



# RecyclApp

Livrable #2 : Modèle de conception et Architecture logique

**GLO-2004 : Génie logiciel orienté objet**  
**16 octobre 2014**

**Équipe Quicksort**

Bruno C. Coulombe  
Camille Béland  
Maxime Charron  
Jean-Daniel Pearson

## Table des matières

Vision .....	2
Risques .....	3
Diagramme de cas d'utilisation .....	4
Modèle du domaine.....	5
Diagramme de séquence système .....	6
Diagramme de classe .....	7
Texte explicatif du diagramme de classes.....	8
Esquisse de l'interface.....	9
Diagrammes de séquence .....	10
Ajout d'un convoyeur .....	10
Ajout d'un équipement.....	11
Mise à jour des calculs .....	12
Affichage du réseau .....	13
Sélection d'un élément.....	14
Diagramme d'état .....	15
Glossaire.....	16
Annexe .....	17
Trello.....	17

## 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.

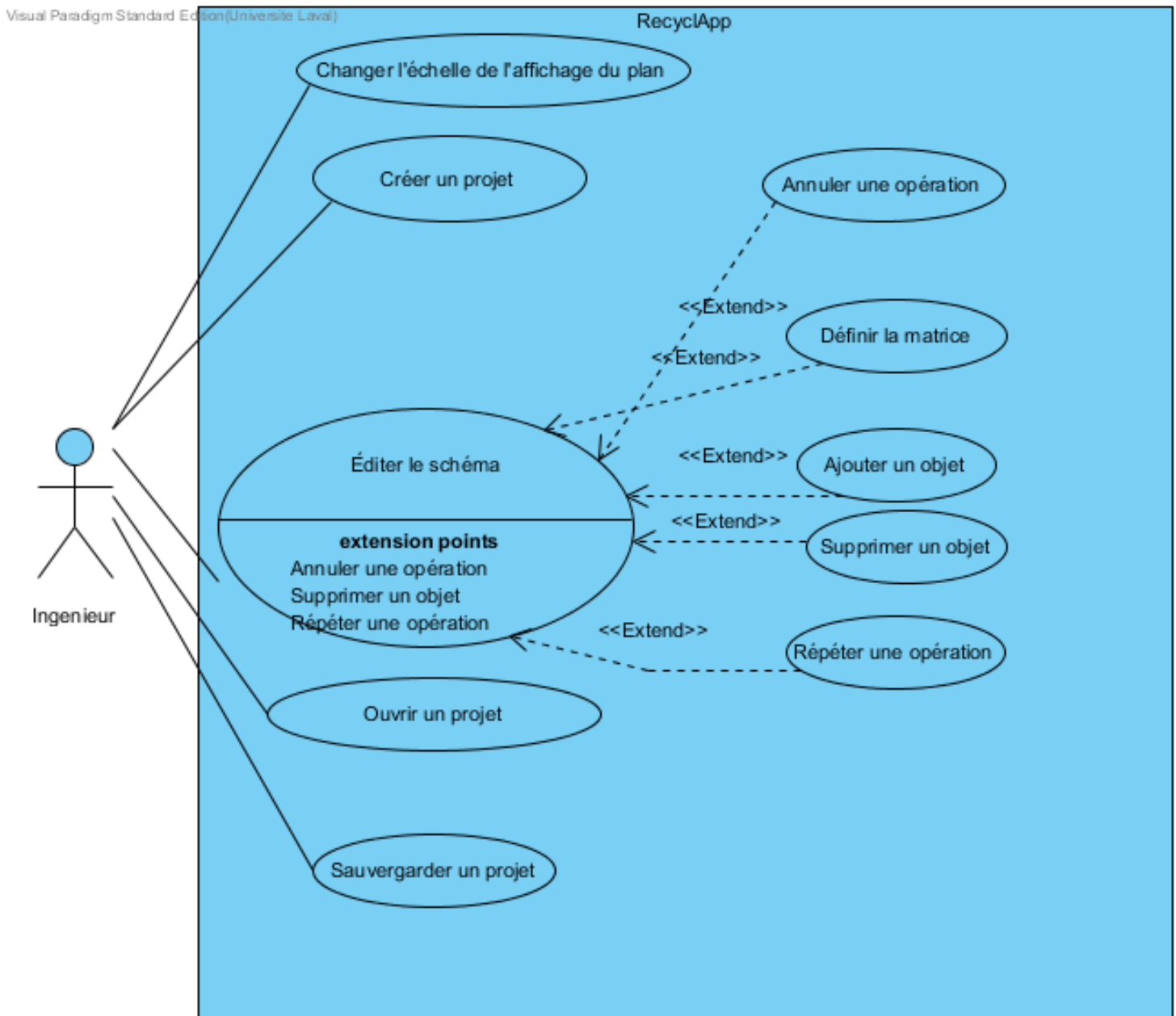
L'application RecyclApp peut apporter une solution au problème. En effet, celle-ci permettra aux ingénieurs de concevoir le centre de tri de manière interactive. Par exemple, ceux-ci pourront disposer les diverses stations du centre de tri sur un plan de travail et relier celles-ci par des arcs représentant 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, la description, ainsi que 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. 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 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.

## **Risques**

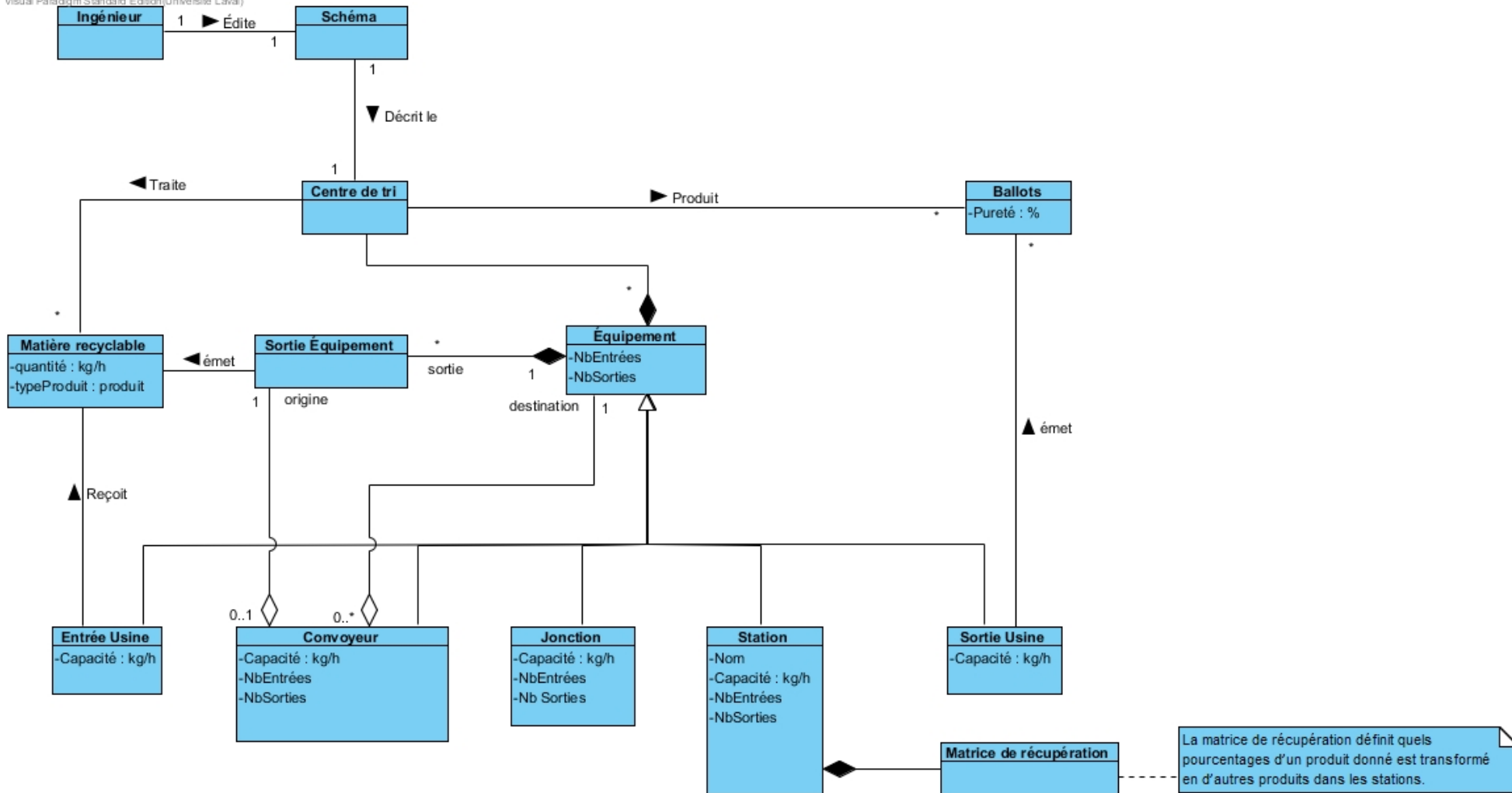
La conception d'un centre de tri en fonction des exigences du client est une tâche extrêmement complexe. Le logiciel doit donc être assez configurable pour que l'utilisateur puisse simuler ces scénarios complexes de façons simple et exacte. Le client demande que l'application soit programmée en java, ce qui l'avantage d'être multiplateformes. L'équipe de projet ayant peu d'expérience avec ce langage de programmation, une attention supplémentaire devra être portée pendant le développement.

## Diagramme de cas d'utilisation

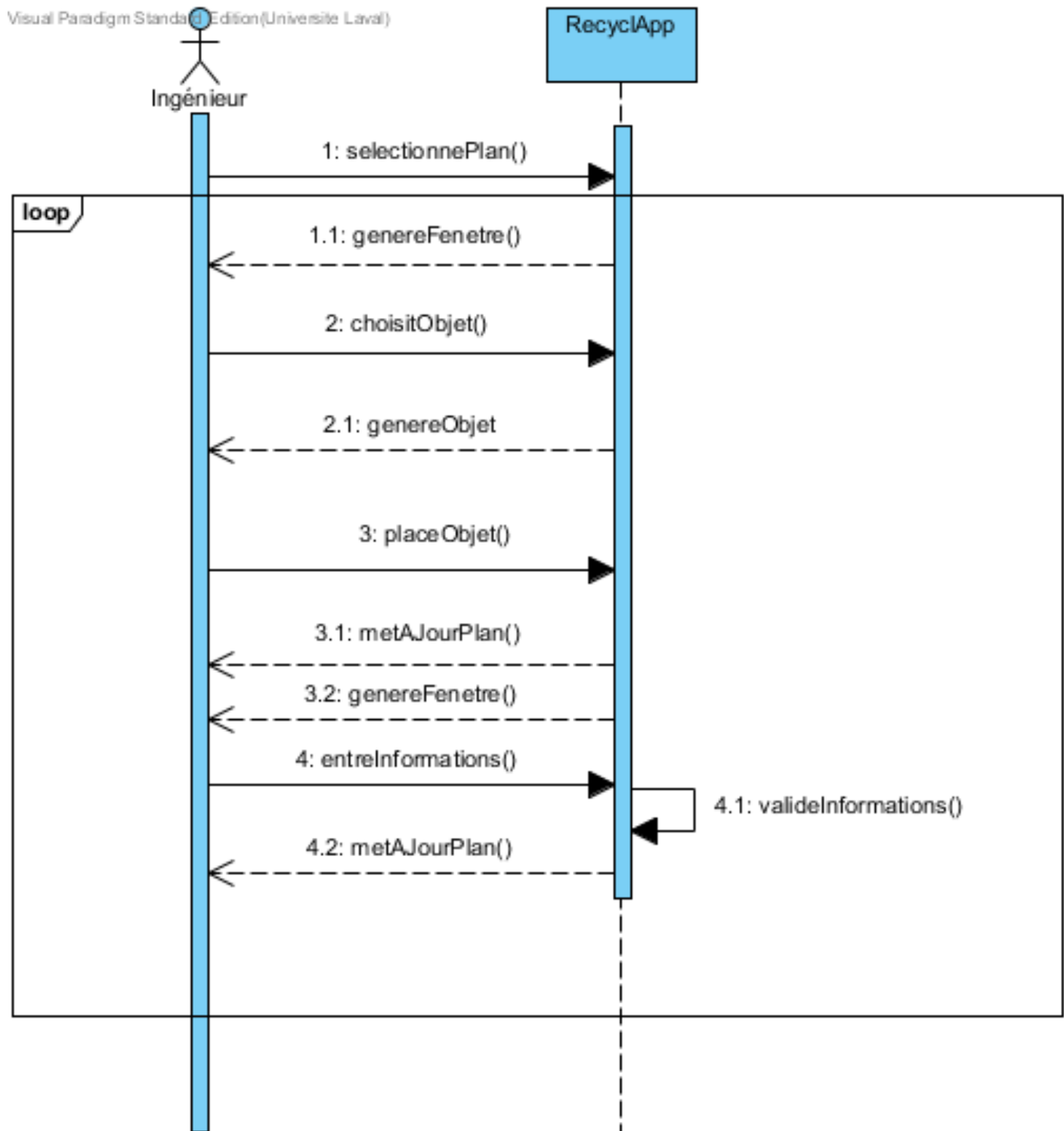


## Modèle du domaine

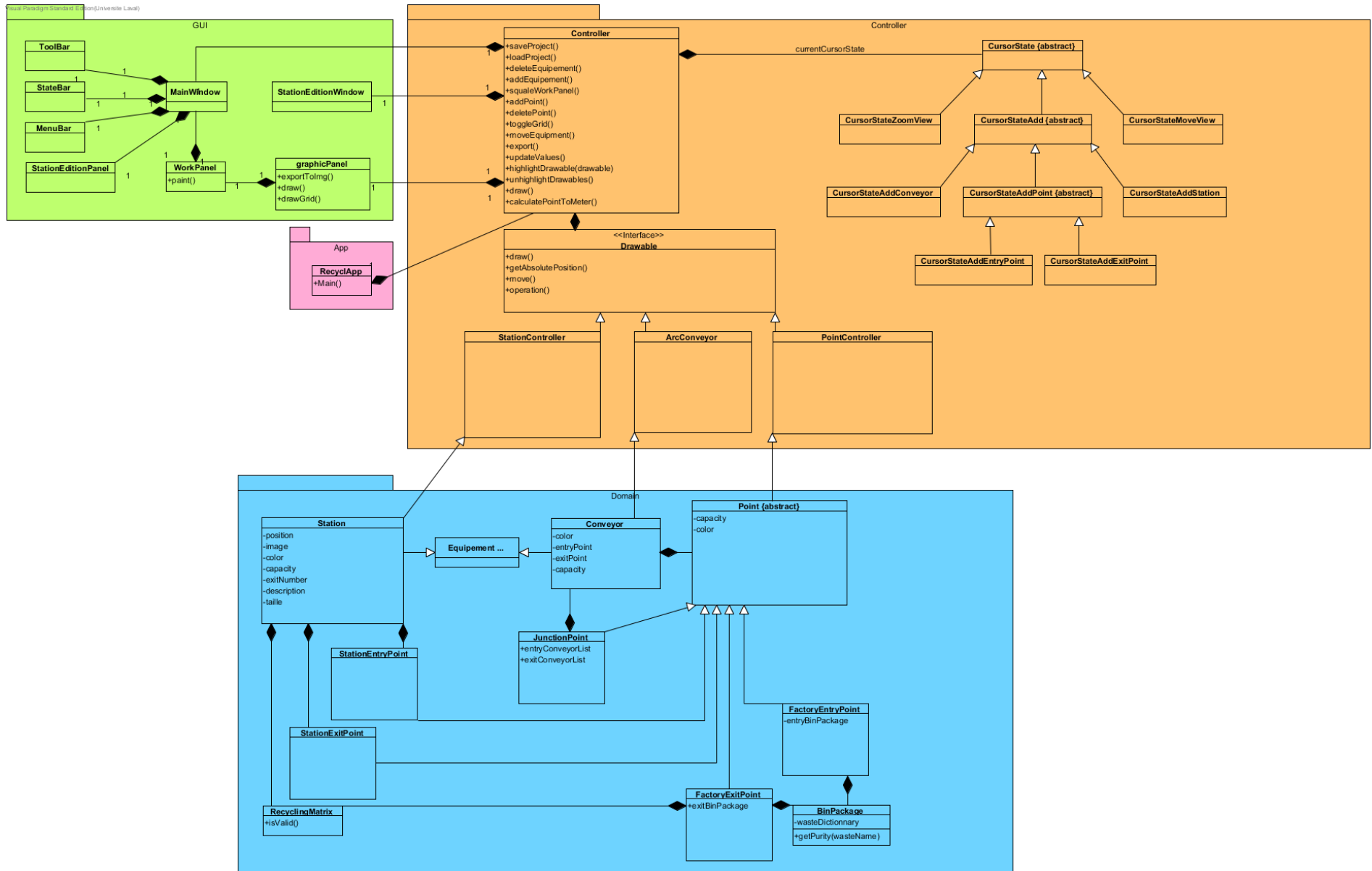
Visual Paradigm Standard Edition (Université Laval)



## Diagramme de séquence système



## 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.

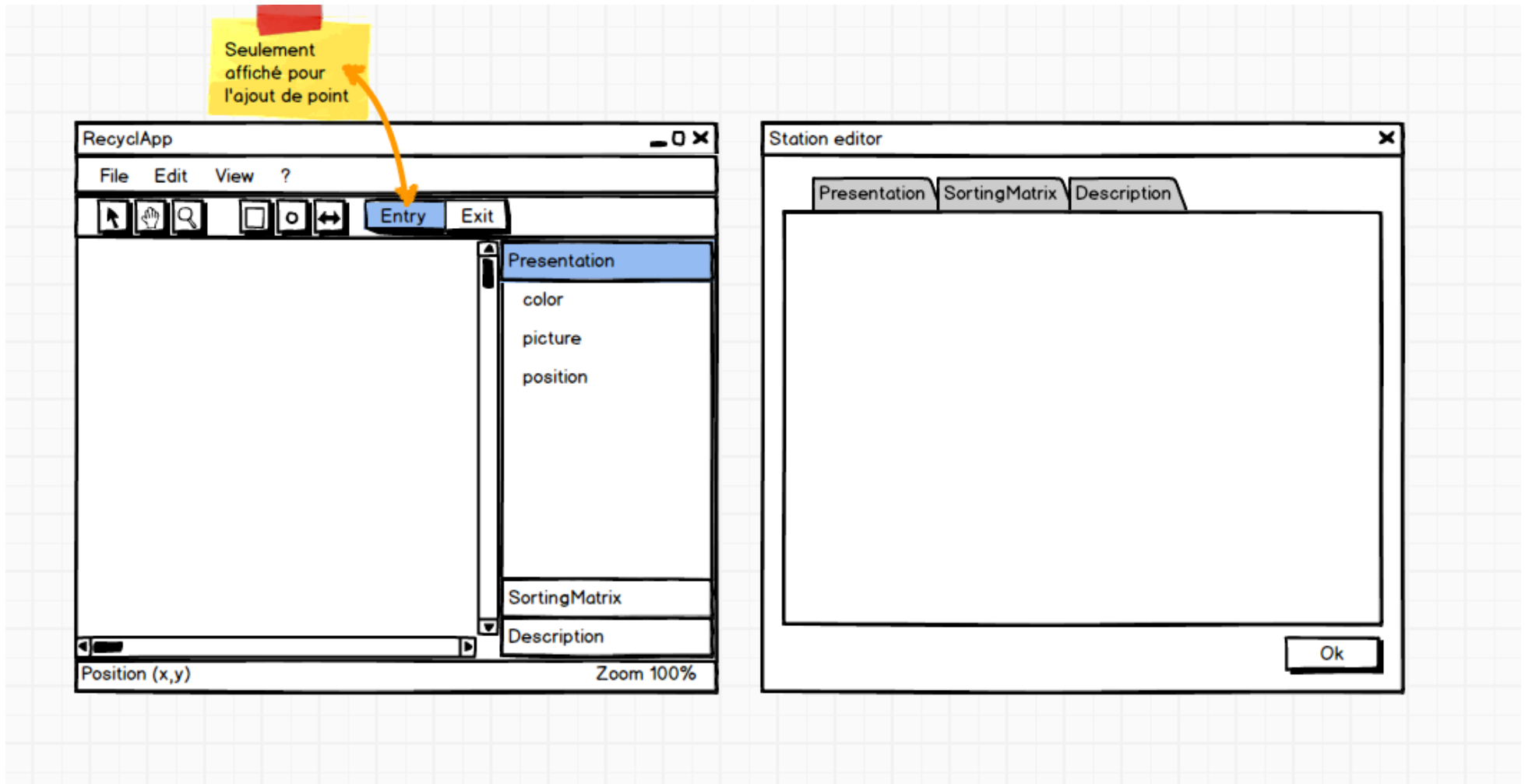
Les classes faisant partie du package «GUI» seront en grande partie des extends de composants Swing. «MainWindow» sera notre JFrame principal et fera la plupart de sa besogne avec la classe controller. Windows sera composé de :

- 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 fera afficher la plupart du contenu 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».

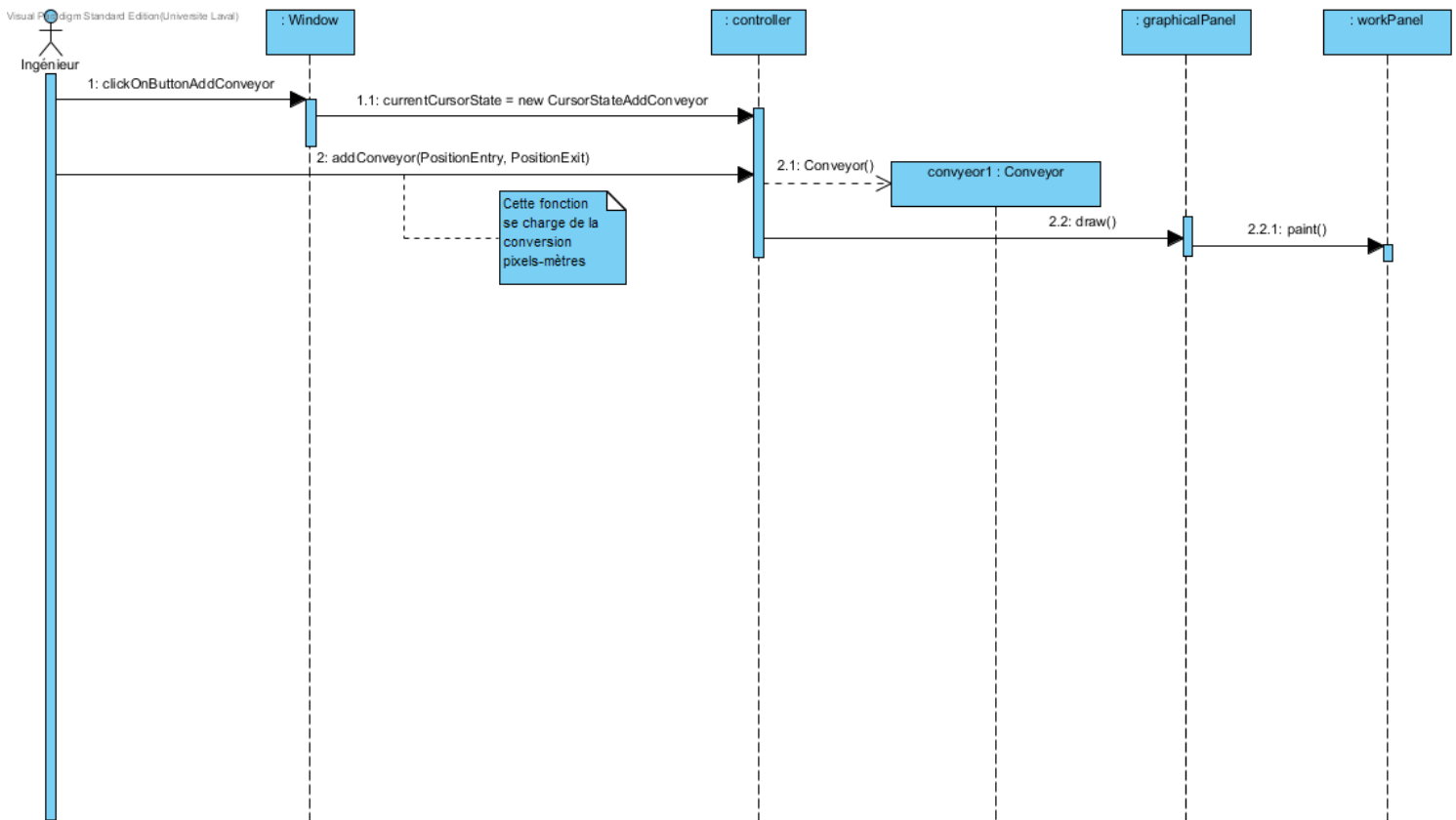
Les différents «Point» ont été unifiés au maximum pour faciliter l'affichage. Une station possèdera un ou aucun «StationEntryPoint», mais possèdera au moins un ou plusieurs «StationExitPoint». L'ingénieur pourra éditer le «BinPackage» d'un «FactoryEntryPoint», mais le «BinPackage» de «FactoryExitPoint».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

## Esquisse de l'interface



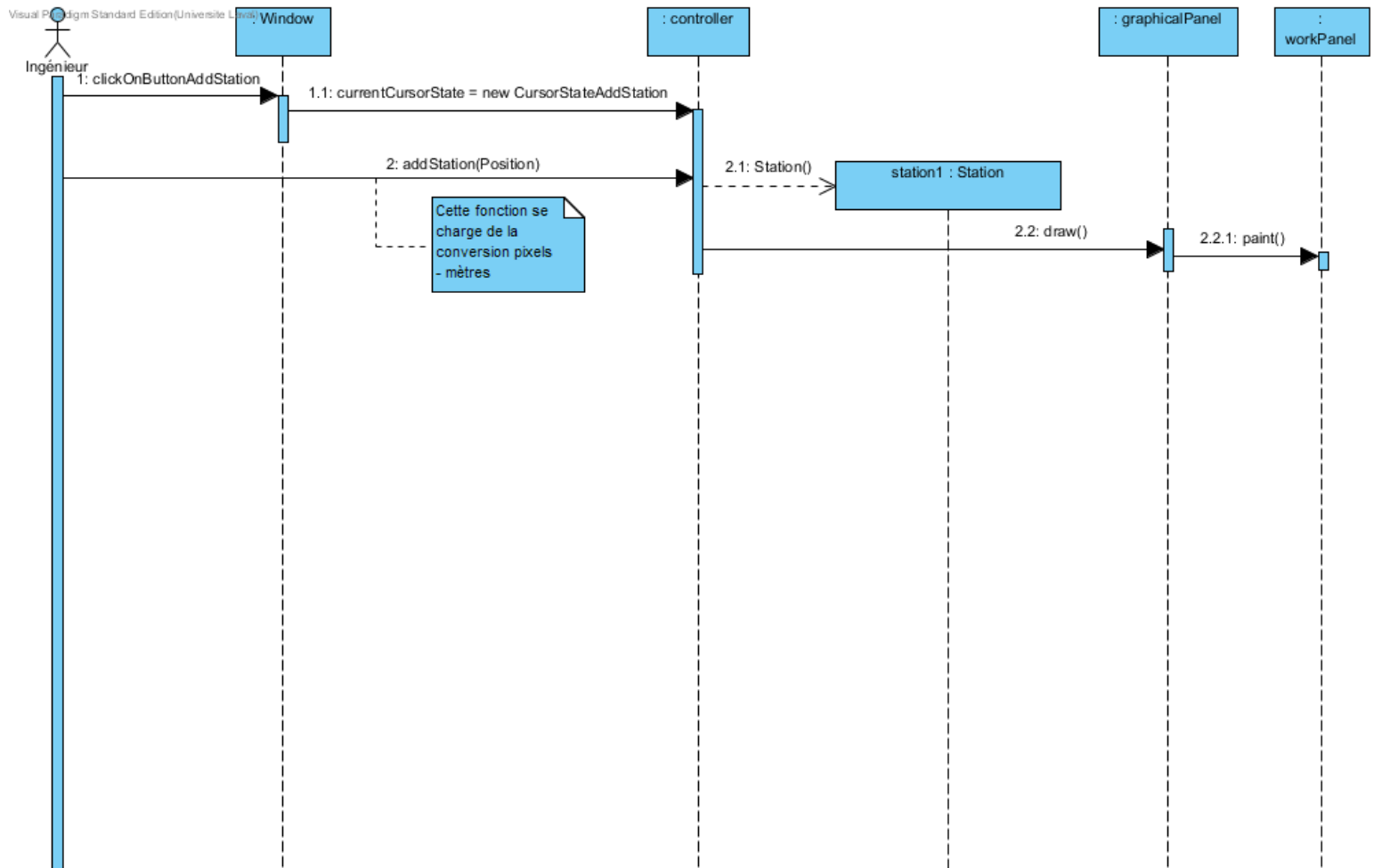
## Diagrammes de séquence

### Ajout d'un convoyeur



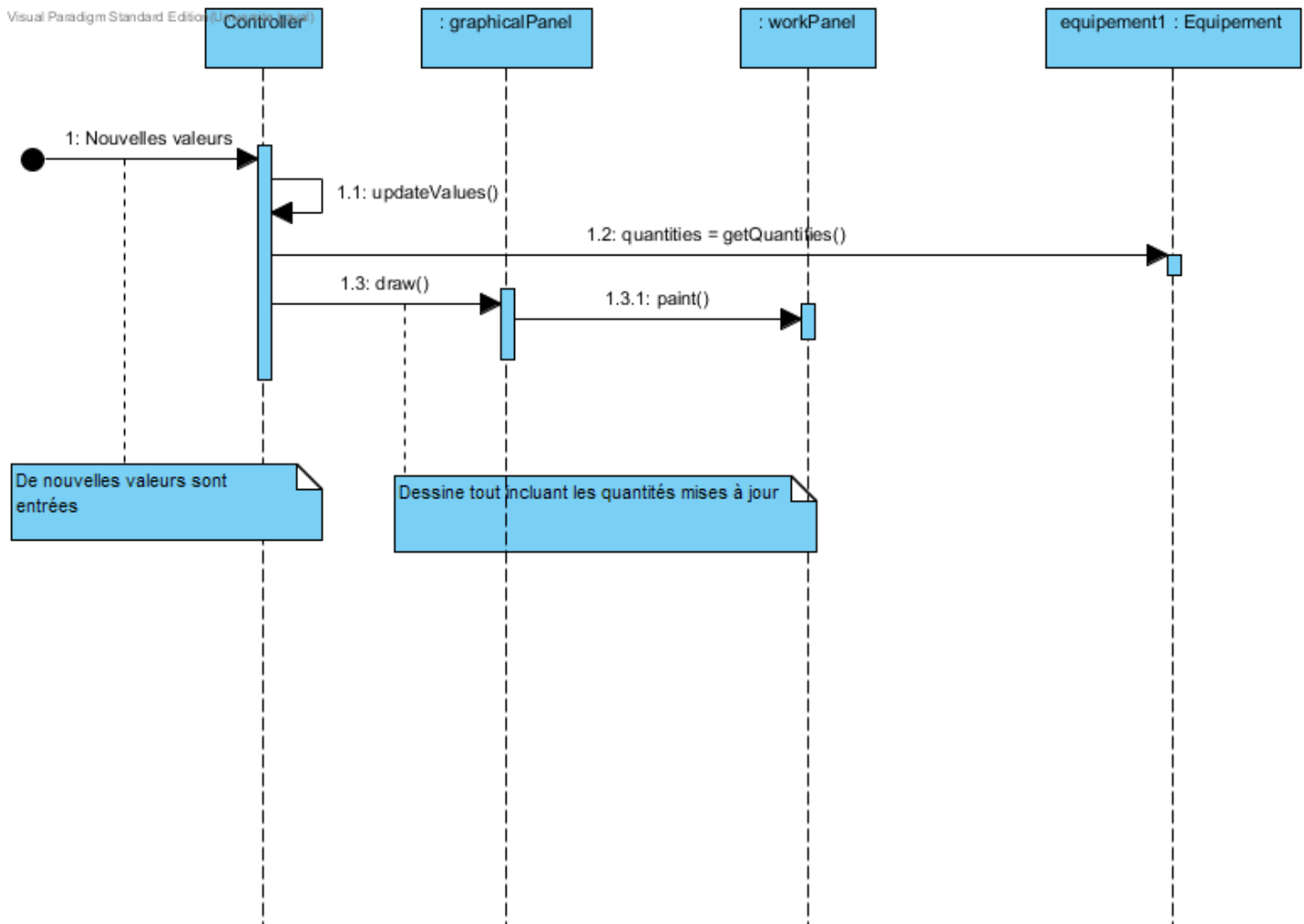
La séquence concernant l'ajout d'un convoyeur est bien simple. En fait, l'ingénieur doit d'abord cliquer sur le bouton d'ajout d'un convoyeur. Le curseur doit ensuite entrer en mode d'ajout de convoyeur. L'ingénieur doit ensuite cliquer pour indiquer le point d'entrée du convoyeur et son point de sortie. Avec ces informations, la fonction `addConveyor` se charge de la conversion de la position de mètres vers pixel et de pixel vers bytes. Avec ces informations, un nouvel objet `Conveyor` est créé. Le contrôleur appelle ensuite la méthode `draw` du `graphicalPanel` et le `graphicalPanel` appelle la méthode `paint` du `workPanel`. Ceci permet de mettre à jour l'affichage avec le nouveau convoyeur.

## Ajout d'un équipement



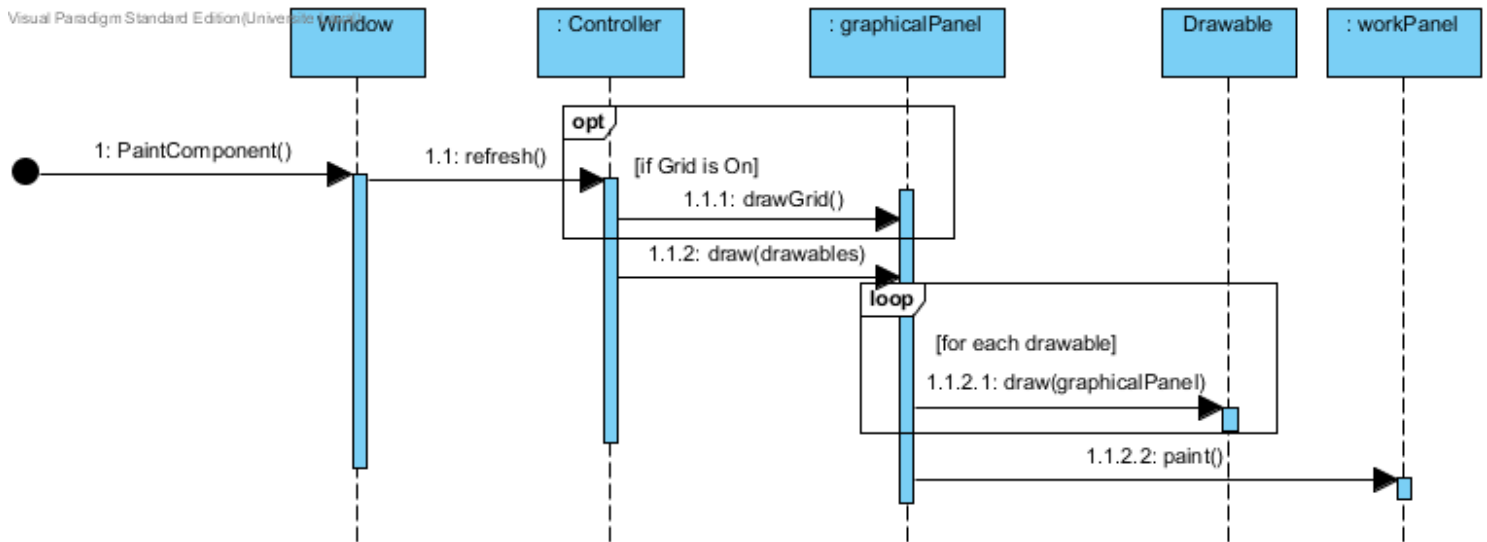
Premièrement, l'ingénieur clique sur le bouton d'ajout d'une station de la Window et le curseur change d'état pour celui d'ajout de station. Cette information est envoyée au contrôleur. Lorsque l'ingénieur clique sur un point du plan avec le curseur en mode ajout d'équipement, l'événement est transmis au contrôleur qui crée un nouvel objet Station avec la fonction `addStation`. Cette fonction se charge de la conversion de la position de mètres vers pixel et de pixel vers bytes. Le contrôleur appelle alors la fonction `draw` de `graphicalPanel` pour dessiner les objets de l'affichage incluant la nouvelle station créée. Le `graphicalPanel` appelle ensuite la fonction `paint` de `workPanel` qui affiche les objets dessiner à l'écran.

## Mise à jour des calculs



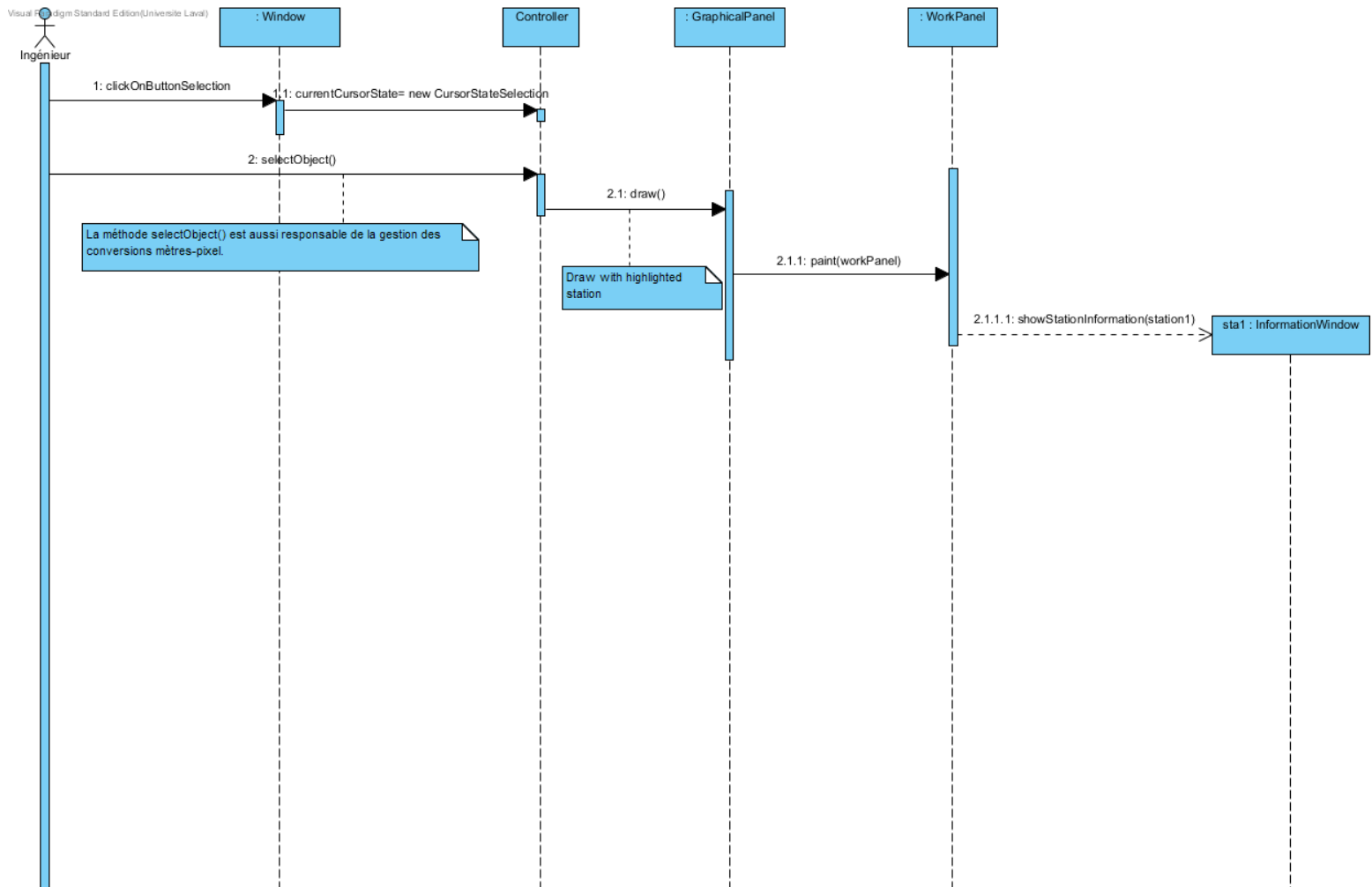
La mise à jour des calculs est faite en temps réel (avec la méthode `updateValues`), lorsque de nouvelles valeurs sont entrées. Le contrôleur appelle alors la méthode `getQuantities` des équipements pour aller chercher les quantités qui transitent dans chaque équipement (le diagramme de séquence est l'exemple du traitement d'un seul équipement) et qui ensuite appelle la méthode `draw` du `graphicalPanel` pour mettre à jour les quantités affichées. Finalement, la méthode `paint` du `workPanel` est appelée pour afficher à l'écran la mise à jour effectuée.

## Affichage du réseau



Le point de départ est le `PaintComponent()` qui est appelé lors d'un nouvel affichage de réseau. La `Window` fait alors appel à la méthode `refresh` du contrôleur. Ensuite, le contrôleur appelle la méthode `draw` avec tous les `drawables` en paramètre et si la grille doit être affichée, celui-ci appelle aussi la méthode `drawGrid` du `graphicalPanel`. Chaque `drawable` est alors dessiné sur le `graphicalPanel` avec leur méthode `draw`. Pour finir, la méthode `paint` du `workPanel` est appelée par le `graphicalPanel` et la représentation graphique du réseau est affichée à l'utilisateur.

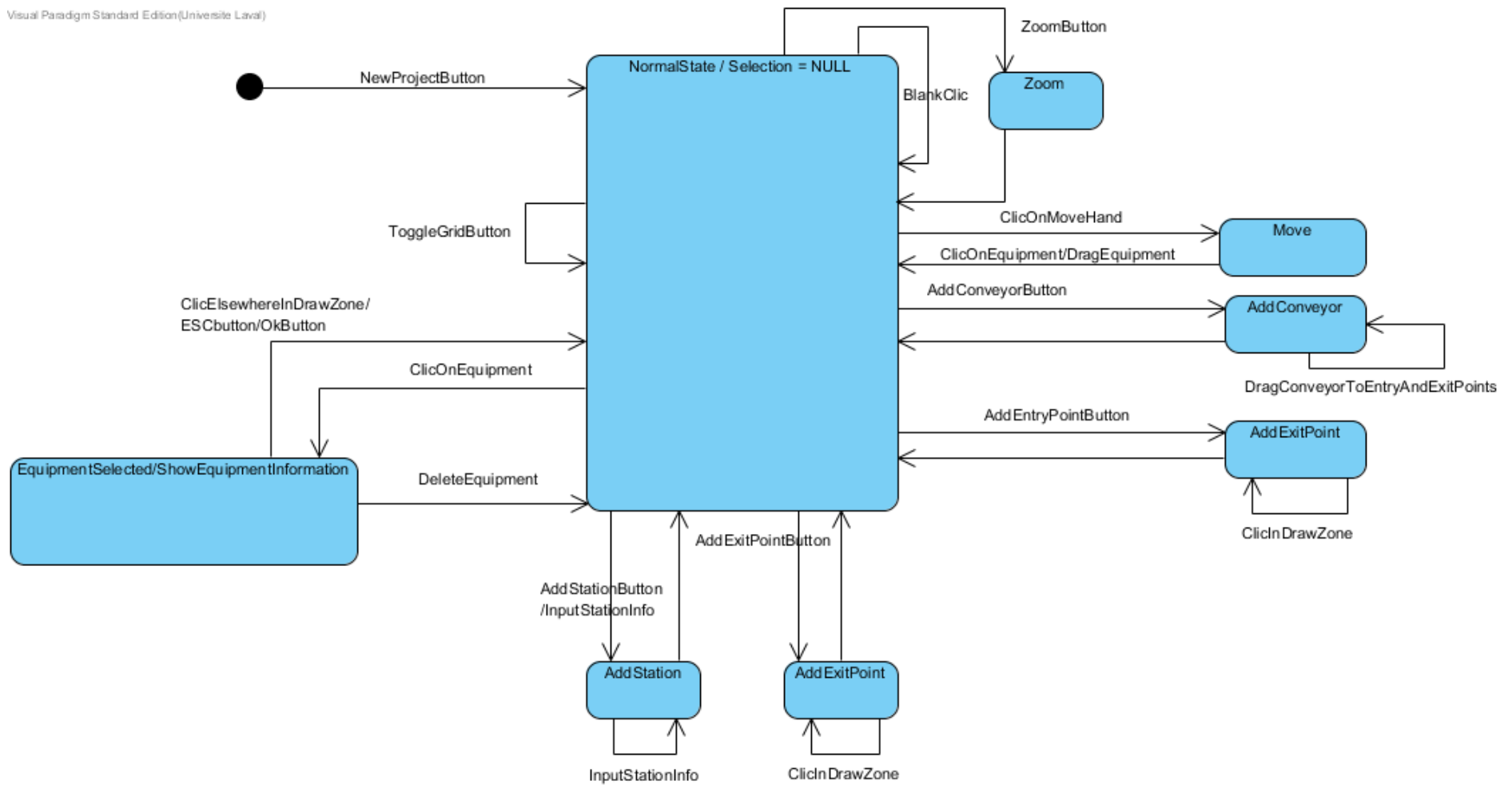
## Sélection d'un élément



La séquence concernant la sélection d'un objet est relativement simple. L'utilisateur doit d'abord cliquer sur le bouton de sélection. L'état du curseur est mis à jour et il entre en mode sélection. Ensuite, l'utilisateur sélectionne l'objet voulu. La fonction selectObject du Controller est appelée. C'est elle qui se charge de convertir la position en mètres en pixel et en bytes. Le Controller appelle ensuite la fonction draw du graphicalPanel qui dessine l'objet en surbrillance. La fonction paint du workPanel est ensuite appelée afin d'afficher l'objet sélectionné en surbrillance.

## Diagramme d'état

Visual Paradigm Standard Edition (Université Laval)





## 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

The screenshot displays a Trello board for the 'GLO-2004 Project'. The board is organized into four columns representing different iterations of the project. Each iteration contains a list of tasks with progress indicators and due dates.

**Iteration #1 - Analyse**

- Exigences: 4/4, Sep 23, MC, BC
- Modélisation domaine d'affaires: 2/2, Sep 23, MC
- Diagramme de séquence système: 3/3, Sep 23, MC
- Construire le Trello: 4/4, Sep 23, BC

**Iteration #2 - Conception**

- Révision du livrable # 1: 4/4, Oct 16, MC, BC
- Mettre à jour le Trello: 3/3, Oct 16, BC
- Esquisses des interfaces utilisateur: 1/1, Oct 16, BC
- Diagramme de package (architecture logique): 2/2, Oct 16, BC
- Diagramme de classe: 2/2, Oct 16, BC
- Diagrammes de séquence: 5/5, Oct 16, MC, BC
- Diagramme d'état: 1/1, Oct 16, BC
- Implémentation du code (Début): 1/1, Oct 16, BC

**Iteration #3 - Construction et démonstration**

- Modèle de conception (mis à jour): 0/3, Nov 25, MC
- Mettre à jour le Trello: 0/2, Nov 25, BC
- Démonstration: 0/2, Nov 25, BC
- Implémentation du code: 0/2, Nov 25, BC

**Iteration #4 - Construction et démonstration**

- Modèle de conception (mis à jour): 0/3, Dec 16, MC
- Mettre à jour le Trello: 0/1, Dec 16, BC
- Démonstration finale: 0/1, Dec 16, BC
- Finaliser l'implémentation du code: 0/2, Dec 16, BC