Compte rendu de réunion n°7

**Objet** : Présentation des corrections à apporter à l'existant, proposition de structure générique pour les propositions, proposition de réseau overlay.

Date : 14/04/2015 à 14h.

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

Rédigé par : Kevin ESPENEL.

**Ordre du jour :**

1 – Propositions de correction des problèmes de sémantique

2 – Proposition de structure générique pour la représentation des propositions

3 – Réflexions autour de la couche overlay

4 – 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 – Propositions de correction des problèmes de sémantique**

**Problème sémantique selected :**

Pour rappel selected permet d'identifier s'il y a des **conflits au niveau d'une proposition.** Tous les agents en lien avec la proposition ont une valeur selected correspondante (true, false).

Pour qu'il n'y ait pas de conflit, il faut que toutes les valeurs de selected soient vraies. Cela ne signifie pas que les agents ont tous selectionné la proposition, seulement qu'il y a un accord.

- Solution 1 : Changer l'algorithme de calcul de la satisfaction et selected en "reserved"

On inverse plus les valeurs reserved en passant pas les nœuds NONode.

On modifie l'algorithme de calcul de la satisfaction : Pour que la proposition P1 propage une satisfaction positive pour l'agent A1, il faut que A1 ait reserved = true pour la proposition et tous les autres reserved = false.

=> Avec cette solution la représentation devient cohérente pour qu'il y ait satisfaction, il faut que seul l'agent lui-même ait reservé les propositions, sinon il y a conflit.

- Solution 2 : Changer selected(Agent, true/false) en influence(Agent, true/false)

=> On retient la solution 1 et on remplace "reserved" par "booked" à cause du double sens. On modifie l'algorithme de calcul de la satisfaction au niveau des propositions.



**Problème des nœuds XOR ordonnés :**

Pour rappel les nœuds XOR sont générés dans un ordre précis dans la solution actuelle. Cependant il n'est pas indiqué que les solutions doivent être ordonnées dans les spécifications.

Solution :

Créer un nouveau type de nœuds XOR plus spécifique OrderedXOR où le fait que les nœuds fils sont ordonnés est spécifié.

Dans certain problèmes l'ordre n'aura pas d'importance. La distinction entre les deux pourra avoir une **incidence au niveau du code pour les structures de données des listes par exemple**. Il n'y a pas d'incidence au niveau du processus de résolution.

Dans la représentation des exemples, il suffira d'indiquer lors de la présentation du problème si l'ordre à une importance où non. Pas besoin de l'indiquer dans tous les schémas, cela serait trop lourd.

=> Ok pour cette solution, spécifier au niveau du nœud lorsqu'il est ordonné ou non, cela peut être représenté par une méta-information.

**2 – Proposition de structure générique pour la représentation des propositions**

- Présentation de la structure permettant une représentation d'un large domaine de propositions (diagramme de classe).

=> Ok pour cette représentation. Améliorer les détails et penser à ce que l'on met dans les builder pour faciliter au maximum la création des éléments. Penser à l'intégration de cette structure dans le projet global (au sein du réseau de chaque agents etc..).

=> Penser à une structure très dynamique pour les propositions, au-delà des variables ajouter la possibilité de mettre des méta-informations (informations sur la réservation de la proposition par les agents etc..).

**3 – Réflexions autour de la couche overlay**

**Objectif :**

On veut partir sur l'idée d'un réseau overlay capable de donner une trace des activités passées pour orienter le comportement des agents.

**Principe :**

- On se base sur un **environnement très dynamique dont la structure peut évoluer rapidement**. On a au départ une base de propositions dans l'environnement. Ce nombre de propositions peut augmenter ou diminuer de même que la répartition de ces propositions dans l'environnement (par ex. une région où le nombre de propositions se multiplie car il y a des conflits importants).

- Les propositions peuvent être de différentes natures et elles ne sont **pas classées naturellement** dans différentes régions.

- **Les agents** **gèrent eux-mêmes la génération de propositions** en fonction de leur perception des conflits dans l'environnement. Ils cherchent conjointement à atteindre le meilleur compromis assurant la meilleure situation globale.

**Proposition :**

La couche overlay est composée d'un ensemble d'agents interconnectés.

La couche overlay doit être capable de s'adapter au dynamisme de l'environnement et d'analyser sa structure pour guider les agents dans leur processus de résolution.

L'overlay représente la mémoire collective du système et les **traces de la résolution passée de conflits doivent permettre d'orienter les agents dans leurs futures tentatives**.

Il peut y avoir **plusieurs types de nœud dans la couche overlay :**

Chaque proposition de l'environnement possède un **nœud observateur**. Ce nœud observateur est chargé d'analyser la situation de la proposition, son intérêt an sein du graphe

Lors de l'apparition de conflits entre propositions un **nœud conflit** est créé pour représenter la zone de conflit. Le nœud **évalue l'importance du conflit** pour représenter le risque pour un agent de choisir cette proposition.

Lors de la résolution d'un conflit un **nœud résolution** est créé. Il sert de trace **de la résolution** (stratégie utilisée par l'agent pour sortir du conflit, nature de la modification effectuée) et influence les agents dans le choix de leur stratégie pour sortir du conflit. Les nœuds de type résolution ont une **force** représentée par l'augmentation de l'intérêt induit chez les agents par la modification et part le nombre de conflits solutionnés.

**Avantages de la couche overlay :**

=> Méthode totalement **décentralisée** où les agents sont des acteurs du partage de l'information.

=> Le réseau overlay est **dynamique** et sa structure s'adapte à celle de l'environnement.

=> Les nœuds de type conflit et résolution ont un **facteur d'évaporation** et disparaissent lorsque les phénomènes qu'ils représentent ne sont plus d'actualité. Ainsi la taille du réseau overlay est limitée.

=> Les informations indiquées dans la couche overlay disposent d'un poids déterminant la probabilité pour un agent de suivre l'indication donnée par l'overlay. Cela préserve une approche **non déterministe** et le compromis exploration/renforcement.

**Question en suspens :**

=> Peut-on ajouter d'autres types de nœuds dans le réseau overlay ? (ex. un nœud par agent chargé d'observer la stratégie de génération adoptée par l'agent). Attention à limiter la taille du réseau.

=> Quelles informations placer au sein des nœuds ? Pour garder une trace de la résolution d'un conflit, l'overlay pourrait contenir la stratégie de génération de proposition utilisée par l'agent (stratégies à définir, il faut quelles représentent également les changements dans l'arbre de proposition de l'agent).

=> La structure du réseau overlay que j'ai présenté contient un trop grand nombre de nœuds. Il faudrait déporter l'information contenue dans les nœuds observateurs vers les propositions elles-mêmes (méta-information). Cela évite d'avoir un réseau overlay de trop grande taille (un nœud observateur par proposition).

=> Les nœuds de type résolution n'apporte pas d'avantages par rapport aux nœuds de type conflit. On peut fusionner ces deux types de nœuds et déporter les informations de résolution vers les nœuds conflits.

=> Ajouter la possibilité de communication entre les nœuds conflits pour l'échange d'information sur la résolution des conflits (politique adoptée par les agents pour la génération de nouvelles propositions).

=> La communication entre les nœuds de l'overlay peut également permettre d'identifier des dépendances entre certaines régions (par exemple problème de coloration de graphe).

**v2** => L'overlay peut également fournir un moyen de visualisation pour l'administrateur et permettre d'identifier les causes des problèmes rencontrés. Par exemple pour les politiques de transport, on remarque des régions de conflits qui peuvent inciter à augmenter le nombre d'aires de livraison par exemple.

Objectifs pour la suite :

=> Continuer la réflexion autour du réseau overlay pour **avoir une implémentation le plus rapidement possible et faire des tests** sur des instances de problème.

=> Pour le parallélisme ne pas se concentrer sur cette aspect au niveau du processus de résolution. Implémenter une solution avec un parallélisme faible où l'on ne repense pas le processus de résolution pour permettre la simultanéité. Les agents qui ne sont pas en conflit peuvent agir simultanément car ils n'essayent pas d'accéder aux mêmes ressources.

**4 - Remarques / Questions**

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

=>Lundi 27 avril à 9h.

**=> Je devrais partir plus tôt (15h30) jeudi 16/04 et mardi 21/04, je rattraperai les heures sur les autres jours.**

**Actions :**

KES

- Poursuivre la représentation générique des propositions et penser leur intégration dans le projet global

- Avancer sur le réseau overlay, penser à comment implémenter la structure et faire des tests sur des exemples pour étudier le comportement du système.

VDE / SHA

- Eventuelles remarques ou idées pour améliorer les solutions proposées.

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