

Agents et Systèmes multi-agents

Introduction

Claude Moulin

Université de Technologie de Compiègne

IA04

Sommaire

- 1 Introduction
Définitions
- 2 Intérêt
- 3 Principaux concepts
Agents et Objets
Types d'agents
SMA

Sommaire

- 1 Introduction
- 2 Intérêt
- 3 Principaux concepts

Sommaire

- 1 Introduction
Définitions
- 2 Intérêt
- 3 Principaux concepts
Agents et Objets
Types d'agents
SMA

Paradigme

- Message-Oriented Middleware (MOM) : communication asynchrone et persistante.
- Chaque entité peut envoyer et recevoir des messages.
- Les messages sont envoyés une fois au destinataire (1 à 1 ou 1 à plusieurs).

Agent : définition

- Un agent est un système informatique :
 - **situé** dans un environnement,
 - qui agit d'une façon **autonome** et **flexible** pour atteindre les objectifs pour lesquels il a été conçu.

Situé

- Un agent peut recevoir des entrées sensorielles provenant de son environnement et peut effectuer des actions qui sont susceptibles de changer cet environnement.
- Le monde réel et Internet sont des exemples d'environnements où les agents peuvent être situés.

Autonome et flexible

- Système Autonome :
 - un agent est capable d'agir sans l'intervention directe d'un humain ou d'un autre agent et a le contrôle de ses actions et de son état interne.
- Système Flexible :
 - un agent peut répondre rapidement aux changements qui se produisent dans son environnement ;
 - un agent peut prendre des initiatives au moment approprié ; il n'agit pas seulement en réponse à son environnement ;
 - un agent peut interagir avec d'autres agents ou avec des humains pour collaborer à certaines tâches.

Vocabulaire - 1 : Compétence

- Un agent est capable de faire un certain nombre de choses, de rendre un certain nombre de services : ses **compétences**.
- Une compétence se traduit par les tâches, les actions, les raisonnements que l'agent peut entreprendre pour mettre en oeuvre cette compétence.
- Il peut être intéressant d'indiquer à d'autres entités les compétences d'un agent.

Vocabulaire - 2 : Action

- On considère un agent implémenté dans un langage à objets.
- Action : un agent peut exécuter des actions
 - méthode avec éventuellement des prises en compte de problèmes d'exécution (try ... catch).
 - exemples : parser un fichier, avancer dans un espace, changer la fréquence d'acquisition d'un capteur, recevoir, envoyer un message, etc.

Vocabulaire - 3 : Tâche

- Un agent doit ou peut réaliser certaines **tâches**. Il a été programmé pour cela.
 - une tâche représente ce que l'agent est capable de faire.
 - une tâche peut avoir des préconditions et des effets.
 - une tâche implique une sélection d'actions.
 - une tâche peut être implémentée par une classe.
 - exemples : détecter les spam, jouer le prochain coup, arrêter le ballon, etc.
- Une compétence implique en général plusieurs tâches mais peut se réduire à n'être capable que d'une seule.
 - ex : savoir jouer aux échecs

Vocabulaire - 4 : But

- Un agent peut devoir atteindre un **but**.
 - maximiser une certaine fonction d'utilité.
 - atteindre une situation désirable.
- L'agent raisonne ou délibère pour réaliser un but.
- Plusieurs stratégies peuvent s'appliquer.
 - Utiliser un système expert.
 - Etablir un plan qui reflète la suite de tâches (ou d'actions) à effectuer en vue de réaliser son but.

Vocabulaire - 5 : Monde

- Le **monde** est l'environnement où l'agent est situé (réel, numérique).
- Un agent peut se bâtir une représentation interne du monde, en général une vue incomplète.
 - Structure de données persistente (base de données, système d'information).
 - Un agent doit aussi avoir des informations sur la manière dont ses propres actions affectent le monde autour de lui.

Système

- Définition Wikipédia :
 - Un système est un ensemble d'éléments interagissant entre eux selon certains principes ou règles.
- Un système est déterminé par :
 - la nature de ses éléments constitutifs ;
 - les interactions entre ces derniers ;
 - sa frontière, c'est-à-dire le critère d'appartenance au système (déterminant si une entité appartient au système ou fait au contraire partie de son environnement).

Système Multi-agents (SMA)

- Dans beaucoup de situations plusieurs agents co-existent et interagissent avec d'autres agents.
Exemples : agents logiciels sur Internet, robots footballeurs.
- un système multi-agents (SMA) est un système comprenant un certain nombre d'agents qui peuvent interagir entre eux.
- Le champ de l'IA qui traite des principes et de la conception des SMA s'appelle l'IA distribuée.

Sommaire

- 1 Introduction
- 2 Intérêt**
- 3 Principaux concepts

Domaines d'application - 1

- Énergie : achat de puissance d'une manière intelligente, gestion des réseaux, support d'un centre de crise.
- Industrie : automatisation des processus et de la production, logistique, robots coopératifs, maison intelligente, smart cartes.
- Communication (y compris télécommunication) : gestion de réseaux, commerce électronique, maison intelligente, services du réseau personnel, calcul mobile.
- Information : assistance personnelle, recherche d'Information, gestion de workflow, maison intelligente.
- Santé : supervision des malades, systèmes d'aide.

Domaines d'application - 2

- Transport : logistique, aide à la mobilité, information des voyageurs, contrôle de trafic.
- Militaire : fouille de données, espionnage.
- Internet : Vente aux enchères, commerce électronique, fouille de données distribuée, veille technologique.
- Sciences sociales : agents personnels qui agissent pour le compte d'utilisateurs.

Utilisation d'agents : Robotique

- Les agents (logiciels ou robots)
 - sont situés dans des endroits différents, parfois inaccessibles à des humains,
 - reçoivent des informations sur des capteurs qui sont géographiquement distribués,
 - doivent coordonner leurs actions pour assurer un comportement optimal (le meilleur possible) du système.
- Exemples :
 - véhicules conçus pour découvrir des environnements hostiles : lune, mars ;
 - automobile : pilotage automatique, actions à la place du conducteur, conseils au conducteur (alertes de somnolence, de dangers) ;
 - compétitions de robots : équipes de robots footballeurs, combats de robots

Utilisation d'agents : Réalité virtuelle

- Dans le domaine de la réalité virtuelle, des jeux, et même des jeux sérieux, le challenge est de construire des agents :
 - qui présentent un comportement intelligent ;
 - qui sont les avatars d'autres participants ;
 - qui simulent des personnages vivants.
 - utilisation des sciences cognitives pour déterminer des traits de caractères et des comportements (joie, tristesse, panique, etc.).

Utilisation d'agents : Simulation

- Gestion d'un éco-système
- Modèle écologique
- Comportement de population
- Phénomène physique : houle
- Etude comparative de plusieurs stratégies

Sommaire

- 1 Introduction
- 2 Intérêt
- 3 Principaux concepts**

Sommaire

- 1 Introduction
Définitions
- 2 Intérêt
- 3 Principaux concepts
Agents et Objets
Types d'agents
SMA

Calcul d'une Factorielle

- Algorithme : calcul de fac (n) ou $n!$

```
si  n = 0
    alors  0! = 1
sinon si n = 1
    alors  1! = 1
sinon
     $n! = n * (n-1)!$ 
```


Approche objet - 1

- Une classe (MathFactorielle) implémente :

```
public long fact(int n) {  
    long r = Integer.MIN_VALUE;  
    if (n == 0 || n == 1)  
        return 1;  
    if (n > 1)  
        return n * fact(n - 1);  
    return r;  
}
```

Approche objet - 2

- Un objet instancie un objet de type `MathFactorielle` et appelle la méthode d'instance de cet objet `fact` :

```
MathFactorielle mf = new MathFactorielle();  
this.n = 20;  
this.result = mf.fact(n);  
System.out.printf("Factorielle %d = %d", n, result);
```

- Résultat : Factorielle 20 = 2432902008176640000

Approche objet - 3

- Un objet encapsule un état et un comportement
- L'état correspond à l'ensemble des valeurs de ses propriétés.
- Un objet envoie un message à un autre objet en lui enjoignant de réaliser une action et attend le résultat (aspect synchrone).
- Un objet contrôle son état mais ne contrôle pas son comportement.
- Un type d'objets présente les méthodes auxquelles les objets de ce type répondent (interface).

Approche objet - 4

- Objets dépendants.
- Objets intéressés par l'état d'un autre.
 - L'objet source envoie une notification à un objet cible.
 - Cette notification déclenche une méthode de l'objet cible qui réagit si la notification l'intéresse.
 - L'objet cible peut être une fonction.
- L'objet source n'appelle pas directement une méthode d'un autre objet.

Approche agent - 1

- Un agent (`AgentMathFactorielle`) implémente aussi le calcul de factorielle.
- L'agent a le contrôle sur son comportement :
 - Il peut décider par exemple de ne pas répondre à un message à partir d'un entier négatif ;
 - Il peut décider de répondre par un texte explicatif, etc.
- Un type d'agent ne présente pas les méthodes auxquelles il répond mais plutôt ce qu'il sait faire (compétence).
- D'où le problème de communication entre agents.

Approche agent - 2

- Un agent encapsule aussi un état et un comportement
- L'agent a le contrôle sur son comportement
- L'agent exerce ce contrôle de différentes manières (réactif, dirigé par les buts, social)
- Un SMA a plusieurs flux de contrôle alors qu'un système à objets n'a a priori qu'un seul flux de contrôle.

Sommaire

- 1 Introduction
Définitions
- 2 Intérêt
- 3 Principaux concepts
Agents et Objets
Types d'agents
SMA

Trois types

- Agents réactifs
- Agents délibératifs
- Agents hybrides

Agent réactif

- ne fait que réagir aux changements qui surviennent dans l'environnement ;
- se contente d'acquérir des informations et d'y réagir en appliquant certaines règles prédéfinies ;
- destinés à réagir très rapidement.

Agent délibératif

- effectuent une certaine délibération pour choisir leurs actions.
 - elle peut se faire en se basant sur les buts de l'agent ;
 - sur une certaine fonction d'utilité ;
 - elle peut prendre la forme d'un plan : suite d'actions que l'agent doit effectuer.

Agent hybride

- Pour la majorité des problèmes cependant, ni une architecture complètement réactive, ni une architecture complètement délibérative n'est appropriée.
- Les agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe),
- et dans d'autres, ils doivent avoir un comportement plus réfléchi.

Architecture hybride

- Généralement plusieurs couches logicielles :
 - une couche purement réactive : prend ses décisions en se basant sur des données brutes en provenance des capteurs ;
 - une couche intermédiaire : travaille plutôt avec une vision qui se situe au niveau des connaissances de l'environnement ;
 - une couche supérieure : se charge des aspects sociaux de l'environnement, contact avec d'autres agents.

Sommaire

- 1 Introduction
Définitions
- 2 Intérêt
- 3 Principaux concepts**
Agents et Objets
Types d'agents
SMA

Plusieurs situations

- Un agents isolé.
- Plusieurs agents cohabitants.
- Système multi-agents.
- Système multi-agents confronté à des agents ou un autre système multi-agent.

Caractéristiques

- Conception
- Environnement
- Perception
- Contrôle
- Connaissance
- Communication

Approche SMA

- La création d'un SMA nécessite une plateforme (développement - exécution) SMA.
- Exemples :
 - JADE (Java Agent DEvelopment) : <http://jade.tilab.com/> ; open-source et basé sur le langage Java.
 - OMAS (Open Multi-Agent Asynchronous Systems) : plate-forme de recherche développée par l'équipe d'IA de l'UTC (Jean-Paul Barthès).
 - MASON (Multi-Agent Simulator Of Neighborhoods... or Networks) : <http://cs.gmu.edu/~eclab/projects/mason/>

Plateforme multi-agents

- Plateforme permettant de développer des systèmes multi-agents.
- Une infrastructure de logiciels utilisée comme environnement pour le déploiement et l'exécution d'un ensemble d'agents.
- Un SMA est un ensemble d'agents développés à partir de ces outils.
- Une plateforme fournit une manière confortable de créer et tester des agents :
 - elle prend en charge la gestion des agents (cycle de vie) et l'échange des messages ;
 - elle est basé sur un modèle particulier d'agents.

Avantages de la technologie des SMA

- Rapidité et efficacité à cause de processus asynchrones et parallèles.
- Robustesse et fiabilité : le système peut continuer même si un agent tombe.
- Extensibilité : il est possible (facile ?) d'ajouter de nouveaux agents à un système.
- Développement, coût et réutilisation : il est plus simple, donc moins onéreux, de développer et maintenir un système modulaire qu'un système monolithique.