**GLO-2004 : Génie logiciel orienté objet**

**16 Décembre 2014**

**Équipe Quicksort**

Bruno C. Coulombe

Camille Béland

Maxime Charron

Jean-Daniel Pearson

RecyclApp

Livrable #4 : Modèle de conception et Architecture logique

# Vision

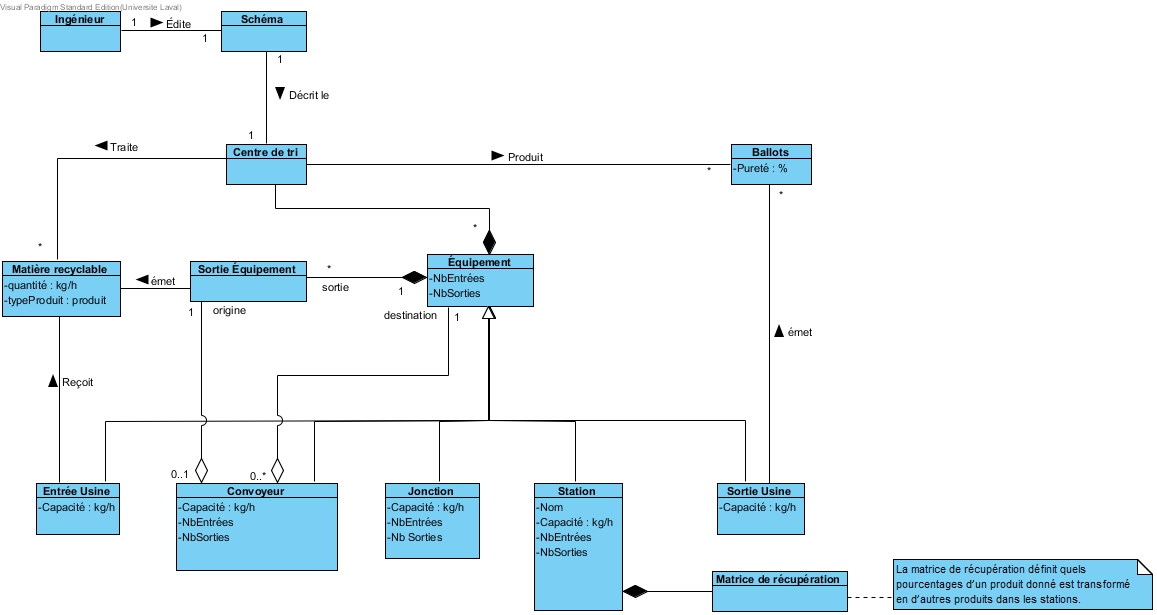
Actuellement, la conception d’un centre de tri de matières recyclables est un travail fastidieux. En effet, plusieurs machines ou stations manuelles doivent être synchronisées afin de produire le taux de pureté des matières recyclées exigé par le client à la sortie. Celles-ci doivent aussi fonctionner ensemble de manière fluide afin d’éviter les goulots d’étranglements. Présentement, la conception du plan de l’usine de recyclage est assurée par des ingénieurs d’expérience et leur tâche est fastidieuse, car ils ne disposent pas de technologie adéquate. En effet, les plans sont réalisés avec le logiciel Visio et les calculs avec Excel et ceux-ci ne permettent pas de modéliser avec réalisme et précision la disposition des stations de tri du centre. Beaucoup d’essais doivent donc être réalisés avant de trouver la combinaison optimale.

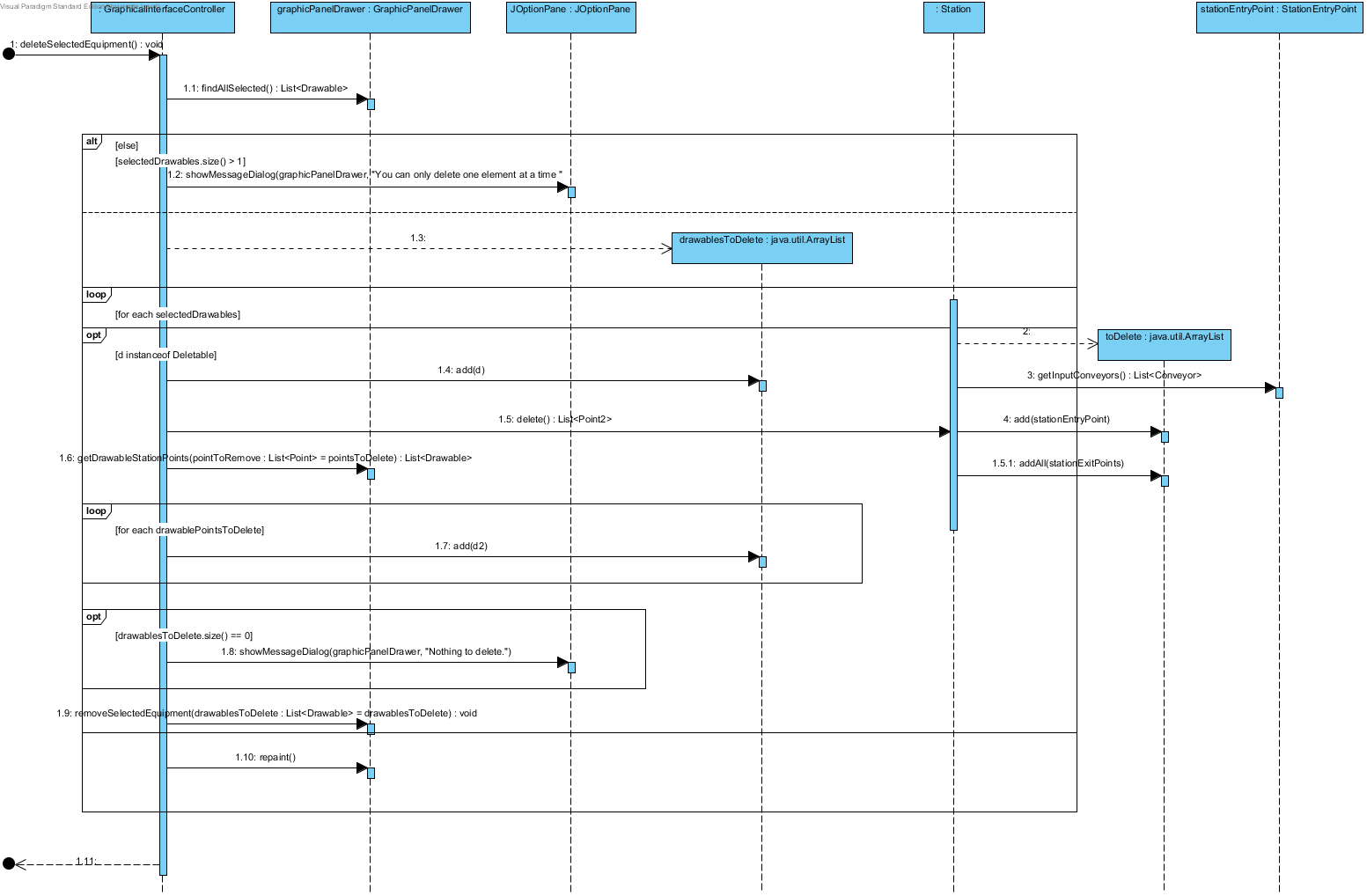
L’application RecyclApp peut apporter une solution au problème. En effet, celle-ci permettra aux ingénieurs de concevoir le centre de tri de manière interactive. Par exemple, ceux-ci pourront disposer les diverses stations du centre de tri sur un plan de travail et relier celles-ci par des arcs représentant les convoyeurs. L’application permet aussi de définir des points de jonction entre plusieurs convoyeurs. Les informations relatives à chaque station, telles que le nom et le nombre de sorties, peuvent être spécifiées. Il est aussi possible de préciser la quantité de chaque produit en entrée, ainsi que la quantité sortante en pourcentage. L’application permet aussi de transformer un produit en d’autres produits. Par ailleurs, RecyclApp permet la modification ou la suppression d’un convoyeur ou d’une station. De plus, la quantité de chaque produit voyageant d’une station à l’autre est calculée automatiquement en kilogrammes par heure et ces données peuvent être visibles par l’utilisateur en tout point du réseau. Des fonctionnalités de modification de l’affichage telles que la possibilité d’agrandir ou de rapetisser le plan de travail ou encore de faire apparaître une grille, magnétique ou non, facilitent la conception du centre de tri pour les ingénieurs. Finalement, RecyclApp utilise les mètres comme unité de mesure ce qui permet de concevoir un plan du centre de tri réaliste et à l’échelle. Chaque projet peut être sauvegardé et réouvert plus tard pour lui apporter des modifications.

Grâce à ses fonctionnalités adaptées aux besoins des utilisateurs et son interface intuitive, RecyclApp est une application tout-en-un qui permet de concevoir rapidement et avec facilité un centre de tri des matières recyclables. Les ingénieurs pourront avoir une vue d’ensemble sur la disposition des équipements du centre sans avoir à déplacer l’équipement réel. Puisque la quantité de matière recyclée est visible en tout point du réseau, les goulots d’étranglement pourront être repérés facilement. Les coûts du centre de tri seront réduits, car le temps de mise en place d’une nouvelle configuration du centre sera diminué et, avec la modélisation logicielle de la disposition des stations, il sera plus facile de trouver la disposition optimale afin de maximiser le rendement du traitement des matières recyclées. Enfin, avec une disposition de l’équipement optimale, la capacité du centre de tri peut être augmentée sans que cela nécessite d’équipement supplémentaire.

# Diagramme de cas d’utilisation

# Modèle du domaine

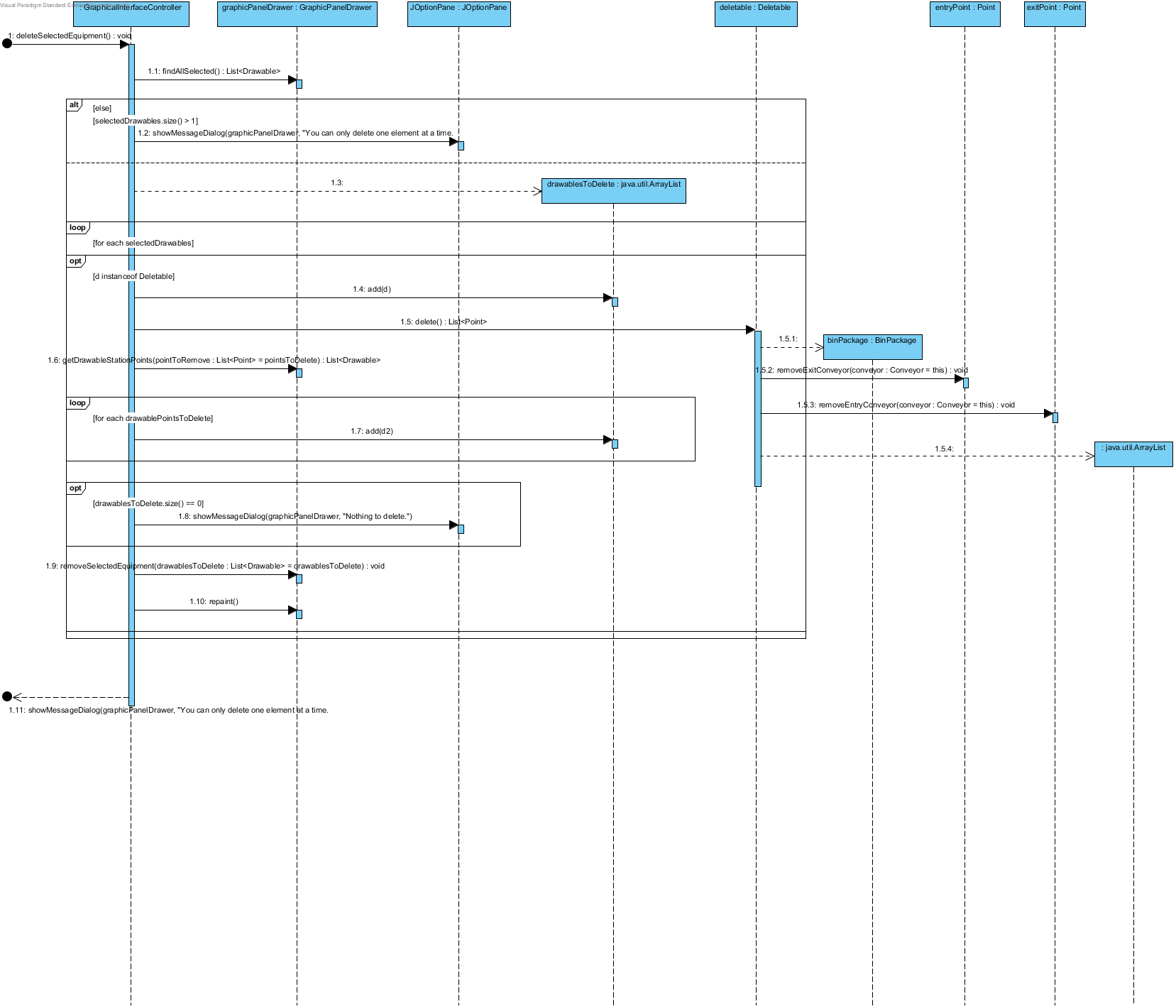


**Diagramme(s) de séquence pour la suppression d’une station (avec texte explicatif).**

La suppression d’une station commence par la fonction deleteSelectedEquipment() lors du clic sur le bouton delete par l’utilisateur. Cette fonction appelle findAllSelected() qui retourne les éléments sélectionnés du plan. Ensuite, si la liste contient des éléments sélectionnés, le programme vérifie si ceux-ci sont supprimables. Si oui, le programme appelle la fonction delete() associée à l’objet. Si cet objet est une station, le programme ajoute les points d’entrées associés à celle-ci à une liste. La fonction getDrawableToDelete() s’occupe ensuite de supprimer les points d’entrée et de sorite de la station de la liste de Drawable. Finalement, la fonction removeSelectedEquipment() enlève ensuite la station de la liste de Drawable. La fonction repain() est appelée et la liste de Drawable sans les éléments supprimés est affichée sur le plan.

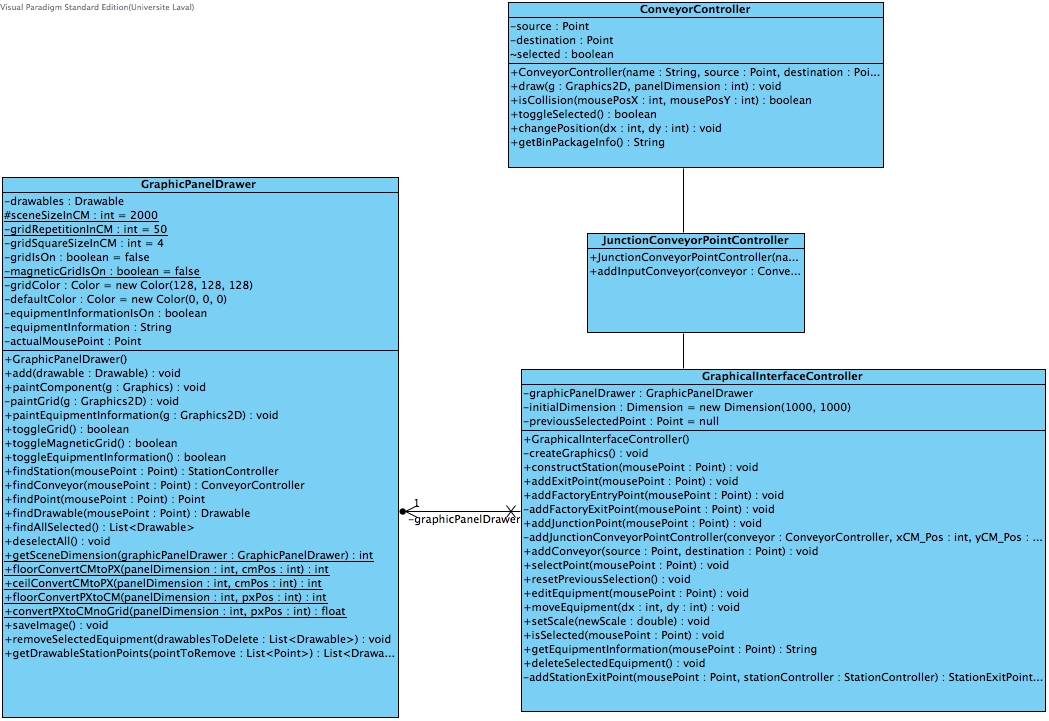
**Diagramme(s) de séquence pour la suppression d’un convoyeur/arc (avec texte explicatif).**

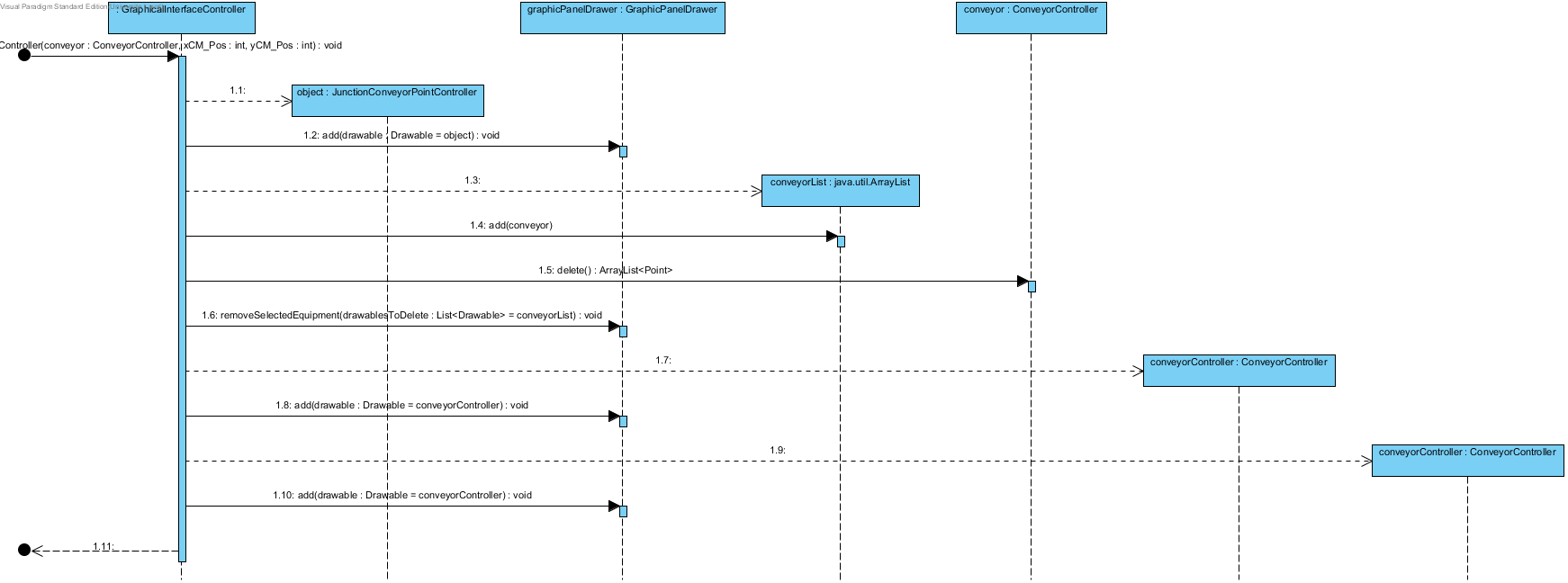
La suppression d’une station commence par la fonction deleteSelectedEquipment() lors du clic sur le bouton delete par l’utilisateur. Cette fonction appelle findAllSelected() qui retourne les éléments sélectionnés du plan. Ensuite, si la liste contient des éléments sélectionnés, le programme vérifie si ceux-ci sont supprimables. Si oui, le programme appelle la fonction delete() associée à l’objet. Si cet objet est un convoyeur, le programme supprime les références sur ce convoyeur dans son point d’entrée et de sortie en appelant les foncions removeExitConveyor() en appelant removeEntryConveyor(). Finalement, la fonction removeSelectedEquipment() enlève ensuite le convoyeur de la liste de Drawable. La fonction repain() est appelée et la liste de Drawable sans les éléments supprimés est affichée sur le plan.



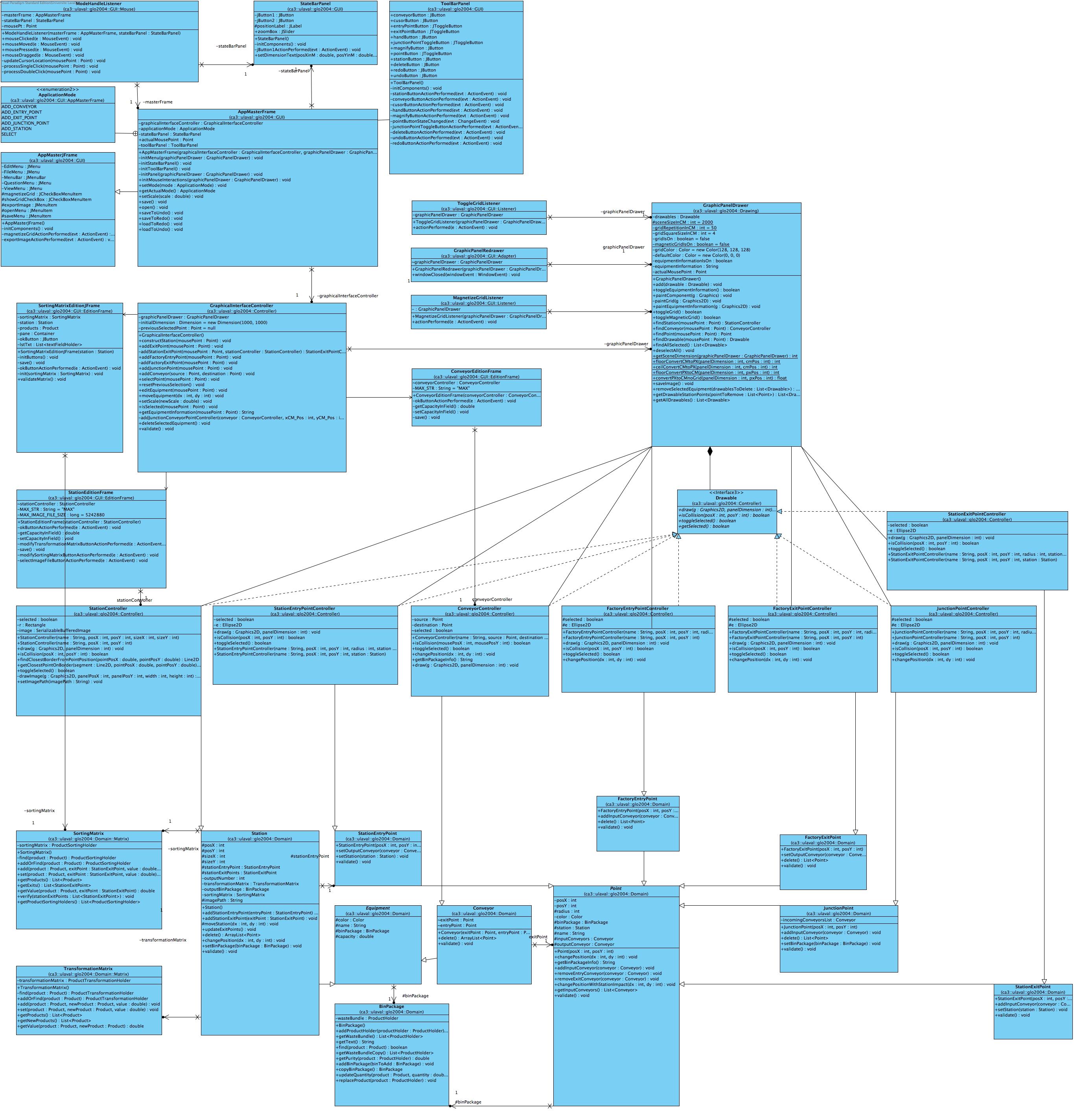
**Fonctionnalité préférée : ajout de point de flexion aux convoyeurs.**

La fonctionnalité dont nous sommes le plus fier est la possibilité d’ajouter d’un point de flexion aux convoyeurs. Premièrement, lorsque l’utilisateur ajoute un point de flexion au convoyeur, un JunctionConveyorPointController est créé et ajouté à la liste de Drawable. Ensuite, la fonction delete() prépare le convoyeur concerné en supprimant sa référence sur les deux points qui lui sont associés. Ensuite, deux convoyeurs sont créés avec la méthode add() de part et d’autre du point de flexion ajouté.





# Diagramme de classe



## **Texte explicatif du diagramme de classes**

Le diagramme de packages est composé de cinq packages. Tous les éléments qui auront un lien direct avec l'affichage se retrouveront dans le «GUI». La logique de l'application se retrouvera dans le «Domain» et jamais le «Domain» ne communiquera avec le package du «GUI». Afin de faire le pont entre le «GUI» et le «Domain», le package «Controller» contiendra différentes classes pour assurer la coordination. Le package «App», bien que minimaliste, s'occupera de lancer le «Controller» du package «Controller» avec les bons paramètres (projets à ouvrir, configuration...). Le package «App» est vraiment le point d'entrée du programme et il ne communiquera qu'avec le package «Controller».

Le diagramme de classes montre les différentes classes dans les packages de l'application.

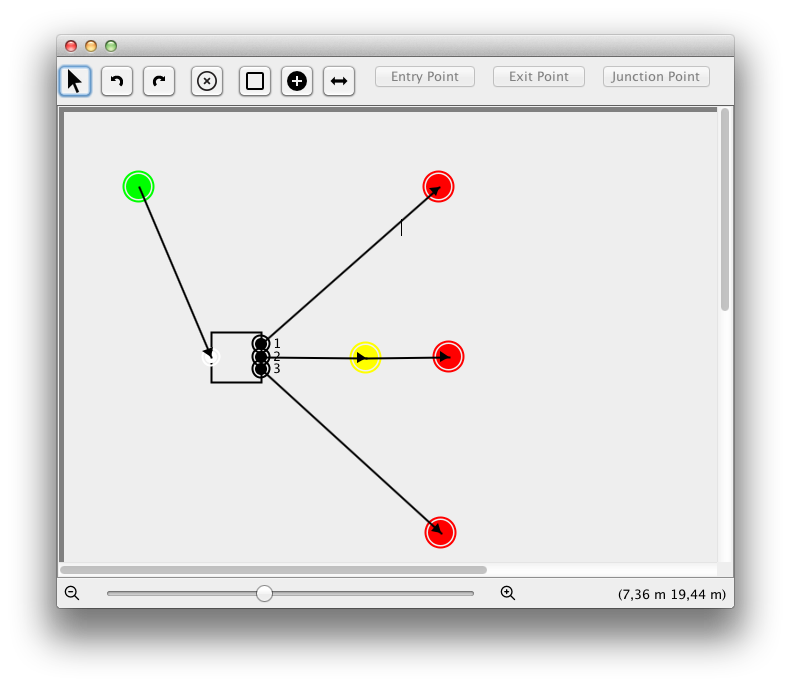
Les classes faisant partie du package «GUI» seront en grande partie des extends de composants Swing. «MainWindow» sera notre jFrame principal et fera la plupart de sa besogne avec la classe controller. Windows sera composé de :

* ToolBar qui contiendra les menus Fichier, Éditer, Affichage, Aide.
* MenuBar qui contiendra les boutons principaux de l'interface, comme ajouter une station.
* StateBar qui affichera des informations utiles à l'utilisateur, comme la position du curseur en mètres et le zoom.
* StationEditionPanel qui sera un menu d'affichage rapide des informations de l'objet sélectionné.
* WorkPanel la zone d'affichage du plan de travail.

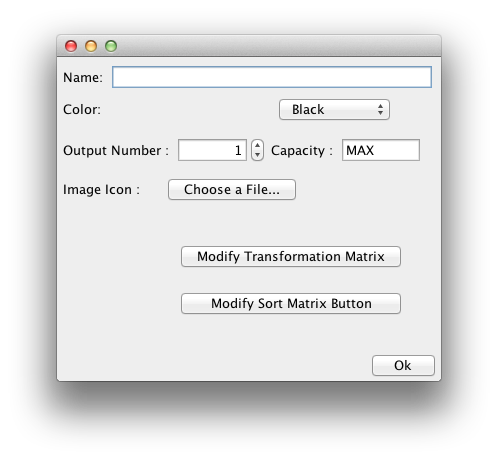
«Controller» sera le chef d'orchestre de l'application et adoptera la forme telle que vue par Larman. Le «Controller» contiendra l'état de curseur qui sera une classe héritant de «CursorState». Dépendamment de l'état du curseur, le contrôleur ne Il fera afficher la plupart du contenue dans le «GraphicalPanel» qui se chargera d'exporter en image pour impression ou, le plus souvent, d'envoyer le rendu graphique à «WorkPanel». Pour créer le rendu, la seule chose à faire sera de dessiner tous les objets «Drawable».

Les différents «Point» ont été unifiés au maximum pour faciliter l'affichage. Une station possèdera un ou aucun «StationEntryPoint», mais possèdera au moins un ou plusieurs «StationExitPoint». L'ingénieur pourra éditer le «BinPackage» d'un «FactoryEntryPoint», mais le «BinPackage» de «FactoryExitPoint». Les «Conveyor» permettent de relier deux «Point».

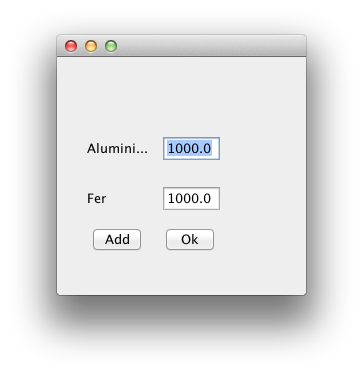
# Esquisse de l’interface



**Édition d’un point d’entrée**



**Édition de la matrice de tri**



# Glossaire

|  |  |
| --- | --- |
| Terme | Définition |
| Arc | Synonyme de convoyeur aux vues du système RecyclApp. |
| Ballot de matière | Matières recyclées traitées obtenues à la sortie du centre de tri. |
| Centre de tri | Usine qui reçoit et traite le contenu des bacs à recyclage. |
| Convoyeur | Équipement de machinerie reliant les différentes stations, entrées, sorties et point de jonction entre eux. |
| Entrée d’une station | Endroit de la station où les matières à être traitée entrent dans celle-ci. |
| Entrée de matière | Endroit du réseau de convoyeur où des matières non traitées entrent dans le système de tri. |
| Nœud | Station, entrée, sortie et point de jonction qui est relié à d’autres équipements à l’aide d’arcs. |
| Plan de travail | Espace de l’interface du logiciel dans lequel est élaboré le schéma de l’usine. |
| Point de jonction | Endroit du réseau de convoyeur où 2 d’entre eux ou plus fusionnent en un seul. |
| Schéma | Représentation visuelle du contenu de l’usine dans le plan de travail. |
| Station | Endroit où s’effectue la transformation ou le tri de matière. Ce peut être une pièce de machinerie ou des ressources humaines. |
| Sortie de matière | Endroit du réseau de convoyeur où les matières sortent du système de tri après avoir été triées. |
| Sortie d’une station | Endroit de la station où les matières triées sortent de celle-ci. Chaque station a une sortie par défaut et peut en posséder plus d’une. |

# Annexe

## **Trello**

