Simulation de l’ordonnancement des missions d’une flotte de robots autonomes

Table des matières

[Introduction 2](#_Toc375075026)

[Modélisation géométrique et topologique 2](#_Toc375075027)

[Topologie du site 2](#_Toc375075028)

[Abstractions 3](#_Toc375075029)

[Trajectoires 3](#_Toc375075030)

[Places 3](#_Toc375075031)

[Modélisation des robots 4](#_Toc375075032)

[Taches et exécution concurrente 4](#_Toc375075033)

[Utilisation des abstractions géométriques 5](#_Toc375075034)

[Suivi de trajectoires 5](#_Toc375075035)

[Suivi dans un contexte concurrent 5](#_Toc375075036)

[Utilisation des primitives de dessin dans un contexte concurrent 6](#_Toc375075037)

[Modélisation des missions 6](#_Toc375075038)

[Agence 6](#_Toc375075039)

[Messages et boite aux lettres des robots 6](#_Toc375075040)

[Répartition des missions et parking 6](#_Toc375075041)

# Introduction

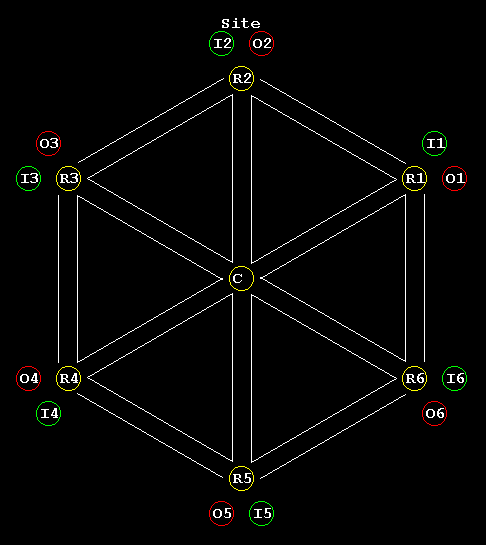
Le projet consiste en l’ordonnancement de missions réalisées par des robots évoluant sur un site commun. Le but est d’obtenir une simulation ou les robots évoluent simultanément sans collisions lors de leurs déplacements.

# Modélisation géométrique et topologique

## Topologie du site

Le site comporte des lieux, appelés places, pouvant être catégorisés en 3 types :

* Les places d’entrés (input places) : Points d’entrée sur le site, les missions des robots commencent sur une place d’entrée.
* Les places de sorties (output places) : Points de sortie du site, les missions des robots finissent sur une place de sortie.
* Les places « transitoires » (ring places, center) : Points de passage temporaires assimilables à des carrefours.



**Figure 1. Topologie du site dans lequel évoluent les robots. Les places I (en vert) sont les places d’entrée, les places O (en route) sont les places de sortie et les places R et C sont des places temporaires (en jaune).**

Des routes relient des places suivant une topologie en hexagone comme décrit dans la figure 1 dans notre cas de simulation.

## Abstractions

### C:\Users\Karl\Downloads\umlet_12.2\Umlet\new.jpgTrajectoires

**Figure 2. Diagramme de classes UML des trajectoires.**

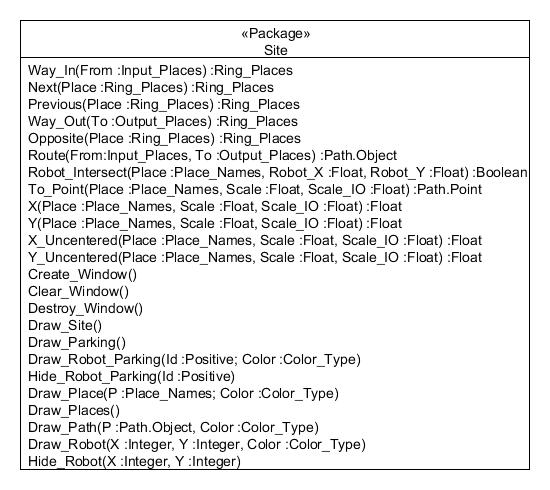
Les trajectoires sont représentées par une structure de données (file FIFO) contenant des places. Ces trajectoires contiennent un chemin constitué d’une liste de points définissant les coordonnées d’une étape.

### Places

Les places et la topologie du site sont décrites dans un package « Site » s’occupant également des opérations de dessin. Ce package contient aussi différents types énumérés pour décrire les places.

Les différentes méthodes permettent d’obtenir d’une place, ses voisins, entrées et sorties du site et différentes opérations utilisées dans les déterminations des trajectoires. La suite de places d’une trajectoire est déterminée par la fonction route, la trajectoire se construit autour de son résultat. Pour déterminer un chemin et les coordonnées des points à atteindre qu’il contient, des fonctions permettant de connaitre les coordonnées des places ont été ajoutées.

Les opérations de dessins sont regroupées dans un objet protégé pour assurer une exclusion mutuelle lors de leurs appels. En effet plusieurs robots évolueront en parallèle et des problèmes de dessins pourraient survenir sans cette protection.



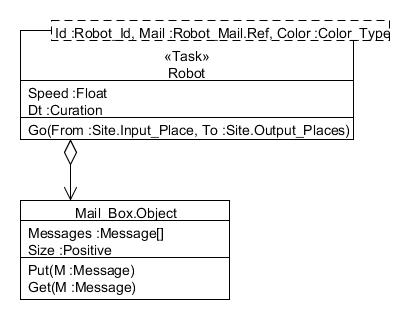
**Figure 3. Contenu du package Site avec les procédures de dessin et de description de la topologie du site.**

# Modélisation des robots

## Taches et exécution concurrente

Les robots sont modélisés par une tache Ada, ils ont une vie propre et indépendante les uns des autres et on communique avec eux grâce à des messages. Les deux messages possibles sont un appel pour lancer une mission (suivi d’une trajectoire) ou un appel pour un arrêt complet (fin de la mission en cours et plus de nouvelles missions ne sont acceptés). Les robots possèdent une vitesse et un taux de rafraichissement utilisé pour s’afficher et faire les calculs de déplacements à intervalle régulier.

Un robot nécessite lors de son instanciation d’être associé à une couleur, une référence vers une boite mail et un identifiant. La boite mail est utilisée lors de la gestion des missions pour emmètre des rapports de fin de mission et indiquer que le robot est disponible pour en réaliser une autre. L’identifiant sert à identifier le robot dans ses rapports et la couleur sera celle utilisée dans les appels aux fonctions de dessin du site pour s’afficher. Ainsi tous les robots partagent la même vitesse et le même taux de rafraichissement, mais leurs identifiants, mail et couleur leur sont propres à chacun.



**Figure 4. La tache modélisant le robot et le lien vers la boite mail qu’il utilise pour notifier ses fins des missions.**

## Utilisation des abstractions géométriques

### Suivi de trajectoires

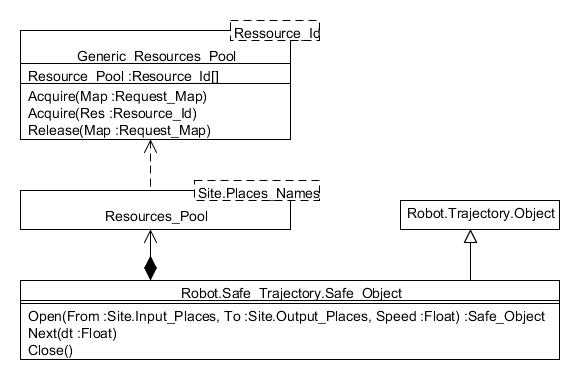
Le robot a pour mission de suivre une trajectoire, cette opération se fait lors d’un appel à « Go ». Cet appelle lance la demande de la liste de places à visiter pour aller d’une place d’entrée donnée à une place de sortie. La trajectoire expose une interface d’itération sur un chemin calculé à partir de la liste de places. Les coordonnées des places ont été déterminées par des appels aux fonctions du package Site. Le robot itère à sa fréquence et ses positions sont rafraichies suivant sa fréquence de rafraichissement, lors d’une avancée du curseur d’itération, un calcul dépendant de la vitesse est du temps est réalisé pour conserver une vitesse constante le long des segments du chemin. Le Robot suit ainsi son chemin et dépose un rapport de fin de mission une fois terminé.

Le robot possède un étant interne indiquant sa disponibilité ou non pour une mission, en début de mission le robot se verrouille et aucune autre mission ne peut lui être demandé, il est de nouveau disponible une fois la mission terminée. Lors d’un appel à « Go », les prise en compte des informations de mission (place de départ et d’arrivé, verrouillage du robot en état de réalisation de mission) sont faites de façon atomique. Le parcours et la réalisation de la mission ne le sont pas. Ainsi plusieurs robots peuvent réaliser des missions en parallèle sur le même site.

### Suivi dans un contexte concurrent

Plusieurs robots évoluent dans le site en simultané, il est donc nécessaire de prévoir un mécanisme pour éviter les collisions, en effet deux trajectoires peuvent se croire, ou pire mettre deux robots en collision frontale. Une réservation des places est donc présente : quand un robot veut réaliser une mission, il réserve l’ensemble des places par les quelles il passera, les autres robots ayant besoin de cette place attendent qu’elle se libère. La libération est faite quand le robot quitte une place (une distance de sécurité est présente, la place est libérée une fois quittée et également une fois certaine distance effectuée).

La gestion des réservations (demandes, libération, attentes) est fait par un package générique contenant un lot de ressources. Ce package est utilisé avec des places qui sont ainsi réservées et libérées par les robots. Des trajectoires sûres sont ajoutées prenant en compte les réservations lors des opérations nécessaires dans les robots.



**Figure 5. Réservation de places et trajectoires sûres**

### Utilisation des primitives de dessin dans un contexte concurrent

De même que les déplacements, plusieurs robots peuvent appeler les fonctions de dessin en simultané. Les primitives « Adagraph » ne sont pas protégées ni prévues pour des appels concurrents. Les fonctions de dessins du site sont donc dans un objet protégé utilisé pour rentre les opérations de dessin sures.

# Modélisation des missions

## Agence

L’orchestration des missions de robots est centralisée dans une agence. Elle a connaissance de la flotte de robots, de la boite mails qu’ils utilisent pour donner leurs rapports de fin de mission et possède un « listener » qui s’occupe de donner des missions aux robots. Elle permet aussi l’envoi d’un message d’arrêt généralisé à tous les robots et à l’écouteur de mission pour un arrêt propre de la simulation.

## Messages et boite aux lettres des robots

Lors de la fin d’une mission, un robot envoie un rapport de fin de mission en y indiquant son identifiant. L’écouteur de mission prend connaissance de ce robot disponible et lui donne une nouvelle mission. Le robot se voie attribuer une nouvelle mission et demande sa trajectoire en faisant les réservations de places. Il réalise sa mission une fois la réservation terminée.

## Répartition des missions et parking

Les robots sans missions ou sans place d’entrée de mission accessible attendent dans un parking. Une fois la place d’entrée sur le site libre ils s’y rendent et commencent le suivi de la trajectoire une fois toutes les autres places réservées. Une fois la mission terminée ils envoient un rapport et retournent au parking.

## D:\Users\Karl\Documents\Documents\umlet_12.2\Umlet\seq.jpgCommunication avec les robots

**Figure 6. Diagramme de séquence de l’attribution et de la réalisation d’une mission par un robot.**