

H4202

PLANCHE Benjamin – PHAN Nicolas

BARDY Fabien – DRIAI Sahra

BAUDRY Daniel – FOULQUIER Manon – GEVREY Loïc



TP Développement Orienté Objet

Bagage Interface Manager Prototype du SGBag

Compte-Rendu



Table des matières

Introduction	3
Etude Préliminaire.....	4
Introduction	4
Organisation du projet.....	4
Répartition des rôles.....	4
Gestion du projet	5
Méthodologie.....	5
Outils de gestion	6
Livrables	8
Liste des livrables	8
Tableau récapitulatif des documents et des acronymes correspondants.....	8
Diagramme de précéderance des livrables.....	9
Planning prévisionnel.....	10
Périmètre du projet SGBag-Partiel	11
Liste de besoins potentiels.....	11
Modèle du domaine de SGBag	12
Conclusion.....	14
Elaboration.....	15
Introduction	15
Capture des besoins.....	15
Modèle de cas d'utilisations	15
Diagrammes de séquences	21
Analyse.....	22
Diagramme de paquetages d'analyse	22
Diagramme séquence pour le scénario d'arrivée des bagages	23
Diagramme de séquence pour le scénario d'arrivée de top d'horloge	24
Conclusion.....	37
Construction.....	38
Introduction	38
Analyse	38
Conception XML.....	38
Conception.....	40



Diagramme de séquence	40
Diagramme de classes Vues + Noyau (classes vides mais avec liaison).....	50
Rétro-engineering du code vers le diagramme complet en fin de projet	51
Réalisation.....	52
Présentation de l'IHM	52
Transition	57
Introduction	57
Charge de travail	57
Répartition des activités	57
Bilan	57
Commentaires.....	57
Retour sur le programme généré	58
Retour sur la gestion.....	58
Suggestions	59
Conclusion.....	59
Conclusion.....	60



Introduction

Désirant innover son système de gestion des bagages de l'aéroport Saint Exaspéré, la Chambre des Commerces de l'INSA a lancé un ambitieux chantier d'automatisation. Nommé SysBag, ce nouveau système se veut à la fois plus fiable, moins contraignant, et plus simple à prendre en main pour les différents acteurs.

Suite à un appel d'offres pour l'élaboration, dans un premier temps, du module de simulation du transport intra-terminal des bagages, notre équipe d'ingénieurs-spécialistes s'est portée volontaire face au challenge. Après l'élaboration d'une première maquette, notre projet a donc été retenu, initiant une collaboration étroite avec la maîtrise d'ouvrage.

Pleinement conscients de la complexité de ce projet d'innovation, nous avons engagé tout notre savoir-faire et notre expérience dans la gestion de projet, à l'élaboration d'un premier prototype. Par la mise en place de la méthodologie itérative et incrémentale USDP, associée à certains concepts Agile, nous avons pu prendre en compte pleinement les besoins clients, et instaurer un processus de développement interactif et parallélisé.

Nous espérons que cet ensemble de rapports, reprenant les principales activités USDP, ainsi que notre prototype baptisé *Bagage Interface Manager (BIM)*, sauront vous convaincre de notre implication et expertise, afin de poursuivre ensemble l'aventure SysBag.

En vous souhaitant bonne lecture, et vous remerciant pour la confiance que vous nous avez offerte.



Etude Préliminaire

Introduction

Phase d'initialisation du cycle USDP, l'étude préliminaire permet d'évaluer le domaine du projet et ses risques. Elle sera également pour nous l'occasion de définir clairement l'ensemble des paramètres (rôles de chacun, méthodologie, etc.) influant sur la gestion du projet.

Nous éclaircrons ainsi dans un premier temps ces choix méthodologiques puis définirons l'ensemble des livrables, *mile-stones* du projet, et poursuivrons avec l'étude du système.

Organisation du projet

Répartition des rôles

Après une brève étude du projet, une réunion d'équipe a été organisée pour discuter du contenu du projet, des compétences pertinentes de chacun, et de ses aspirations. L'élection d'un chef de projet a suivi, ainsi que d'un responsable Qualité et d'un responsable technique :

- **Chef de projet** : Benjamin Planche
 - *Tâches principales* :
 - Définir le calendrier et la distribution des tâches.
 - Coordonner les différentes équipes (réunions, Redmine, mails, ...).
 - Suivre l'avancement de chaque tâche pour garder un point de vue d'ensemble.
- **Responsable Qualité** : Manon Foulquier
 - *Tâches principales* :
 - Référencer et normaliser les notions et termes manipulés.
 - Vérifier la cohérence d'ensemble.
 - Mettre en place une démarche Qualité (s'assurer de la maintenabilité, de la clarté, etc.).
- **Responsable Technique** : Nicolas Phan
 - *Tâches principales* :
 - Former les autres, si besoin, aux technologies employées (Swing, SVN, Redmine).
 - Prendre en main et adapter les modules et exemples fournis.
 - Effectuer la maintenance du dépôt SVN et Redmine.

Durant la phase préliminaire, différents tâches ont été réalisées simultanément (étude du domaine, étude des cas d'utilisation, rédaction des scénarios associés, etc.).

Par la suite, un découpage précis en rôles a été élaboré, parallélisant les tâches indépendantes. Ce découpage, mis en place pour la phase d'élaboration fut appliqué également pour la conception. Il fut défini selon le modèle suivant :

- **Moteur dynamique et gestion des collisions**
 - Responsable : Sahra Driai, aidée de Fabien Bardy
 - Tâches :
 - Étudier le scénario du *Top Horloge*, couvrant toute l'évolution dynamique du chariot et ses cas limites (collisions, prise ou dépôt de bagage, ...)
 - Élaborer les diagrammes de séquences associés, en précisant niveau par niveau.



- Implémenter et tester unitairement ces procédures.

- **Persistance XML**

- Responsable : Daniel Aubry, aidé de Loïc Gevrey
- Tâches :
 - Étudier les associations entre les éléments du noyau afin d'extraire une stratégie de persistance.
 - Élaborer le diagramme de séquences associé.
 - Implémenter et tester unitairement ces procédures.

- **Fonctionnement manuel**

- Responsable : Manon Foulquier
- Tâches :
 - Étudier le scénario du *Clic en fonctionnement Manuel*, assurant à l'utilisateur un contrôle total sur les actions des chariots et la génération des bagages.
 - Élaborer les diagrammes de séquences associés, en précisant niveau par niveau.
 - Implémenter ces procédures.

- **Vues et Affichage**

- Responsable : Nicolas Phan
- Tâches :
 - Définir une stratégie pertinente d'affichage, n'altérant pas le noyau et son fonctionnement.
 - Élaborer le diagramme de séquences associé.
 - Implémenter et tester unitairement ces procédures.
 - Intégrer, avec l'aide de l'équipe concernée, le noyau (associations, récupération des changements d'état).
 - Fournir les images et éléments graphiques nécessaires.

- **Interface Homme-Machine**

- Responsable : Benjamin Planche
- Tâches :
 - Transcrire en Swing la maquette précédemment réalisée.
 - Intégrer, avec l'aide de l'équipe concernée, les vues à l'IHM (options d'affichage, double-buffering).
 - Intégrer, avec l'aide de l'équipe concernée, les interactions manuelles à l'IHM (traitement d'évènements).

Gestion du projet

Méthodologie

Ce projet étant de courte durée, il ne s'avère pas possible de pleinement tirer bénéfices d'une méthodologie de gestion de projet ou d'une démarche Qualité. Néanmoins, le respect, même relatif, de ces notions, permet d'obtenir une clarté structurelle, tant au niveau de l'organisation que de la conception, facilitant l'avancement.



Nous avons donc opté, sur la totalité du projet, pour l'USDP, avec un découpage en quatre étapes (Etude préliminaire, Elaboration, Conception et Transition), aboutissant au prototype présenté.

Remarque :

Sur un délai plus long, ce cycle aurait été répété, aboutissant à chaque fois à une nouvelle version, présentable au client pour obtenir sa position vis-à-vis du produit.

De plus, reprenant les principes des méthodes Agile, nous avons découpé l'étape de construction en sous-cycles. Chacun de ces cycles débutait par une réunion d'ensemble, permettant de mettre en commun les remarques et définir un sous-planning. S'ensuivait, pour chaque équipe, une phase d'élaboration puis de conception. Enfin, le cycle se terminait par une phase d'intégration des éléments pertinents (préalablement testés et validés).

Cette méthode dynamique nous permit d'avancer par incrément valides, et donc de tester au fur et à mesure chaque fonctionnalité. De plus, les diverses réunions et phases d'intégration (rassemblant au moins deux équipes) permirent à chacun de prendre connaissance du travail des autres et d'avoir une vue d'ensemble sur l'application.

Outils de gestion

Afin de mettre en œuvres cette organisation, nous avons employé différents outils de synchronisation et de gestion de projet, dont voici les deux principaux :

SVN

La mise de place d'un serveur SVN rendit possible des mises en commun plus fréquentes du travail de chacun. Il requiert cependant la mise en place de règles strictes de gestion du dépôt (ne pas *commit* des éléments non-testés, bloquer les fichiers en modification, explorer l'arborescence, etc.).

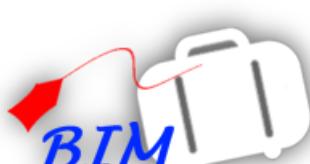
Redmine

[Redmine](#) est une application web Open Source de gestion complète de projet. Elle permet notamment de gérer les rôles attribués aux différents membres, de lancer des demandes d'intervention (tâches à effectuer, erreur à corriger, etc.), de définir le temps estimé, puis le temps passé, sur chaque tâche.

Il peut également être couplé au SVN, afin d'avoir également un aperçu constant des modifications apportées.

Cet outil de gestion nous permit notamment de réaliser un véritable calendrier prévisionnel et dynamique, de générer le diagramme de Gant, d'assigner chaque tâche à une équipe, et de totaliser le temps passé sur le projet.

Voici un exemple d'attribution de tâche :



TP DOO & IHM

Aperçu Activité Planning Demandes Nouvelle demande Gantt Calendrier Annances Documents Wiki Fichiers Dépôt Configuration

Création #42

✍ Mettre à jour ⌚ Saisir temps ⭐ Ne plus surveiller 🔗 Dupliquer 📄 Copier 👉 Déplacer 🗑 Supprimer

[VUE] GERER VUES

Ajouté par Benjamin Planche il y a environ 20 heures. Mis à jour il y a environ 2 heures.

Statut:	Nouveau	Début:	14/11/2011
Priorité:	Urgent	Echéance:	24/11/2011
Assigné à:	Nicolas Phan	% réalisée:	<div style="width: 100%;">100%</div>
Catégorie:	-	Temps passé:	22.00 heures
Version cible:	-	Temps estimé:	24.00 heures

Sous-tâches

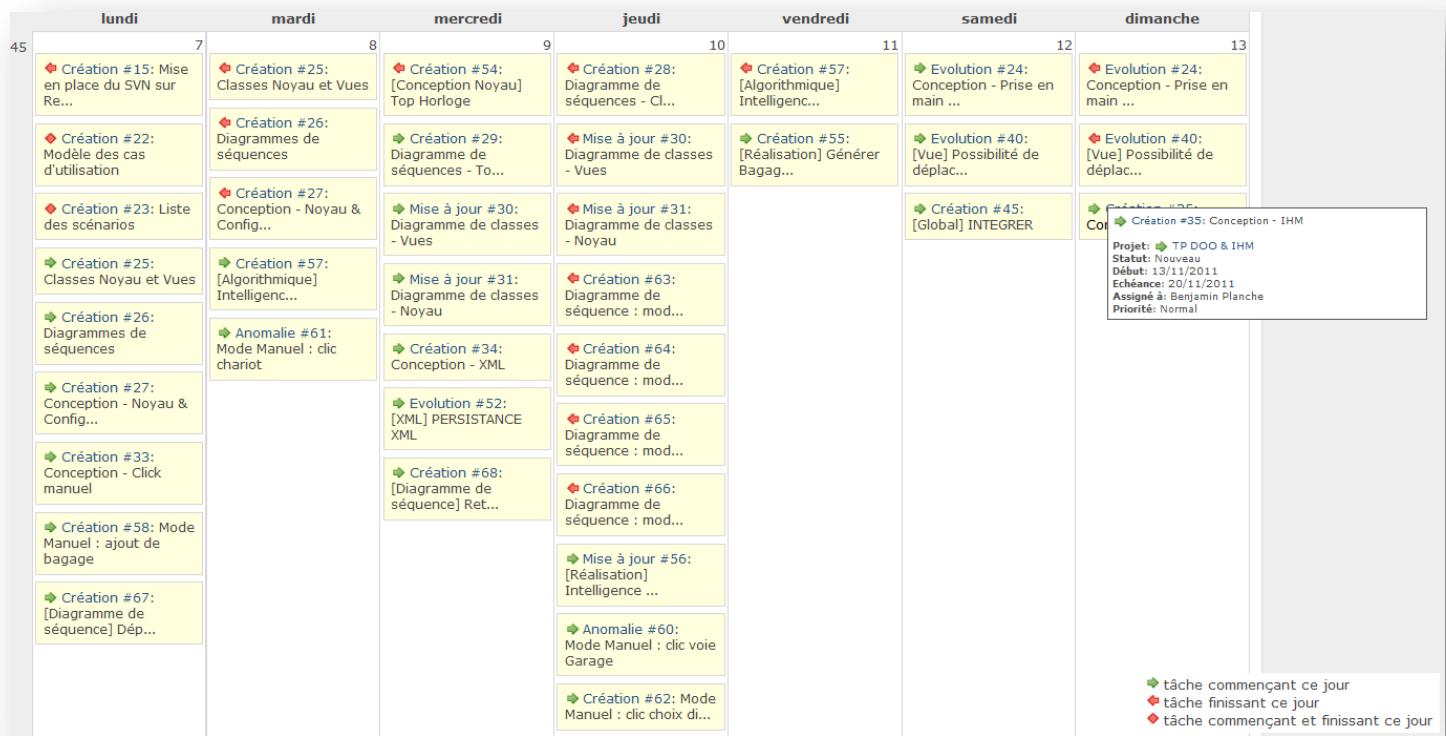
Anomalie #44: [Vue] Création des classes vues affichant les éléments de...	Nouveau	Nicolas Phan	<div style="width: 100%;">100%</div>	Ajouter
Création #47: [Vue] Diagramme de classe des vues	Nouveau	Nicolas Phan	<div style="width: 100%;">100%</div>	
Création #48: [Vue] Diagramme de séquence de la méthode Dessiner	Nouveau	Nicolas Phan	<div style="width: 100%;">100%</div>	
Correction #49: [Vue] Correction des vues suite à l'intégration du module...	Nouveau	Nicolas Phan	<div style="width: 100%;">100%</div>	

Demandes liées

Ajouter

Figure 1 Exemple de gestion d'intervention sous RedMine

Et un exemple du calendrier ainsi créé :



Livrables

Liste des livrables

Dans un souci de communication optimale avec le client, il est important dans un projet de préalablement définir l'ensemble des livrables à retourner.

Dans notre cas, ces livrables ont été condensés en un seul rapport, dont le sommaire est la traduction directe du tableau suivant.

Tableau récapitulatif des documents et des acronymes correspondants

- XXX - Etude préliminaire
- XXX - Elaboration
- XXX - Construction
- XXX - Transition

Document	Acronyme
Tableau activités / personne sur l'ensemble du projet	PPT-a
Diagramme de précédence des livrables	PPD
Liste de besoins potentiels sur SGBag partiel	LBP
Modèle du domaine du SGBag : liste de classes et diagramme de classes associé (sans attribut)	MD
Liste des acteurs et leur définition	DA-a
Diagramme des cas d'utilisation de SGBag partiel	DCU-a
Description des cas d'utilisation en définissant les rôles des acteurs	RACU-a
Liste des acteurs et leur définition	DA-b
Diagramme des cas d'utilisation de SGBag partiel	DCU-b
Description des cas d'utilisation en définissant les rôles des acteurs	RACU-b
Liste des scénarios des cas d'utilisation pour la Simulation (identification) de ceux à réaliser)	LSCU-a
Diagrammes de séquence pour les scénarios d'arrivée d'un bagage	DSbag-a
Diagrammes de séquence pour les scénarios de top d'horloge	DSclock-a
Diagramme de paquetages d'analyse (pour SGBag complet)	DPA
Diagramme séquence pour le scénario d'arrivée des bagages	DSbag-b
Diagramme séquence pour le scénario d'arrivée de top d'horloge	DSclock-b
Diagramme de classes du Prototype	DCP-a
Diagrammes de séquences pour les scénarios additionnels (ex : lecture du	DSA

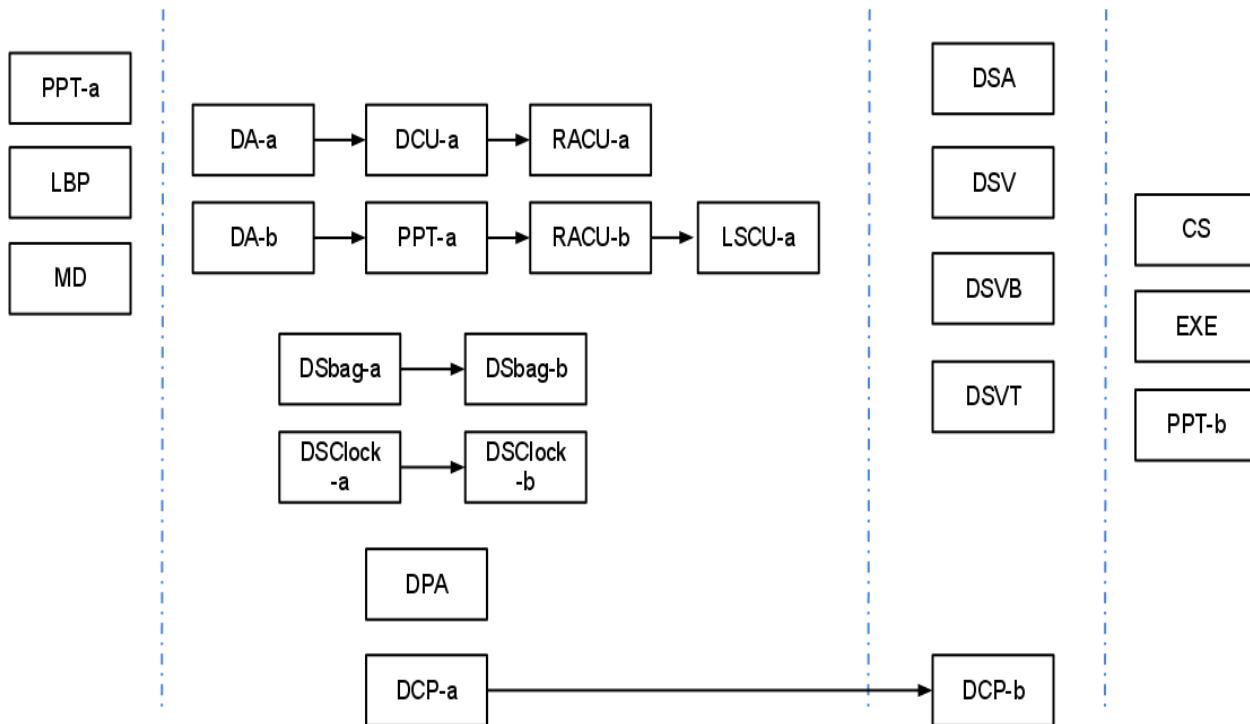


fichier de configuration)	
Diagrammes de séquences des scénarios précédents avec les classes "vue" des objets	DSV
Diagrammes de séquences pour l'arrivée manuelle d'un bagage avec les classes "vue"	DSVB
Diagrammes de séquences pour le changement de vitesse d'un tapis roulant avec les classes "vue" des objets	DSVT
Diagramme de classes complet de l'application	DCP-b
Code Source	CS
Exécutable	EXE
Tableau activités / personne sur l'ensemble du projet final	PPT-b

Diagramme de précédence des livrables

Cette liste suit un ordre d'élaboration logique, associé à l'avancement du projet et de ses étapes. Néanmoins, une réflexion sur son organisation permet de mettre en évidence des possibilités de parallélisation de certaines tâches, et donc des rapports associés.

Le diagramme suivant, consulté tout au long du projet, présente le fruit de cette réflexion



Planning prévisionnel

Tâches fondamentales du chef de projet, la répartition des différentes tâches et l'élaboration d'un planning sont essentiels à la concrétisation d'un projet dans le temps imparti.

Pour se faire, nous avons opté pour les outils fournis par l'application qui nous ont permis de construire le calendrier prévisionnel et d'assigner les multiples tâches à chacun.

Nous avons initialement défini notre planning par un diagramme de Gantt, généré par

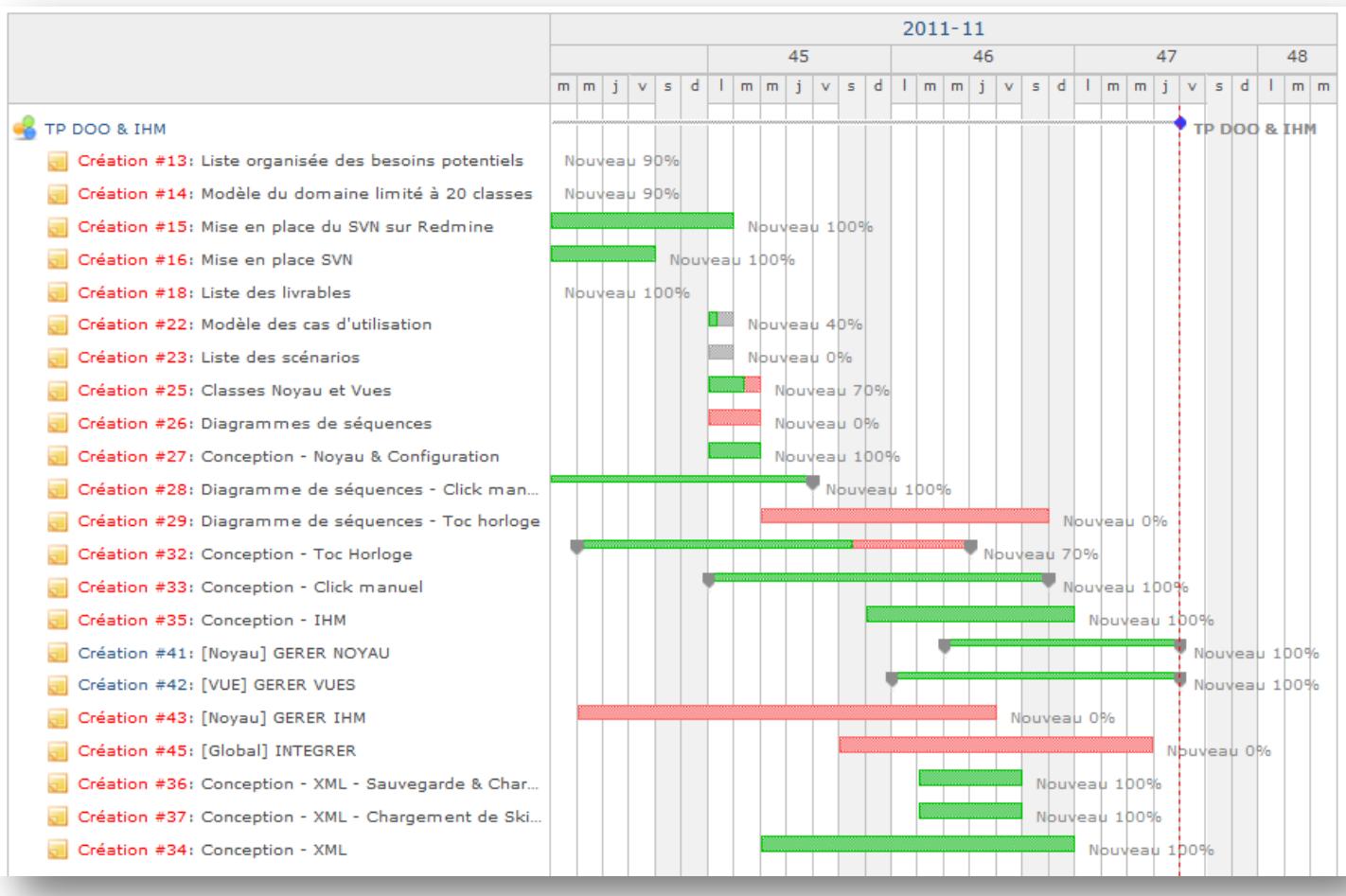


Figure 3 Diagramme de Gantt

Redmine, après évaluation des différentes tâches, de leur ordonnancement et de leur durée :

De la même manière, la répartition des tâches ne s'est pas faite à l'aide d'un tableau à double-entrée, mais par génération de demandes sur Redmine.

Pour une liste exhaustive des tâches et de leur répartition, voir la répartition des rôles.



Périmètre du projet SGBag-Partiel

L'étude préliminaire a notamment pour but de déterminer la faisabilité et le périmètre du projet, avant de se lancer dans toute analyse plus poussée. Nous renseignerons ainsi dans cette partie la liste des besoins potentiels relevés pour l'application complète, ainsi que le modèle du domaine retenu.

Liste de besoins potentiels

- **Sous-système Supervision**
 - **Authentification**
 - Voir statistiques
 - Sauvegarder statistiques
 - Réinitialiser statistiques
 - Accéder à zone (zoom, dé zoom)
 - Accéder aux informations d'un élément
 - **Configuration**
 - Sauvegarder / Charger configuration
 - Ajouter/Modifier/Supprimer élément:
 - Parcours
 - Élément (chariot, carrousel...)
 - Zones
 - Modifier Paramètres : vitesse tapis, ...
 - **Maintenance**
 - Activer/désactiver dispositif (tapis, chariot...)
 - Vérifier paramètres
 - Créer/Modifier/Stocker fiche intervention
 - **Simulation**
 - Passage d'un mode à l'autre
 - Diffuser statistiques (au responsable technique après simulation)
 - Mode Automatique :
 - Choisir loi de comportement
 - Lancer la simulation
 - Mode Manuel :
 - Lancer un évènement (panne, bagage suspect...)
 - Lancer évènement bagage (= arrivée manuelle)
 - Modifier paramètres bagages (nombre/minute etc.)
 - Choix direction/action des chariots
 - Modifier paramètres dispositifs



- **Exploitation**
 - Observer signaux d'un ou plusieurs capteurs spécifiques
 - Capteurs actifs, capteurs passifs
 - Visualiser PDA personnel
 - Alerter un ou plusieurs PDA de personnel
 - Pilotage manuel:
 - Activer/désactiver dispositif
 - zone
 - chemin
 - objet (tapis...)
 - Chariot:
 - démarrer
 - envoyer en maintenance
 - sortir de maintenance
 - modifier destination
 - modifier chemin
 - Débloquer bagage
 - ...
- **Sous-système Réclamation des bagages**
 - Visualisation trajet d'un bagage avec le numéro identifiant
 - Dossier de litige
 - Créer (numéro identification + mot de passe)
 - Modifier
 - Clore
 - Visualiser

Modèle du domaine de SGBag

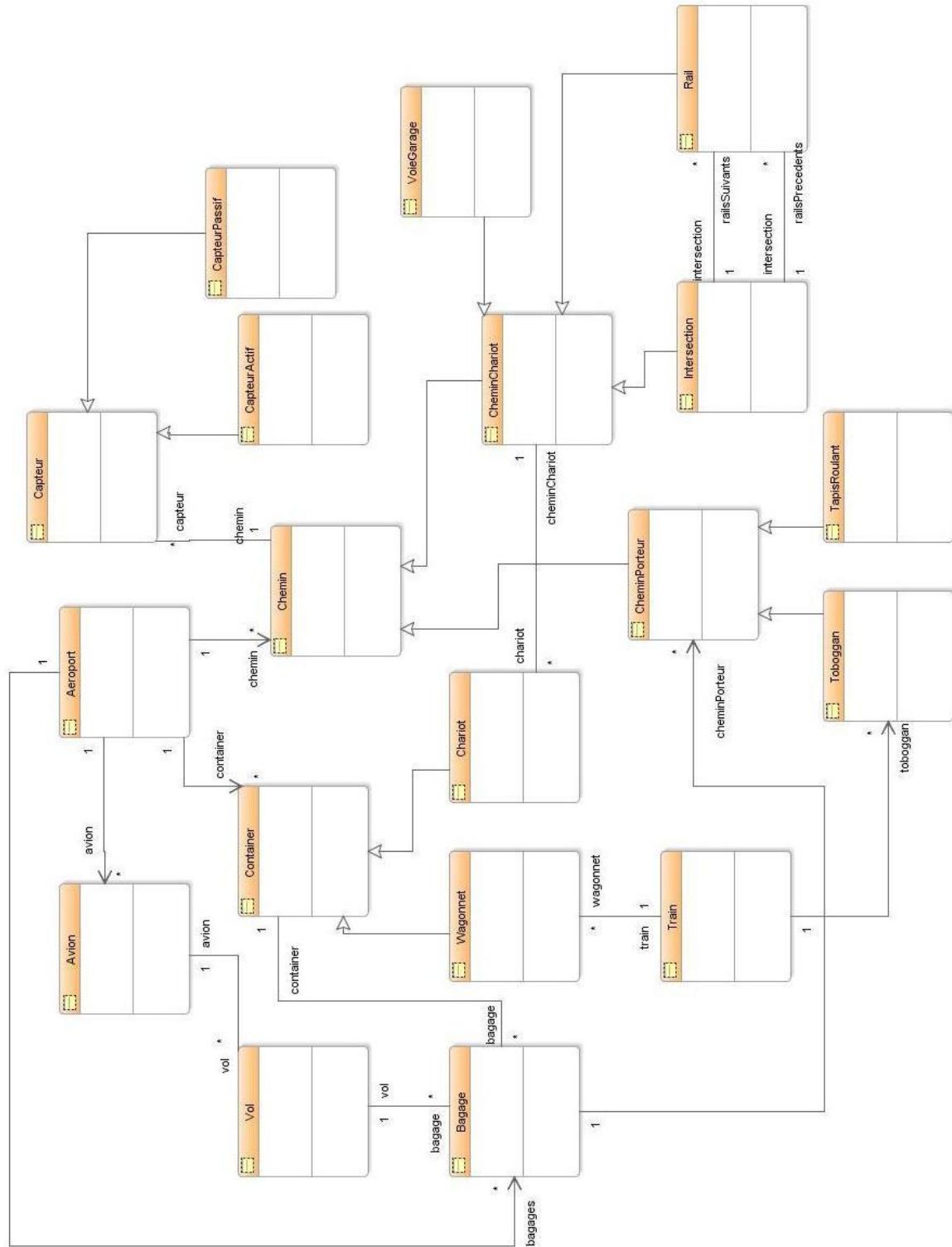
Liste des éléments :

- Aéroport
- Avion
- Vol
- Bagage
- Container
- Waggonnet
- Chariot
- Train
- Capteur
- CapteurActif
- CapteurPassif
- Chemin
- CheminPorteur



- CheminChariot
 - TapisRoulant
 - Toboggan
 - Intersection
 - Rail
 - VoieGarage

Diagramme de classe représentant le système SGBag :



Conclusion

Cette étude préliminaire nous a permis de vérifier que le projet était viable et nous avons pu, pour l'aborder, mettre en place une démarche de gestion de projet en définissant le rôle de chacun des membres de l'hexanôme.



Elaboration

Introduction

Dans ce document d'élaboration, nous avons pu cerner les acteurs de notre programme, et les cas d'utilisation dans lesquels ils interviennent. Nous nous concentrerons ensuite sur la partie *Simulation*, et réaliserons les scénarios correspondant à ces cas précis. A l'aide de diagramme de séquence nous explicitons nos choix, spécialement pour la gestion manuelle d'ajout de bagage et la gestion du temps (top horloge).

Capture des besoins

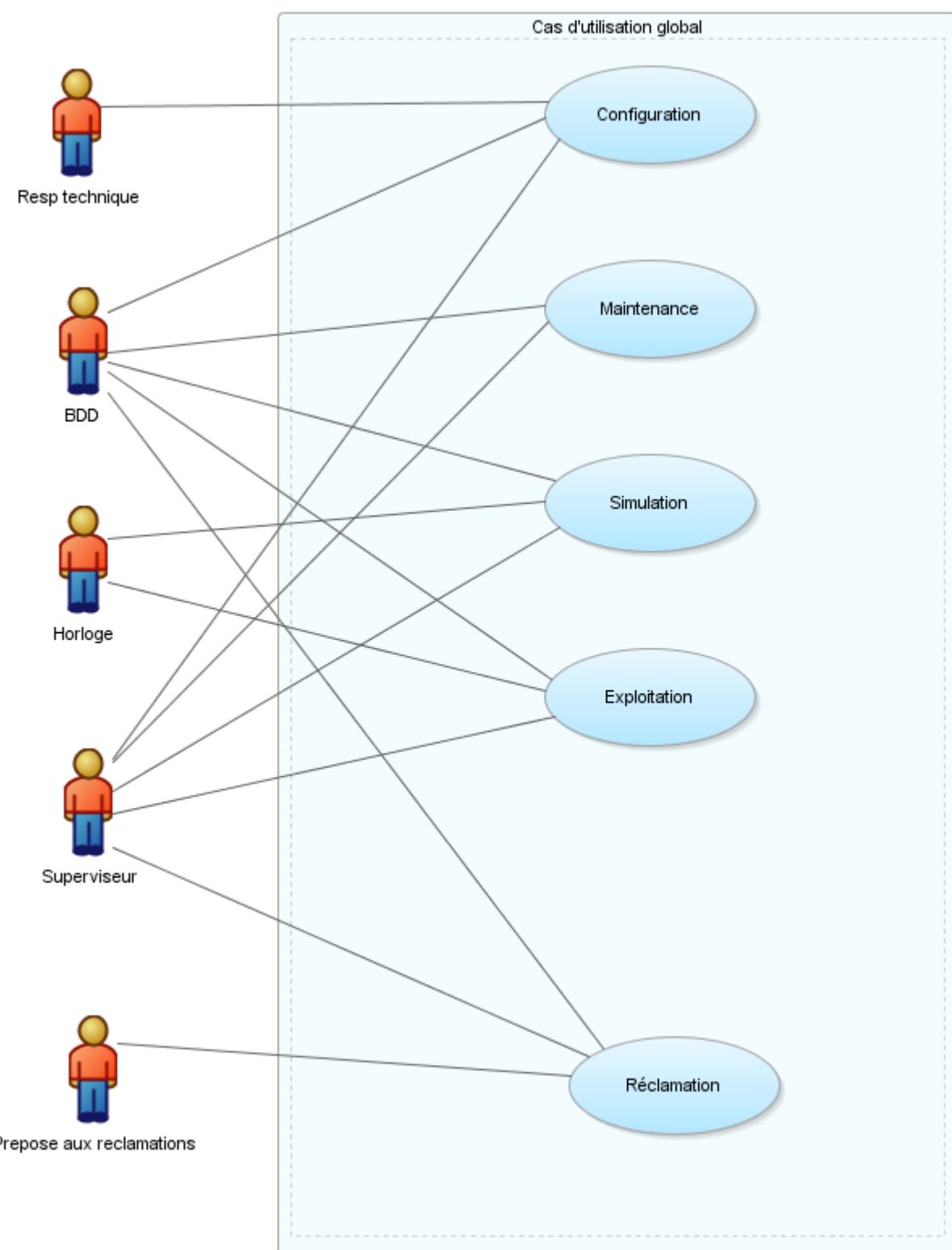
Modèle de cas d'utilisations

SGBag Global

Liste des acteurs et leur définition

- **Responsable Technique :**
 - Gère la configuration du SGBag. Il peut également s'occuper de la simulation mais dans ce cas il prend la casquette de superviseur.
- **Superviseur :**
 - Gère le système SGBag à partir d'une configuration chargée. (simulation, exploitation, maintenance)
- **Horloge :**
 - Gère le temps dans le SGBag, générant les *tops-horloge* correspondant à un instant.
- **BDD :**
 - Contient et gère l'ensemble des données relatives à l'application (données sur le noyau, les vues, etc.).
Remarque : Implicite, il n'apparaît donc pas dans la suite de l'étude.
- **Préposé aux réclamations :**
 - C'est l'acteur qui reçoit/ gère les réclamations.



Diagramme des cas d'utilisation de SGBag partiel

Description des cas d'utilisation en définissant les rôles des acteurs

Sous-système Supervision

- **Créer Configuration**
 - C'est le Responsable Technique qui va définir la configuration du SGBag, en choisissant différents paramètres, l'architecture de l'aéroport.
- **Sauvegarder Configuration**
 - C'est le Responsable Technique qui sauvegarde la configuration précédemment créée.
- **Charger Configuration**
 - Le superviseur va charger une configuration créée par le Responsable Technique. Il ne peut la modifier.
- **Simulation Manuelle**
 - Le superviseur gère la simulation manuelle, il peut ajouter des éléments, des événements en temps réel.
- **Simulation Pilote Auto**
 - Le superviseur choisit certains paramètres de la simulation (loi d'arrivée des bagages par exemple), puis lance la simulation.
 - L'horloge effectue la gestion du temps.
- **Demande de Maintenance**
 - Le superviseur peut faire une demande de maintenance au Service Maintenance. Il aura alors une fiche à remplir.
- **Exploitation**
 - Le superviseur gère l'exploitation en intervenant quand cela est nécessaire.
 - L'horloge effectue la gestion du temps.

Sous-système Réclamations

- **Recevoir Réclamation**
 - Le Préposé aux réclamations reçoit la réclamation d'un passager ou d'un aéroport, va observer les données sur le bagage, créer un dossier de litige etc.

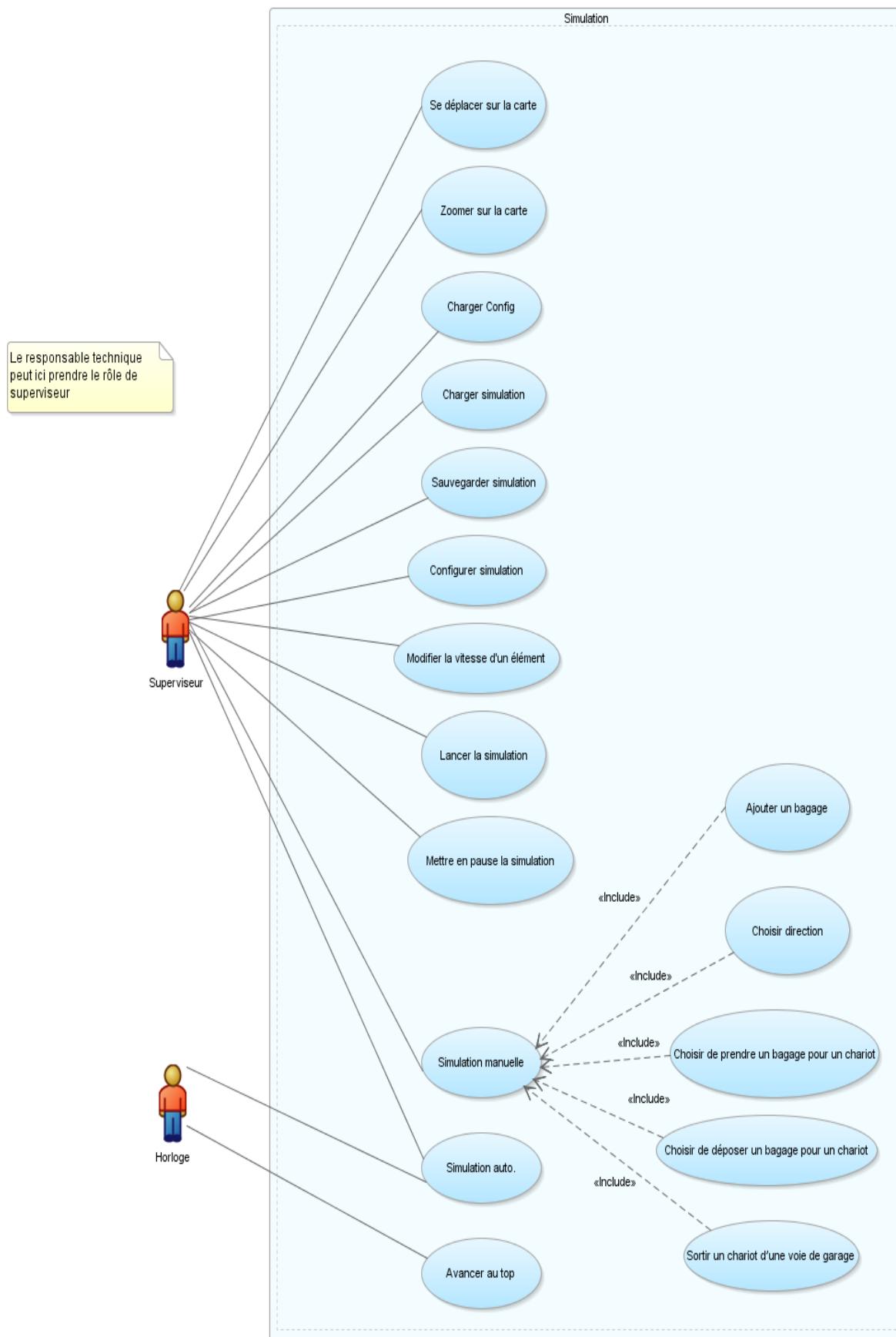
Simulation

Liste des acteurs et leur définition (DA-b)

- **Superviseur :**
 - Gère le système SGBag à partir d'une configuration chargée. (simulation, exploitation, maintenance)
- **Horloge :**
 - Gère le temps dans le SGBag, générant les *tops-horloge* correspondant à un instant.



Diagramme des cas d'utilisation de SGBag partiel (DCU-b)



Description des cas d'utilisation en définissant les rôles des acteurs (RACU-b)

- **Dans les deux cas :**
 - *Visualiser les informations d'un objet (Superviseur)*
 - Le Superviseur clique sur un objet, ses informations s'affichent dans le panel d'information.
- **Simulation Manuelle :**
 - *Ajouter un bagage (Superviseur)*
 - Le Superviseur choisit un vol, clique sur le bouton *ajout bagage*, les tapis roulants s'affichent en surbrillance, il clique sur un tapis, cela lui ajoute le bagage.
 - *Choisir la direction/action d'un chariot (Superviseur)*
 - Le chariot arrêté à une intersection est affiché en surbrillance, le superviseur clique sur le chariot, les choix possibles de direction/action sont affichés : rail suivant, poser le bagage ou prendre un bagage, le Superviseur clique sur l'option de son choix.
 - *Avancer d'un top (Horloge)*
 - Au top de l'horloge, tous les éléments avancent. (position ou progression)
- **Simulation Pilote Auto :**
 - *Configurer simulation (choix du paramétrage de la loi d'arrivée ...) (Superviseur)*
 - Le Superviseur choisit le paramètre de la loi de Poisson d'arrivée des bagages, la vitesse de mise à jour du moteur, la fréquence de rafraîchissement de l'écran, ...
 - *Lancer la simulation (Superviseur)*
 - *Avancer d'un top (Horloge)*

Liste des scénarios des cas d'utilisation pour la Simulation (identification de ceux à réaliser)

- **Modifier la vitesse d'un élément**
 - Stimulus :
 - L'utilisateur clique sur un élément "sélectionnable" (chariot, tapis roulant, toboggan) de la carte.
 - L'utilisateur peut entrer une nouvelle valeur de vitesse pour cet élément dans le panel présentant celui-ci situé en bas à gauche de l'écran.
- **Avancer au top**
 - Stimulus : L'intervalle de temps entre deux actualisations du système arrive à échéance
- **Lancer la simulation**
 - Stimulus : L'utilisateur clique le bouton de démarrage de la simulation
- **Mettre en pause la simulation**
 - Stimulus : L'utilisateur clique le bouton de mise en pause de la simulation



- **Arrêter la simulation**
 - Stimulus : L'utilisateur clique le bouton d'arrêt de la simulation
- **Configurer la simulation**
 - Stimulus : L'utilisateur fait varier un paramètre de la simulation (vitesse des moteurs, Fréquence d'actualisation des vues, génération d'événements)
- **Changement de mode de simulation**
 - Stimulus : L'utilisateur clique sur le *radio button* "Manu." ou "Auto." pour passer d'un mode à l'autre.
- **Charger une configuration**
 - Stimulus :
 - L'utilisateur clique sur le bouton "Fichier" de la barre de menu puis sur charger la configuration.
 - L'utilisateur navigue dans l'arborescence de sa machine et sélectionne le fichier XML correspondant à une configuration.
- **Sauvegarder une simulation**
 - Stimulus :
 - L'utilisateur clique sur le bouton "Fichier" de la barre de menu puis sur "Sauvegarder Simulation".
 - L'utilisateur navigue dans l'arborescence de sa machine et enregistre le fichier XML correspondant à la simulation dans le répertoire sélectionné.

- **Charger une simulation**

Stimulus : L'utilisateur clique sur le bouton "Fichier" de la barre de menu puis sur "Charger Simulation".

- L'utilisateur navigue dans l'arborescence de sa machine et sélectionne le fichier XML correspondant à une simulation.

- **Zoomer sur la carte**
 - Stimulus : L'utilisateur utilise la molette de la souris pour augmenter le zoom ou le diminuer.
- **Se déplacer sur la carte**
 - Stimulus : L'utilisateur utilise les flèches pour se déplacer sur la carte.
- **Ajouter un bagage (mode manuel)**
 - Stimulus :
 - Le mode manuel est activé et l'utilisateur clique sur le bouton "Ajouter bagage" (en bas du panel de l'onglet simulation avec l'icône représentant un bagage)
 - L'utilisateur sélectionne parmi une liste, le vol à affecter au bagage
 - L'utilisateur clique sur l'un des tapis roulants de la carte
- **Choisir direction (mode manuel)**
 - Stimulus :
 - Le mode manuel est activé et un chariot arrive à un embranchement.
 - L'utilisateur clique sur l'une des directions possibles pour le chariot (des flèches s'affichent à l'écran)



- **Choisir de prendre un bagage pour un chariot (mode manuel)**
 - Stimulus :
 - Le mode manuel est activé et un chariot arrive à une intersection correspondant à l'arrivée d'un tapis roulant qui possède au moins un bagage.
 - L'utilisateur clique sur une icône représentant un chariot présentée au-dessus du chariot.

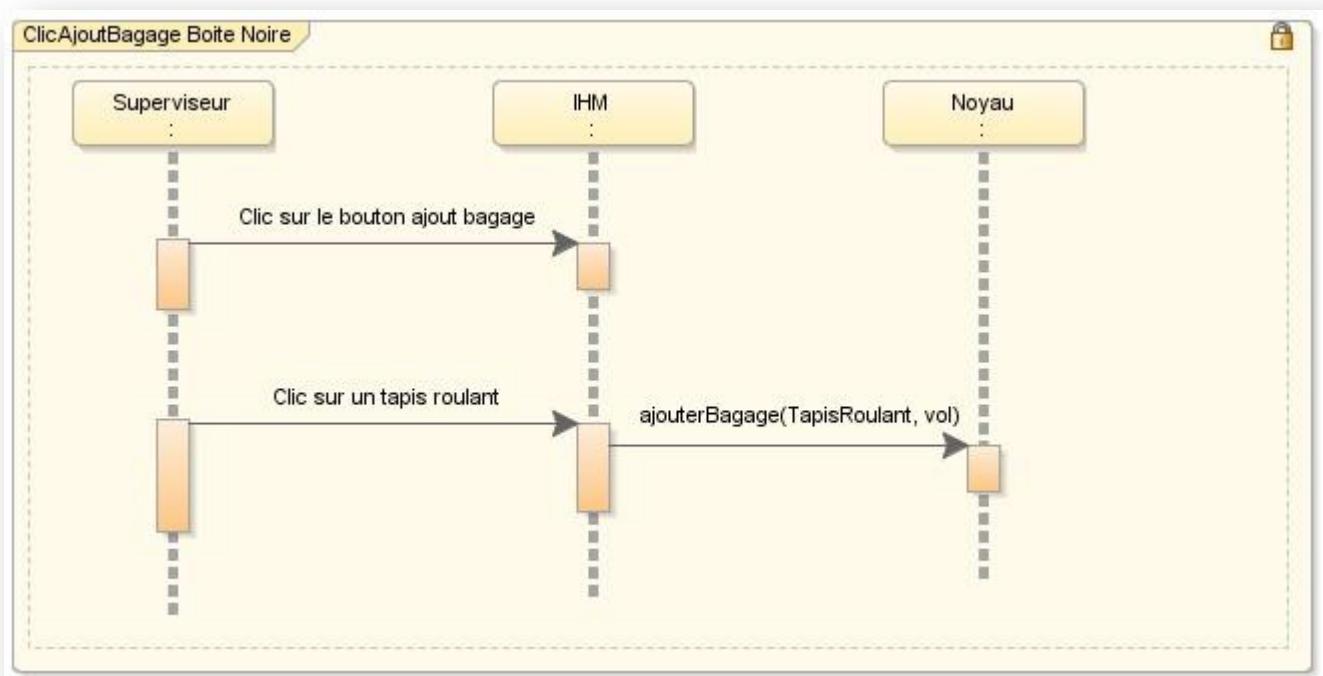
- **Choisir de déposer un bagage pour un chariot (mode manuel)**
 - Stimulus :
 - Le mode manuel est activé et un chariot arrive à une intersection correspondant au toboggan relatif au bagage chargé dans le chariot.
 - L'utilisateur clique sur une icône représentant un chariot en direction du toboggan présentée au-dessus du chariot.

- **Sortir un chariot d'une voie de garage (mode manuel)**
 - Stimulus : Le mode manuel est activé et l'utilisateur clique sur une voie de garage contenant au moins un chariot.

Diagrammes de séquences

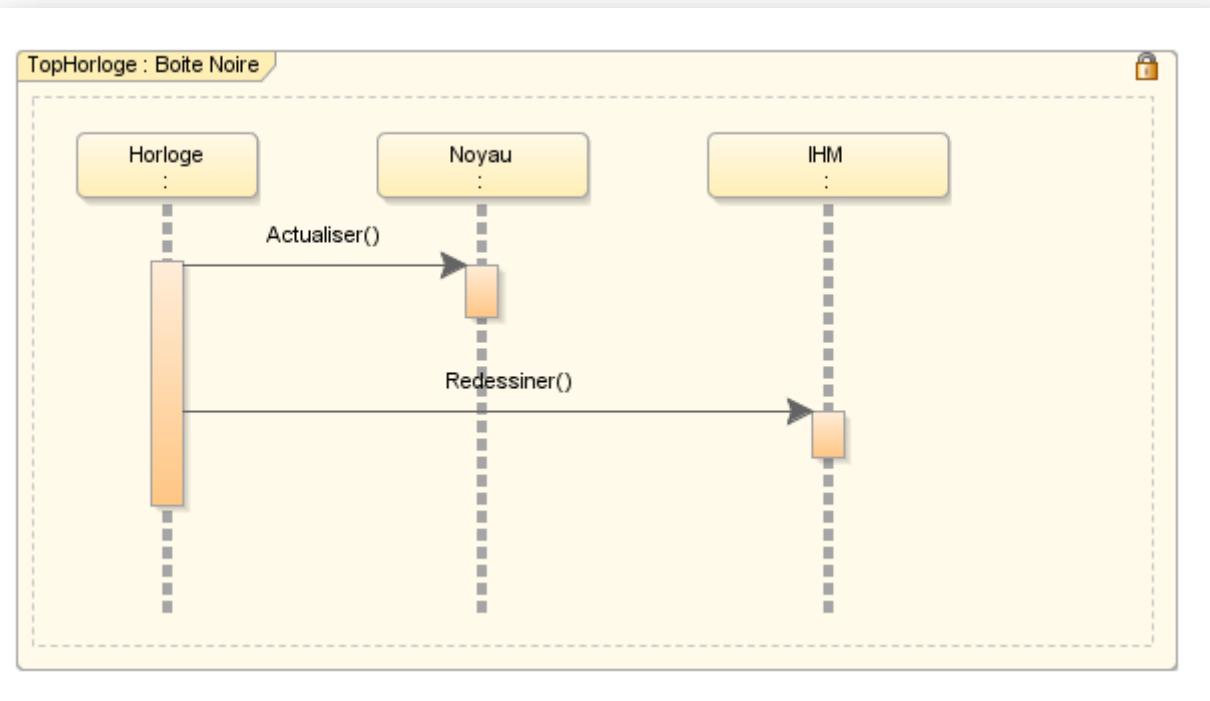
Arrivée d'un bagage

Le diagramme suivant transcrit le scénario d'ajout de bagage en mode manuel, effectué par le superviseur (boîte noire).



Top d'horloge

Le diagramme suivant transcrit le scénario des procédures lancées à chaque top horloge (boîte noire)



Analyse

Diagramme de paquetages d'analyse

Voici le découpage en package et sous package du système SGBag complet.

- **vue**
 - configuration
 - simulation
 - maintenance
 - exploitation
 - authentification
 - statistiques
 - reclamation_de_bagage
 - divers
- **noyau**
 - configuration
 - simulation
 - maintenance
 - exploitation
 - authentification



- statistiques
- reclamation_de_bagage
- divers
- ihm
 - fenetre_configuration
 - fenetre_simulation
 - fenetre_maintenance
 - fenetre_exploitation
 - fenetre_authentification
 - fenetre_statistiques
 - fenetre_reclamation_de_bagage

Nous remarquons trois paquetages principaux, découpant l'application SGBag. Ceux-ci correspondent globalement au système Modèle/Vue/Contrôleur (MVC). Chacun de ses paquetages sera sous-découpé afin de séparer chaque fonctionnalité possible du logiciel.

Le paquetage "divers" permettra de stocker les méthodes utiles au bon fonctionnement général.

Diagramme séquence pour le scénario d'arrivée des bagages

Le diagramme suivant transcrit le scénario d'ajout de bagage en mode manuel, effectué par le superviseur, en développant les éléments appelés au niveau du noyau (boîte grise).

Cette procédure intervenant plus au niveau de l'IHM qu'au niveau noyau, ce diagramme ne dévoile pour l'instant que peu d'informations.

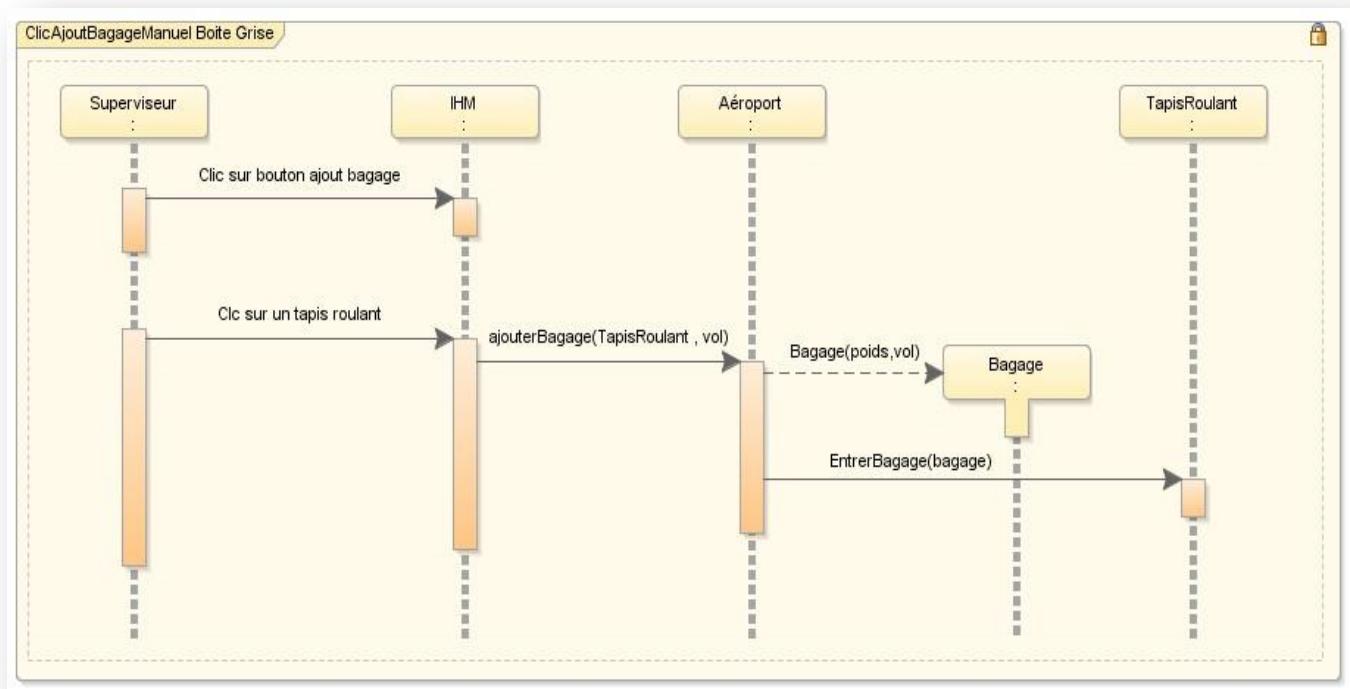
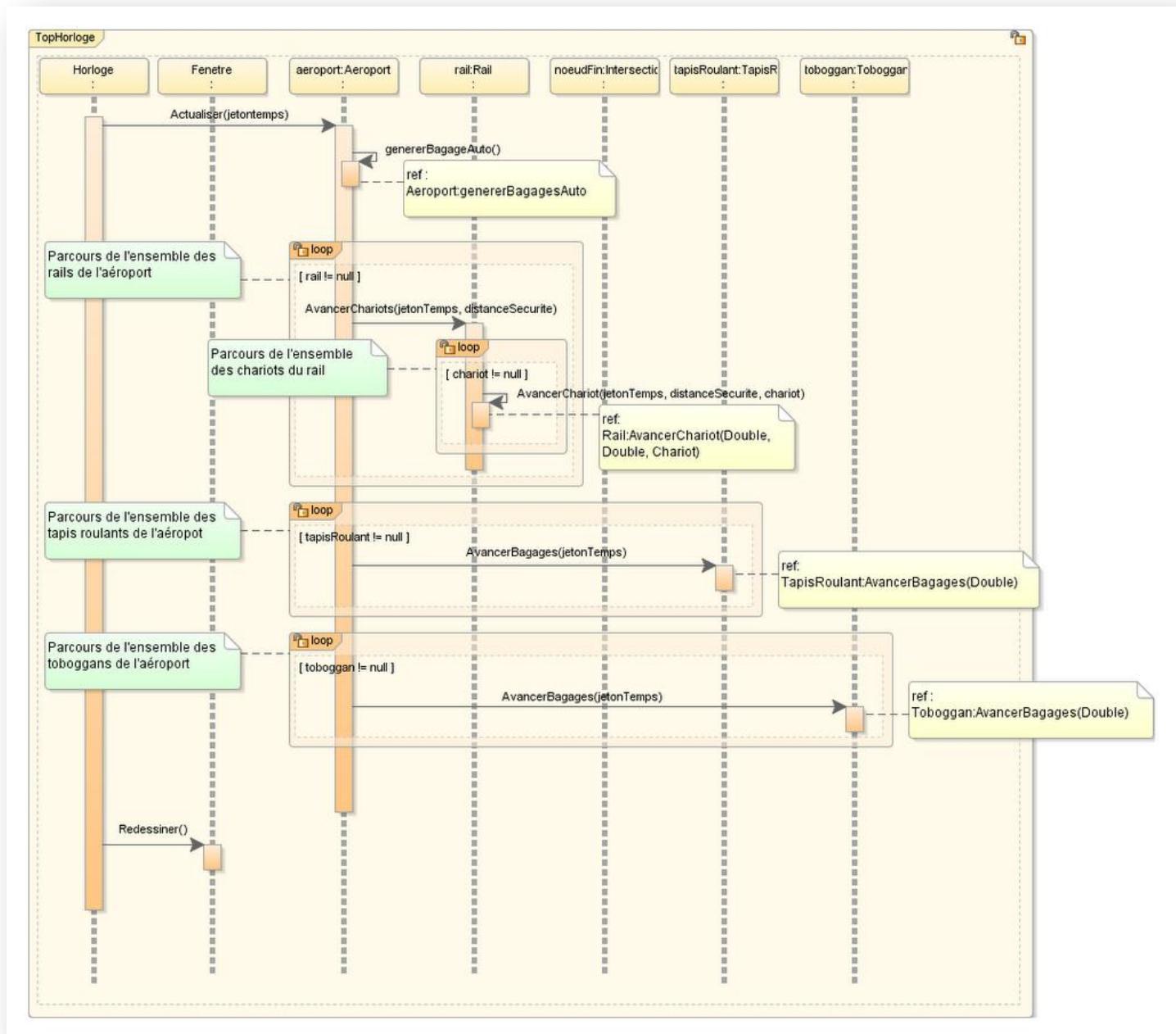


Diagramme de séquence pour le scénario d'arrivée de top d'horloge

Le diagramme suivant représente les actions réalisées sur le noyau suite à un top horloge du système, stimulus lançant son actualisation. En effet, le noyau est mis à jour à des intervalles de temps régulier, afin de garder le système dans un état au plus proche de la réalité.

Seuls les éléments Noyau sont développés (boîte grise).

Diagramme de séquence TopHorloge



Les références à l'intérieur des diagrammes de séquences correspondent à d'autres diagrammes de séquences présentés ci-après.



Diagramme de séquence GénérerBagageAuto

Ce diagramme présente la génération automatique des bagages sur les tapis roulants de l'aéroport.

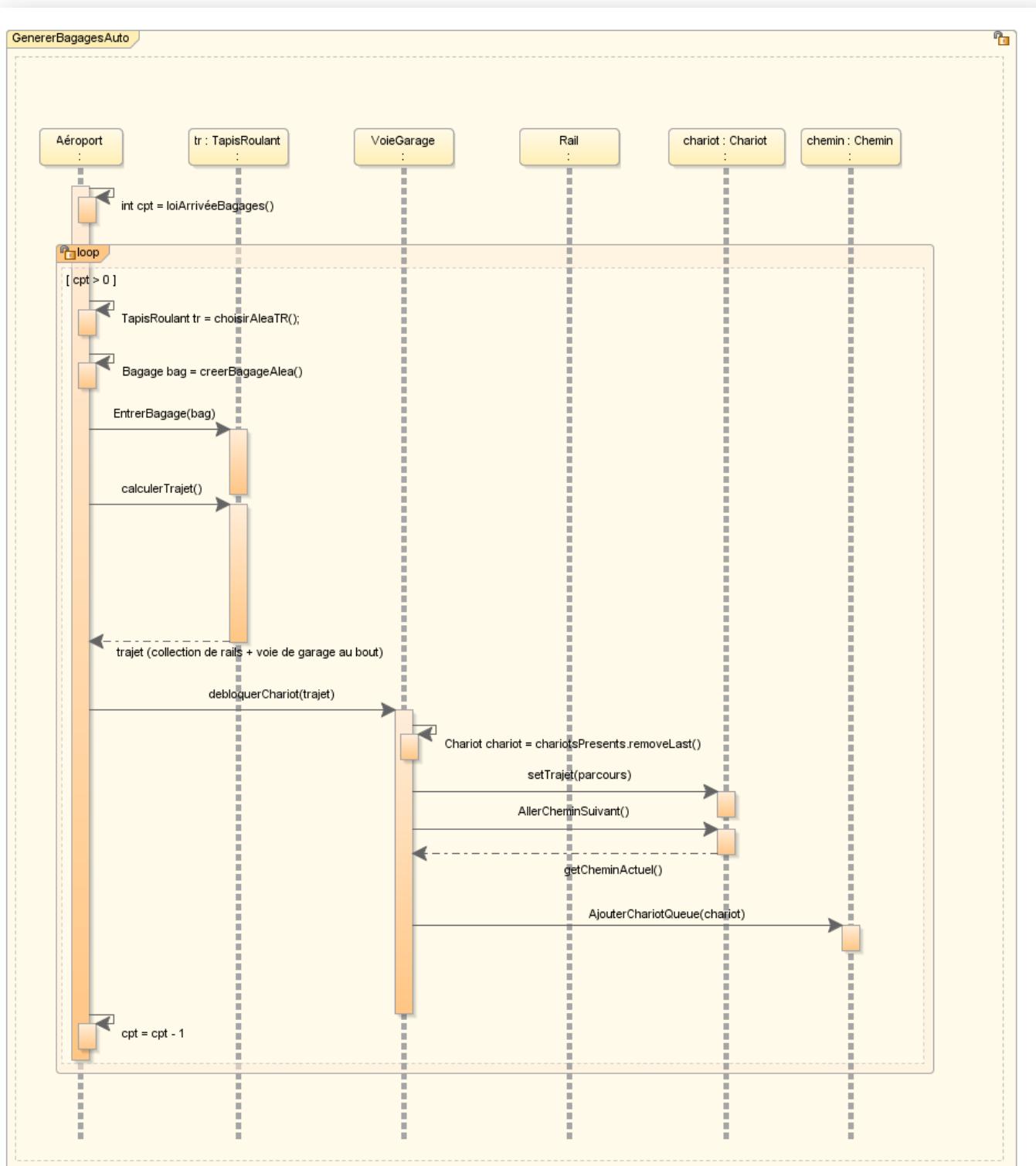


Diagramme de séquence AvancerBagagesTapis

Ce diagramme présente la manière dont un tapis roulant déplace les bagages qui sont posés dessus.

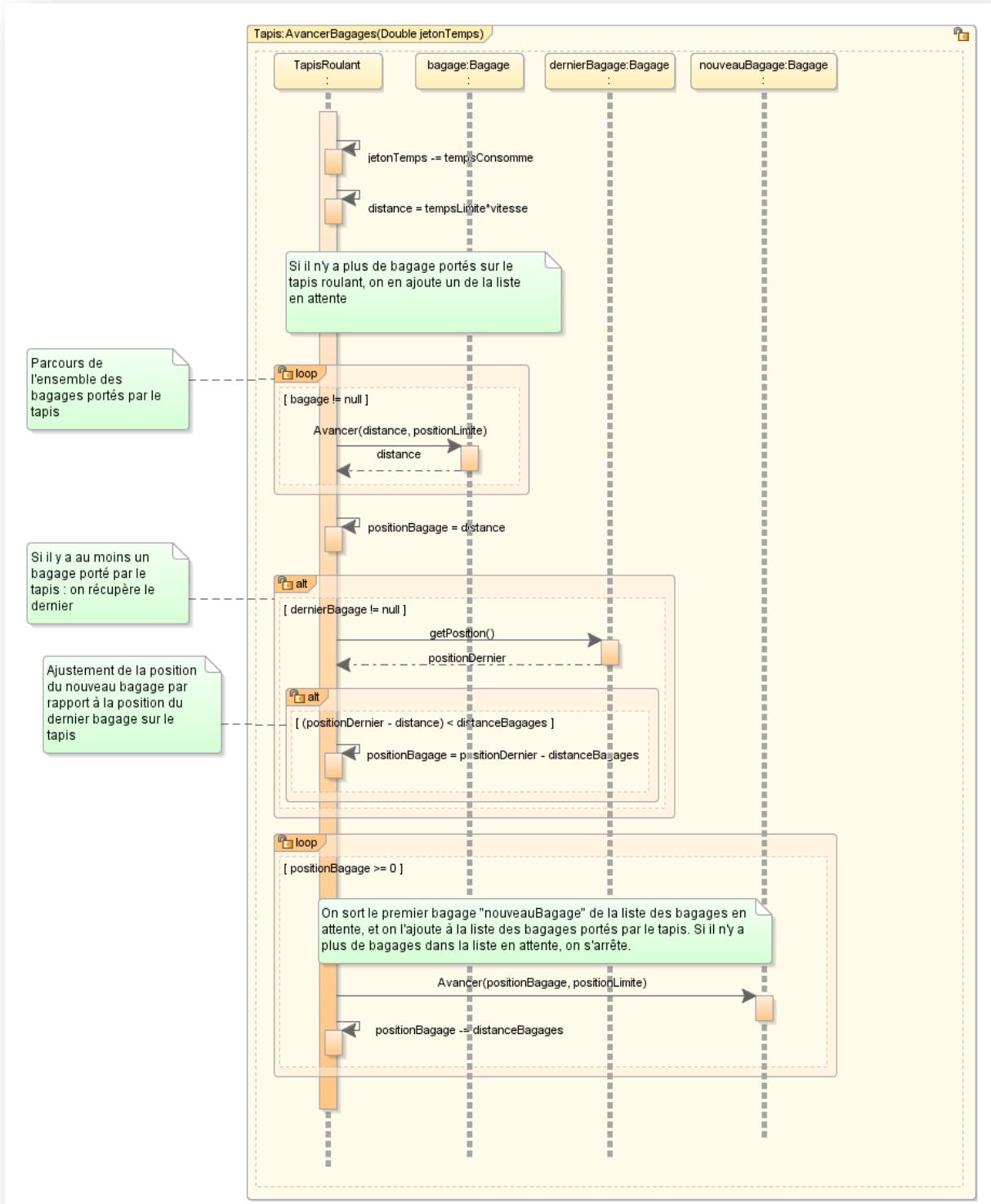


Diagramme de séquence AvancerBagagesToboggan

Ce diagramme présente la manière dont un toboggan déplace les bagages qui sont posés dessus.

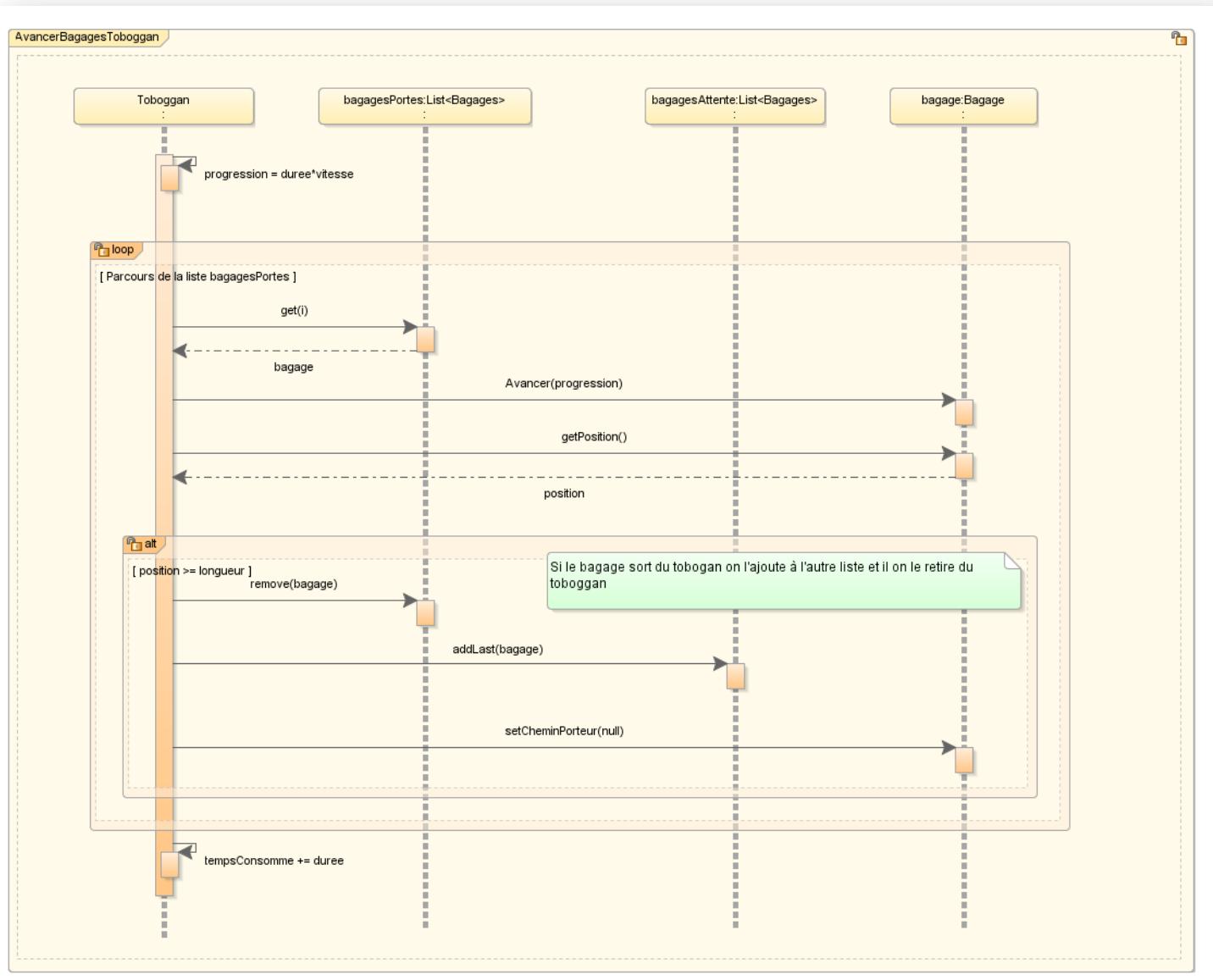


Diagramme de séquence AvancerChariot_Rail

Ce diagramme de séquence présente la méthode d'avancement des chariots pour un rail.

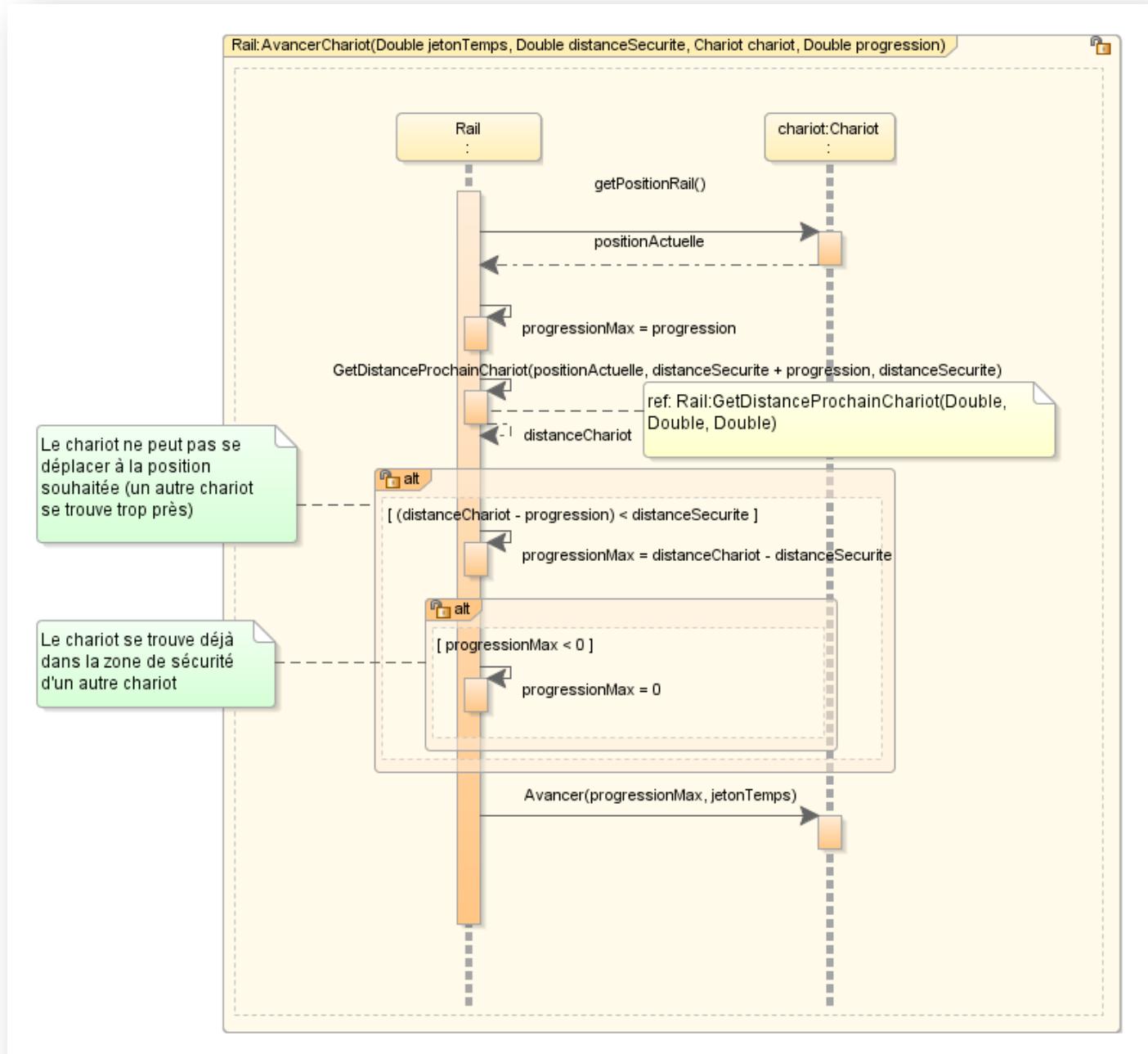


Diagramme de séquence GetDistanceProchainChariot_Rail

Ce diagramme de séquence présente la méthode de la classe Rail qui permet d'obtenir la distance jusqu'au au prochain chariot situé quelque part sur les rails suivants.

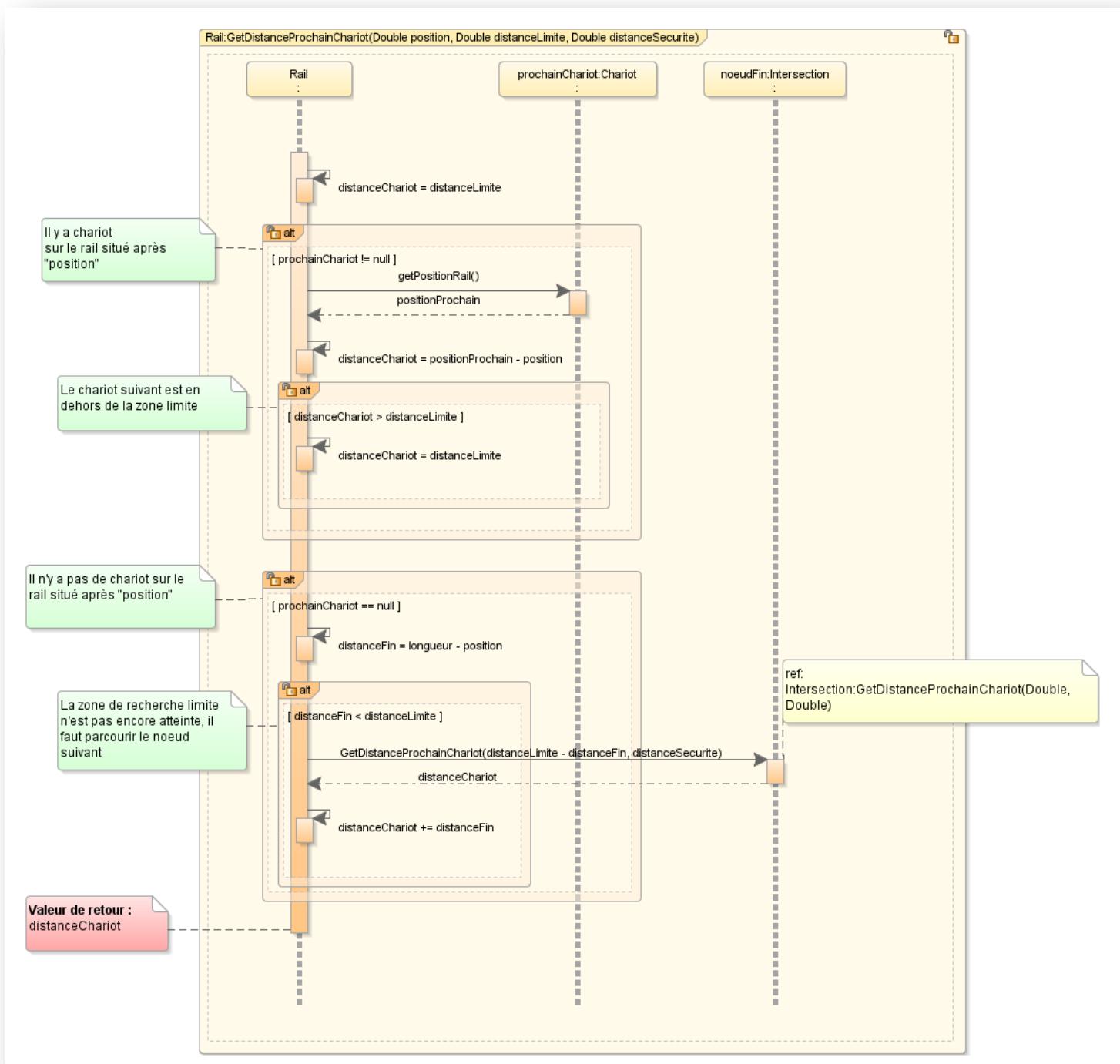
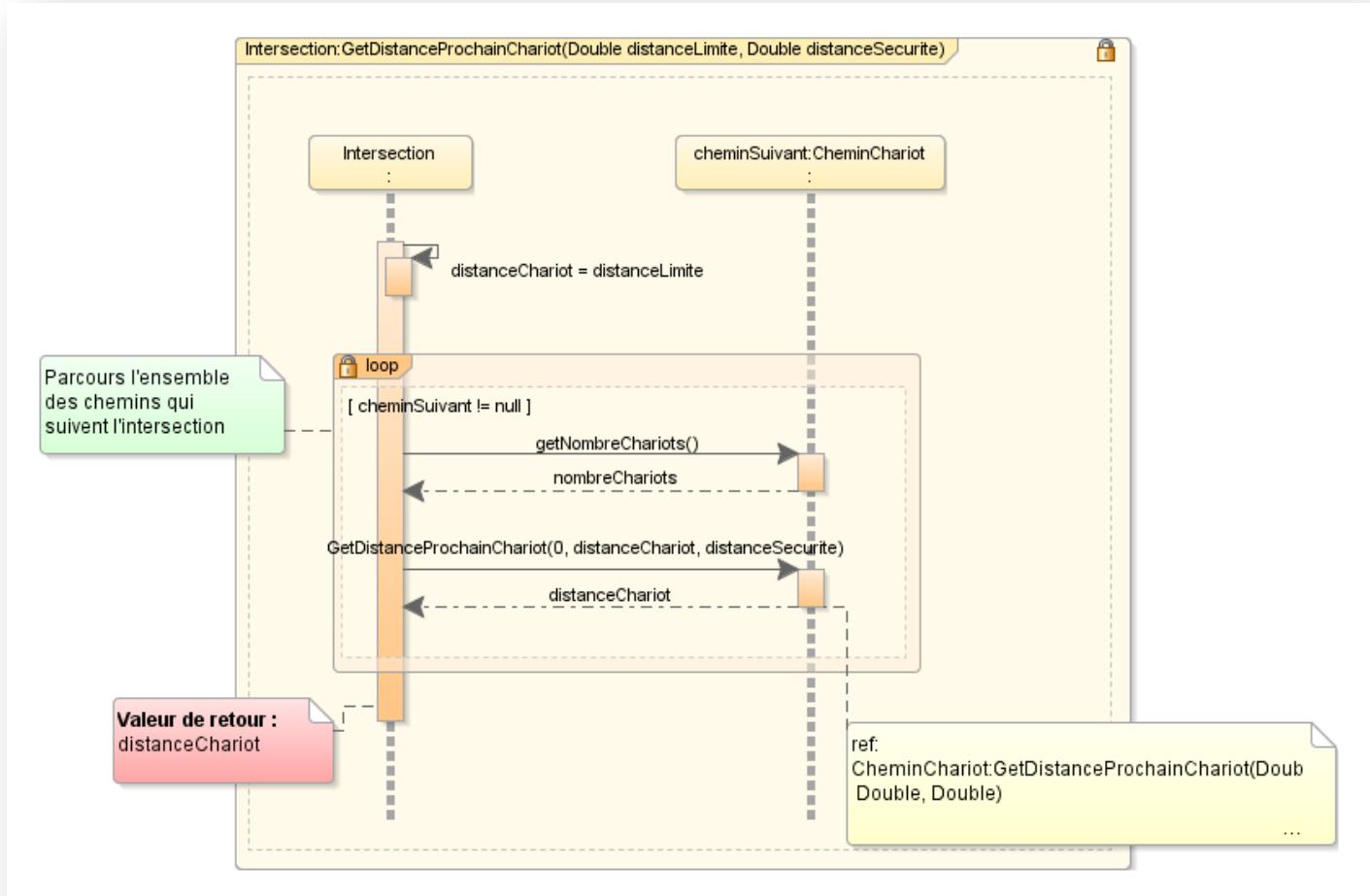


Diagramme de séquence GetDistanceProchainChariot_Intersection

Ce diagramme de séquence présente la méthode de la classe Intersection permettant de parcourir l'ensemble des chemins qui suivent l'intersection et qui retourne la distance jusqu'au prochain chariot.



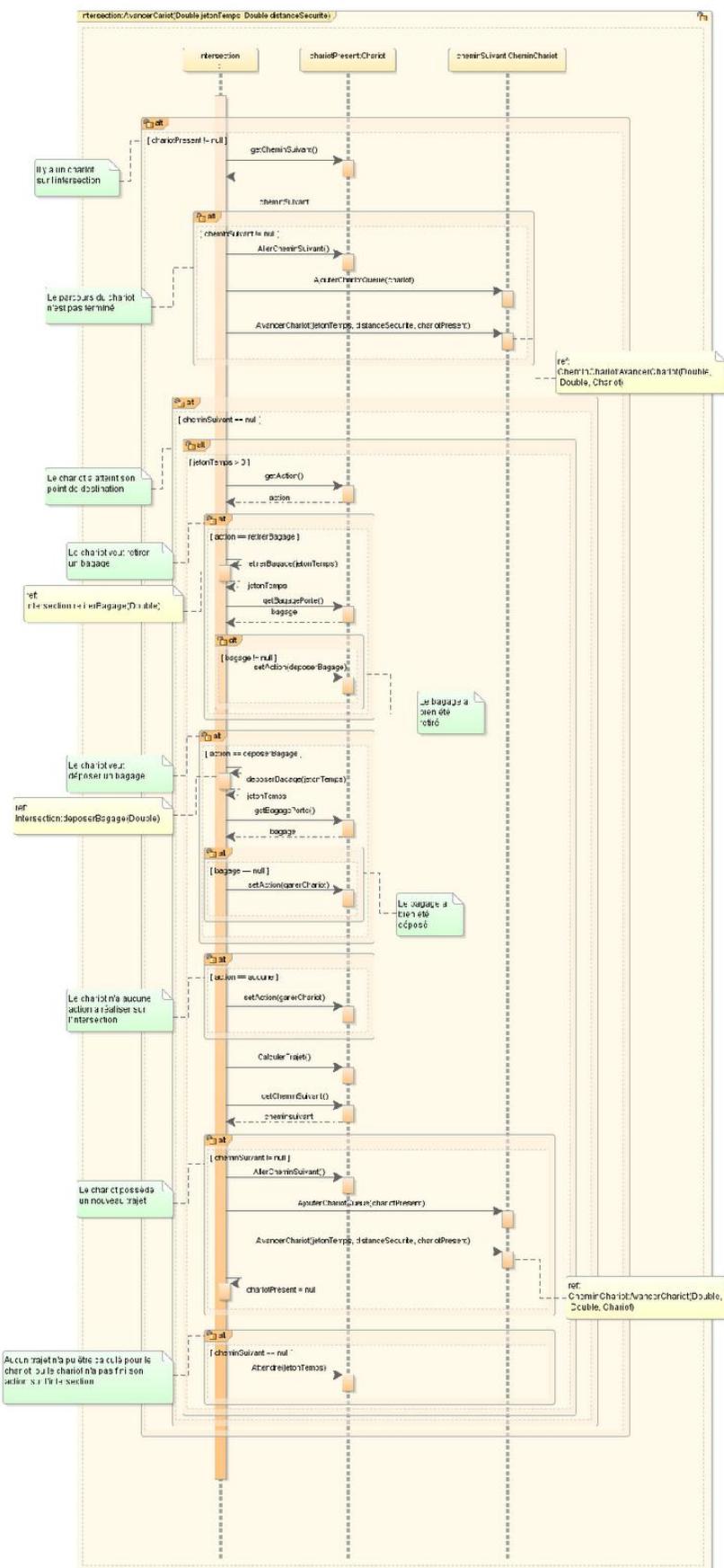


Diagramme de séquence DéposerBagage_Intersection

Ce diagramme de séquence présente le dépôt de bagage dans un toboggan depuis une intersection correspondant à l'embouchure de celui-ci.

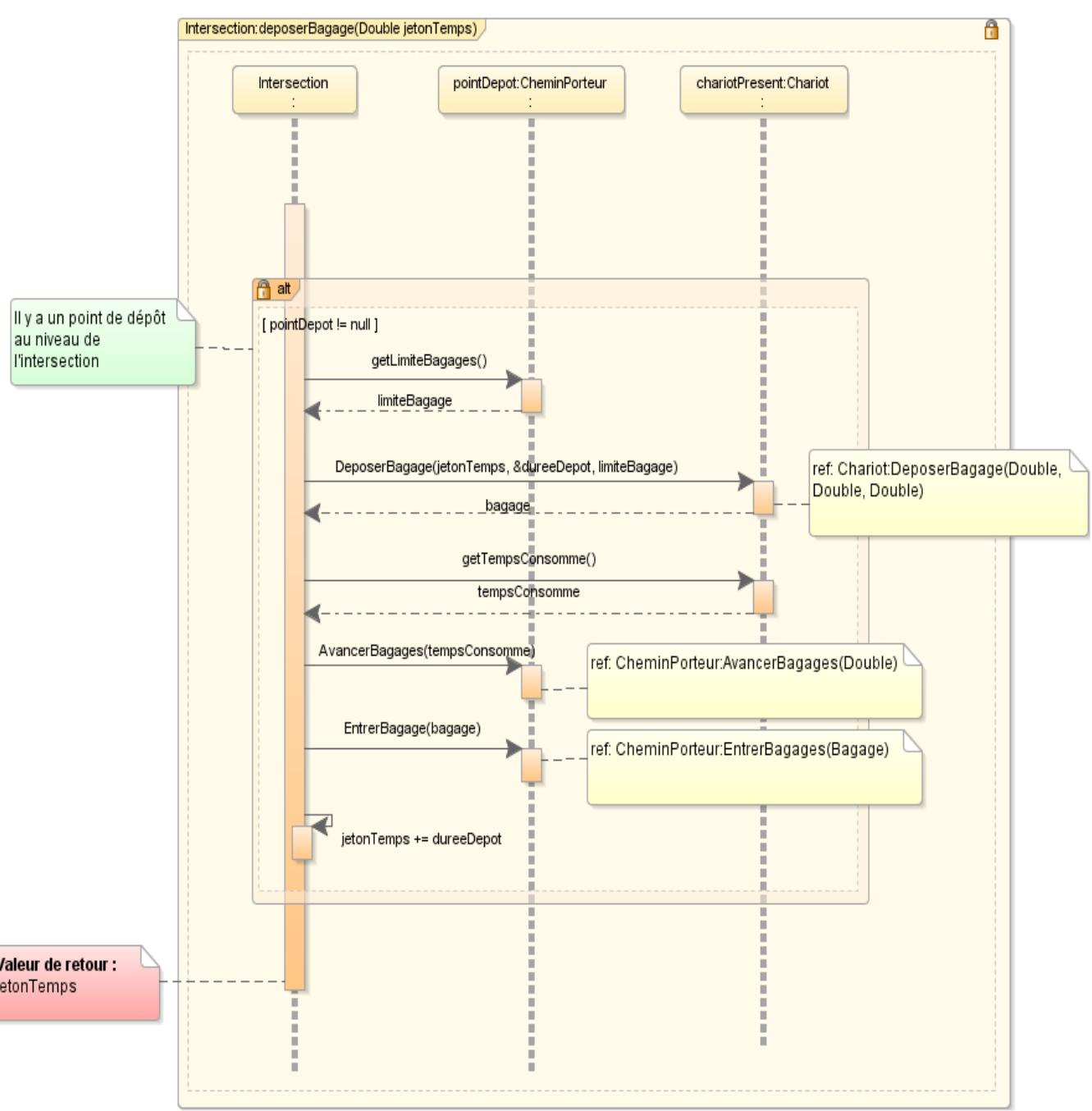


Diagramme de séquence DéposerBagage_Chariot

Ce diagramme de séquence présente le dépôt de bagage dans un toboggan pour un chariot.

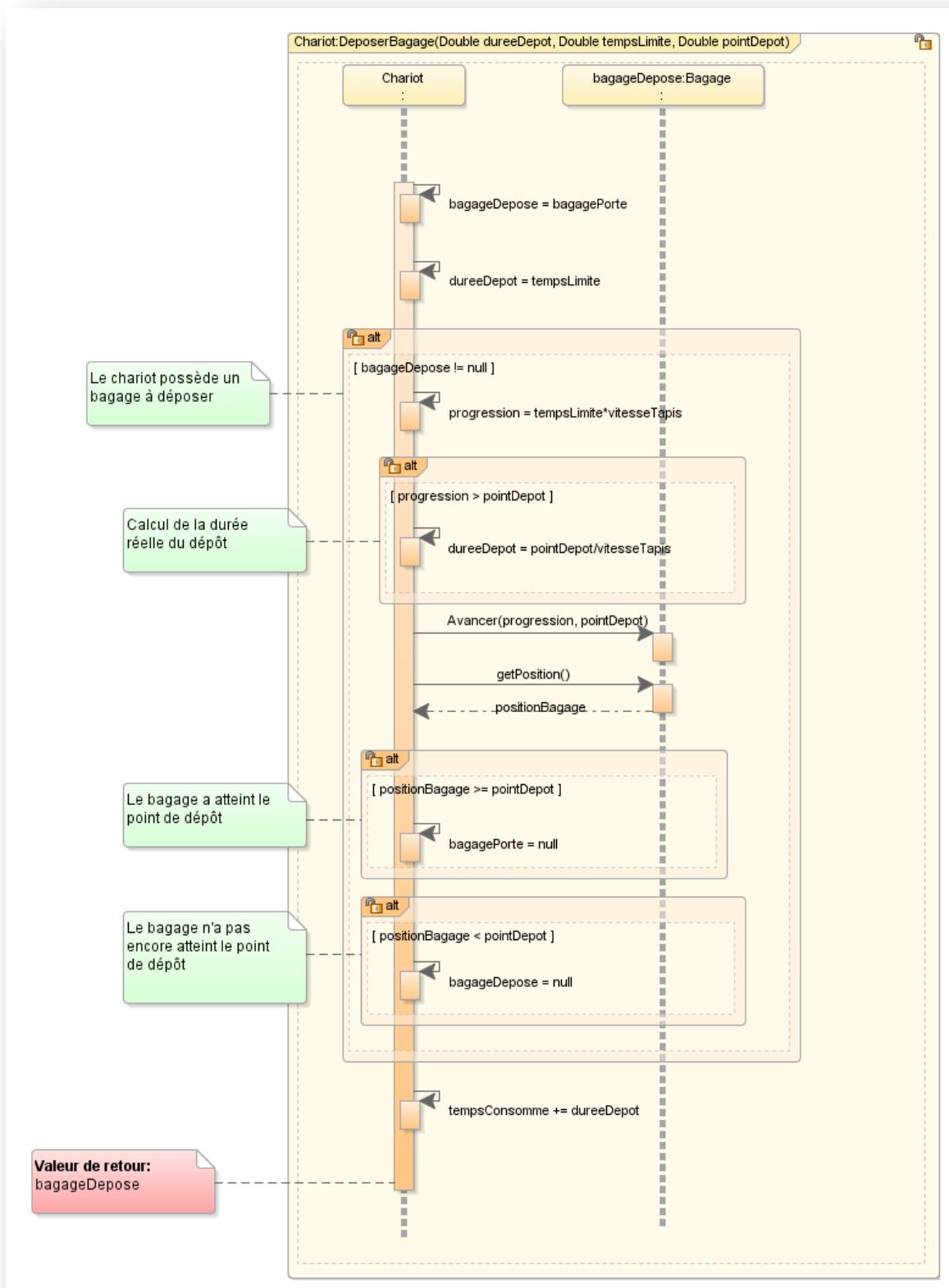


Diagramme de séquence RetirerBagage_Intersection

Ce diagramme de séquence présente le retrait de bagage d'un tapis roulant depuis une intersection correspondant à sa sortie.

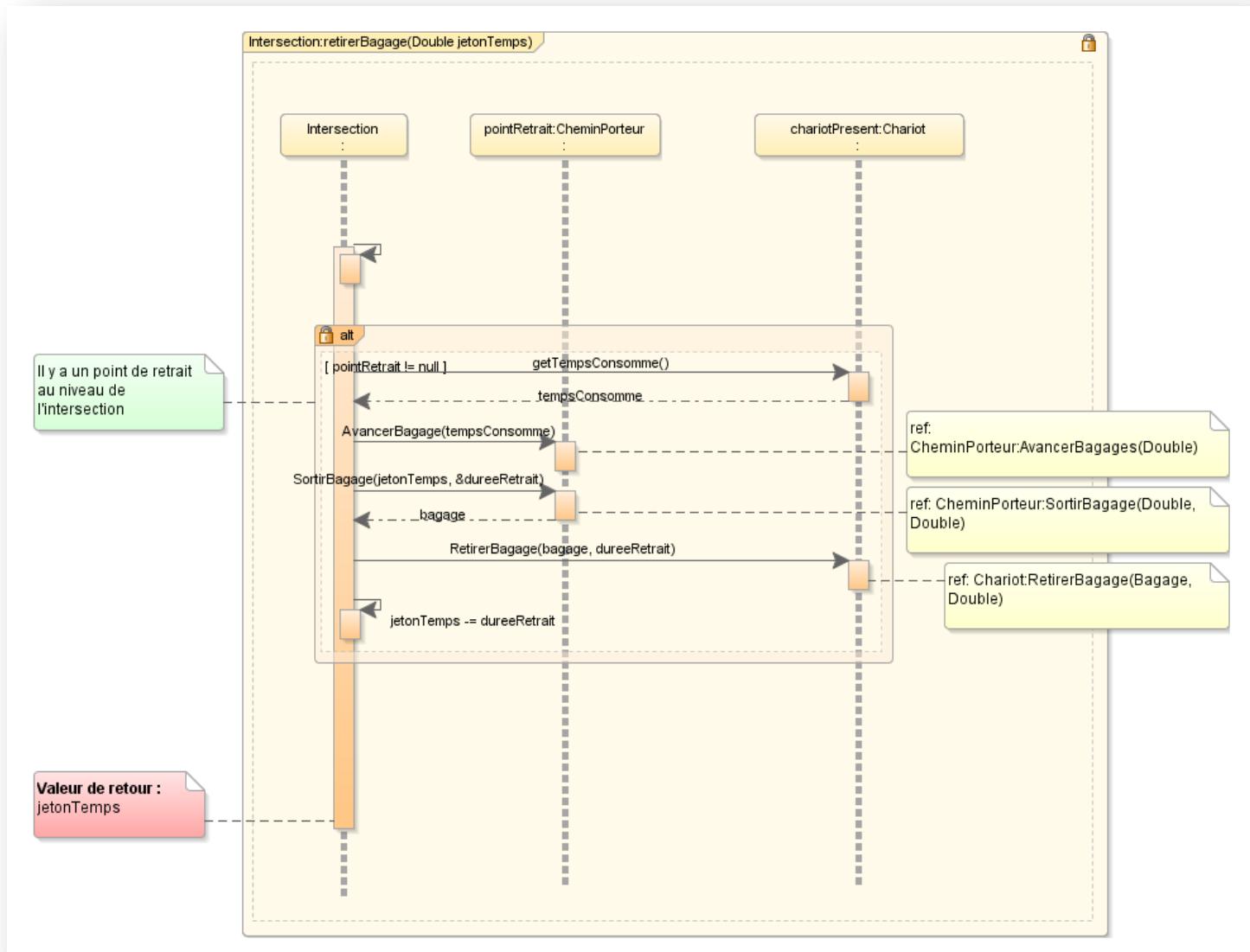
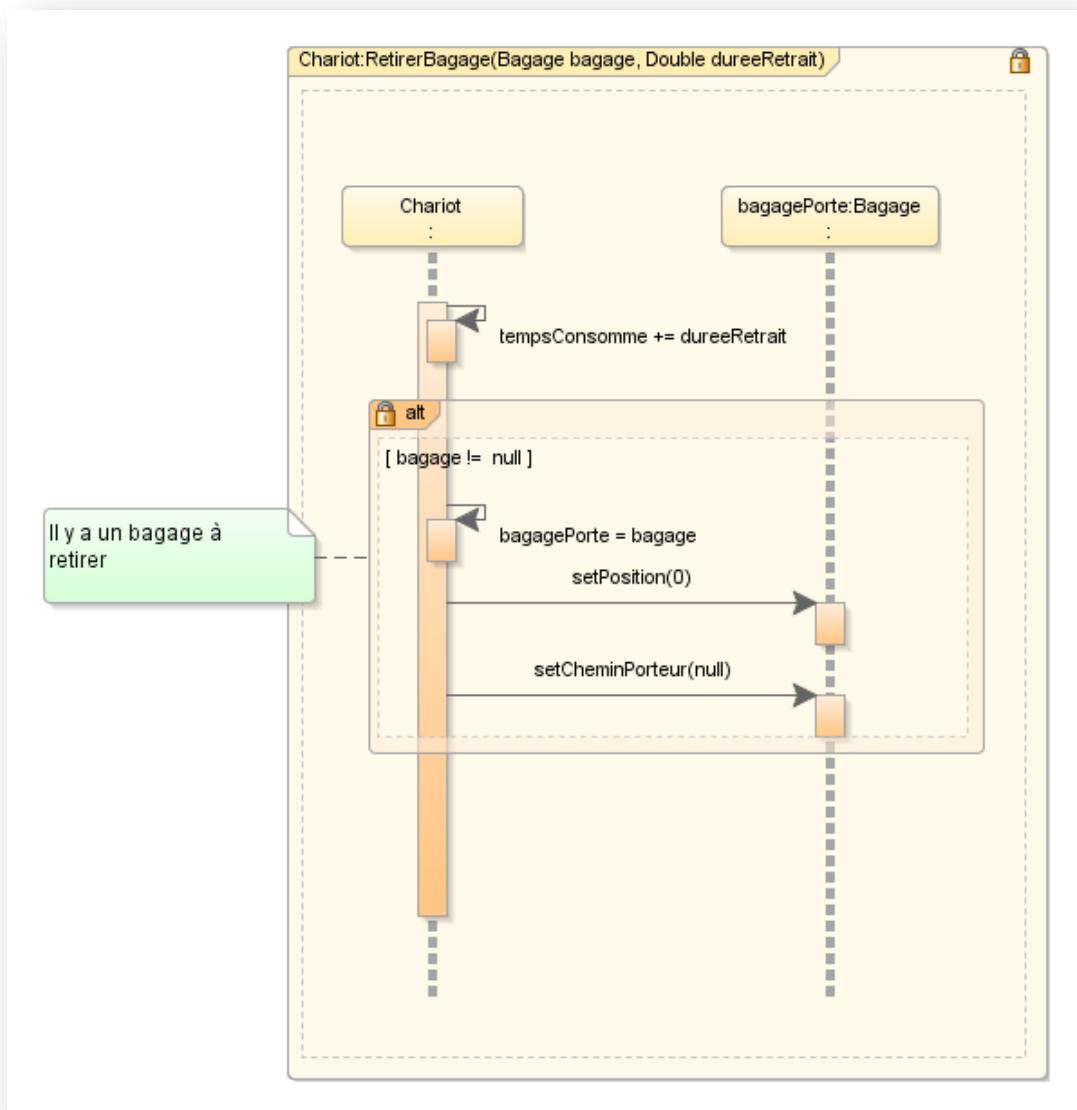
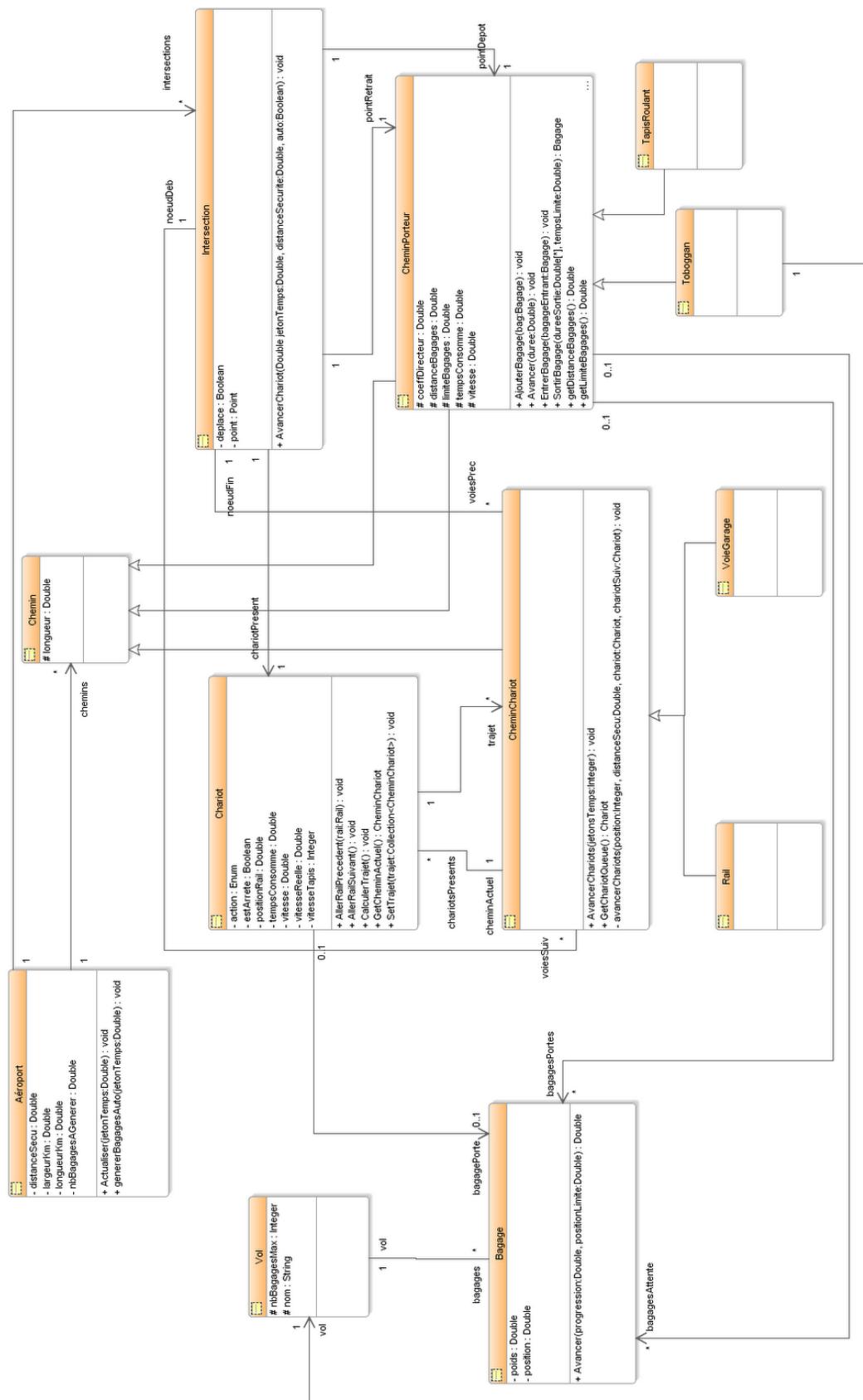


Diagramme de séquence RetirerBagage_Chariot

Ce diagramme de séquence présente le retrait de bagage d'un tapis roulant pour un chariot.





Conclusion

Grâce à l'élaboration des différents cas d'utilisations, nous savons maintenant le rôle de chaque utilisateur dans l'application. Nous avons pu également définir un découpage en paquetage de l'application complète, que l'on pourra réutiliser pour SGBag partiel.

Enfin, les diagrammes de séquences du top horloge, tout comme ceux de l'arrivée des bagages, nous ont permis d'appréhender avec plus d'assurance le fonctionnement du moteur ainsi que la gestion des événements (ajout manuel d'un bagage, etc.).



Construction

Introduction

Dans ce document de construction, étape aussi importante que lourde, nous nous attacherons à cerner et analyser les scénarios additionnels. Nous mettrons en place ensuite des diagrammes de séquences complets et détaillés afin de préparer au mieux le codage de l'application. Pour dresser le squelette de ce code nous établirons des diagrammes de classes complets et précis. Pour finir nous présenterons un diagramme de classe résultant d'un rétro-engineering du code de l'application.

Analyse

Au fur et à mesure de notre conception et développement de l'application, des scénarios imprévus dans nos diagrammes de séquence sont apparus. En plus des scénarios causés par une conception encore imprécise, des fonctionnalités supplémentaires ont été pensé pour le SGBag, tels que le chargement et la sauvegarde d'une configuration à partir d'un fichier XML. (cf. conception XML)

Conception XML

Afin de développer une application performante et réaliste, notre équipe a été heurtée au problème de la persistance des données. Il fallait trouver une solution afin de pouvoir gérer et exploiter les données de simulations.

Pour cela notre application sera capable de sauvegarder et charger une simulation ainsi qu'une configuration prédéterminée à partir d'un fichier XML choisi par l'utilisateur.

Chargement de la configuration :

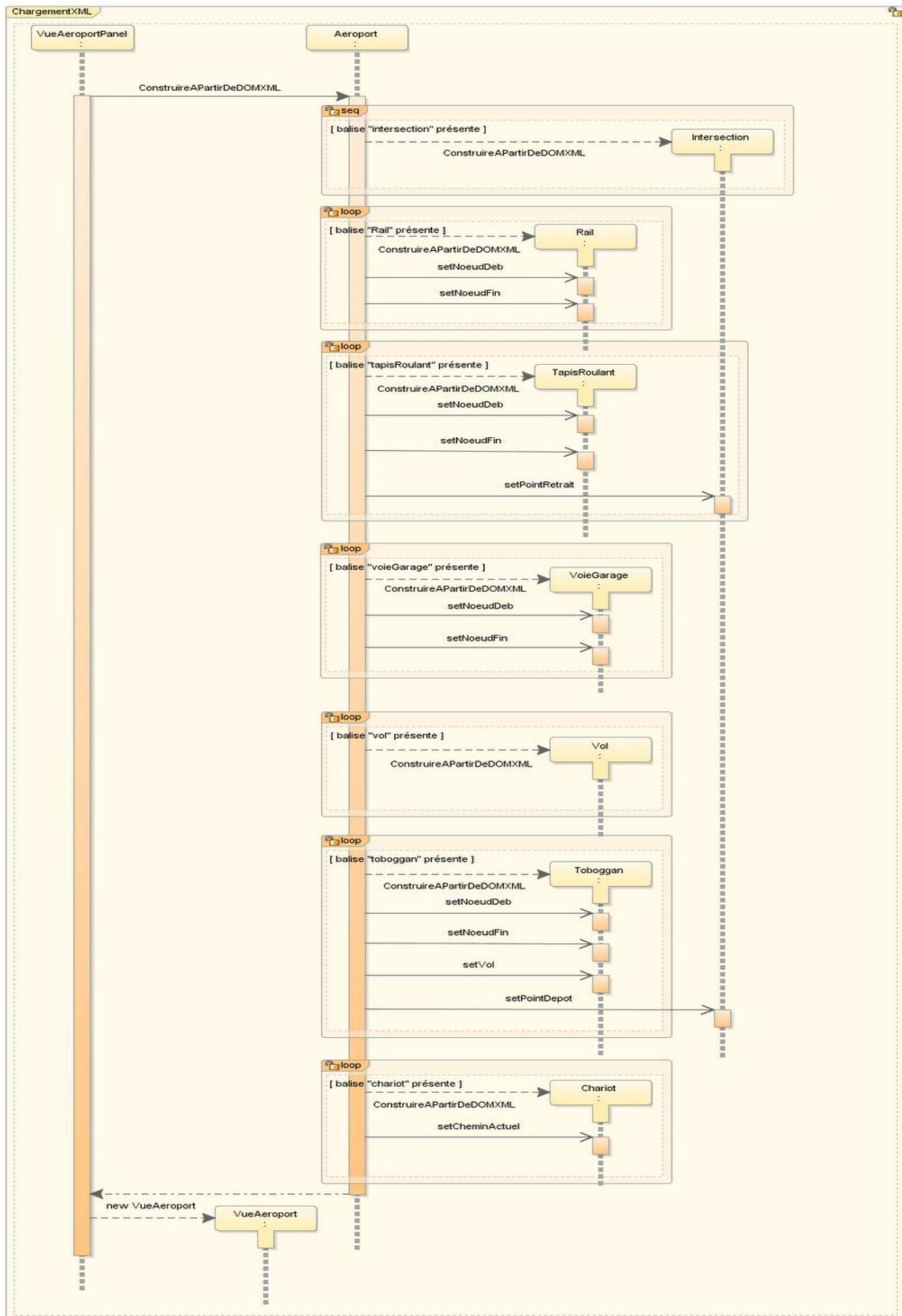
Lors du chargement de la configuration du fichier XML, l'application va créer chaque élément (du noyau) nécessaire au bon fonctionnement, puis va instancier les vues associées.

Remarque :

Nous rappelons que l'instanciation du noyau et des vues associées sont totalement indépendantes. Différents modèles de vues pourraient être rattachés au noyau ainsi recréé par persistance XML.

L'ordre de création des éléments à une incidence directe sur le bon fonctionnement de la création. L'enchaînement des objets est détaillé dans le diagramme de séquence "Chargement XML" ci-dessous.





Quant au formalisme de notre fichier XML, en voici un exemple :

```
<noyau.Aeroport id="0" largeur="16" longueur="25" distanceSecu="1.7" lambdaPoisson="1">
    <noyau.Intersection id="1" x="12" y="2" />
    <noyau.Intersection id="2" x="17" y="2" />
    <noyau.Intersection id="3" x="17" y="7" />

    <noyau.Rail id="4" idDepart="1" idArrivee="2" />
    <noyau.Rail id="5" idDepart="2" idArrivee="3" />

    ...

```

Chaque objet possède une balise du type : *package.nomClass*.

Chacune de ces balises stocke les attributs nécessaires à sa création (ex : id, caractéristiques spécifiques, ...)

Chargement de la simulation :

Ici nous avons le même fonctionnement que pour le chargement de la configuration. Le principe est le suivant : lorsque l'utilisateur choisit le fichier XML à charger, l'application charge dans un premier temps la configuration du fichier, puis les paramètres spécifiques à la simulation (i.e. les bagages et les positions des bagages, le temps restant de la simulation, ...).

Le choix de charger la configuration en même temps que la simulation est un choix logique. En effet, si l'utilisateur choisit de charger un fichier particulier en mode simulation, il n'a aucun intérêt à vouloir charger uniquement la simulation et non la configuration.

De toute manière l'application ne permettrait pas de charger une configuration et une simulation totalement indépendantes l'une de l'autre : positions et objets différents, ...

De plus, cela évite de charger une configuration puis une simulation, qui au final ont leur description au sein du même fichier XML.

Sauvegarde de la simulation (configuration) :

Afin d'assurer la persistance des données, notre application est dotée de la capacité à écrire un fichier XML stockant chaque donnée importante de la simulation en cours, afin que dans un futur plus ou moins proche cette même simulation puisse être reprise où elle s'est arrêtée.

Conception

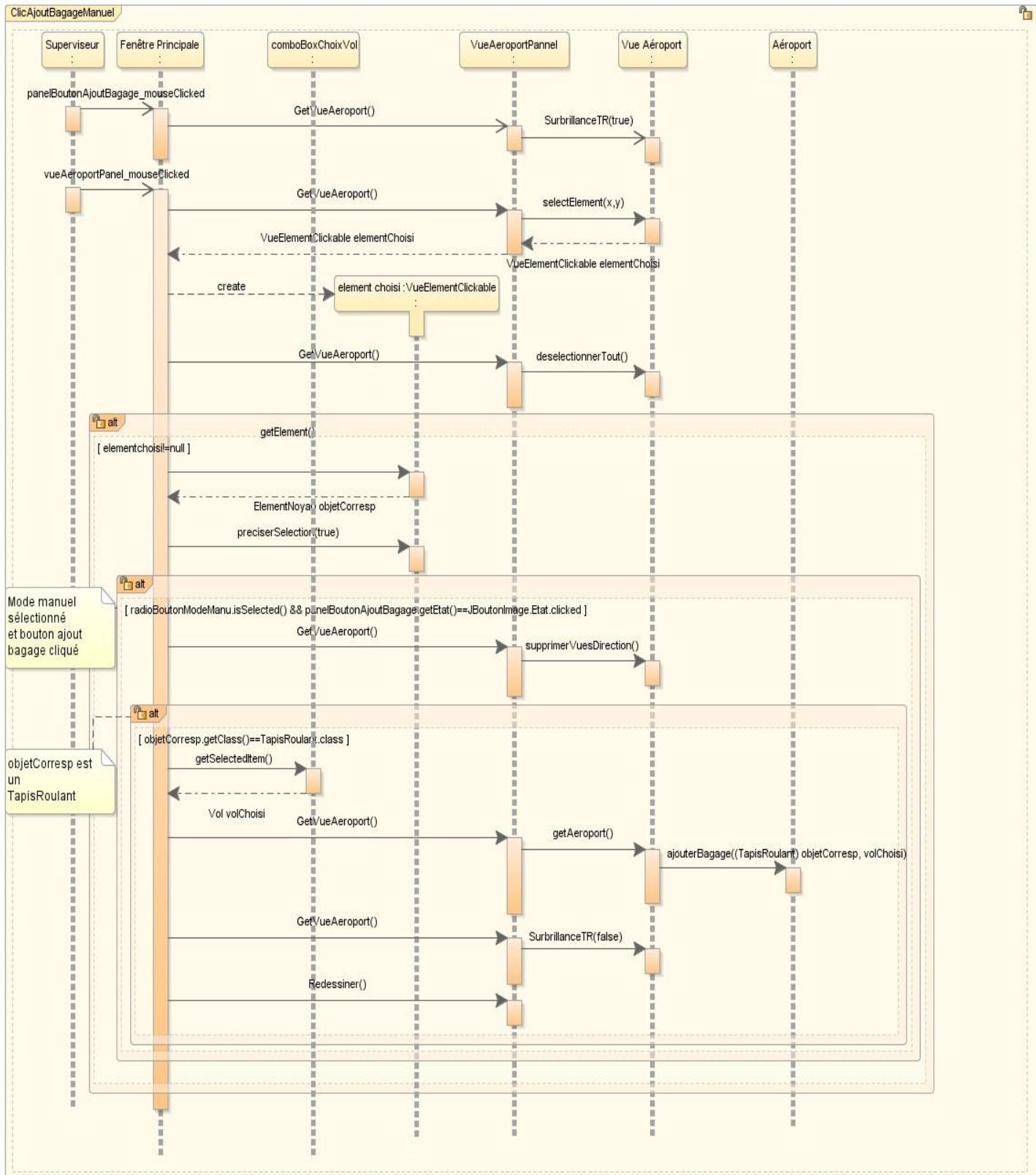
Lors de notre conception du noyau, il a aussi fallu commencer à réfléchir sur l'architecture du module de vue permettant l'affichage des éléments dans un cadre graphique. Nous avons alors travaillé sur une architecture permettant au mieux de séparer les vues du noyau, pour des soucis de maintenabilité et de cohérence. En effet il est important de dissocier ce qui, au sein d'une classe, est utilisé pour le fonctionnement de l'aéroport, et pour l'affichage de la vue dans l'aéroport.

Diagramme de séquence

Le diagramme ci-dessous transcrit le scénario d'ajout de bagage manuel, effectué par le superviseur.

Les fonctions *SurbrillanceTR* et *selectElement* sont détaillées ci-après, afin de ne pas surcharger la compréhension du diagramme.



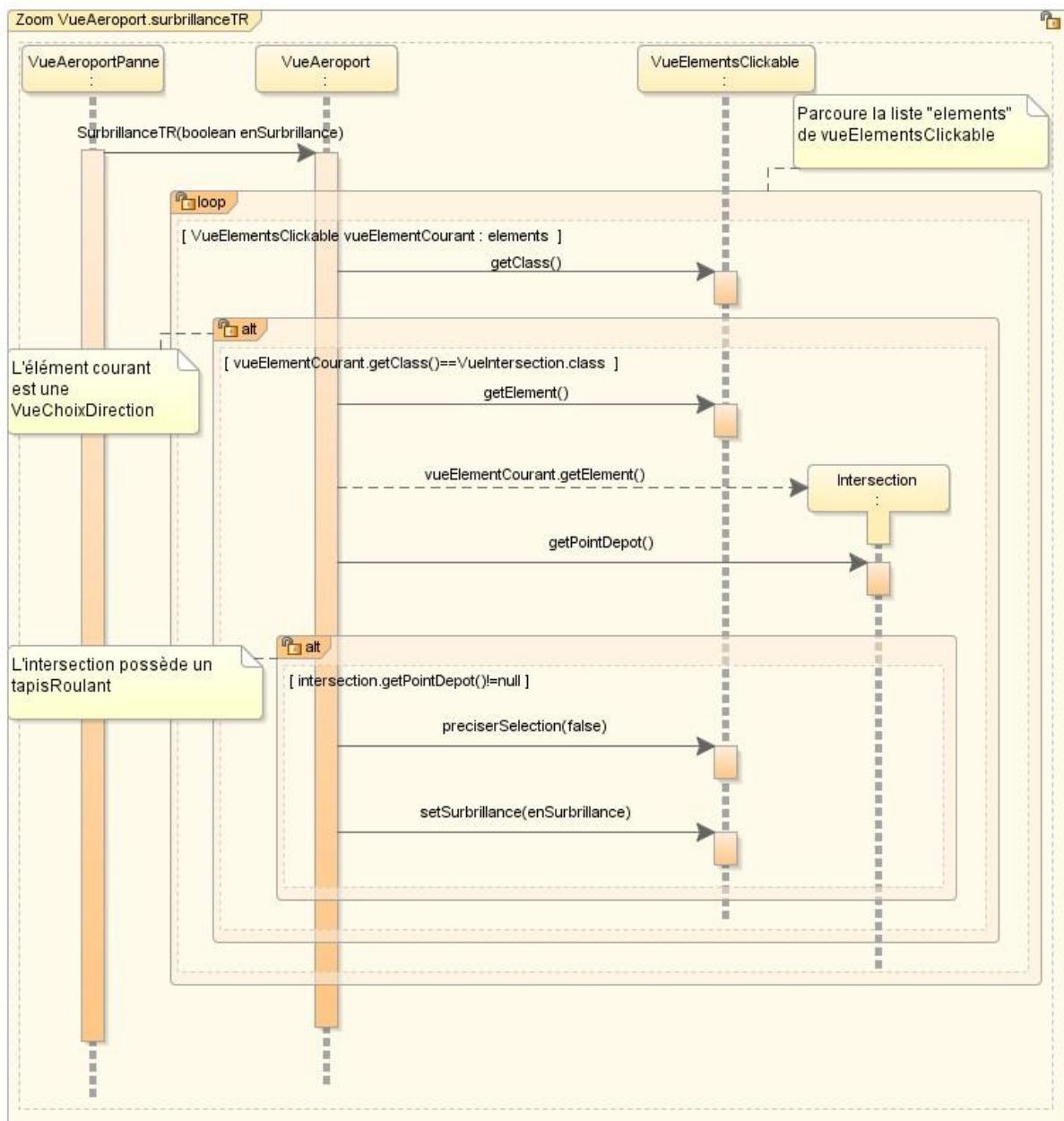


Enfin, les autres diagrammes de séquence de la gestion en mode manuel sont présentés : clic sur un chariot, choix d'une direction/action, et clic sur une voie de garage.

Fonction SurbrillanceTR



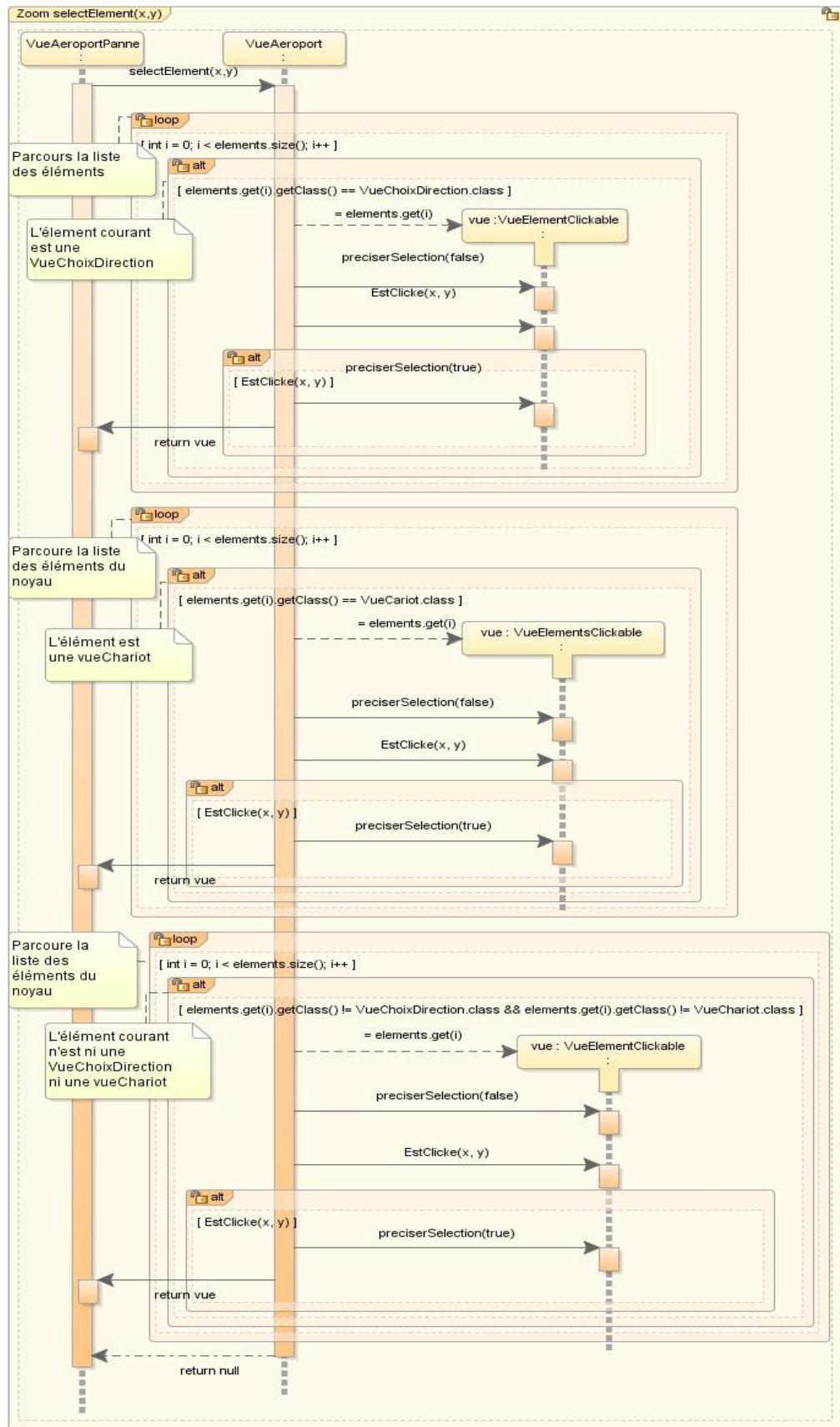
Les tapis roulants sont mis en surbrillance



Fonction `selectElement`

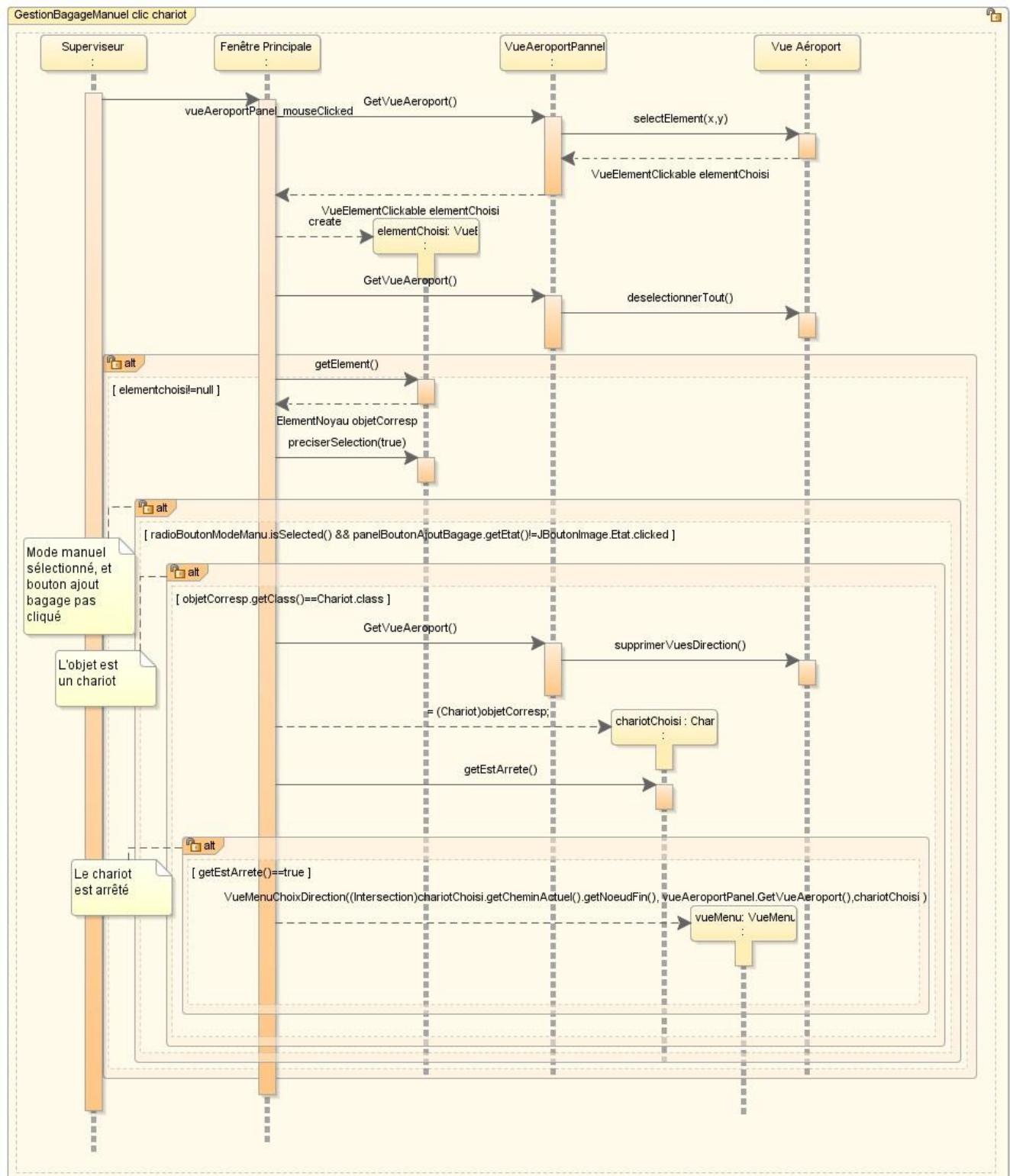
`vueAeroport` recherche parmi ses éléments celui qui a été cliqué par le superviseur, et le retourne.





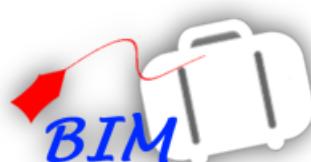
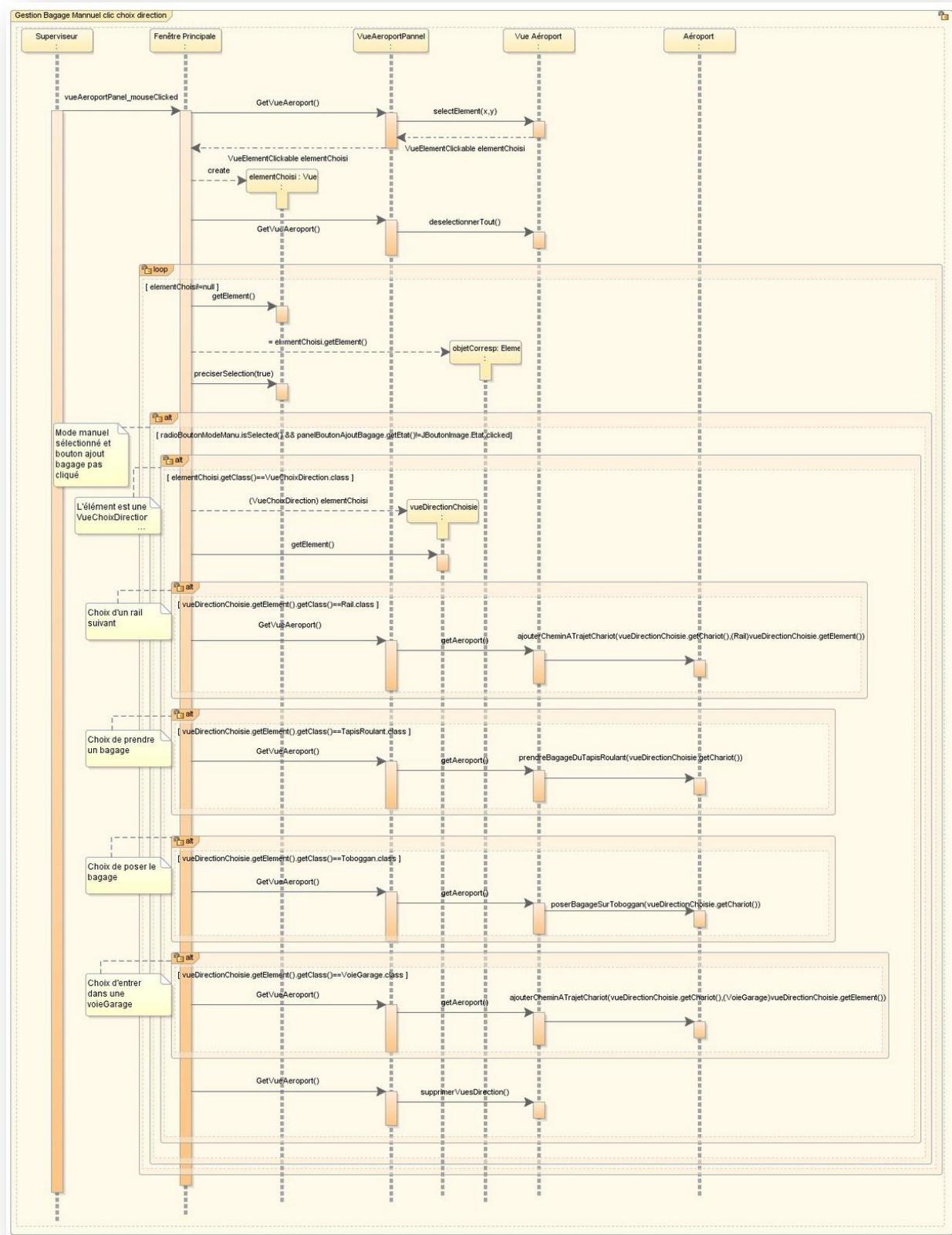
Clic sur un chariot

Affichage des options de direction/action : rail et/ou poser bagage et/ou prendre bagage.



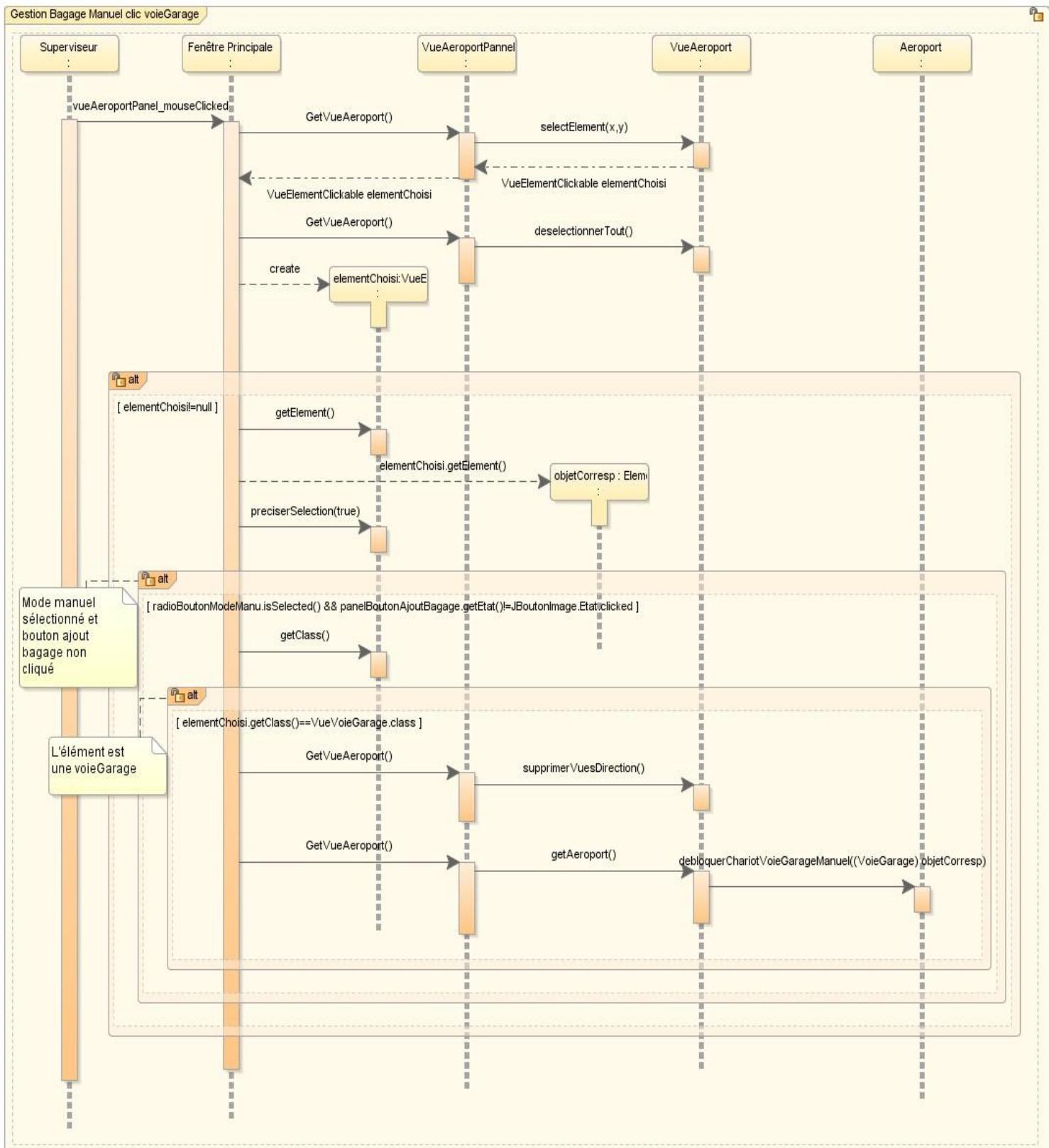
Clic sur un choix de direction/action

Sélection d'une direction ou action par l'utilisateur.



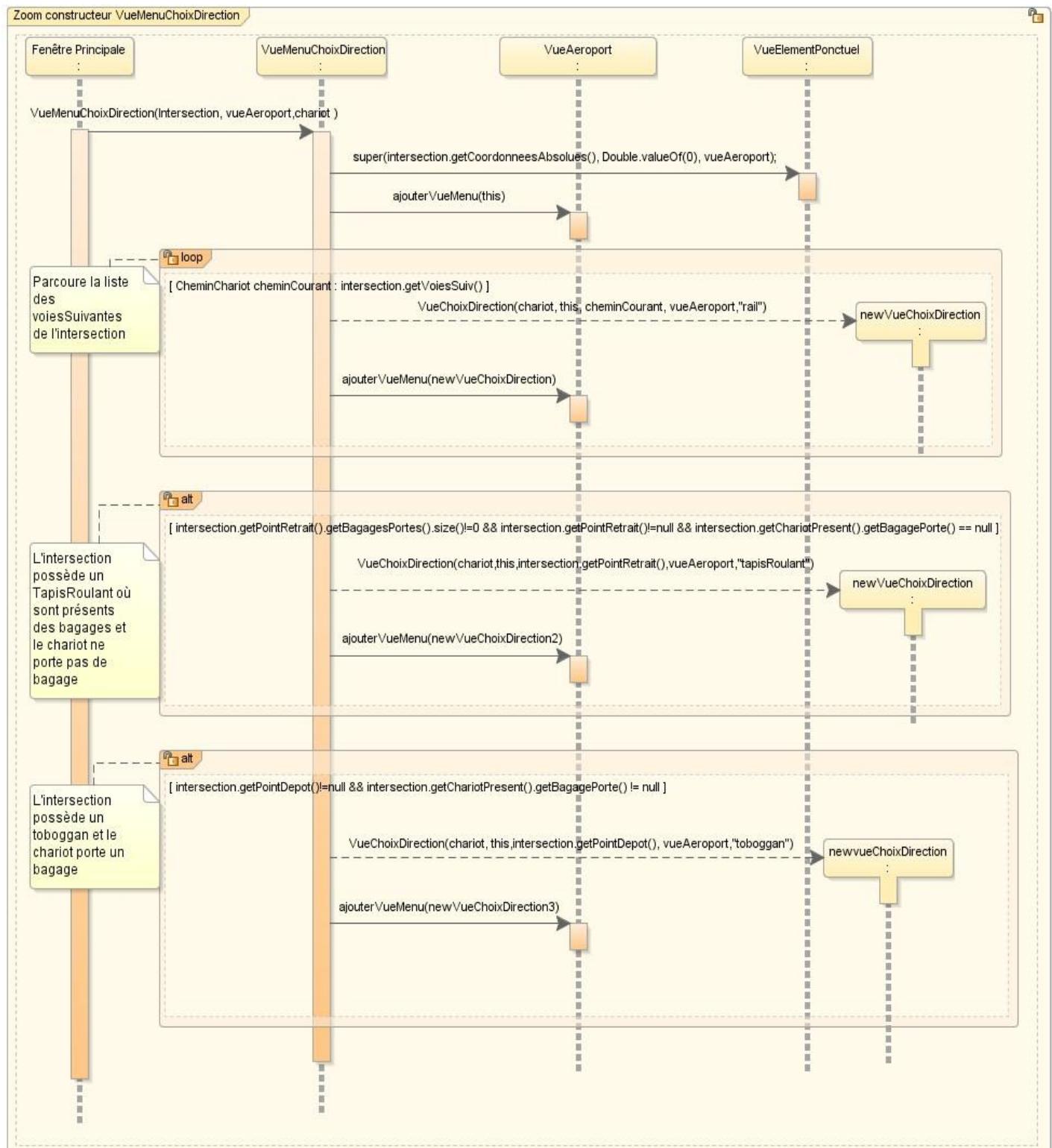
Clic sur une voie de garage

Déblocage d'un chariot en voie de garage.



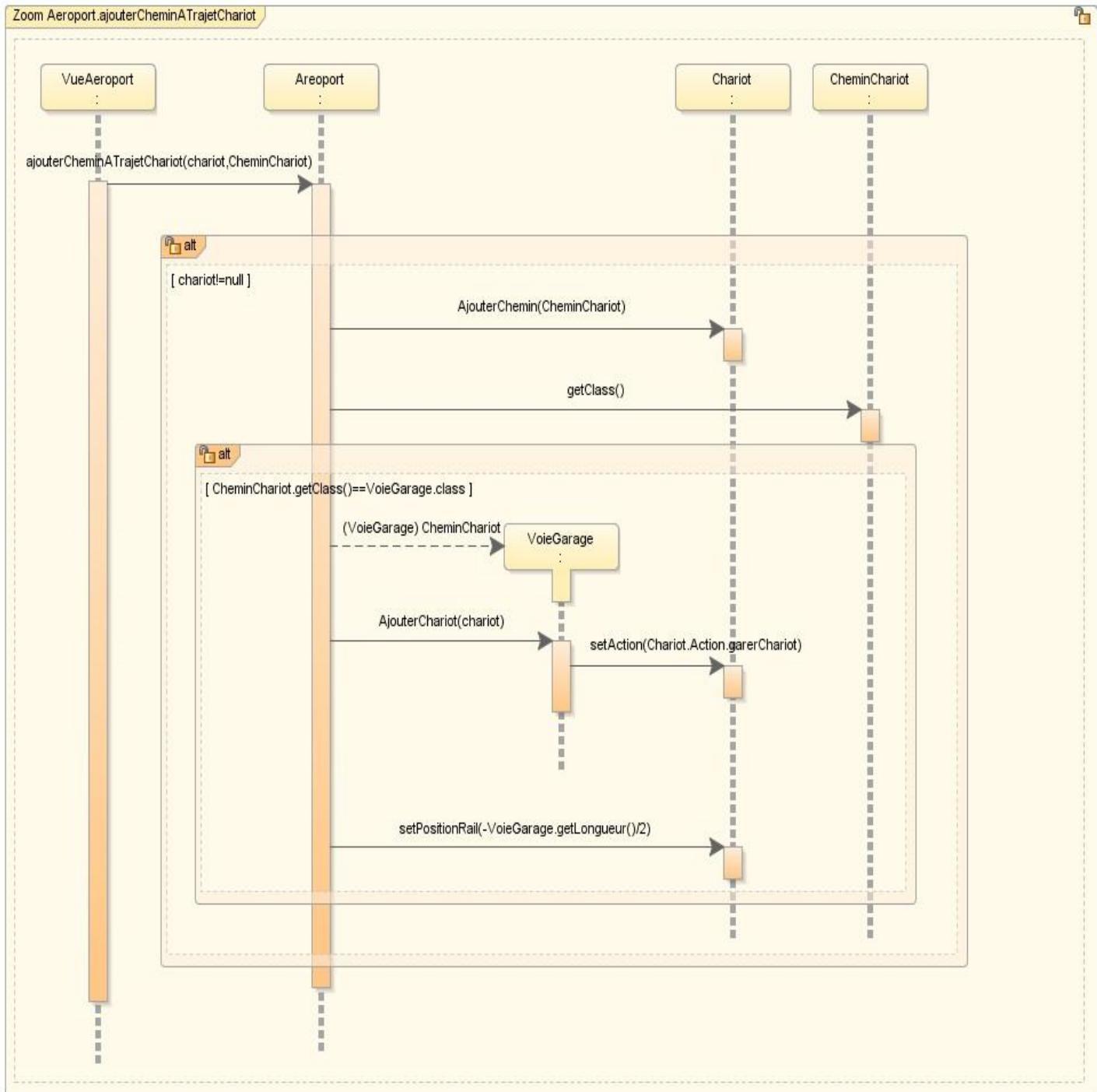
Constructeur de VueMenuChoixDirection

Création des VueChoixDirection correspondant aux choix possibles : rail et/ou toboggan et/ou tapis roulant.

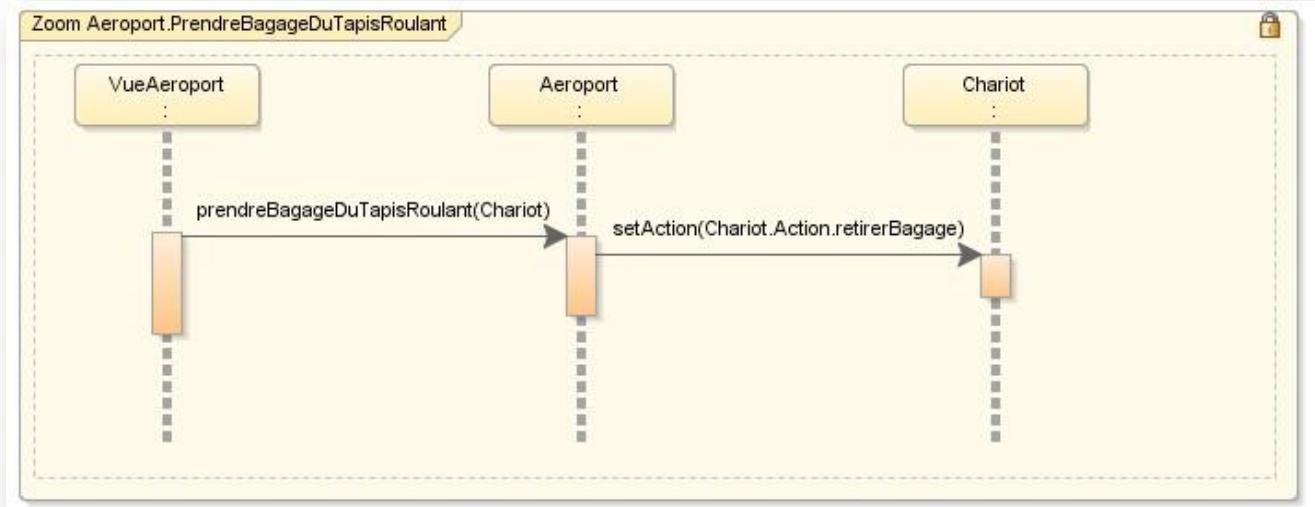


Fonction ajouterCheminATrajetChariot

Ajoute un chemin (rail ou voie de garage) au trajet suivi par un chariot.



Fonction PrendreBagageDuTapisRoulant



Fonction PoserBagageSurToboggan

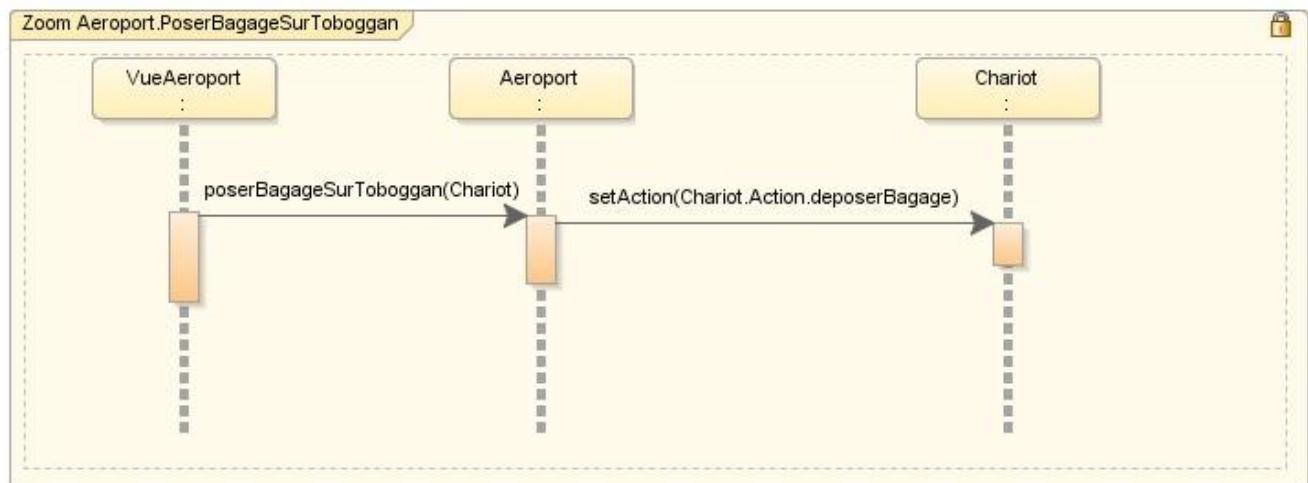
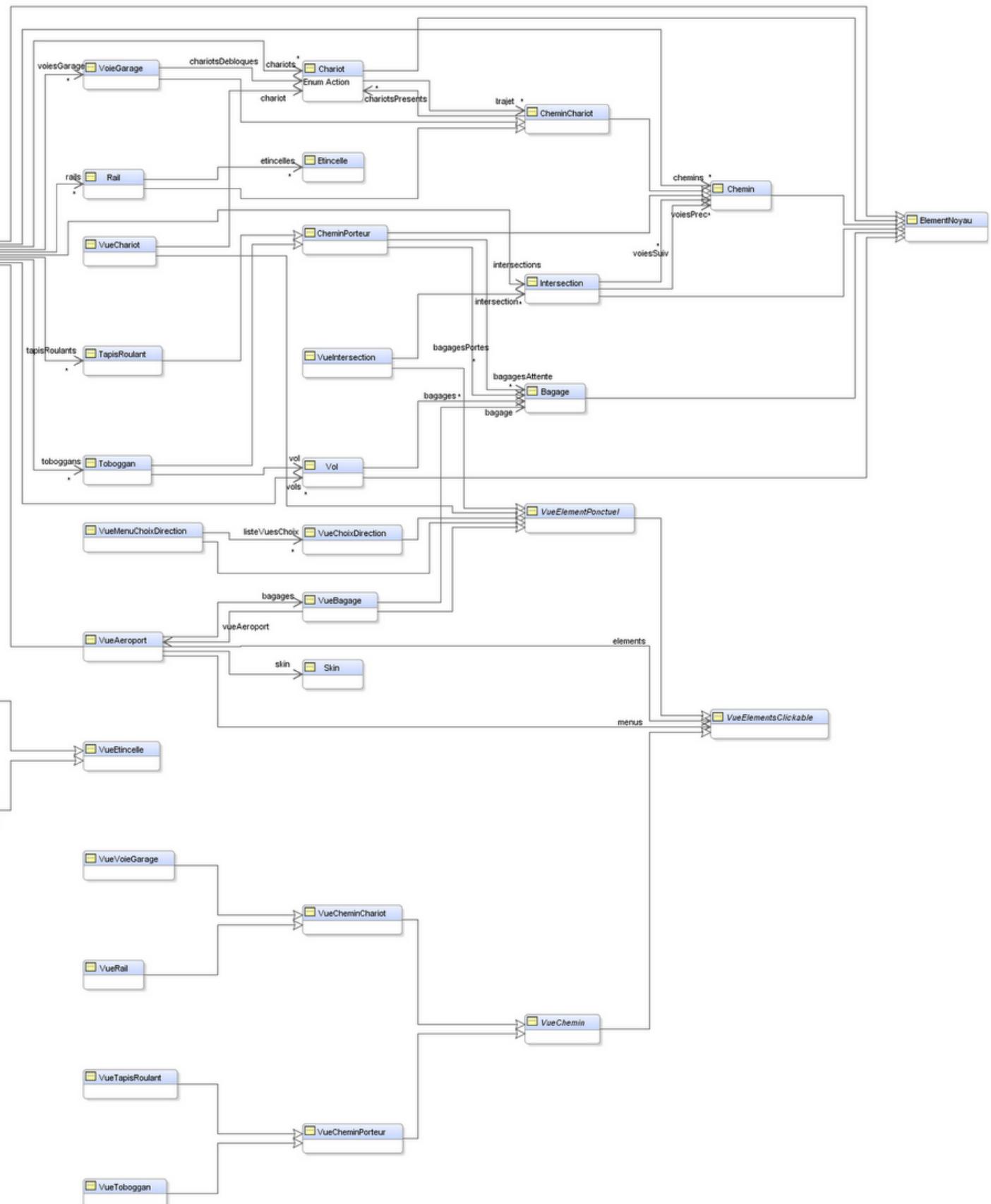
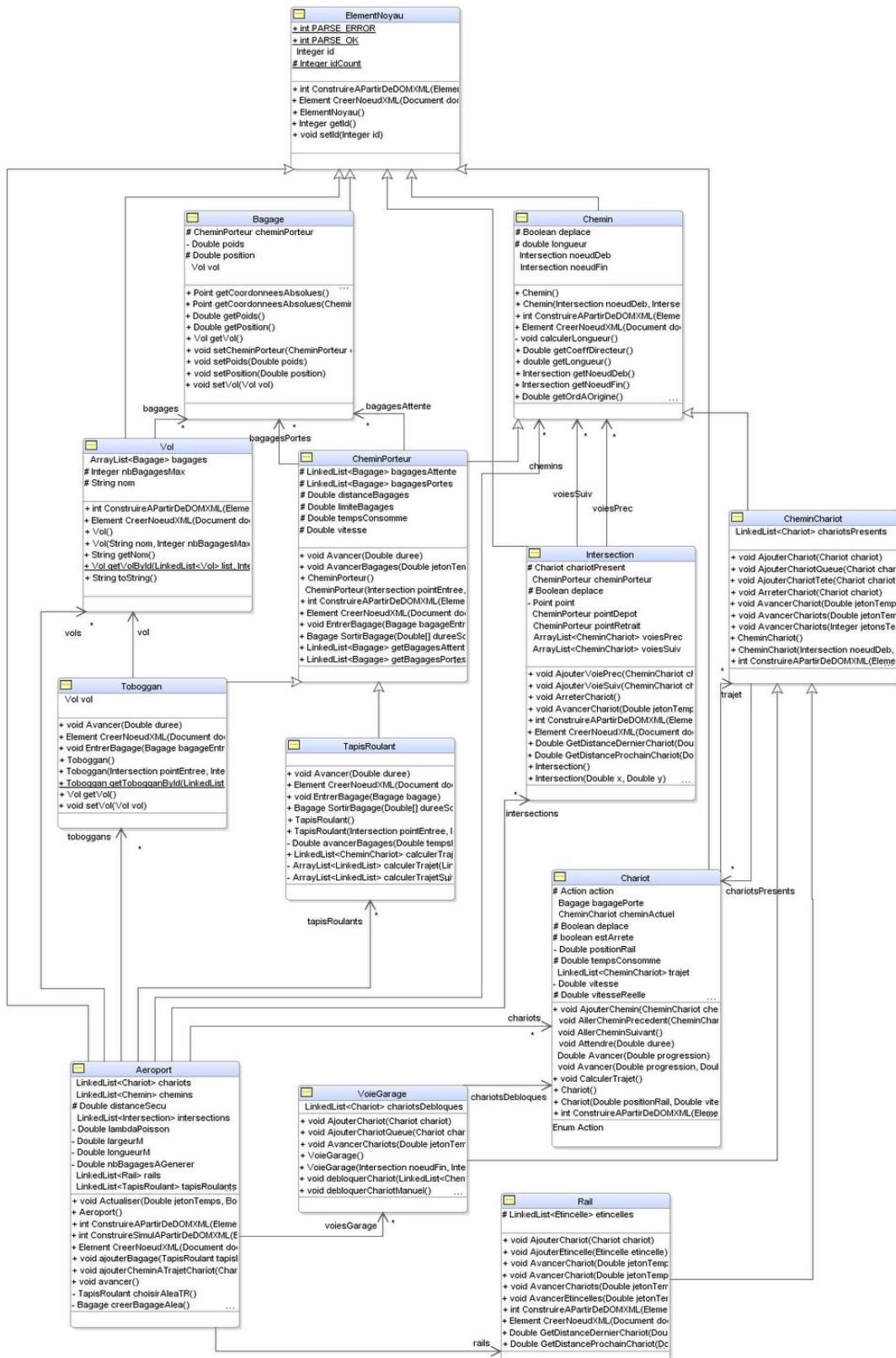


Diagramme de classes Vues + Noyau (classes vides mais avec liaison)



Rétro-engineering du code vers le diagramme complet en fin de projet

Le diagramme suivant (page suivante) représente la rétro-engineering du noyau. Il présente les classes, leurs attributs et méthodes, les relations d'héritage ...



Réalisation

L'ensemble des fichiers de codes, ainsi que tout document utile au fonctionnement de l'application (fichiers XML et images) sont joints à ce dossier.

S'y trouve également la Javadoc structurée, extraite de l'ensemble des commentaires.

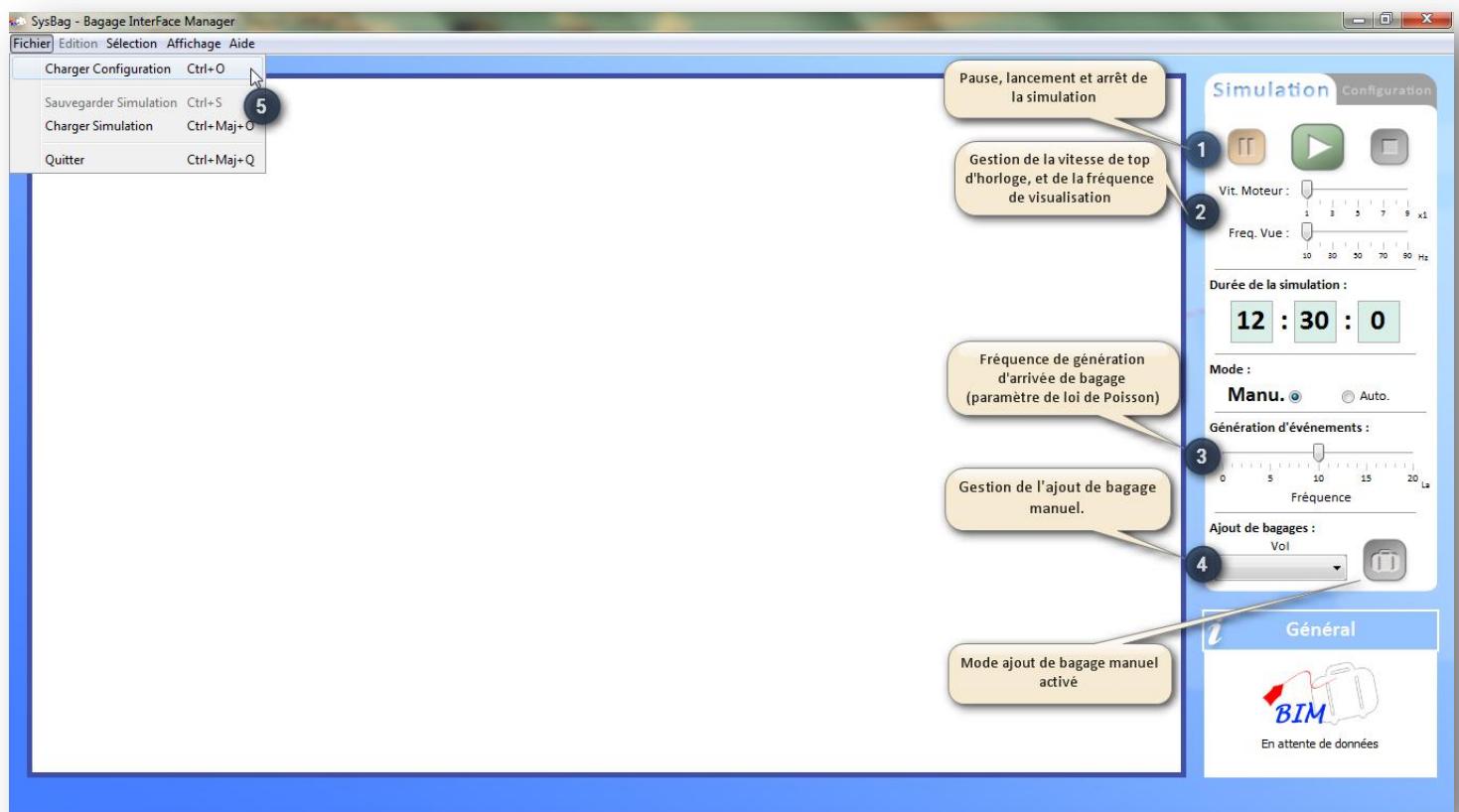
Sera présenté ici un extrait de l'IHM montrant sommairement les fonctionnalités de l'application.

Enfin, divers fichiers XML de configuration et un fichier de sauvegarde sont donnés, afin de pouvoir tester le fonctionnement.

Présentation de l'IHM

Remarque :

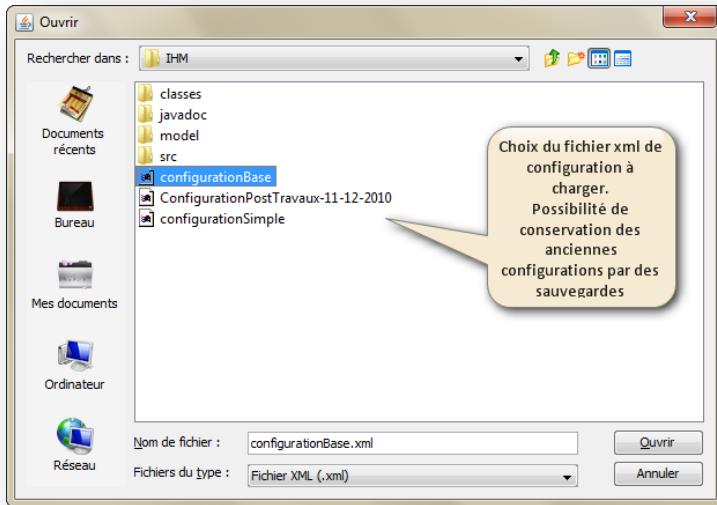
La plupart des explications et commentaires est présente dans les images.



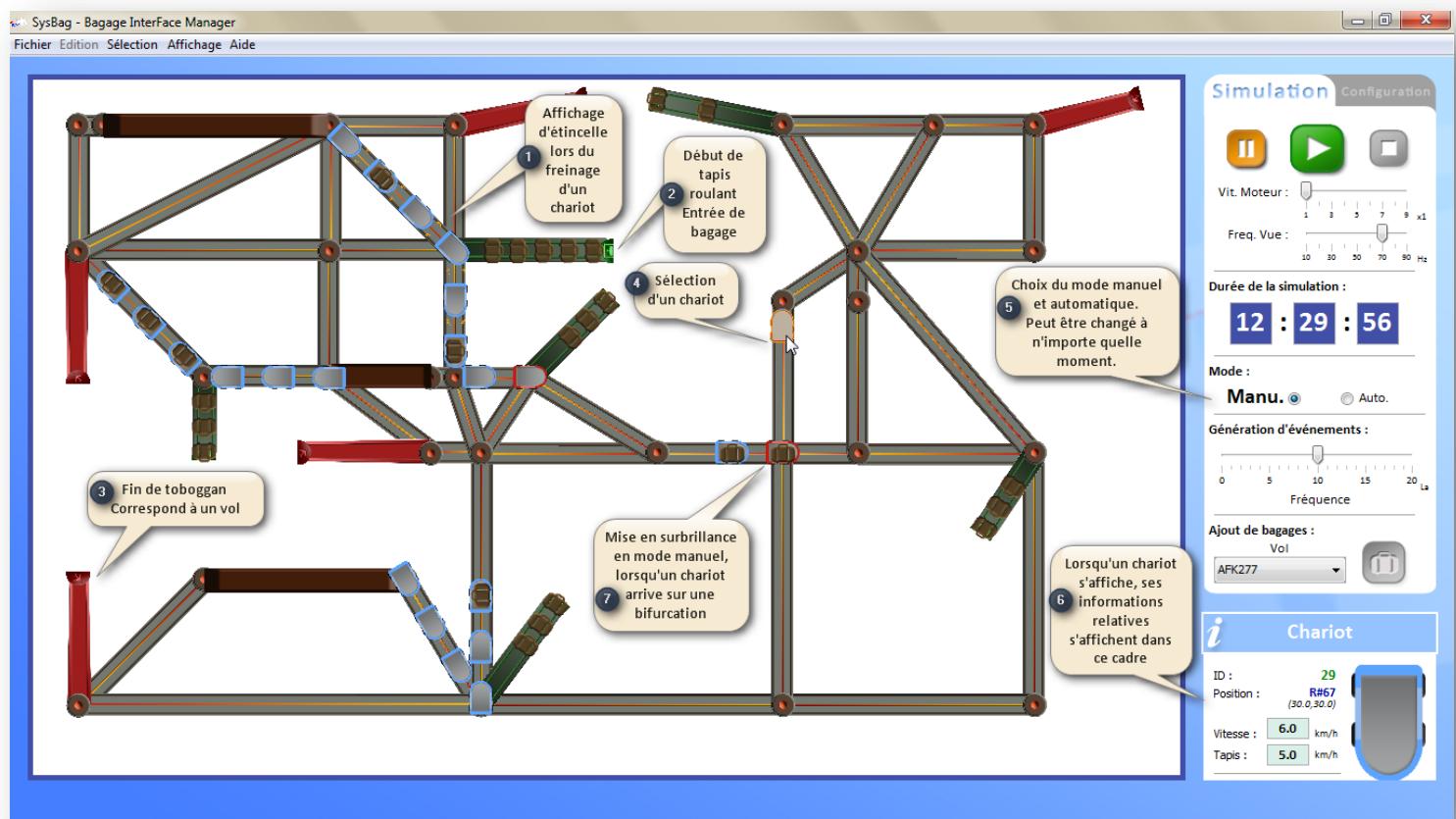
Capture 1

Fenêtre principale de l'application



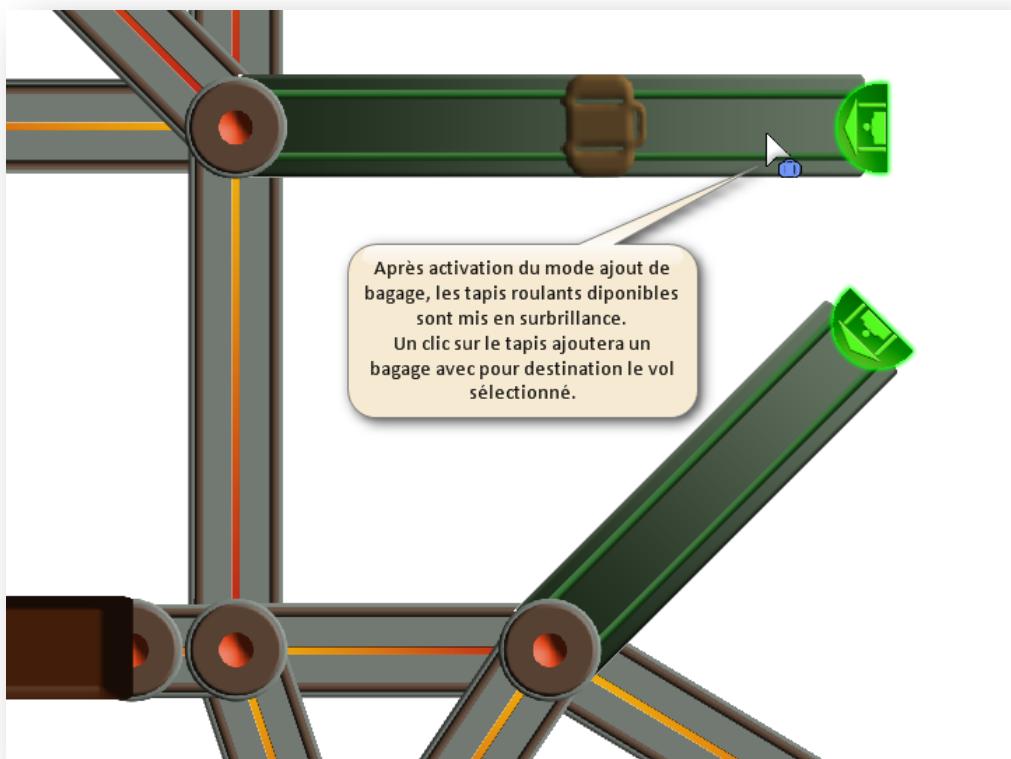


Capture 2
Choix du fichier XML dans le chargement de configuration

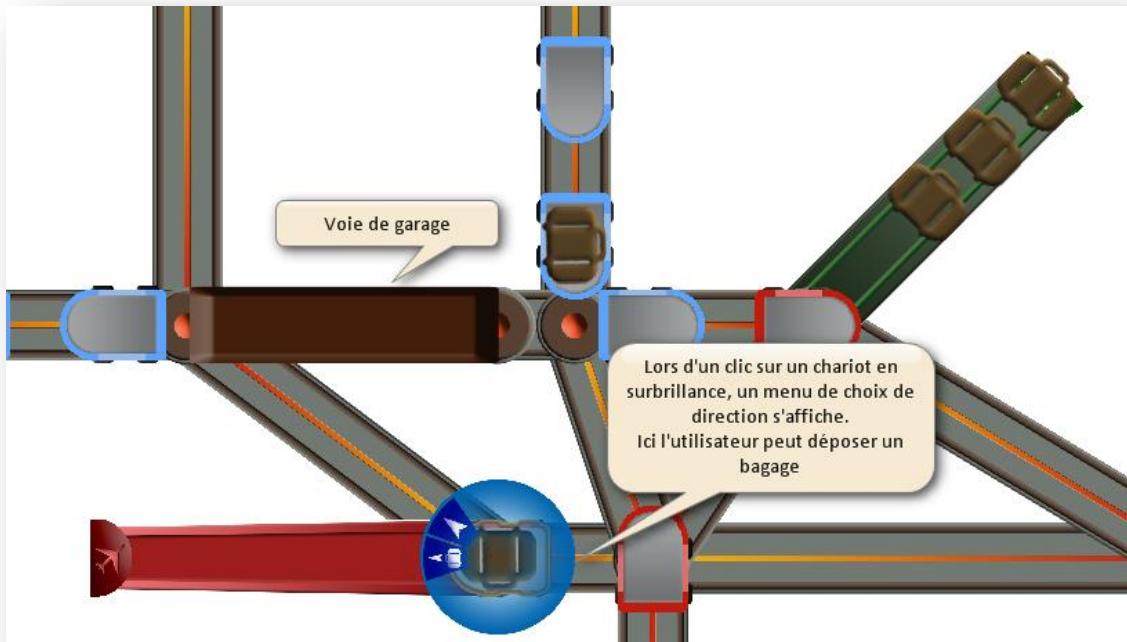


Capture 3
Présentation de l'affichage de l'aéroport, du cadre d'information (en bas à droite) et du système de mise en couleur et surbrillance sur les chariots.

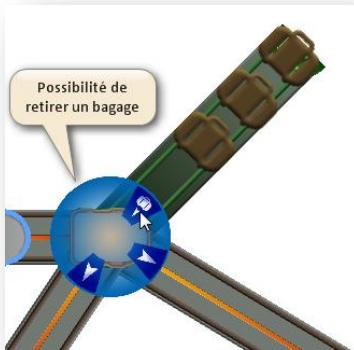


**Capture 4**

Ajout d'un bagage manuellement

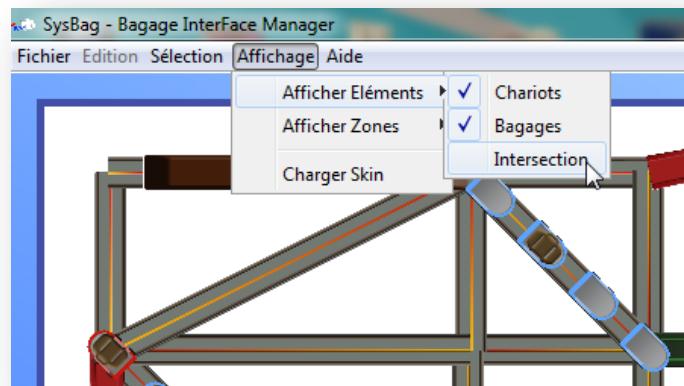
**Capture 5**

Choix de direction lorsqu'un chariot arrive sur une bifurcation



Capture 6
Retrait de bagage sur un tapis roulant

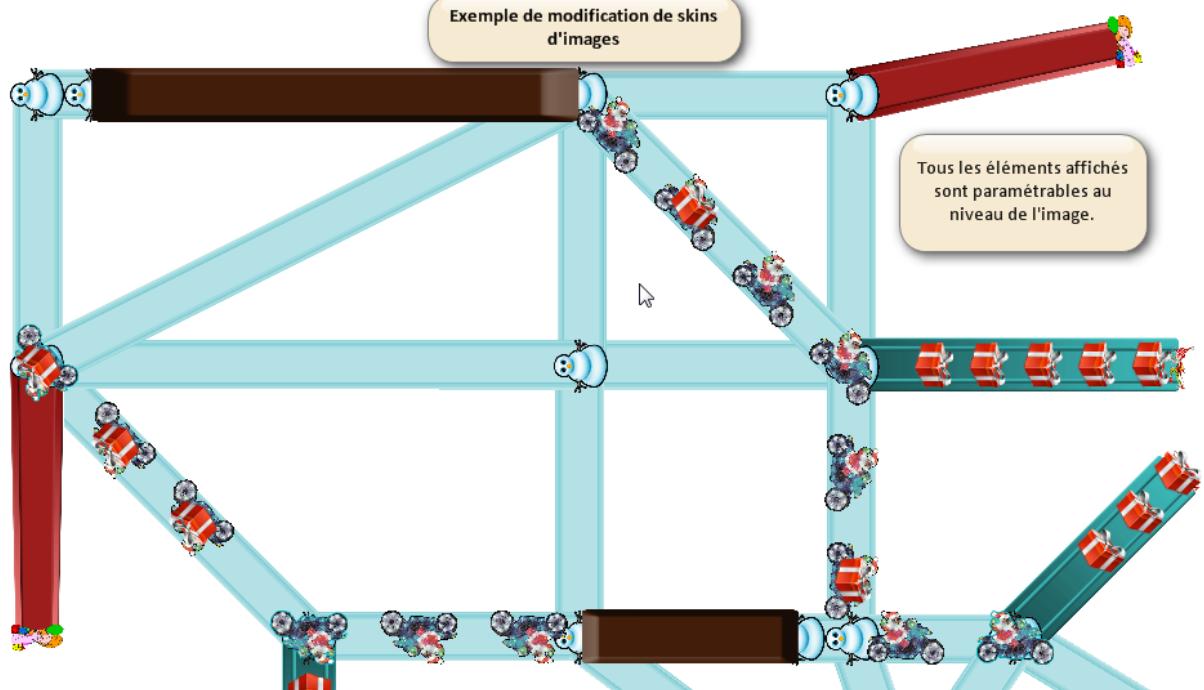
Capture 7
Possibilité d'afficher/de masquer des éléments ou des zones
Les intersections sont actuellement masquées.



Toboggan	Tapis-Roulant
ID : 192 Vitesse : 5.0 Longueur : 5.0 Vol : AFK277 En transfert : 0	ID : 119 Vitesse : 5.0 Longueur : 4.242640... En transfert : 3 En contrôle : 20
Voie de Garage	Chariot
ID : 240 Chariots : 9	ID : 29 Position : R#11 (17.0,17.0) Vitesse : 6.0 km/h Tapis : 5.0 km/h

Capture 8
Dans le cadre d'information, affichage des informations relatives à un élément après un clic sur celui-ci.





Capture 9

Les vues sont modulables, il est possible de modifier les skins des images afin de mettre en valeur des éléments plus que d'autre.

Transition

Introduction

Correspondant normalement à la fin d'un cycle USDP, l'étape de transition permet un retour sur le travail effectué, afin de préparer le prochain cycle en laissant une trace suffisamment explicite pour les clients et les prochaines équipes (le cycle suivant pourrait très bien être réalisé par des individus différents).

Ayant clairement explicité notre élaboration et conception dans les chapitres précédents, nous ne reviendrons, dans cette partie, que sur la répartition des tâches et la charge de travail réelles, avant de conclure par un bilan critique de notre projet.

Charge de travail

Au fur et à mesure de l'avancée du projet, notre intérêt pour ce sujet a grandi, et donc nos exigences avec. Nous n'avons ainsi pas hésité à nous investir plus qu'initialement prévu, dans l'optique d'atteindre un résultat le plus professionnel et complet possible.

Vous trouverez donc ci-dessous la répartition finale des tâches sur l'ensemble du projet, ainsi qu'un bilan récapitulatif sur ce sujet.

Répartition des activités

Tout au long du projet, nous avons donc continué à définir et répartir les tâches (élaboration, conception, intégration, correction, etc.) à l'aide de notre plateforme Redmine.

Grâce à ses outils, nous avons ainsi pu générer un diagramme calendaire final dévoilant le planning appliqué, ainsi qu'un tableau de répartition des activités (demandes).

Vous trouverez ces deux documents joints à ce dossier.

Bilan

Bien que nous ayons dépassé les prévisions initiales de charge de travail, nous sommes globalement satisfaits de notre répartition. En effet, le choix de pousser plus loin la qualité de notre projet ayant été fait en commun, les charges supplémentaires générées ont été équitablement réparties dans l'hexanôme.

Les rôles affectés à chacun en début de projet se sont révélés plutôt homogènes en quantité de travail, ce qui a donc permis à chacun de rester fidèle à ses tâches et devenir "spécialiste" dans son domaine auprès du reste de l'équipe.

Nous avons comptabilisé sur ce projet un total d'environ 265 heures, soit une quarantaine d'heures par individu (avec une légère surcharge pour le chef de projet et la responsable qualité, à travers leurs activités de coordination et de vérification).

Remarque :

Ce calcul prend en compte les heures effectuées durant les séances.

Commentaires



Retour sur le programme générée

Fruit du travail équilibré de chacun, tant dans la spécification que l'implémentation, nous sommes globalement très satisfait de notre application, ainsi que de notre gestion générale du projet.

Comme dit précédemment, nous avons été fortement intéressés par ce sujet, d'où un investissement plutôt considérable de notre part, nous ayant permis d'implémenter diverses fonctionnalités subsidiaires. Il est évident que dans un cadre professionnel, nous n'aurions pas poussé aussi loin notre travail, en vue des frais que génère la sur-qualité. Mais dans le cadre d'un projet à but éducatif, un tel investissement nous semblait non seulement pertinent mais très formateur (tant au niveau technique que managériale).

Finalement, nous avons donc pu implémenter l'ensemble des fonctionnalités spécifiées, requises par le client, ainsi que divers éléments améliorant la qualité du modèle (gestion poussée des collisions, prise en compte du temps de transfert des bagages, sauvegarde de simulation, calcul et affichage des étincelles, etc.), ou de l'interface (double-buffering, possibilité de zoom et de déplacement, paramétrage total de l'affichage, etc.).

Dans un esprit critique, nous pouvons cependant regretter certains soucis non résolus.

Par exemple, la possibilité d'accélérer la simulation au niveau du noyau génère des bugs, une fois une certaine vitesse dépassée. L'erreur est d'ailleurs connue : le déplacement des chariots, lors d'un top Horloge en mode accéléré, est trop grand, et il arrive que les chariots aient le temps de parcourir plusieurs rails. Notre algorithme récursif de gestion des collisions n'est alors pas capable de suivre une telle demande. Résoudre cette erreur nécessiterait donc une réflexion dépassant nos capacités dans un délai aussi court ...

En vue du nombre de cas limites, nous n'avons également pu faire une couverture de tests totale. Il nous arrive donc parfois de découvrir certains légers bugs (comme par exemple la non-réactivation du bouton Ajout Bagage lors du passage du mode automatique au mode manuel).

Néanmoins, grâce à notre soucis constant de modularité (distinction claire entre noyau, vue et interface ; développement orienté objet ; etc.), il serait également aisément d'implémenter certaines fonctionnalités supplémentaires ou de récupérer certains modules à d'autres fins.

Retour sur la gestion

Grâce à des règles de conduite précises au niveau de l'organisation et de la conception (élaboration continue de la Javadoc et commentage du code, vérifications et tests avant tout ajout sur le SVN, réunion régulières, ...), nous n'avons pas eu à souffrir de graves problèmes d'incohérence ou de pertes de données.

De plus, l'utilisation poussée de mails et des outils de notifications Redmine permirent de garder un contact permanent.

Nous regrettions, par manque de temps, la non-réalisation de livrables additionnels nous semblant pertinents, comme un guide d'utilisation ou une aide.

Nous avons pu découvrir pleinement l'avantage du cycle USDP et des méthodes Agile, qui nous ont permis d'avancer par incrément valides et de garder une cohérence tout au long des semaines.



Suggestions

Ce projet, en plus de nous sembler totalement pertinent dans notre formation d'ingénieurs informatiques, nous a beaucoup plu. Cet intérêt grandissant est dû à l'aspect professionnel et réaliste du sujet, nous invitant à réaliser de bout en bout une véritable application.

Nous ne voyons donc guère de reproches ou de suggestions à apporter. Les seuls soucis rencontrés sont liés à l'environnement employé : bien que couvrant de nombreuses fonctionnalités, JDeveloper peut causer certaines difficultés de prise en main. Nous avons notamment eu beaucoup de mal à le coupler avec notre serveur SVN, des erreurs de *versioning* apparaissant régulièrement.

Conclusion

Cet unique cycle fut riche en élaborations et en apprentissages personnels. Nous avons pu appliquer des méthodes nouvelles de gestions de projet, et découvrir de nouvelles technologies.

Dommage que le temps imparti n'ait permis un second cycle ...



Conclusion

Ce projet s'achève sur une application valide et agréable, pour laquelle chacun pu apporter ses méthodes et ses connaissances. Globalement satisfaits de notre travail et des enseignements tirés, tant sur le point méthodologiques que techniques, nous espérons pouvoir les appliquer rapidement à de prochains défis.

