

Méthodologie de développement de systèmes multi-agents adaptatifs

Conception de logiciels à fonctionnalité émergente

Gauthier Picard

Thèse en Informatique
Institut de Recherche en Informatique de Toulouse (IRIT)
Université Paul Sabatier (UPS)

10 décembre 2004

UNIVERSITE
PAUL
SABATIER



TOULOUSE III

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
- 3 La méthode ADELFE
- 4 Emploi du temps adaptatif
- 5 Collectif de robots transporteurs
- 6 Conclusion

Introduction

Motivations & Orientations

Constat

- Environnements des logiciels de plus en plus complexes (ouverture, indéterminisme, dynamique, ...)
- Fonctionnalité difficile à spécifier et besoin d'adaptation globale (non-linéarité, organisation inconnue, ...)
- Les systèmes sont décomposables et leurs parties spécifiées

Inspiration naturelle ou artificielle

- Phénomènes *émergents* et mécanismes d'*auto-organisation*
- Systèmes *multi-agents*
- La *coopération* comme mécanisme de micro-niveau

Vers une méthodologie plus adaptée

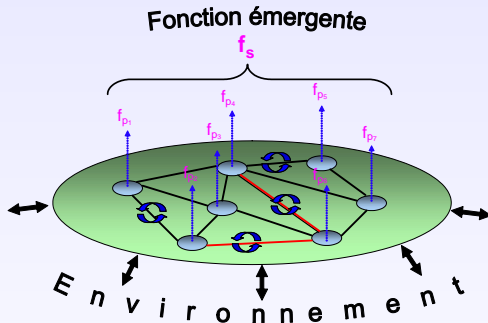
- De telles techniques sont trop peu nombreuses et peu connues
- Besoin de méthodes orientées émergence ➔ **ADELFE**

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
 - Émergence et auto-organisation
 - Les systèmes multi-agents adaptatifs
 - Ingénierie orientée agent
 - Besoins pour les AMAS
- 3 La méthode ADELFE
- 4 Emploi du temps adaptatif
- 5 Collectif de robots transporteurs
- 6 Conclusion

Auto-organisation

Un mécanisme de l'émergence



- 1 organisation \Rightarrow 1 fonction
- Changer d'organisation \Rightarrow changer de fonction
- Adaptation et apprentissage
- **Besoin d'un critère *local* de réorganisation non *explicitement* informé de la connaissance de la fonction globale**

Les systèmes multi-agents adaptatifs (AMAS)

La coopération comme moteur de l'auto-organisation

Adaptation par auto-organisation coopérative (Équipe SMAC)

- L'auto-organisation est un moyen de construire des systèmes *adaptatifs*
- Le critère d'auto-organisation est la *coopération*

Coopération

Un agent est coopératif si C_{per} et C_{dec} et C_{act} :

C_{per} Les signaux perçus par l'agent sont compris sans ambiguïté

C_{dec} L'information reçue est utile au raisonnement de l'agent

C_{act} Le raisonnement de l'agent mène à des actions utiles aux autres

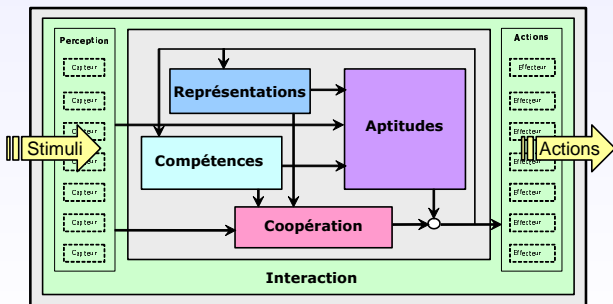
➔ Approche proscriptive : les agents, pour bien fonctionner, doivent éviter et résoudre les situations non coopératives (SNC) : $\neg C_{per}$ ou $\neg C_{dec}$ OU $\neg C_{act}$

Les systèmes multi-agents adaptatifs (AMAS)

Un modèle d'agent coopératif

Architecture

- Cycle de vie "perception-décision-action"
- Plusieurs modules représentent une partition des capacités physiques, cognitives ou sociales
- Attitude sociale coopérative



Les systèmes multi-agents adaptatifs (AMAS)

Méthodologie

Adéquation fonctionnelle

Pour tout système fonctionnellement adéquat dans un environnement donné, il existe au moins un système ayant un milieu intérieur coopératif qui réalise une fonction équivalente

Conséquences

- Conception ascendante privilégiée
- Le comportement global du système émerge des interactions coopératives entre les agents du système

➔ Comment concevoir de tels agents et systèmes ?

➔ Besoin d'une méthode de conception : le projet ADELFE

Ingénierie orientée agent

État de l'art

Influences

- Le paradigme *objet* : A-UML, AOR, ... [Odell, 2002]
- La *nature* : modèles sociaux et organisationnels
- L'*ingénierie des connaissances* : KADS ou CommonKADS [Iglesias et al., 1998]
- L'*ingénierie des besoins* : i^* [Yu and Mylopoulos, 1994]
- L'*ingénierie orientée plate-forme* : Zeus [Nwana et al., 1999]

➔ Beaucoup de méthodes existent, mais...

Ingénierie orientée agent

État de l'art

Caractéristiques des méthodes [Shehory and Sturm, 2001]

- Les concepts clés et propriétés
 - ✓ Interactions et rôles souvent abordés
 - ✗ Adaptation et environnement peu ou pas pris en compte
- Les notations et langages de modélisation
 - ➔ Modularité avec UML
 - ➔ Précision avec approches logiques
- Le processus
 - ✓ Analyse et conception détaillées
 - ✗ Peu de ponts vers le développement et le déploiement
- La pragmatique
 - ✗ Trop peu de ressources disponibles
 - ✗ Trop peu de réelles applications développées

Besoins pour les AMAS

Analyse de l'existant

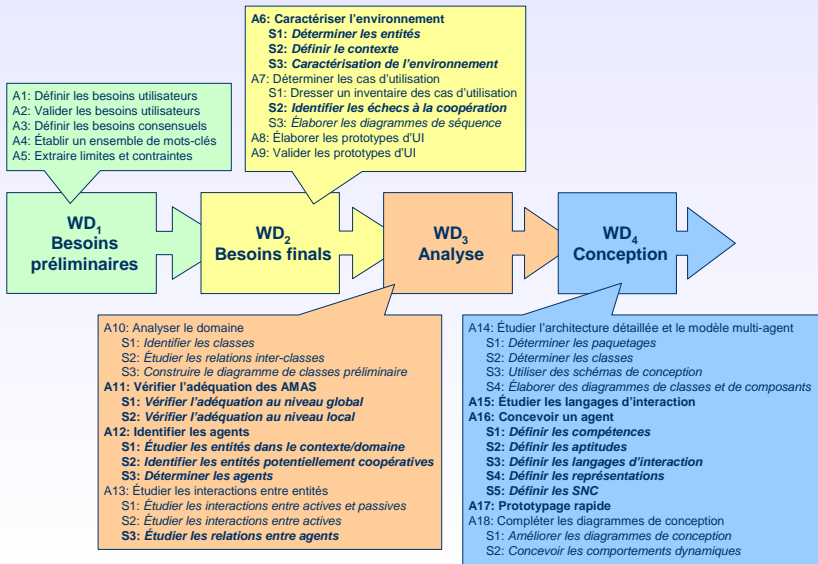
- *Doit-on s'intégrer dans une des méthodes existantes ou repartir de zéro ?*
 - Méthodes spécifiques à des architectures (BDI, FIPA, ...)
 - Organisations fixes et fermées
 - *Mais* méthodes objets assez génériques pour s'ouvrir aux agents (ex : RUP et MESSAGE)
 - ➔ **Oui**, intégration dans une méthode objet existante
- *Quels notations et langages doit-on adopter/créer/réutiliser ?*
 - Compromis difficile entre précision (logique) et expressivité (UML)
 - *Mais* l'agent est vu comme une extension de l'objet
 - ➔ **UML** et son extension orientée agent, A-UML
- *Existe-t-il des outils manipulant les notions avancées ou facilement modifiables/réutilisables ?*
 - Peu de méthodes sont munies d'outils
 - Les outils ne sont d'aucune aide pour le suivi du processus
 - *Mais* des outils de conception objet ouverts
 - ➔ **OpenTool**, et d'autres si nécessaire

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
- 3 La méthode ADELFE**
 - Le processus de développement
 - Les notations – UML et A-UML
 - Les outils
- 4 Emploi du temps adaptatif
- 5 Collectif de robots transporteurs
- 6 Conclusion

ADELFE

Processus - modélisation en SPEM



ADELFE

Processus

Particularités & nouveautés

- *Analyse de l'environnement* en suivant la classification de [Russel and Norvig, 1995]
- *Adéquation des AMAS* mise en question
- *Identification des agents* par traçage des flots à risque
- *Étude des langages d'interactions* par protocoles AIP
- *Conception des agents* par implémentation d'un modèle d'agent coopératif
- *Prototypage rapide* des comportements sociaux des agents

ADELFE

Notations : stéréotypage

Intérêt

- Attachement de règles de cohérence à l'utilisation de classes du méta-modèle UML
- Intégration simple dans la plupart des outils OO
- Modification *en surface* du métamodèle UML

Les stéréotypes d'ADELFE

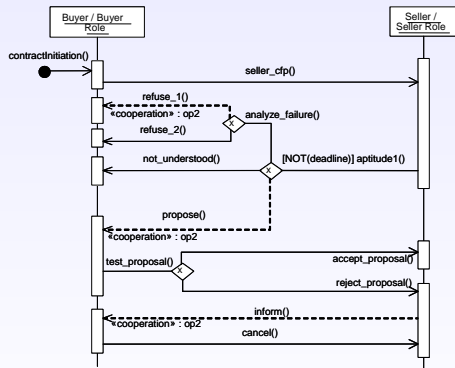
- 1 pour les classes d'agents coopératifs : «cooperative agent»
- 1 pour les propriétés intrinsèques : «characteristic»
- 1 par module : «interaction», «perception», «action», «representation», «skill», «aptitudes»
- 1 pour les règles de coopération : «cooperation»

ADELFE

Notations : protocoles

Protocoles A-UML étendus

- Représentation des interactions sociales
- Diagrammes de séquence génériques d'UML 1.4
- Notion de rôle encapsulée dans le protocole
- **Branchements plus riches**
- **Coopération lors de la réception**

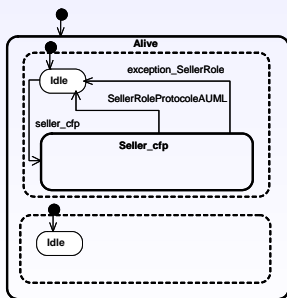
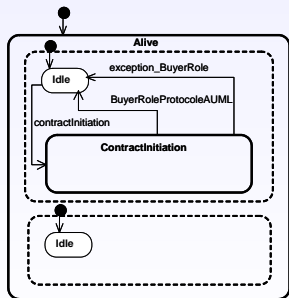


ADELFE

Notations : machines à états finis

Utilisation

- Modélisation du comportement dynamique des agents
- Modélisation des interactions, par transformation de protocoles
- Machines concurrentes pour protocoles concurrents
- Simulation possible dans OpenTool



ADELFE

Outils

Support des notations : OpenTool

- Étendu aux notations d'ADELFE
- Transformation automatique des protocoles en machines à états
- Simulation de machines à états

The screenshot displays the OpenTool interface. On the left, a project tree shows a 'new' folder containing 'Class1 (agent)', 'Class2', and 'Standard'. The main workspace shows a class diagram with three classes: 'Class1 (agent)', 'Class2', and 'CooperativeAgent (Standard)'. 'Class1 (agent)' is a stereotype with a dashed red border and contains a 'representation' compartment with methods '- op2 ()', '+ att: Undefined', '- op1 ()', '- perceve ()', '- decide ()', and '- act ()'. 'Class2' has methods '+ att1: Undefined' and '- op1 ()'. 'CooperativeAgent (Standard)' is a base class. A 1-to-1 association labeled '<characteristic>' connects 'Class1 (agent)' and 'Class2'. A bottom status bar shows 'Class1 (agent) (new)' and a 'Form - Classe: Class1 (agent) <new>' window with error messages: '1 Une classe agent doit avoir au moins un attribut ou une opération (locale) ou héritée(e)) pour chacun des stéréotypes' and '1 Un agent doit hériter directement ou indirectement de la classe CooperativeAgent'.

ADELFE

Outils

ADELFE
Atelier de Développement de Logiciels à Fonctionnalité Emergente

Concevoir des logiciels qui sont capables de s'adapter à un environnement dynamique implique une méthodologie de conception rigoureuse qui doit être différente des approches descendantes classiques.

ADELFE est un projet financé par le Réseau National des Technologies Nouvelles (RNTL - <http://www.telecom.gouv.fr/rntl/>) qui s'est déroulé de décembre 2000 à août 2003.

Les partenaires de ce projet sont :

- Actal Technologies (<http://www.actal.fr/>) ;
- L'Institut de Recherche en Informatique de Toulouse (<http://www.irit.fr/>) pour l'université Paul Sabatier ;
- La Laboratoire Informatique, Image, Interaction (<http://www.univ-lr.fr/labo3i5/>) de l'université de La Rochelle ;
- TRÉ-VALOISYS (<http://www.tré-valoisys.com/>).

ADELFE V1.2
Atelier de Développement de Logiciels à Fonctionnalité Emergente

Aide

Les liens, hypertextes ou le texte sont colorés de différentes manières selon l'information que l'on peut acquérir en les suivant ou le lisant :

- Le bleu est utilisé pour marquer un lien que vous devrez à suivre la méthodologie ;
- Le violet indique un lien vers un mot du glossaire ;
- Le vert indique un lien vers la définition d'un acronyme utilisé dans la méthodologie.

Support du processus : AdelfeToolkit

- Livre interactif présentant le processus
- Raccourcis vers les logiciels de conception
- Adéquation des AMAS

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
- 3 La méthode ADELFE
- 4 Emploi du temps adaptatif**
 - Présentation
 - Les agents dans ETTO
 - Coopération
 - Expérimentations
 - Analyse et bilan
- 5 Collectif de robots transporteurs
- 6 Conclusion

ETTO

Présentation

Cahier des charges

- Solveur dynamique d'emplois du temps universitaires
- Les contraintes des acteurs sont modifiables à tout moment
- Les résolutions ne doivent pas se reconstruire à chaque modification

- ➔ AMAS pertinents en terme de réorganisation dynamique
- ➔ Emergent Time Tabling Organisation

Les RepresentativeAgents

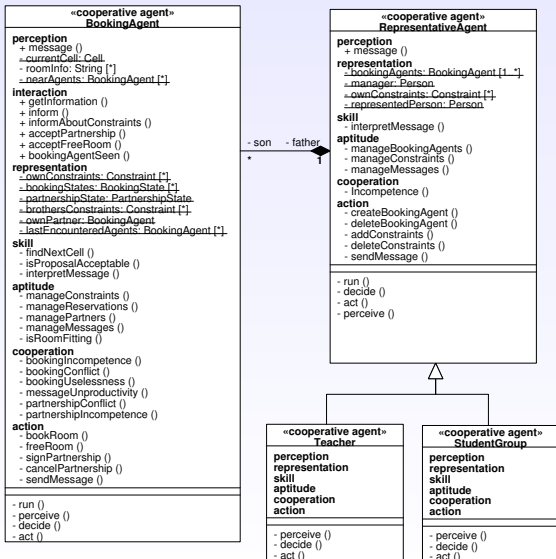
- Représentent les acteurs humains (enseignants, étudiants)
- Regroupent les contraintes de disponibilité, d'équipements, ...
- Délèguent la recherche de créneaux et de salles aux BookingAgents

Les BookingAgents

- Représentent un enseignement d'un acteur (à suivre ou à donner)
- Réservent les créneaux horaires et les salles de manière coopérative
- La cohérence de leurs réservations est assurée par leur RepresentativeAgent

ETTO

Les agents



ETTO

Coopération des BookingAgents

	Pas de réservation		Avec réservation	
	Pas de partenaire	Avec partenaire	Pas de partenaire	Avec partenaire
Perception	Aucune SNC lors de la phase de perception			
Décision	<ul style="list-style-type: none"> • Incompétence Incompétence de partenariat • Improductivité Improductivité de message 	<ul style="list-style-type: none"> • Incompétence Incompétence de partenariat • Improductivité Improductivité de message 	<ul style="list-style-type: none"> • Incompétence Incompétence de partenariat Incompétence de réservation • Improductivité Improductivité de message 	<ul style="list-style-type: none"> • Incompétence Incompétence de partenariat Incompétence de réservation • Improductivité Improductivité de message
Action	<ul style="list-style-type: none"> • Conflit Conflit de partenariat Conflit de réservation 	<ul style="list-style-type: none"> • Conflit Conflit de partenariat Conflit de réservation • Inutilité Inutilité de réservation 	<ul style="list-style-type: none"> • Conflit Conflit de partenariat Conflit de réservation 	<ul style="list-style-type: none"> • Conflit Conflit de partenariat Conflit de réservation • Inutilité Inutilité de réservation

ETTO

Coopération des BookingAgents

SNC = condition de réorganisation

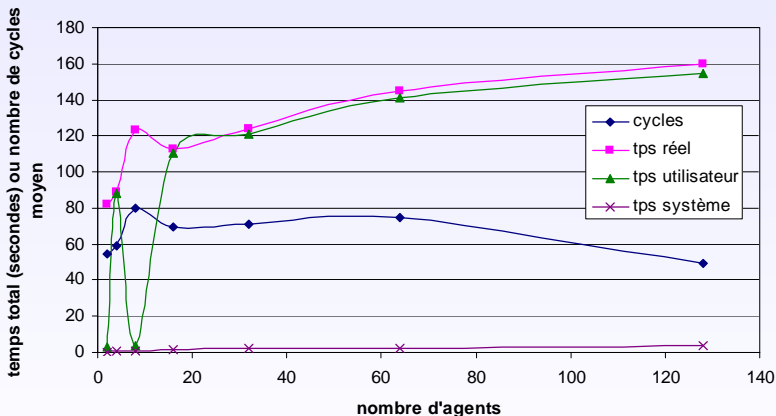
SNC → action de résolution

Les 6 SNC dans ETTO

- Incompétence de partenariat
 - ↳ déplacement + mémorisation
- Incompétence de réservation
 - ↳ déplacement + mémorisation
- Improductivité de message
 - ↳ retour du message
- Conflit de partenariat
 - ↳ négociation (déplacement ou partenariat)
- **Conflit de réservation**
 - ↳ négociation (déplacement ou réservation)
- Inutilité de réservation
 - ↳ déplacement

ETTO

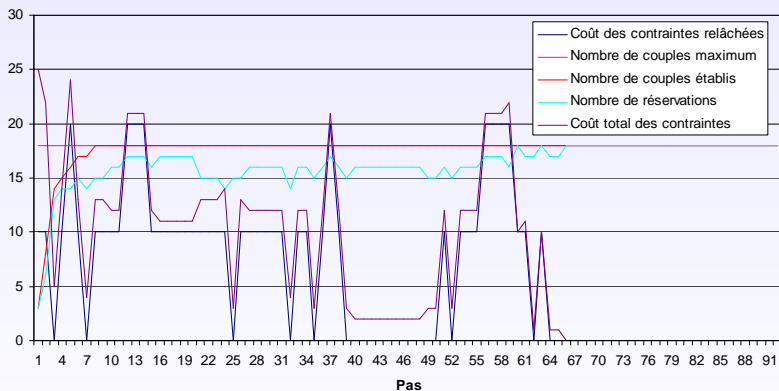
Expérimentations



Variation du temps de résolution en fonction du nombre d'agents

ETTO

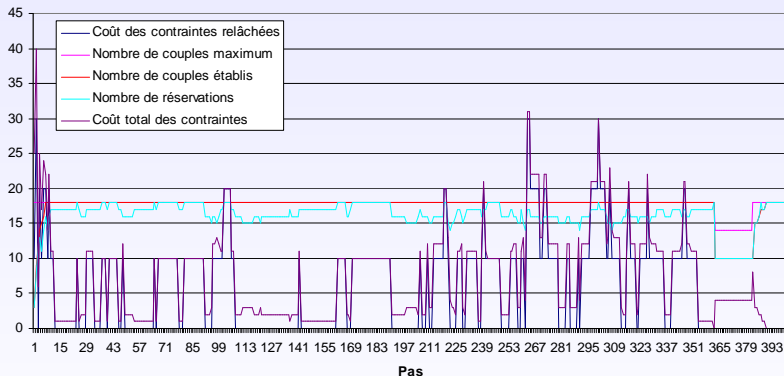
Expérimentations



Variation des contraintes en cours de résolution

ETTO

Expérimentations



Variation des contraintes en cours de résolution, avec suppression de 8 BookingAgents après stabilisation du système

ETTO

Expérimentations

ETTO

Analyse et bilan

- Application d'ADELFE à un problème d'emploi du temps dynamique
- Didacticiel d'ADELFE
- Émergence d'une organisation adéquate

Remarques

- Le système ne possède aucun critère d'arrêt
- ✓ L'ajout d'agents surnuméraires peut favoriser l'amélioration de la solution
- ✓ Les actions à mener sont relativement simples
- ✓ ADELFE a aidé à identifier les deux niveaux d'agents
- ✗ Les agents de réservation ne sont pas efficaces dans leur exploration de la grille
- ✗ Seules les modifications concernant les agents ont été implémentées car elles étaient prioritaires
- ✗ ETTO n'a pas été comparé à d'autres approches, plus classiques

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
- 3 La méthode ADELFE
- 4 Emploi du temps adaptatif
- 5 Collectif de robots transporteurs**
 - Modules
 - Décision des actions
 - Comportement coopératif
 - Expérimentations et résultats
 - Analyse et bilan
- 6 Conclusion

Collectif de robots transporteurs

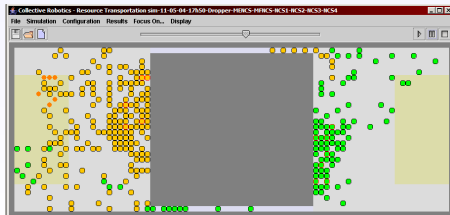
Présentation

Robots

- Autonomes
- Tâche de transport de ressources
- Entités de micro-niveau
- **Non communicants**

Environnement

- 2 salles
- n couloirs étroits séparent les salles



➔ **Interférences spatiales**

➔ Focalisation sur la conception

Robot transporteur «cooperative agent»

Modules & Stéréotypes

- «perception» : cône, proximité, types, position
- «action» : *rest, pick, drop, forward, backward, left* et *right*
- «skill» : préférences sur les actions nominales
- «characteristics» : poids transportable, vitesse, réflexes
- «representation» : position, direction, but et actions précédentes
- «aptitude» : méthodes qui choisissent la prochaine action nominale parmi les actions disponibles (Monte Carlo)
- «coopération» : méthodes choisissent la prochaine action parmi les actions disponibles pour être le plus **coopératif** + méthodes de détection des SNC

Décision des actions

Implémenter en comportement nominal

Comportement nominal – évaluation des actions

$$V_{r_i}^{nomi}(act_j, t) = wp_{r_i}(act_j, t) + wm_{r_i}(act_j, t) + wr_{r_i}(act_j)$$

- $wp_{r_i}(act_j, t)$: valeur calculée en fonction des perceptions
- $wm_{r_i}(act_j, t)$: valeur calculée en fonction de la mémoire
- $wr_{r_i}(act_j, t)$: valeur calculée en fonction des réflexes

➔ Un vecteur de préférences des actions est généré

Comportement coopératif – évaluation des actions

$$V_{r_i}^{coop}(act_j, t) = wp'_{r_i}(act_j, t) + wm'_{r_i}(act_j, t) + wr'_{r_i}(act_j)$$

Comportement composite de l'agent

$$V_{r_i}(t) = V_{r_i}^{nomi}(t) \prec V_{r_i}^{coop}(t)$$

Comportement coopératif

Coopération réactive & coopération anticipative

2 SNC de blocage (dans le module de coopération)

- *Un robot est bloqué*
 - ➔ "Se déplacer sur le côté" ou "Le robot le plus proche de son but l'emporte"
- *Un robot se retourne*
 - ➔ "Se déplacer sur le côté" ou "Continuer tout droit jusqu'à un blocage"

1 SNC d'anticipation (dans le module de coopération)

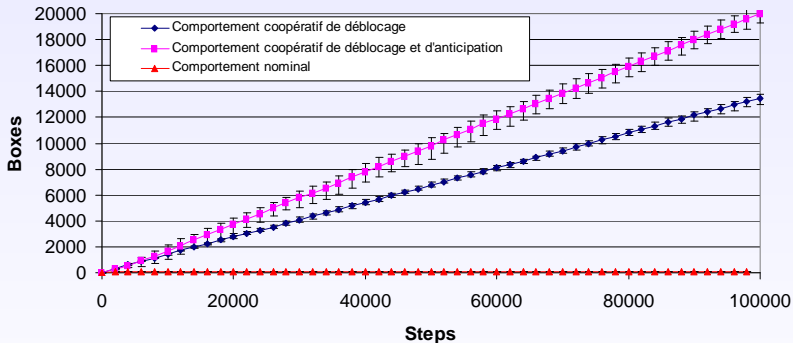
- Pour éviter les zones à risque
- *Un robot voit un robot antinomique*
 - ➔ "Se déplacer sur le côté" ou "Se déplacer en avant"

Mémoire (dans le module de représentation)

- Balises virtuelles : $\langle posX(r_j, t), posY(r_j, t), goal(r_i, t), w \rangle$

Expérimentations et résultats

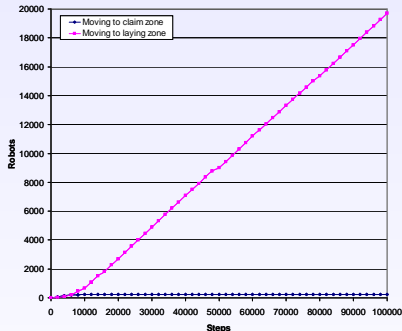
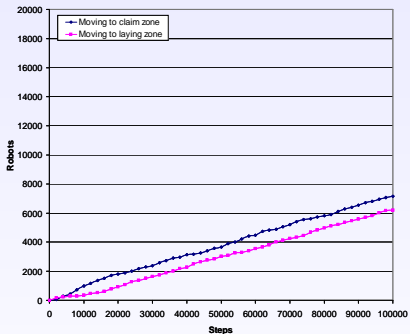
Réaction et anticipation



Nombre de boîtes transportées pour 15 simulations (300 robots, 2 couloirs)

Expérimentations et résultats

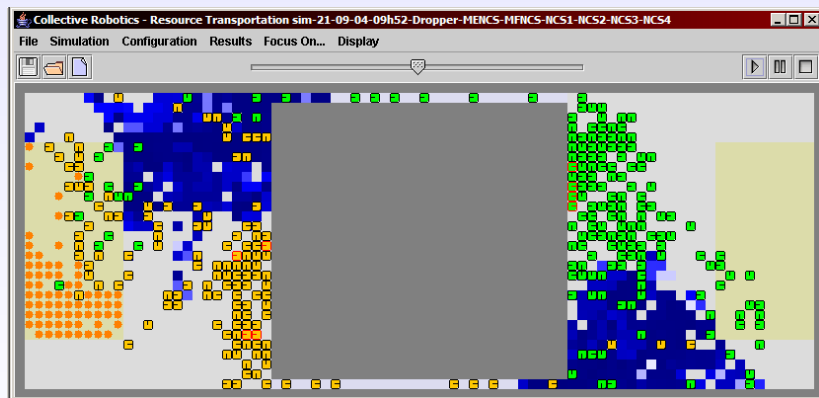
Émergence - Fréquentation



Nombre de robots entrant dans un couloir pour deux comportements coopératifs : déblocage réactif (à gauche) et déblocage avec anticipation (à droite)

Expérimentations et résultats

Émergence - Position des balises



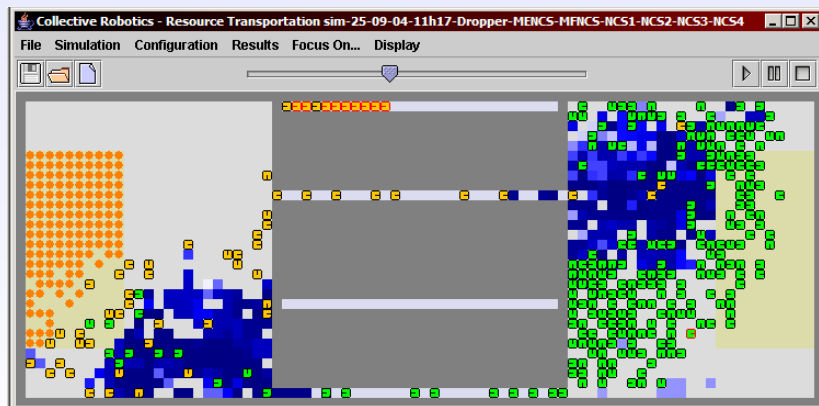
Position de toutes les balises de tous les robots (somme de toutes les mémoires individuelles)

Expérimentations et résultats

Émergence - Comportement global

Expérimentations et résultats

Émergence - Adaptation



Placement des balises virtuelles dans un environnement dynamique à quatre couloirs dont deux sont obturés, pour 300 robots

Collectif de robots transporteurs

Analyse et bilan

- Application d'ADELFE à un problème de transport de ressources multi-robot
- Émergence d'un comportement global efficace et adaptatif

Remarques

- ✓ ADELFE : conception modulaire
- ✓ Pas de communication
- ✓ Pas de mémoire partagée
- ✓ Comportement robuste et adaptatif
- ✓ Pas de feedback global
- ✗ Position absolue
- ✗ Les paramètres sont difficiles à déterminer
- ✗ ADELFE n'aide pas à trouver les instances de SNC, car problème de régulation, plus que de réorganisation

Sommaire

- 1 Introduction
- 2 Contexte et état de l'art
- 3 La méthode ADELFE
- 4 Emploi du temps adaptatif
- 5 Collectif de robots transporteurs
- 6 Conclusion**
 - Analyse d'ADELFE
 - Perspectives pour ADELFE
 - Bilan

Conclusion

Analyse d'ADELFE

Applications d'ADELFE

- ETTO, robotique collective
- SYNAMEC, MicroMéga, Adaptive JRules

↳ Enseignements

Points forts & points faibles

- ✓ Concepts : *adaptation, ouverture*, interaction et réactivité
- ✓ Notations : exécution et modularité
- ✓ Processus : analyse, conception, *délivrables* et *gestion*
- ✓ Pragmatique : *ressources*
- ✗ Manque de définition des concepts
- ✗ Défauts de l'approche Objet/UML
- ✗ Manque de formalisme

Conclusion

Perspectives pour ADELFE

Vers plus de formalisme

- Tentatives passées : B, ensembles, ...
- Automates formels

Vers un processus plus complet et plus vivant : *Living Design*

- Glissement vers les phases "vivantes" du processus
- Décalage de processus objet-agent par chevauchement
- Prototypage rapide plus évolué

Vers une méthode plus adaptée : *MétADELFE*

- Idée : les travaux et activités d'un processus doivent s'agencer afin de répondre aux besoins des concepteurs
- Utilisation d'ADELFE sur le domaine des méthodes (SPEM)

Conclusion

Bilan

Objectif : définir un cadre de développement de systèmes émergents pour les non spécialistes

Les choix

- Les AMAS
- L'ingénierie orientée objet et le RUP

Apports

- Méthodologiques : SPEM, A-UML, AdelfeToolkit
- AMAS : modèle d'agent coopératif, coopération

Exemples illustratifs

- ETTO : agents communicants orientés utilisateurs
- Robotique collective : agents quasi-réactifs non communicants

?

Bibliographie



Ali, S., Zimmer, R., and Elstob, C. (1997).
The question concerning emergence : Implication for Artificiality.
In Dubois, D.M., editor, *Computing Anticipatory Systems : CASYS'97 - First International Conference*.



Goldstein, J. (1999).
Emergence as a Construct : History and Issues.
Journal of Complexity Issues in Organizations and Management,
1(1).



Iglesias, C. A., Garijo, M., González, J. C., and Velasco, J. R. (1998).
Analysis and design of multiagent systems using MAS-CommonKADS.
In Singh, M. P., Rao, A., and Wooldridge, M. J., editors, *Intelligent Agents IV : Agent Theories, Architectures and Languages*, volume 1365 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer-Verlag.



Langton, C. (1990).