

# Faciliter et réguler les interactions: une responsabilité de l'environnement

J. Saunier<sup>1</sup>, F. Balbo<sup>1,2</sup>

<sup>1</sup>LAMSADE, Université Paris-Dauphine

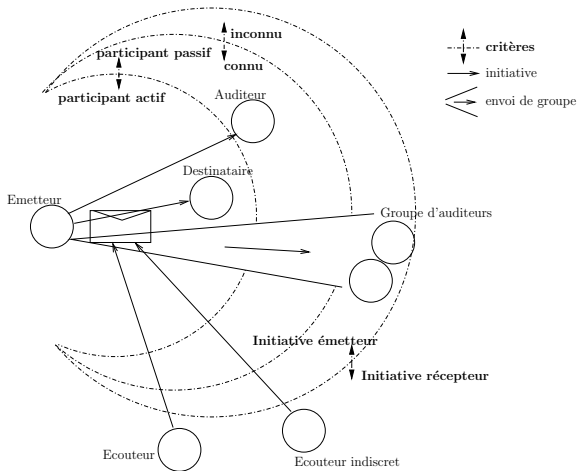
<sup>2</sup>INRETS GRETIA, Institut National de REcherche sur les Transports et leur Sécurité

Groupe de travail D2A2

# Sommaire

- 1 Positionnement
  - Communications Multi-parties
  - Awareness
- 2 Environnement de communication
  - Présentation générale
- 3 Régulation et politiques
  - Rôle de l'environnement
  - Filtres négatifs
  - Politiques
- 4 Conclusion

# Communications Multi-Parties



## Critères

- Intention
- Connaissance
- Rôle attendu
- Initiative

# Notion d'*awareness*

Conscience (des autres), sensibilité, connaissance, attention?

## Environnement, Stimuli

- Etat
- Résultat de stimuli
- Vecteur: environnement
- Non choisi

→ L'*awareness* est subi

## Décisions d'action

- Activité de perception
- Organe/capteur
- Action
- Choisi

→ L'*awareness* est actif

⇒ L'*awareness* est une conjonction de stimuli non désirés, de capacités de perception et de décision d'actions.

Ces décisions peuvent être contradictoires avec les sollicitations de l'environnement

# Property-Based Coordination

Une interaction a lieu:

- lorsqu'un certain nombre de **conditions** sur l'**état du SMA** sont réunies
- L'**état du SMA** est représentable par une description des entités le composant.
- Les conditions peuvent provenir de règles de l'environnement, ou exprimer des besoins des agents

Ce calcul nécessite une connaissance de l'état du SMA et des conditions. Notre modélisation propose d'effectuer cela par une médiation de l'interaction via un environnement explicite.

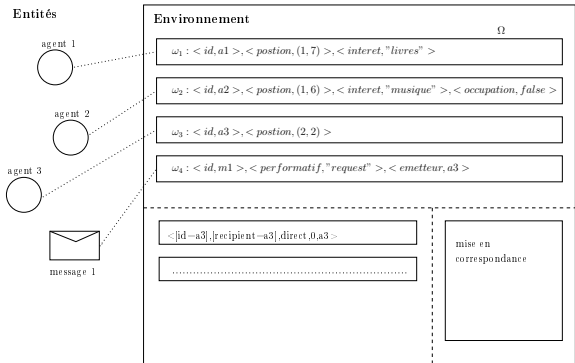
# Environnement (Environment as Active Support of Interaction)

Formalisation issue de l'analyse de données symboliques:

- classification
- expressivité

L'environnement est composé de:

- $\Omega$  l'ensemble des entités
- $\mathcal{F}$  l'ensemble des filtres



# Description des entités

## Agents

### Propriétés

- Générales: capacités, préférences...
- Contexte: occupation, intérêt ponctuel...
- Changeantes: localisation...

## Objets d'interaction

### Exemple

- Messages
- Traces
- Drapeaux

### Propriétés

- Message: Sujet, performatif...
- Contexte: Emetteur, destinataire, date, localisation...

Exemple: FIPA-ACL

## Objets

## Autres objets

### Exemple

- Ressources
- Interfaces

### Propriétés

- Informatif: accessibilité, position...

# Filtres - intuition

## Objectif

Appariement des objets d'interaction avec les agents

## Comment?

- Conditions sur des propriétés  
Exemple:  $[available(a) = true]$  ou  $[performative(io) = "request"]$
- Trois ensembles de conditions:
  - Agent
  - Donnée
  - Contexte (facultatif)
- Appariement de valeurs entre les entités
- Méta informations sur le filtre: nom, priorité, initiateur...



# Filtres

## Définition

Un filtre  $f \in \mathcal{F}$  est un tuple  $\langle f_{ag}, f_{pe}, [f_{co}], n_f, [priorité], [initiateur] \rangle$  où:

- $f_{ag}$  est la description en intention de l'agent récepteur telle que:  
 $a \in A, f_{ag}(a) = \bigwedge_{p_i \in P_{f_{ag}}} [p_i(a) R_{p_i}^{f_{ag}} d_{p_i}^{f_{ag}}]$ .
- $f_{pe}$  est la description en intention du percept telle que:  
 $\omega \in \Omega, f_{pe}(\omega) = \bigwedge_{p_i \in P_{f_{pe}}} [p_i(\omega) R_{p_i}^{f_{pe}} d_{p_i}^{f_{pe}}]$ .
- $f_{co}$  est la description en intention (optionnelle) du contexte telle que:  
 $C \subset \Omega, f_{co}(C) = \bigwedge_{\forall c \in C} f_{co}(c)$ , avec  
 $f_{co}(c) = \bigwedge_{p_i \in P_c} [p_i(c) R_{p_i}^c d_{p_i}^c]$ .
- $n_f$  est le nom du filtre.
- *priorité* est la priorité du filtre.
- *initiateur* est l'initiateur du filtre.

# Filtres – Exemple

## Communications dyadiques

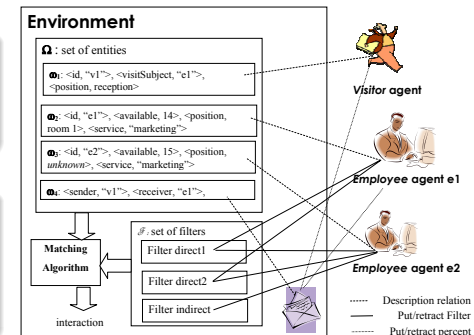
- Récepteur:  $[id(a) = ?x]$
- Message:  $[receiver(io) = ?x]$

## Orienté données

- Récepteur:  $[id(a) = 13542]$
- Message:  $[pos(io) \in MR] \wedge [sub(io) = "available"]$

## Interception

- Récepteur:  $[id(a) = 13542]$
- Message:  $[sender(io) = 25554]$



## Adressage par propriété

- Récept:  $[dep(a) = ?x] \wedge [ava(a) = true]$
- Message:  $[dep(io) = ?x]$

# Rôle de l'environnement

## Filtres des agents

⇒ Expression des besoins des agents - influences

## Filtres de l'environnement

⇒ Définir un comportement global - règles

- Décharger les agents de cette tâche
- Aide à la conception
- Assurer à l'exécution la transmission des messages

# Filtres négatifs

## Problème

Les filtres permettent d'ajouter des perceptions. Dans les systèmes ouverts et hétérogènes, cela présente des risques: agents malveillants, mal conçus...

Définir des règles de l'environnement, positives et négatives: capacité de transmission....

Donner aux agents le moyen d'exprimer leurs décisions d'action relatives à leur *awareness*

## Réguler

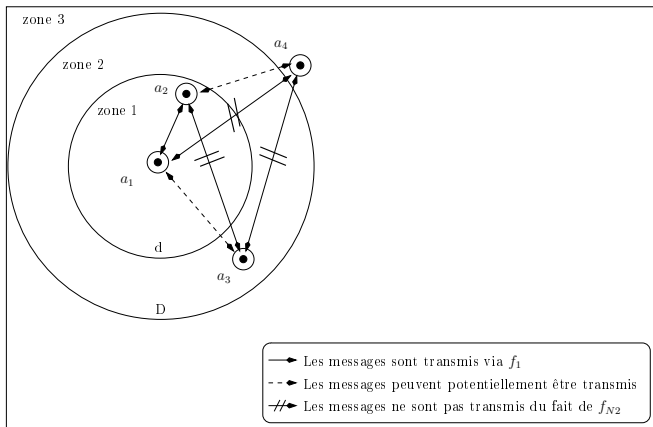
Par le même système de définition de filtres, il s'agira donc de bloquer la perception au lieu de la créer.  
Résolution à l'exécution de l'algorithme.

# Algorithme d'appariement modifié

- Créer une liste contenant  $FPer_a \cap Channel_{io} \cap FContext_C$ , triée par ordre de priorité décroissante
- Effectuer le même algorithme que précédemment, sur l'ensemble des percepts, agents, contextes et filtres
- Lorsqu'un filtre est valide pour un agent et un percept:
  - Si le filtre est positif, transmettre le message et passer au filtre suivant
  - Si le filtre est négatif, passer au couple agent/percept suivant

Le choix de la priorité joue donc un rôle primordial sur la façon dont se déroule l'algorithme.

# Exemple: environnement spatial

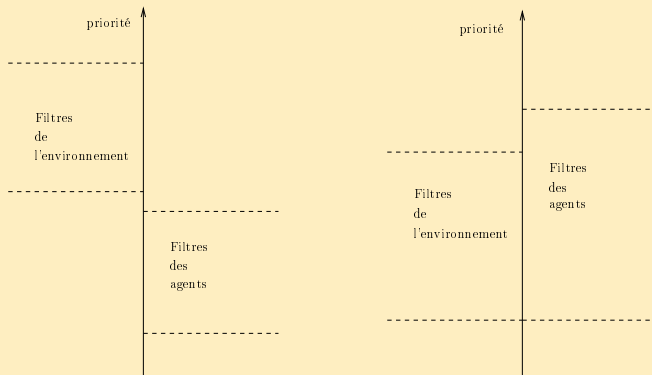


$$\langle [dist(a, a_2) < d], [sender(io) = ?x], [id(a_2) = ?x], "proxim", p_1, env \rangle \in \mathcal{F}$$

$$\langle [dist(a, a_2) > D], [sender(io) = ?x], [id(a_2) = ?x], "eloign", p_2, env \rangle \in \mathcal{F}_N$$

# Exemples de politiques: Environnement

## Prédominance de l'environnement - Environnement facilitateur



# Filtres personnels

## Problème

Que faire si un agent tente de bloquer les perceptions d'un autre?

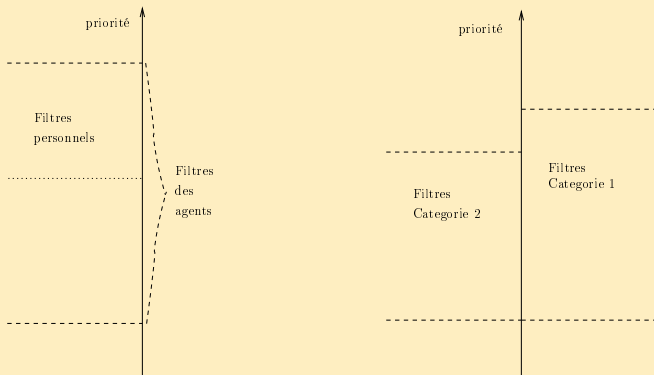
## Définition

$f$  est un *filtre personnel* de l'agent  $a_x$  si et seulement si  $initiator_f = a_x$  et  $E(f_{ag}) = \{a_x\}$

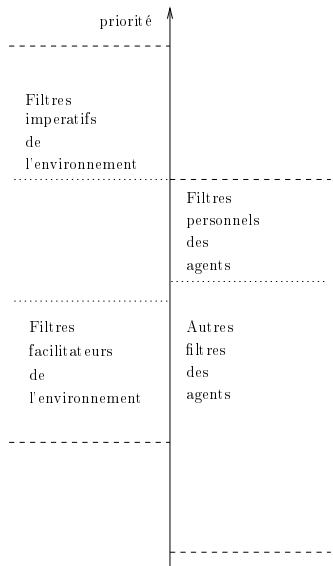


# Exemples de politiques: Agents

## Prédominance personnelle - Politique différenciée



# Exemple de politique: Un environnement “naturel”



- Soumission aux règles inviolables de l'environnement
- Les agents définissent leurs propres décisions d'action
- Règles de fonctionnement normal de l'environnement d'interaction
- Autres sollicitations

# Conclusions

- L'environnement: abstraction explicite pour gérer les interactions
- Un modèle à base de filtres expressifs pour les normes et règles du SMA
- Mise en place de politiques de régulation
- Un modèle implémentable

# Perspectives

- **Politiques de régulation-** Introduction de la logique déontique dans la formalisation des filtres
  - vérification à la conception
  - validation à l'exécution
- **Algorithmes-** Algorithmes de stockage des informations (descriptions des entités et filtres) pour optimiser leur traitement  
Différencier le traitement des propriétés en fonction du taux de mise à jour
- **Analyse de Données Symboliques-** Introduction des algorithmes de classification de données de l'ADS
  - Retour d'analyse *a posteriori* pour le concepteur du déroulement de l'exécution du SMA
  - Aide à l'exécution pour optimiser la classification des entités, surveillance à partir de données agrégées

# Parutions I



M. Zargayouna, J. Saunier, and F. Balbo.

Property based coordination.

In I. J. Euzenat and J. Domingue, editors, *Artificial Intelligence: Methodology, Systems, Applications*, volume 4183 of *Lecture Notes in Artificial Intelligence*, pages 3–12. Springer Verlag, 2006.

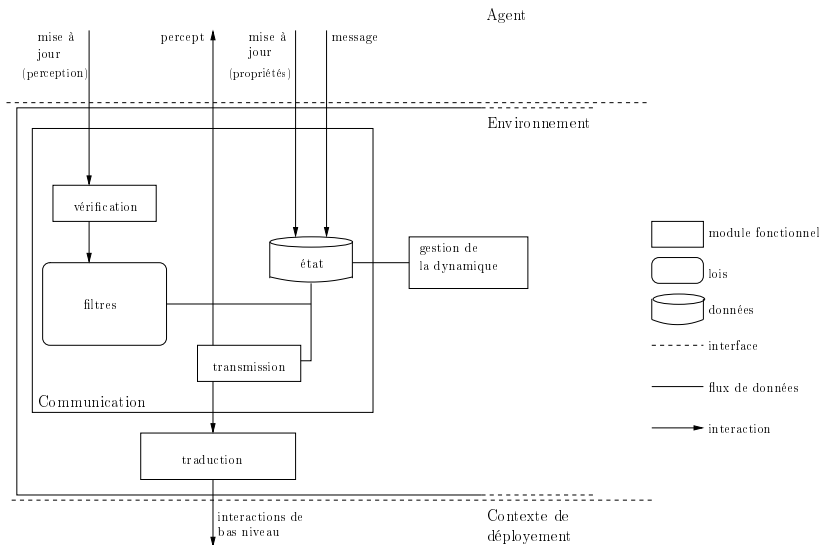


J. Saunier, F. Balbo, and F. Badeig.

Environment as active support of interaction.

In *Proceedings of Environment for Multi-Agent Systems, Workshop held at the Fifth Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 4389 of *LNAI*, pages 87–105. Springer Verlag, 2007.

# Architecture



# Algorithmes spécialisés I

## Idée

Utiliser les informations structurelles  
Restreindre les ensembles

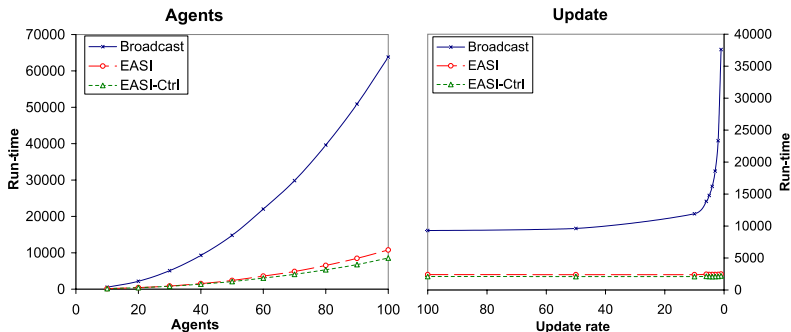
## Informations / Exemple

$Channel_{io}$ : ensemble des filtres  $f$  liés au percept  $io$  tel que cet  $io$  appartient à l'extension de chaque  $f$

Nom	Définition
$Channel_{io}$	$\{f \in \mathcal{F} \mid io \in E(P_{f_{pe}})\}$
$Receiver_{io}$	$\{a \in A \mid \exists f \in Channel_{io},$ $a \in E(P_{f_{ag}})\}$
$Context_{io}$	$\{C \subset \Omega \mid \exists f \in Channel_{io},$ $C \in E(P_{f_{co}})\}$
$FPer_a$	$\{f \in \mathcal{F} \mid a \in E(P_{f_{ag}})\}$
$FContext_C$	$\{f \in \mathcal{F} \mid C \in E(P_{f_{co}})\}$

Existence des propriétés: statique

# Algorithmes spécialisés: tests (filtres négatifs)



- Intelligence ambiante
- Broadcast / algorithmes
- Nombre agents et mise à jour
- Etude centralisée

⇒ Pas de surcoût dû aux filtres négatifs (coupe circuit)  
 ⇒ Globalement meilleur que la diffusion