
Modèle formel et architecture décisionnelle

Le projet IWARD vise à développer une approche robotique capable de répondre dans un même temps à différents besoins logistiques en milieu hospitalier tel le nettoyage, le transport, le guidage, la téléconférence. Nous nous focalisons sur la couche « swarm/ team » (essaim / équipe) dont le but est d'optimiser l'utilisation de l'ensemble des robots pour répondre au mieux à différentes demandes en logistique et réagir à différentes perturbations.

Ce chapitre présente un modèle formel du système robotique considéré. Ce modèle permet d'identifier les différents composants du système, leur caractéristiques, leur comportement statique et dynamique. Il permet aussi de caractériser les différentes activités/et missions demandées aux robots et leur exigences.

Le modèle formel est décliné en deux sous-modèles : le modèle statique et le modèle dynamique. Le modèle statique représente les différents constituants du système ainsi que leurs caractéristiques statiques qui n'évoluent pas au cours du temps. Ce modèle statique permet donc de comprendre la constitution du système robotique que nous allons étudier dans cette thèse. Le modèle dynamique décrit les comportements dynamiques des différents composants du système et les données importantes ou variables d'état de ces constituants dont l'évolution influe fortement sur le comportement du système robotique.

3.1. *Modèle Statique*

Dans cette thèse, nous considérons un système composé de N robots de base et M fonctionnalités ou modules permettant d'avoir différentes configurations des robots. L'environnement hospitalier est considéré à travers des points de stationnement pour les robots et des points d'intérêt. Le modèle statique caractérise également les missions pouvant être confiées aux robots. Nous formulons par la suite les caractéristiques de ces composants de notre système robotique, nécessaires pour le pilotage et la coordination du système.

3.1.1. *Caractéristiques des robots*

Un robot de base que nous appelons simplement « robot » est une plate-forme de base sur laquelle tous les modules fonctionnels seront montés. Les principales caractéristiques de la plateforme de base sont :

- L'identificateur unique de la base, qui est une façon de distinguer les différents robots.
- La vitesse par défaut et la vitesse maximale du robot de base (Le projet IWARD a proposé une méthode issue de l'intelligence artificielle pour estimer la vitesse moyenne d'un robot, mais par manque de ressources cette méthode n'a pu être adoptée.
- L'autonomie de la batterie (en kW) et la consommation d'énergie du robot de base dans les 2 états : en attente et en mission.
- Le temps de recharge du robot.

Exemple en XML :

```
<RobotBase>
```

```

<idRobot>2</idRobot>
<Speed> 1.6 </ Speed >
<MaxSpeed> 1.6 </ MaxSpeed >
<IdleAutonomy> 480 </ IdleAutonomy >
<BusyAutonomy > 480 </ BusyAutonomy >
<RechargingTime> 150 </RechargingTime>
</RobotBase>

```

3.1.2. Fonctionnalités et Modules

Une des caractéristiques du projet IWARD est la modularité des robots : les différents modules peuvent être facilement branchés ou débranchés d'un robot. Chacun de ces modules fournit une ou plusieurs fonctionnalités comme le nettoyage de routine, le nettoyage des déversements, la surveillance de l'environnement, l'orientation des patients et le transport.

Les caractéristiques principales de ces modules et qui doivent être prises en compte par le système robotique (essaim / groupe) sont:

- L'identifiant unique de chaque module.
- Le type ou la catégorie du module.
- Les différentes fonctionnalités fournies par ce module.
- La consommation d'énergie et l'autonomie d'alimentation (le cas particulier de l'autonomie du module de nettoyage doit être pris en compte)
- L'influence sur la vitesse du robot.
- La liste des modules ne pouvant pas être montés sur le même robot.
- La procédure de montage et le degré de difficulté pour le montage et le démontage du module, précisant si la présence d'un technicien qualifié est nécessaire pour l'opération de montage.

Exemple en XML :

```

<Module>
  <idModule> 2 </ idModule >
  <listOfTaskAccomplissable>
    <taskType>2</taskType>
  </listOfTaskAccomplissable>
  <ConflictingModules>
    <idModule>1</idModule>
  </ConflictingModules>
  <MainBatteryConsumption> 50 </MainBatteryConsumption>
  <OwnBatteryAutonomy>500</OwnBatteryAutonomy>
  <PowerSource> Own Battery </ PowerSource >
  <IntelligentModule> Yes </IntelligentModule>
  <CanBeMountedAutomaticly>Yes</CanBeMountedAutomaticly>
  <CanBeDismountedAutomaticly>No</CanBeDismountedAutomaticly>
</Module>

```

3.1.3. Configurations

La configuration d'un robot doit respecter les contraintes suivantes :

- Le nombre maximal de modules pouvant être montés sur le robot.
- La compatibilité des modules.

En raison du petit nombre de modules possibles sur un robot (2 à 4), une énumération de toutes les configurations possibles semble être raisonnable.

Exemple en XML :

```
<Configuration>
  <idConfiguration> 3 </idConfiguration>
  <ConfigurationsElements>
    <idModule>2</idModule>
    <idModule>5</idModule>
  </ConfigurationsElements>
</Configuration>
```

3.1.4. Points de stationnement

A- Station de recharge et de configuration (C/R)

La plupart des robots doivent être reconfigurés afin de répondre aux changements de demandes durant une journée. De même, l'autonomie limitée de la batterie d'un robot oblige un rechargement régulier de sa batterie, le rendant indisponible sur des périodes de rechargement relativement longues (plusieurs heures). Il est alors nécessaire de redistribuer les modules des robots qui se rechargent sur d'autres robots actifs. Cela motive le choix d'un même lieu pour la recharge et la configuration.

Durant le projet, nous avons également envisagé l'utilisation de batteries amovibles, ce qui permettrait d'accroître le nombre de robots disponibles, puis ce qu'il il suffit de remplacer une batterie épuisée par une autre chargée. Les tests techniques avec les batteries amovibles n'étaient cependant pas concluants.

La station de recharge et de reconfiguration (station C /R) est représentée par les données suivantes:

- L'identifiant unique de la station C/R. (Généralement, il y a une seule station C/R par étage)
- L'emplacement de la station (cordonnées x, y ou bien identifiant de l'emplacement)
- Le nombre de chargeurs disponibles
- Le nombre maximal de robots pouvant être hébergés dans une station donnée.

Exemple en XML :

```
<CRStation>
  <idCRStation> 3 </idCRStation>
  <Capacity> 2 </Capacity>
  <NbCharger>1</NbCharger>
  <idLocation > 2 </idLocation>
</CRStation>
```

B- Station d'attente

La station d'attente est l'endroit où les robots retournent après avoir terminé leurs tâches, dans l'attente de l'arrivée d'autres tâches.

Une station d'attente peut être modélisée ou représentée à l'aide des données suivantes:

- L'identifiant unique de la station d'attente
- L'emplacement de la station d'attente
- La capacité maximale de stationnement ou le nombre maximal de robots pouvant y être stationnés

Exemple en XML:

```
<HomeStation>
  <idHomeStation> 3 </idHomeStation>
  <Capacity> 2 </Capacity>
  <idPOI > 2 </idPOI>
</HomeStation>
```

3.1.5. Points d'intérêt

Les points d'intérêt sont des repères connus du système robotiques comme les chambres, les couloirs, la pharmacie, ... Ils permettent d'identifier les lieux de missions, les positions et les trajectoires des robots.

La formulation d'un point d'intérêt est décrite par les données suivantes:

- Identification du point d'intérêt
- Coordonnées X et Y du point
- Nom formel du point d'intérêt
- Nom du département ou du service auquel appartient le point d'intérêt

Exemple possible en XML :

```
<POI>
  <idPOI>2</idPOI>
  <POIxCoord>150</ POIxCoord>
  < POIyCoord>230</ POIyCoord>
  <POIdépartement>Pharmacy</POIdépartement>
  <POIName>Home Station 1 - Pharmacy</POIName>
</POI>
```

3.1.6. Caractéristiques des missions

Le projet IWARD considère cinq familles ou types de missions: la livraison, le nettoyage, la téléconférence, la surveillance et le guidage des patients. Ces missions nécessitent des modules différents et donc une configuration adaptée aux missions des robots.

Chaque type de mission peut être modélisé ou représenté en utilisant les données suivantes :

- Identifiant du type de mission
- Nom attribué à ce type de mission
- Liste des missions incompatibles pour les exécutions en parallèle
- Description formelle des interruptions possibles aussi bien que les procédures de prise en charge de ces interruptions

Exemple en xml d'une classe/ type de missions :

```
<TaskType>
```

```

    <idTaskType >2</idTaskType>
    <Label> cleaning </Label>
    <IncompatibleTasks>
      <idTaskType >2</idTaskType>
      <idTaskType >5</idTaskType>
      <idTaskType >1</idTaskType>
    </IncompatibleTasks>
  </Interruptions>
</TaskType>

```

Les robots doivent accomplir des missions demandées par le personnel hospitalier. Ces demandes peuvent être régulières comme pour les missions de nettoyage de routine ou imprévues comme une mission de nettoyage suite à un accident qui peut avoir lieu à n'importe quel moment de la journée.

Dans cette thèse, une mission est caractérisée par les informations suivantes :

- Identifiant unique de la mission
- Type de mission
- Fonctionnalités ou modules exigés
- Lieu de départ et lieu de fin de mission
- Date de début au plus tôt, date de début souhaitée et date de début au plus tard
- Date de fin souhaitée et date de fin au plus tard
- Priorité de la mission
- Personnes à contacter en début et fin de mission
- Durée estimée de la mission
- Possibilité d'interruption et d'abandon de la mission
- Trajectoire de la mission (chemin, points d'intérêt à visiter)
- Mission mono-robot ou multi-robots

Exemple en XML :

```

<Task>
  <idTask> 88 </idTask>
  <taskType>2</taskType>
  <idStartLocation> 2 </idStartLocation>
  <idEndLocation> 5 </idEndLocation>
  <PreferredStartingTime>12:10:02</PreferredStartingTime>
  <LatestStartingTime>12:25:02</LatestStartingTime>
  <DueDate>12:59:59</DueDate>
  <LatestDueDate >13:09:59</LatestDueDate>
  <DueDate>13:59:59</DueDate>
  <Priority>2<Priority>
  <RequestingUser>2</RequestingUser>
  <ReceivingUser>3</ReceivingUser>
  <Scheduled> Yes </Scheduled>
  <CanBeInterrupted>Yes</CanBeInterrupted>
  <CanCollaborate>No</CanCollaborate>
  <MissionPath>
    <idLocation>12</idLocation>
    <idLocation>5</idLocation>
    <idLocation>2</idLocation>
    <idLocation>10</idLocation>
  </MissionPath>
  <SingleRobotTask>Yes</SingleRobotTask>
  <CanBeDelegated>Yes</CanBeDelegated>
  <ExpectedMissionDuration>45</ExpectedMissionDuration>
  <idStartingEvent>4</idStartingEvent>

```

```

    <idEndEvent>5</idEndEvent>
  </Task>

```

3.2. Modèle dynamique

L'une des particularités de notre projet est la notion de « *Shared knowledge* », (connaissances partagées) sorte de base de données distribuée qui est mise à jour en permanence par la couche de communication. Cette base de données contient les informations statiques du système robotique présenté précédemment mais aussi les données dynamiques du système

Le pilotage du système robotique présenté dans cette thèse repose sur les informations suivantes : état des robots, état des missions, état des modules, plan de navigation des robots, base de connaissances partagées « Shared Knowledge ».

3.2.1. Etats des robots

Le comportement dynamique d'un robot peut être décrit par le diagramme d'états-transitions de la Figure 5 où l'on distingue quatre cycles de fonctionnement. Plus généralement, un robot possède les quatre états suivants :

- (i) Etat d'attente. Le robot est alors soit sans missions connues soit en attente de la prochaine mission programmée à une date lointaine.
- (ii) Etat de travail, en exécutant une mission ou même plusieurs en parallèle.
- (iii) Etat de reconfiguration, soit pour ajouter soit pour supprimer des modules au robot.
- (iv) Etat de recharge de ses batteries pour pouvoir exécuter de futures missions.

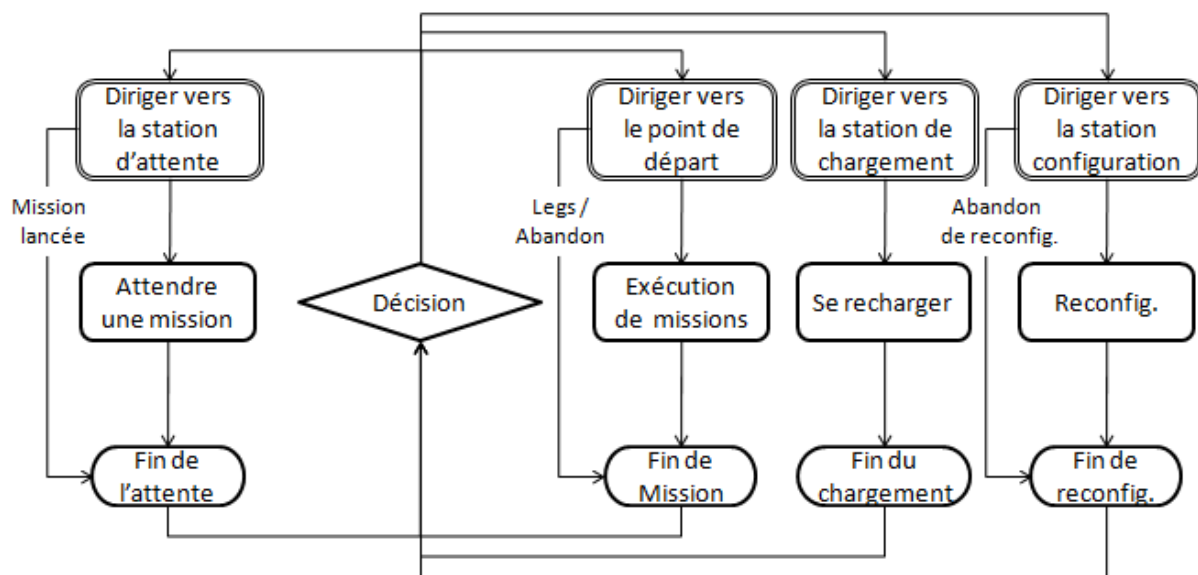


Figure 5 : Diagramme d'états-transitions des robots

En plus de ces états, d'autres données dynamiques concernant les robots sont nécessaires pour trouver la meilleure coordination. Ces informations sont synthétisées par la liste suivante :

- emplacement ou localisation exacte du robot,
- liste des tâches affectées au robot,

- niveau d'énergie restante dans les batteries du robot,
- état de la tâche en cours (En exécution, En conflit, Abandon ...),
- charge de travail affectée au robot,
- modules attribués au robot,
- station d'attente assignée au robot,
- liste des robots à proximité. Une méthode se basant sur la technologie du Bluetooth, devait être implémentée pour déterminer les robots sont dans un certain diamètre autour du robot.

3.2.2. Etats des Missions

La Figure 6 est le diagramme d'états-transitions d'une mission. Le processus démarre par la réception d'une nouvelle mission, ensuite l'agent de décision responsable de l'affectation des missions, en prenant en compte les caractéristiques de la mission ainsi que l'état des robots, affecte la mission au(x) robot(s) le(s) plus approprié(s).

Une fois la mission affectée, le composant responsable de l'exécution des missions ; muni des caractéristiques de la demande ainsi que le type de la mission ; instancie une squelette de missions du type adéquat. Cette instance est mise à jour au fur et à mesure de l'avancement de la tâche.

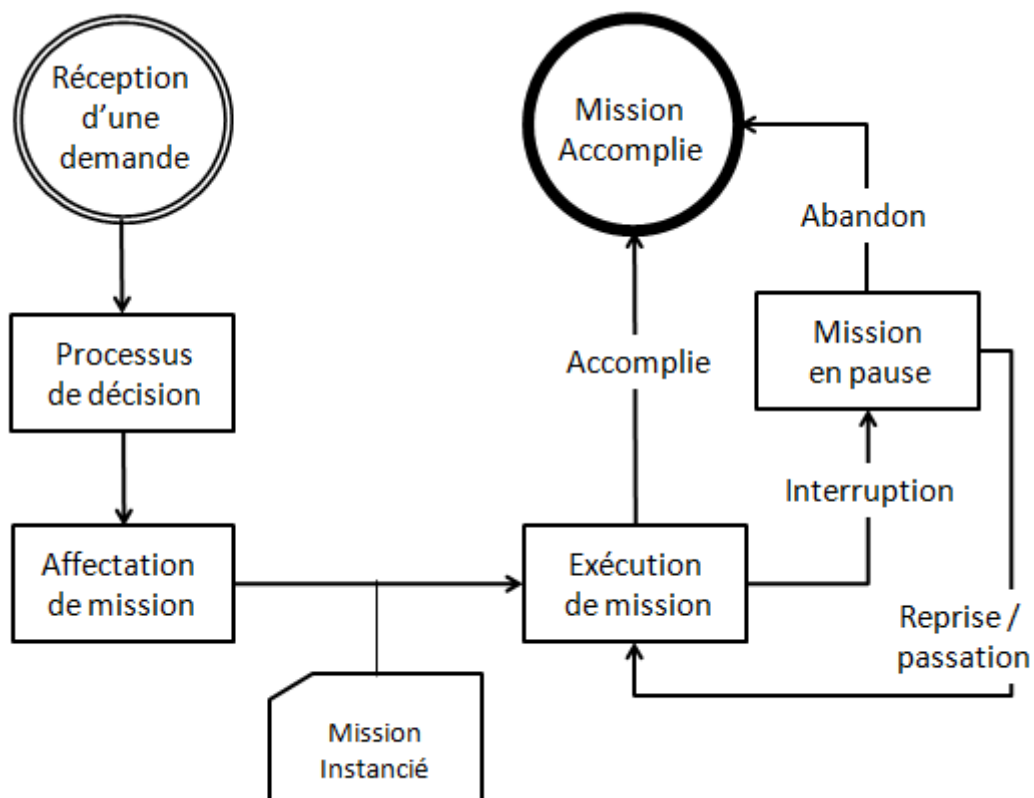


Figure 6 : Graphe d'état-transitions d'une mission

Les principales données dynamiques concernant l'état d'une mission sont ;

- Etat de la mission,
- Etape en cours d'exécution,

- Emplacement actuel du robot exécutant la mission,
- Direction vers laquelle se dirige le robot pour accomplir l'étape suivante,
- Estimation du temps restant de l'étape actuelle, ainsi qu'une estimation de la date de fin de la mission. (Ces informations devaient être mise à jour par un agent d'apprentissage artificiel)

Un exemple d'une instance dynamique d'une mission est illustré par l'entité XML suivante :

```
<MissionRuntimeObject>
  <idMission>88</idMission>
  <idTask>9</idTask>
  <missionType>spillage_clean</missionType>
  <currentMissionStep>spillCleanStartCleaning </currentMissionStep>
  <currentNearestPOI>2</currentNearestPOI>
  <nextPOI>3</nextPOI>
  <StartDate>13:59:59</StartDate >
  <ExpectedMissionDuration>45</ExpectedMissionDuration>
  <EndDate/>
  <MissionState>Mission Execution</MissionState>
  <currentExecutionRobot>3</currentExecutionRobot>
</MissionRuntimeObject>
```

Notons quatre principaux états d'une mission : (i) L'état instancié, où la mission est planifiée et attend la date de début pour être exécuté (ii) L'état d'exécution, où la mission est exécutée par le robot, ensuite soit la mission est interrompue et elle sera en (iii) état de pause, soit elle est achevée et elle sera en (iv) état terminé. On note qu'à partir de l'état de pause, la mission peut, soit être reprise par le même robot, soit être transmise à un autre robot, soit être abandonnée.

3.2.3. Etats des modules

La base de connaissances distribuées « *Shared knowledge* »(connaissances partagées) contient les informations sur l'état des modules. Les données dynamiques d'un module sont synthétisées par la liste suivante :

- Etat du module soit en stock, soit monté sur un robot,
- Emplacement physique du module, s'il n'est pas monté sur un robot,
- Identifiant du robot sur lequel est monté le module, s'il n'est pas stocké.

3.2.4. Plan de navigation dynamique

Le plan de navigation dynamique sert à décrire l'état de l'environnement, des chemins et des liaisons ainsi que l'emplacement actuel de chaque robot. Il est caractérisé par les informations suivantes :

- Etat de chaque chemin (arc reliant deux points d'intérêt POI).
- Durée moyenne estimée pour traverser chaque chemin.
- Emplacement actuel de chacun des robots.
- Destination de chacun des robots.
- Dernier emplacement connu de chaque robot.

3.2.5. Connaissances partagées « Shared Knowledge »

Le tableau suivant, réunit les entités dynamiques ainsi que statiques qui figurent dans la base de connaissance partagée « *Shared Knowledge* ».

<i>Plan:</i>
• <i>Schedule (Plan) for each robot and</i>
<i>Robots:</i>
• <i>Actual Configuration: modules mounted on each robot</i> • <i>Preferred Configuration: configuration preferred by plan</i> • <i>Home station: current home station</i> • <i>Scheduled recharging time: next scheduled battery charging time</i> • <i>Missions :</i> ○ <i>missions requested</i> ○ <i>missions assigned</i> ○ <i>missions in progress</i> ○ <i>current step / submission</i> ○ <i>next target/location of the robot</i> • <i>Location status: actual location of the robot given by the navigation module</i> • <i>Battery status: remaining power level provided by power monitoring device of the base robot</i>
<i>Missions:</i>
• <i>List of all missions ordered including their status</i>
<i>Modules:</i>
• <i>Location of the module</i> • <i>Busy or idle state of the module</i> • <i>On which robot it is mounted in case of busy status</i>

3.3. Evénements

Le système robotique implémenté repose sur une architecture de décision réactive, qui inspecte les événements et par conséquent prend des décisions. La compréhension des événements est une étape cruciale. La détection d'un événement incite une série de décisions et déclenche une série d'actions pouvant changer dans certain cas l'état du système robotique.

Nous pouvons classer les événements en 2 catégories : les événements initiés par les acteurs humains (demande d'une nouvelle mission) et les événements initiés par un robot suite à l'interprétation des données recueillies par les différents capteurs du robot (détection d'une personne allongée par terre ou arrivée à une destination).

Nous pouvons aussi classer les événements selon leur impact sur le système robotique. Deux classes d'événements sont considérées: les événements globaux affectant l'ensemble des robots et les événements locaux n'affectant que le robot lui-même.

Nous considérons les événements globaux suivants qui doivent être pris en compte par l'ensemble des robots:

- (i) l'arrivée d'une nouvelle demande à exécuter,
- (ii) la réaffectation d'une tâche,
- (iii) la remise en cause d'une des tâches,
- (iv) l'achèvement d'une mission,
- (v) le départ d'un robot du système robotique,
- (vi) l'intégration d'un nouveau robot,
- (vii) la fin d'une période et le début d'une nouvelle période,
- (viii) les événements issus des agents d'apprentissage, où un robot informe ses pairs, des informations qu'il a découvert et qui pourrait être utiles pour les autres membres de l'équipe, exemple : un obstacle bloquant un chemin.

Les événements locaux suivants sont considérés :

- (i) l'affectation d'une tâche,
- (ii) le démarrage d'une mission,
- (iii) l'interruption d'une tâche.

Dans certains cas, un événement local peut déclencher une série d'actions pouvant engendrer des événements globaux.

3.4. *Architecture de décision*

Dans cette thèse, nous nous limitons aux décisions suivantes :

- La gestion des plannings de rechargement des robots,
- La gestion de la configuration des robots et des modules,
- La localisation des stations d'attente des robots,
- L'affectation des missions aux différents robots,
- La planification et l'ordonnancement des missions affectées,
- La gestion et le contrôle de l'exécution des missions,
- La gestion de l'exécution des missions en parallèle,
- La gestion des interruptions de mission,
- La détection de la possibilité de passation de missions,
- La gestion locale d'énergie.

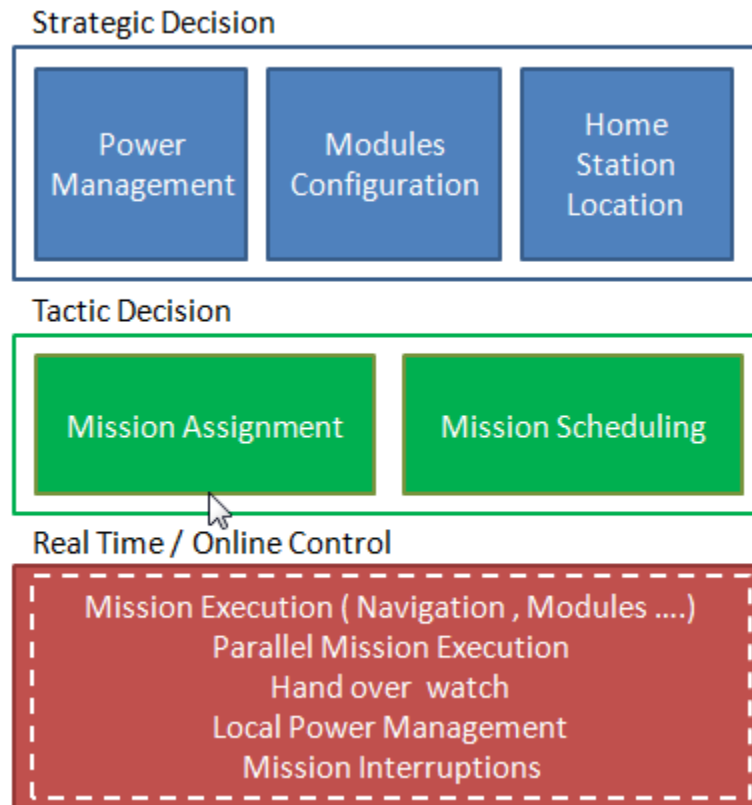


Figure 7 : Trois niveaux de décision

La Figure 7 illustre notre architecture de décision en trois niveaux :

- a) Le niveau stratégique,
- b) Le niveau tactique,
- c) Le niveau opérationnel et temps réel.

3.4.1. Niveau stratégique

Les décisions du niveau stratégique, illustrées par la Figure 8 et considérées dans cette thèse, sont :

- La gestion des plannings de rechargement des robots. Sachant que les robots sont équipés de batteries fixes nécessitant des durées de rechargement assez conséquentes, une bonne coordination des rechargements des robots est nécessaire afin d'assurer une disponibilité suffisante des robots dans le temps.
- La gestion des configurations des robots. Une des particularités du robot IWARD est la modularité permettant de modifier les fonctionnalités d'un robot en ajoutant ou supprimant des modules. Malgré la simplicité de cette opération, ces reconfigurations nécessitent une intervention humaine, d'où la nécessité d'un agent de décisions pour indiquer les configurations des robots actifs, tenant compte des besoins à venir.
- La localisation des stations d'attente. Afin de répondre efficacement et rapidement aux futures demandes, les robots doivent être correctement positionnés dans l'hôpital pour être proches des lieux de missions.

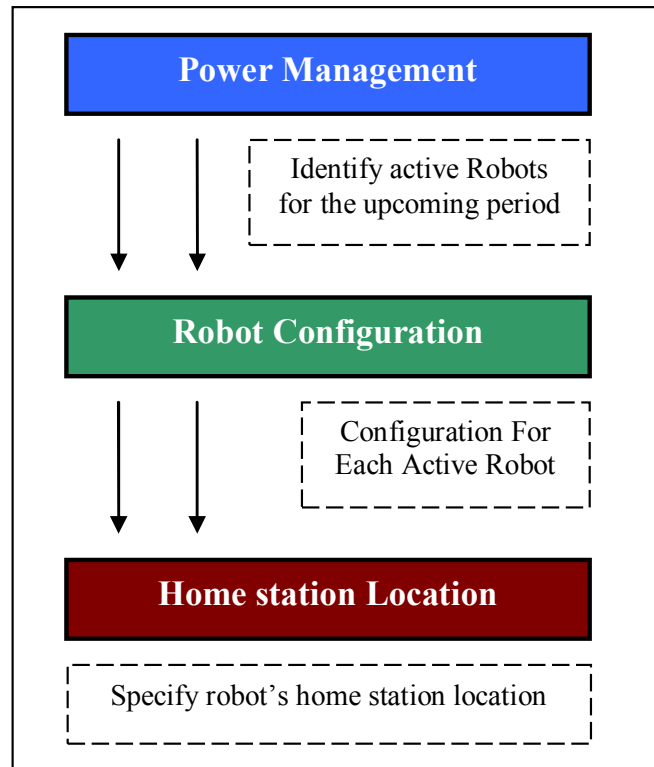


Figure 8 : Niveau stratégique

Un des points communs de ces agents de décisions stratégiques, c'est que le temps et l'horizon de planification, durant lesquels ces décisions sont prises, sont relativement longs et concernent généralement plusieurs périodes consécutives voire même une ou plusieurs journées. Ces décisions ne devront pas être remises en cause fréquemment. Généralement, les changements des décisions stratégiques coïncident avec le changement de postes qui accompagne généralement un changement des demandes de missions.

Afin d'établir ces décisions, l'état global du système robotique doit être considéré : notamment les robots, leur état, leur niveau d'énergie restant, les modules montés ainsi que les modules disponibles en stock, l'état des missions en cours ainsi qu'une prévision des futures missions. Toutes ces informations doivent être prises en considération afin d'établir, pour la prochaine période, quels robots vont se recharger et quels robots sont opérationnels, et établir pour chacun des robots opérationnels la configuration qui lui sera attribuée ainsi que la station d'attente qui lui sera assignée.

3.4.2. Niveau tactique

Au niveau tactique, nous nous intéressons aux problèmes d'affectation des missions aux robots et d'ordonnancement des missions sur chaque robot.

Ces décisions doivent prendre en compte les charges de travail des robots, l'état des robots, les recommandations du niveau stratégique, et les configurations des robots. Ces décisions doivent être prises assez rapidement, surtout pour les missions urgentes, afin de ne pas retarder leur exécution.

Les missions multi-robots ne sont pas prises en compte directement et nous supposons qu'elles sont décomposées en plusieurs sous-missions mono-robot que nous pouvons affecter séparément à différents robots. Néanmoins un système de compatibilité de missions est mis en place afin de ne pas affecter deux sous-missions au même robot.

Deux types de missions sont considérés : les missions régulières et connues en avance qui sont programmées régulièrement (transport régulier de médicaments, nettoyage programmé) et les missions imprévues qui ne sont pas connues en avance.

Deux approches de prise de décisions sont considérées : décisions centralisées et décisions distribuées.

3.4.2.1. *Prise de décision centralisée*

La caractéristique principale de cette approche est que les décisions sont prises par un agent central. Sachant que ces agents doivent s'occuper de l'affectation et de l'ordonnancement des missions, il est essentiel que ces agents soient informés en permanence de l'état des robots et des missions. La prise en compte de toutes ces informations complique le problème et risque de retarder la prise de décision, ce qui peut se traduire par un retard d'exécution des missions. En revanche, vu que ces agents ont une vue globale, la solution proposée peut être l'optimale.

Ce système peut être avantageux notamment dans le cas où la majorité des missions sont des missions régulières connues en avance, ce qui permet d'établir une affectation et un ordonnancement a priori des missions, et de disposer suffisamment de temps pour la prise de décision.

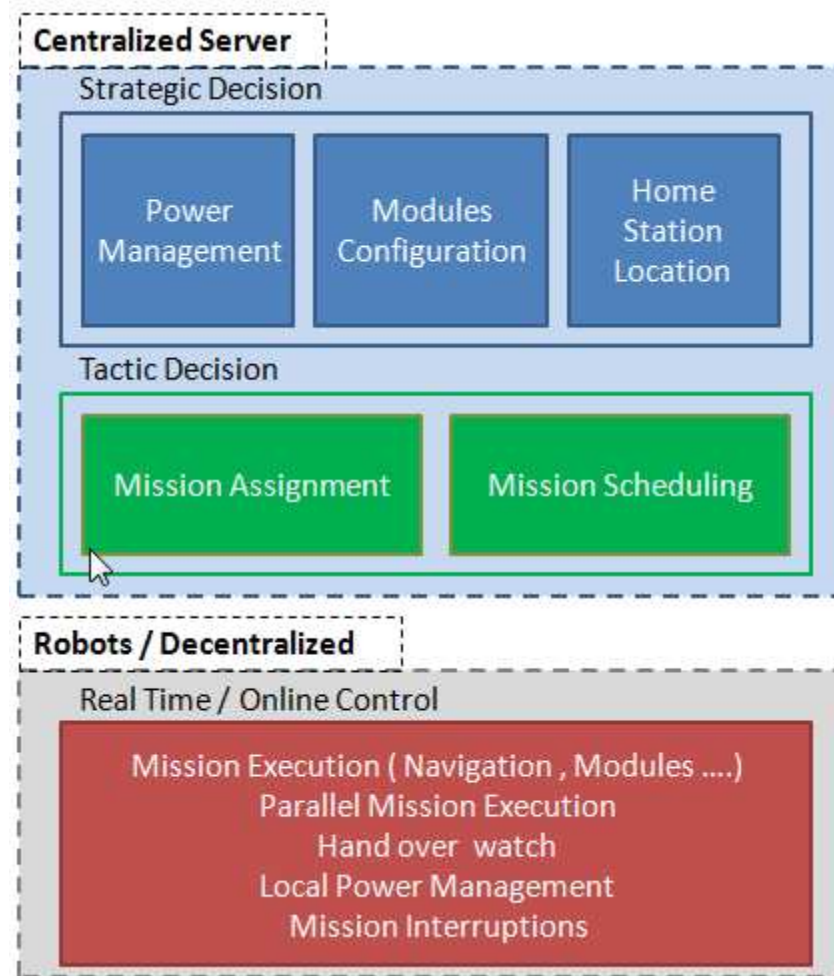


Figure 9: Approche centralisée de décisions tactiques

3.4.2.2. Prise de décision décentralisée

L'utilisation d'un agent central de décisions rend le système sensible aux défaillances des robots et des systèmes de communication. Pour cela, nous proposons également une approche de décision décentralisée. Le système décentralisé profite des capacités de calcul de chaque robot ainsi que de leurs facultés de communication. Dans ce système les robots sont indépendants, et les décisions sont prises en concertation avec les autres robots, le système implémenté repose sur des techniques dites d'enchères inversés et de négociation.

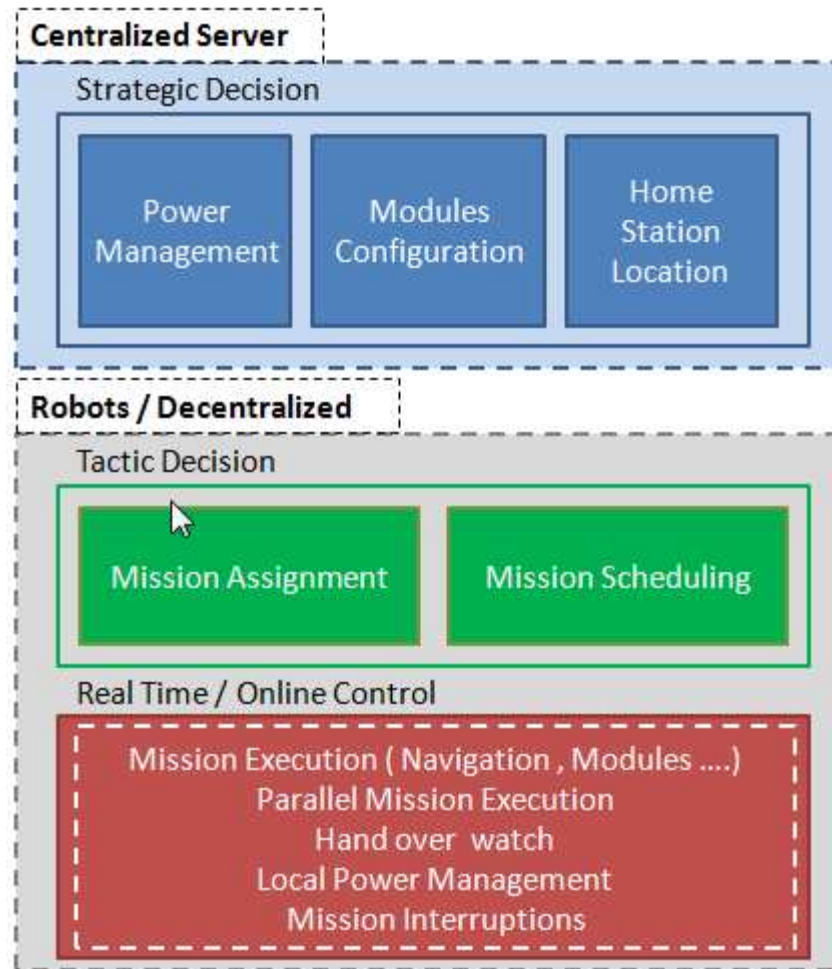


Figure 10 : Décisions tactiques décentralisées.

Afin de tester l'efficacité du système décentralisé, ce système est comparé à l'aide d'une simulation avec un système centralisé basé sur les algorithmes génétiques.

D'autres systèmes hybrides peuvent être envisagés, notamment un système d'affectation centralisé couplé avec un système d'ordonnancement local propre à chaque robot. Une autre possibilité serait d'utiliser le système centralisé pour les missions régulières, lors de l'arrivée de nouvelles missions urgentes ces décisions seront reconsidérées par un système décentralisé. Ce système décentralisé jouera alors un rôle réparateur qui s'adapte aux aléas en temps réel.

3.4.3. Niveau opérationnel et temps réel

Ce niveau gère l'exécution des missions. Le composant principal de ce niveau est l'exécuteur, qui a pour rôle d'exécuter les ordres du niveau tactique, et de veiller au bon déroulement des missions. L'exécuteur doit interagir avec les éléments robotiques, analyser les données reçues par les capteurs, identifier les événements qui doivent être transmis au niveau tactique, et transmettre des ordres aux éléments robotiques afin d'exécuter les différentes étapes des missions. D'autres composants sont conçus pour identifier des opportunités d'optimisation. Le système de prospection de missions parallèles permet d'identifier les missions pouvant être

exécutées en parallèle par le même robot. Un autre système détecte la possibilité de transmettre la mission en-cours d'un robot à un autre aux alentours afin que le premier puisse être libéré pour une mission plus urgente.