin, personnelHospitalier)
dRole(infirmier, personnelHospitalier)
dRole(specialiste,medecin)
dRole(generaliste,medecin)
dRole(chirurgien,specialiste)
dRole(pneumologue,specialiste)
dRole(anesthesiste,specialiste)
dRole(cardiologue,specialiste)
```

Les règles régissant le modèle $F_{politique}$ permettent de déduire :

```
hRole(medecin, personnelHospitalier)
hRole(specialiste,medecin)
hRole(generaliste,medecin)
hRole(chirurgien,specialiste)
...
hRole(specialiste,personnelHospitalier)
hRole(generaliste,personnelHospitalier)
hRole(chirurgien,medecin)
dRole(chirurgien,personnelHospitalier)
...
```

Tableau 2. *Exemple de déduction à l'aide de dépendances*

La règle principale et celles permettant de calculer la couverture transitive de la relation d'héritage, peuvent être exprimées à l'aide de dépendances de la classe appelée Total Tuple-Generating Dependencies, une classe dite décidable (Beeri *et al.*, 1984) (cf. section 3.2). Prenons par exemple la manipulation de la hiérarchie de rôles présentée en figure 6. Soit la relation $dRole(RoleDescendant, RoleParent)$ qui modélise les relations d'héritage directes entre rôles (par exemple le rôle "cardiologue" hérite directement du rôle "spécialiste", ou autrement exprimé un "cardiologue" *est-un* "spécialiste"). Il faut calculer $hRole(RoleDescendant, RoleParent)$, *la fermeture transitive* de cette relation, c'est à dire la relation transitive minimale contenant la relation $dRole$ (cf. figure 5), est calculée à l'aide des deux dépendances suivantes :

$$dRole(RoleDescendant, Role) \wedge hRole(Role, RoleParent) \rightarrow hRole(RoleDescendant, RoleParent) \quad (1)$$

$$dRole(RoleDescendant, Role) \rightarrow hRole(RoleDescendant, Role) \quad (2)$$

Ce calcul est illustré par la tableau 4.3 qui présente l'exemple d'utilisation de dépendances pour calculer la fermeture transitive de la hiérarchie de rôles de la figure 6.

Basée sur ces principes, voici la traduction en dépendance de la règle principale de dérivation d'autorisation :

$$\begin{aligned} Affectation(Sujet, Role_a, Structure_a) \wedge hStruct(Structure_a, Structure) \wedge \\ PRA(Operation, Document, Role) \wedge hRole(Role_a, Role) \wedge \\ Contenu(Document, Dossier) \wedge Heberge(Document, Structure) \wedge \\ Autorise(Patient, Sujet) \wedge Possede(Patient, Dossier) \wedge \\ \rightarrow Autorise(Sujet, Operation, Document) \end{aligned}$$

5. Ingénierie des Rôles et des Contraintes Métier

La mise en place d'une politique d'autorisation intégrant la notion de rôles est un projet important. Comme une politique à base de rôles est "calquée" sur la structure de l'organisation, cela nécessite de formaliser cette structure. La définition des rôles en présence dans les organisations ainsi que le regroupement des permissions est un domaine de recherche à part entière, à la croisée du management et de la sécurité : l'ingénierie des rôles (Coyne, 1996).

La notion de *contrainte métier* a été introduite pour représenter les spécificités des systèmes et limiter les privilèges des usagers du système, c'est un aspect des plus détaillés dans la littérature RBAC voire même une des principales motivations (Ahn *et al.*, 2000). Les contraintes métiers sont un ensemble de restrictions devant être garanties dans la politique et dépendantes des organisations, à la différence des *contraintes d'intégrités* qui doivent être satisfaites par instance de politique.

Les contraintes métiers ont été proposées pour garantir *la séparation des responsabilités* des acteurs et éviter qu'un utilisateur dispose de trop de permissions. Les auteurs de (Ferraiolo *et al.*, 2003) abordent la notion de contraintes entre utilisateurs et rôles et entre rôles, aboutissant à un ensemble de propriétés devant être respectées par la politique pour que les contraintes soient cohérentes. Les contraintes métiers permettent d'exprimer dans la PA les règlements établis par les organisations.

Nous allons illustrer notre propos par l'étude de la notion, *d'exclusion mutuelle* statique entre deux rôles, car c'est une contrainte métier à la sémantique répandue. Supposons qu'il existe une règle qui spécifie que "les rôles spécialistes et généralistes ne peuvent pas être assumés par un même médecin", il s'agit d'un cas d'exclusion mutuelle entre les rôles "spécialiste" et "généraliste" (cf. figure 6).

Pour exprimer cette contrainte métier, nous introduisons une relation supplémentaire $ssd(Role_1, Role_2)$ qui modélise le fait que les rôles $Role_1$ et $Role_2$ sont en exclusion mutuelle. Nous utiliserons les dépendances pour modéliser les propriétés l'exclusion mutuelle doit respecter. Comme pour l'expression des principes de fonctionnement du modèle de contrôle d'accès, les dépendances utilisées font partie d'une classe décidable. Voici un extrait des propriétés que l'exclusion mutuelle doit respecter (le prédicat $affectation(U, R, S)$ représente l'attribution du rôle R à un utilisateur U dans la structure S, $\perp$ s'interprète comme un état inconsistant) :

– un utilisateur *ne peut pas être affecté* à deux rôles en exclusion mutuelle :
$affectation(U, R_1, S), affectation(U, R_2, S), ssd(R_1, R_2) \rightarrow \perp$,

– l'exclusion mutuelle entre rôles est *symétrique* :
$ssd(R_1, R_2) \rightarrow ssd(R_2, R_1)$,

– un rôle ne peut pas être en exclusion mutuelle *avec lui même* :
$ssd(R_1, R_1) \rightarrow \perp$,

– deux rôles en exclusion mutuelle *ne peuvent pas hériter* l'un de l'autre :
$hRole(R_1, R_2), ssd(R_1, R_2) \rightarrow \perp$,

– il ne doit pas y avoir de rôle qui hérite de deux rôles en exclusion mutuelle : $hRole(R, R_1), hRole(R, R_2), ssd(R_1, R_2) \rightarrow \perp$,

– l'exclusion mutuelle est *propagée* par la relation d'héritage : $hRole(R, R_1), ssd(R_1, R_2) \rightarrow ssd(R, R_2)$.

Notre cadre nous permet d'exprimer toute une variété de contraintes qui généralisent l'exclusion mutuelle. Par exemple, on peut grâce aux dépendances exprimer une contrainte imposant que "un rôle r soit en exclusion mutuelle avec tous les autres" ($role(R), R \neq r \rightarrow ssd(R, r)$). Ceci permet d'interdire l'augmentation des privilèges des utilisateurs invités : un utilisateur mal intentionné qui à partir d'un compte *invité* parviendrait à obtenir un nouveau rôle rendrait la politique inconsistante et serait remarqué.

Le cadre permet également d'exprimer des contraintes complexes comme "un professionnel de santé ne peut pas s'autoriser lui-même" ($autorise(U, U) \rightarrow \perp$), ou "s'il existe un utilisateur chirurgien dans l'organisation, alors il existe *un autre* utilisateur anesthésiste dans la même structure" ($affectation(U, chirurgien, S) \rightarrow \exists U'\ affectation(U', anesthesiste, S), U' \neq U$) . Ce dernier type de contraintes n'a pas, à notre connaissance, été jusqu'alors abordé sous cette forme dans la littérature sur le CA.

6. Administration

6.1. *Interrogation des Politiques*

Notre proposition se place dans un cadre de base de données munies de dépendances qui permettent d'imposer l'existence de données en fonction d'autres, comme par exemple la déduction de la fermeture transitive d'une relation d'héritage. De plus, ces dépendances permettent d'exprimer des *contraintes d'intégrité* permettent de garantir la cohérence des politiques.

Les langages de requêtes déclaratifs comme SQL (Structured Query Language) permettent d'exprimer toutes les requêtes exprimables sur des données relationnelles *sans connaître les mécanismes* entrant en jeu pour leur évaluation. Il s'agit là d'une des motivations pour l'utilisation du cadre théorique que nous avons choisi pour notre approche : nul besoin de réaliser des modules *ad-hoc* pour pouvoir interroger les politiques. Pour notre application, les langages déclaratifs permettent d'interroger la politique. Pour les relations déduites (comme $hRole$ déduite à partir de $dRole$), deux approches sont envisageables :

– *calculer a priori* l'ensemble des faits qui peuvent être déduits, les stocker dans un système de gestion de base de donnée puis utiliser les fonctionnalité d'interrogation du système. Cette hypothèse est envisageable car les dépendances utilisées pour exprimer les règles qui régissent le fonctionnement de la PA-GMSIH font partie d'une classe de dépendances décidable.

– utiliser un moteur de *base de données déductives* (Barker, 2002), par exemple XSB [2] qui permet d'interroger directement à la fois des relations en extension ou en intention, sans calculer *a priori* l'ensemble des faits déductibles.

6.2. *Intégrité des Politiques*

Gavrila et Barkley ont étudié la spécification formelle des relations entre rôles et entre utilisateurs et rôles (Gavrila *et al.*, 1998). Ils ont défini un ensemble de propriétés qui doivent être garanties pour que la politique soit *consistante*. Nous avons repris ces propriétés et les avons exprimées à l'aide de dépendances. Nous traitons donc dans cette section des "contraintes d'intégrité sur la politique". Ces dépendances sont des "contraintes de déni" (denial constraints) que toute instance de la politique doit satisfaire.

– aucun rôle n'hérite de lui même, pas de cycle dans la hiérarchie des rôles :
$hRole(R, R) \rightarrow \bot$,

– aucun utilisateur ne doit être affecté à des rôles héritant entre eux :
$affectation(U, R_1, S), affectation(U, R_2, S), hRole(R_1, R_2) \rightarrow \bot$.

La politique est consistante *si et seulement si* le contenu de la politique $F_{données}$ et des dépendances $F_{politique}$ régissant son fonctionnement satisfont l'ensemble de dépendances $F_{intégrité}$. Il s'agit d'une vérification automatisable qui est effectuée avant chaque opération d'administration sur la politique. Par exemple, lors de nouvelles affectations on vérifie qu'aucun utilisateur n'est affecté à deux rôles héritant entre eux.

D'autres contraintes d'intégrité de la politique peuvent être mises en place. En effet, avec la multiplicité des intervenants impliqués dans les grandes organisations, plusieurs administrateurs peuvent se retrouver à collaborer sur une même politique. Afin d'éviter les dysfonctionnements, il peut être intéressant de mettre en place des contraintes imposant un minimum d'affectations, qualifiées de *contraintes de prérequis*. On pourrait dire informellement "qu'elles imposent *l'existence* de certains faits sur des termes inconnus à l'avance".

Nous donnons ici des exemples contraintes d'intégrité que l'ensemble des faits $F_{données}$ doit satisfaire, le prédicat $pra(A, D, R)$ représente l'affectation du droit d'accès A sur le document D au rôle R, $possede(P, D)$ indique que le patient P possède le dossier D, $autorise(P, M)$ indique que la professionnel M est autorisé par le patient P à accéder aux données le concernant :

– chaque utilisateur est associé à *au moins* un rôle :
$user(U) \rightarrow \exists R, S\ affectation(U, R, S)$,

– chaque rôle est associé à *au moins* un utilisateur :
$role(R) \rightarrow \exists U, S\ affectation(U, R, S)$,

2. `http://xsb.sourceforge.net/`

– chaque rôle dispose d'*au moins* une permission :
$role(R) \rightarrow \exists A, D\ pra(A, D, R)$,

– chaque dossier *appartient* à un patient :
$dossier(D) \rightarrow \exists P\ possede(P, D)$,

– *au moins* un professionnel de santé doit pouvoir accéder à chaque dossier :
$dossier(D), possede(P, D) \rightarrow \exists M\ autorise(M, D)$.

6.3. *Méthodologie*

Les principales étapes de l'établissement d'une politique d'autorisation sont (Ferraiolo *et al.*, 2003) :

1) le choix du modèle de contrôle d'accès,

2) la définition de la politique (ex : utilisateurs, rôles, permissions),

3) la réorganisation des rôles et des structures en hiérarchies,

4) la définition des contraintes métier.

Ces différentes étapes font intervenir les différents ensemble de dépendances $F_{politique}$, $F_{intégrité}$ et $F_{contraintes}$ ainsi que les données $F_{données}$. Ainsi, lors de la conception de la politique, différents besoins d'administration doivent être pris en compte :

1) lors de la mise en place de la PA-GMSIH sur le DPPR, on peut raisonner sur $F_{politique}$ pour comprendre le fonctionnement du modèle,

2) pendant la définition de la politique $F_{données}$, on a besoin d'interroger la politique, pour connaître les droits des usagers,

3) ensuite, on définit $F_{intégrité}$ pour mettre en place les contraintes de pré-requis afin de s'assurer que chacun dispose d'un minimum de permissions,

4) pendant la réorganisation des rôles en hiérarchie, on va avoir besoin d'interroger la politique pour trouver les permissions communes à plusieurs rôles,

5) pendant la définition des contraintes métiers on souhaite trouver les faits qui violent les dépendances de $F_{contraintes}$,

6) une fois les contraintes métiers définies, on souhaite minimiser la taille de $F_{contraintes}$ afin de simplifier la gestion de la politique,

Chacune de ces problématiques peut être résolue en utilisant les procédures de preuves de l'implication ou la satisfaction logique classique des dépendances. Le tableau 3 synthétise les besoins d'administrations en classe de problème et apporte une réponse basée sur le cadre théorique des dépendances de données.

Besoins d'administration	*Réductions aux dépendances*
Vérifier qu'une propriété est vraie dans *toute* instance du modèle PA-GMSIH	Modéliser la propriété sous forme de dépendance f et vérifier son implication
Eliminer les contraintes métier redondantes	si $F_{politique} \cup F_{intégrité}$ implique une contrainte, alors la supprimer car elle est redondante
Interroger la politique	Modéliser la requête sous forme logique, et utiliser le moteur de la BdD
Vérifier que la politique est consistante	Vérifier que $F_{contraintes} \cup F_{intégrité}$ sont bien toutes satisfaites par $F_{politique} \cup F_{données}$

Tableau 3. *Réduction en dépendances des besoins d'administration*

6.4. *Exemple de Vérification Automatisée*

Prenons comme exemple l'existence d'un rôle $administrateur$, qui hérite de tous les autres. Si une contrainte d'exclusion mutuelle est definie dans la politique, le rôle $administrateur$ *ne peut pas exister* sans violer les règles qui régissent l'exclusion mutuelle. Soit F l'ensemble des dépendances régissant l'exclusion mutuelle (section 5) ainsi que la dépendance f_{admin} modélisant ce rôle particulier. Soit $g : ssd(R_1, R_2), role(R_1), role(R_2) \rightarrow \perp$ la propriété que l'on veut savoir vérifiée "une contrainte d'exclusion mutuelle entre deux rôles *quels qu'ils soient* et l'existence d'un rôle $administrateur$ rendent *toute* politique inconsistante" :

1) $f_{admin} : role(R) \rightarrow hRole(administrateur, R)$,
2) $f_1 : hRole(R_1, R_2), ssd(R_1, R_2) \rightarrow \perp$,
3) $f_2 : hRole(R, R_1), hRole(R, R_2), ssd(R_1, R_2) \rightarrow \perp$,

Les procédures de preuves (Maher *et al.*, 1996; Beeri *et al.*, 1984; Coulondre, 2003) que nous avons évoqué en section 3.2 permettent de répondre à ce type de problèmes, et prouvent sur cet exemple que effectivement $F \models g$, l'existence d'une contrainte d'exclusion mutuelle interdit la présence d'un rôle $administrateur$.

7. Conclusion et Perspectives

Le DPPR est un dossier médical partagé entre les acteurs de la santé et réparti sur plusieurs établissement, dont l'échelle et la diversité illustre le besoin de modèle de contrôle d'accès expressif, capable de représenter des autorisations complexes. Cet article apporte une réponse cohérente aux problèmes de *représentation*, *stockage*, et d'*interrogation* des politiques d'autorisation, permettant d'*évaluer les décisions* de contrôle d'accès, d'*exprimer des contraintes* métiers complexes et de s'assurer de

l'*intégrité* des politiques. Nous avons illustré notre approche avec un adaptation au DPPR de la politique d'autorisation proposée par le GMSIH.

En effet, nous pensons que la prise en compte nécessaire de la confidentialité de l'information médicale doit se faire de la façon la plus homogène et la plus complète possible. Le domaine des BdD propose un large éventail d'outils théoriques et pratiques. La fertilisation croisée entre les notions d'*intégrité* et de *confidentialité* des données nous a permis de proposer une nouvelle gamme de contraintes sur les politiques : des contraintes d'intégrité qui permettent de s'assurer de la consistance à l'expression des contraintes métiers.

La taille considérable des SI actuels, rend l'administration centralisée des politiques d'autorisation difficile. C'est pour répondre à cette problématique d'administration distribuée qu'ont été proposés les modèles d'administration Administrative-RBAC (Sandhu *et al.*, 1999) et Scoped Administration of RBAC (Crampton *et al.*, 2003). Ces modèles définissent comment donner des privilèges aux différents administrateurs d'une politique RBAC. Nous pourrions adapter ces propositions au modèle PA-GMSIH et utiliser les récents travaux sur la vérification et simplification des dépendances pour s'assurer que les droits donnés aux administrateurs ne conduiront pas à une politique inconsistante.

Une perspective de recherche active est celle du contrôle d'accès impliquant des contraintes spatio-temporelles. Plusieurs propositions ont abordé le problème de l'intégration du temps et de l'espace dans le CA à base de rôles (Barker *et al.*, 2003; Joshi *et al.*, 2005). D'un autre côté, l'introduction des contraintes arithmétiques dans les dépendances semble être une manière élégante de pouvoir gérer des contraintes d'intégrité temporelles. Ceci permet d'exprimer des droits d'accès *contextuels*. Par exemple on peut avec un tel cadre pour restreindre l'utilisation d'un rôle r sur un intervalle $[h_1, h_2]$.

Dans un tout autre axe de recherche, nous pensons qu'il est possible de proposer un atelier complet de gestion des politique d'autorisation. Les graphes conceptuels par exemple, sont candidats à une représentation adéquate des politiques RBAC (Thion *et al.*, 2006a) et pourraient être intégrés à un tel atelier. Enfin, l'intégration d'autres propositions attenantes aux politiques de contrôle d'accès permettrait de proposer un outil avec lequel superviser la gestion des politiques RBAC de bout en bout : de l'ingénierie des rôles jusqu'à l'administration quotidienne en passant par la vérification.

8. Bibliographie

Abiteboul S., Hull R., Vianu V., *Foundations of Databases*, Addison-Wesley, 1995.

Ahn G.-J., Sandhu R. S., « Role-based authorization constraints specification. », *ACM Trans. Inf. Syst. Secur.*, vol. 3, n° 4, p. 207-226, 2000.

Barker S., « Deductive Database Security. », *in* E. Gudes, S. Shenoi (eds), *DBSec*, vol. 256 of *IFIP Conference Proceedings*, Kluwer, p. 103-114, 2002.

Barker S., Stuckey P. J., « Flexible access control policy specification with constraint logic programming. », *ACM Trans. Inf. Syst. Secur.*, vol. 6, n° 4, p. 501-546, 2003.

Beeri C., Vardi M. Y., « A Proof Procedure for Data Dependencies. », *J. ACM*, vol. 31, n° 4, p. 718-741, 1984.

Coulondre S., « A top-down proof procedure for generalized data dependencies. », *Acta Inf.*, vol. 39, n° 1, p. 1-29, 2003.

Coyne E. J., « Role engineering », *RBAC '95 : Proceedings of the first ACM Workshop on Role-based access control*, ACM Press, New York, NY, USA, p. 4, 1996.

Crampton J., Loizou G., « Administrative scope : A foundation for role-based administrative models », *ACM Trans. Inf. Syst. Secur.*, vol. 6, n° 2, p. 201-231, 2003.

Deswarte Y., Kalam A. A. E., « Modèle de sécurité pour le secteur de la santé », *Technique et Science Informatiques*, vol. 23, n° 3, p. 291-321, 2004.

Durand T., Spacagna H., Verdier C., Biron P., Flory A., « The Rhône-Alpes Health Platform », *Methods of Information in Medicine*, 2007. to appear.

Fagin R., « Inverting schema mappings », *Proceedings of the 25th symposium on Principles Of Database Systems*, ACM Press, New York, NY, USA, p. 50-59, 2006.

Ferraiolo D., Kuhn R., Chandramouli R., *Role-Based Access Control*, Artech House Publishers, 2003.

Gavrila S. I., Barkley J. F., « Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management. », *ACM Workshop on RBAC*, p. 81-90, 1998.

GMSIH, Politique de Sécurité des Systèmes d'Information des Etablissements de Santé - Politique d'autorisation, Technical report, GMSIH, Mars, 2003.

Halpern J. Y., Weissman V., « Using First-Order Logic to Reason about Policies. », *CSFW*, IEEE Computer Society, p. 187-201, 2003.

Harrison M. A., Ruzzo W. L., Ullman J. D., « Protection in operating systems », *Commun. ACM*, vol. 19, n° 8, p. 461-471, 1976.

Joshi J., Bertino E., Latif U., Ghafoor A., « A Generalized Temporal Role-Based Access Control Model. », *IEEE Trans. Knowl. Data Eng.*, vol. 17, n° 1, p. 4-23, 2005.

Kalam A. A. E., Politiques et Modèles de Sécurité pour les domaines de la santé et des affaires sociales, PhD thesis, Institut National Polytechnique de Toulouse (INPT), 2003.

Maher M. J., Srivastava D., « Chasing Constrained Tuple-Generating Dependencies. », *PODS*, ACM Press, p. 128-138, 1996.

Rihaczek K., « The harmonized ITSEC evaluation criteria », *Comput. Secur.*, vol. 10, n° 2, p. 101-110, 1991.

Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E., « Role-Based Access Control Models. », *IEEE Computer*, vol. 29, n° 2, p. 38-47, 1996.

Sandhu R. S., Munawer Q., « The ARBAC99 Model for Administration of Roles. », *ACSAC*, IEEE Computer Society, p. 229-240, 1999.

Thion R., Coulondre S., « Representation and Reasoning on Role-Based Access Control Policies with Conceptual Graphs. », *14th International Conference on Conceptual Structures*, vol. 4068 of *Lecture Notes in Computer Science*, p. 427-440, 2006a.

Thion R., Coulondre S., « Un Modèle Homogène pour la Confidentialité et l'Intégrité des Données Relationnelles. », *in* D. Laurent (ed.), *BDA*, 2006b.