Compte rendu de réunion n°6

**Objet** : Présentation de l'exemple illustratif, idées pour la suite

Date : 03/04/2015 à 10h30.

Participants : Véronique DESLANDRES (**VDE**), Salima HASSAS (**SHA**), Kevin ESPENEL (**KES**).

Rédigé par : Kevin ESPENEL.

**Ordre du jour :**

1 – Présentation de l'exemple illustratif

2 – Idées et pistes pour la suite

3 – Remarques / Questions

**Déroulement :**

Pour garder une trace précise de ce qui a été dit dans l'ordre du jour, j'y ai reporté mes notes concernant tout ce que j'ai souhaité aborder durant la réunion. Les réponses et les échanges sont représentés en **vert**.

1 – Présentation de l'exemple illustratif

=> Présentation de l'exemple perception/influence/calcul satisfaction

**Améliorer les zones de légende dans la présentation des exemples illustratifs afin que n'importe qui puisse comprendre le fonctionnement sans rien avoir lu du projet avant (signification de lambda, satisfaction, attributs "selected"…).**

**Continuer d'enrichir la valeur de lambda sur la base des travaux de Noura. Intégrer des éléments de valeur propre aux propositions et pas seulement à leur appartenance à une certaine tournée de livraison.**

**=> Remarques sur le processus de résolution :**

- Ceux dont la satisfaction est plus grande convergent plus vite

- Lorsque deux agents ont une forte préférence pour des propositions en conflit, il est très probable que le premier agent agissant lâche la proposition et en sélectionne une autre. Or cela peut être mauvais si c'était la meilleure proposition pour lui et une solution moyenne pour l'agent avec lequel il est en conflit.

**Formaliser les notions qui sont implicites dans le processus de résolution. Ex. l'ordre des XOR a une importance car ils définissent l'ordre de livraison des groupes et des clients, il faut formaliser cela par exemple en appelant les XOR des Ordered XOR.**

**Changer le nom de "selected" au niveau des propositions ou l'algorithme de traitement, il y a un problème de sémantique.**

**Etudier les langages de proposition utilisés en IA par exemple pour réaliser des inférences. On a des patterns récurrents (XOR/AND/XOR/AND). Voir si les algorithmes utilisés dans ce domaine peuvent être intéressants pour nous.**

**Problème de la convergence trop rapide qui peut être constatée (plus l'agent a une bonne solution, plus il converge vite) : Introduire un facteur aléatoire pour améliorer l'exploration des solutions, éviter les comportements systématiques. Se concentrer ensuite sur le paramétrage dans ce type de problème.**

=> Ainsi pour la suite il pourrait être intéressant de modéliser des scénarios de négociation afin d'étudier le maximum de cas possibles. Cela permettrait d'identifier des comportements qui pourraient être contre productifs pour les améliorer.

2 – Idées et pistes pour la suite

**=> Etude des design patterns pour identifier les bonnes pratiques adaptées à notre problème**

- Exemple design pattern Observer :

Les agents observent l'environnement et sont notifiés lorsqu'ils y a des modifications qui touche leur partie locale de l'environnement.

- Pattern Etat :

En fonction de leur état (satisfait, non satisfait, proche d'être satisfait etc…) les agents adoptent une stratégie différente (autre manière de générer des propositions etc…).

- Pattern Mediator :

Au lieu d'interagir entre eux directement, les agents passent par un médiateur qui dispatche ensuite les messages (ici on peut voir l'environnement comme le Médiateur ?).

**Ok bonne approche, continuer à rechercher les design patterns intéressants pour avoir une bonne solution générique.**

**=> Etudier la programmation orientée agents :**

- Frameworks existants en JAVA

- Autres plateformes.

**Ne pas perdre trop de temps à étudier ces plateformes, se concentrer sur la solution JAVA. Les frameworks JAVA sont trop rigides et contraignant quand le système devient complexe.**

**Ne pas utiliser de framework pour le projet mais étudier leur fonctionnement, voir s'il y a des patterns réutilisables. Se concentrer sur MadKit plutôt que Jade.**

**=> Les design patterns multi-agents :**

- Faire l'état de l'art de design patterns qu'il pourrait être intéressant de réutiliser pour le projet

**Voir le cours de Smart Environment donné en Master IADE, il y a des patterns intéressants qui ont été présentés, résolution de problème complexes récurrents en utilisant une approche bio-inspirée.**

**=> Organisation dans les SMA, négociation**

**Org dans les SMA non, les travaux sont très vastes et pas très adaptés à notre problème.**

**La négociation dans les SMA oui, comme vu lors de notre précédente réunion.**

**=> Méthodes de guidage interne de l'émergence**

**Travaux très généraux dans le domaine, difficile dans retirer quelques chose pour notre problème spécifique. Mais peut être intéressant de s'inspirer des démarches employées.**

**Il serait intéressant de faire une mindmap pour visualiser le positionnement du projet au sein des différents domaines de la recherche.**

=> **Réseau overlay qui servirait de trace des négociations passées**. Les négociations laissent une trace locale dans l'environnement pour apporter une information supplémentaire aux agents qui voudraient générer des propositions touchant à cette région.

- Représentation de la trace par message structuré ? Simple modification de valeur type lambda ?

**Se concentrer sur l'approche par réseau overlay, réfléchir sur sa structure (ensemble de nœuds agents ?, messages propagés dans l'environnement?...)**

**Favoriser une approche d'intelligence décentralisée, ne pas avoir un contrôleur central dirigeant le fonctionnement du système.**

**Penser l'intégration de la couche overlay dès le départ pour qu'elle soit correctement intégrée.**

**=> Revoir plus en profondeur l'architecture du projet et proposé un système "Threadé"**

- Intéressant pour moi au niveau ingénierie

- Intéressant au niveau des performances (exploitation des performances des processeurs multi-cœurs, les agents localisé dans des parties différentes de l'environnement peuvent agir simultanément)

- Système où les agents peuvent agir simultanément, intéressant pour une approche SMA.

**Ok approche intéressante niveau ingénierie, évaluer les risques pour ne pas perdre trop de temps avec ce type d'approche et se concentrer sur la partie résolution. S'inspirer de madkit qui est multi-threadé.**

**Aborder l'environnement comme un graphe, avec des régions regroupant plusieurs nœuds, s'inspirer des approches dans les réseaux d'influence.**

**=> Approfondir encore un peu les méthodes de gestion de projet et notamment Agiles**

- Modélisation

- Développement itératif

- Tests

- etc

3 – Remarques / Questions

**=> Sur quels domaines je dois concentrer ma biblio** ? (partie état de l'art du rapport de stage)

- Stigmergie

- Emergence dans les systèmes multi-agents

- Elargir également à la prog multi-agents et design patterns

**Mettre à jour l'état de l'art, voir les domaines des réseaux d'influence, VRP, négociation, stigmergie, travaux sur les designs patterns en SMA, eco laws**

=> Quand est-ce que l'on fixe la prochaine réunion

**Lundi 13/04 à 14h**

**Actions :**

KES

- Apporter les améliorations à l'exemple illustratif pour le rendre plus clair

- Corriger les problèmes de sémantique

- Tirer du concret pour avancer dans le projet et pouvoir orienter le travail pour la suite

VDE / SHA

- Mise à disposition d'articles intéressants pour la biblio sur le wiki