DÉveloppement de Systèmes Informatiques par Raffinement des contraintes Sécuritaires

Projet DESIRS

PaRISTIC

Bordeaux 21-22-23 Novembre 2005

Chantiers du projet

1. **Modélisation incrémentale d'un système par raffinement successif (B événementiel)**

2. **Modélisation par raffinement en logiques non-classiques (B non-classique)**

3. **Formalisation des politiques de sécurité en logiques non-classiques (contrôle d'accès)**

4. **Etudes de cas et outils**

Partenaires du projet

- ◇ **Centre de Recherche en Informatique de Lens CRIL (Salem Benferhat)**
- ◇ **ENST Bretagne, Rennes, Laboratoire RS (Frédéric Cuppens)**
- ◇ **IRIT Equipe LILAC (Philippe Balbiani)**
- ◇ **LISI/ENSMA Université de Poitiers (Yamine Ait-Ameur)**
- ◇ **LORIA UMR 7503 Equipe MOSEL (Dominique Méry)**

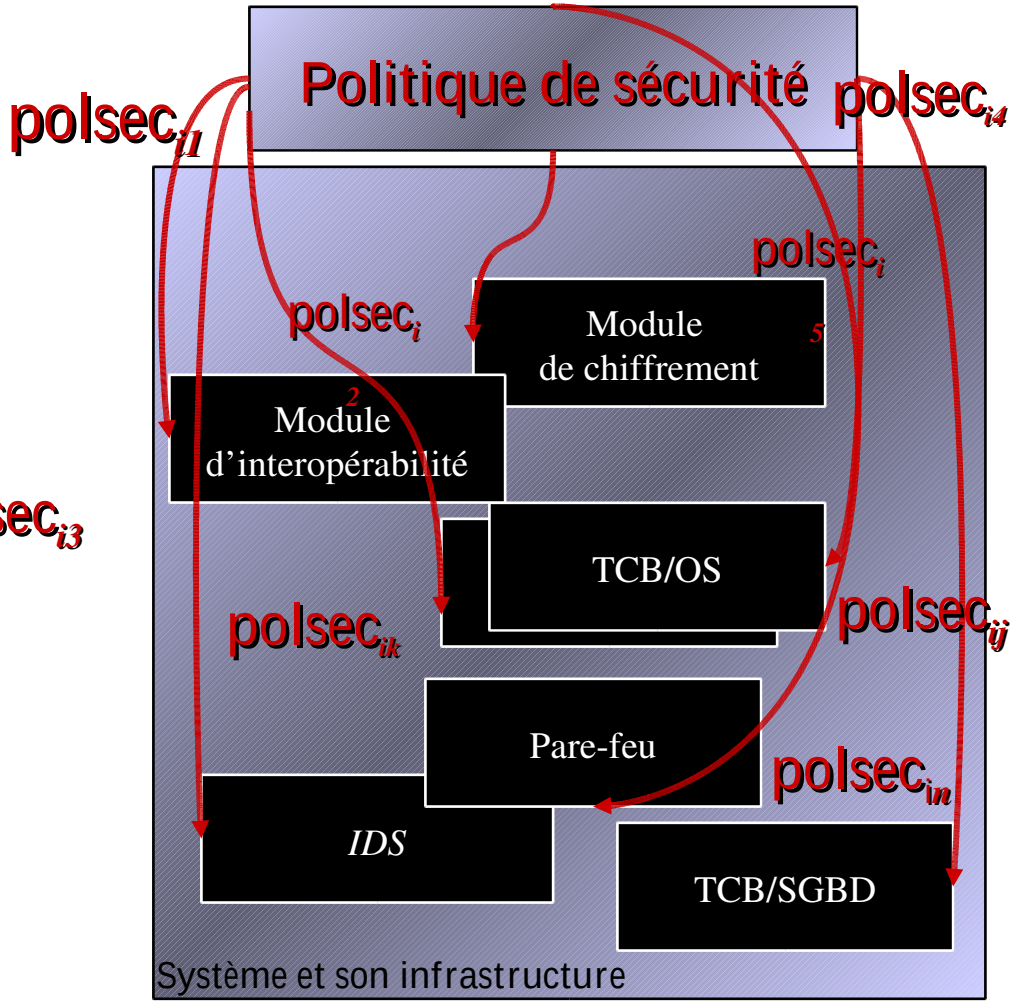
Problématique

- ◇ **Intégration d'une politique de sécurité donnée au sein d'un système à développer ou déjà développé.**
- ◇ **Intégration fondée sur le raffinement.**
- ◇ **Structuration des modèles OrBaC**
- ◇ **Question sur l'analyse de la cohérence de la politique de sécurité**
- ◇ **Question sur le lien entre les modèles ORBAC, B**
- ◇ **Contrôle d'accès et contrôle de flux**

Objectif : Une approche globale

- Définition de politiques de sécurité globales des systèmes d'information (SI)

- Processus de raffinement/décomposition/ affectation
  - Configurer la politique de sécurité déléguée à chacun des composants d'une architecture de sécurité mise en place par les SI

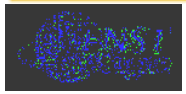


Approche descendante

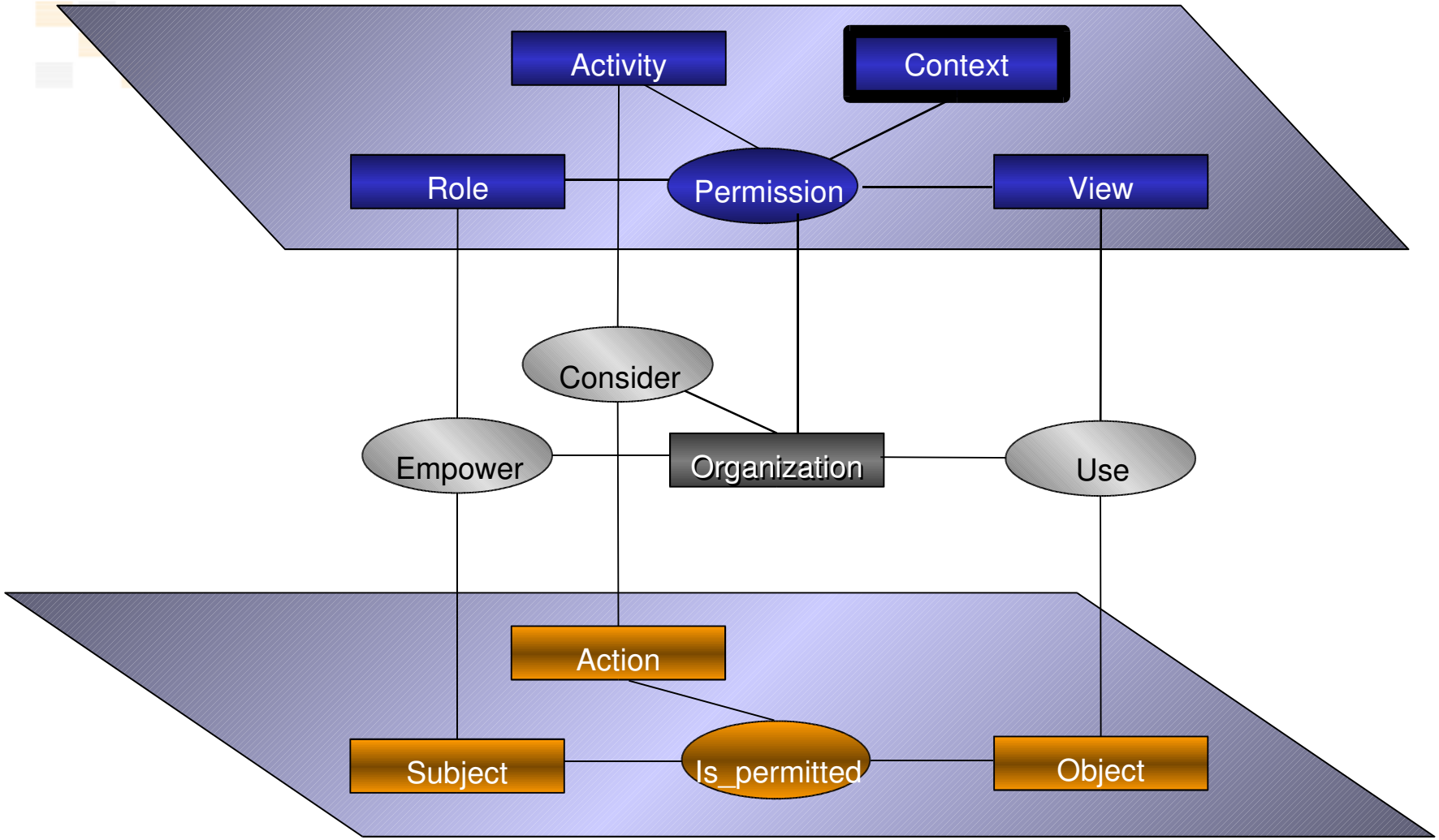
- Basée sur le modèle Or-BAC
  - Or-BAC : Organization Based Access Control

- Or-BAC
  - Modèle centré sur le concept d'organisation
  - Abstractions Sujet → Role
    Action → Activité
    Objet → Vue
  - Permission, Interdiction, Obligation

- Politique de contrôle d'accès et d'usage
  - Application de Or-BAC aux politiques de sécurité réseau
    - ➔ Configuration de pare-feu et d'IDS



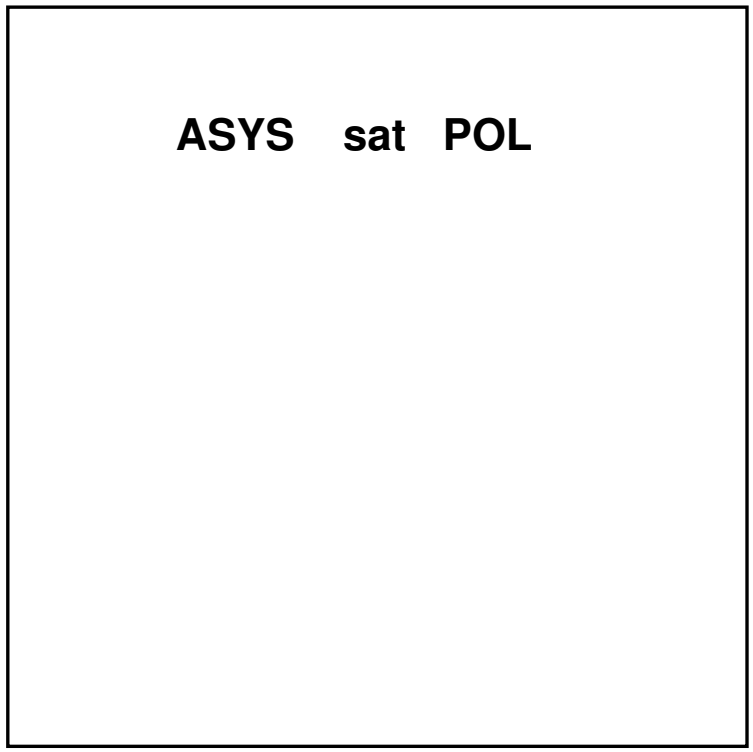
Le modèle Or-BAC



Contrôle d’accès

- ◇ **Politique de sécurité pour les permissions et les interdictions.**

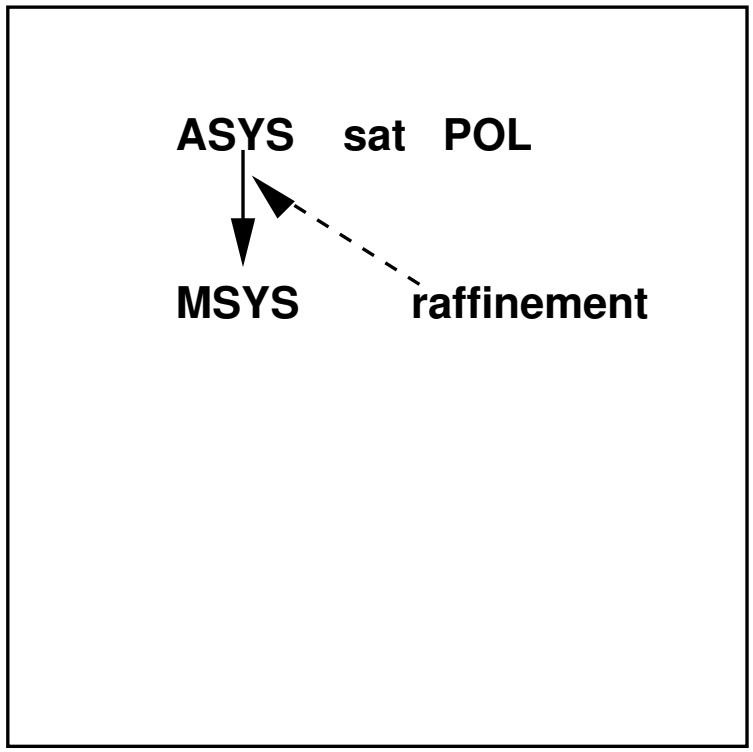
- ◇ **Mise en œuvre de cette politique de sécurité dans le cadre d’un système.**



Contrôle d'accès

- ◇ **Politique de sécurité pour les permissions et les interdictions.**

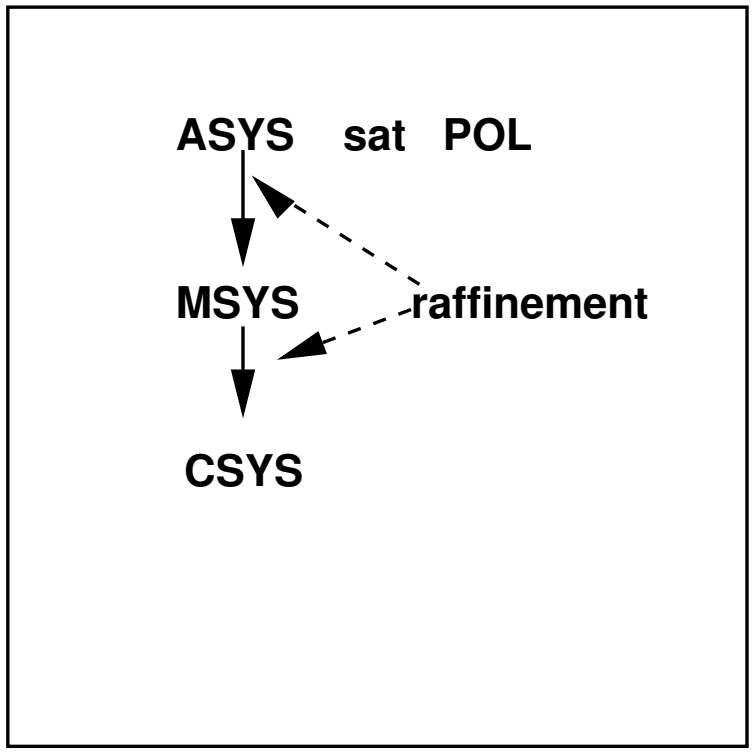
- ◇ **Mise en œuvre de cette politique de sécurité dans le cadre d'un système.**



Contrôle d'accès

- ◇ **Politique de sécurité pour les permissions et les interdictions.**

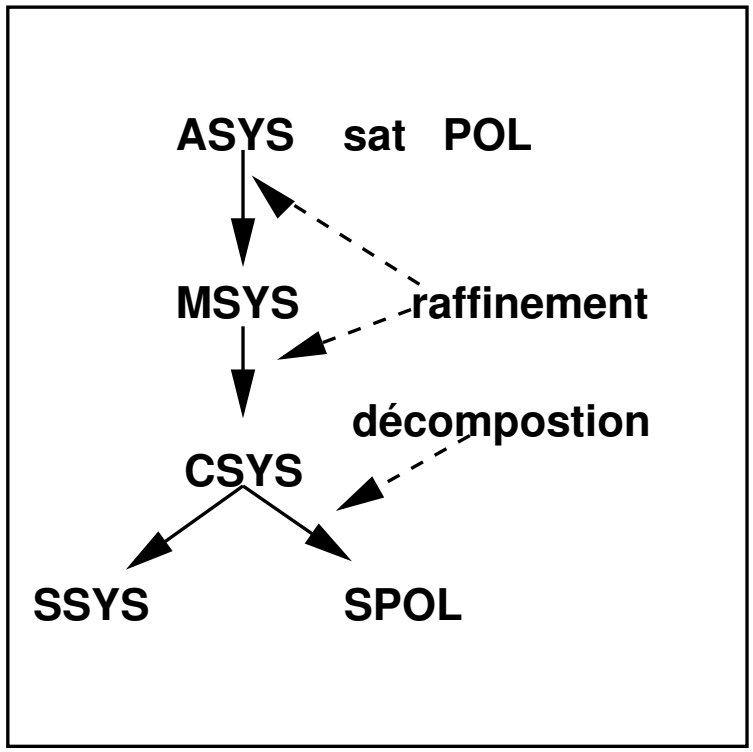
- ◇ **Mise en œuvre de cette politique de sécurité dans le cadre d'un système.**



Contrôle d’accès

- Politique de sécurité pour les permissions et les interdictions.

- Mise en œuvre de cette politique de sécurité dans le cadre d’un système.



Problème des conflits

- ORBAC : Modèle expressif
  - Contextes
  - Organisations
  - Permission / Interdiction sont inclues.
- Permissions positives
  - Pas de conflits
  - Logique classique suffit
- Permissions + Interdictions
  - Risque de conflicts
  - Besoin de logiques non-classiques

Un exemple de politique de sécurité

- ◇ **R1. Le personnel a la permission de modifier (écrire) les dossiers administratifs des patients.**
- ◇ **R2. Il est interdit aux médecins de modifier les dossiers administratifs des patients.**
- ◇ **R3. Un médecin est un membre du personnel.**
- ◇ **R4. Une secrétaire est un membre du personnel.**
- ◇ **R5. Tout utilisateur autorisé à modifier le dossier administratif de JO est également autorisé à le lire.**

Supposons que dans un hôpital A nous avons: ,

- **un patient ("JO"),**
- **deux personnels : un médecin ("Bob") et une secrétaire ("Mary")**
- **un objet : le dossier administratif de JO.**

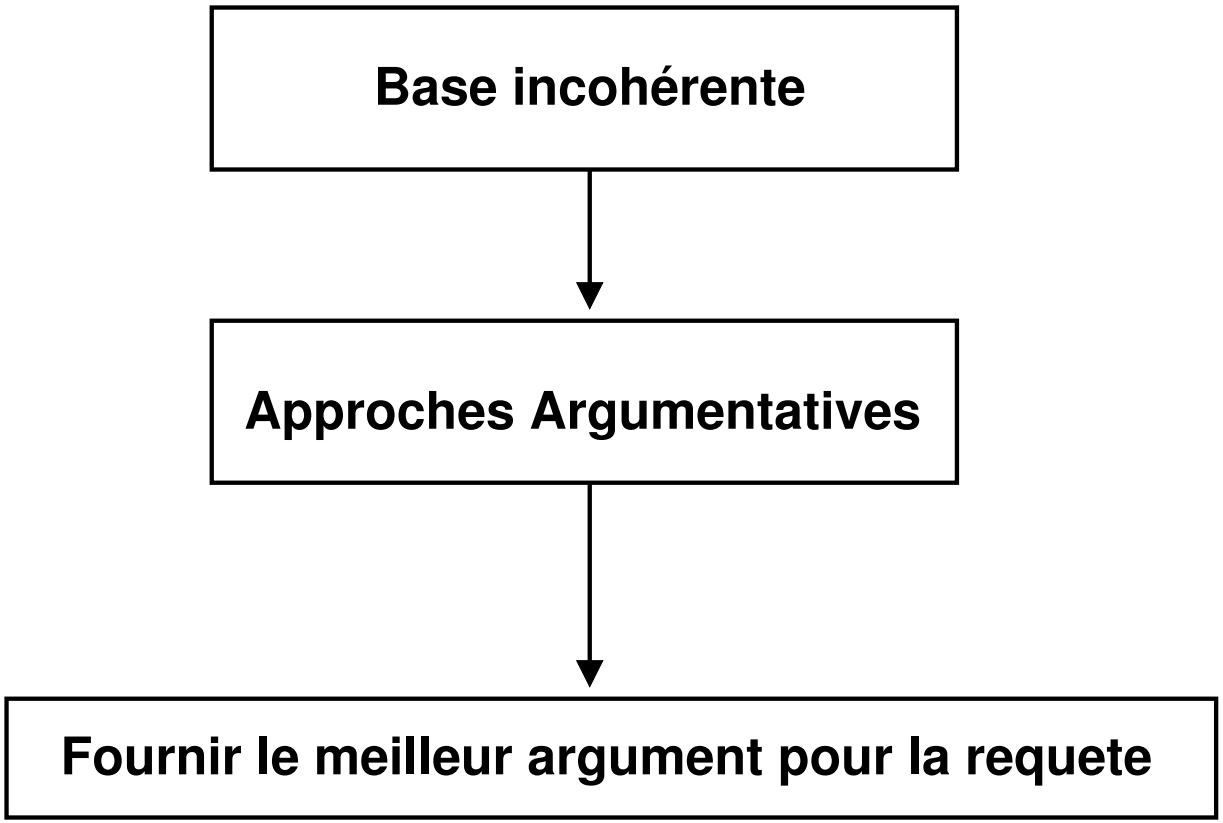
Un exemple de politique de sécurité

- **R1. Le personnel a la permission de modifier les dossiers administratifs des patients.**
- **R2. Il est interdit aux médecins de modifier les dossiers administratifs des patients.**
- **R3. Un médecin est un membre du personnel.**
- **R4. Une secrétaire est un membre du personnel.**
- **R5. Tout utilisateur autorisé à modifier le dossier administratif de JO est également autorisé à le lire.**
- **F1 un médecin ("Bob")**
- **F2 une secrétaire ("Mary")**
- **F3 un objet : le dossier administratif de JO.**

Requête: Est-il permis à Bob de modifier le dossier administratif de JO?

Conflits: Interdiction (F1+R2), Permission(F1+R3+R1) !

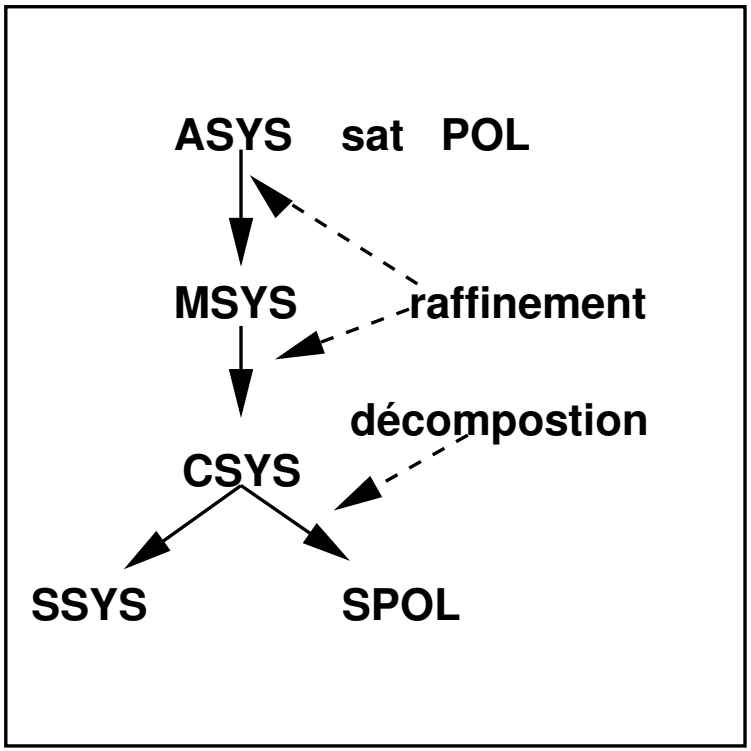
Traitement des incohérences: Approches argumentatives



Politique de sécurité composée avec un système événementiel via le raffinement

- Donner une relation entre un modèle OrBaC et un modèle événementiel
- Analyser la cohérence de la politique de sécurité par le raffinement
- Produire un modèle de référence OrBaC pour développer ultérieurement le modèle B du système conforme à la politique de sécurité explicitée

Approche système du raffinement



Problème du bureau payeur

- **Problème d'un bureau payeur (sécurité sociale par exemple)**

- **Lorsqu'un malade transmet sa requète en vue d'un remboursement, son dossier est d'abord traité par un agent administratif puis validé par le chef de service et finalement un chèque à l'intention du malade est émit par le comptable :**

  1. **Traitement du dossier par un agent administratif.**
  2. **Validation par le chef de service.**
  3. **Emission d'un chèque par le comptable.**

Problème du bureau payeur

- **un dossier peut être traité par le chef de service ou le comptable, en revanche la validation et l'émission des chèques sont exclusivement du ressort du chef de service et du comptable respectivement.**

- **un problème workflow car l'ordre d'exécution des tâches est important**

- **Les contraintes workflow décrites en ORBAC via les contextes seront exprimées dans le modèle événementiel via les variables d'état.**

Problème du bureau payeur

En plus des fichiers malades, la base de données du bureau comporte également les fichiers du personnels ainsi que ceux de la comptabilité. L'accès à ces fichiers est régi par les règles suivantes:

1. **Les fichiers du personnel peuvent être consultés par le chef de service et le comptable.**

2. **Les fichiers de comptabilité peuvent être consultés par le chef de service et le comptable, mais ne peuvent être modifiés que par ce dernier.**

Problème du bureau payeur

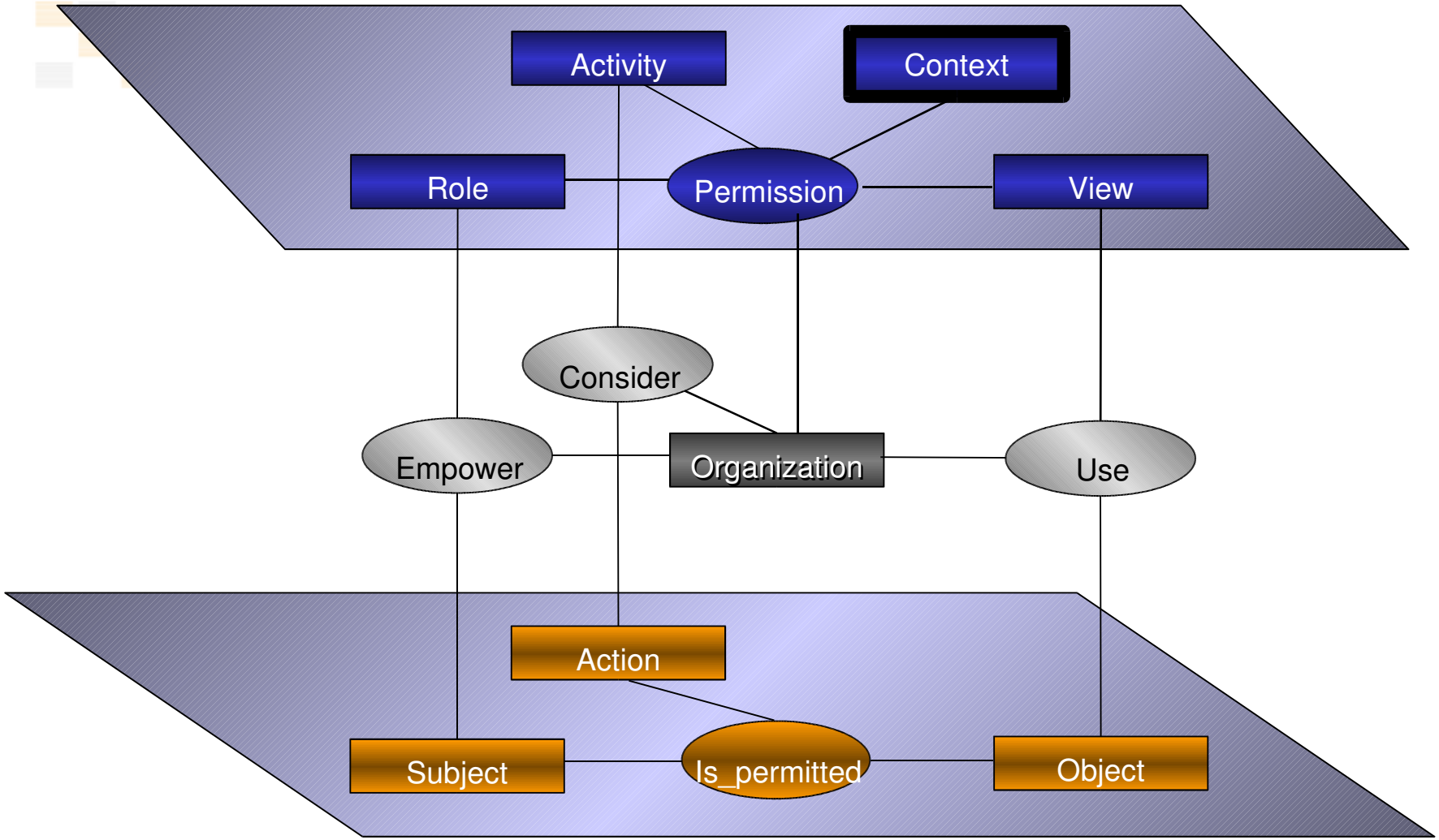
- **Garantir la séparation des pouvoirs dans le processus du traitement des dossiers, la validation et l'émission des chèques ne doivent être effectuée par la même personne même si celle-ci cumule les fonctions de chef de service et de comptable.**

- **Contraintes:**

  1. **Le comptable doit veiller à mettre à jour les fichiers de comptabilité après l'émission de chaque chèque (–>obligation).**

  2. **A l'issue de la validation un dossier peut être rejeté, dans ce cas le chèque n'est évidemment pas émis.**

Le modèle Or-BAC



Vue ORBAC

```
ROLES={agt_admin,ch_serv,compt}
ACTIONS={consult,traiter,valider,emettre,modifier}
VIEWS={f_malade,f_compt,f_personnel,cheques}

OBJECTS={fm1,fm2,fm3,fp1,fp2,fc,cheque}
SUBJECTS={emp1,emp2,emp3,chef_service,comptable}
```

Vue ORBAC

```
use(f_malade,fm1)
use(f_malade,fm2)
use(f_malade,fm3)
use(f_malade,fm1)
use(f_personnel,fp1)
use(f_personnel,fp2)
use(f_compt,fc)
use(cheques,cheque)

empower(agt_admin,emp1)
empower(agt_admin,emp2)
empower(agt_admin,emp3)
empower(ch_serv,chef_service)
empower(compt,comptable)
```

Vue ORBAC

Les notions d'activité et d'action se confondent à cause de la simplicité de l'exemple.

```
// Les permissions
        permission(bureau,agt_admin,traiter,f_malade)
        permission(bureau,ch_serv,valider,f_malade)
        permission(bureau,ch_serv,consult,f_personnel)
        permission(bureau,ch_serv,consult,f_compt)
        permission(bureau,compt,emettre,cheques)
        permission(bureau,compt,consult,f_compt)
        permission(bureau,compt,modifier,f_compt)
// La hiérarchie des rôles :
        specialized_role(compt,agt_admin)
        senior_role(ch_serv,agt_admin)
```

Vue ORBAC

- **Le rôle comptable est un sous rôle de agent administratif de type specialized rôle**
- **il héritera donc de ses permissions et de ses interdictions.**
- **Par contre le rôle chef de service est un sous rôle de agent administratif de type senior rôle, il n'héritera par conséquent que de ses permissions.**

Approche proposée

◇ **L'approche proposée ici pour le passage vers le modèle événementiel est composée de deux étapes.**

1. **La première étape consiste à obtenir un modèle abstrait concernant uniquement les organisations, rôles, actions et vues.**

2. **La seconde étape sera un raffinement de la première dans laquelle les notions de subject et objects seront introduites.**

Modèle événementiel

MODEL
m
SETS
s
CONSTANTS
c
PROPERTIES
$P(s, c)$
VARIABLES
x
INVARIANT
$I(x)$
SAFETY
$A(x)$
INITIALISATION
$x : INIT(x)$
EVENTS
$e_1, \ldots, e_n$
END

- ☑ **Un modèle a un nom name m**
- ☑ **Un modèle utilise une théorie définie par les clauses SETS, CONSTANTS et PROPERTIES: politique de sécurité**
- ☑ **L'invariant $I(x)$ est maintenu par les événements**

Evénements

Evénement : E	Before-After Predicate
BEGIN $x : P(x_0, x)$ **END**	$P(x, x')$
SELECT $G(x)$ **THEN** $x : P(x_0, x)$ **END**	$G(x) \quad \wedge \quad P(x, x')$
ANY t **WHERE** $G(t, x)$ **THEN** $x : P(x_0, x, t)$ **END**	$\exists\, t \cdot (\, G(t, x) \quad \wedge \quad P(x, x', t)\,)$

Conditions de vérification pour un modèle

	Condition
(INV1)	$\Gamma(s,c) \vdash Init(x) \Rightarrow I(x)$
(INV2)	$\Gamma(s,c) \vdash I(x) \land BA(e)(x,x') \Rightarrow I(x')$
(DEAD)	$\Gamma(s,c) \vdash I(x) \Rightarrow (\mathsf{grd}(e_1) \lor \ldots \mathsf{grd}(e_n))$
(SAFE)	$\Gamma(s,c) \vdash I(x) \Rightarrow A(x)$
(FIS)	$\Gamma(s,c) \vdash I(x) \land \mathsf{grd}\,(E) \Rightarrow \exists x' \cdot P(x,x')$

Première étape: obtenir un modèle abstrait décrivant les permissions et les interdictions

- **Les actions doivent elles se dérouler dans le bon ordre et respecter les différentes contraintes du problème (variable $historique$).**

- **Cohérence du modèle: L'héritage des permissions et interdictions entres rôles peut générer des conflits: r1 est un sous rôle de r2**

```
interdiction (o,r2,a,v)=> interdiction (o,r1,a,v)
```

- **Si r1 dispose d'une permission : permission (o,r1,a,v), r1 ne pourra pas en profiter à cause de l'interdiction hérités de r2.**

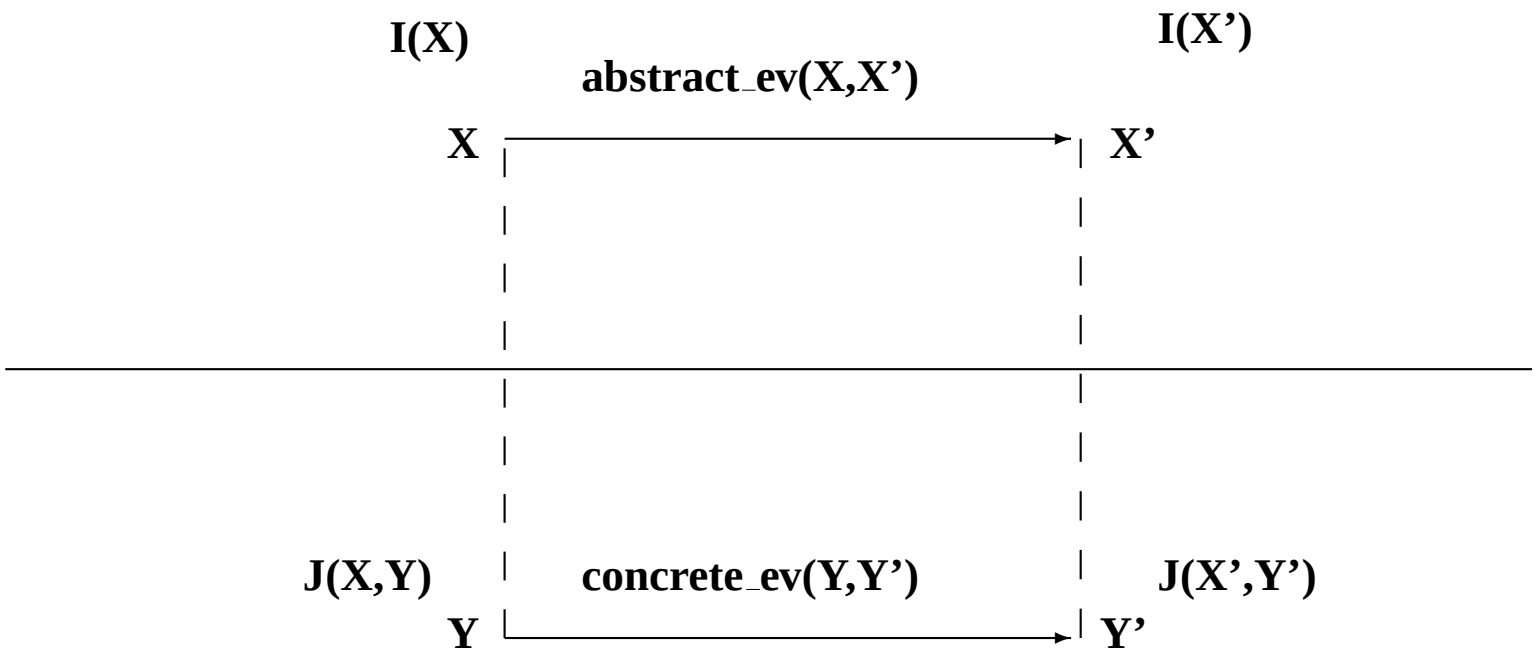
- **olution possible: privilégier les permissions sur les interdictions, mais la permission peut être héritée:**

```
permission (o,r2,a,v)=> permission (o,r1,a,v)
```

Refinement

REFINEMENT r
REFINES m
SETS t
CONSTANTS d
PROPERTIES $Q(t, d)$
VARIABLES y
INVARIANT
$J(x, y)$
SAFETY
$B(y)$
INITIALISATION
$y : INITC(y)$
EVENTS
$e_1, \ldots, e_n$
END

Schéma du raffinement



Conditions de vérification pour le raffinement

(REF1) $\textsc{Initc}(y) \Rightarrow \exists x \cdot (\textsc{Init}(x) \land \mathtt{J}(x,y))$:

Conditions initiales

(REF2) $\mathtt{I}(x) \land \mathtt{J}(x,y) \land BAC(y,y') \Rightarrow \exists x'.(BAA(x,x') \land \mathtt{J}(x',y'))$:

Conditions pour un événement

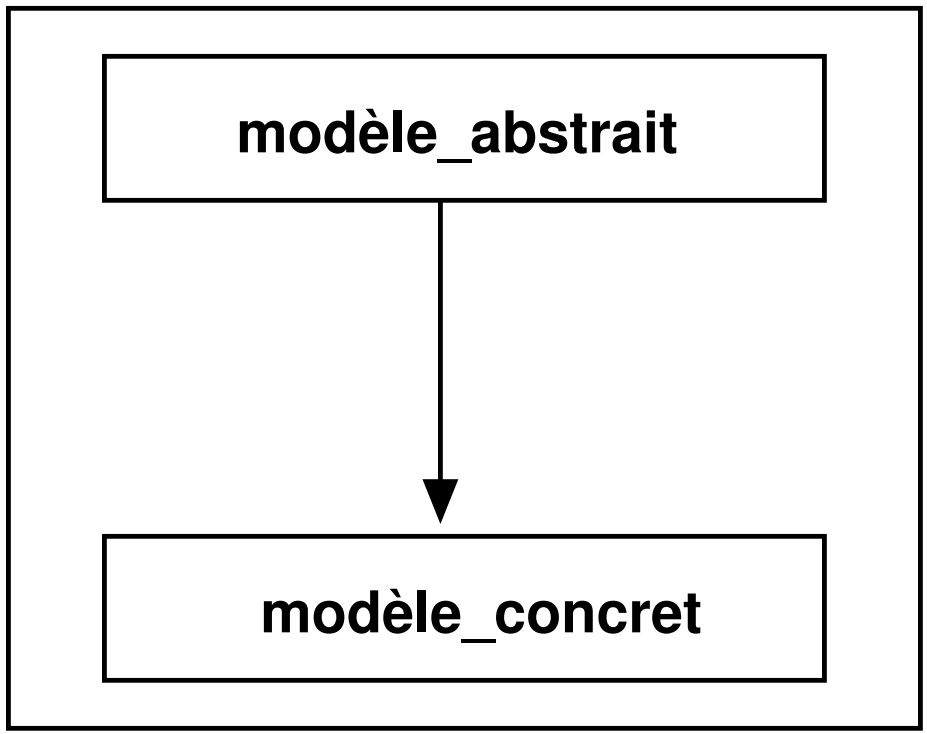
(REF3) $\mathtt{I}(x) \land \mathtt{J}(x,y) \land BAC(y,y') \Rightarrow \mathtt{J}(x,y')$:

Conditions pour un événement

Seconde étape: Obtenir le modèle abstrait global

- Fusion des modèles de rôles pour former un modèle abstrait de l'organisation.
- Raffinement du modèle abstrait pour introduire les notions de subject et object
- et pour exprimer les nombreuses contraintes relatives à la nature workflow du problème.
- Afin de pouvoir contrôler le flux d'exécution des différentes actions, une variable contenant l'historique des actions exécutée est utilisée.
- la variable "historique" dans le modèle abstrait puis la variable "histC" dans le modèle concret.
- Leur utilisation permet de contrôler si l'ordre d'exécution des actions est conforme aux problème tel qu'il a été posé.
- L'utilisation de ces variables ouvrent de nombreuses perspectives dans le sens ou elles permettent non seulement de contrôler l'ordre d'exécution des actions mais aussi de garantir la séparation des pouvoirs ou encore de déterminer les responsabilités en cas d'erreur..

Seconde étape: Modélisation



MODEL model_abstrait

```
SETS
ROLES={agt_admin,ch_serv,compt};
ACTIONS={consult,traiter,valider,emettre,modifier};
VIEWS={f_malade,f_compt,f_personnel,cheques}
CONSTANTS
permission
PROPERTIES
permission<:ROLES*ACTIONS*VIEWS    &
(agt_admin|->traiter|->f_malade):permission  &
(ch_serv|->valider|->f_malade):permission &
!(aa,vv).((aa:ACTIONS & vv:VIEWS & (agt_admin|->aa|->vv):permission)=>(ch_se
(ch_serv|->consult|->f_personnel):permission &
(ch_serv|->consult|->f_compt):permission  &
(compt|->emettre|->cheques):permission &
!(aa,vv).((aa:ACTIONS & vv:VIEWS & (agt_admin|->aa|->vv):permission)=>(compt
(compt|->consult|->f_compt):permission &
(compt|->modifier|->f_compt):permission &
! Perm. ((Perm <: ROLES*ACTIONS*VIEWS  &
(agt_admin|->traiter|->f_malade):Perm &
(ch_serv|->valider|->f_malade):Perm &
!(aa,vv).((aa:ACTIONS & vv:VIEWS & (agt_admin|->aa|->vv):Perm)=>(ch_serv|->a
(compt|->emettre|->cheques):Perm &
!(aa,vv).((aa:ACTIONS & vv:VIEWS & (agt_admin|->aa|->vv):Perm)=>(compt|->aa|
     )=> permission<:Perm)
```

MODEL model_abstrait

VARIABLES

$historique$

INVARIANT

$historique \subseteq ROLES \times ACTIONS \times VIEWS \wedge$

$historique \subset permission \wedge$

$\forall (r, v).((r \in ROLES \wedge v \in VIEWS \wedge (r \mapsto valider \mapsto v) \in historique)$

$\Rightarrow (\exists r2.(r2 \in ROLES \wedge (r2 \mapsto traiter \mapsto v) \in historique))) \wedge$

$\forall (r, v).((r \in ROLES \wedge v \in VIEWS \wedge (r \mapsto emettre \mapsto v) : historique)$

$\Rightarrow (\exists r2.(r2 \in ROLES \wedge (r2 \mapsto valider \mapsto v) \in historique)))$

événement action

action $=$

ANY r, v, a **WHERE**

$r \in ROLES \wedge$

$v \in VIEWS \wedge$

$a \in ACTIONS \wedge$

$(r \mapsto a \mapsto v) \in permission \wedge$

$a \neq traiter \wedge a \neq emettre \wedge a \neq valider$

THEN

$historique := historique \cup \{(r \mapsto a \mapsto v)\}$

END

événement traiter

traiter $=$

ANY r, v, a **WHERE**

$r \in ROLES \wedge$

$v \in VIEWS \wedge$

$a \in ACTIONS \wedge$

$(r \mapsto a \mapsto v) \in permission \wedge$

$a = traiter$

THEN

$historique := historique \cup \{(r \mapsto a \mapsto v)\}$

END

événement valider

valider $=$

ANY r, v, a **WHERE**

$r \in ROLES \wedge$

$v \in VIEWS \wedge$

$a \in ACTIONS \wedge$

$(r \mapsto a \mapsto v) \in permission \wedge$

$a = valider$

$\exists r2.(r2 \in ROLES \wedge (r2 \mapsto traiter \mapsto v) \in historique)$

THEN

$historique := historique \cup \{(r \mapsto a \mapsto v)\}$

END

événement emettre

emettre $=$

ANY r, v, a **WHERE**

$r \in ROLES \wedge$

$v \in VIEWS \wedge$

$a \in ACTIONS \wedge$

$(r \mapsto a \mapsto v) \in permission \wedge$

$a = emettre$

$\exists r2.(r2 \in ROLES \wedge (r2 \mapsto valider \mapsto v) \in historique)$

THEN

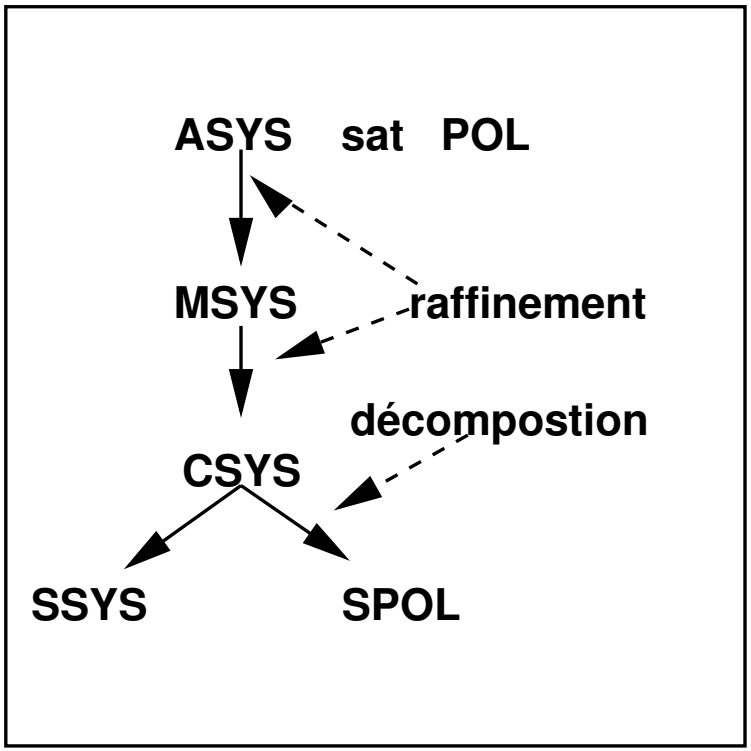
$historique := historique \cup \{(r \mapsto a \mapsto v)\}$

END

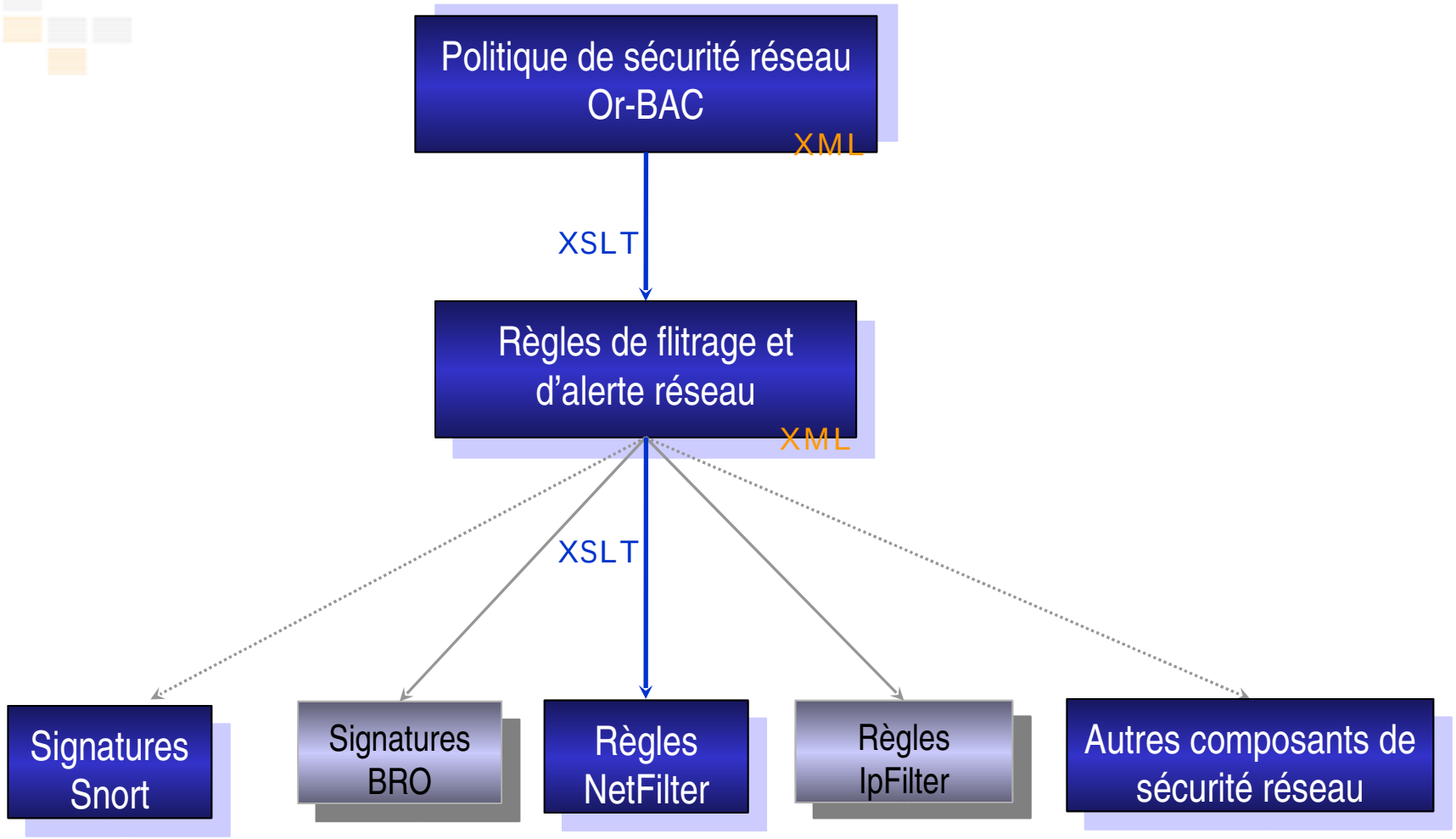
Que faire avec le modèle concret?

- **Le modèle concret constitue le modèle de la politique de sécurité dans le monde événementiel**
- **Le raffinement de modèle permet alors de concevoir un système conforme à cette politique de sécurité**
- **Est-ce que la politique de sécurité doit être vue globalement ou progressivement?**
- **Est-ce que nous avons une méthodologie?**
- **Extension du modèle OrBaC pour prendre en compte les obligations**
- **Applications aux workflows**

Approche système du raffinement



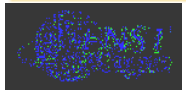
Approche descendante : application réseau



Approche descendante : démarche et résultats

- Comment obtenir la configuration d'un ensemble de composants conformément à la politique de sécurité réseau ?
  - Spécifier formellement la politique de sécurité réseau
  - Dériver la configuration des composants de cette spécification
  - Prendre en compte la topologie et l'organisation hiérarchique des composants

- Résultats
  - Approche globale basée sur Or-BAC
    - → MotOr-BAC : outil de gestion de politiques de sécurité Or-BAC
  - Configuration de firewalls et d'IDS
  - Reconfiguration dynamique par changement de contexte
    - → Réaction aux attaques



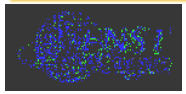
Approche ascendante

- Besoin de combiner l'approche descendante avec une approche ascendante
  - Analyser les configurations existantes

- Deux problèmes différents
  - Analyse intra composant
    - → Détecter les anomalies dans la configuration d'un composant
  - Analyse inter composant
    - → Détecter les conflits entre configurations de composants différents

- Objectif
  - Méthode générale pour analyser différents types de composants
  - Firewall, VPN, IDS, …

- Résultats
  - Algorithmes complets de détection d'anomalies intra et inter-firewall



Bilan intermédiaire

- Outil DIXIT: éditeur de diagrammes de prédicats constituant des abstractions de système (déposé et accessible http://dixit.loria.fr)
- Outil de gestion OrBaC: moteur d'inférences avec une interface pour l'administratic des politiques de sécurité.
- Publications
- Projets déposés

Publications

R. El Baida. *Gestion des incohérences dans les systÃĺmes de contrôle d'accès*. PhD thesis, Thèse de l'Université d'Artois.

P. . Balbiani. A uniform approach to modelling timed protection. In *IASTED International Conference on Artificial Intelligence and Applications (AIA 2005).*, 2005.

P. Balbiani. Constitution et développement d'une logique des modalités aléthiques, déontiques, dynamiques, et temporelles en vue de la formalisation du raisonnement sur les actions et sur les normes. In *Troisièmes journées francophones modèles formels de l'interaction (MFI'05).*, 2005.

P. Balbiani. A formal examination of roles and permissions in access control with hierarchy and separation. In *3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-05*, 2005.

P. Balbiani and F. Cheikh. Une approche uniforme de la modélisation des systèmes de protection temporisés. In *Formalisation des activités concurrentes (journées FAC'2005).*, 2005.

F. Cuppens, Nora Cuppens-Boulahia, and A. Miège. Héritage de privilèges dans le modèle or-bac : application dans un environnement réseau. In *Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC04)*, Rennes, Juin 2004.

F. Cuppens, Nora Cuppens-Boulahia, and A. Miège. Inheritance hierarchies in the or-bac model and application in a network environment. In *Foundations of Computer Security (FCS04)*, 2004.

Frédéric Cuppens, Nora Cuppens-Boulahia, Thierry Sans, and Alexandre Miège. Formal approach to specify and deploy a network security policy. In *Formal Aspects in Security and Trust (FAST 2004)*, 2004.

F. Cuppens, N. Cuppens-Boulahia. A Provably Secure Approach to Configure Network Security Components. 12th Workshop of HP OVUA. Porto, Portugal, Juillet 2005.

Publications

F. Cuppens, N. Cuppens -Boulahia, and J. García. Detection and Removal of Firewall Misconfiguration. IASTED International Conference on Communication, Network and Information Security (CNIS 2005). Phoenix, AZ, USA, Novembre 2005.

F. Cuppens, N. Cuppens -Boulahia, and J. García. "Misconfiguration Management of Network Security Components. 7th International Symposium on System and Information Security (SSI05), Sao Paolo, Brazil, Novembre 2005.

L. Fejoz, D. Méry, and S. Merz. The DIXIT tool. Technical report, LORIA, 2004. http://www.loria.fr/equipes

L. Fejoz, D. Méry, and S. Merz. Dixit: a graphical tool for predicate abstrcations. In *AVIS'05*, 2005.

Leonor Prensa-Nieto and Gilles Barthe. Formally verifying information flow type systems for concurrent and thread systems. In Michael Backes, David Basin, and Michael Waidner, editors, *2nd ACM Workshop on Formal Methods in Security Engineering : From Specifications to Code 2004 - FMSE'04, Washington D.C., Etats-Unis*, ACM SIGSAC, pages 13–22. ACM, Oct 2004.

Salem BENFERHAT and Rania EL BAIDA. Handling conflicts in access control models. Poster, Aout 2004. 16th European Conference on Artificial Intelligence (ECAI'04).

Salem BENFERHAT et Rania EL BAIDA. Gestion des conflits dans les systèmes de contrôles d'accès basés sur l'organisation (orbac). In *14ième congrès francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2004)*, pages 1185–1194, 2004.

Nathalie CHETCUTI-SPERANDIO et Fabio MASSACCI. Sémantique et raisonnement automatique pour une infrastructure à clés publiques. In *14ieme congrès francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2004)*, pages 1164–1174, Toulouse, Janvier 2004.

WANG Tao. Raisonnement automatique pour l'échange sécurisé d'informations. Dea, Université d'Artois, 2004.