Compte rendu de réunion n°9

**Objet** : Bilan du développement de la structure générique des propositions et avancée de l'intégration à la solution initiale

Date : 11/05/2015 à 8h50.

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

Rédigé par : Kevin ESPENEL.

**Ordre du jour :**

1 – Bilan du développement de la structure générique des propositions

2 – Avancées de l'intégration à la solution initiale

3 – Organisation pour la suite de l'intégration

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 – Bilan du développement de la structure générique des propositions**

- Développement de la structure générique terminé, structure adapté pour modéliser le problème des tournées de livraison

**Il faut finir les tests unitaires pour s'assurer de la stabilité de la solution + javadoc.**

**Voir si la solution est adaptée pour modéliser des benchmarks et l'adapter s'il y a des manques.**

**2 – Avancées de l'intégration à la solution initiale**

- Les propositions ont pu être modélisées à partir des données de l'ancienne solution, les contraintes sur les propositions également.

- Il faut finir l'intégration générique à l'ancienne version (beaucoup de classes touchées).

**Il faut revoir la structure actuelle des nœuds pour la rendre plus générique. Elle est actuellement trop couplée à la modélisation des propositions pour les tournées de livraison. Le lambda qui joue un rôle central lors des phases de propagations devient une propriété de la proposition.**

**Il faut également intégrer le rôle des méta-informations.**

**Revoir la structure des nœuds permettra de facilité ensuite l'intégration de nouveaux types de nœuds et la communication avec l'overlay.**

**3 – Organisation pour la suite de l'intégration**

- Il faut finir de modifier les classes existantes pour les agents et l'environnement et les adapter à la nouvelle structure. Par exemple dans l'ancienne version le code définissant les contraintes sur les propositions était définit au niveau de la classe RoutePhaseProposition. Dans la nouvelle version il est définit dynamiquement au sein des contraintes (globales ou binaires) passées à l'environnement.

**=> Continuer à intégrer la nouvelle structure de propositions et faire les tests unitaires. => 3 jours pour adapter les classes existantes et pouvoir comparer les résultats à ceux obtenus dans la version précédente.**

**A faire ensuite :**

**=> Créer la dynamique du problème traité. Les benchmarks fournissent des données d'entrée mais il faut être capable de générer des perturbations à la manière des agents perturbateurs dans l'ancienne version. On peut s'appuyer sur les domaines de la nouvelle structure pour créer des générateurs de proposition et gérer la dynamique.**

**Le développement de générateurs de propositions prendrait plusieurs jours.**

**=> Nettoyer la structure des nœuds pour pouvoir en ajouter de nouveaux plus facilement. Anticiper la possibilité d'échange avec l'overlay et la gestion des métainformations.**

**=> Etudier la manière de récupérer les résultats en sortie (notamment travaux Romain avec les macros excel). S'il reste du temps, rendre paramétrable la génération des données de sortie pour que les résultats soient facilement comparable à l'existant.**

**4 - Remarques / Questions**

**Prochaine réunion pour faire le point sur l'avancement mercredi 13/05 à 10h30.**

**Actions :**

KES

- Finir l'intégration à l'ancienne solution (modification des anciennes classes et adaptation au nouveau comportement) et les tests unitaires (stabilisation de la solution) pendant les 3 prochains jours.

VDE / SHA

- Identifier des benchmarks pour pouvoir mettre à l'épreuve la nouvelle structure et permettre d'identifier d'éventuels points à améliorer dans la structure.