Gestion des Conflits dans les Systèmes de Contrôles d'Accès Basés sur l'Organisation (OrBAC)

Handling Conflicts in Organization-Based Access Control (OrBAC)

Salem BENFERHAT Rania EL BAIDA
CRIL-CNRS, Université d'Artois, Rue Jean Souvraz, SP 18 62307 LENS Cedex, FRANCE
E-mail:{benferhat,elbaida}@cril.univ-artois.fr

Résumé

La modélisation formelle des politiques de sécurité, contrôlant l'accès aux informations sensibles, est un problème important dans de nombreux domaines, en particulier dans celui de la santé. Récemment, un nouveau système de contrôle d'accès, appelé OrBAC (Organization-Based Access Control) a été proposé. Il apporte de nombreuses solutions aux systèmes de contrôle d'accès existants, comme la prise en compte du contexte ou encore la possibilité de représenter les différentes formes de privilèges : permission, interdiction et obligation. Cependant, ce système ne permet pas de résoudre les conflits dus aux traitements simultanés des permissions ou obligations et des interdictions. Cet article traite le problème de la gestion des conflits dans le système OrBAC, modélisé formellement par une base de connaissances de la logique du premier ordre. Nous montrons que les approches de la gestion des incohérences proposées pour les bases de connaissances propositionnelles ne sont pas appropriées dans le cas de bases de connaissances du premier ordre. Nous proposons une approche basée sur l'affaiblissement des formules responsables du conflit.

Mots Clef

Contrôle d'accès basé sur l'organisation, gestion des conflits, bases de connaissances stratifiées.

Abstract

Modelling information security policies is being an important task in many domains, in particular in the health sector. Recently, a new access control system, called OrBAC (Organization-Based Access Control) has been proposed. This model brings many solutions to the existing access control systems. For instance, OrBAC takes into account the context and is able to represent various kinds of privileges: permission, prohibition and obligation. However, this system does not deal with conflicts due to the joint handling of permission or obligation and prohibition policies. This paper deals with the problem of handling conflicts in the OrBAC system, modelled by first order logic knowledge bases. We show that approaches suggested for handling conflicts in propositional knowledge bases are not adapted for inconsistent first order knowledge bases. We propose an approach in which we weak formulas responsible of conflicts.

Keywords

Organisation-based access control, handling conflicts, stratified knowledge bases.

1 Introduction

La modélisation formelle des politiques de sécurité est un problème important dans de nombreux domaines. Dans le secteur de la santé, par exemple, une altération illégale des informations contenues dans le dossier médical peut avoir des conséquences graves sur le diagnostic du patient. D'un autre côté, interdire à un médecin d'accéder au dossier médical dans une situation d'urgence peut avoir des effets désastreux sur l'état du patient. Assurer la sécurité d'un système consiste à garantir les propriétés de sécurité définissant les caractéristiques de confidentialité, d'intégrité et de disponibilité qui doivent être maintenues dans le système.

Il existe déjà plusieurs modèles de politiques de sécurité pour le secteur de la santé (voir [14] par exemple). Certains de ces modèles se basent sur la notion de rôles associés aux utilisateurs comme dans le système RBAC (*Role-Based Access Control*) [13] et TMAC (*Team-Based Access Control*) [7] basé sur la notion de groupes. Dans ces modèles, les droits d'accès sont attribués aux utilisateurs uniquement en fonction du rôle qu'ils exercent dans le système d'information.

Les systèmes à base de rôle existants ne sont pas entièrement satisfaisants. Par exemple, dans le système RBAC, les utilisateurs exerçant le rôle "médecin"' ont la permission de lire les dossiers médicaux des patients. Si cette règle assure la propriété de la disponibilité, elle ne garantit pas cependant la propriété de confidentialité puisqu'un médecin peut accéder aux dossiers de tous les patients, y compris ceux qu'il ne traite pas.

Récemment, un nouveau système de contrôle d'accès, appelé OrBAC (*Organisation-Based Access Control*) [1], a

été proposé et modélisé en logique du premier ordre. Ce système apporte trois contributions importantes au système RBAC. D'abord il est centré sur la notion d'organisation. Ceci permet de résoudre un problème important qui se résume par le fait que les politiques de sécurité en général diffèrent d'une organisation à une autre. Ensuite, OrBAC permet de prendre en compte le contexte, ce qui lui permet de représenter facilement les règles conditionnelles. Finalement, OrBAC permet de modéliser les trois modalités déontiques qui sont : obligation, permission et interdiction, alors que dans RBAC seules les permissions sont codées.
Cependant, le système OrBAC ne permet pas de résoudre les conflits dus aux traitements simultanés des permissions (ou obligations) et des interdictions.
Par exemple, un conflit peut être du au fait qu'une politique de sécurité définit une activité qu'un utilisateur est obligé d'exécuter et il existe des situations où l'exécution d'une telle activité est interdite. Prenons par exemple les deux règles suivantes : (i)"le médecin a l'obligation d'informer la famille du patient si ce dernier est atteint d'une maladie grave", et (ii)"le médecin a l'interdiction d'informer la famille du patient si ce dernier a préalablement interdit la révélation de sa maladie à sa famille". Si le patient est atteint d'une maladie grave et a interdit la révélation de sa maladie à sa famille, on obtient un conflit. Ce conflit peut être résolu en considérant que la règle (ii) est prioritaire sur la règle (i).
L'un des objectifs de ce travail est d'étudier le problème des conflits dans le système OrBAC. Il existe déjà plusieurs travaux qui traitent le problème des conflits dans les bases de connaissances propositionnelles. Parmi ces travaux on trouve des approches dites de "cohérence" dont le principe est :
i)sélectionner une ou plusieurs sous-bases cohérentes de la base incohérente, puis
ii)appliquer la relation d'inférence classique sur chacune de ces sous-bases sélectionnées.
Parmi ces approches, on trouve l'approche possibiliste [5], l'approche basée sur les sous-bases maximales pour l'inclusion [4], l'approche dite d'ajustement maximal [15] et l'approche dite par cardinalité (ou lexicographique) basé sur les sous-bases maximales cohérentes pour les critères de la cardinalité [2, 9].
Si certaines de ces approches sont satisfaisantes dans le cas propositionnel, leurs applications à la logique du premier ordre donnent des résultats non intuitifs. Par exemple, l'approche par cardinalité dans le cas du premier ordre souffre du problème dit de la noyade [2], et produit des résultats "indésirables".
Les sections 2 et 3 présentent les différents concepts de base utilisés dans le modèle OrBAC ainsi que son codage en logique du premier ordre. Les sections 4, 5 et 6 présentent le problème des conflits dans OrBAC, puis montrent les limites des approches propositionnelles et proposent une extension au premier ordre.

2 Le contrôle d'accès basé sur l'organisation

L'objectif d'un système de contrôle d'accès est de limiter l'accès aux objets du système d'information uniquement aux utilisateurs ayant une autorisation définie par une politique de sécurité. Une politique de sécurité définit les règles d'attributions des privilèges aux utilisateurs.
En général, l'attribution des privilèges ne se fait pas d'une façon explicite pour chaque utilisateur, et pour chaque objet. Les privilèges sont plutôt définis en fonction du rôle que chaque utilisateur joue dans une organisation. De même les privilèges ne portent pas directement sur les objets, mais plutôt sur des classes d'objets, appelés vues, partageant des propriétés similaires.
Cette section rappelle les principaux concepts du modèle OrBAC [1], présentés sous forme d'entités-associations.

2.1 Les organisations

L'entité centrale du modèle OrBAC est l'*organisation*. Elle peut être vue comme un groupe de sujets jouant certains rôles. Dans le domaine médical, un exemple d'organisation peut être l'unité de soins intensifs de l'hôpital Rangueil.

2.2 Les sujets et les rôles

Dans le modèle OrBAC, un *Sujet* peut être soit un utilisateur (ex., Jean), soit une organisation (ex., le service des urgences de l'hôpital Purpan).
L'entité *Rôle* est utilisé pour structurer le lien entre les sujets et les organisations. Les rôles associent des sujets qui remplissent les mêmes fonctions. Dans le domaine médical, les rôles "infirmière" et "cardiologue" sont joués par des utilisateurs alors que les rôles "service des urgences" et "unité des soins intensifs" sont joués par des organisations.
Comme les sujets jouent des rôles dans des organisations, la relation *Habilite(org,s,r)* signifie que l'organisation org habilite le sujet s à jouer le rôle r. Par exemple, la relation *Habilite(Purpan,Jean,cardiologue)* signifie que l'hôpital Purpan habilite Jean dans le rôle cardiologue.

2.3 Les objets et les vues

Dans le modèle OrBAC, l'entité *Objet* représente principalement les entités non actives. Dans le domaine médical, un exemple d'objet est le dossier médical d'un patient.
L'entité *vue* regroupe un ensemble d'objets qui satisfait une propriété commune. Par exemple, la vue "dossiers médicaux" correspond aux dossiers médicaux des patients. Dans la mesure où les vues caractérisent la manière dont les objets sont utilisés dans l'organisation, la relation *Utilise(org,o,v)* signifie que l'oragnisation org utilise l'objet o dans la vue v. Par exemple, la relation $Utilise(Purpan, F31.doc, dossier\_médical)$ signifie que l'hôpital Purpan utilise F31.doc comme un dossier médical.

2.4 Les actions et les activités

Dans le modèle OrBAC, l'entité *Action* englobe principalement les actions informatiques comme "lire", "écrire", etc. De la même façon que les rôles et les vues sont des abstractions des sujets et des objets, l'entité *Activité* correspond à des actions qui ont un objectif commun (ex., consulter, modifier, etc.).
La relation *Considère* sera utilisée pour associer les entités *organisation, Action* et *Activité*. Plus précisément, $Considère(org, \alpha, a)$ signifie que l'organisation org considère l'action α comme faisant partie de l'activité a. Par exemple, la relation *Considère(Purpan,lire,consultation)* signifie que l'hôpital Purpan considère l'action lire comme faisant partie de l'activité consultation.

2.5 Les contextes

Les contextes sont utilisés pour spécifier les circonstances concrètes dans lesquelles les organisations accordent des permissions de réaliser des activités sur des vues. La relation *Definit(org,s,α,o,c)* donne la définition du contexte c par l'organisation org à partir du sujet s, de l'action α et de l'objet o. Par exemple, la relation *Definit(Purpan,Jean,lire,F31.doc,médecin-traitant)* signifie que si Jean est le médecin traitant du patient auquel appartient l'objet $F31.doc$, alors Jean a accès en lecture au fichier F31.doc.

2.6 Les privilèges

Dans le modèle OrBAC, les relations *Permission, Interdiction* et *Obligation* correspondent à des relations entre les organisations, les rôles, les vues, les activités et les contextes. La relation *Permission(org,r,a,v,c)* signifie que l'organisation org accorde au rôle r la permission de réaliser l'action a sur la vue v dans le contexte c. Les relations *Interdiction(org,r,a,v,c)* et *Obligation(org,r,a,v,c)* sont définies de façon similaire.
Les relations *Permission, Interdiction* et *Obligation* sont introduites comme des faits. Elles ne sont pas directement associées aux utilisateurs, actions et objets mais à leur abstraction respectives rôles, activités et vues.
Par exemple, la politique de sécurité de l'hôpital Purpan peut contenir le fait suivant :
$Permission(Purpan, médecin, consulter, dossier\_médical, médecin\_traitant)$
qui signifie que l'hôpital Purpan accorde aux médecins la permission de consulter les dossiers médicaux des patients dont ils sont les médecins traitants.

2.7 Les autorisations concrètes

Les relations *Permission, Interdiction* et *Obligation* permettent à une organisation donnée de spécifier les permissions accordées suivant le contexte. De telles permissions correspondent à une relation entre les rôles, les vues et les activités. Or dans la pratique, les requêtes concernent des permissions (resp. interdictions et obligations), dites concrètes, accordées aux sujets pour effectuer des actions sur des objets. Dans le but de décrire des actions concrètes que réalisent les sujets sur les objets, nous introduisons les relations $Est\_permis, Est\_interdit$ et $Est\_obligatoire$. La relation $Est\_permis(s, \alpha, o)$ signifie qu'il est permis au sujet s de réaliser l'action α sur l'objet o. Les relations $Est\_interdit$ et $Est\_obligatoire$ sont définies de la même manière. Ces relations sont dérivées logiquement des permissions accordées aux rôles, vues et activités par les relations *Permission, Interdiction* et *Obligation*.
Notons que les instances explicites des relations $Est\_permis, Est\_interdit$ et $Est\_obligatoire$ peuvent être utilisées pour exprimer des exceptions à la politique de sécurité générale spécifiée par les relations *Permission, Interdiction* et *Obligation*.
La figure 1 résume le modèle de sécurité OrBAC.

3 Codage logique du modèle OrBAC

Dans cette section, nous présentons le codage du modèle OrBAC en logique du premier ordre. La logique du premier ordre est suffisante car les modalités déontiques (Permission, Interdiction, Obligation) dans le modèle OrBAC portent sur des actions simples. Le modèle OrBAC ne contient pas d'imbrication de modalités et les modalités ne portent pas sur des disjonctions d'actions.
Le vocabulaire utilisé contient des symboles de constantes, de variables individuelles, de symboles de relations et de symboles de fonctions.
Les symboles de constantes correspondent aux instances des entités du diagramme. Les symboles de constante et les variables individuelles $(x, y, z \ldots)$ peuvent être de l'un des types *Organisation, Sujet, Objet, Action, Rôle, Vue, Activité* et *Contexte* (représentant les domaines des entités du modèle OrBAC). Les symboles de relation (prédicats) corrrespondent aux relations du modèle OrBAC. Les symboles de fonction permettent de dénommer des valeurs d'attributs pour des entités, par exemple $médecin\_traitant(x)$ représente un attribut de x, où x est de type sujet, et la valeur de $médecin\_traitant(x)$ est de type sujet.
Nous utiliserons des relations binaires pour pouvoir comparer les valeurs d'attributs d'entités (ex., $=$, $>$, $<$, etc.). Ainsi, si t et u sont des termes de types *Sujet*, alors $médecin\_traitant(t) = médecin\_traitant(u)$ signifie que les sujets t et u ont les mêmes médecins traitants.
Les formules du langage sont construite à partir des connecteurs logiques : $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$. Par exemple, la formule $\forall s(Habilite(Purpan, s, médecin) \rightarrow Habilite(Purpan, s, personnel\_soignant))$ signifie que tous les médecins de l'hôpital Purpan sont habilités comme personnel soignant.

3.1 Différentes composantes de la base OrBAC

Les informations disponibles sont codées par une base de connaissances de la logique du premier ordre. Elle est com-

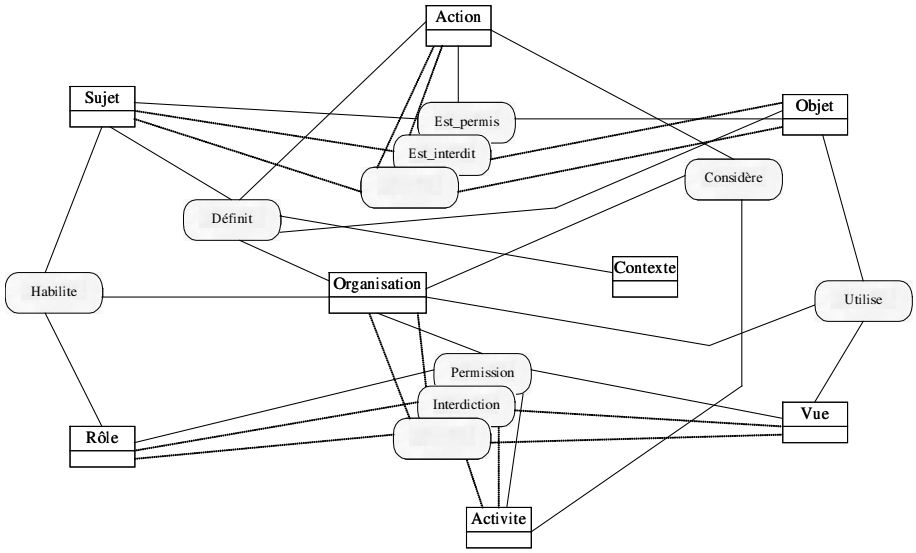

Figure 1: *Le modèle OrBAC*

posée des parties suivantes :

- **Ensemble de faits** : Nous distinguons deux sortes de faits :
  a- *Des faits codant les relations OrBAC* : Ils concernent les relations décrites dans la section précédente (i.e., relations du schéma relationnel de la figure 1) à l'exception de la relation $Definit$ qui n'est pas donnée comme fait. Par exemple, on peut avoir les faits suivants :
  - $Permission(Purpan, infirmière, consultation, champs\_chirurgical, urgence)$ (i.e., dans l'hôpital Purpan, une infirmière a la permission de consulter le champs chirurgical des patients dans le contexte d'urgence);
  - $Habilite(Purpan, Mary, infirmière)$ (i.e., l'hôpital Purpan habilite Mary pour être une infirmière);
  - $Utilise(Purpan, champs\_chirurgical\_JO, champs\_chirurgical)$ (i.e., l'organisation Purpan utilise le champs chirurgical du patient JO comme un champs chirurgical);

  b- *Des faits reliés aux attributs des entités du modèle OrBAC* : Ces faits donnent les informations nécessaires concernant les attributs des relations. Par exemple, on peut avoir les faits suivants :
  - $en\_grève(Jean)$, qui signifie que le sujet $Jean$ est en grève.
  - $Nom(dossier\_médical\_JO) \in Patient(Jean)$, qui signifie que le patient JO auquel correspond l'objet $dossier\_médical\_JO$ est un des patients du sujet Jean.
- **Ensemble de règles certaines** : Nous distinguons trois sortes de formules certaines :
  a- *les règles d'hiérarchie* : Les rôles, les vues et les activités sont organisés sous formes de hiérarchies. Par exemple, les médecins d'un hôpital font parties du personnel soignant de cet hôpital. Cette hérarchie implique l'héritage des privilèges. Par exemple, les médecins hériteront les privilèges accordés au rôle personnel soignant. De ce fait, les règles de hiérarchie entre les rôles sont écrites en deux parties :
  1.Hiérarchie des rôles : par exemple,
  $\forall s(Habilite(Purpan, s, médecin) \rightarrow Habilite(Purpan, s, personnel\_soignant))$:
  pour exprimer que tout utilisateur habilité à jouer le rôle médecin, est également habilité à jouer le rôle personnel soignant dans l'organisation hôpital Purpan.

  2.Héritage des privilèges : par exemple,
  $\forall a \forall v \forall c$
  $(Permission(Purpan, personnel\_soignant, a, v, c) \rightarrow Permission(Purpan, médecin, a, v, c))$: qui exprime le fait que dans l'hôpital Purpan, les médecins héritent les permissions accordées au rôle personnel soignant.

  Le même principe s'applique à une hiérarchie entre les activités ou encore sur les vues. Par exemple, la vue $champs\_chirurgical$ est une des sous vues de la vue $dossier\_médical$. Cette hiérarchie implique une hiérarchie entre la vue $champs\_chirurgical$ et la vue $dossier\_médical$ par la relation
  $\forall o, Utilise(Purpan, o, champs\_chirurgical) \rightarrow Utilise(Purpan, o, dossier\_médical)$,
  et un héritage des permissions :
  $\forall r \forall a \forall c$
  $(Permission(Purpan, r, a, dossier\_médical, c) \rightarrow$

$Permission(Purpan, r, a, champs\_chirurgical, c)$).

b- *les règles décrivant les contextes* : Contrairement aux autres relations de la figure 1 qui sont données comme des faits, la relation $Definit(org, s, \alpha, o, c)$ dans le modèle OrBAC n'est pas donnée comme un fait, mais exprimée par des formules du premier ordre, composées de conditions qui portent sur les attributs de paramètres de "$Definit$", à savoir org, α, o, s.
Par exemple, le contexte $en\_grève$ est défini par :
$\forall s \forall \alpha \forall o (Definit(Purpan, s, \alpha, o, en\_grève) \leftrightarrow en\_grève(s) = vrai)$ (i.e., dans l'hôpital Purpan, le contexte "$en\_grève$" est vrai entre le sujet s, l'action α et l'objet o si et seulement si le sujet s est en grève).

c- *les liens entre modalités* : Dans tout le reste de l'article, nous ne traiterons que les permissions et les interdictions. Cette restriction est faite pour des raisons de simplicité, et ne complique pas particulièrement le problème de gestion des incohérences dans les bases de connaissances du premier ordre.
Nous avons besoin de cet axiome pour la relation Est_permis et $Est\_interdit$,
$\forall s \forall \alpha \forall o$
$(Est\_permis(s, \alpha, o) \leftrightarrow \neg Est\_interdit(s, \alpha, o))$:
tous ce qui est permis n'est pas interdit.

- **Ensemble de règles incertaines (avec exceptions)**
La dernière partie de la base concerne le passage des permissions abstraites $Permission(org, r, a, v, c)$ (resp. *Interdiction* et *Obligation*), qui portent sur les rôles, activités et vues, aux permissions concrètes $Est\_permis(s, \alpha, o)$ (resp. $Est\_interdit$ et $Est\_obligatoire$)), qui portent sur les sujets, actions et objets. Ce passage est obtenu par l'ensemble d'axiomes suivants (pour chaque contexte c) :
$\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Permission(org, r, a, v, c) \wedge$
$Habilite(org, s, r) \wedge$
$Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, c)$
$\rightarrow Est\_permis(s, \alpha, o)$:
Pour toute organisation org, si org, dans le contexte c, accorde la permission au rôle r de réaliser l'activité a sur la vue v, et si org habilite le sujet s dans le rôle r, et si org utilise l'objet o dans la vue v, et si org considère l'action α comme faisant partie de l'activité a et si au sein de l'organisation org le contexte c est vrai entre s, α et o, alors le sujet s a la permission de réaliser l'action α sur l'objet o.
Le passage à la relation "$Est\_interdit$" se fait de façon similaire.
Remarquons que ces axiomes ne sont pas quantifiés par le quantificateur universel $\forall$ par rapport au contexte c. Pour prendre en compte la hiérarchie entre les rôles, les activités et les vues, ces règles devront aussi être instanciées par rapport à chaque rôle, activité et vue. Pour des raisons de clarté, on se restreint dans ce papier à instancier uniquement le contexte.

3.2 Exemples

Nous illustrons la description de la base à travers l'exemple de règles conflictuelles suivant qui sera repris dans le reste de l'article.

Exemple 1 *Supposons que nous avons l'hôpital Purpan, dans lequel nous supposons que nous avons un patient JO, un médecin Jean qui est le médecin traitant de JO, l'activité consultation qui revient à l'action lire, la vue dossier médical du patient qui contient le dossier médical du patient JO.*
Supposons que la politique de sécurité contient les deux règles suivantes :

1. *Un médecin a la permission de consulter le dossier médical des patients dont il est le médecin traitant.*
2. *Un médecin en grève a l'interdiction de consulter le dossier médical des patients.*

De ces deux règles, on extrait les deux rôles "patient" et "médecin", ainsi que les deux contextes "médecin traitant" qui prend la valeur vrai si le patient est l'un des patients du sujet jouant le rôle médecin et "en grève" qui sera validé si le médecin est en grève. Supposons que Jean est en grève.
L'ensemble des règles se code comme suit :

- **ensemble de faits :**
$\mathbf{R_1}$. $Permission(Purpan, médecin, consultation, dossier\_médical, médecin\_traitant)$.
$\mathbf{R_2}$. $Interdiction(Purpan, médecin, consultation, dossier\_médical, en\_grève)$.
$\mathbf{R_3}$. $Considère(Purpan, lire, consultation)$.
$\mathbf{R_4}$. $Utilise(Purpan, dossier\_médical\_JO, dossier\_médical)$.
$\mathbf{R_5}$. $Habilite(Purpan, Jean, médecin)$.
$\mathbf{R_6}$. $Habilite(Purpan, JO, patient)$.
$\mathbf{R_7}$. $Nom(dossier\_médical\_JO) \in Patient(Jean)$.
$\mathbf{R_8}$. $en\_grève(Jean) = vrai$.

- **ensemble de règles certaines :**
$\mathbf{R_9}$. $\forall s \forall \alpha \forall o$
$(Est\_permis(s, \alpha, o) \leftrightarrow \neg Est\_interdit(s, \alpha, o))$.
$\mathbf{R_{10}}$. $\forall s$
$Definit(Purpan, s, lire, dossier\_médical\_JO, médecin\_traitant) \leftrightarrow$
$(Nom(dossier\_médical\_JO)^1 \in Patient(s)^2)$.
$\mathbf{R_{11}}$. $\forall s$
$Definit(Purpan, s, lire, dossier\_médical\_JO, en\_grève) \leftrightarrow (en\_grève(s) = vrai)$.

[1] Nom(o) retourne le nom du patient correspondant à l'objet o.
[2] Patient(s) retourne la liste des patients du sujet s.

- **ensemble des règles incertaines :**
$\mathbf{R_{12}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Permission(org, r, a, v, médecin\_traitant) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, médecin\_traitant) \rightarrow$
$Est\_permis(s, \alpha, o)$.

$\mathbf{R'_{12}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Permission(org, r, a, v, en\_grève) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, en\_grève) \rightarrow$
$Est\_permis(s, \alpha, o)$.

$\mathbf{R_{13}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Interdiction(org, r, a, v, médecin\_traitant) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, médecin\_traitant) \rightarrow$
$Est\_interdit(s, \alpha, o)$.

$\mathbf{R'_{13}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Interdiction(org, r, a, v, en\_grève) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, en\_grève) \rightarrow$
$Est\_interdit(s, \alpha, o)$.

On remarque qu'à partir de cette base, et en utilisant les règles $\{R_1, R_3, R_4, R_5, R_6, R_7, R_{10}\}$, *on infère* "$Est\_permis(Jean, lire, dossier\_médical\_JO)$" *de la règle* R_{12}. *Tandis que si on utilise les règles* $\{R_2, R_3, R_4, R_5, R_6, R_8, R_{11}\}$, *on infère* "$Est\_interdit(Jean, lire, dossier\_médical\_JO)$" *de la règle* R'_{13}*; Ce qui aboutit à un conflit (règle* R_9*).*

4 Résolution des conflits : Application des approches propositionnelles

La logique classique n'est pas appropriée pour la gestion des bases de connaissances incohérentes, puisqu'elle ne produit que des conclusions triviales.
Il est bien connu que la notion de priorité est très importante pour la résolution des conflits[6]. L'exemple précédent peut-être facilement résolu si on dispose d'une relation de priorité entre les règles (en particulier entre les règles R_{12} et R_{13} puisque ce sont les seules qui ne sont pas certaines).
La stratification peut être donnée par un expert ou bien calculée automatiquement. Un exemple de calcul automatique est l'algorithme de Pearl [11] ou son équivalent l'algorithme de minimun de spécificité détaillé dans [3], lorsque les conflits sont causés par les règles avec exceptions. La stratification reflète le niveau de spécificité entre les règles ayant des exceptions. Dans ces systèmes,

- les règles sans exceptions (i.e., règles certaines) et les faits sont toujours préferées aux règles avec exceptions (i.e., règles incertaines).
- les règles avec exceptions sont entre elles ordonnées comme suit : une règle codant une situation exceptionnelle doit être préferée à une règle codant une situation générale (sinon elle ne sera jamais appliquée).

Une règle est dite "exceptionnelle" si le fait de rendre l'antécédant de cette règle vrai cause l'incohérence de la base (voir [3] pour plus de détails).
Une base stratifiée est de la forme $\Sigma = S_1 \cup \ldots \cup S_n$ où les formules de S_i ont toutes le même degré de priorité et est telle que S_1 contient les formules les moins certaines et S_n contient les formules les plus certaines.
On s'intéresse à savoir si à partir de Σ on arrive à inférer une observation ψ. Plusieurs approches ont été proposées pour inférer des conclusions non-triviales à partir d'une base de croyances, incohèrente et stratifiée [2], [4], [10], [15]. Parmi ces approches, on trouve celles dites de cohérence qui suivent le schéma suivant pour la déduction des conclusions plausibles, proposées par Rescher et Manor [12] : sélectionner une ou plusieurs sous-bases cohérentes de Σ, puis ψ est considérée plausible si, par inférence classique, on obtient ψ de chacune de ces sous-bases sélectionnées.
Dans cet article, nous rappelons deux de ces méthodes qui sont l'inférence possibiliste [5] et l'inférence lexicographique [2], [9].

L'inférence possibiliste : La logique possibiliste est une extension de la logique classique. Elle associe à chaque formule un degré qui reflète son niveau de priorité par rapport aux autres formules de la base.
Pour savoir si une formule ψ est une conséquence possibiliste de la base Σ, nous sélectionnons une sous-base de Σ, notée $\delta(\Sigma)$, et définie comme suit : $\delta(\Sigma) = S_i \cup \ldots \cup S_n$, où i est tel que $S_i \cup \ldots \cup S_n$ est cohérent, mais cependant $S_{i-1} \cup S_i \ldots \cup S_n$ ne l'est pas.
Puis ψ est dite une conséquence possibiliste de Σ si et seulement si $\delta(\Sigma) \vdash \psi$.

Exemple 2 *Reprenons l'exemple 1. On s'intéresse à savoir s'il est permis au médecin Jean de lire le dossier médical de son patient JO, si Jean est en grève.*
Supposons que la formalisation des règles aboutit à la base stratifiée suivante : $\Sigma = S_1 \cup S_2 \cup S_3$*, où*
S_3 *contient les faits et les règles certaines,* S_2 *contient la règle concernant le passage d'une instanciation abstraite à l'interdiction concrète, et* S_1 *concerne le passage à la permission concrète (c'est à dire, on considère que l'interdiction explicite a une grande priorité). Plus précisement,*
$S_3 = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}\}$,
$S_2 = \{R_{13}, R'_{13}\}$,
$S_1 = \{R_{12}, R'_{12}\}$.
Appliquons le principe d'inférence possibiliste. On selectionne la sous-base $\delta(\Sigma) = S_3 \cup S_2$ *(notons que l'ajout*

de S_1 à cette sous-base provoque une incohérence). Nous remarquons, qu'à partir de cette sous-base, nous pouvons inférer l'interdiction de Jean pour lire le dossier médical de JO, c'est-à-dire,
$\delta(\Sigma) \vdash Est\_interdit(Jean, lire, dossier\_médical\_JO)$*, obtenue à partir des règles et faits* $\{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{13}, R'_{13}\}$.

Inférence Lexicographique : L'inférence possibiliste est "efficace" car elle est d'une complexité proche de celle de l'inférence classique ($log_2(n)$ où n est le nombre de strates dans Σ, voir [8] pour plus de détails) , mais elle est cependant prudente et souffre du problème de la noyade [2], comme le montre l'exemple suivant :

Exemple 3 *Prenons l'exemple précedent, et ajoutons à la base la règle suivante :*

- *Une infirmière a la permission de lire le dossier médical des patients,*

ainsi que le fait suivant :

- *Mary est une infirmière.*

Ces deux règles se codent formellement comme suit :
$\mathbf{R''_{12}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Permission(org, r, a, v, cas\_général) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge Definit(org, s, \alpha, o, cas\_général)$
$\rightarrow Est\_permis(s, \alpha, o)$.

$\mathbf{R''_{13}}$. $\forall org \forall s \forall \alpha \forall o \forall r \forall a \forall v$
$Interdiction(org, r, a, v, cas\_général) \wedge$
$Habilite(org, s, r) \wedge Utilise(org, o, v) \wedge$
$Considère(org, \alpha, a) \wedge Definit(org, s, \alpha, o, cas\_général)$
$\rightarrow Est\_interdit(s, \alpha, o)$.

$\mathbf{R_{14}}$. $Permission(Purpan, infirmière, consultation, dossier\_médical, cas\_général)$.
$\mathbf{R_{15}}$. $\forall s, Definit(Purpan, s, lire, dossier\_médical\_JO, cas\_général) \leftrightarrow \top$.
$\mathbf{R_{16}}$. $Habilite(Purpan, Mary, infirmière)$.

L'ensemble de règles $\{R_1, \ldots, R_{16}\}$ *donne lieu à la base stratifiée suivante :*
$S_4 = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{14}, R_{15}, R_{16}\}$,
$S_3 = \{R'_{13}\}$,
$S_2 = \{R_{12}, R'_{12}, R''_{12}\}$,
$S_1 = \{R_{13}, R''_{13}\}$.
Cette base est incohérente et seulement les formules se trouvant dans les strates S_4, S_3 sont selectionnées par l'inférence possibiliste; A partir de cette sous-base nous ne pouvons pas inférer qu'il est permis à Mary de lire le dossier médical de JO.

Pour prendre en compte les formules les moins certaines, une deuxième relation d'inférence, dite lexicographique (appelée aussi l'inférence pour la cardinalité), a été définie [2],[9]. L'idée est de sélectionner non pas une seule sous-base cohérente mais plusieurs sous-bases maximales cohérentes. Une sous-base $A \subseteq \Sigma$ est dite lexicographiquement préférée à une sous-base $B \subseteq \Sigma$, noté $A >_{Lex} B$, s'il existe un niveau $n \geq i \geq 1$ tel que :

- $|A \cap S_i| > |B \cap S_i|$ et
- $\forall j > i, j \leq n, |A \cap S_j| = |B \cap S_j|$.

Nous notons par $Lex(\Sigma)$ les sous-bases lexicographiquement cohérentes préférées de Σ', c'est-à-dire : $Lex(\Sigma) = \{A : A \subseteq \Sigma$ est cohérente et $\nexists B \subseteq \Sigma'$ cohérente, $B >_{Lex} A\}$.

Exemple 4 *Reprenons l'exemple précedent. Il existe une seule sous-base lexicographiquement cohérente, qui est* $A = S_4 \cup S_3 \cup \{R''_{12}\} \cup \{R_{13}\}$.
A partir de cette sous-base, nous pouvons inférer qu'il est permis à Mary de lire le dossier médical de JO (i.e., $A \vdash Est\_permis(Mary, lire, dossier\_médical\_JO)$*).*

5 Limites des approches par cohérence

Les approches par cohérence sont très satisfaisantes pour traiter le problème de la noyade dans le cas des bases propositionnelles. Cependant, elles échouent dans le cas des bases de connaissances composées de formules du premier ordre.
L'application de l'approche lexicographique à une base de connaissances du premier ordre souffre non seulement du problème de la noyade mais peut aussi aboutir à des résultats indésirables. Ceci peut être expliqué par le fait que l'inférence lexicographique procède niveau par niveau en supprimant quelques formules du premier ordre, pour résoudre le conflit. La suppression de ces formules rend non seulement impossible d'appliquer leurs instances non conflictuelles (ce qui provoque le problème du noyade), mais aussi conduit à l'ajout des formules moins prioritaires permettant de conclure des résultats indésirables.
L'exemple 5 illustre une situation où l'inférence lexicographique souffre du problème de la noyade, et l'exemple 6 illustre un cas où l'inférence lexicographique donne un résultat non-désirable.

Exemple 5 *Reprenons l'exemple 1, et rajoutons les faits suivants :*

- *Chris est un médecin.*
- *Chris n'est pas en grève.*
- *Chris est le médecin traitant de JO.*

Ces règles se codent formellement de la façon suivante :
$\mathbf{R_{14}}$. $Habilite(Purpan, Chris, médecin)$.
$\mathbf{R_{15}}$. $en\_grève(Chris) = faux$.
$\mathbf{R_{16}}$. $Nom(dossier\_médical\_JO) \in Patient(Chris)$.

Le codage de l'ensemble des règles $\{R_1, \ldots, R_{20}\}$ *aboutit à la base stratifiée suivante :*
$S_3 = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{14}, R_{15}, R_{16}\}$,
$S_2 = \{R_{13}, R'_{13}\}$,
$S_1 = \{R_{12}, R'_{12}\}$.

Le résultat attendu est qu'il est permis à $Chris$ *de lire le dossier médical de JO (i.e.,* $Est\_permis(Chris, lire, dossier\_médical\_JO)$*).*

En utilisant le principe d'inférence lexicographique, on sélectionne une seule sous-base
$A = S_3 \cup S_2$*; On remarque que*
$A \nvdash Est\_permis(Chris, lire, dossier\_médical\_JO)$.

Dans cet exemple, la règle R_{12} est responsable du conflit. Le fait de la supprimer rend impossible son application pour l'utilisateur $Chris$.

Exemple 6 *Enrichissons l'exemple 5 pour contenir l'ensemble des règles suivantes :*

- *Un personnel de l'hôpital n'a pas la permission de lire les dossiers médicaux des patients.*
- *Un médecin a la permission de lire le dossier médical des patients dont il est le médecin traitant.*
- *Un médecin en grève n'a pas la permission de lire le dossier médical des patients.*
- *Un médecin est membre du personnel de l'hôpital.*
- *JO est un patient de l'hôpital Purapn.*
- *Jean est un médecin de l'hôpital Purpan.*
- *Chris est un médecin de l'hôpital Purpan.*
- *Jean est le médecin traitant de JO.*
- *Chris est le médecin traitant de JO.*
- *Jean est en gève.*
- *Chris n'est pas en grève.*

Les résultats attendus sont :
Il est interdit a Jean de lire le dossier médical de JO (i.e., $Est\_interdit(Jean, lire, dossier\_médical\_JO)$*) et il est permis a Chris de lire le dossier médical de JO (i.e.,* $Est\_Permis(Chris, lire, dossier\_médical\_JO)$*).*

La formalisation de ces règles donne l'ensemble des règles $\{R_1, \ldots, R_{16}\}$ *de l'exemple 5 avec l'ajout des règles suivantes :*
$\mathbf{R_{17}}$. $Interdiction(Purpan, personnel, consultation, dossier\_médical, cas\_général)$.
$\mathbf{R_{18}}$. $\forall s, Habilite(Purpan, s, médecin) \rightarrow Habilite(Purpan, s, personnel)$.
$\mathbf{R_{19}}$. $\forall a \forall v \forall c$
$Permission(Purpan, personnel, a, v, c) \rightarrow Permission(Purpan, médecin, a, v, c)$;
$\mathbf{R_{20}}$. $\forall s, Definit(Purpan, s, lire, dossier\_médical\_JO, cas\_général) \leftrightarrow \top$.

L'ensemble de ces règles aboutit à la stratification suivante :
$S_4 = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}, R_{20}\}$,
$S_3 = \{R'_{13}\}$,
$S_2 = \{R_{12}, R'_{12}, R''_{12}\}$,
$S_1 = \{R_{13}, R''_{13}\}$.
Le résultat attendu est que $Chris$ *doit avoir la permission de lire le dossier médical de JO (i.e.,* $Est\_permis(Chris, lire, dossier\_médical\_JO)$*).*

De cette base stratifiée, on peut avoir une seule sous base lexicographique :
$A = S_4 \cup S_3 \cup S_1$. *De cette sous-base, on infère*
$A \vdash Est\_interdit(Chris, lire, dossier\_médical\_JO)$.

Ce résultat non intuitif est expliqué d'abord par le fait de la suppression de la règle R_{12} (solution radicale pour résoudre les conflits des règles prioritaires de S_4 et S_3). Ensuite, le fait de supprimer cette règle, rend possible l'ajout de la règle R''_{13}, ce qui explique l'inférence de la conclusion non-intuitive.

6 Adaptation des approches par cohérence à la logique du premier ordre

Nous venons de voir que l'inférence lexicographique telle que, définie dans le cadre de la logique propositionnelle, n'est pas appropriée pour les bases incohérentes du premier ordre. Au fait, ces résultats contre intuitifs s'appliquent généralement pour les autres approches tolérant l'incohérence et basées sur la restauration de la cohérence. Par exemple, l'approche basée sur les sous-bases préférées pour la cardinalité [4], l'approche linéaire [10] et l'approche dite d'ajustement maximal [15], toutes échouent à résoudre correctement l'incohérence des bases de connaissances données dans les exemples 5 et 6.
Les limites des approches basées sur la restauration de la cohérence sont expliquées par le fait que la suppression d'une formule du premier ordre, revient à la suppression d'un ensemble de formules propositionnelles. Ceci n'est pas satisfaisant, car par exemple s'il s'avère qu'une formule de la forme $\forall x \phi(x)$ est responsable du conflit, alors il est rare que toutes les instances de cette formule soient également responsables du conflit.
A partir de cette remarque, une formule du premier ordre responsable d'un conflit ne doit pas être supprimée, sauf si toutes ses instances sont également responsables du con-

flit. Elle doit être affaiblie en restreignant l'application de la formule aux seules instances non responsables du conflit.
Par exemple, si la formule $\forall x \phi(x)$ est seulement conflictuelle pour $x = a$, alors cette formule sera remplacée par $\forall x, \neg(x = a) \rightarrow \phi(x)$.
Appliquons maintenant cette remarque au système OrBAC. Les formules à affaiblir concernent toujours la règle de passage des permissions (resp. interdictions) abstraites aux permissions (resp. interdictions) concrètes.
Pour des raisons de simplicité, notons $\phi_P(org, r, a, v, c)$ (resp. $\phi_I(org, r, a, v, c)$) les formules de passage des permissions (resp. interdictions) abstraites aux permissions (resp. interdictions) concrètes pour un contexte donné c, c'est-à-dire :
$\phi_P(org, r, a, v, c, s, o, \alpha) \equiv$
$\forall org \forall r \forall a \forall v \forall s \forall o \forall \alpha$
$Permission(org, r, a, v, c) \wedge Habilite(org, s, r) \wedge$
$Utilise(org, o, v) \wedge Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, c) \rightarrow Est\_permis(s, \alpha, o)$

(resp. $\phi_I(org, r, a, v, c, s, o, \alpha) \equiv$
$\forall org \forall r \forall a \forall v \forall s \forall o \forall \alpha$
$Interdiction(org, r, a, v, c) \wedge Habilite(org, s, r) \wedge$
$Utilise(org, o, v) \wedge Considère(org, \alpha, a) \wedge$
$Definit(org, s, \alpha, o, c) \rightarrow Est\_interdit(s, \alpha, o)$).

Notons, "*Différent*$(s, i_s, \alpha, i_\alpha, o, i_o)$" comme l'abréviation de la formule $\neg((s = i_s) \wedge (\alpha = i_\alpha) \wedge (o = i_o))$.

Une formule $\phi_{Aff_P}(org, r, a, v, c, s, o, \alpha)$ (resp. $\phi_{Aff_I}(org, r, a, v, c, s, o, \alpha)$) est dite une formule affaiblie de $\phi_P(org, r, a, v, c, s, o, \alpha)$ (resp. $\phi_I(org, r, a, v, c, s, o, \alpha)$) si elle est de la forme $A_i \rightarrow \phi_P(org, r, a, v, c, s, o, \alpha)$ (resp. $A_i \rightarrow \phi_I(org, r, a, v, c, s, o, \alpha)$) où
$A_i = \{Différent(s, i_s, \alpha, i_\alpha, o, i_o)\}$.

La contre-partie d'une sous-base cohérente dans le cas propositionnel devient une base de connaissances affaiblie.

Définition 1 *Une base $\Sigma' = S'_1 \cup \ldots \cup S'_n$ est dite une base affaiblie de $\Sigma = S_1 \cup \ldots \cup S_n$ si elle est obtenue à partir de Σ en remplaçant les formules ϕ_P (resp. ϕ_I) par les formules affaiblies ϕ_{Aff_P} (resp. ϕ_{Aff_I}).*

Définition 2 *Soit $\phi_{Aff_P} = A \rightarrow \phi_P$ (resp. $\phi_{Aff_I} = A \rightarrow \phi_I$) une formule affaiblie de ϕ_P (resp. ϕ_I).*
On définit le degré de ϕ_{Aff_P}, noté $degré(\phi_{Aff_P})$, comme la cardinalité de A, c'est-à-dire :
$degré(\phi_{Aff_P}) = |A|$.

Définition 3 *Soit $\Sigma' = S'_1 \cup \ldots \cup S'_n$ la base affaiblie de $\Sigma = S_1 \cup \ldots \cup S_n$.*
On définit le degré de S'_i, $degré(S'_i)$, par :

$$degré(S'_i) = \sum_{\phi_{Aff_P} \in S'_i} degré(\phi_{Aff_P}) + \sum_{\phi_{Aff_I} \in S'_i} degré(\phi_{Aff_I})$$

L'extension de l'ordre lexicographique est définie par :

Définition 4 *Soit $\Sigma' = S'_1 \cup \ldots \cup S'_n$ et $\Sigma'' = S''_1 \cup \ldots \cup S''_n$ deux bases cohérentes affaiblies de Σ.*
Σ' est dite $1\_Lex\ préférée$ à Σ'', noté $\Sigma' >_{1\_Lex} \Sigma''$, si $\exists i, 1 \leq i \leq n$ tel que :

- $|degré(S'_i)| < |degré(S''_i)|$,
- $\forall j > i, |degré(S'_i)| = |degré(S''_i)|$.

C'est-à-dire, une base est dite $1\_Lex\ préférée$ si on a affaibli le moins de formules possibles.

Finalement, nous définissons les bases préférées ainsi que l'inférence.

Définition 5 *Soit Σ' une base affaiblie cohérente de Σ.*
Σ' est dite $1\_Lex\ préféré$ de Σ, s'il n'existe aucune Σ'' cohérente et affaiblie de Σ tel que : $\Sigma' >_{1\_Lex} \Sigma''$.
Une formule ψ est une $1\_Lex\ préférée$ de Σ, si elle est conséquence de toutes les bases affaiblies cohérentes 1_Lex préférées de Σ.

Appliquons maintenant cette approche aux exemples 5 et 6.

Exemple 5 (suite) *Dans l'exemple 5, la base affaiblie Σ' de Σ est obtenue en remplaçant la formule ϕ_P de R_{12} par sa formule affaiblie ϕ_{Aff_P}.*

et R_{12} devient alors :
$\phi_{Aff_P} = Différent(s, Jean, \alpha, lire, o, dossier\_médical\_JO) \rightarrow$
$\phi_P(org, r, a, v, médecin\_traitant, s, o, \alpha)$.
$degré(\phi_{Aff_P}) = 1$.

$\Sigma' = S'_3 \cup S'_2 \cup S'_1$, *est tel que :*
$S'_3 = S_3$; $S'_2 = S_2$ *et* $S'_1 = \{\phi_{Aff_P}, R'_{12}\}$.
$degré(S'_1) = 1$.
Σ' est cohérente car les seules instances responsables du conflit dans Σ sont enlevées par la formule ϕ_{Aff_P}.
On peut montrer que Σ' est la seule base $1\_Lex\ préféré$ de Σ.
Nous pouvons maintenant vérifier que nous obtenons le résultat
$\Sigma' \vdash Est\_permis(Chris, lire, dossier\_médical\_JO)$.

Exemple 6 (suite) *Reprenons maintenant l'exemple 6. La base affaiblie Σ' de Σ est obtenue en remplaçant la formule ϕ_P de R_{12} et la formule ϕ_I de R''_{13} par leurs formules affaiblies ϕ_{Aff_P} et ϕ_{Aff_I}, avec*

$\phi_{Aff_P} = Différent(s, Jean, \alpha, lire, o, dossier\_médical\_JO) \rightarrow$
$\phi_P(org, r, a, v, médecin\_traitant, s, o, \alpha)$.
$\phi_{Aff_I} = Différent(s, Chris, \alpha, lire, o, dossier\_médical\_JO) \rightarrow \phi_I(org, r, a, v, cas\_général, s, o, \alpha)$,

$\Sigma' = S'_4 \cup S'_3 \cup S'_2 \cup S'_1$, *est tel que :*
$S'_4 = S_4$; $S'_3 = S_3$; $S'_2 = \{\phi_P, R'_{12}, R''_{12}\}$ *et* $S'_1 = \{R_{13}, \phi_{Aff_I}\}$.

$degré(S'_1) = 1$. $degré(S'_2) = 1$
Σ' *est cohérente et est* $1\_Lex\ preféré$ *de* Σ.
A partir de Σ'*, on obtient les résultats désirables :*
$\Sigma' \vdash Est\_permis(Chris, lire, dossier\_médical\_JO)$,
et
$\Sigma' \vdash Est\_interdit(Jean, lire, dossier\_médical\_JO)$.

7 Conclusion

Ce papier a traité le problème de la gestion des conflits dans le système OrBAC, modélisé formellement par une base de connaissances de la logique du premier ordre.
Nous avons montré que les approches de la gestion des incohérences proposées pour les bases de connaissances propositionnelles ne peuvent pas être appliquées dans le cas du premier ordre. C'est le cas par exemple de l'approche dite par cardinalité qui souffre non seulement du problème de la noyade mais peut aussi aboutir à des résultats indésirables dans le cas du premier ordre. Cette limitation est du fait que la suppression d'une formule du premier ordre conduit à la suppression des instances ce cette formule qui ne sont pas responsables du conflit.
Nous avons proposé une solution basée sur l'affaiblissement d'une formule incertaine, concernant le passage des privilèges abstraites aux privilèges concrètes. L'affaiblissement consiste à restreindre l'application d'une formule aux seules instances non responsables du conflit. Ceci a permis de résoudre les problèmes des approches dites par cohérence dans le cas des bases du premier ordre.

8 Remerciements

Ce travail est soutenu par le projet national RNRT MPSSICSS. Les auteurs remercient A.Abou El Kalam, P.Balbiani, F.Cuppens, Y.Deswarte, A.Miège, C. Saurel, G.Trouessin et D.Vinsonneau pour leurs commentaires.

References

[1] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miège, C. Saurel, and G. Trouessin. Organization based access control. In *Policy'03*, pages 120–131, 2003.

[2] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *IJCAI'93*, pages 640–645, 1993.

[3] S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artificial Intelligence Journal*, 1997.

[4] G. Brewka. Preferred Subtheories: an extended logical framework for default reasoning. In *IJCAI'89*, pages 1043–1048, 1989.

[5] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In *Handbook of Logic in Artifical Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[6] P. Gärdenfors. Knowledge in Flux - Modeling the Dynamic of Epistemic States. In *The MIT Press*, MIT Press, Cambridge, MA, 1988.

[7] C.K. Georgiadis, I. Mavridis, G. Pangalos, and R.K. Thomas. Flexible Team-Based Access Control Using Contexts. In *SACMAT'01*, Chantilly, Virginia, USA, May 3-4 2001.

[8] J. Lang. Possibilistic logic: complexity and algorithms . In *Algorithms for Uncertainty and Defeasible Reasoning,Eds: Kluwer Academic Publishers , Dordrecht, The Netherlands*, pages 179–220, 2001.

[9] D. Lehmann. Another perspective on default reasoning. In *Annals of Mathematics and Artificial Intelligence*, volume 15(1), 1995.

[10] B. Nebel. Base revision operator and schemes: semantics representation and complexity. In *11th European Conference on Artifical Intelligence*, pages 341–345, Amesterdam, 1994.

[11] J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. In *TARK'90*, 1990.

[12] N. Rescher and R. Manor. On inference from inconsistent premises. In *Theory and Decision 1*, 1970.

[13] R. Sandhu. Role-Based Access Control. In Academic Press, editor, *Advances in Computers*, volume 46, 1998.

[14] W. Wilikens, S. Feriti, and M. Masera. A context-related authorization access control method based on RBAC : a case study from the healthcare domain. In *SACMAT'02*, June 3-4 2002.

[15] M. Williams. A practical approach to belief revision: reason-based change. In *KR'96*, pages 412–421, 1996.