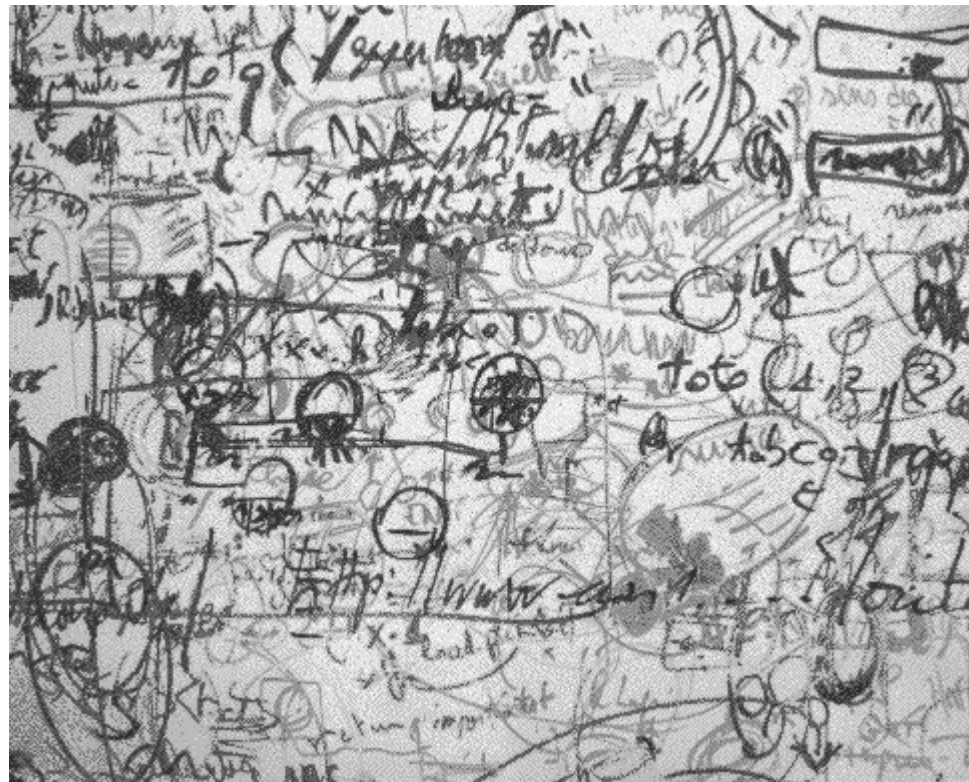


# Méta-modélisation et Référentiels d'Entreprise

---

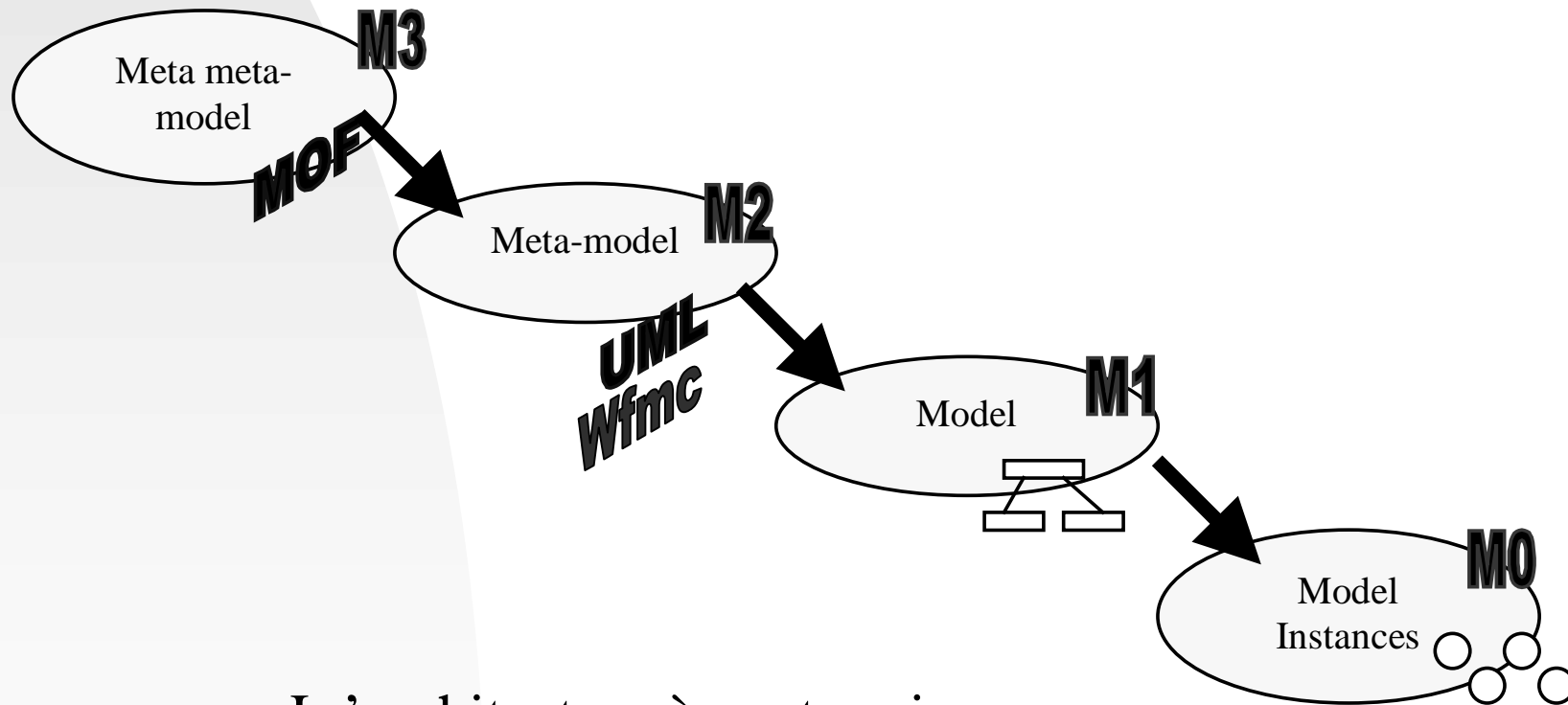
Mariano Belaunde,  
France Telecom - CNET



# PLAN

- 1) Principes de la méta-modélisation
- 2) L'Outillage *Univers@lis* pour la méta-modélisation
- 3) Construction de Référentiels d'Entreprise

# Principes de la méta-modélisation (1)



L'architecture à quatre niveaux  
(standard OMG MOF)

# Principes de la méta-modélisation (2)

classe, relation, héritage, attribut ...

**M3**

Langage universel pour décrire des concepts

activité, acteur, ressource, séquencement ...

**M2**

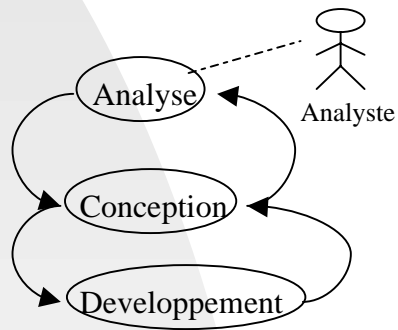
Mon modèle générique de processus

analyse/conception/dev,  
source C, chef de projet...

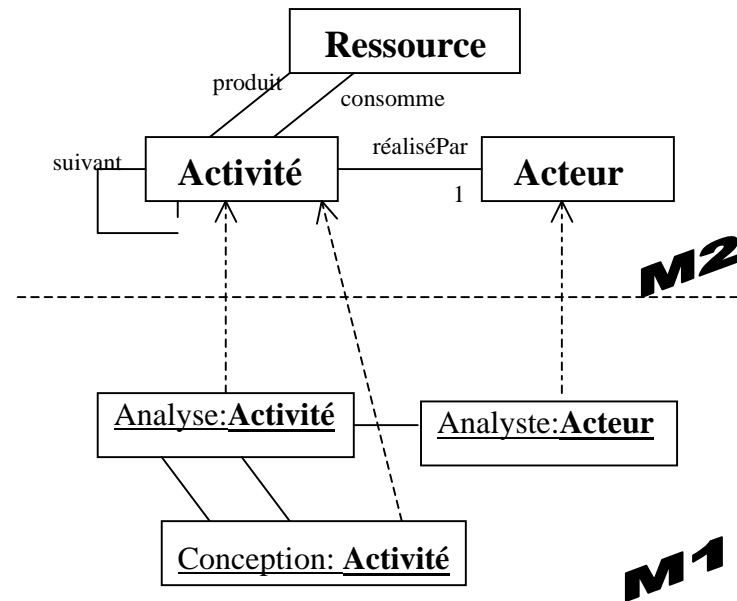
**M1**

Mon processus de développement

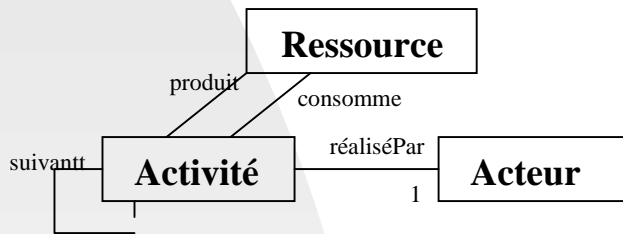
# Principes de la méta-modélisation (3)



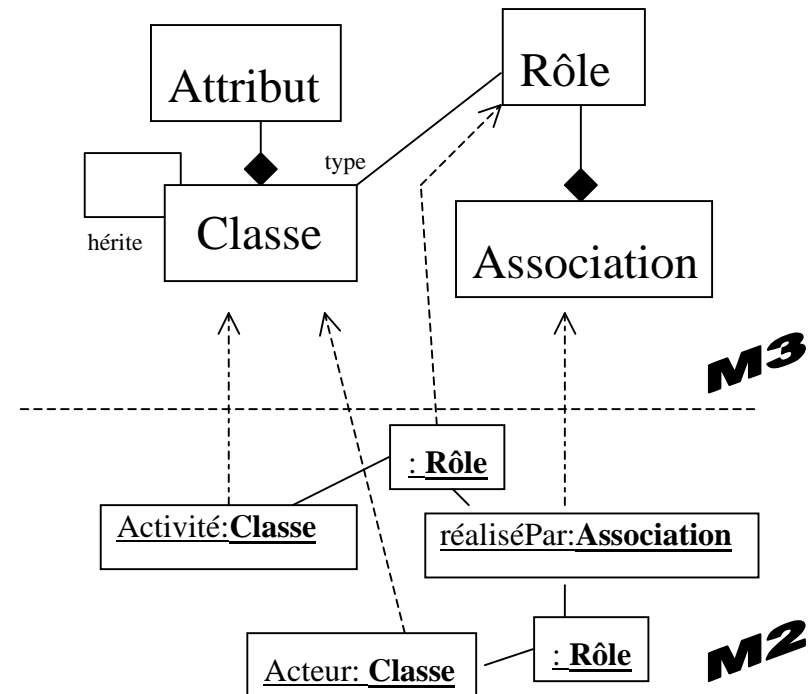
Mon processus de développement



# Principes de la méta-modélisation (4)



Mon modèle générique de processus



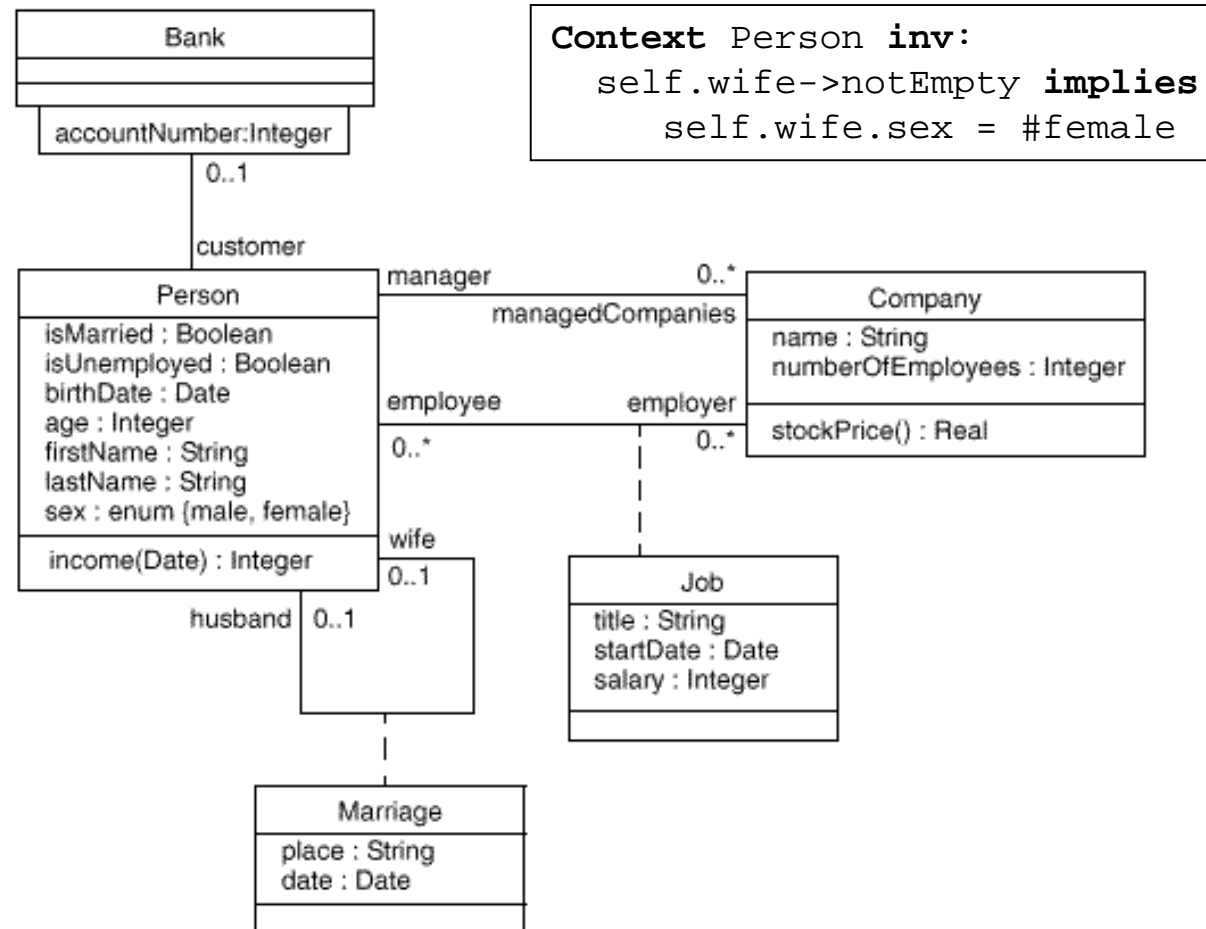
# Principes d'Implantation

- *Mapping* direct vers des langages de programmation objet (classes, héritage, relations, *instanciation*).
- Uniformisation des API (génération automatique et règles génériques)
- Uniformisation de la représentation textuelle pour l'externalisation des instances.

# Positionnement des standards

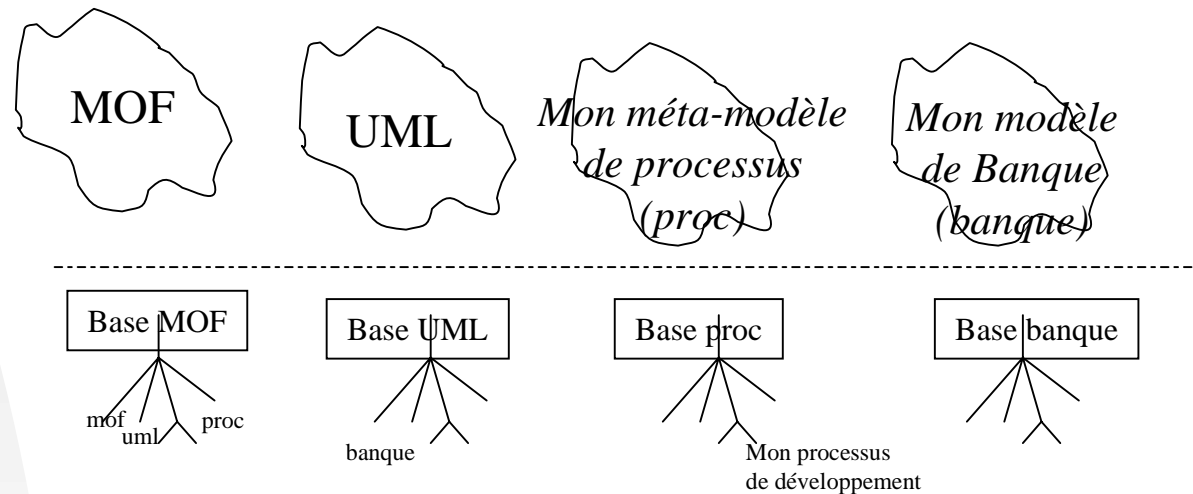
- **OMG MOF (Méta Object Facility)**
  - ◆ standard de méta-modélisation
- **OMG UML**
  - ◆ notation graphique pour la modélisation
- **OCL (Object Constraint language)**
  - ◆ Pour mieux approcher la sémantique d'un modèle.
- **XML**
  - ◆ Publication des contenus sur le WEB, format d'échange (XMI)

# La norme OCL



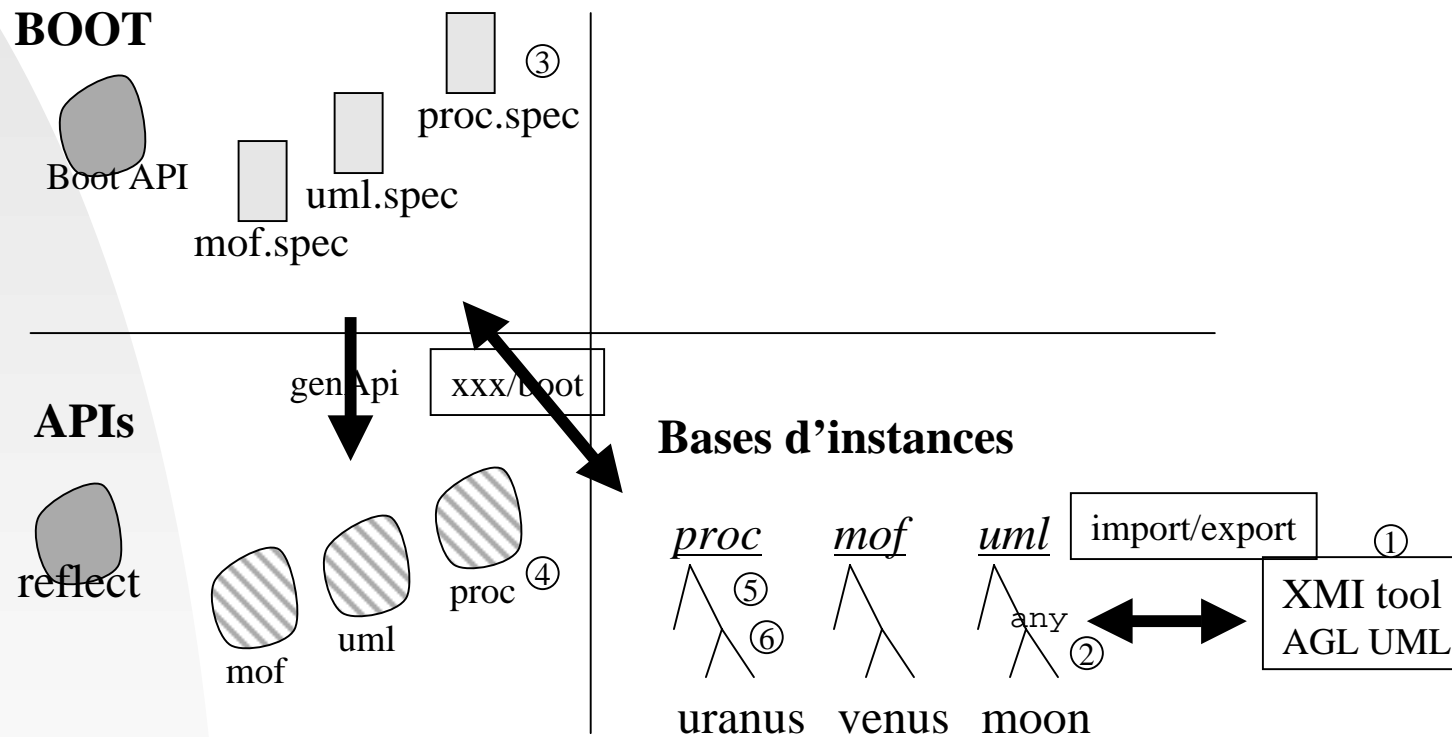
# Univers@lis: Un référentiel d'instances générique

<http://universalis.elibel.tm.fr/>

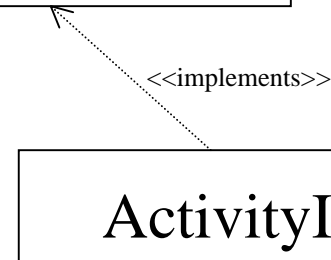
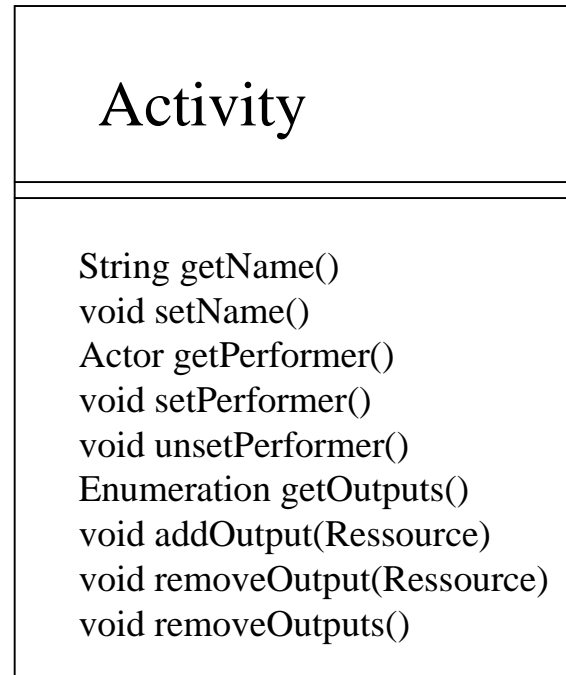
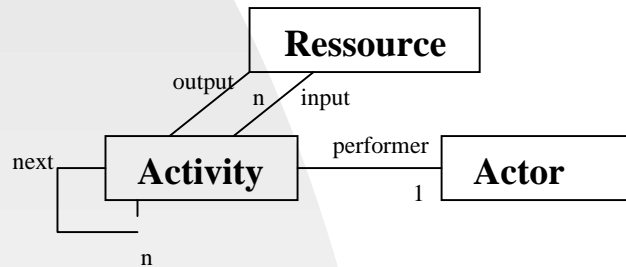


- Support opérationnel d'un modèle objet (par la production automatique de la hiérarchie de classes associée).
- Stockage persistant des instances d'un modèle (enregistrement dans un espace hiérarchique)

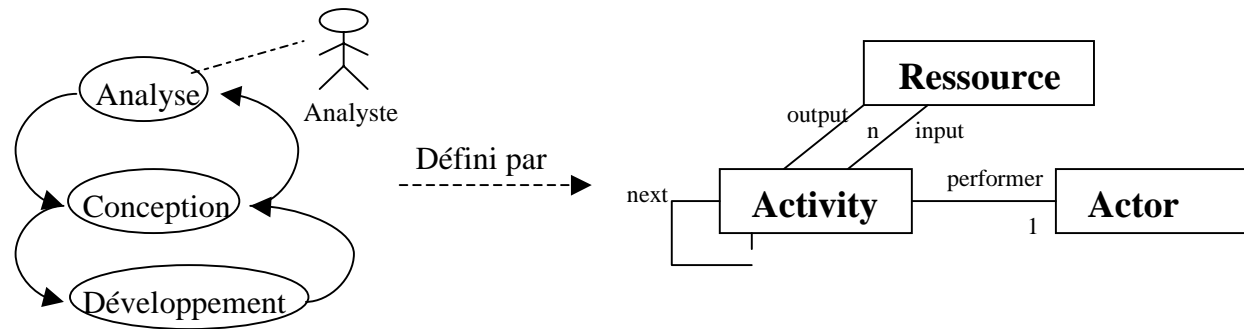
# Univers@lis: Architecture



# Univers@lis: Génération des API



# Univers@lis: Représentations textuelles Génériques



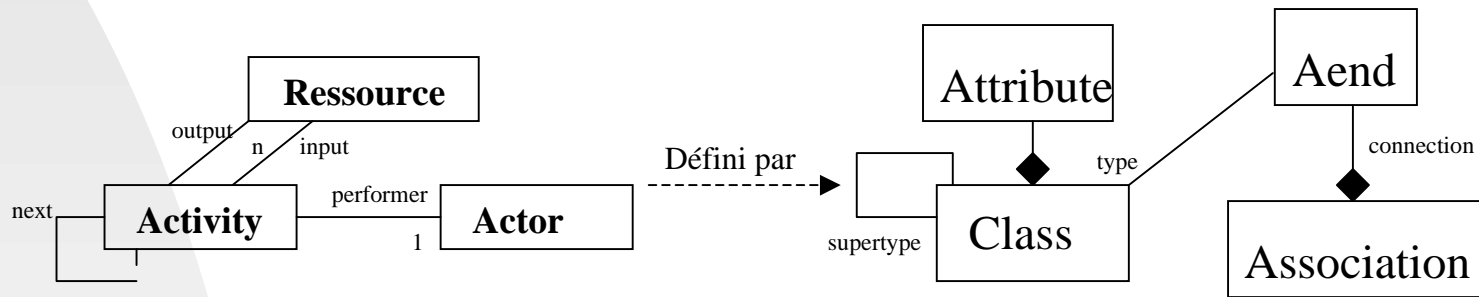
```

Activity analyse { JMI
  name=« Analyse »;
  performer:
    ref Actor analyste;
  next:
    ref Activity conception;
  ....
}
  
```

```

<Activity xmi.id=analyse> XMI
  <name>Analyse</name>
  <performer>
    <Actor xmi.idref=analyste\>
  </performer>
  <next>
    <Activity xmi.idref=conception\>
  </next>
  ....
</Activity>
  
```

# Univers@lis: Représentations textuelles Génériques (2)



```

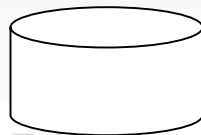
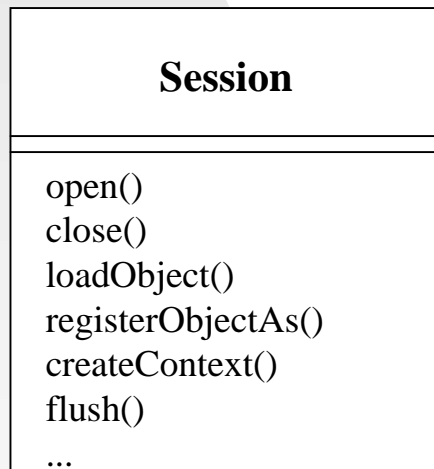
JMI
Association performer {
  connection:
  Aend src{
    type: ref Class actor;
  }
  AEnd dest {
    type: ref Class activity;
  }
}
    
```

```

<Association xmi.id=performer> XMI
  <connection>
    <Aend >
      <type>
        <Class xmi.idref=actor>
      </type>
    </Aend>
    <Aend >
      <type>
        <Class xmi.idref=activity>
      </type>
    </Aend>
  </connection>
</Association>
    
```

# Univer@lis:Propriétés

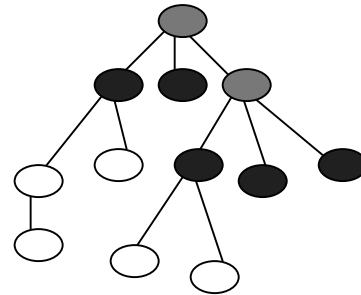
## Encapsulation



File System,  
Data bases,...



## Enregistrement



### Naming Hierarchy

- contexts
- root objects
- sub-objects

### Explicit Registration

## Flexibilité

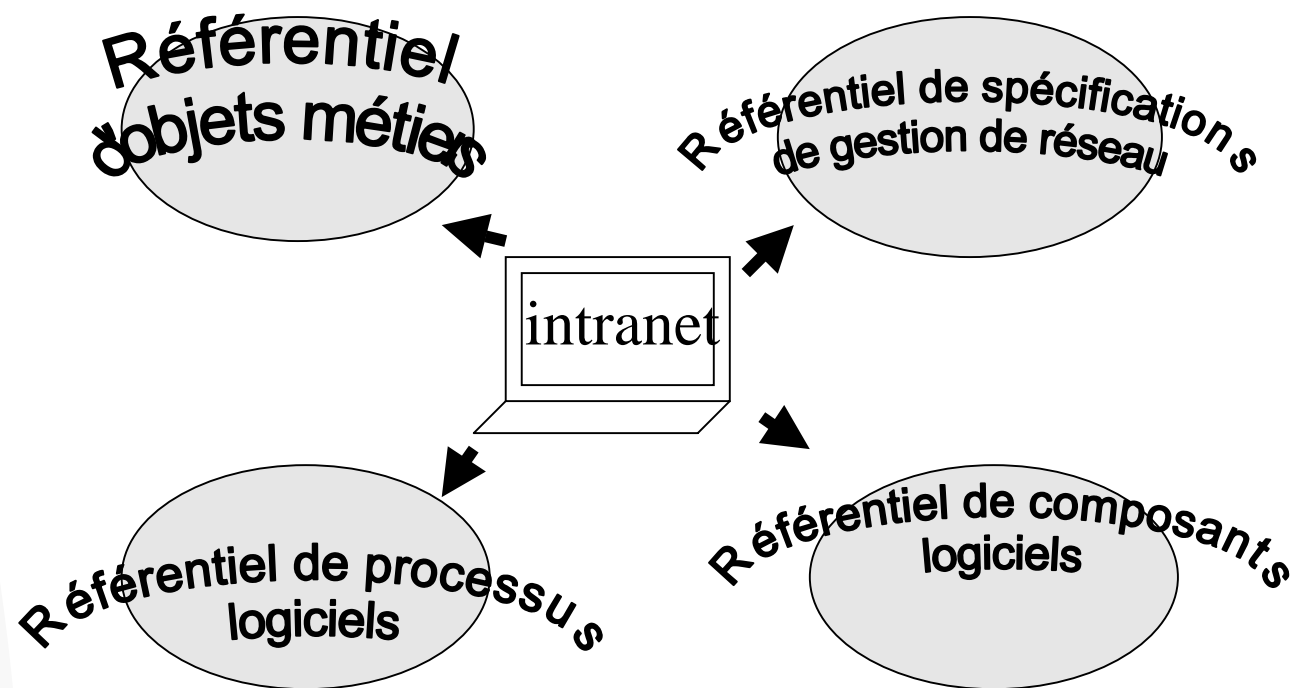
Model containment  
versus OID containment

Attribute Default Values

Proxy Management

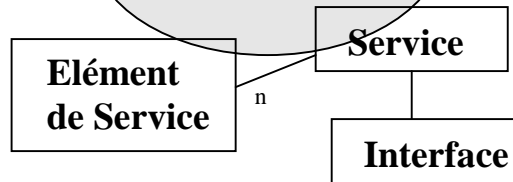
Restricted Navigability

# Construction de Référentiels d'Entreprise

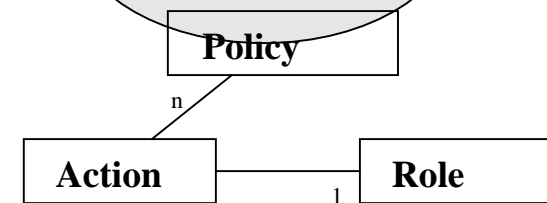


# Construction de Référentiels d'Entreprise (2)

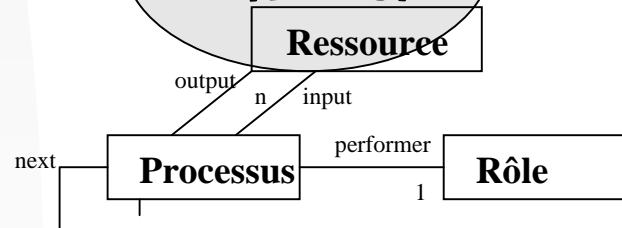
## Référentiel d'Éléments de Service



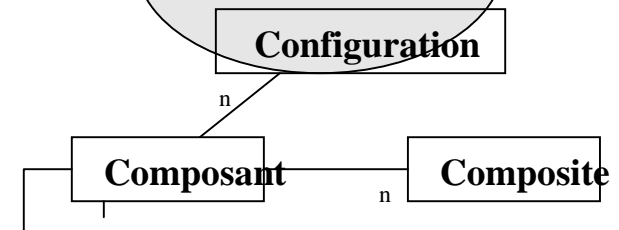
## Référentiel de spécifications de gestion de réseaux



## Référentiel d'objets métier



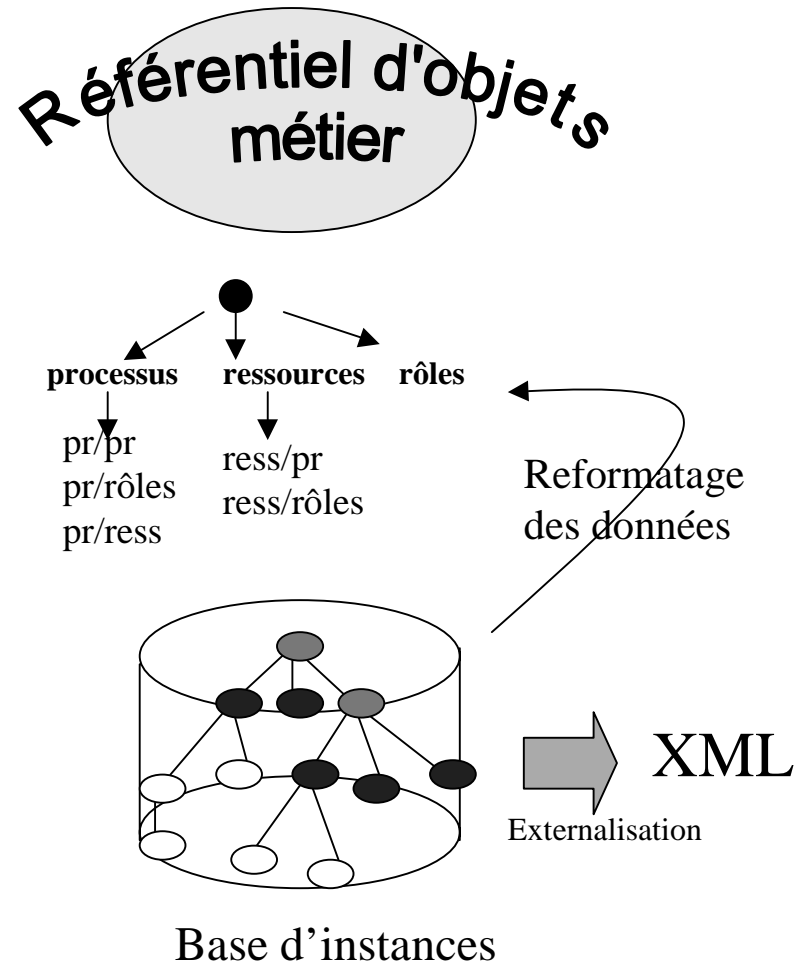
## Référentiel de composants logiciels



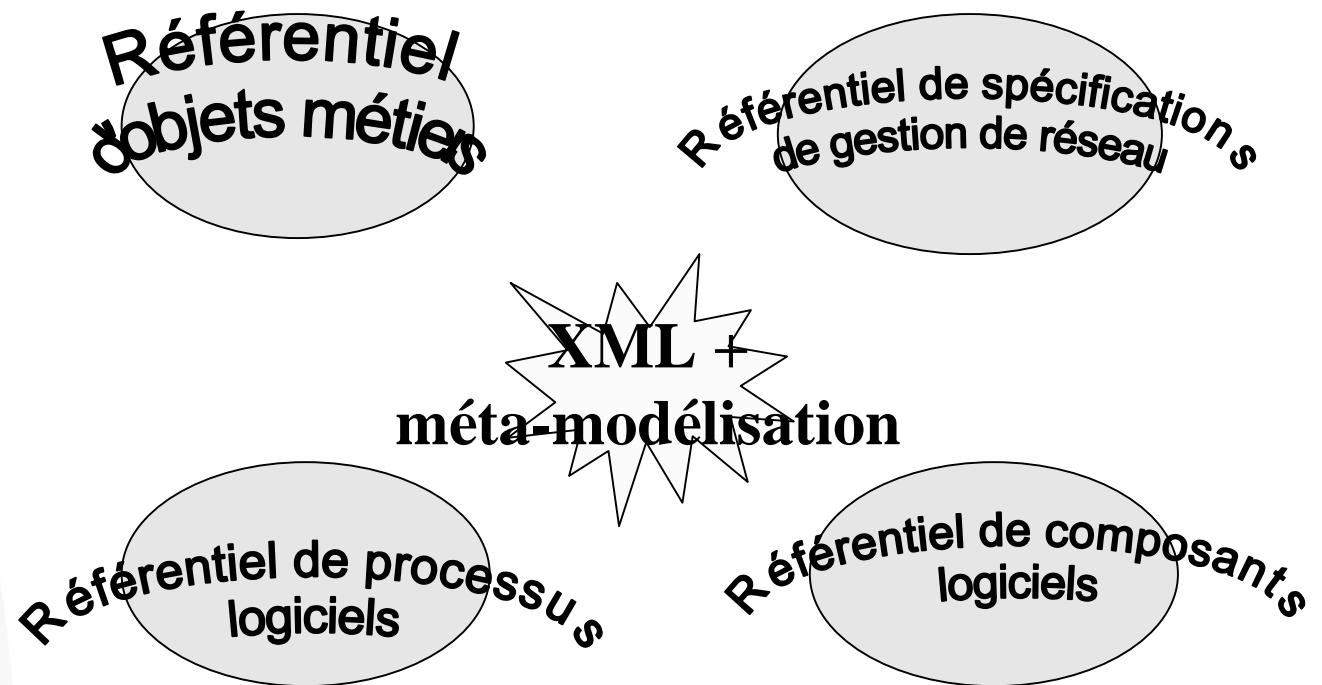
# Construction de Référentiels d'Entreprise (3)

Vers l'automatisation:

- Règles de transformation
- Modélisation de la structure du référentiel



# Construction de Référentiels d'Entreprise (4)



Partage de l'information via les référentiels

# Conclusion

- Les techniques de méta-modélisation peuvent aider:
  - ◆ A mieux maîtriser la cohérence de des données de l'entreprise.
  - ◆ Fournit un cadre permettant d'homogénéiser la représentation de l'information.
- La standardisation (MOF, UML, XML) progresse mais reste incomplète.