Introduction aux Systèmes Multi-Agents
Le Dîner des Philosophes

FRANÇOIS HERNANDEZ - LÉO PONS
CentraleSupélec
January 28, 2017

Contents

Introduction

Un Système Multi-Agents est un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Le fonctionnement du système est défini par le fonctionnement des agents, le système central ne fait qu'office d'environnement d'évolution et de communication des différents agents. On parle d'intelligence artificielle distribuée.

Ce TD a pour objectif de construire une plateforme générique permettant de représenter et simuler des systèmes multi-agents de différentes natures. Un premier SMA simulé sera celui du dîner des philosophes.

Le problème des philosophes peut être défini de différentes manières, mais l'idée générale est toujours la même.

La situation est la suivante :

- Un certain nombre de philosophes se trouvent autour d'une table.
- Chacun des philosophes a devant lui un plat de spaghetti.
- À gauche de chaque plat de spaghetti se trouve une fourchette.

Un philosophe a un certain nombre d'états possibles, par exemple :

- Il peut penser, ce qui lui donne faim.
- Il peut être dans un état de famine, s'il a faim et qu'il ne peut pas manger.
- Il peut être en train de manger.

Des contraintes extérieures s'imposent à cette situation :

- Pour manger, un philosophe a besoin de deux fourchettes : celle qui se trouve à gauche de sa propre assiette, et celle qui se trouve à droite (c'est-à-dire les deux fourchettes qui entourent sa propre assiette).
- Quand un philosophe a faim, il attend que les fourchettes soient libres.

Le problème consiste à définir le comportement des philosophes tel qu'ils puissent tous manger, chacun à leur tour.

I Plateforme SMA Synchrone

Dans un premier temps, nous créons un package 'plateforme' contenant les éléments principaux de la représentation générique d'un SMA.

Notre plateforme fonctionne de manière synchrone : Le premier agent observe son environnement et effectue une action dont les critères d'exécution sont satisfaits par les données observées, puis c'est au tour du second et ainsi de suite. Arrivé au dernier agent, on recommence en repartant du premier.

Notre package contient les classes suivantes :

- `Agent` : Représente chaque agent du système, liés à un `Environnement`, à une liste d'actions possibles et à une boîte aux lettres (pour la réception des messages, détaillée plus loin).
- `Environnement` : Représente l'environnement dans lequel évoluent les agents, contenant des `Donnee` et une liste d'agents `Agent`.
- `Donnee` : Classe générique représentant les différentes données d'un problème.
- `Action` : Classe abstraite destinée à représenter les différentes actions possibles, chaque action sera une sous-classe de celle-ci. Deux méthodes cruciales définissent une action : `check_conditions` qui évalue les critères d'exécution de l'action, et `effectuer` qui exécute l'action proprement dit et modifie donc les données affectées par cette action.
- `Message` : Les messages seront décrits dans un deuxième temps. Ils permettent la communication entre les agents.

Vous trouverez dans le code Java des commentaires détaillant l'implémentation des différentes classes et de leurs méthodes.

II Application au problème du dîner des philosophes

Afin d'implémenter le problème des philosophes, nous créons un nouveau package 'philosophes' contenant différentes classes héritant des classes génériques de la plateforme, ainsi que de nouvelles destinées à représenter d'autres aspects spécifiques au problème.

Les classes héritant de la plateforme sont les suivantes :

- `Philosophe`, héritant de la classe `Agent`.
- `Table`, héritant de la classe `Environnement`.
- `philosophes.actions`, package de classes héritant de la classe `Action`. Chacune de ces classes définit une action exécutable par les philosophes.
- `Fourchettes`, héritant de la classe `Donnee` et contenant la représentation de l'état des N fourchettes de l'environnement.

Les classes supplémentaires sont les suivantes :

- `Etat`, type énuméré représentant l'état d'un `Philosophe`.
- `main`, permettant de faire tourner le système.

Tests et conclusion