

IPST-CNAM  
Intranet et Designs patterns  
**NSY 102**  
Vendredi 9 Avril 2021

Durée : **1 h 30**  
Enseignants : LAFORGUE Jacques

1ère Session NSY 102

---

Correction de l'examen

**PROBLEME** [50 points] (1h30)

On veut réaliser le Système d'Information permettant de gérer la mise à jour des panneaux d'affichage des vols d'un grand aéroport.

Toutes les compagnies aériennes déposent les plans de vol (PLN) dans une base de données unique (BDPLN), au plus tard la veille de la date de départ de chacun des vols. (On ne traite pas les compagnies aériennes dans ce sujet).

Un poste IHM d'administration [COMPOSANT 3] de l'aéroport permet de configurer un serveur [COMPOSANT 1] afin qu'il récupère de la BDPLN, automatiquement, par exemple à 4h du matin, tous les plans de vols prévus pour la journée.

Les plans de vols récupérés sont stockés en mémoire du serveur. Le serveur est propre à l'aéroport.

Pendant la journée, à tout moment, un administrateur peut utiliser ce poste pour demander au serveur de mettre en standby, annuler, modifier l'heure de départ d'un vol ou modifier toute autre information du plan de vol. Il peut aussi créer, exceptionnellement, un nouveau plan de vol.

Le poste affiche tous les plans de vol et permet de faire toutes les commandes nécessaires sur un plan de vol.

Le rôle de ce serveur est aussi de gérer tous les panneaux d'affichage [COMPOSANT 2] répartis dans l'aéroport.

Il existe différents types de panneaux entre ceux destinés au public et ceux destinés à des techniciens : (enregistrement, douane, bagagerie, ...). Ainsi, en fonction de son type, chaque panneau n'affiche pas les mêmes informations. (Pour l'écriture de vos diagrammes, prenez 2 types de panneaux possibles)

Pour simplifier à notre niveau, un panneau est vu comme un composant d'IHM graphique lourd qui s'abonne, dès son lancement, au serveur en lui indiquant sa nature.

En fonction de sa nature, le serveur devra lui notifier les informations qu'il attend.

Pour fixer les idées (inutile de mettre tous ces attributs dans votre diagramme, cela étant implicite), voici un exemple des informations contenues dans un plan de vol : aéroport de départ, aéroport d'arrivée, heure de départ, heure d'arrivée, identifiant du vol, le type d'avion, le nom de la compagnie, le nombre de passager prévus, la nature du frète si exceptionnel, le niveau global de risque potentiel pour la douane, l'identité des personnes potentiellement dangereuses ou signalés comme suspects, le nombre de bagages, une demande de révision ou d'entretien pour l'avion, un texte libre d'information, ...

Dynamiquement, à chaque fois que le serveur met à jour une information d'un plan de vol, les panneaux d'affichage sont aussitôt notifiés de la modification.

Le serveur contient tout un ensemble de traitements algorithmiques (recherche, planification, filtre, tri, ...). Ces traitements utilisent directement les plans de vol mémorisés dans le serveur et peuvent donc aussi les modifier.

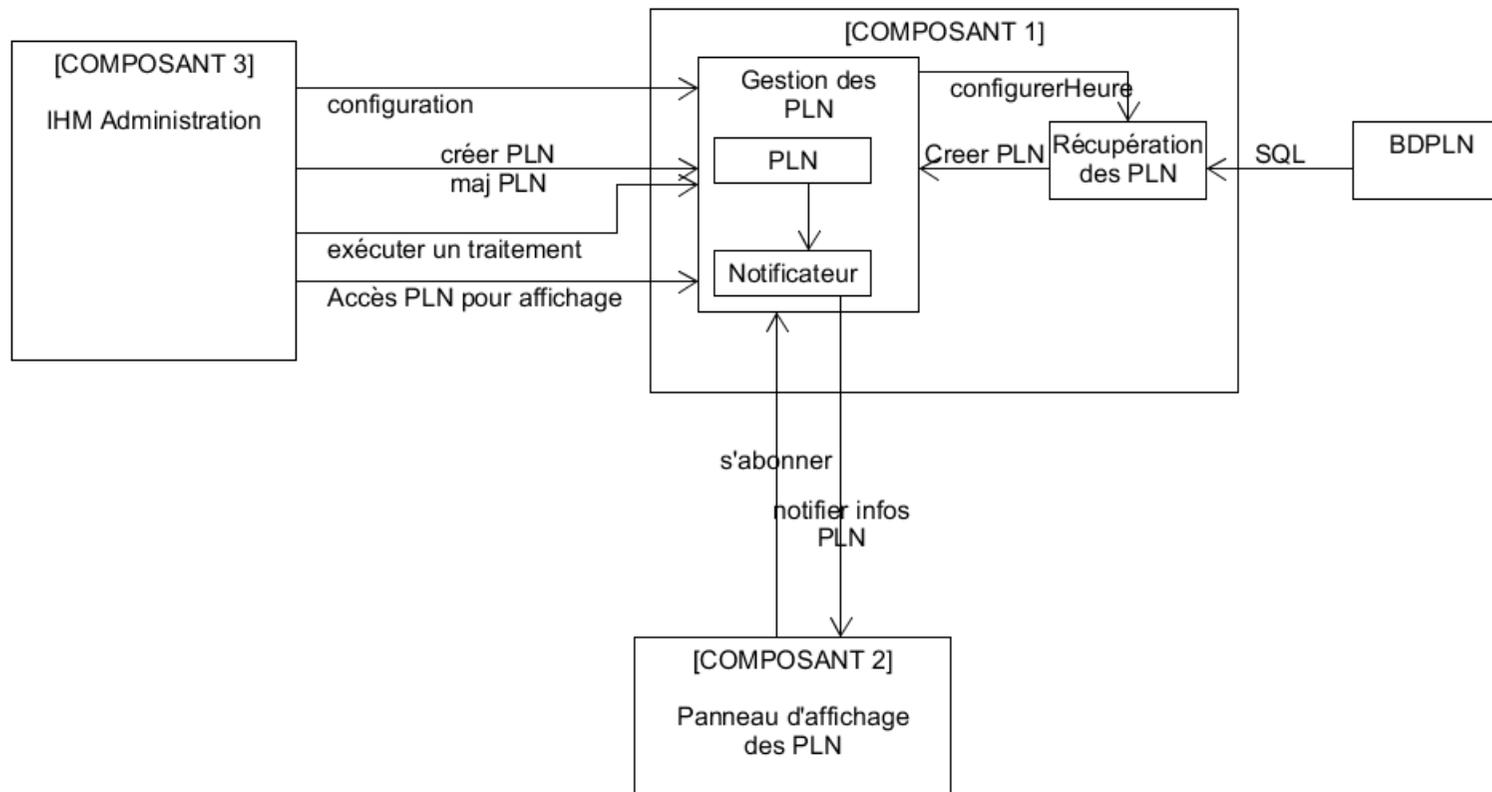
(Suite du sujet page suivante)

**PARTIE 1 [15 points] :**

Faites le diagramme de communication de ce Système d'Information [10 points].

Commentez votre schéma (rôles des composants, comportement dynamique général, échanges des informations, localisation des données) [5 points].

Nous rappelons que ce schéma doit permettre de connaître vos choix d'organisation des composants et des composants internes (sous-composants) de chacun des trois COMPOSANTS de ce SI, et vos choix d'appel entre les composants et sous-composants.



L'IHM Administration [COMPOSANT 1] utilise à distance le composant "Gestion des PLN" pour réaliser les traitements suivants :  
 - configurer le serveur [COMPOSANT 3] qui à son tour va changer l'heure de récupération des PLN.

- créer un PLN, mettre à jour un PN, exécuter un traitement
- accéder à tous les PLN afin de les afficher dans son IHM.

Le serveur [COMPOSANT 3] est constitué de deux sous-composants :

- "Gestion des PLN" qui gère en mémoire tous les PLN. Ce sous-composant permet aux panneaux de s'abonner et notifie les panneaux à chaque fois qu'un PLN est créé ou modifié.
- "Récupération des PLN" qui active automatiquement un traitement à l'heure configuré. Ce traitement lit la BD et met à jour 'Gestion des PLN' avec les PLN lus en BD.

Chaque panneau est une occurrence du [COMPOSANT 3]. Dès qu'un panneau est exécuté il s'abonne au "Gestion des PLN" qui est aussitôt notifié de tous les PLN, et qui sera ensuite également notifié à chaque fois qu'un PLN sera modifié.

## **PARTIE 2 [35 points] : (cette partie contient 3 sous-parties)**

1/ Faites le diagramme de classe UML du [COMPOSANT 1] en mettant en évidence les Designs Patterns vus en cours. **sur 15 points**  
Commentez le diagramme. **sur 5 points**



On crée un Factory de PLN (classe FactoryPLN) (DP Factory) qui est aussi un singleton (DP Singleton). Les PLN créés en local par ce factory sont couplés à un ProxyPLN dont le rôle sera de notifier les panneaux à chaque fois que l'on utilisera un setteur d'un PLN (modification d'un PLN).

Le factory maintient une liste de tous les panneaux à notifier. Cette liste est enrichie à chaque abonnement d'un panneau.

Le factory gère un Observable d'Observer distant (autant d'observer distant que de panneaux) (DP Observable distant).

Les panneaux sont gérés en autant de classe qu'il existe de type de panneau (DP Stratégie). Chaque classe permet ainsi de créer les informations spécifiques à chaque panneau à partir d'un PLN. C'est cette information qui est notifiée par l'ObservablePanneaux à chacun des panneaux.

Quand un panneau s'abonne, il précise sa nature (type de Panneau) et son interface d'observer distant.

Chaque panneau est un observer local.

Pour que l'IHM puisse utiliser le FactoryPLN à distance on crée un factory distant (FactoryPLNOD) qui utilise le FactoryPLN en singleton (DP OD par Composition).

La récupération des PLN en base de données se fait par la classe RecupererPLN qui va exécuter un thread dont le rôle est de lire la BD afin de récupérer les PLN qui concerne notre aéroport et mettre à jour le GestionPLN à travers son singleton.

2/ Ecrire le pseudo-code ou le code Java des traitements suivants du COMPOSANT 1 :

- la création d'un nouveau PLN
- l'abonnement d'un panneau au serveur
- la notification des panneaux, suite à la modification d'un plan de vol.

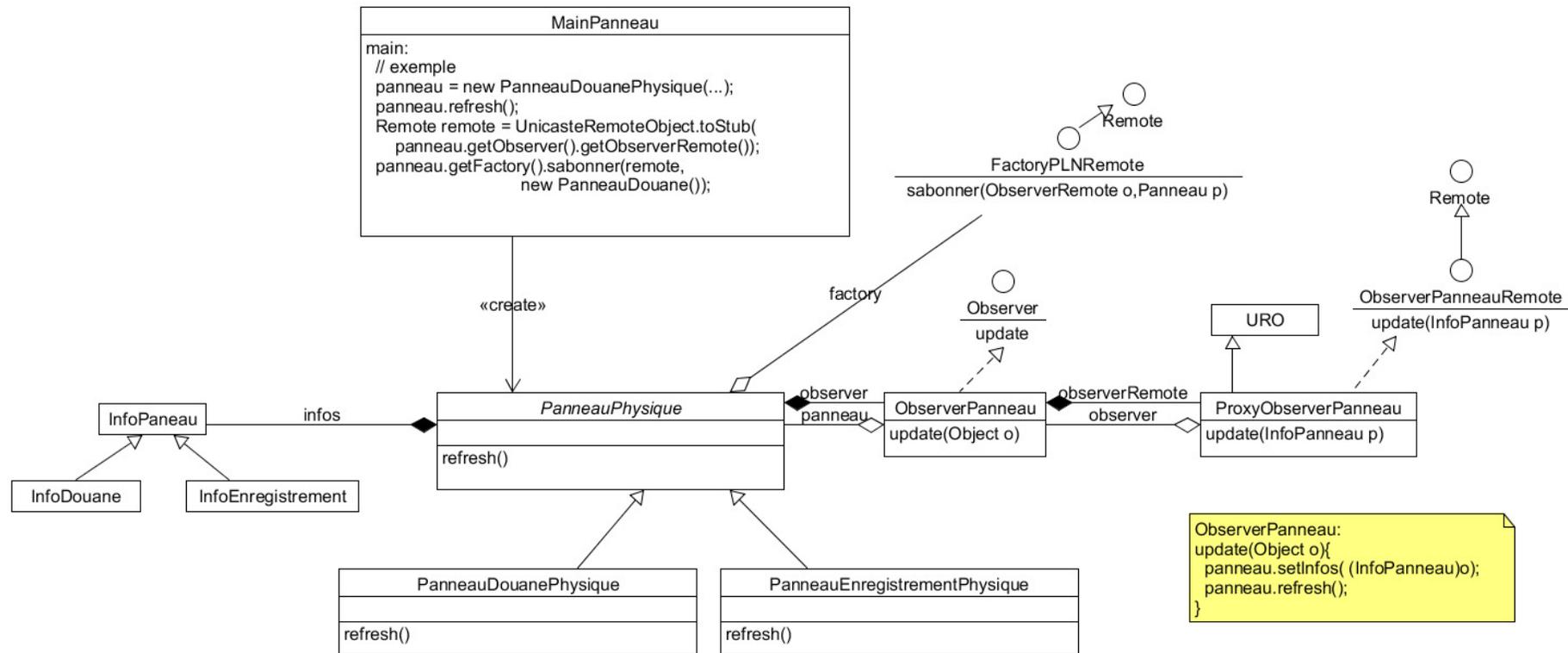
/5 points

Voir les codes en jaune dans le schéma précédent.

**(Suite du sujet page suivante)**

3/ Faites le diagramme de classe UML du [COMPOSANT 2] en mettant en évidence les Designs Patterns vus en cours. Commentez chacun de ces diagrammes.

/10 points



Un Panneau physiquement est une instance d'une classe qui hérite de PanneauPhysique.

Ce panneau utilise l'interface distante du factory distant pour faire son abonnement.

Chaque PanneauPhysique gère un observer local (ObserverPanneau) qui à sont tour gère un observer distant (ProxyObserverPanneau). C'est ce dernier qui reçoit les notifications provenant du serveur.

En fonction de sont type, le serveur notifie les informations spécifiques au type de panneau (InfoDouane, InfoEnregistrement, ... qui héritent de InfoPanneau.

Le traitement de notification est de modifier les informations du panneau avec celles reçues, puis de faire le rafraichissement de l'affichage du panneau.

Conseils :

Un composant logiciel [COMPOSANT X] correspond à une JVM ou process. Cela signifie que les COMPOSANTS X communiquent sur le réseau à travers des interfaces distantes.

Ainsi, pour une description précise de vos diagrammes de classe, on fait le choix que toutes les communications distantes entre les composants sont réalisées en RMI (utilisation de la classe URO = UnicastRemoteObject et de l'interface Remote).

**Fin du sujet**