

Exercice 10

Réalisation d'un simulateur en LOCAL d'un SI de la livraison de petits colis par Drone dans une grande ville

| | |
|---|----------|
| 1. PRÉSENTATION | 2 |
| 2. ENONCÉ | 4 |
| 2.1. DIAGRAMME DE COMMUNICATION | 4 |
| 2.2. DIAGRAMMES DE CLASSE | 4 |
| 3. CORRECTION | 5 |
| 3.1. DIAGRAMME DE COMMUNICATION | 5 |
| 3.2. CONCEPTION | 7 |
| 3.2.1. COMPOSANT 1 : « SERVEUR » | 7 |
| 3.2.2. COMPOSANT 2 : IHM DE GESTION DES COMMANDES | 7 |
| 3.2.3. COMPOSANT 3 : IHM DE SUIVI | 8 |
| 3.2.4. COMPOSANT 4 : LES DRONES | 8 |
| 3.2.5. COMPOSANT 5 : STRUCTURE D'ACCUEIL | 8 |
| 3.2.6. PROGRAMME PRINCIPAL | 8 |
| 3.3. DIAGRAMMES DE CLASSES | 10 |
| 3.3.1. COMPOSANT 1 : « SERVEUR » | 11 |
| 3.3.2. COMPOSANT 2 : IHM COMMANDES | 13 |
| 3.3.3. COMPOSANT 3 : IHM SUIVI | 14 |
| 3.3.4. COMPOSANT 4 : DRONES | 15 |
| 3.3.5. COMPOSANT 5 : Structure d'Accueil | 16 |

1. Présentation

Cet exercice consiste à s'exercer à faire des **Designs Patterns** dans un diagramme de classe d'un SI particulier dont tous les composants s'exécutent dans un même programme (JVM).

Tous les composants de ce SI sont donc liés par des appels de méthodes, aucune communication distante.

L'objectif est de créer un programme simulateur d'un Système d'Information (SI) permettant de gérer, depuis un dépôt unique, la livraison de petits colis par drone dans une grande ville.

La livraison se fait sur les toits des buildings qui sont équipés d'une structure d'accueil spécifique des drones.

Les drones suivent un parcours prédéfini, aller-retour, entre le dépôt et un des bâtiments. Ces bâtiments sont tous équipés d'un accès publique ouvert en journée.

Le Système d'Information (SI) est composé :

- d'un Serveur [COMPOSANT 1 (unique)],
- une IHM de gestion des commandes des colis [COMPOSANT 2 (unique)],
- des IHM permettant de suivre en temps réel les drones (1 IHM par drone) [COMPOSANT 3 (en N exemplaires)],
- les drones [COMPOSANT 4 (en N exemplaires)] et
- la structure d'accueil de livraison [COMPOSANT 5 (en K exemplaires)].

Le serveur utilise une interface distante sur laquelle il reçoit les commandes à livrer. Le serveur gère les COMMANDES, les PARCOURS prédéfinis de livraison et les DRONES.

Ici, on suppose que le colis est déjà préparé.

Le COMPOSANT 2 permet d'identifier le colis correspondant à une commande contenant l'adresse mail du client et l'adresse du building de livraison (que le client aura sélectionné lors de la création de sa commande via un site internet dédié). Un opérateur utilise le COMPOSANT 2 pour identifier la première IHM de suivi d'un drone (COMPOSANT 3) disponible. Puis une fois le drone chargé de son colis, le parcours est transmis au drone (COMPOSANT 4). L'opérateur déclenche son départ. Le drone est donc autonome pour réaliser son parcours.

Chaque drone (COMPOSANT 4) envoie à son IHM de suivi (COMPOSANT 3), en temps réel, son état (identification, position, et altitude) et le flux vidéo d'une caméra embarquée.

L'IHM de suivi (COMPOSANT 3) permet de suivre sur une carte le vol du drone et d'afficher ce qu'il voit à travers la caméra embarquée. En cas de tout dysfonctionnement ou par décision de l'opérateur, l'IHM de suivi permet de passer en mode manuel.

Une fois le drone arrivé à destination, le colis est déposé automatiquement et la structure d'accueil (COMPOSANT 5) dépose automatiquement le colis dans une boîte sécurisée munie d'un digicode. La structure d'accueil prévient alors le serveur (COMPOSANT 1) que la livraison a été réalisée en lui donnant le numéro du casier. Le drone revient à sa base automatiquement. Le serveur génère un code aléatoire pour accéder au casier et programme le casier avec ce code. Il envoie un mail au client de la commande contenant le code d'accès pour le prévenir que la livraison a été effectuée.

Etant donné que l'on veut simuler le comportement de ce SI :

- le serveur sera simulé par une classe
- les IHM seront simulés par des classes qui propose une interface graphique réduite
- les DRONES seront simulés par des instances d'une même classe. Chaque instance contient un thread qui simule le comportement du Drone

- les structures d'accueil de livraison sont simulées par des instances d'une même classe élémentaire
- l'envoi de mail est simulé par une trace dans un fichier de log

2. Enoncé

2.1. Diagramme de communication

Faire le diagramme de communication de ce SI indépendamment que ce SI soit simulé.

Compléter ce diagramme de communication avec l'aspect simulateur, si besoin.

2.2. Diagrammes de classe

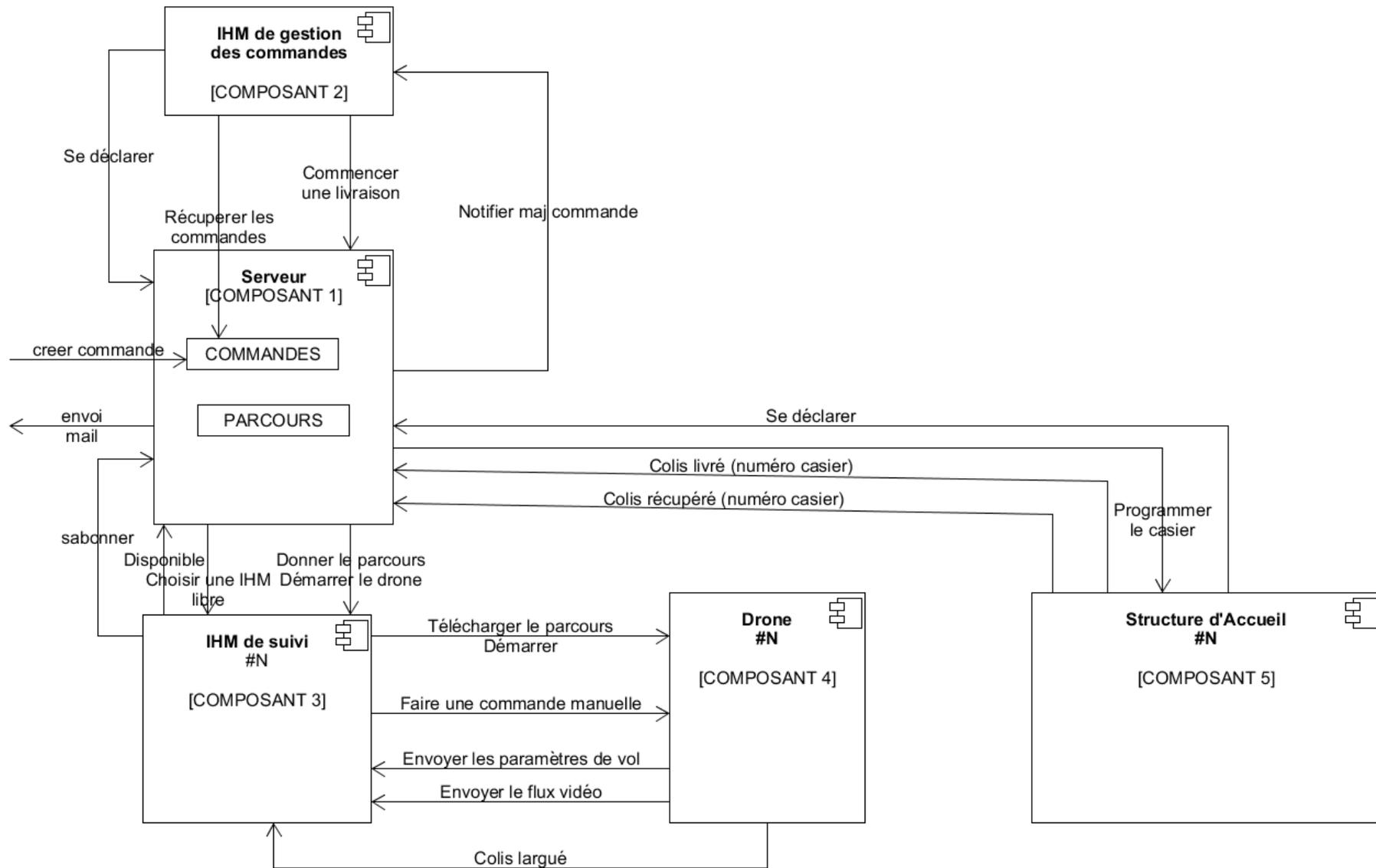
Faire les diagrammes de classes de ce programme en faisant la part des choses entre les classes propres au SI et les classes propres au simulateur.

Vous pourrez faire un diagramme de classe par COMPOSANT.

Pensez à bien compartimenter la communication entre ces packages à travers des **interfaces** précises car lors de la réalisation de ce SI en situation réelle les COMPOSANTS deviendront des composants distants et les classes propres au SI pourront être réutiliser et compléter.

3. Correction

3.1. Diagramme de communication



Le composant « **Serveur** » récupère les **commandes** permettant de simuler le SI.

Quand l'**IHM de gestion des commandes** s'exécute elle **récupère les commandes** du « serveur ».

L'opérateur choisit une commande (qui a été préparée sur un drone) et **commence une livraison** en prévenant le « serveur » en lui donnant le numéro du drone.

Le « serveur » **choisit** une IHM libre en interrogeant toutes les **IHM de suivi** qui **renvoie leur disponibilité**. Ensuite, il récupère le lieu de livraison (structure d'accueil) dans la commande (choisit par le client) et **donne le parcours** prédéfini correspondant à l'IHM de suivi. Puis demande à l'IHM de suivi de démarrer le drone.

L'IHM de suivi **télécharge le parcours** sur le drone, le **démarre**. À tout moment l'IHM de suivi peut **faire une commande manuelle** sur le drone. Le drone **envoie ses paramètres de vol** à l'IHM de suivi qui visualise le déplacement du drone, ainsi que le **flux vidéo envoyer** par le drone.

Une fois le drone arrivé à destination, il prévient son IHM de suivi que le **colis est largué**. Et revient à son point de départ.

La **Structure d'accueil** réceptionne le colis, choisit un casier libre, dépose le colis dans le casier, et prévient le « serveur » que le **colis est livré** en lui donnant le numéro du casier. Le « serveur » génère un code d'accès au casier, **programme le casier** sur la Structure d'accueil. Puis, **envoie un mail** au client que le colis est disponible avec le numéro d'accès, et met à jour la commande qui est **notifié** à l'IHM de gestion des commandes.

Quand le client récupère le colis, la Structure d'Accueil prévient le serveur que le **colis a été récupéré**. Le « serveur » met à jour la commande et **notifie** l'IHM de gestion des commandes. Le « serveur » **envoie un mail** au client.

Quand le drone est revenu à sa position de départ, l'IHM de suivi se met à disponible.

3.2. Conception

3.2.1. COMPOSANT 1 : « Serveur »

Le « serveur » va contenir :

- La classe SERVEUR qui est la classe principale et est un SINGLETON.
- Un FACTORY de COMMANDES qui sera aussi un SINGLETON
 - Chaque COMMANDE sera couplée à un PROXY pour notifier chaque maj de la commande à l'IHM de gestion des commandes.
 - Une COMMANDE contient un état (non préparé, préparé, en cours de livraison, livrée, récupérée) et l'adresse mail du client, le numéro du casier de livraison, le code d'accès au casier.
- Un FACTORY de PARCOURS qui sera aussi un SINGLETON. Un PARCOURS est une liste de position GPS dont le point d'arrivé est la position d'une Structure d'Accueil. Les PARCOURS sont dans un fichier de simulation

Afin de décorréler le « Serveur » et les IHM de suivi on va utiliser un OBSERVABLE. Les IHM se déclarent au serveur.

Pour déterminer quelles sont les IHM de suivi disponibles, le « Serveur » notifie toutes les IHM, et chaque IHM prévient le serveur qu'elle est disponible ou pas. Le « Serveur » doit gérer une liste des IHM disponibles.

Toutes les fonctions d'entrée sur le « Serveur » sont réalisées à travers une INTERFACE.

L'envoi de mail sera simulé par une écriture dans un journal.

3.2.2. COMPOSANT 2 : IHM de gestion des commandes

Une classe qui affiche la liste des commandes, permet de sélectionner une commande afin de commencer la livraison d'une commande.

Chaque notification rafraîchit la liste des commandes affichée.

A sa création, la classe se déclare au « Serveur ».

Toutes les fonctions d'entrée sur ce composant sont réalisées à travers une INTERFACE.

3.2.3. COMPOSANT 3 : IHM de suivi

On va simuler la création de 2 IHM de suivi.

A la création chaque IHM s'abonne au « Serveur » et crée un OBSERVER pour recevoir la demande de disponibilité.

Toutes les fonctions d'entrée sur chacun de ces composants sont réalisées à travers une INTERFACE (une par IHM).

Elle affiche l'image reçue par le drone dans une zone.

On simule le déplacement du drone en affichant les positions GPS reçues (on n'affiche pas de carte).

On simule une commande manuelle en envoyant le nom de la commande et ses paramètres au drone simulé.

3.2.4. COMPOSANT 4 : Les DRONES

On crée un FACTORY de DRONES pour gérer les drones qui sont la vision logique des Drones physiques (COMPOSANT 4).

Pour simuler les drones, on va créer un deuxième FACTORY de DRONES dont chaque produit contient un Thread qui envoie des paramètres de vol à l'IHM de suivi en simulant le déplacement du drone de son point de départ à son point d'arrivée. Le flux vidéo sera simulé par l'envoi d'images (l'image du dépôt, des images d'une rue, l'image d'une structure d'accueil).

Quand le parcours est terminé, le thread prévient l'IHM de suivi que le colis est largué.

Ces 2 factories sont décrits dans un DP abstraits de Factory.

L'instanciation de chacun des drones dans le factory est réalisé en fonction d'un fichier de configuration qui décrit le type de comportement de chacun des Drones (vol nominal, perte du signal GPS en cours de parcours, ...). On utilise un DECORATEUR pour cumuler les différents comportements).

Toutes les fonctions d'entrée sur un DRONE sont réalisées à travers une INTERFACE (une par drone)

On simule la prise en compte d'une commande manuelle en l'écrivant dans un fichier de log.

On simule la prise en compte du parcours (téléchargement) en l'écrivant dans le même fichier de log.

3.2.5. COMPOSANT 5 : Structure d'Accueil

On va simuler la création de 2 Structures d'Accueil.

A la création chaque Structure d'Accueil se déclare au « Serveur ».

Toutes les fonctions d'entrée sur chacun de ces composants sont réalisées à travers une INTERFACE (une par Structure d'Accueil).

Pour simuler les actions réalisées par la Structure D'accueil, on crée une interface supplémentaire afin que l'IHM de Suivi puisse simuler la réception du colis par la structure d'accueil et la récupération du colis par le client.

3.2.6. Programme principal

Le programme principal réalise le traitement suivant :

Début

Instanciation du Serveur (qui instancie les Factory de COMMANDES et de PARCOURS)

Instanciation de la classe principale de l'IHM de gestion des commandes (qui se déclare au serveur)

Instanciation des classes principales des 2 IHM de Suivi (qui s'abonnent au serveur) avec leur Id

Instanciation du Factory de simulation des drones et création de 2 Drones.

Instanciation des 2 Structures d'Accueil (qui se déclarent au serveur) avec leur Id

Envoyer des commandes au Serveur (jeu de données)

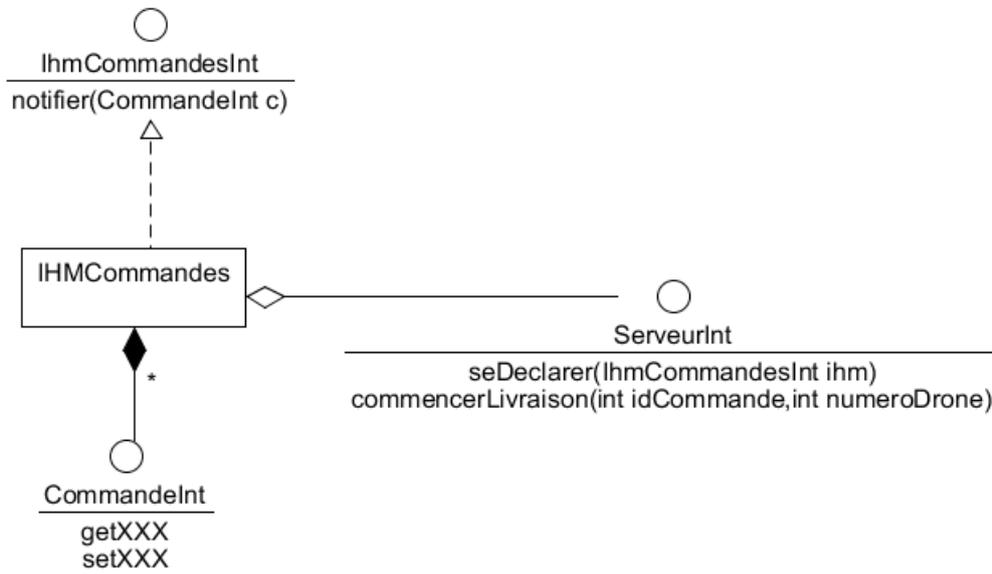
Fin

3.3. Diagrammes de classes

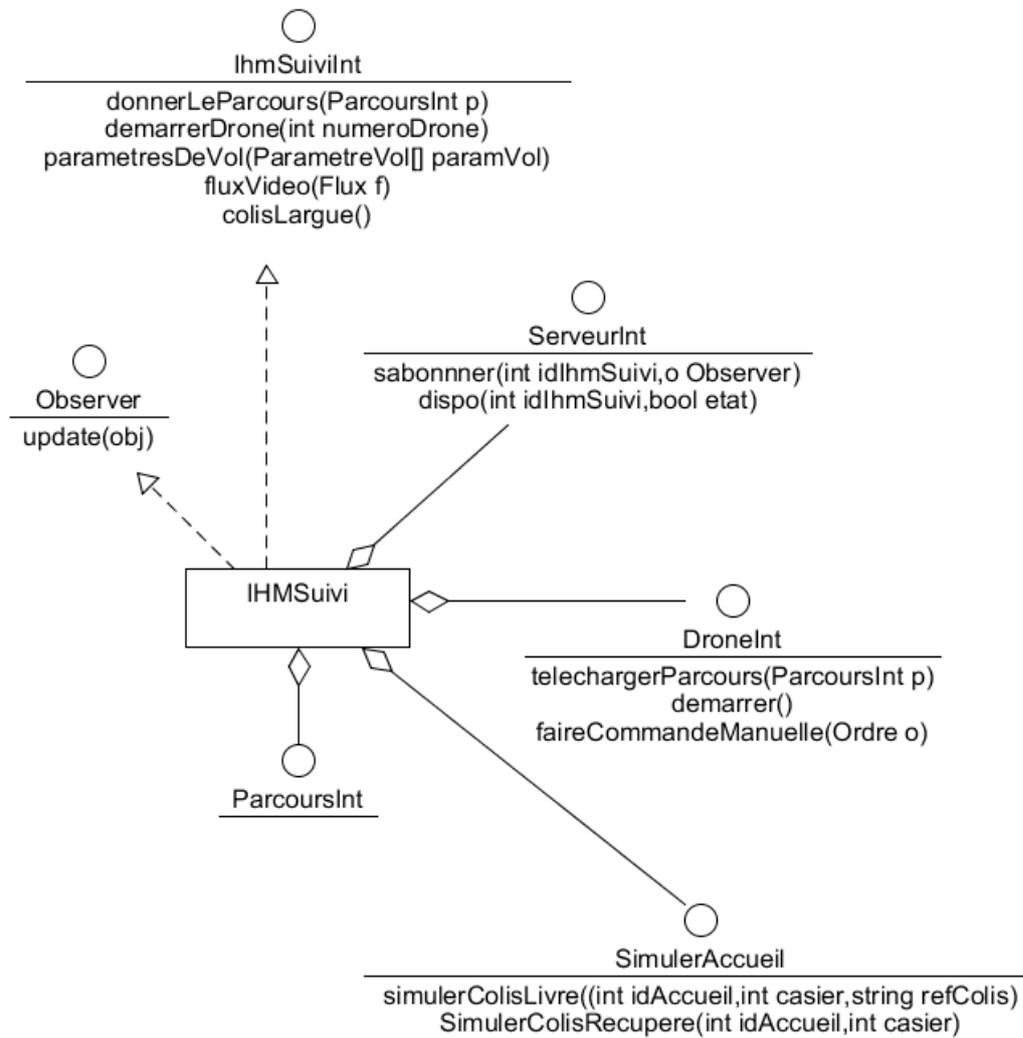
COMMENTAIRES EN SEANCE DE COURS

3.3.1. COMPOSANT 1 : « Serveur »

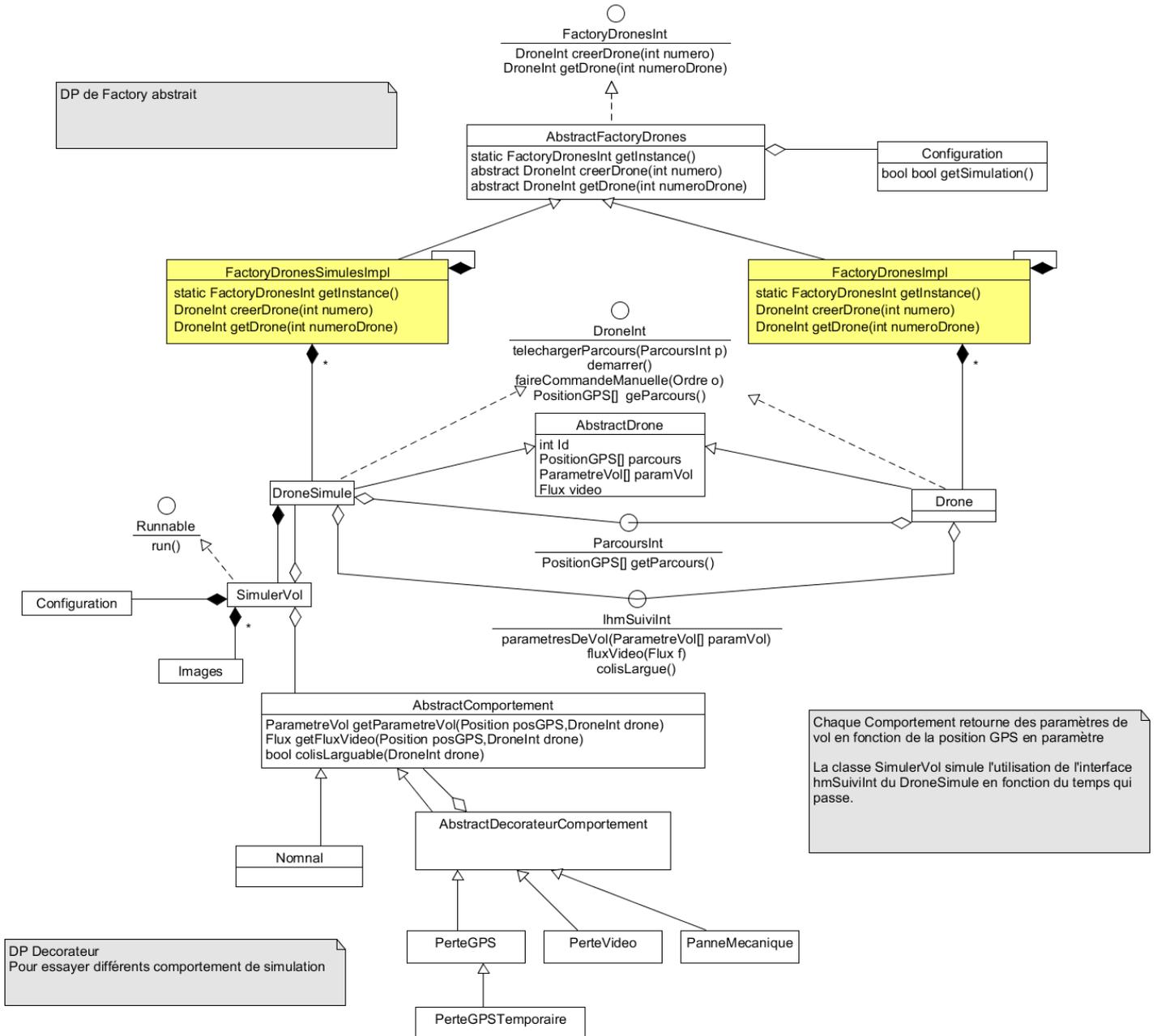
3.3.2. COMPOSANT 2 : IHM Commandes



3.3.3. COMPOSANT 3 : IHM Suivi



3.3.4. COMPOSANT 4 : DRONES



3.3.5. COMPOSANT 5 : Structure d'Accueil

