



# **Méta-ontologies et ontologies dans MIMOSA**

Jean-Pierre Müller  
CIRAD-TERA-GREEN/LIRMM  
jean-pierre.muller@cirad.fr

COSMAGEMS - 8 janvier 2006



## **Introduction**

## Les ontologies

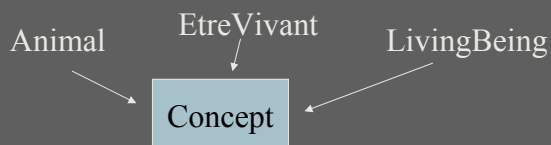
- ⇒ En philosophie:
  - Domaine de la métaphysique traitant de l'être
- ⇒ En informatique:
  - Spécification d'une conceptualisation d'un domaine
    - Conceptualisation: identification d'objets pertinents et regroupement en catégories et leurs relations
    - Spécification: formalisation de cette conceptualisation
  - Ensemble de concepts/classes/catégories/types structuré par des relations taxinomiques et sémantiques, portant sur des structures d'individus/objets/entités

## Les méta-ontologies

- ⇒ « Animal » est un concept/classe/catégorie
- ⇒ « Classe » est un méta-concept/classe/catégorie
- ⇒ Dire que nous allons décrire une réalité par les « objets » qui la compose et allons la structurer en « catégories » ou en « classes » est introduire une méta-ontologie (pauvre!).

## Limites (1)

- ⇒ Animal/Etre Vivant/Living Being: synonymes?
- ⇒ Solution 1:
  - Séparer terminologie et concepts



- Concepts absolus indépendants de la langue (Chomsky, Platonisme)

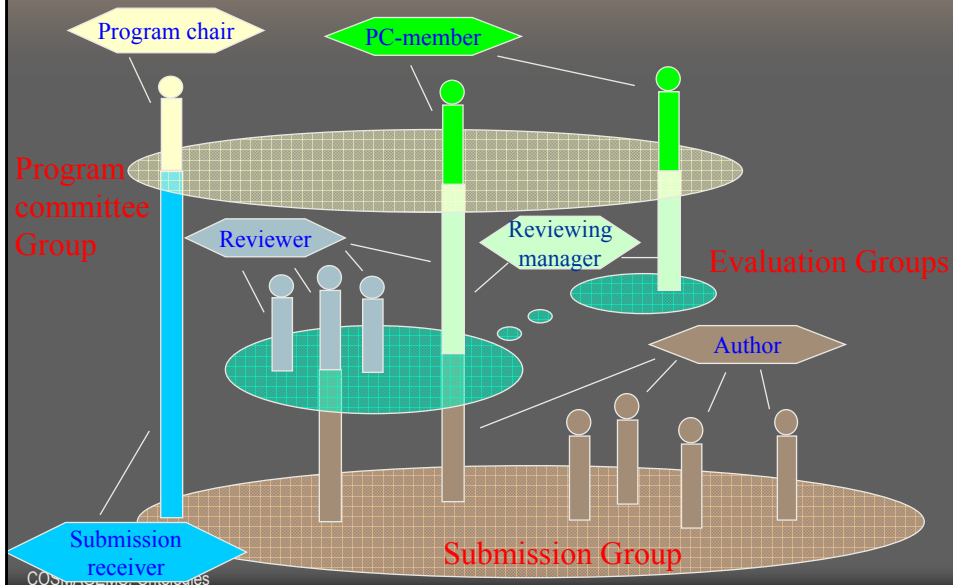
## Limites (2)

- ⇒ Solutions 2:
  - Il n'y a ni réalité ni concepts absolus: seulement des discours selon des points de vue
  - Les termes sont contextuellement en relation, éventuellement synonymes.

# Et les SMA?

- ⇒ Les systèmes multi-agents orientés-agents
  - Des agents en interaction dans un environnement
  - Types d'agents: catégorisation des agents
- ⇒ Les systèmes multi-agents orientés-organisation
  - Des rôles en interaction dans des organisations
  - Les agents coordonnent ces rôles (ou aspects) en participant à diverses organisations
  - Organisations comme points de vue sur la structure/dynamique du système

# Exemple du comité de programme



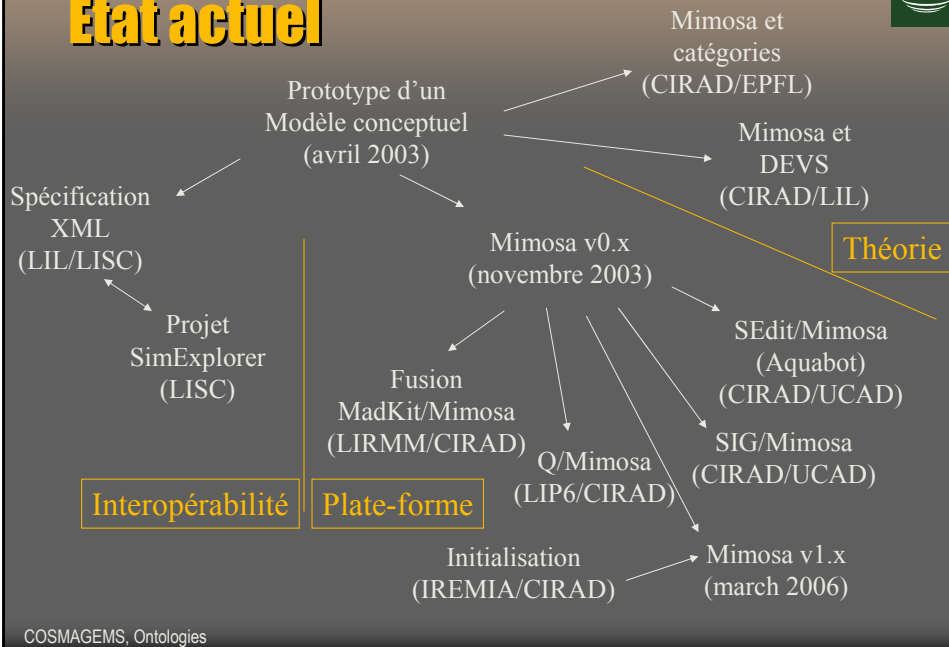
## Cahier des charges

- ⇒ Enrichir les méta-ontologies:
  - Des domaines différents produisent des méta-ontologies éventuellement différentes
- ⇒ Les méta-ontologies comme grammaires:
  - Les méta-ontologies structurent le discours
- ⇒ Les ontologies comme points de vue
  - Articulations d'aspects et de catégorisations relativement un certain point de vue

## L'initiative MIMOSA

CIRAD, LIRMM, IRD, LIP6, CEMAGREF,  
INRA, ENST-Bretagne, ENIB, LIL, LIFL,  
IREMIA, EMSE, DIAM-IUSPIM

# Etat actuel



Interopérabilité

Plate-forme

Théorie

# Architecture (1)

## ⇒ Niveau 0:

- Niveau de couplage des modèles
  - Spécification des aspects dynamiques
- Méta-ontologie: entités et types d'entités

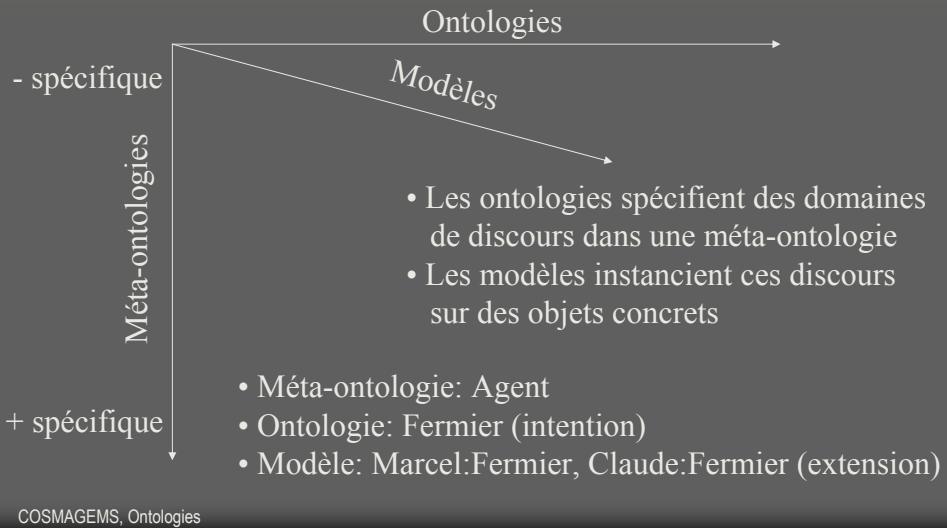
## ⇒ Niveau 1:

- Niveau abstrait de structuration des méta-ontologies
  - Spécification des aspects structurels
- Méta-ontologie: composants, composés, relations et leurs types

## ⇒ Niveau 2:

- Catalogue de méta-ontologies pour décrire les ontologies
  - Ouverture vers la multi-formalisation
- Méta-ontologies comme spécialisations de la méta-ontologie de niveau 1

# Architecture (2)



# Niveau 0

# Méta-ontologie (1)

## ⇒ Entité:

- Un objet du domaine du discours
- Vue interne:
  - Communique via des ports (noms)
- Vue externe:
  - Un port est lié à une ou plusieurs entités

## ⇒ Un modèle est

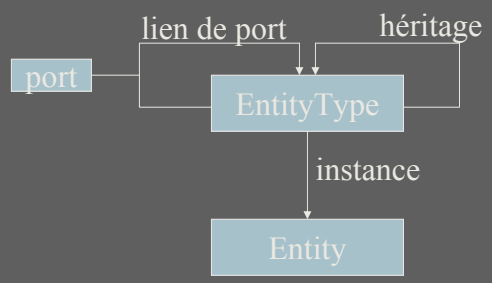
- Un ensemble d'entités liées entre elles
- Une spécification de l'état initial pour chaque entité
- Ex.: diagramme d'objets d'UML

# Méta-ontologie (2)

## ⇒ Type d'entité:

- Décrit ce qu'il y a de commun à plusieurs entités
  - Attributs
  - Ports: à quels types et à combien d'entités on peut les lier
    - ◆ Liens sémantique
  - Super type
    - ◆ Lien taxinomique
- Ex.: diagramme de classes d'UML avec deux types de liens:
  - Héritage
  - Association mono-directionnelle

# Méta-ontologie (3)

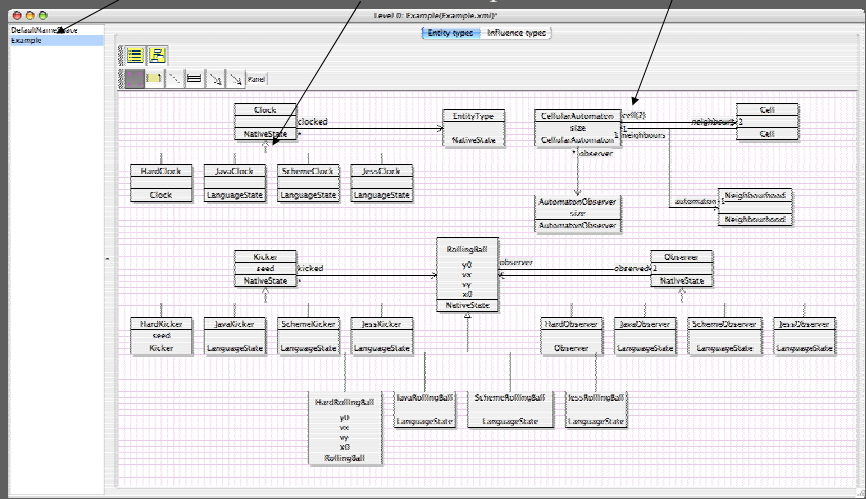


# Editeur conceptuel

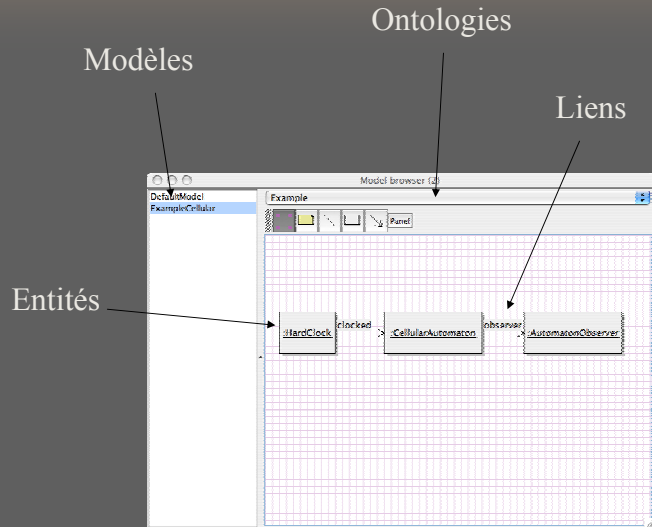
Ontologies (Name space)

Liens sémantiques

Liens taxinomiques



# Editeur de modèles



COSMAGEMS, Ontologies

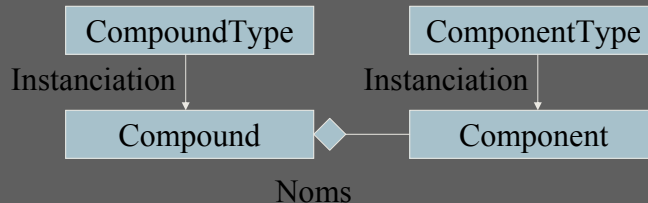


# Niveau 1

# Méta-ontologie (1)

## Les entités:

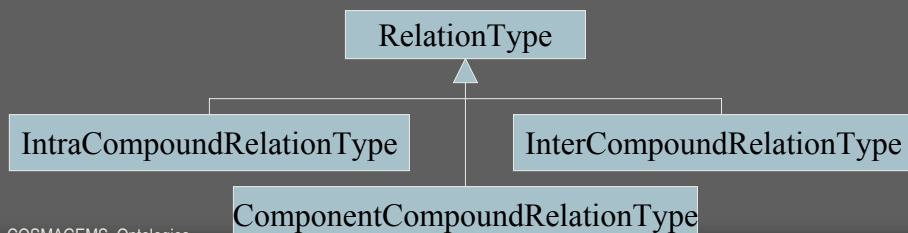
- Composants (éléments):
  - Entités atomiques
  - Types et instances: Component, ComponentType
- Composés (ensemble):
  - Entités décomposables
  - Types et instances: Compound, CompoundType



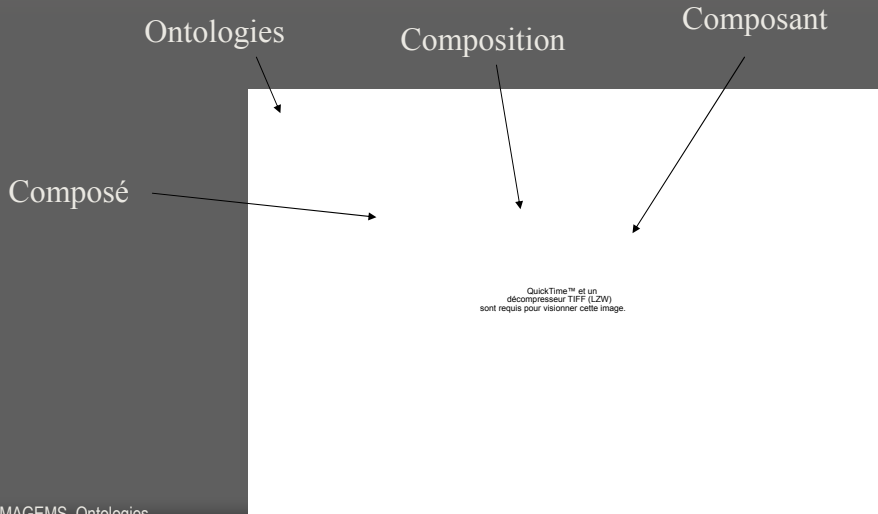
# Méta-ontologies (2)

## Les relations:

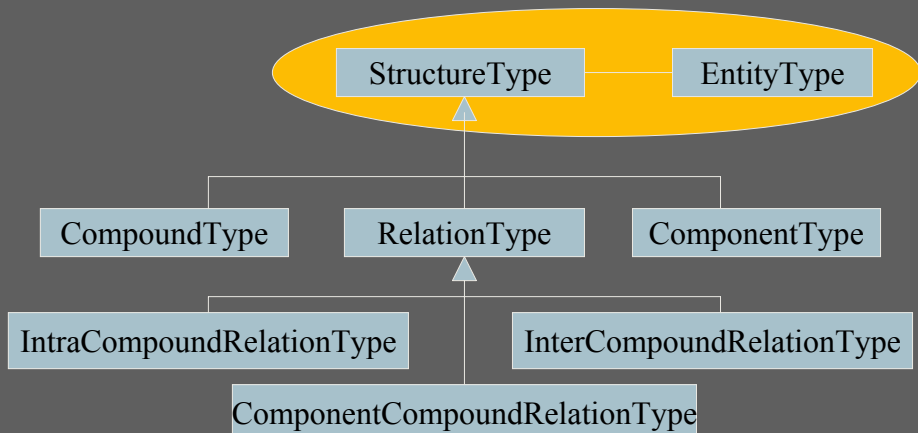
- Relations entre les composants d'un composé
  - Relations sur un ensemble
- Relations entre deux composés
  - Relation entre deux ensembles
- Relations entre un composant et un composé
  - Hiérarchie



# Editeur conceptuel



# Lien niveau 1/niveau 0



## Niveau 2: les méta-ontologies du thématicien

- ⇒ Espace
- ⇒ Temps
- ⇒ Concepts
- ⇒ Automates cellulaires
- ⇒ SMA
- ⇒ ...

## Espace (1)

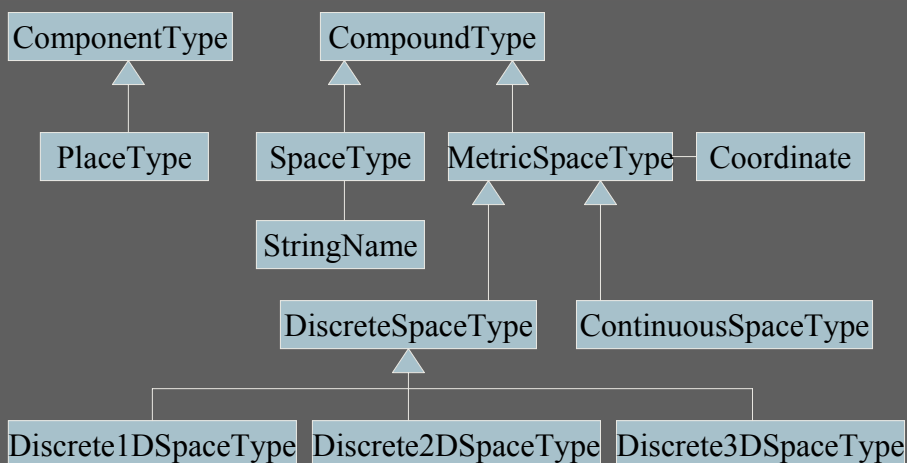
- ⇒ Structure:
  - Composant = lieu
  - Composé = espace (ensemble de lieux)
  - Nommage:
    - Les lieux peuvent avoir des noms (Montpellier, France, ...)
    - ◆ Relations qualitatives entre lieux:
      - Adjacence, inclusion, etc.
      - Relations de Allen sur l'espace
    - Les lieux peuvent avoir des coordonnées discrètes ou continues
    - Un composé peut être considéré comme un champs (un ensemble de fonctions de coordonnées dans un domaine de valeur)

## Espace (2)

### Relations:

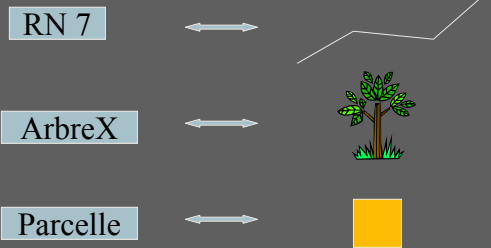
- Sur les lieux avec des noms:
  - Qualitatives sur l'espace: adjacence,...
  - Qualitatives entre espaces
  - Hiérarchies spatiales
- Sur des lieux avec coordonnées
  - Induites des coordonnées elles-mêmes
  - Changement de coordonnées entre espaces
  - Multi-échelle

## Espace (3)



## Remarque

⇒ Les lieux sont distincts de leur(s) représentation(s) géométrique(s)



## La méta-ontologie géométrique (1)

⇒ Composants: les objets géométriques

- Les objets ponctuels:
  - Icônes, textes
  - Attributs: position, orientation, couleur, police, taille, etc.
- Les objets linéaires:
  - Lignes, courbes, combinaisons
  - Attributs: position, orientation, taille, épaisseur, couleur, etc.
- Les objets surfaciques:
  - Ellipses, polygones
  - Attributs: position, orientation, couleur intérieure, couleur/épaisseur du bord, taille, etc.
- Les objets volumiques:

## La méta-ontologie géométrique (2)

### ⇒ Composés:

- Ensemble d'objets géométriques
- Visualisables

### ⇒ Relations:

- Entre tout composé et un composé géométrique:
  - Définition de sa forme visuelle
  - Plusieurs formes possibles
  - Liens entre les attributs des composants et les attributs géométriques



Vers une modélisation sémiotique

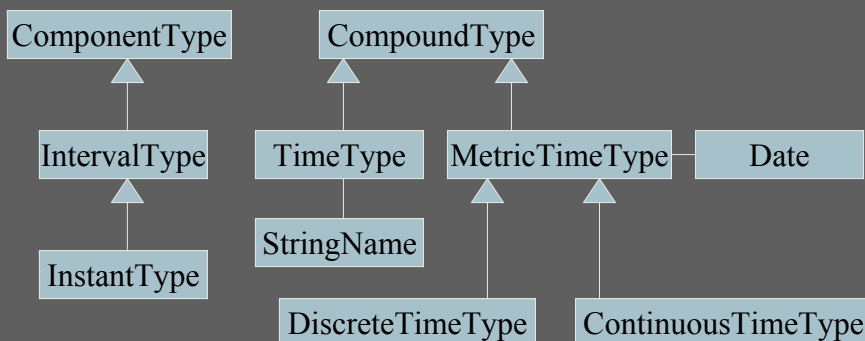
## Niveau 2: les méta-ontologies du thématicien

- ⇒ Espace
- ⇒ Temps
- ⇒ Concepts
- ⇒ Automates cellulaires
- ⇒ SMA
- ⇒ ...

## Différents types de temps

- ⇒ Temps du calcul:
  - Séquence des évènements par l'enchaînement du calcul
- ⇒ Temps simulé:
  - Interprétation du temps du calcul comme écoulement du temps du phénomène simulé
- ⇒ Temps de l'observation:
  - Le temps dans lequel les processus sont mesurés
- ⇒ Temps représenté:
  - La description des évènements dans le temps: équivalent à l'espace

## Méta-ontologie du temps représenté



## Niveau 2: les méta-ontologies du thématicien

- ⇒ Espace
- ⇒ Temps
- ⇒ Schémas
- ⇒ Automates cellulaires
- ⇒ SMA
- ⇒ ...

## Représentations centrées-objets

- ⇒ Les langages centrés-objets ( $\neq$  orientés-objets!!!) décrivent les ontologies:
  - Ex.: FRL, SRL, KRL, Sherpa
- ⇒ Propriétés:
  - Visibilité de leurs descriptions
  - Pas de distinction classe-instance
  - Similaire aux graphes conceptuels
- ⇒ Actuellement, l'ingénierie des connaissances utilise les logiques de description (LD)

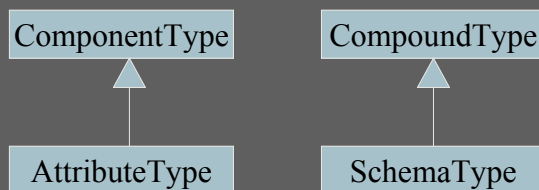
## Extensions dans Mimosa

- ⇒ Structuration en espaces de noms (i.e. ontologies)
- ⇒ Taxonomies des points de vue sur les objets plutôt que les objets eux-mêmes
- ⇒ Etat associé pour en décrire la dynamique
  - Distinction classe-instance plus importante (entre centré-objet et orienté-objet)
- ⇒ Nommage des instances
  - Le nom d'un concept dépend du contexte
  - Les noms peuvent être indexés

## Schémas

- ⇒ Description structurée d'un ensemble d'éléments:
  - Exemple: une voiture (composé) est composée de 4 roues, un moteur, etc. (ses composants).
- ⇒ Structure:
  - Composants:
    - Les attributs avec leur type et leur cardinalité
  - Composés:
    - Les schémas comme ensemble d'attributs accessibles par leur nom d'attribut
  - Les relations peuvent être quelconques.

# Implémentation



## Niveau 2: les méta-ontologies du thématicien

- ⇒ Espace
- ⇒ Temps
- ⇒ Concepts
- ⇒ Automates cellulaires
- ⇒ SMA
- ⇒ ...

# Les automates cellulaires

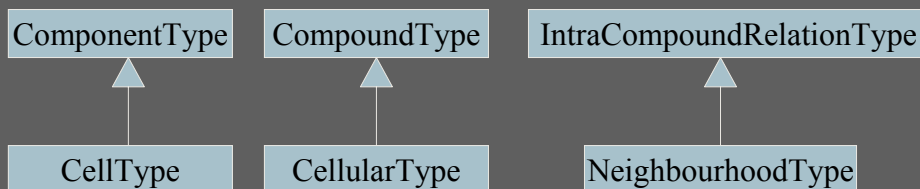
## ⇒ Structure:

- Composants: les cellules:
  - Fonction de transition
- Composés: la matrice 1 à 3 dimensionnelle de cellules:
  - Dérivés de la notion d'espace discret
- Relations:
  - Voisinages de Von Neumann et de Moore, toroïdaux ou pas

## ⇒ Dynamique (cellules et automates):

- Evolution synchrone ou asynchrone sur un « tick » d'horloge

# Implémentation



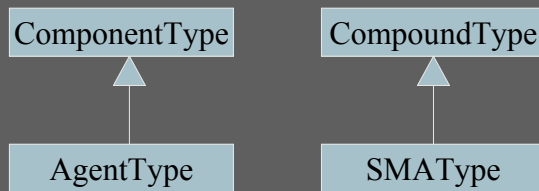
## Niveau 2: les méta-ontologies du thématicien

- ⇒ Espace
- ⇒ Temps
- ⇒ Concepts
- ⇒ Automates cellulaires
- ⇒ SMA
- ⇒ ...

## Les agents

- ⇒ Les agents sont un point de vue particulier sur les entités en tant qu'agents (et non pas environnement, objet, etc.)
- ⇒ Structure:
  - Composants: les agents
    - L'état contient une boîte aux lettres pour stocker les événements (= messages)
  - Composés: les populations d'agents
    - Contient tous les agents d'un modèle
    - Différent des groupes

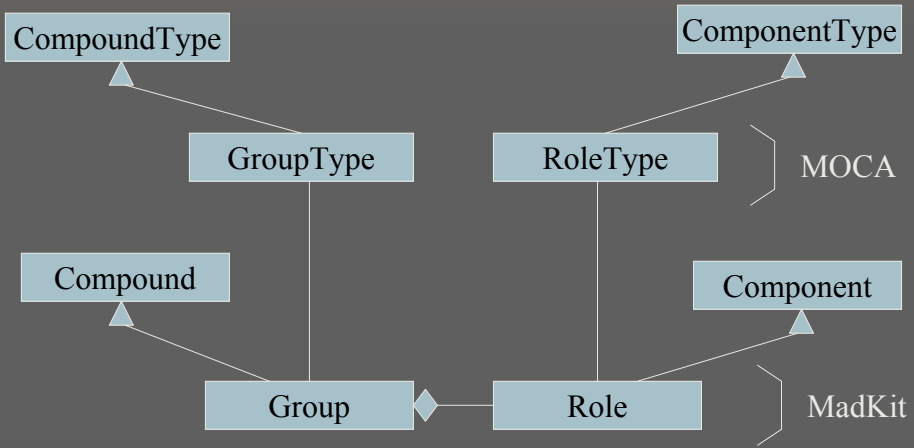
# Implémentation



# Le modèle AGR

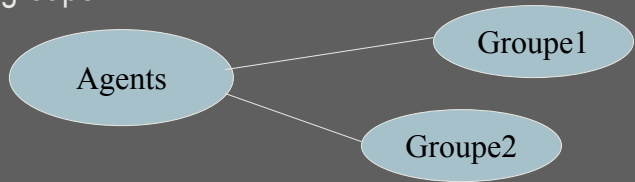
- ⇒ Les agents prennent des rôles dans des groupes:
  - Ils peuvent participer à plusieurs groupes
- ⇒ Structure:
  - Composants: les rôles
    - Ils définissent ce qu'un agent ou une entité peut faire dans une organisation
  - Composés: les groupes
    - Ils définissent les règles d'échange entre les rôles d'une même organisation

# Implémentation



# Prise de rôle

- ⇒ Prise d'un rôle dans un groupe par un agent:
  - Relation entre la population des agents et chaque groupe



- Prise de rôle = modification de la relation

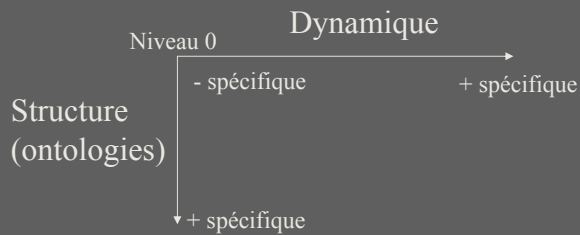
# Généralisation

- ⇒ Tous les objets ont des rôles/aspects multiples selon le point de vue (pas seulement les agents)
- ⇒ Généralisation:
  - Considérer tout composé comme un point de vue
  - Considérer les relations entre composés comme des articulations entre points de vue
- ⇒ Remarque:
  - Même considérer un ensemble d'objets(d'agents) est un point de vue!!!

# Dynamique

# Articulation ontologie/dynamique

- ⇒ L'articulation se fait au niveau 0
- ⇒ On peut spécifier un comportement dans chaque type d'entité



COSMAGEMS, Ontologies

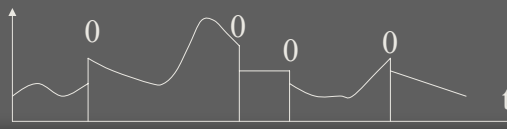
# Dynamique de base: DEVS atomique

QuickTime™ et un  
décompresseur TIFF (LZW)  
sont requis pour visionner cette image.

COSMAGEMS, Ontologies

# Propriétés

- ⇒ Comportements pro- et réactifs
- ⇒ La réaction à un événement est toujours avec retard (contrainte physique)
- ⇒ Le temps est spécifié par intervalles
  - Dans X temps, faire Y
- ⇒ La nature d'un état n'est pas spécifié
  - Peut être un fonction continue du temps



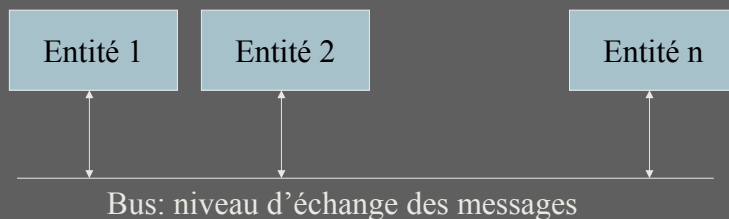
COSMAGEMS, Ontologies

# Protocole //-DEVS étendu

- ⇒ Initialisation:
  - void instanceInitialize()
  - void initialize()
- ⇒ Transitions pures:
  - void internalTransition(interval)
  - Influences outputTransition(interval)
  - Influences externalTransition(interval, influences)[]
  - Influences logicalTransition(interval, influences)
- ⇒ Transitions confluentes:
  - Influences confluentInternalTransition(interval, influences)
  - Influences confluentOutputTransition(interval, influences)
- ⇒ Comportement spontané:
  - InternalInfluence nextTransition(interval)[]

COSMAGEMS, Ontologies

# Couplage d'entité



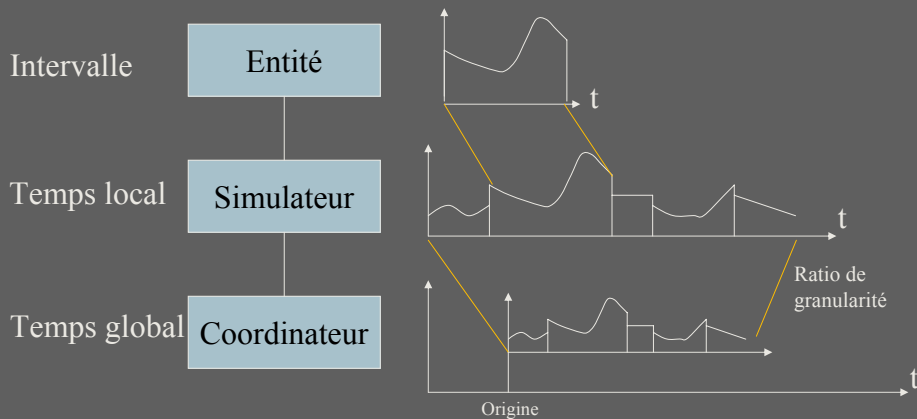
Entité: unité exécutable/communicante

Modèle: un ensemble d'entités couplées

# Extension

- ⇒ //-DEVS:
  - Prise en compte des évènements simultanés
- ⇒ Observation de l'état:
  - Deux possibilités:
    - Observation comme mesure physique
    - Observation absolue: introduction d'évènements instantanés: les événements logiques ←
- ⇒ Chaque entité a une ligne de temps propre (origine, granularité => échelle)
- ⇒ Les entités peuvent être créées/détruites dynamiquement

# Architecture de simulation



# Les temps

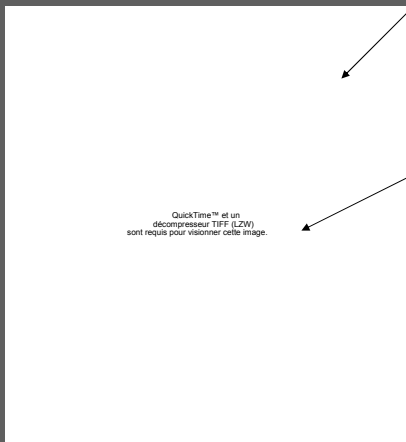
- ⇒ Le temps du calcul
  - L'ordinateur exécute les entités dans un certain ordre
- ⇒ Le temps mathématique
  - Les événements sont annotés par une valeur croissante au fil du calcul
  - MIMOSA garantit la cohérence du temps mathématique local et global
- ⇒ Le temps simulé
  - On ajoute des unités aux valeurs croissantes
  - Interprétation en termes de temps « réel »

# Protocole //-DEVS étendu

- ⇒ Initialisation:
  - void instanceInitialize()
  - void initialize()
- ⇒ Transitions pures:
  - void internalTransition(interval)
  - Influences outputTransition(interval)
  - Influences externalTransition(interval,influences)[]
  - Influences logicalTransition(interval,influences)
- ⇒ Transitions confluentes:
  - Influences confluentInternalTransition(interval, influences)
  - Influences confluentOutputTransition(interval, influences)
- ⇒ Comportement spontané:
  - InternalInfluence nextTransition(interval)[]

# Spécification des comportements

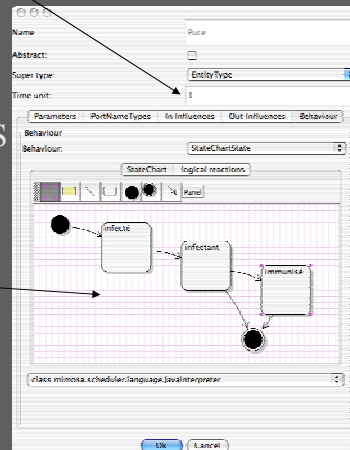
Ratio de granularité



QuickTime™ et un décompresseur TIFF (LZW) sont requis pour visionner cette image.

Spéc. DEVS

Spéc. diagramme d'états



# Conclusion et perspectives

## Conclusion (1)

⇒ MIMOSA est:

- Une architecture de spécification: méta-ontologies, ontologies, dynamiques, états initiaux, visualisation
- Un éditeur générique:
  - Des modèles conceptuels et donc des ontologies thématiques
  - Des modèles instanciés en vue de leur simulation
  - Des visualisations des simulations
- Une architecture d'intégration multi-modèles

## Conclusion (2)

- ⇒ Niveau 0:
  - Méta-ontologie standard minimale
- ⇒ Niveau 1:
  - Méta-ontologie ensembliste comme interprétation du niveau 0
- ⇒ Niveau 2:
  - Méta-ontologies spécifiques aux domaines

## Conclusion (3)

- ⇒ Processus de modélisation:
  - Spécification des méta-ontologies et ontologies:
    - Composants, composés et relations
  - Spécification de la dynamique:
    - Etats, évènements
  - Spécification du temps:
    - Les ratio de granularité, les unités
  - Spécification de la visualisation:
    - Les grandeurs mesurables d'une entité (sondes), les mises en correspondances avec des visualisations

# Perspectives (1)



- ⇒ Niveau 0:
  - Distribution et spécification de la distribution
  - Implémentation d'un ensemble d'entités d'observation/visualisation
  - Méthodes d'initialisation:
    - A partir de bases de données
    - A partir de cartes (cf. GEAMAS)
- ⇒ Niveau 1:
  - En cours de finalisation
  - Exploitation de la structure ensembliste elle-même
- ⇒ Niveau 2:
  - Analyse et implémentation de méta-ontologies exhaustives
    - Espace
    - Temps
    - Agents
    - Géométrie 2D et 3D

# Perspectives (2)



- ⇒ Ontologies des dynamiques:
  - Automates d'états finis
  - Diagrammes d'états/transitions (StateChart)
  - Réseau de Pétri
  - Équations différentielles
  - ...d'une manière générale, « mapping » des formalismes connus dans DEVS