Approche de spécification et validation formelles de politiques RBAC au niveau des processus métiers

Présenté par:

Salim CHEHIDA
Université d' ORAN1, Algérie

Plan

Modélisation des processus métiers

La description des processus métiers permet :

- La compréhension, la formalisation et la mise œuvre des SI
- L'adaptation et la maintenance des SI

Les standards de modélisation
(**UML2 Activity Diagram**, BPMN)

La sécurité

Aspect fondamental dans la description des processus métiers

Les langages de modélisation ne sont pas adaptés à la modélisation des aspects de sécurité

Le contrôle d'accès au SI

Qui, au sein d'un SI, peut exécuter quelles actions et sous quelles conditions

DAC, MAC, **RBAC**, BAC, PBAC

Norme ANSI / INCITS
Le plus répandu dans les Systèmes d'Information

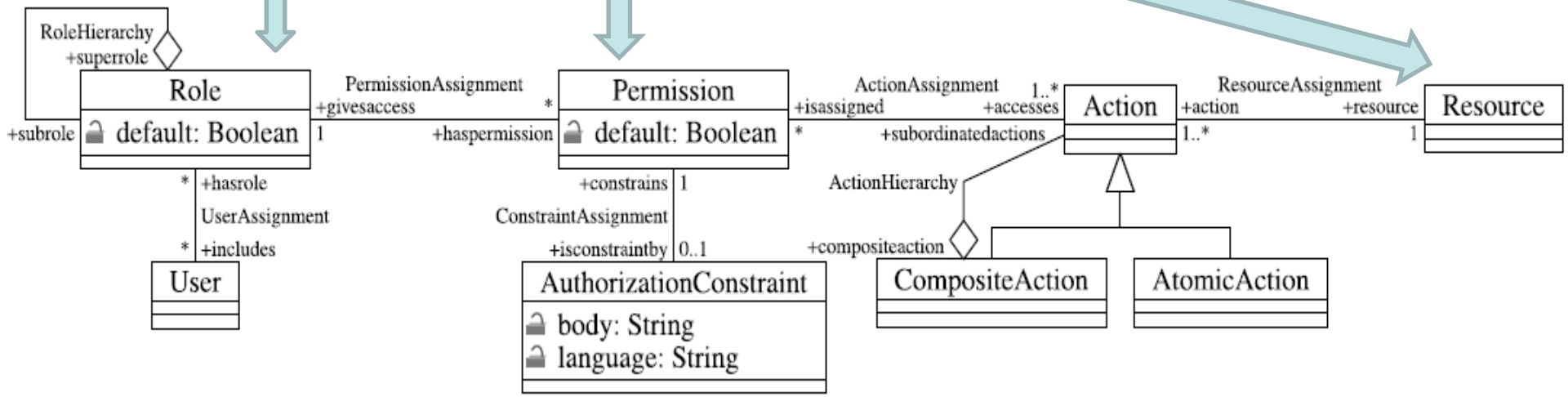
L'accès aux données est accordé à un utilisateur en fonction du **rôle** qui lui est associé

Le profil SecureUML

Méta-modèle pour la spécification de politiques RBAC sur les diagrammes d'UML

Extension du **Diagramme de Classes**

- «Role», «Permission», «Entity»

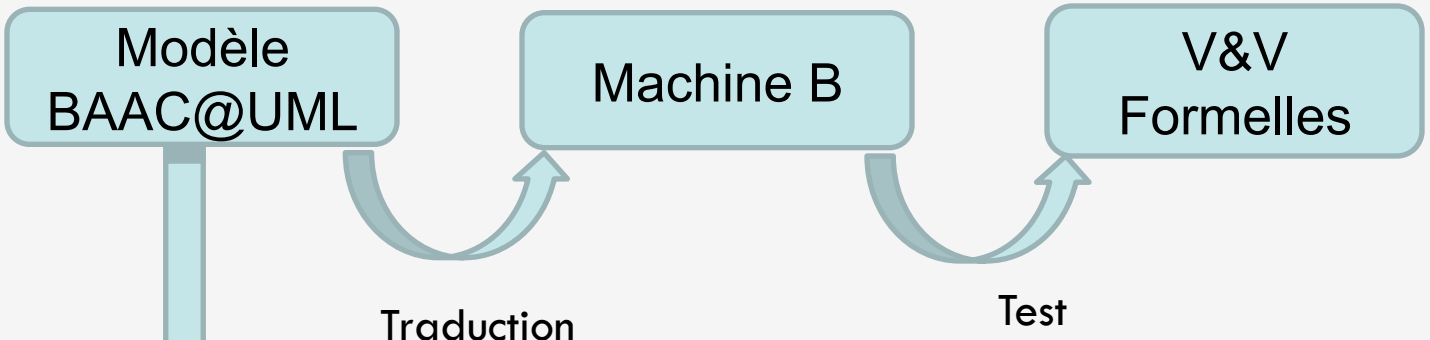


Métamodèle SecureUML [D.Basin et al]

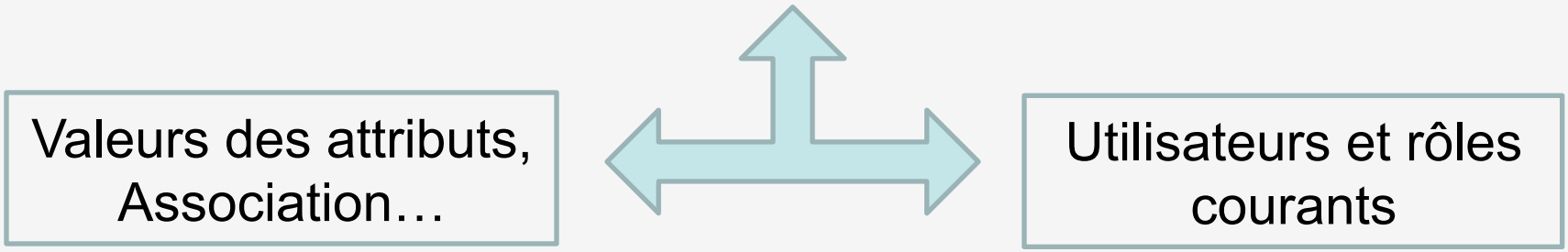
[Automated analysis of security-design models. IST(51, 5).(2009)]

Notre approche

- Extension du **diagramme d'activité** (Profil BAAC@UML) (Stéréotypes, Contraintes)



- **Rôle & Pré-conditions de Contrôle d'accès**



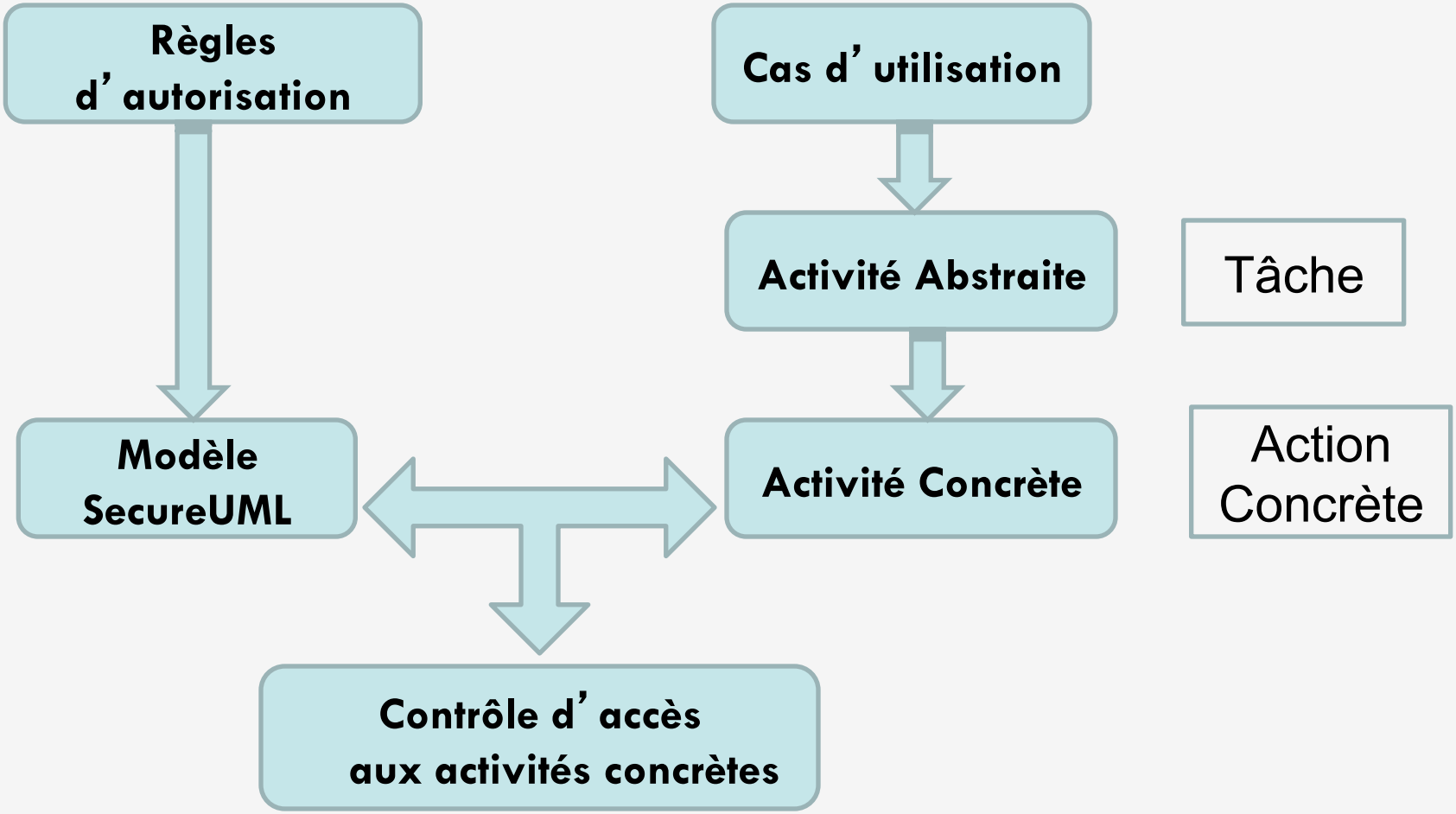
Exemple : Système d'organisation des réunions

Diagramme de cas d'utilisation

Règles d'autorisation:

- Seul l'initiateur peut créer des réunions
- Un initiateur peut lire ou modifier une réunion, pour autant qu'il soit le créateur de cette réunion
- Un participant peut lire les informations d'une réunion, pour autant qu'il fasse partie des personnes invitées à la réunion
- ……

Méthodologie



Modèle SecureUML

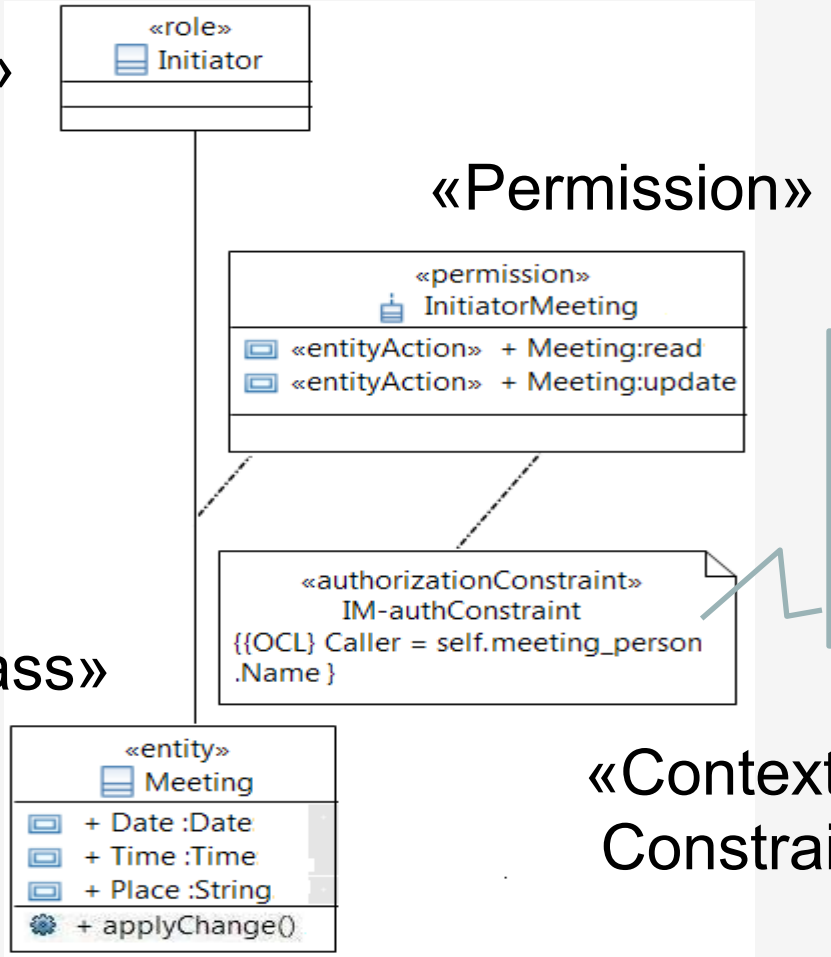
Modélisation de la vue statique de la politique RBAC

«Role»

«Permission»

«Protected Class»

«Contextual Constraint»



Un initiateur peut lire ou modifier une réunion, pour autant qu'il soit le créateur de cette réunion

Modèle BAAC@UML

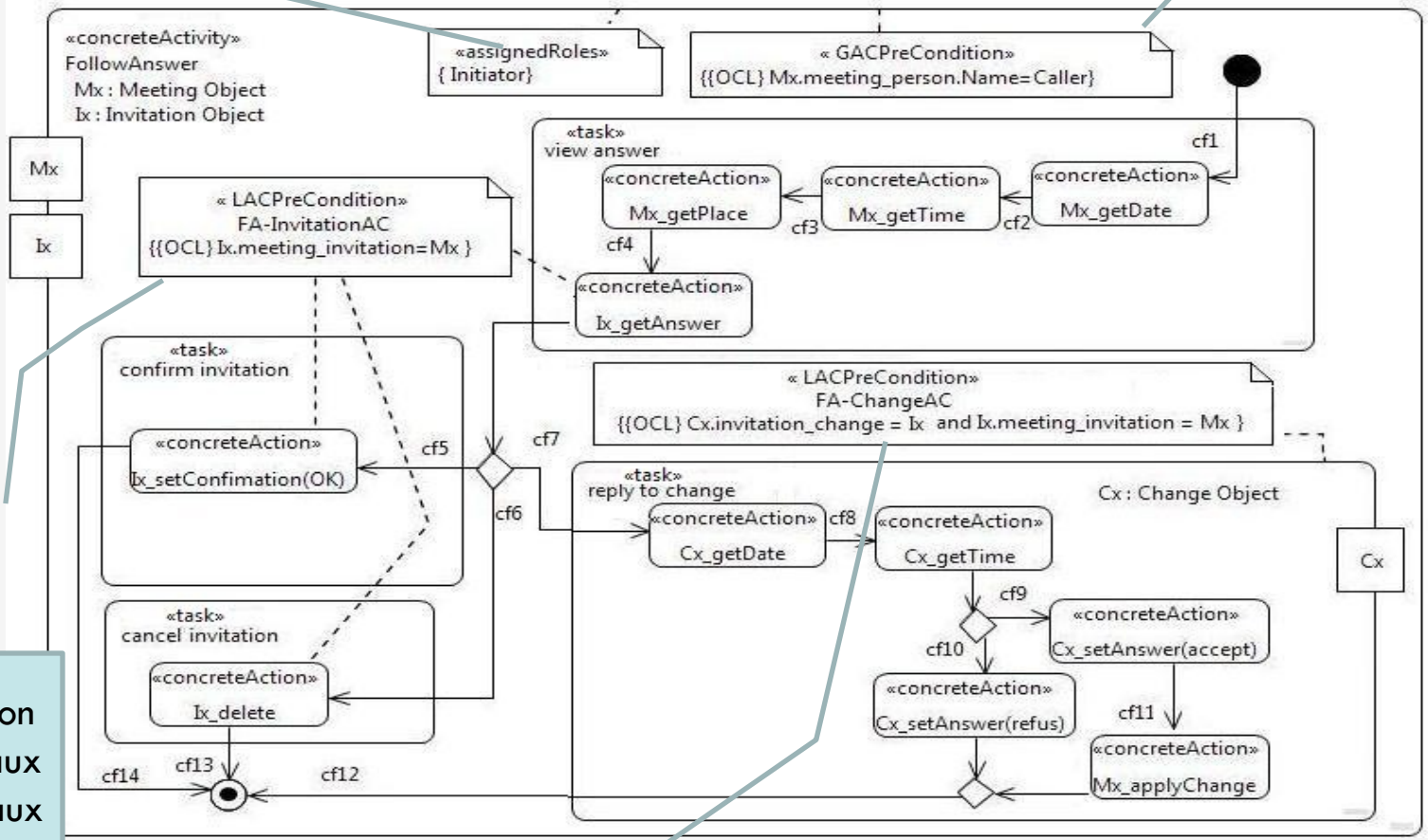
Contrôle d'accès aux activités d'un processus métier

L'utilisateur exécutant l'activité est le créateur de la réunion

Seuls les utilisateurs affectés à Initiator peuvent exécuter l'activité

L'initiateur de la réunion ne peut lire, modifier ou supprimer que les invitations de sa réunion

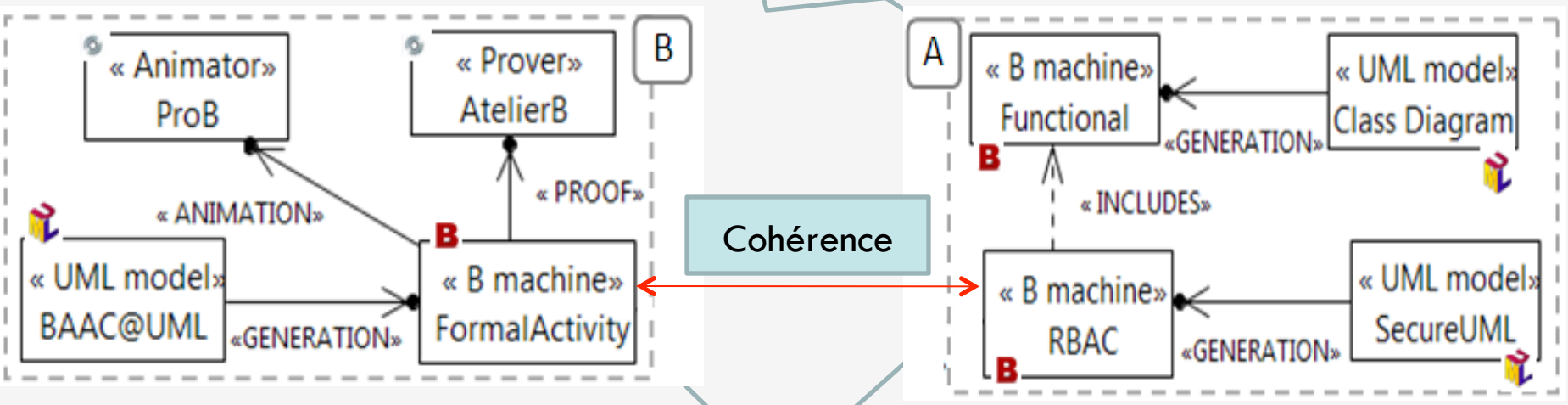
L'initiateur de la réunion ne peut répondre qu'aux changements associés aux invitations de sa réunion



Approche de Formalisation

L'outil B4MSecure (http://b4msecure.forge.imag.fr)

Spécification en B des éléments du diagramme de classes (les classes, les attributs, les opérations et les associations)



Formalisation des éléments fonctionnels et des éléments de sécurité du modèle BAAC@UML

Formalisation de l'affectation des utilisateurs à des rôles et des permissions des rôles aux éléments de classes protégées

La machine FormalActivity (La partie structurelle)

Eléments de spécification du flot de contrôle d'activités

Les objets des classes

AssignedRoles & GACPreCondition

LACPreCondition

MACHINE
FormalFollowAnswer
INCLUDES *RBAC*, *Flow*
VARIABLES
Result, *Mx*, *Ix*, *opened*
INVARIANT
$Result \in STR \wedge Mx \in Meeting \wedge Ix \in INVITATION$
$\wedge\ opened \in BOOL$
$\wedge\ (opened = TRUE \Rightarrow Initiator \in roleOf(currentUser)$
$\wedge\ \{Initiator\} = \mathbf{ran}(Session)$
$\wedge\ meeting\_person(Mx) = currentUser\ )$
INITIALISATION
$Result := nones \parallel Mx :\in Meeting \parallel Ix :\in INVITATION$
$\parallel\ opened := FALSE$
DEFINITIONS
$FA\_InvitationAC == meeting\_invitation(Ix) = Mx$

La machine FormalActivity (La partie exécutive)

Connexion des utilisateurs aux rôles

Ouverture de l'activité

Initialisation du flot de contrôle

Open_activity (*actUser*, *Mi*, *Ii*) =
PRE
actUser ∈ *USERS* ∧ *Initiator* ∈ *roleOf*(*actUser*)
∧ *Mi* ∈ *Meeting* ∧ *Ii* ∈ *Invitation*
∧ *meeting_person* (*Mi*) = *actUser*
∧ *meeting_invitation* (*Ii*) = *Mi*
THEN
setPermissions;
Connect(*actUser*,{*Initiator*});
changeCurrentUser(*actUser*);
Mx := *Mi* ; *Ix* := *Ii* ;
opened := *TRUE* ; setCurrentFlow({*cf1*})
END ;

La machine FormalActivity (La partie exécutive (tâche))

Le flot de contrôle permettant de démarrer la tâche

Les pré-conditions locales de contrôle d'accès

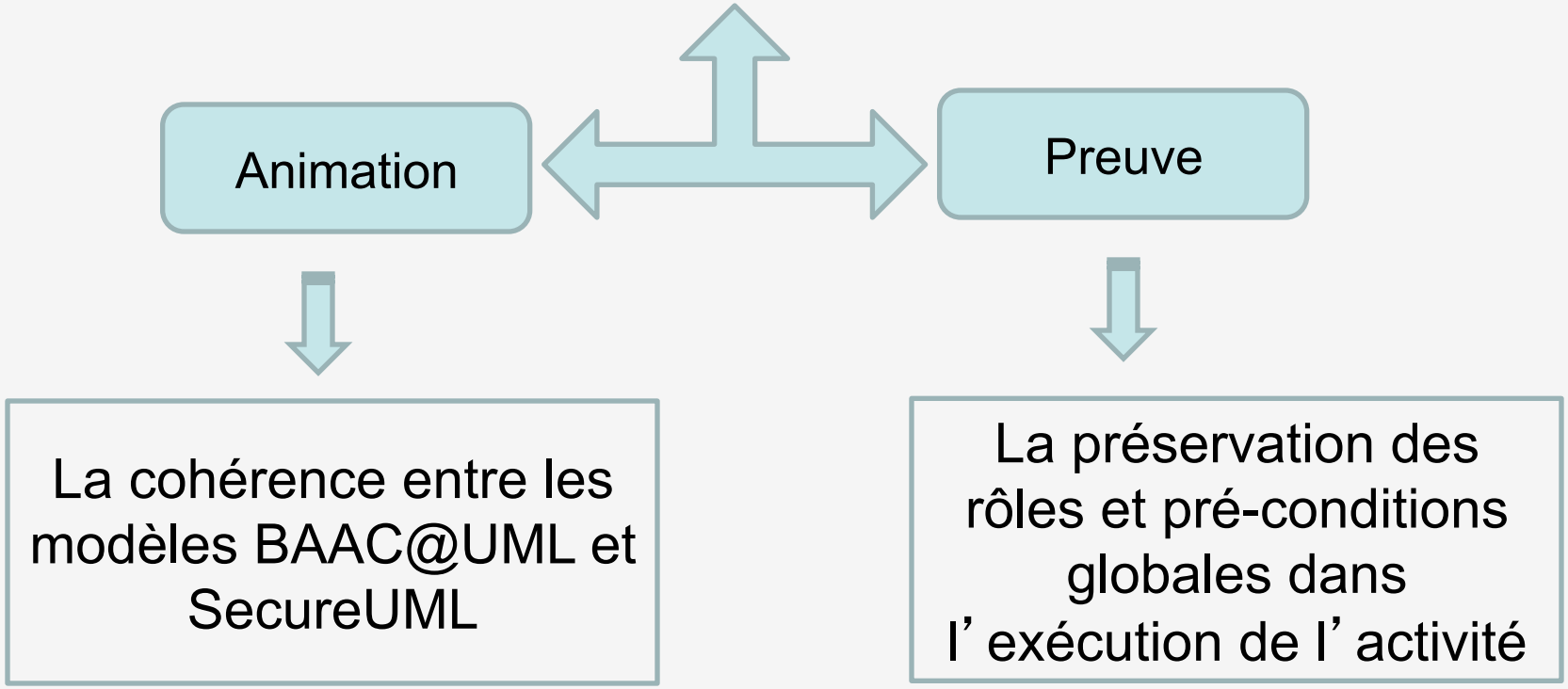
Actions concrètes

```
reply_to_change (Cx) =
PRE cf7 ∈ currentFlow ∧ Cx ∈ Change
    ∧ invitation_change(Cx) = Ix ∧ meeting_invitation(Ix) = Mx
THEN
  Result ← secure_Change__getDate(Cx); addTrace(cf7) ;
  Result ← secure_Change__getTime(Cx); addTrace(cf8) ;
  CHOICE
  secure_Change__setAnswer(Cx, accept); addTrace(cf9) ;
  secure_Meeting__applyChange(Mx) ; addTrace(cf11) ;
  setCurrentFlow({cf12});
  OR
  secure_Change__setAnswer(Cx, refus) ; addTrace(cf10) ;
  setCurrentFlow({cf12})
  END
END;
```

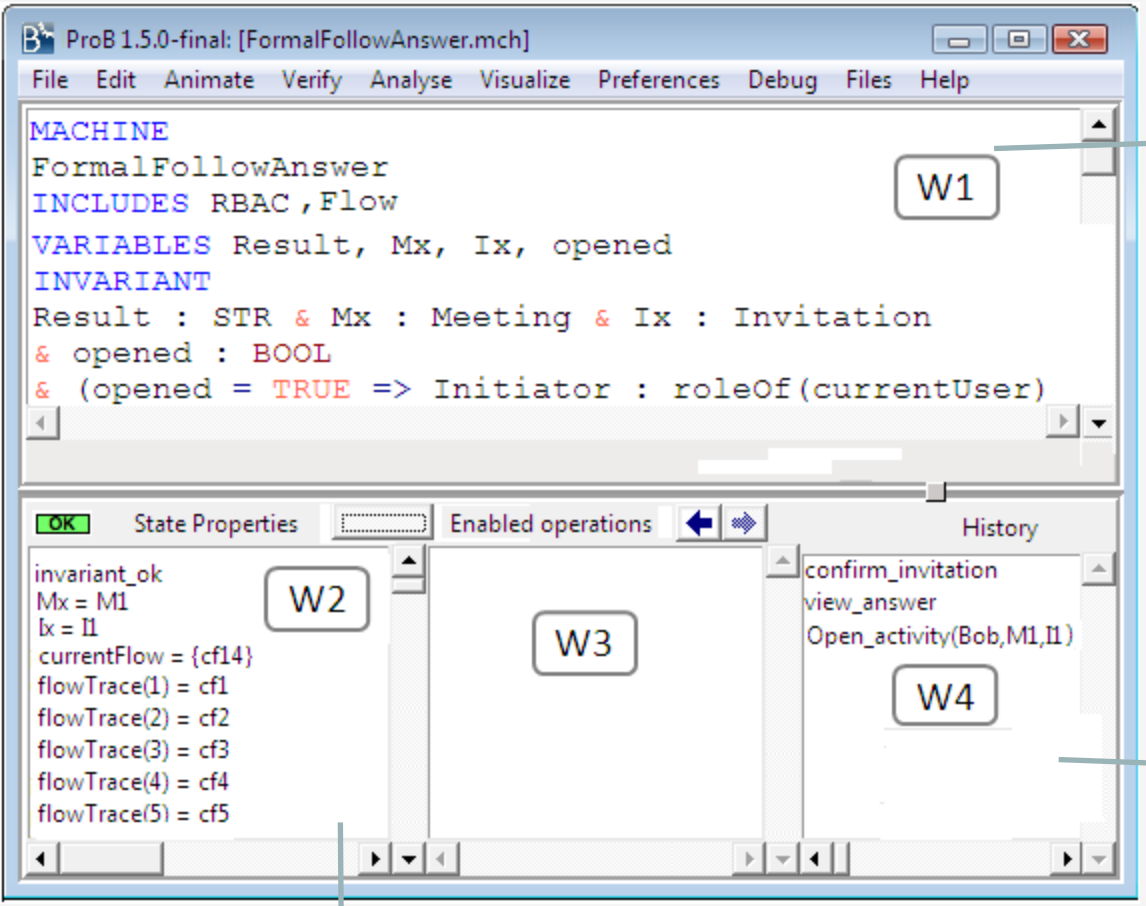
La trace du flux de contrôle

Activités de V&V

But : vérifier rigoureusement la politique RBAC exprimée au niveau des activités



Animation avec ProB



La machine de l'activité Follow answer

La trace d'animation courante

La valeur actuelle de chaque variable de la machine

Preuve avec AtelierB

Génération et démonstration automatiquement des **obligations de preuve** associées aux différentes machines d'activités

L'initialisation et les opérations de la machine d'activité préservent les pré-conditions d'activité *assignedRoles* et *GACPreCondition* exprimées comme invariants

$$
\begin{aligned}
opened = TRUE \Rightarrow\ & Initiator \in roleOf(currentUser) \\
& \wedge \{Initiator\} = \mathbf{ran}(Session) \\
& \wedge meeting\_person(Mx) = currentUser
\end{aligned}
$$

Conclusion et perspectives

Contributions :

- BAAC@UML étend le diagramme d'activités avec les éléments du modèle RBAC
- L'animation et la preuve assurent la cohérence entre les modèles BAAC@UML et les modèles SecureUML

Avantages :

- La séparation entre la spécification des activités et de la politique RBAC
- La spécification graphique et formelle d'une politique RBAC au niveau des activités
- Les modèles BAAC@UML prennent en compte la vue statique de la politique RBAC

Futurs travaux :

- Intégration des TBSoD (Task-based Separation of Duty), BoD (Binding of Duty) et la délégation des tâches
- Utilisation du Model-checking et du prouveur interactif dans les activités de validation

Merci pour votre attention

SalimChehida@yahoo.fr