


# Interopérabilité entre composants dirigée par les modèles



**Frédéric SEYLER**

**Philippe ANIORTE**

**LIUPPA - IUT Bayonne**

# Plan de la présentation



- 1. Positionnement des travaux
- 2. Interopérabilité au niveau conceptuel :  
le modèle de composants
- 3. Interopérabilité au niveau exécution :  
l'exploitation du modèle

# 1. (Re)déploiement de systèmes



- Mise en œuvre de l'évolution des SI (organisation, technologie)
- Gestion de l'éclatement des SI (dissémination des données et des traitements)

# 1. Réutilisation d'éléments existants



- Composants logiciels de haut niveau autonomes à l'exécution
- Réutilisation « a posteriori »
- Intégration de composants hétérogènes

# 1. Distribution



- Architecture logicielle distribuée
- Intergiciel (middleware)
- Interopérabilité

# 1. Interopérabilité entre composants



- Niveau conceptuel (traitements)
- Niveau exécution (interopérabilité entre intergiciels : M2M)
- Synergie entre ces 2 niveaux : approche guidée par les modèles (MDA)

# 1. Interopérabilité au niveau conceptuel



- Modèle de composants
- Principes : découplage entre composants, séparation des flux de données et de contrôle
- Interactions

# 1. Interopérabilité au niveau exécution



- Transformation du modèle initial
- Mise en évidence de la nécessité de ponts d'interopérabilité entre intergiciels
- Implantation des ponts d'interopérabilité



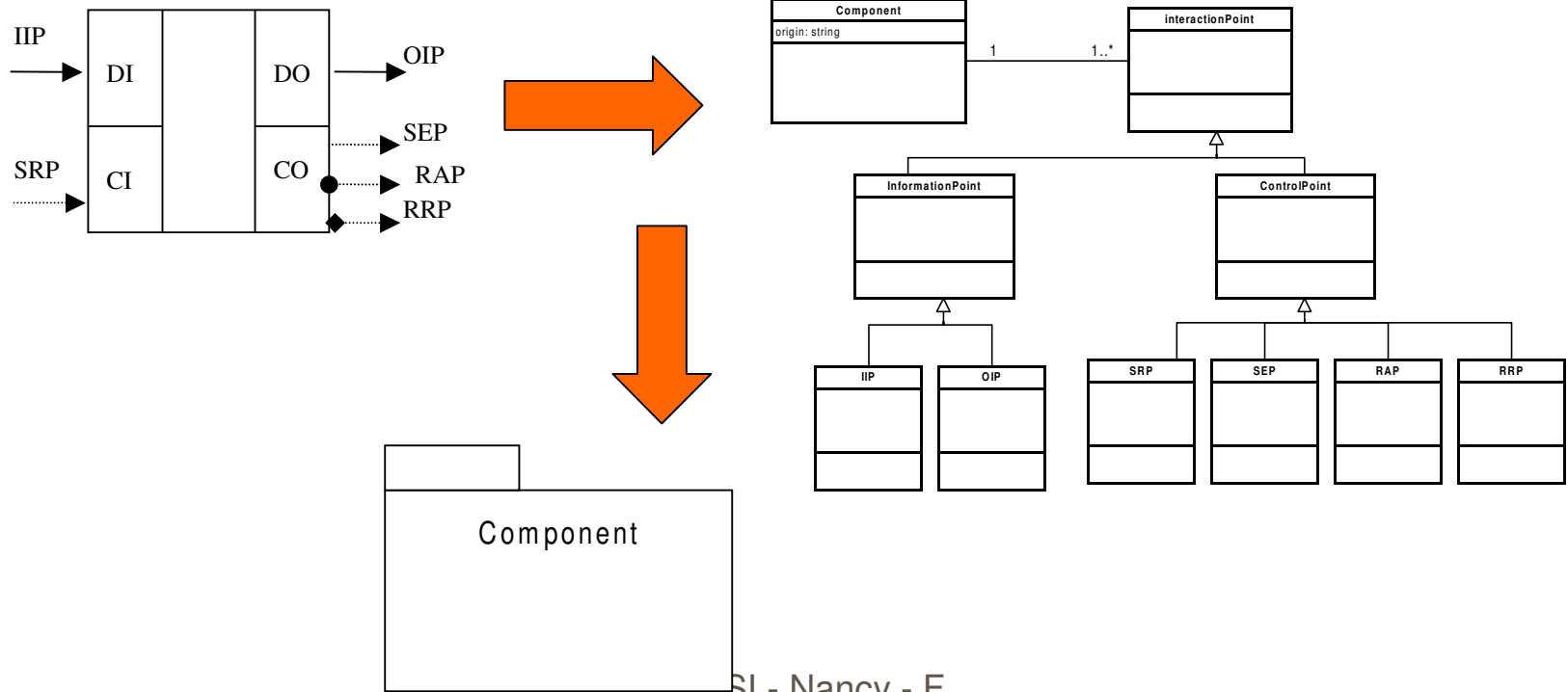
## 2. Méta-modèle « ugatze » : package « component »

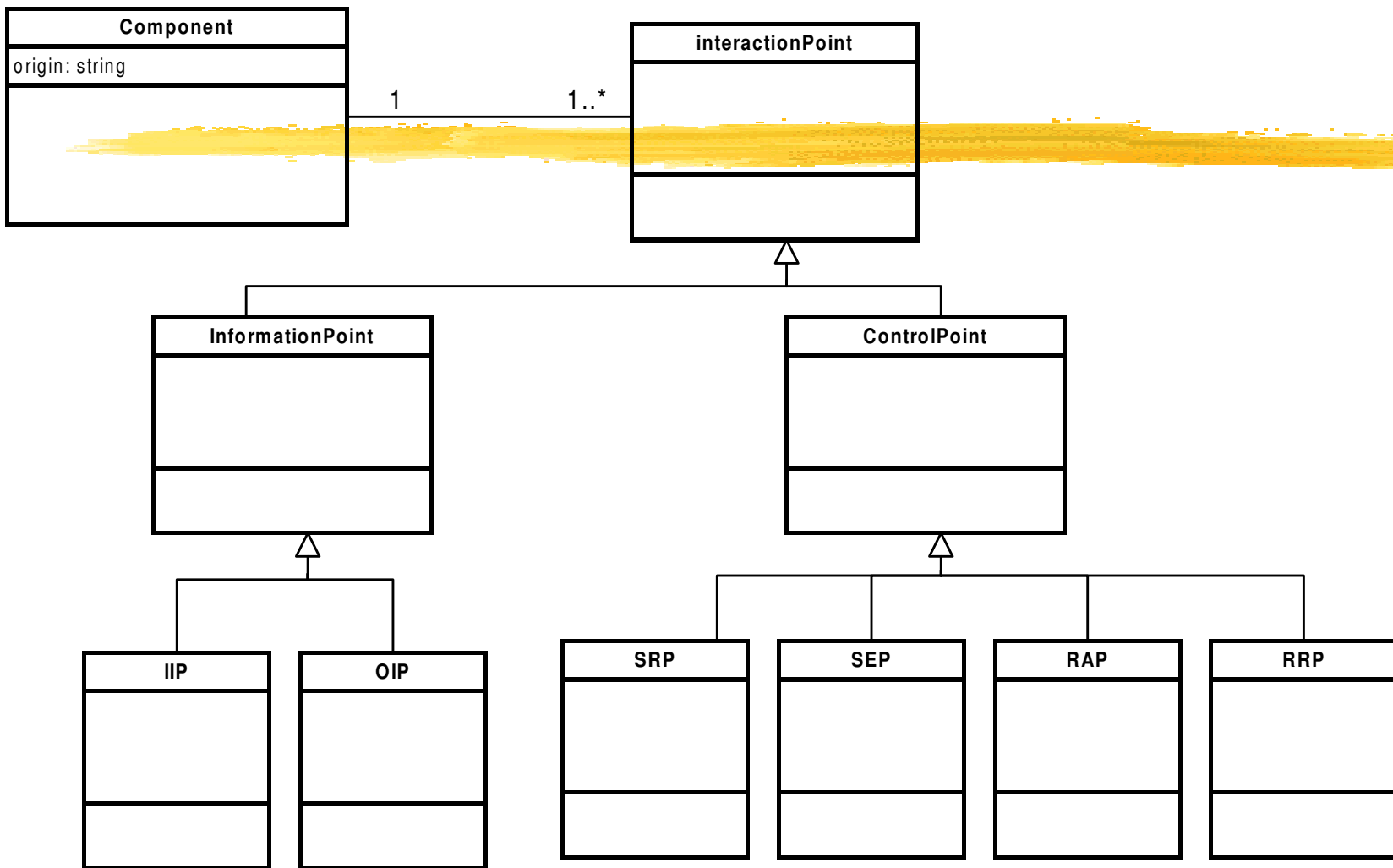


- Représentation des composants (principe d'abstraction)
- Interface : emballage (wrapper)
- Point d'interaction : E/S données/contrôle

## 2. Méta-modèle « ugatze » : package « component »

### Le package MOF « component »





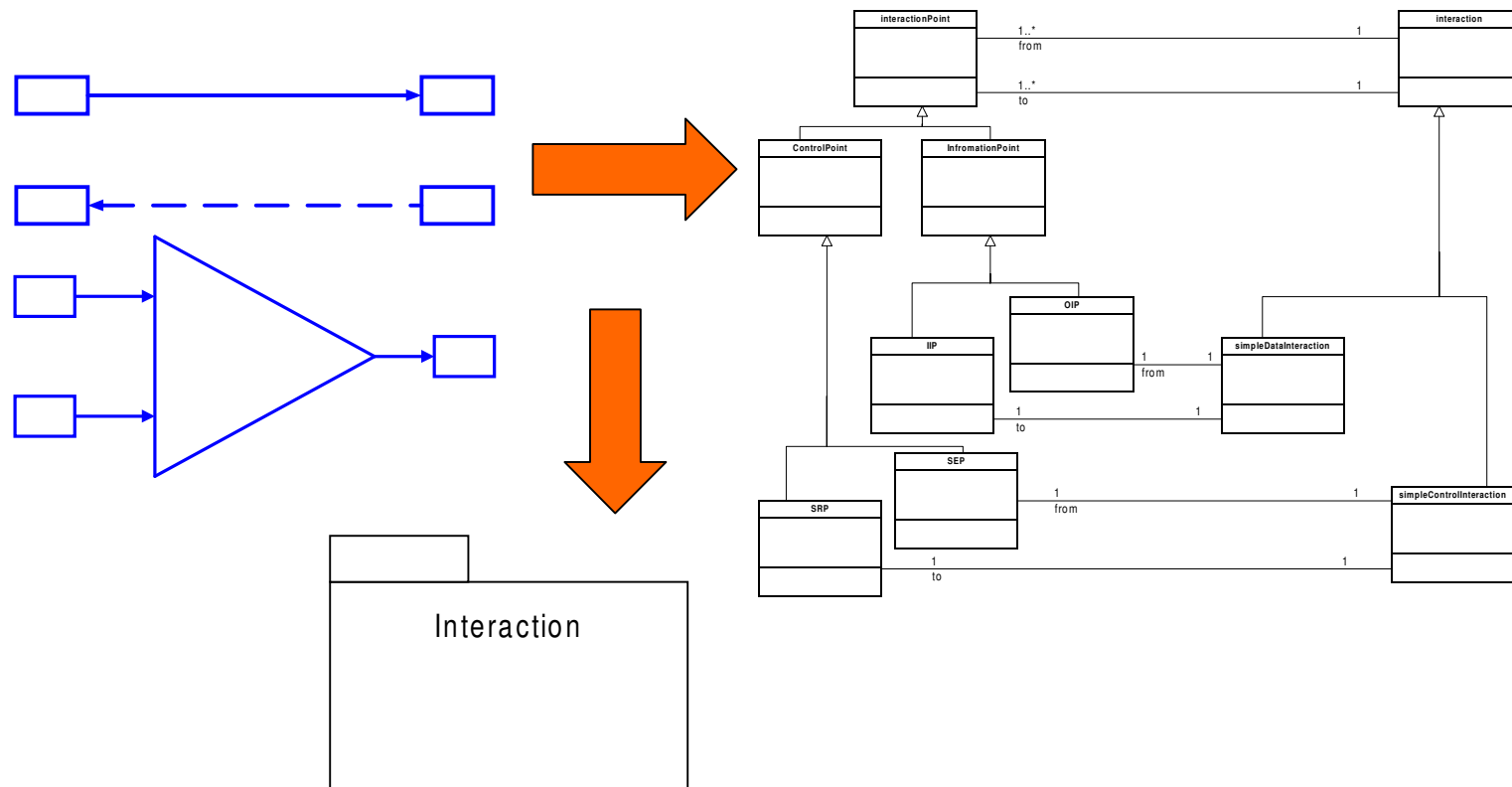
## 2. Méta-modèle « ugatze » : package « interaction »

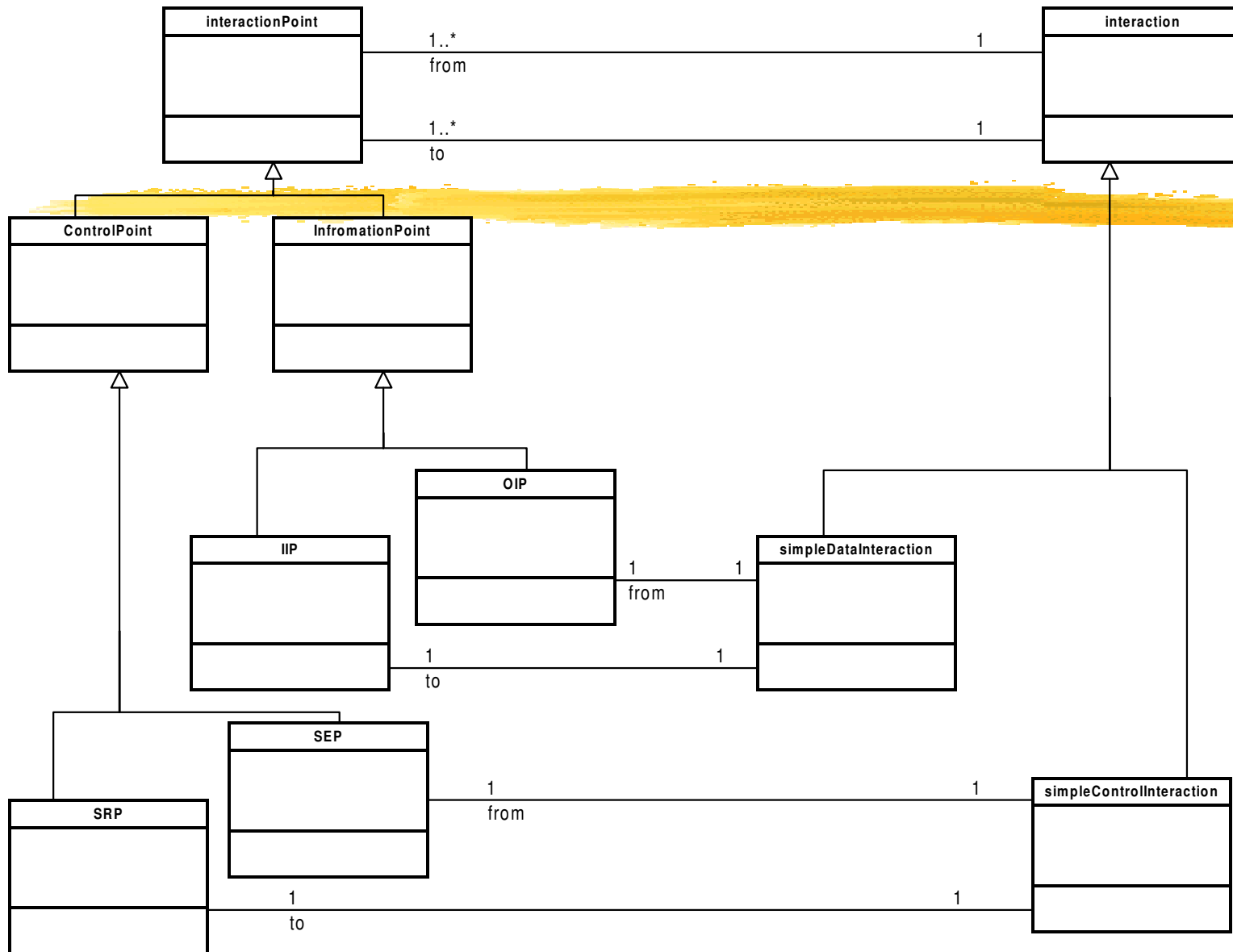


- Représentation des interactions entre composants
- Interactions prédéfinies : données / contrôle
- Interactions à façon

## 2. Méta-modèle « ugatze » : package « interaction »

### Le package MOF « interaction »



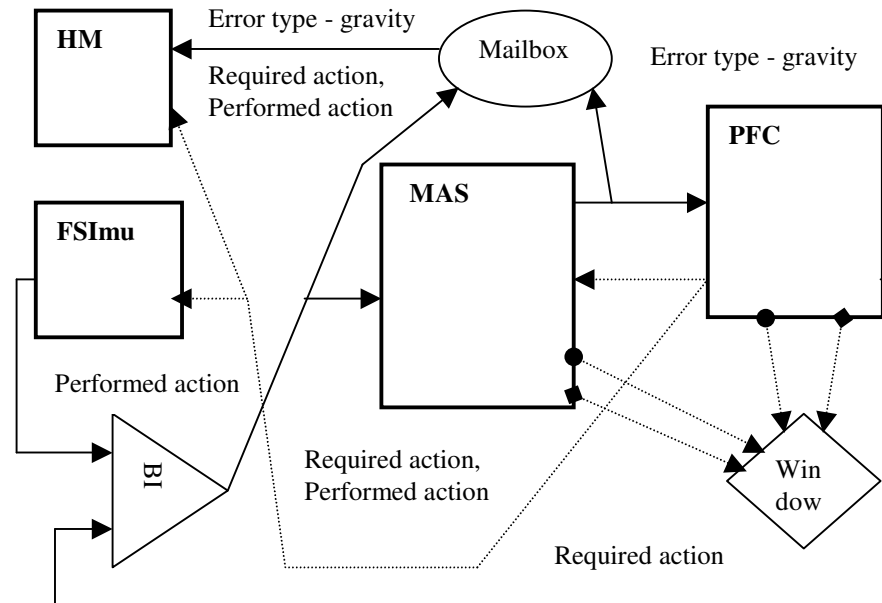


## 2. Instanciation de « ugate » : Exemple de l'application ASIMIL

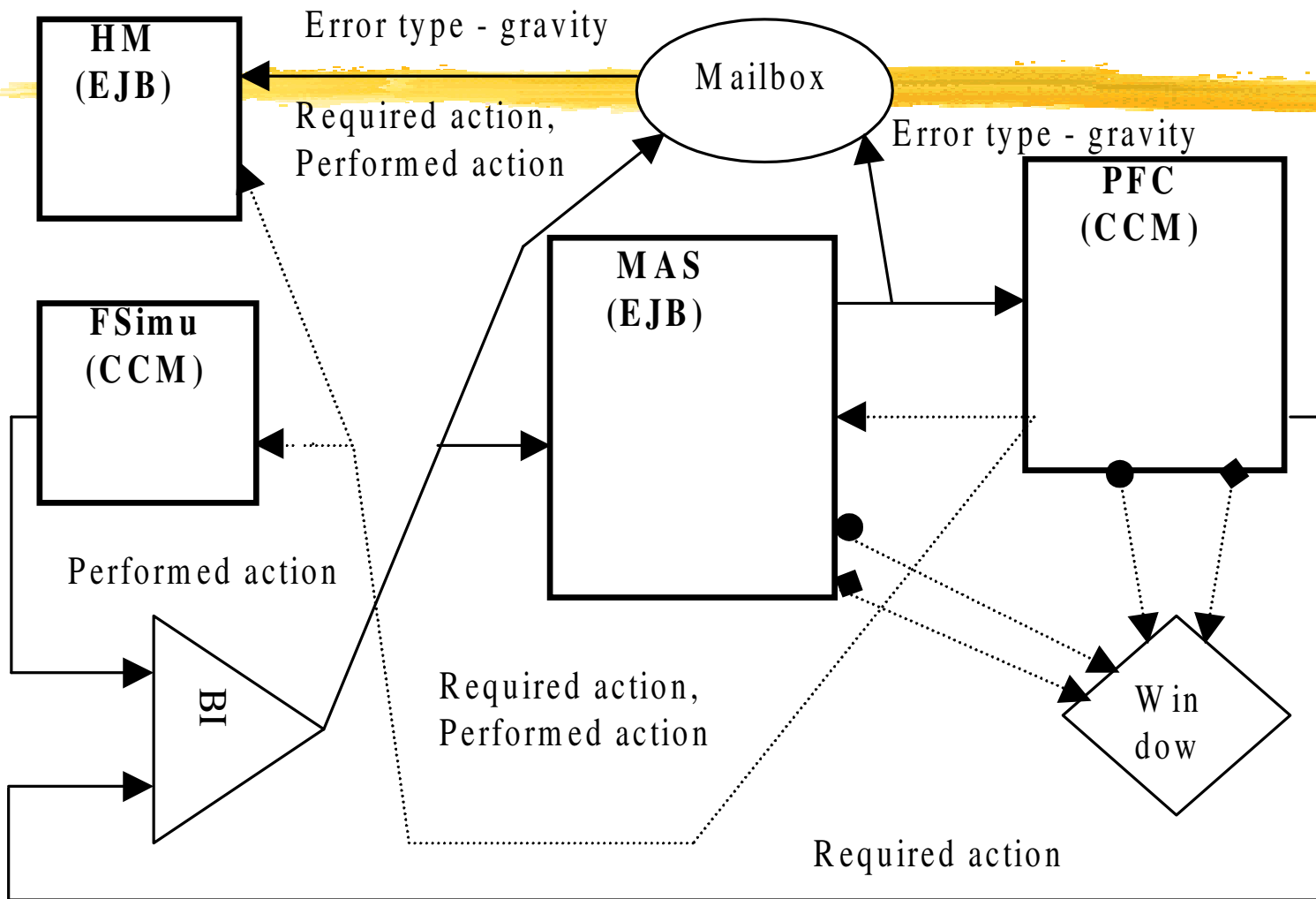


- Aero user-friendly SIMulation-based distance Learning
- Différents composants distribués hétérogènes
- Nécessité d'intégrer ces composants

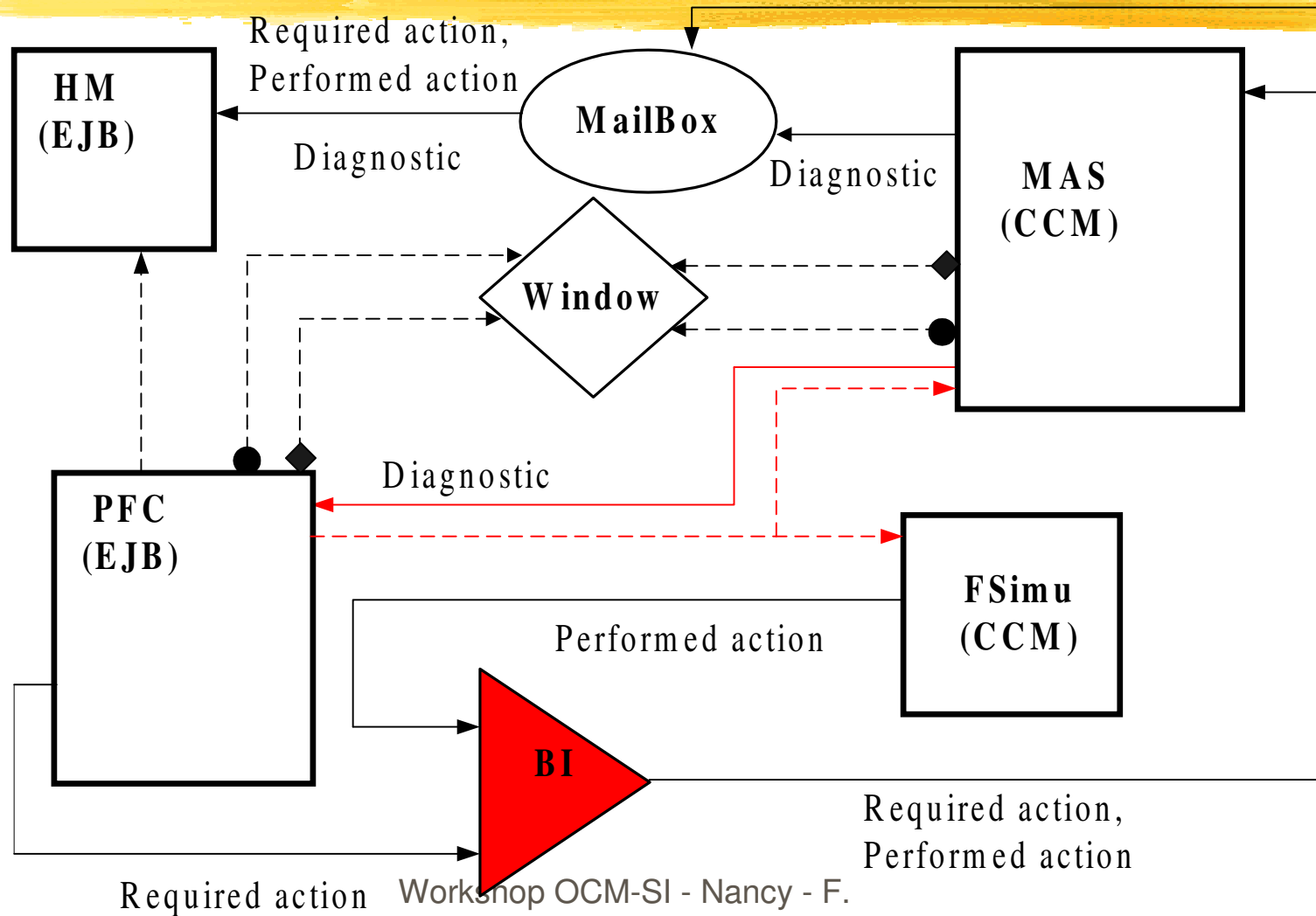
## 2. Modèle d'application d'ASIMIL : Le PIM



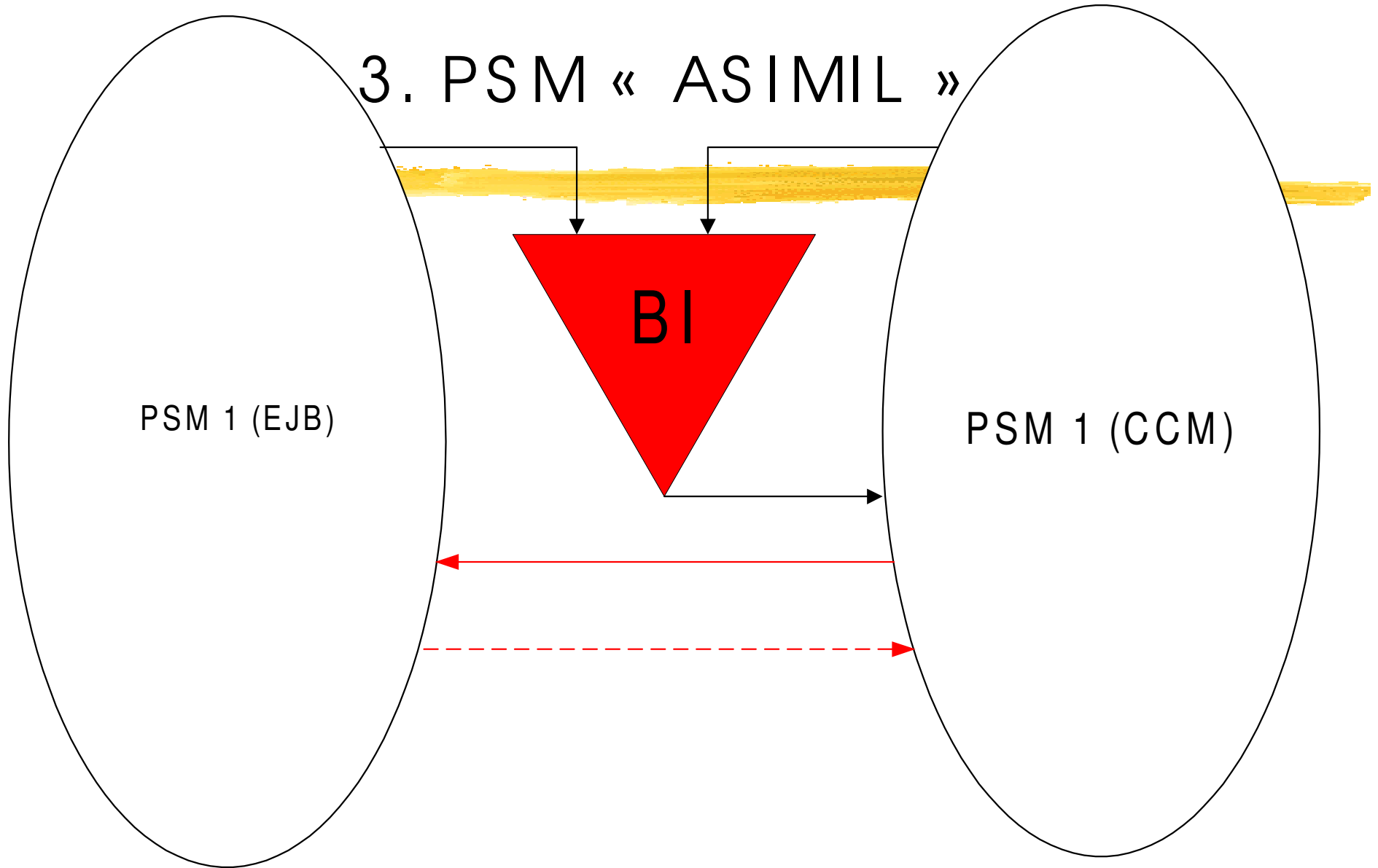




### 3. Transformation du PIM : Interactions inter-intergiciels



### 3. PSM « ASIMIL »



### 3. L'architecture d'intégration

- Un outil pour construire, stocker, importer des modèles d'applications (PIM) conformes au méta modèle « ugatze » : référentiel de modèles MOF-MDA.
- Des outils pour transformer les PIM en modèles exécutables (PSM puis code)
- Des outils génériques correspondant aux interactions prédéfinies

### 3. Le référentiel MOF-Ugatz



- Respecte les 4 niveaux de méta-modélisation
- Offre des outils de construction de PIM (Niveau M1) conformes à ugatz (Niveau M2)
- Permet une représentation graphique des PIM gérée par un outil de visualisation

### 3. Interopérabilité à l'exécution



- Recensement des interactions « inter-organisations »
- Regroupement par types d'interactions
- Mise en oeuvre de l'interopérabilité inter-organisations (M2M)

# 3. Implantation des interactions



- Ponts d'interopérabilité (Corba/EJB, EJB/Corba) : RMI-IIOP ...
- Réutilisation des éléments générés par ces techniques pour implanter les interactions

# Bilan



- Des outils (modèle, démarche ...) adaptés à des besoins spécifiques
- Réutilisation de composants logiciels de haut niveau
- Interopérabilité conceptuelle des traitements
- Synergie entre interopérabilité conceptuelle et interopérabilité à l'exécution
- Champ d'application concret



# Perspectives



- Modèle : points d'interaction, interactions, formalisation
- Développement du référentiel des modèles
- Gestion des ponts d'interopérabilité