Modélisation UML - Étude de cas (1/2)

Le consortium de compagnies aériennes Blue Sky veut un logiciel pour :

- gérer les vols et les avions des compagnies

- gérer les achats de billets et l'enregistrement des passagers. Ces opérations peuvent être faites en ligne par les clients, ou à un guichet tenu par une hôtesse. L'enregistrement peut aussi être fait par le client à un guichet automatique dans un aéroport.

- générer des tableaux de données pour les responsables clientèle des compagnies et pour les services de sécurité, en particulier la liste des passagers d'un vol.

Modélisation UML - Étude de cas (2/2)

Les précisions suivantes sont données par Blue Sky :

- un vol est ouvert et fermé à l'achat de billet et à l'enregistrement sur ordre de la compagnie.

- un client peut acheter un ou plusieurs billets, pour des passagers différents mais il doit fournir un numéro de passeport pour chaque billet.

- un billet concerne un seul passager.

- l'enregistrement peut donner lieu au paiement d'un supplément bagage.

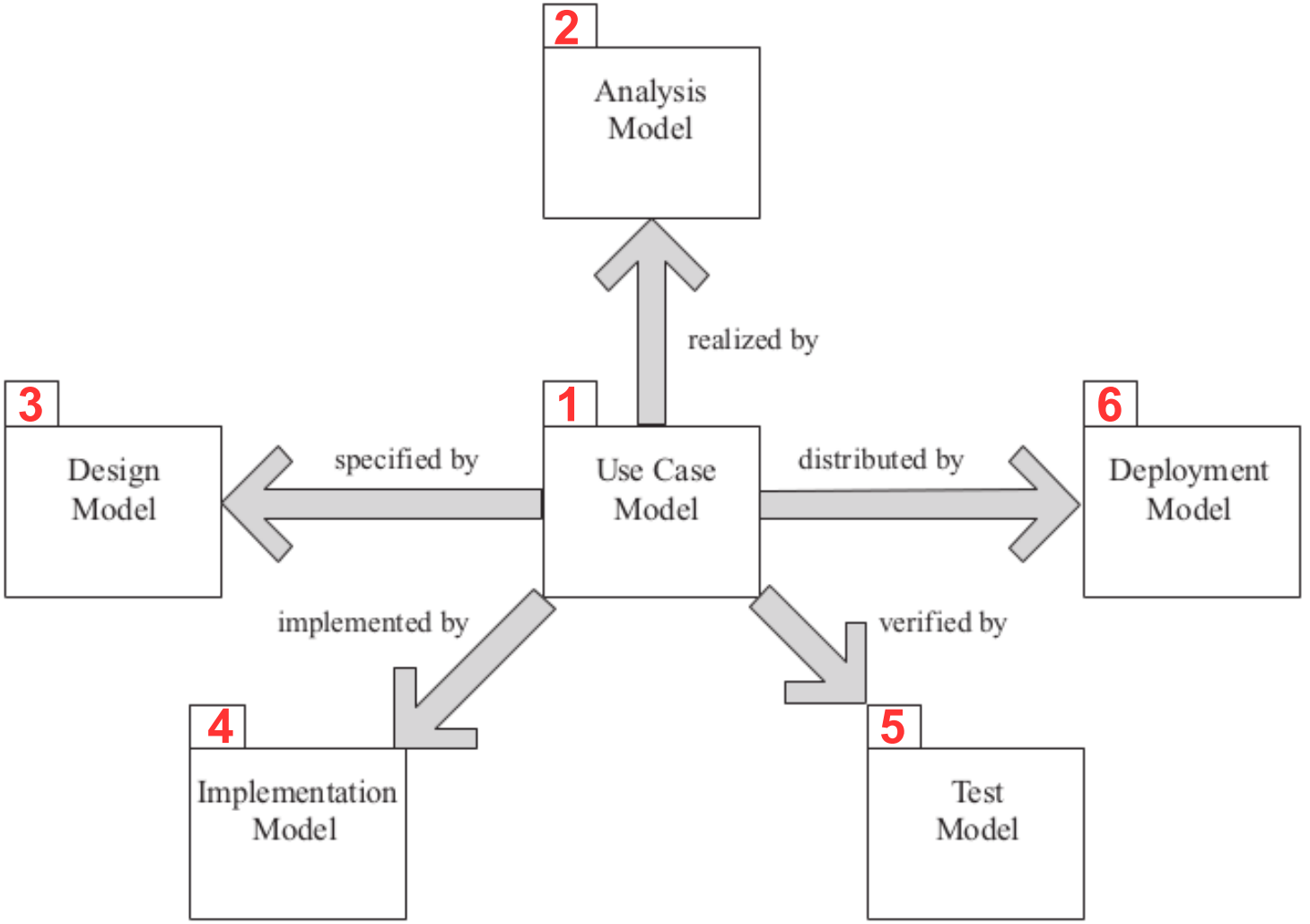
- un billet peut être annulé tant que le vol n'a pas eu lieu.

- un vol a un aéroport de départ et un aéroport d'arrivée ainsi qu'un jour et une heure de départ et d'arrivée.

- un billet peut comporter plusieurs vols avec des escales.

Remarque : on néglige l'identification des acteurs dans le système

Les modèles dans UP



Inception

Dans UP, l'inception, ou analyse des besoins, couvre :

- la définition du périmètre du projet
- la spécification des fonctionnalités
- le recensement des risques et des coûts
- la réalisation d'un premier jet d'architecture.

Les fonctionnalités sont spécifiées dans un modèle de cas d'utilisation :

- *diagramme de cas d'utilisation* pour spécifier les acteurs et les fonctionnalités

- *diagrammes de comportement* si nécessaire pour détailler les fonctionnalités

Modéle de cas d'utilisation (1/7)

Les acteurs doivent être bien identifiés :

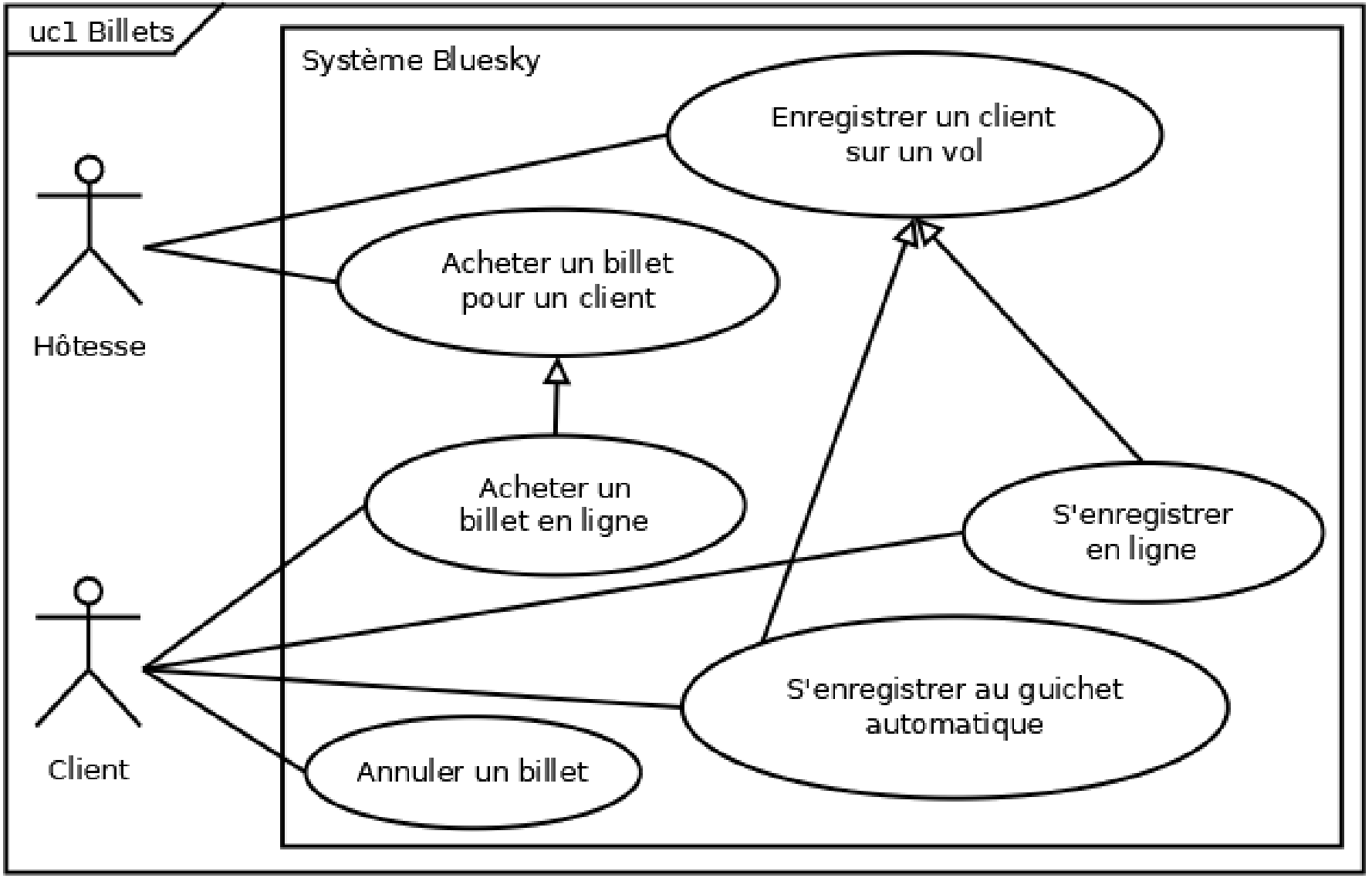
- les clients
- les employés des compagnies
- les services de sécurité

Les employés des compagnies peuvent avoir des rôles différents au regard des fonctionnalités :

- responsables des vols dans les compagnies
- responsables clientèle dans les compagnies
- hôtesses aux guichets

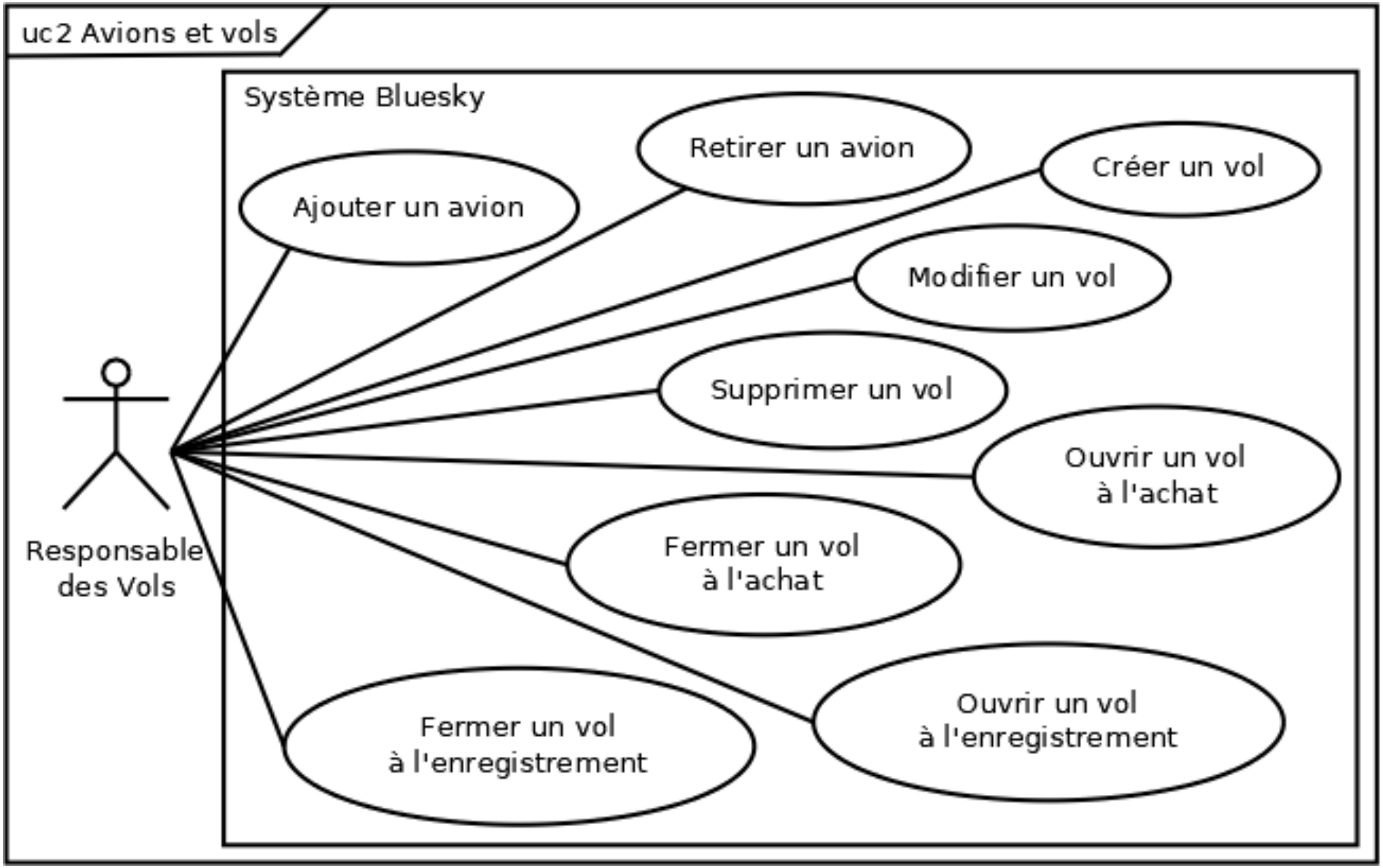
Modèle des cas d'utilisation (2/7)

Certaines informations ne sont pas fournies (par exemple on ne sait pas qui annule les billets), il faut faire un choix.



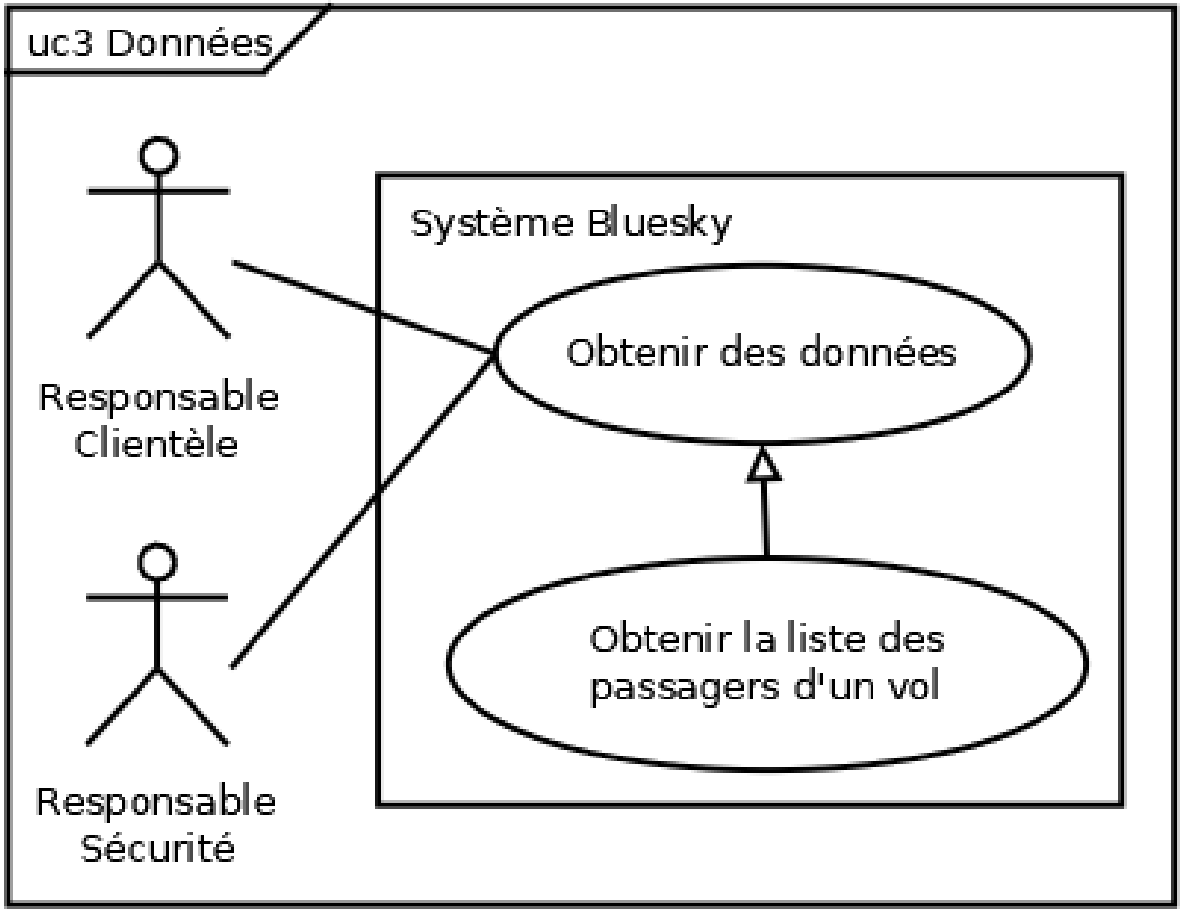
Modèle des cas d'utilisation (3/7)

Certaines fonctionnalités sont floues (« gérer les avions »), elles doivent être précisées.



Modèle des cas d'utilisation (4/7)

Il peut être préférable distinguer des acteurs qui ont des rôles identiques, pour faciliter l'évolution du modèle et surtout du logiciel.



Modèle des cas d'utilisation (5/7)

Détailler le déroulement des cas d'utilisation va permettre de mettre en évidence des éléments structurels :

- classes d'objets
- données caractérisant des objets
- méthodes supportées par des objets
- acteurs extérieurs au système

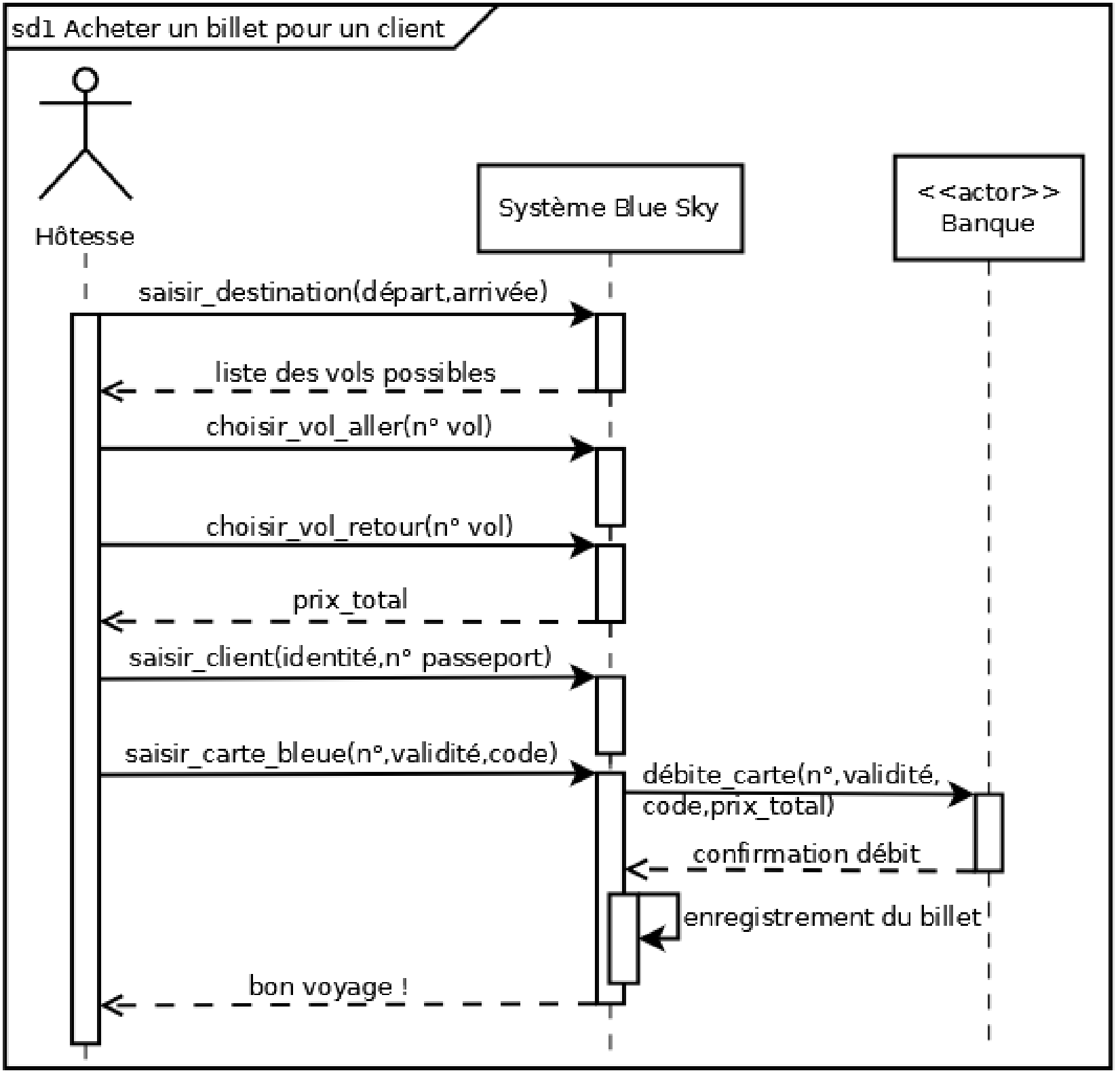
Détailler un cas n'a de sens que si le déroulement du cas n'est pas évident ou si cela apporte des informations utiles pour la suite.

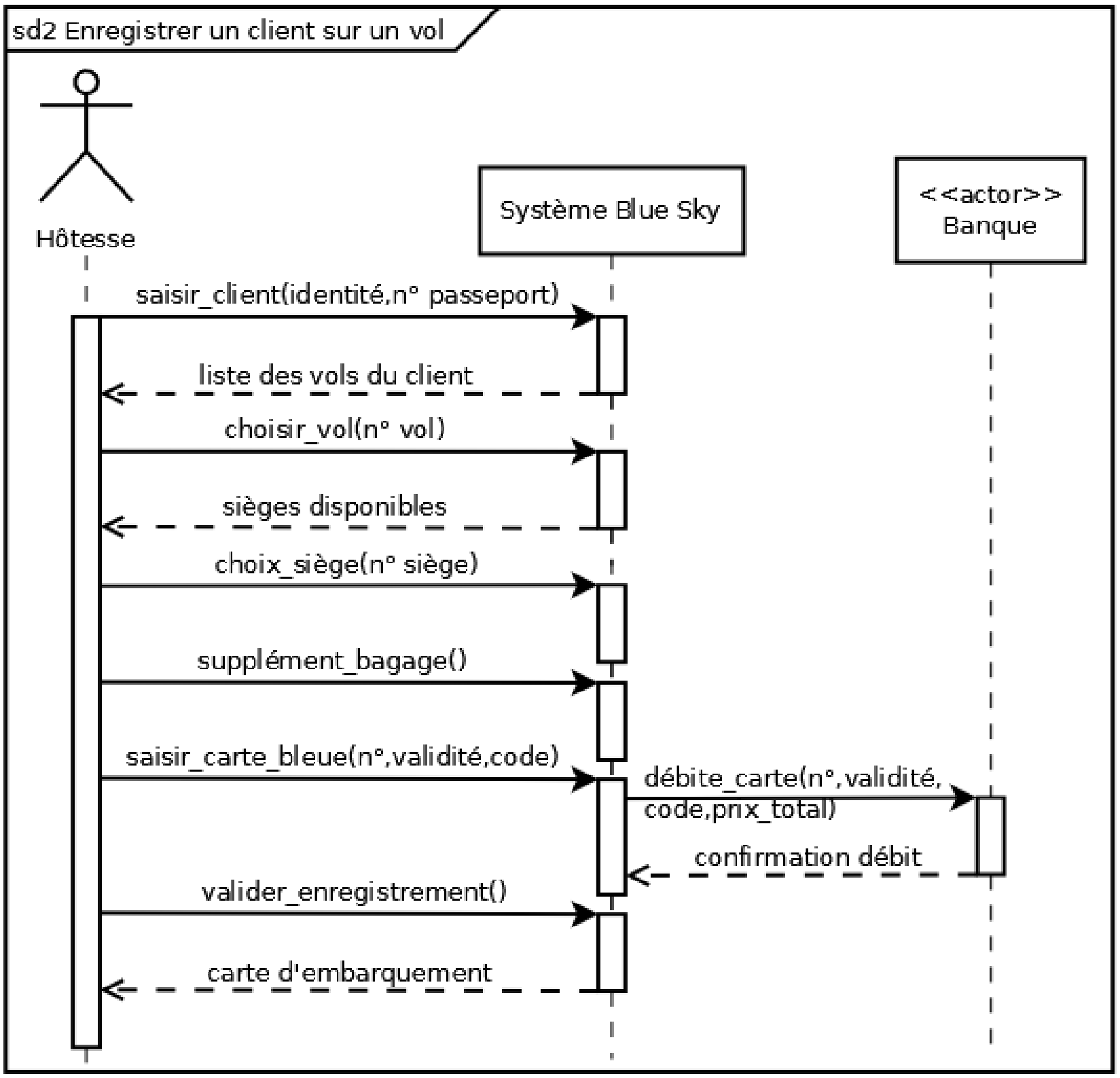
Exemple de l'achat de billet :

- on choisit de séparer le choix du vol aller et du vol retour, il faut le préciser dans le modèle de cas d'utilisation.

- l'enregistrement peut donner lieu à achat d'un supplément de bagage, on l'indique dans un diagramme de séquence.

Il faudrait aussi détailler la création d'un vol et l'ajout d'un avion.

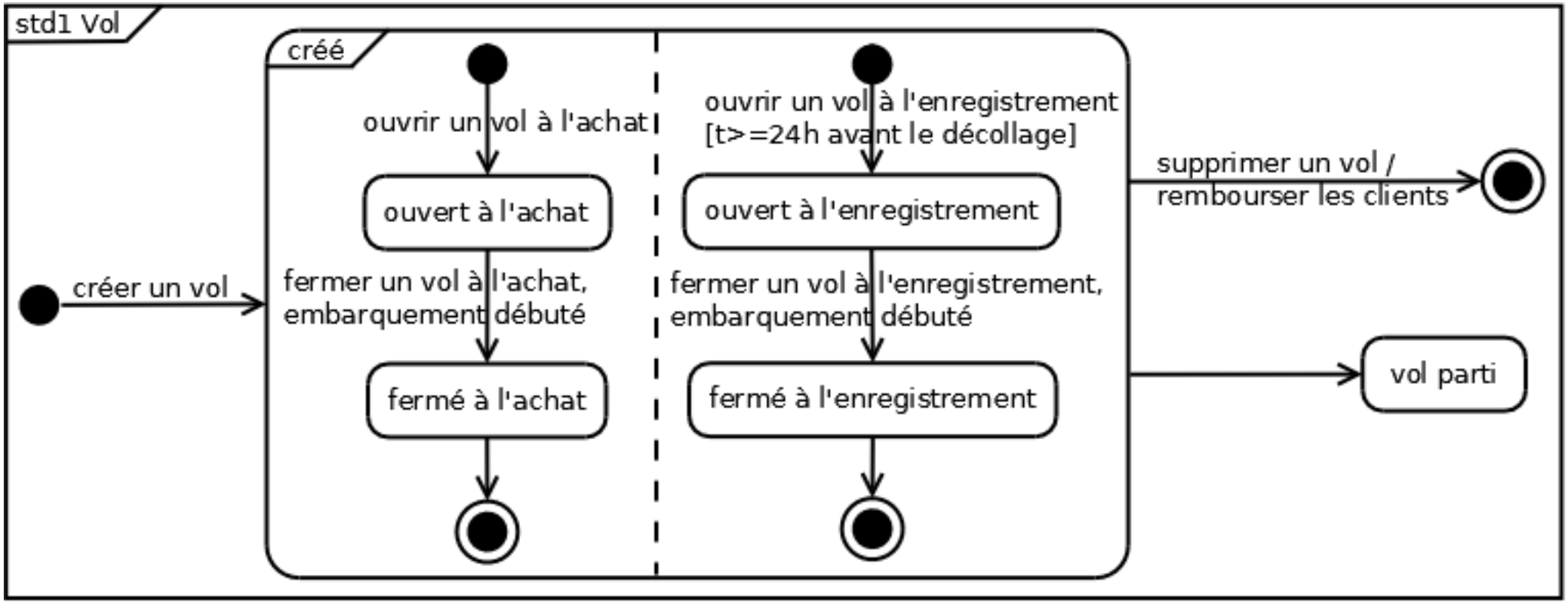

sd1 Acheter un billet pour un client
Hôtesse
Système Blue Sky
<<actor>>
Banque
saisir_destination(départ,arrivée)
liste des vols possibles
choisir_vol_aller(n° vol)
choisir_vol_retour(n° vol)
prix_total
saisir_client(identité,n° passeport)
saisir_carte_bleue(n°,validité,code)
débite_carte(n°,validité,
code,prix_total)
confirmation débit
enregistrement du billet
bon voyage !


sd2 Enregistrer un client sur un vol
Hôtesse
Système Blue Sky
<<actor>>
Banque
saisir_client(identité,n° passeport)
liste des vols du client
choisir_vol(n° vol)
sièges disponibles
choix_siège(n° siège)
supplément_bagage()
saisir_carte_bleue(n°,validité,code)
débite_carte(n°,validité,
code,prix_total)
confirmation débit
valider_enregistrement()
carte d'embarquement

Modèle des cas d'utilisation (6/7)

Il paraît nécessaire de préciser la façon dont les vols sont ouverts et fermés à l'achat et à l'enregistrement.

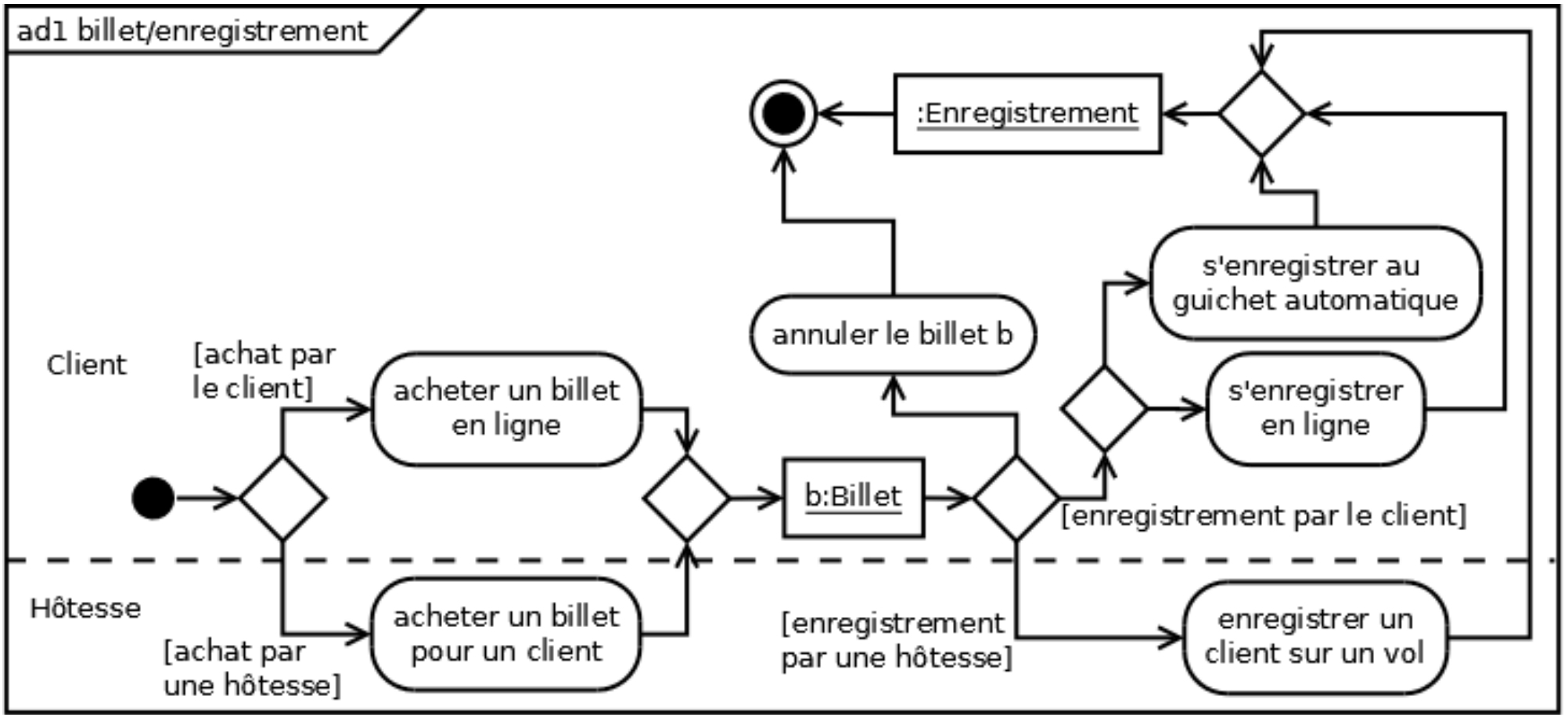
<u>Remarque importante</u> : on reprend autant que possible les éléments des diagrammes précédents !



Modèle des cas d'utilisation (7/7)

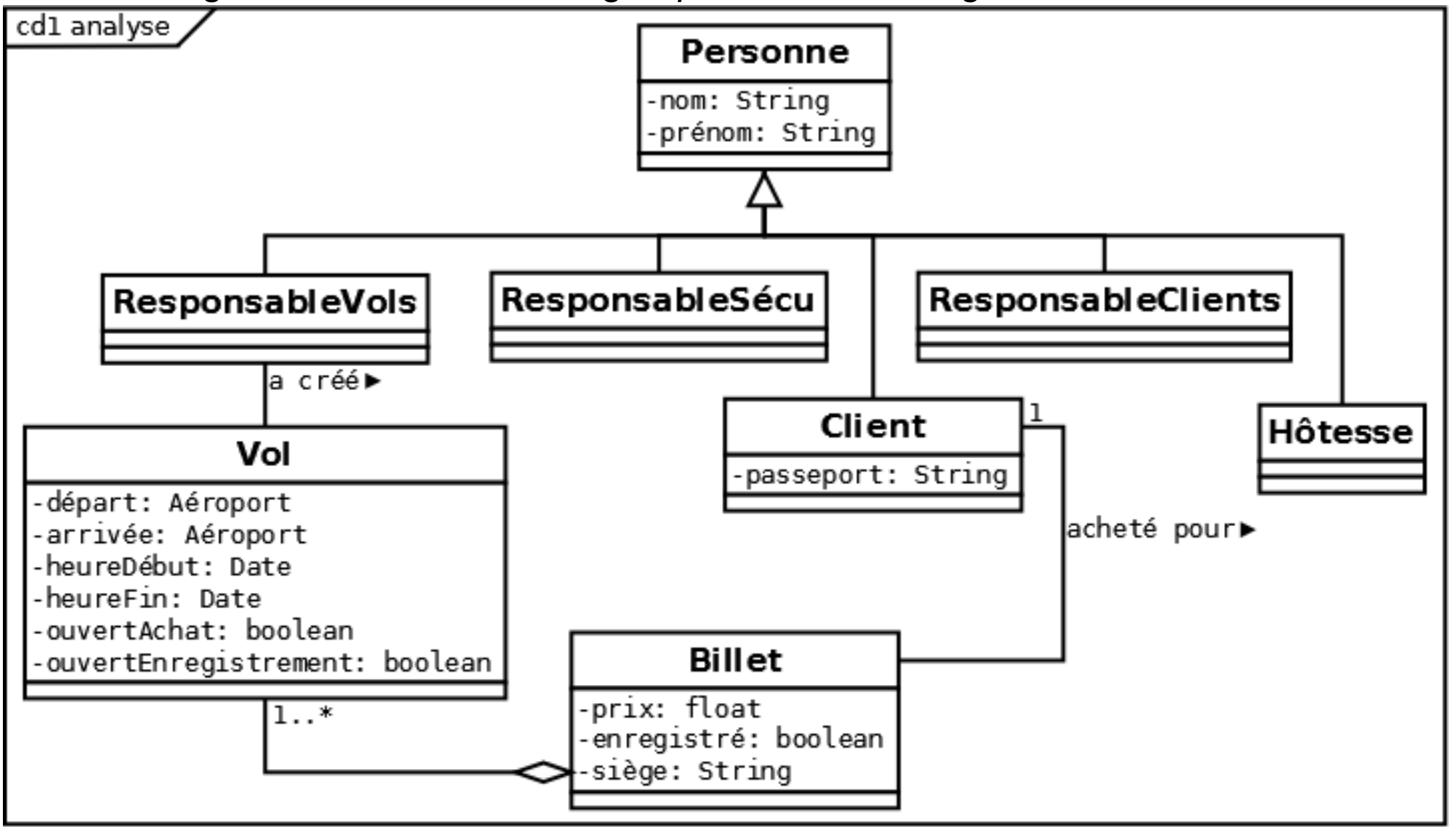
Il est utile de préciser la façon dont les cas d'utilisation vont s'enchaîner, ou pas, à l'aide d'un diagramme d'activités.

Exemple : on ne peut plus annuler un billet après l'enregistrement



Modèle d'analyse (1/6)

Des choix structurels peuvent être faits à partir des cas d'utilisation : *on n'enregistre pas qui a ouvert ou fermé un vol à l'achat ou à l'enregistrement, on ne distingue pas billet et enregistrement, ...*



Elaboration

Dans UP, l'élaboration couvre la spécification et le début de la conception de l'architecture du système et de son fonctionnement.

L'architecture est élaborée dans un modèle d'analyse :

- *diagramme de classes*

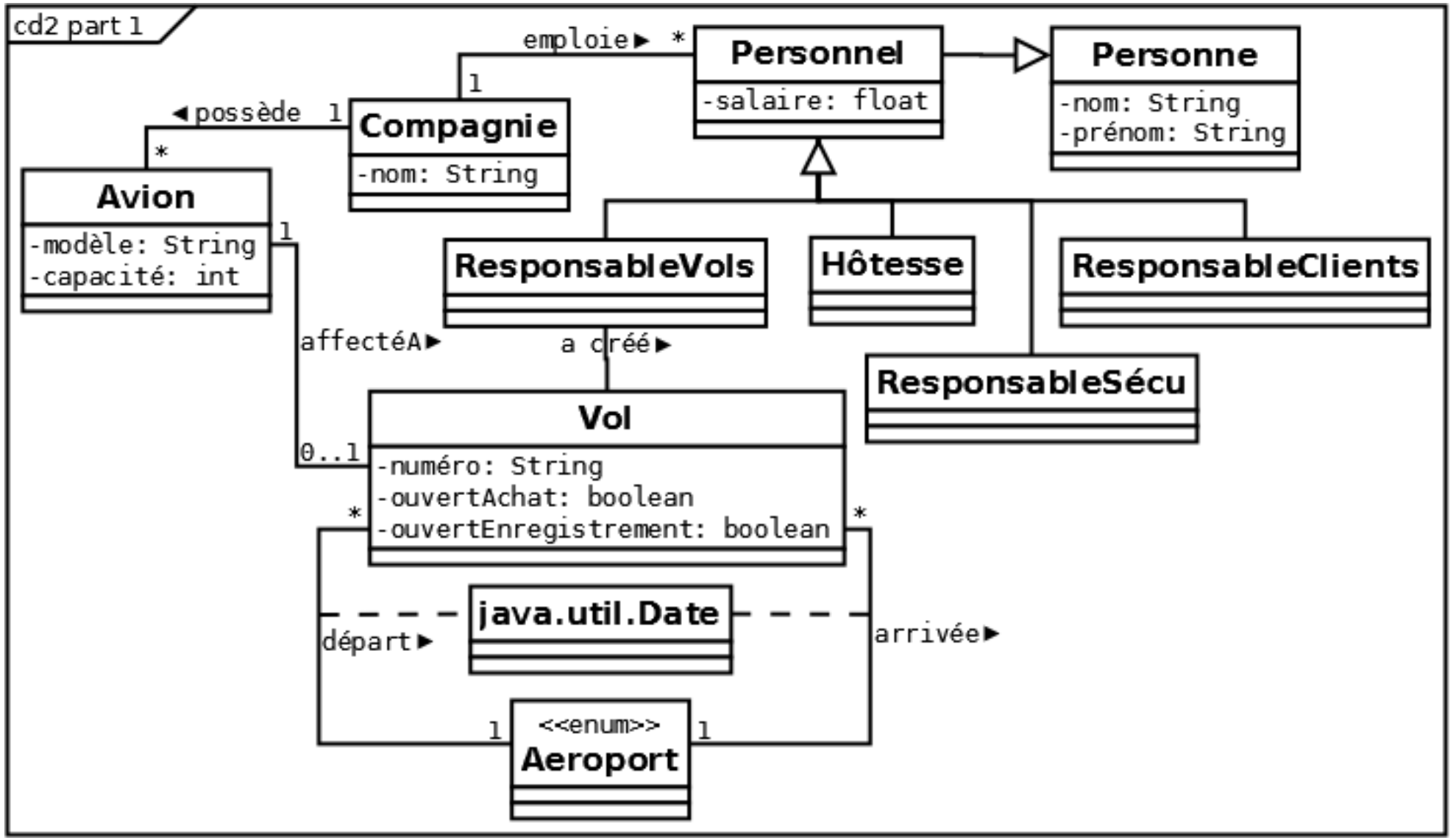
- *diagramme d'objets* si nécessaire pour détailler les états possibles des objets

- *diagrammes de comportement* si nécessaire pour détailler le fonctionnement du système

- *diagramme de paquetages*

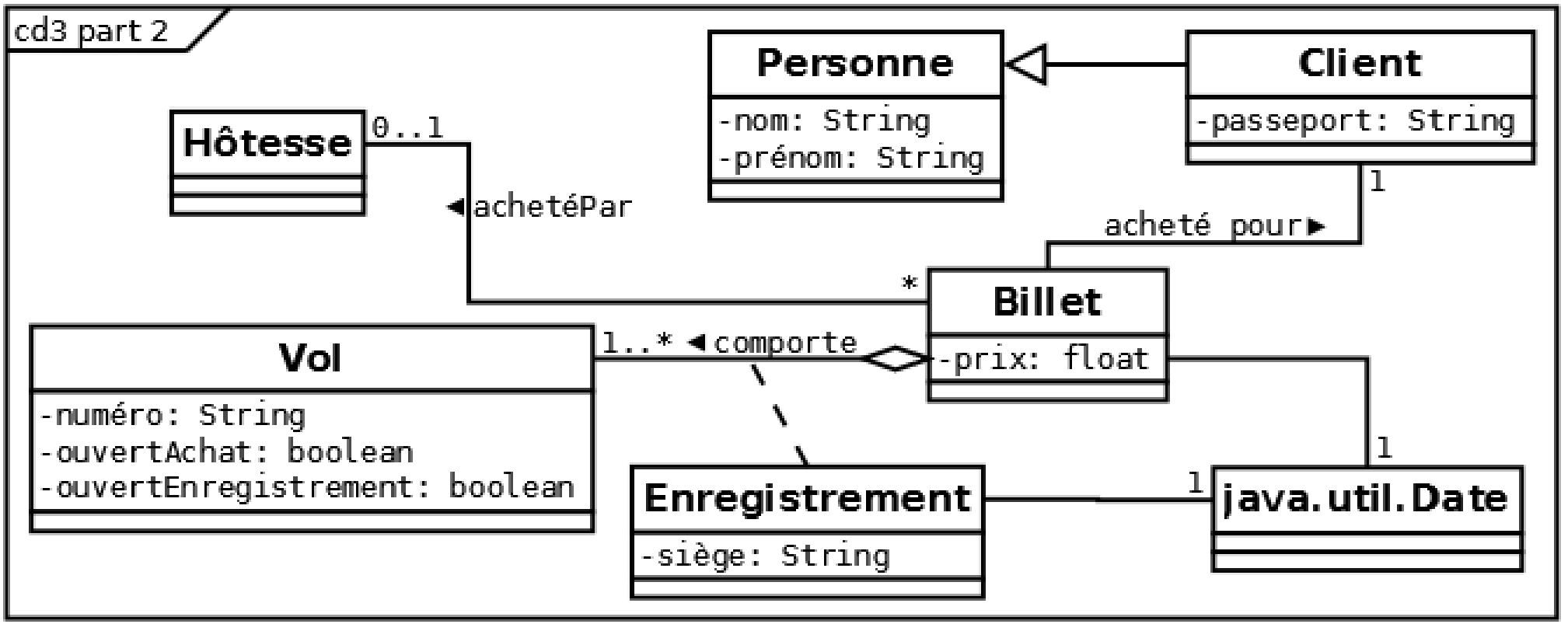
Modèle d'analyse (2/6)

On rend plus modulaire la description des vols, on introduit les avions et les compagnies.



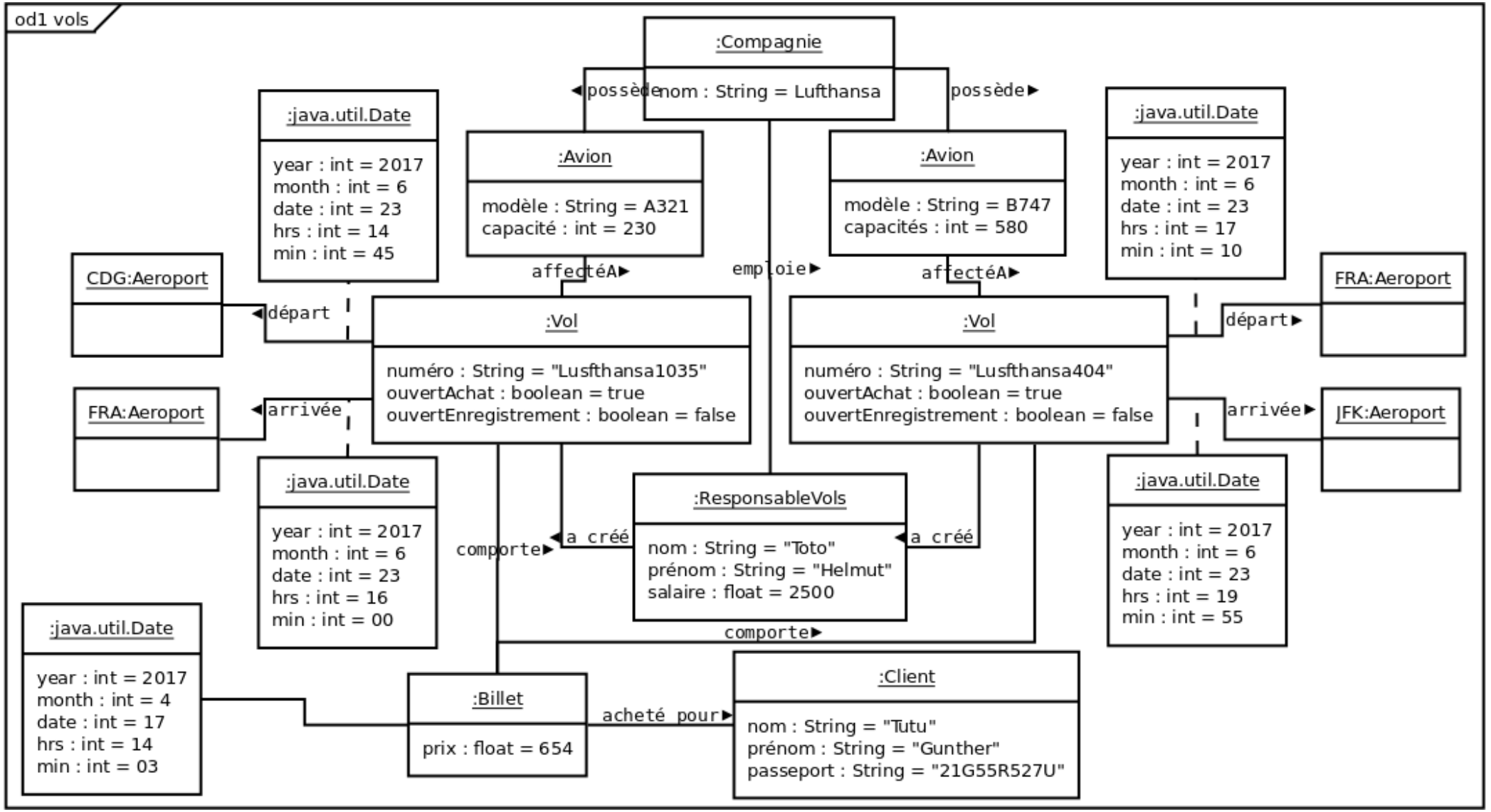
Modèle d'analyse (3/6)

On s'aperçoit que les enregistrements portent sur les vols et pas sur le billet en entier. On ajoute les dates d'achat et d'enregistrement.



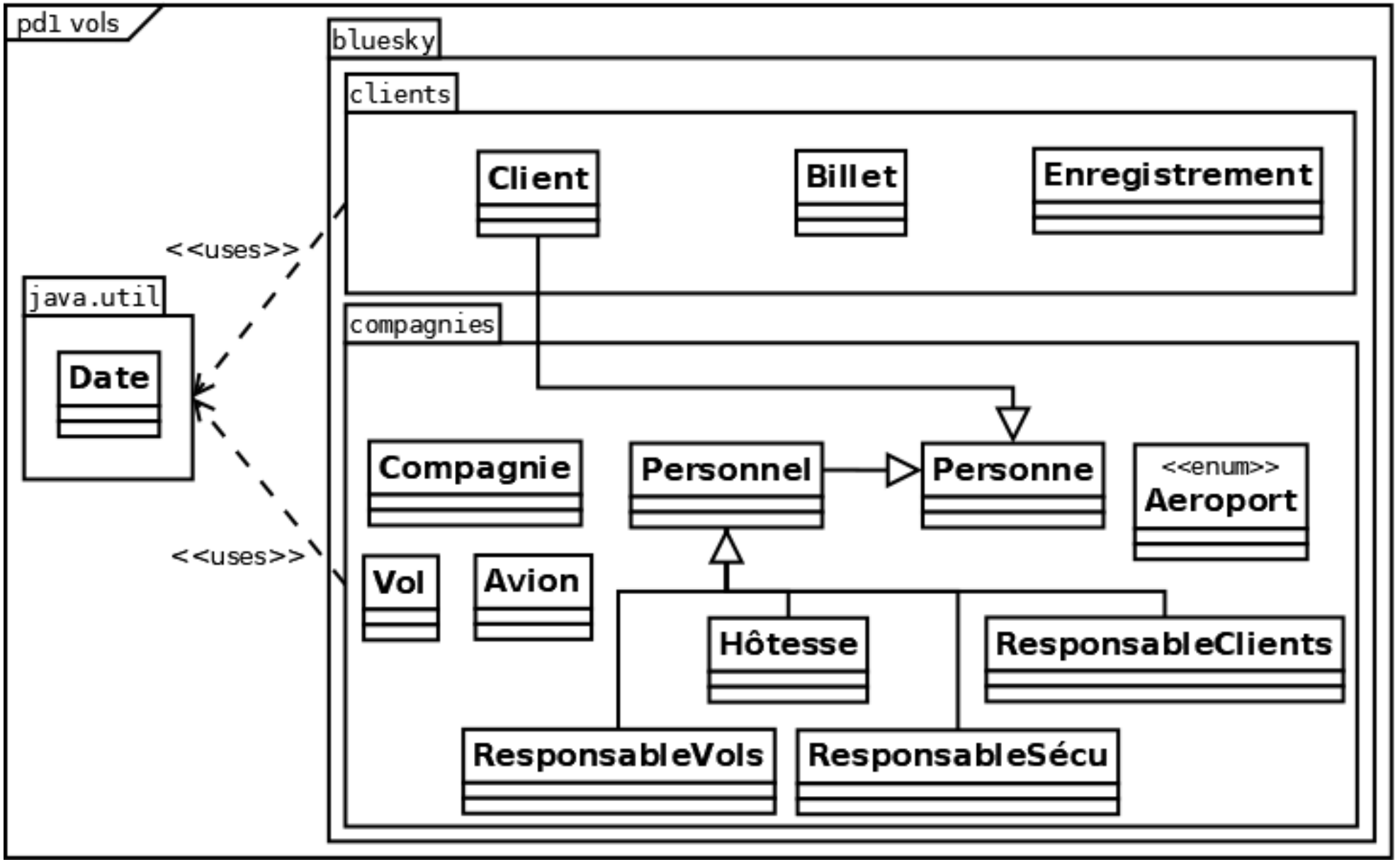
Modèle d'analyse (4/6)

Un diagramme d'objets peut être utile pour vérifier la cohérence du diagramme de classe ou pour l'exemplifier.



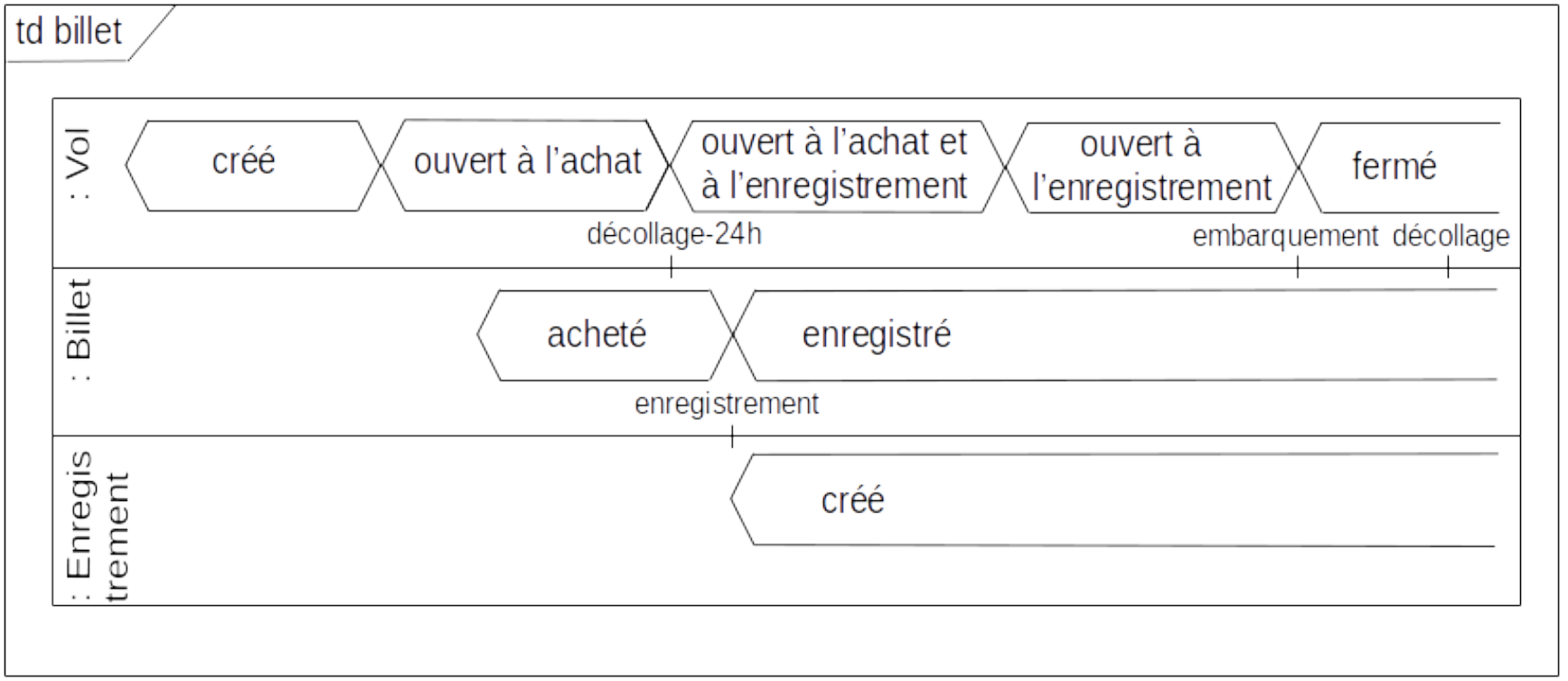
Modèle d'analyse (5/6)

Un diagramme de paquetages permet de spécifier la structuration du programme en fichiers et répertoires.



Modèle d'analyse (6/6)

Un diagramme de timing peut être utile pour visualiser le déroulement des actions et des changements d'état.



Construction

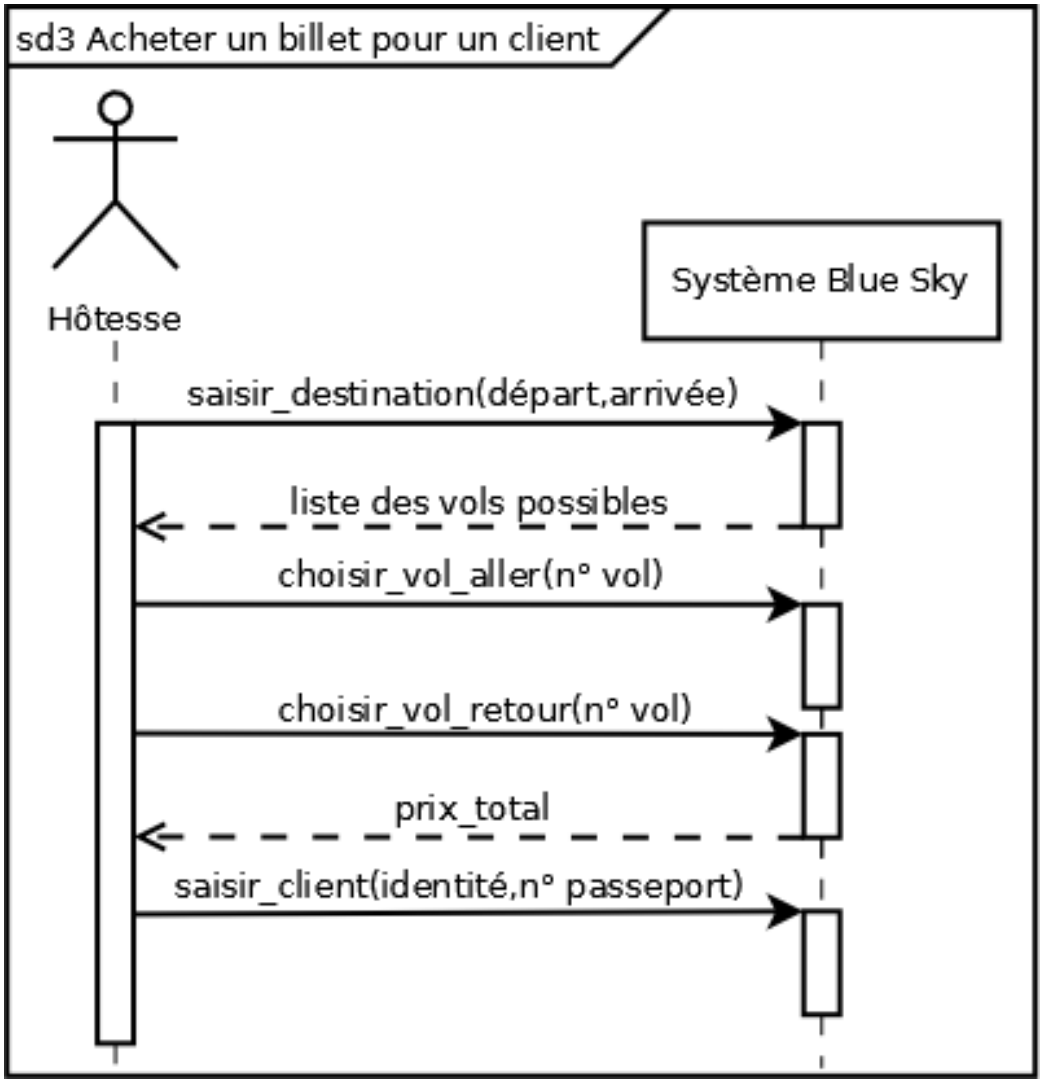
Dans UP, la construction correspond à une synthèse qui fixe le modèle d'analyse dans un modèle de conception.

Le modèle de conception contient les mêmes diagrammes que le modèle d'analyse, mais détaillés au maximum pour pouvoir implémenter « sans se poser de question » (voire de façon automatique).

Les modalités d'implémentation et de test vont être spécifiés à ce moment dans des modèles d'implémentation et de test.

Modèle de conception (1/5)

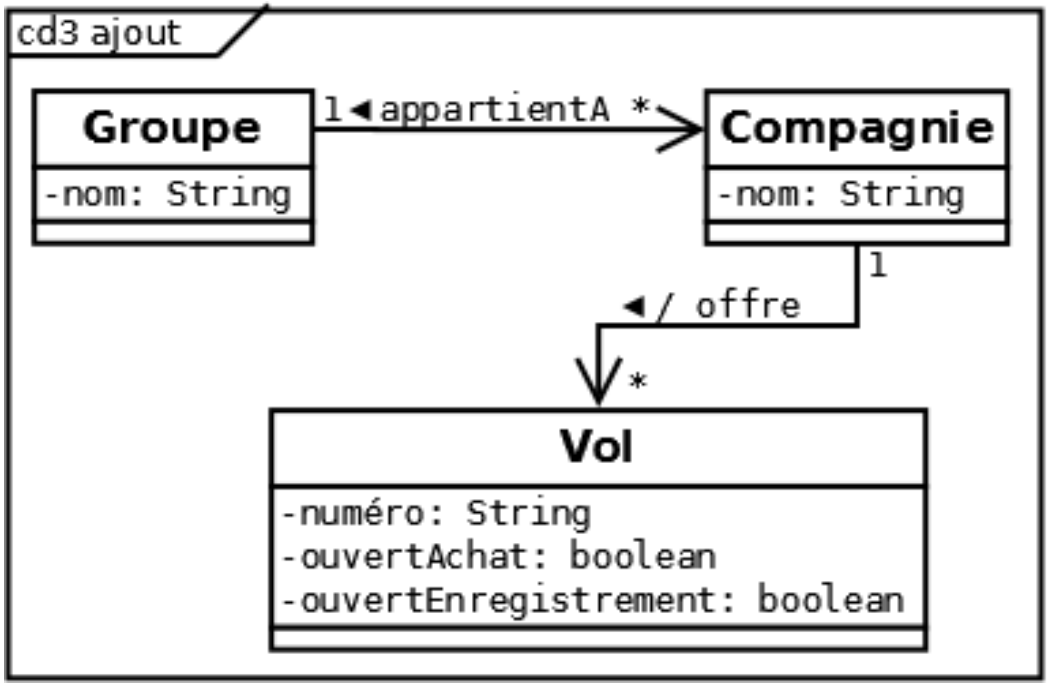
On peut détailler les diagrammes de séquence du modèle de cas d'utilisation pour vérifier la cohérence avec le fonctionnement interne du système.



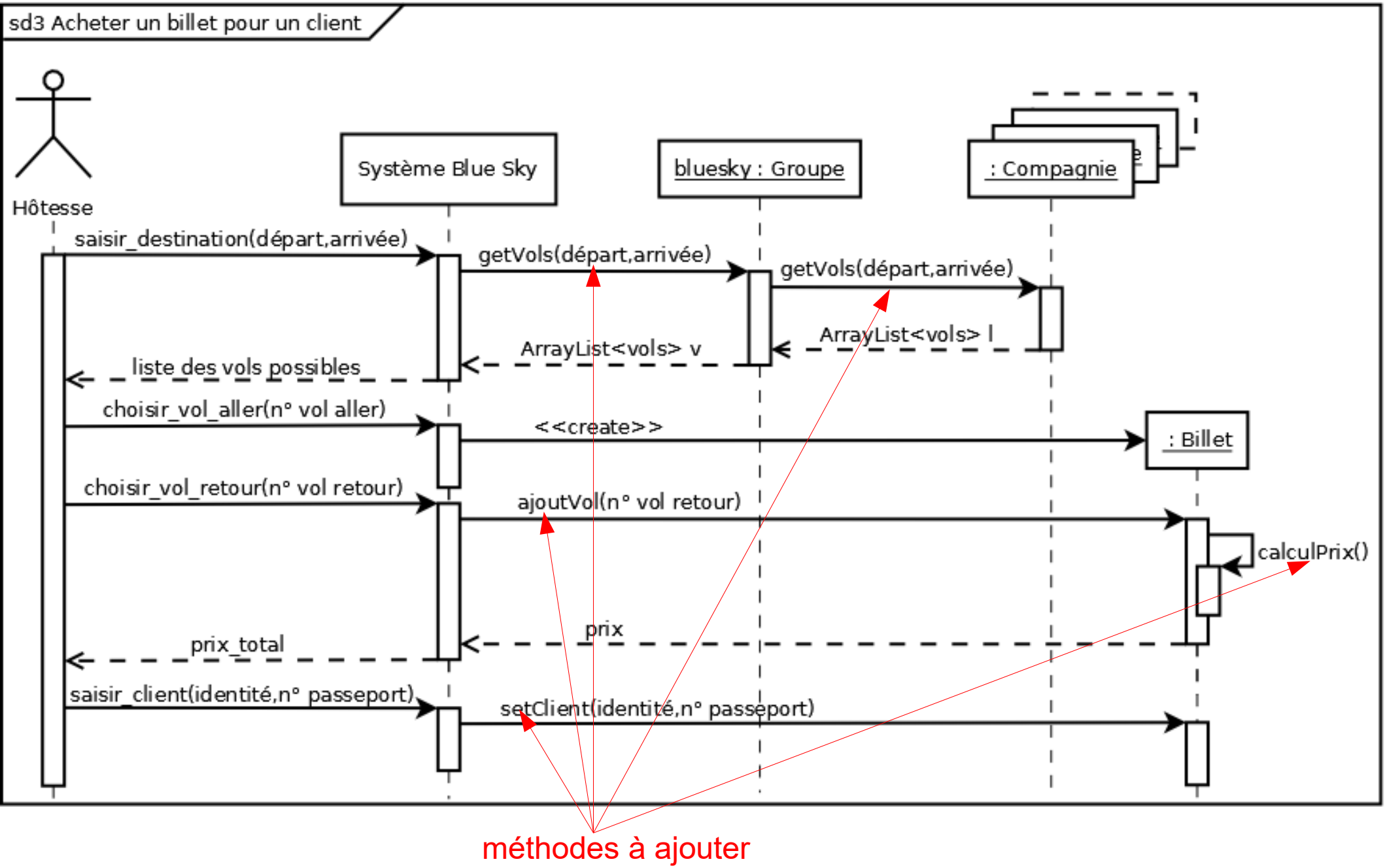
Modèle de conception (2/5)

Problème : les vol ne sont référencés (potentiellement) que par les **`ResponsableVols`** et les **`Avions`** !

Deux solutions, parmi d'autres :

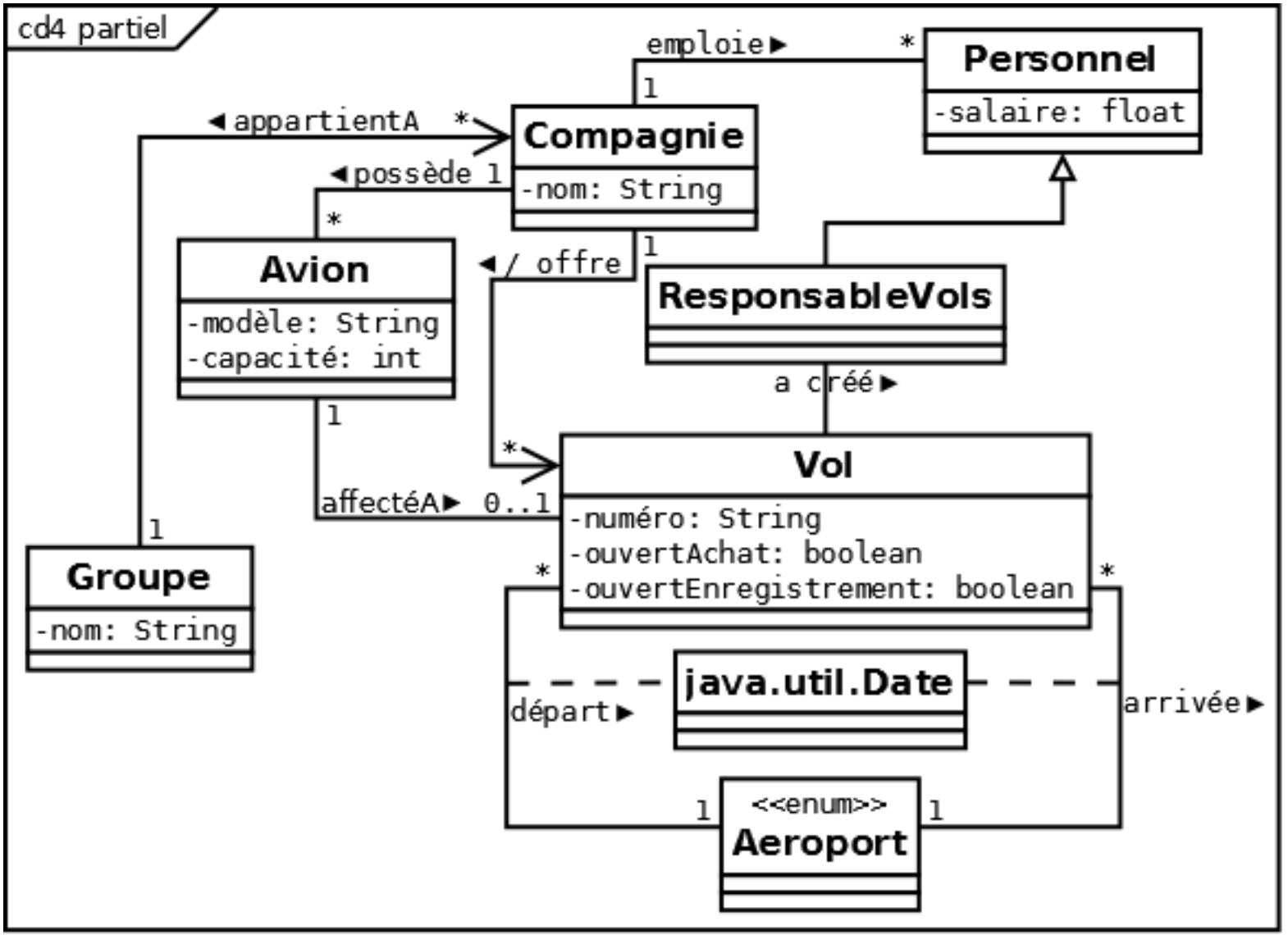


Modèle de conception (3/5)



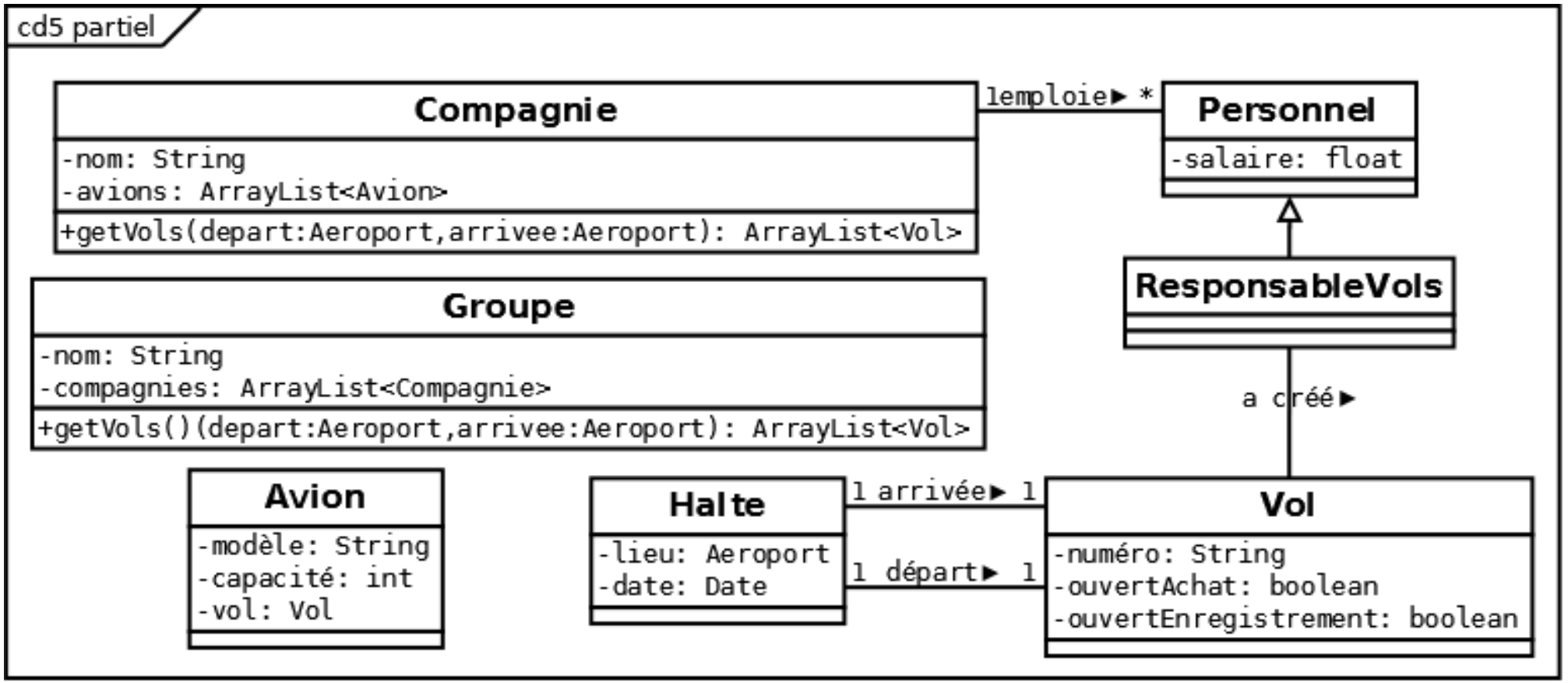
Modèle de conception (4/5)

Le diagramme de classes peut être affiné : détail des attributs et méthodes, détails sur l'implémentation des associations, etc.



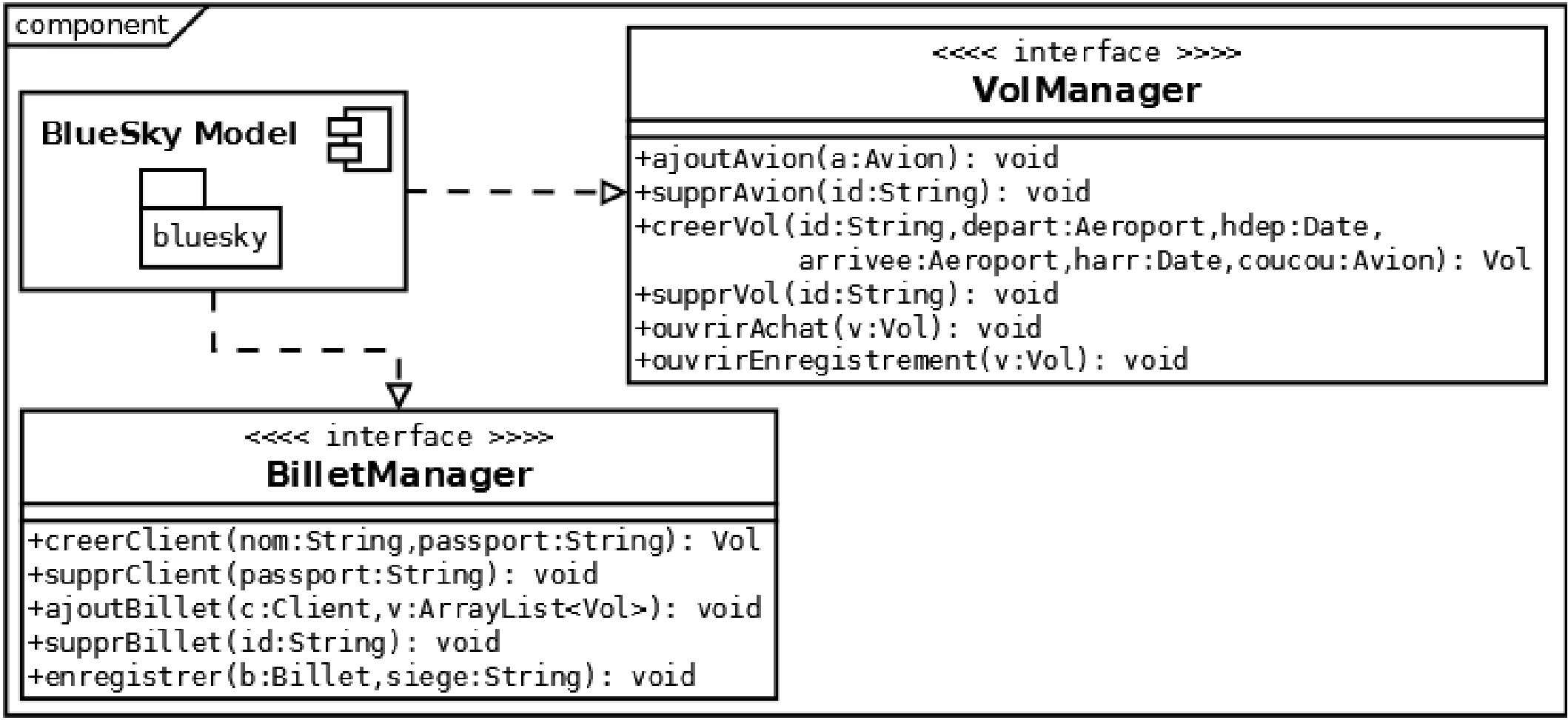
Modèle de conception (5/5)

NB : certains avions peuvent ne pas être affectés à un vol



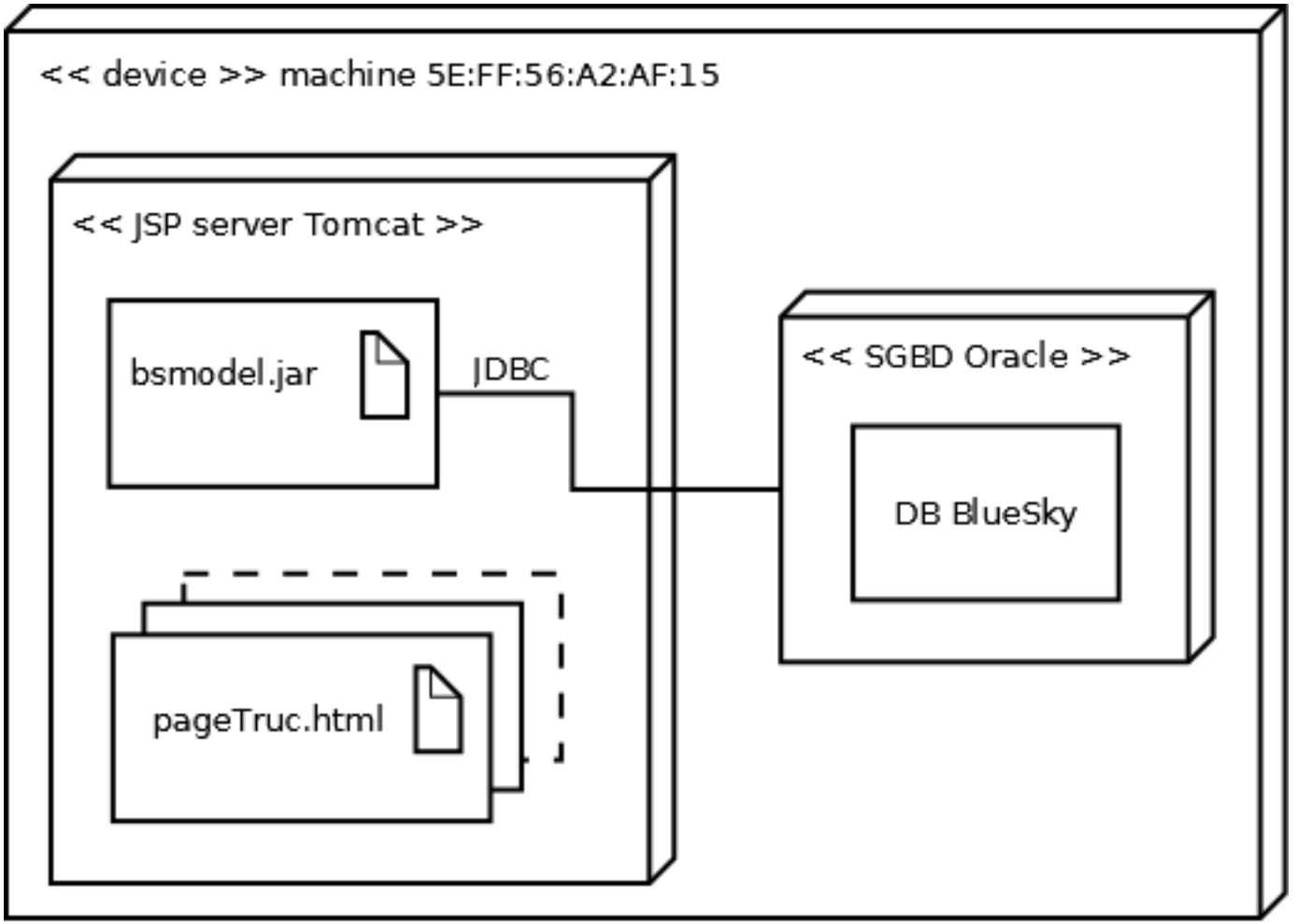
Modèle d'implémentation (1/2)

On peut imaginer que cette partie modèle du logiciel soit destinée à être connectée, comme composant, à une interface graphique. Un diagramme de composant permet de spécifier les interfaces de connection.



Modèle d'implémentation (2/2)

Le système doit être déployé en ligne sous forme de servlet JSP déployée sur un serveur Tomcat. Les données sont sauvegardées sur une base Oracle installée sur la même machine.



Modèle de tests

Les diagrammes de séquence décrivant des scénarios d'utilisation peuvent être utilisés pour des tests unitaires.

Tout diagramme de comportement peut potentiellement servir à produire un test unitaire.

Les diagrammes d'objets peuvent servir à effectuer des tests structurels et comportementaux.