

## Compte-rendu IA04 - TD4

### Simulation et Environnement

### Le Sudoku

---

#### 1. Quels sont les rôles de chaque type d'agent ?

Il y a trois types d'agents au total :

1. Le type **Agent d'environnement** (`EnvironmentAgent` dans le code) : stocke la matrice qui représente la grille de Sudoku chargée initialement. Il conserve également une version qui correspond à l'état actuel de la matrice en cours de résolution. C'est cet agent qui se charge de recevoir les messages en provenance de la console JADE.
2. Le type **Agent de simulation** (`SimulationAgent` dans le code) : fait la jonction entre chaque agent d'analyse et l'agent d'environnement. Il se charge de mettre à jour la grille partiellement résolue en fonction des données envoyées par les agents d'analyse. Il envoie périodiquement à une salve de messages aux agents d'analyses contenant la portion à analyser. Il envoie également de temps en temps un message de mise à jour à l'agent d'environnement pour qu'il *update* sa version actuelle de la grille (partiellement résolue).
3. Le type **Agent d'analyse** (`AnalysisAgent` dans le code) : 27 instances traitent chacune leur partie de ligne et déduisent les possibilités pour chaque case traitée.

#### 2. Quelles sont les tâches de chaque type d'agents (en termes de behaviours simples et composites) ?

`EnvironmentAgent` possède 2 behaviours :

1. `ReceiveFromConsoleEnvironmentBehaviour` : comme son nom l'indique, ce behaviour gère les messages en provenance de la console JADE. Principalement, il peut afficher la grille partiellement résolue actuelle et lire une nouvelle grille (ce qui redémarrera totalement le programme et enchainera une nouvelle série de résolutions). Ce behaviour hérite de `CyclicBehaviour`.
2. `ReceiveFromSimulationEnvironmentBehaviour` : là encore, le nom induit que ce behaviour traite les messages reçus de l'agent de simulation. En effet, ce dernier envoie périodiquement la version la plus à jour de la grille partiellement résolue à l'agent d'environnement. Ce behaviour hérite de `CyclicBehaviour`.

Étant à la jonction des deux autres types d'agents, **SimulationAgent** est très sollicité et possède 4 behaviours :

1. **ReceiveFromAnalysisSimulationBehaviour** : ce behaviour traite les messages reçus des agents d'analyse. Quand un agent a terminé son analyse, il renvoie un message à l'agent de simulation. Quand ce dernier reçoit cette réponse, il décrémente le nombre de messages qu'il attend (sur les 27 messages initiaux envoyés). Ce behaviour hérite de **CyclicBehaviour**.
2. **SimulationToEnvironmentBehaviour** : ce behaviour envoie périodiquement à l'agent d'environnement la version actuelle de la grille partiellement résolue. Ce behaviour hérite de **TickerBehaviour**.
3. **SimulationToAnalysisBehaviour** : l'agent de simulation utilise ce behaviour pour envoyer périodiquement une salve de messages aux 27 agents d'analyses pour qu'ils traitent la portion de grille qui les concerne. Ce behaviour hérite de **TickerBehaviour**.
4. **ReceiveFromEnvironmentSimulationBehaviour** : comme son nom l'indique, ce behaviour traite les messages reçus de l'agent d'environnement. En fait, on pourrait penser qu'il ne s'active qu'une seule fois, mais il est possible en fait de modifier la grille originale bien que le programme soit déjà en cours d'une résolution d'une grille, donc ce behaviour n'est pas un **OneShotBehaviour** mais bien un **CyclicBehaviour**. Son rôle est de mettre à jour la grille à résoudre dans l'agent de simulation.

**3. Quel sont les types des messages échangés (request, inform, subscribe, etc.), leur utilité et leur contenu ?**

**4. En supposant que les agents d'analyse sont situés sur différentes stations, la résolution est-elle encore possible ?**