

# **Livrable 2 - Modèle de conception et Architecture logique**

**Modèle de conception et Architecture logique**

**Par : Gabriel Rousseau, Mathieu Roussel, Samuel Vézeau, William Bourque et William Gingras**

**Cours GLO-2004-IFT-2007  
Université Laval  
Session Automne 2024**

## Table des matières

### Table des matières

Diagramme de classes de conception .....	3
Architecture Logique .....	5
Diagramme de séquence de conception .....	6
Gantt mis à jour .....	9
Contribution des membres de l'équipe .....	9

## Diagramme de classes de conception

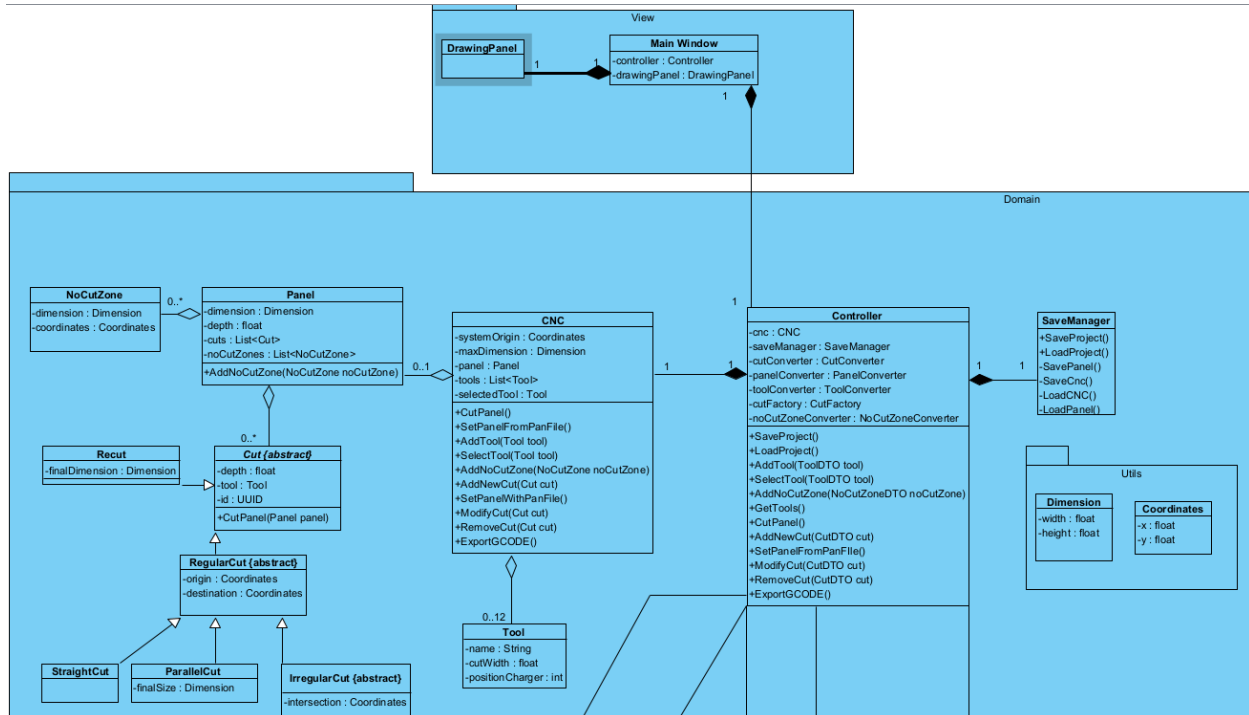


Figure 1: Vue et le domaine du diagramme de classe

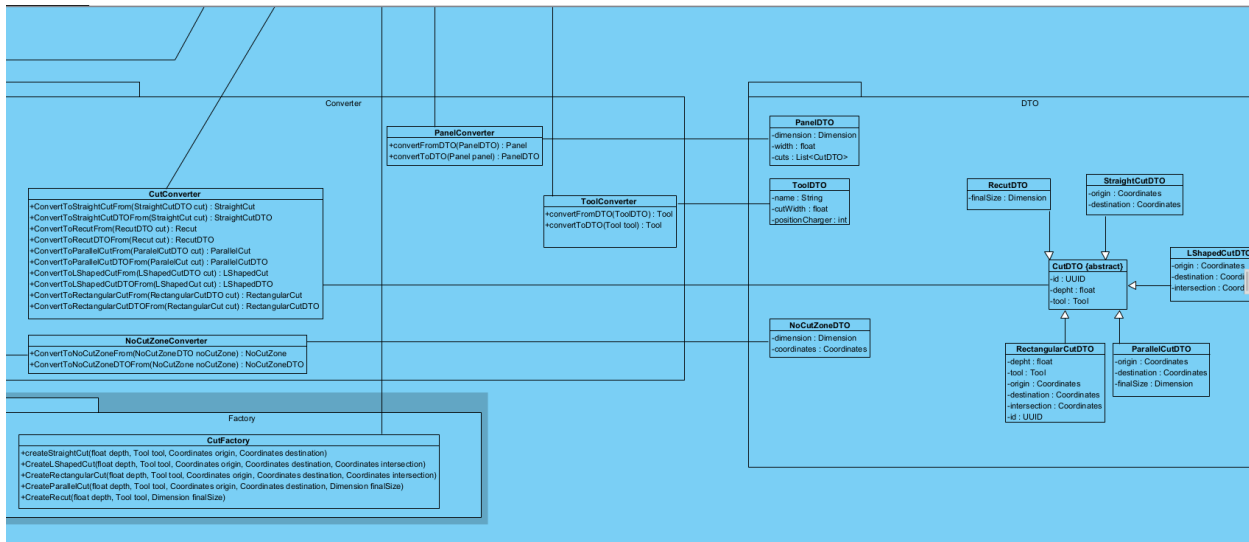


Figure 2: Les Convertisseurs, les DTO et la Factory

La classe **DrawingPanel** fait partie de la vue et est responsable de la représentation visuelle du panneau et des découpes. Elle est associée à la classe **MainWindow**. Cette dernière fait également partie de la vue et contient un **DrawingPanel** ainsi qu'un **Controller**. La classe **MainWindow** entretient une relation de

composition avec **DrawingPanel**, qui est utilisé pour afficher les informations graphiques du panneau en cours, et avec le **Controller**, qui gère l'arrière-plan du programme.

La classe **Controller** est responsable de la gestion des actions de la CNC et de la gestion globale du projet. Elle peut charger et sauvegarder des projets grâce à sa composition avec la classe **SaveManager**. La relation de composition avec la classe **CNC** permet au **Controller** de gérer les outils, le panneau et les découpes. Le **Controller** peut également accéder à tous les convertisseurs disponibles pour le transfert de données entre ses composants.

Le **SaveManager** est responsable des opérations de sauvegarde et de chargement de projets, de panneaux et des paramètres de la CNC. Il est composé par le **Controller**, qui décide des actions de sauvegarde à effectuer. Quant à la classe **CNC**, elle représente la machine CNC réelle. La **CNC** est composée du panneau (**Panel**) et gère directement les outils (**Tool**). Elle peut effectuer des actions comme la découpe du panneau, l'ajout de zones interdites, l'ajout ou la suppression d'outils, et l'exportation d'un fichier GCODE.

La classe **Tool** représente un outil de découpe utilisé par la CNC. Elle contient des attributs tels que le nom, la largeur de la coupe et la position. Elle est directement associée à la classe **CNC**, car c'est cette dernière qui crée les outils.

La classe **Panel** représente le panneau actuellement utilisé par la CNC et contient ses dimensions, sa profondeur, sa liste de découpes appliquées (CUT), ainsi que des zones interdites de coupe (**NoCutZones**). Elle entretient une relation d'agrégation avec la classe **NoCutZone** et la classe abstraite **Cut**.

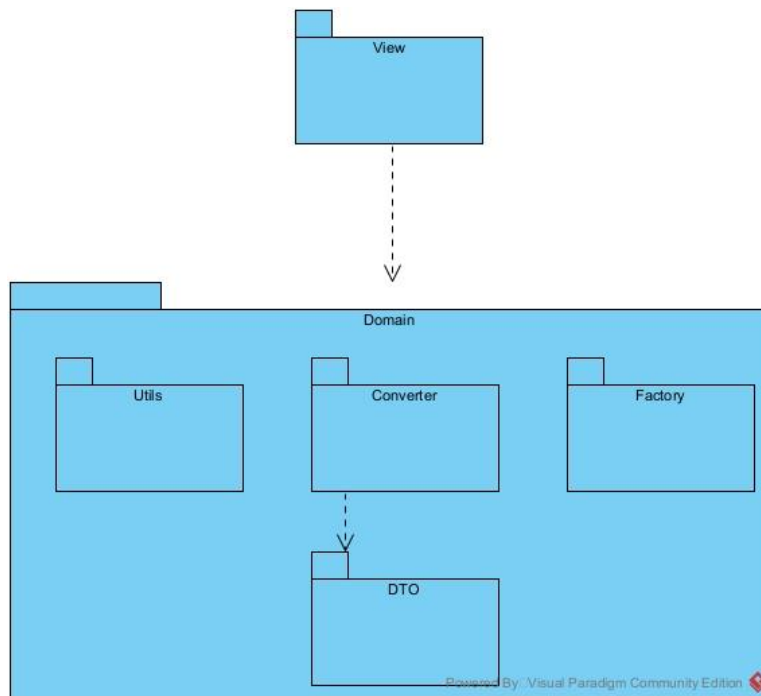
La classe **NoCutZone** représente une zone dans le panneau où aucune découpe ne peut être effectuée. Elle contient des dimensions et des coordonnées et est associée à la classe **Panel** via une agrégation. C'est la classe **CNC** qui crée les zones interdites via la classe **Panel**.

La classe abstraite **Cut** représente une découpe sur un panneau et est définie par des attributs tels que la profondeur, l'outil utilisé et un identifiant unique. Les sous-classes qui héritent de cette classe incluent les différents types de découpe : **RegularCut**, **IrregularCut**, et **ReCut**, pour refaire une découpe via son identifiant. **RegularCut** inclut les coupes droites (**StraightCut**) et les coupes parallèles (**ParallelCut**), tandis que **IrregularCut** inclut les coupes rectangulaires (**RectangularCut**) et en forme de L (**LShapedCut**). Les coupes irrégulières se distinguent par l'utilisation de points d'intersection pour former des découpes plus complexes.

Le **Controller** a accès à tous les convertisseurs de données, à savoir **PanelConverter**, **ToolConverter**, **CutConverter**, et **NoCutZoneConverter**. Chaque convertisseur a accès à son objet de transfert de données respectif (DTO) : **PanelDTO**, **ToolDTO**, **CutDTO**, et **NoCutZoneDTO**. Il est important de noter que **CutDTO** est abstrait et inclut des DTO pour chaque type de découpe (par exemple, **ReCutDTO**, **StraightCutDTO**, **LShapedCutDTO**, **ParallelCutDTO**, et **RectangularCutDTO**).

Enfin, la **CutFactory** est un patron usine, c'est-à-dire une classe qui crée des objets. On utilise ce patron pour créer toutes les découpes afin que le **Controller** puisse les ajouter à la CNC pour découper le panneau.

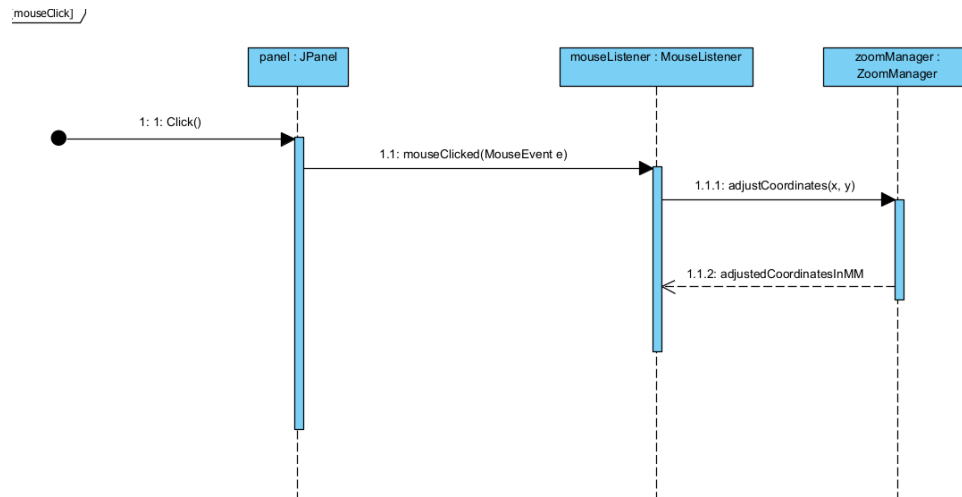
## Architecture Logique



Le package View correspond à la couche de présentation de notre application, c'est-à-dire l'interface utilisateur, et dépend du package Domain pour fonctionner. Le package Domain regroupe tous les éléments principaux pour faire fonctionner notre application. Il est composé de plusieurs sous-packages. Premièrement, il y a le package Utils, qui contient des classes utilitaires que toutes les classes dans le domaine utilisent. Le package Converter contient toutes les classes qui permettent de convertir soit des données en objet de transfert de données en utilisant le package DTO ou de reconvertir ces objets en données, pour faciliter le transfert de données entre les différentes classes du domaine. Finalement le package Factory contient des patrons d'usine, des classes qui permettent de créer différents objets similaires.

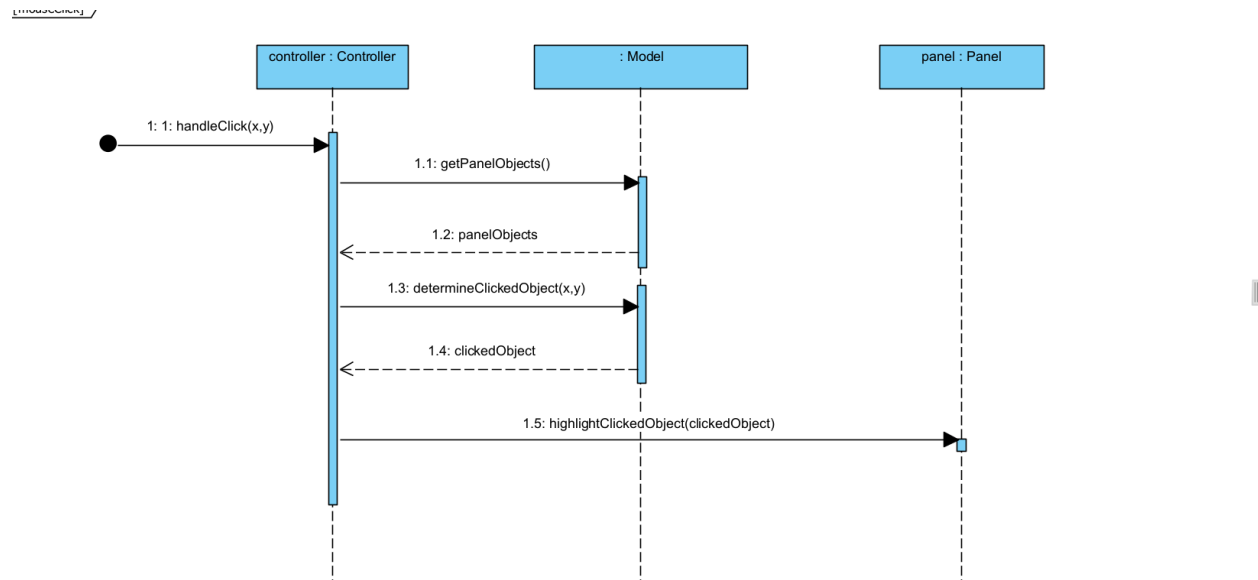
## Diagramme de séquence de conception

### 3.1.1



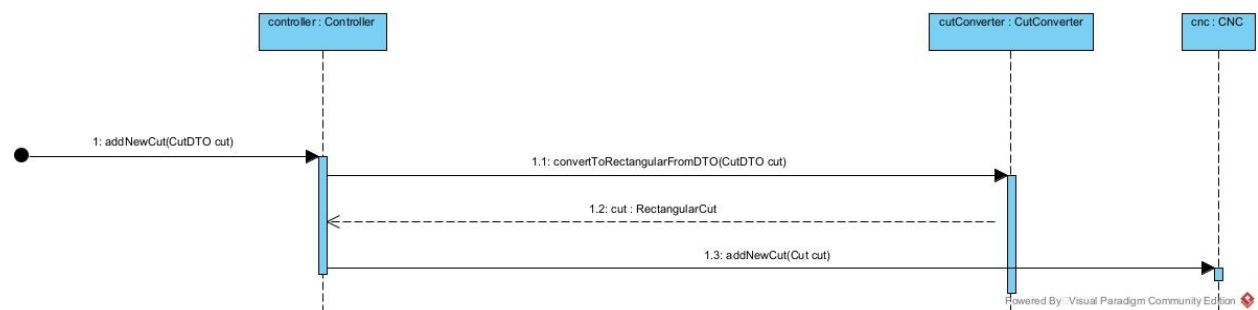
Ce diagramme de séquence montre le processus de gestion d'un clic de souris dans une interface graphique utilisant un JPanel. Lorsqu'un utilisateur clique sur l'interface, le MouseListener de Java capte l'événement de clic de souris. Ensuite, le MouseListener envoie les coordonnées du clic (en pixels) au ZoomManager, qui ajuste les coordonnées en fonction du niveau de zoom en vigueur. Le ZoomManager retourne les coordonnées ajustées en millimètres.

### 3.1.2



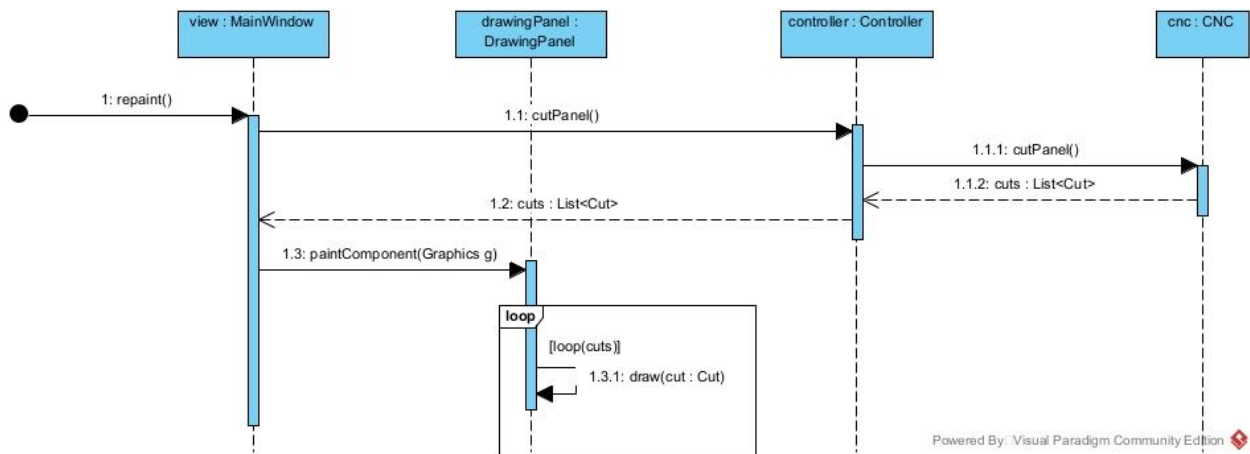
Ce diagramme de séquence illustre comment le Contrôleur gère les coordonnées de clic transmises par le `MouseListener`. Le Contrôleur commence par récupérer la liste des objets du panneau via le Modèle. Il utilise ensuite les coordonnées en millimètres pour déterminer quel objet ou élément a été cliqué. Une fois que l'objet cliqué est déterminé, le Contrôleur envoie un message à la Vue pour mettre à jour l'affichage en surlignant ou en sélectionnant l'objet en question.

### 3.2



Le contrôleur est appelé pour ajouter une nouvelle coupe. La coupe passée en paramètre est un DTO `coupeRectangulaire`. Le contrôleur appelle ensuite le `CutConverter` pour qu'il génère un objet `RectangularCut`. Cet objet coupe `RectangularCut` est ensuite retourné au contrôleur. Le contrôleur le passe ensuite dans le domaine de la CNC.

### 3.3



Un évènement Swing déclenche l'appel de repaint du MainWindow. MainWindow va ensuite appeler le contrôleur pour obtenir les coupes qui sont contenues dans le domaine. Le contrôleur appelle donc la CNC pour obtenir sa liste de coupes. La liste de coupes sera ensuite lue par le MainWindow qui les renverra vers le DrawingPanel pour que les coupes soient affichées.



## Gantt mis à jour

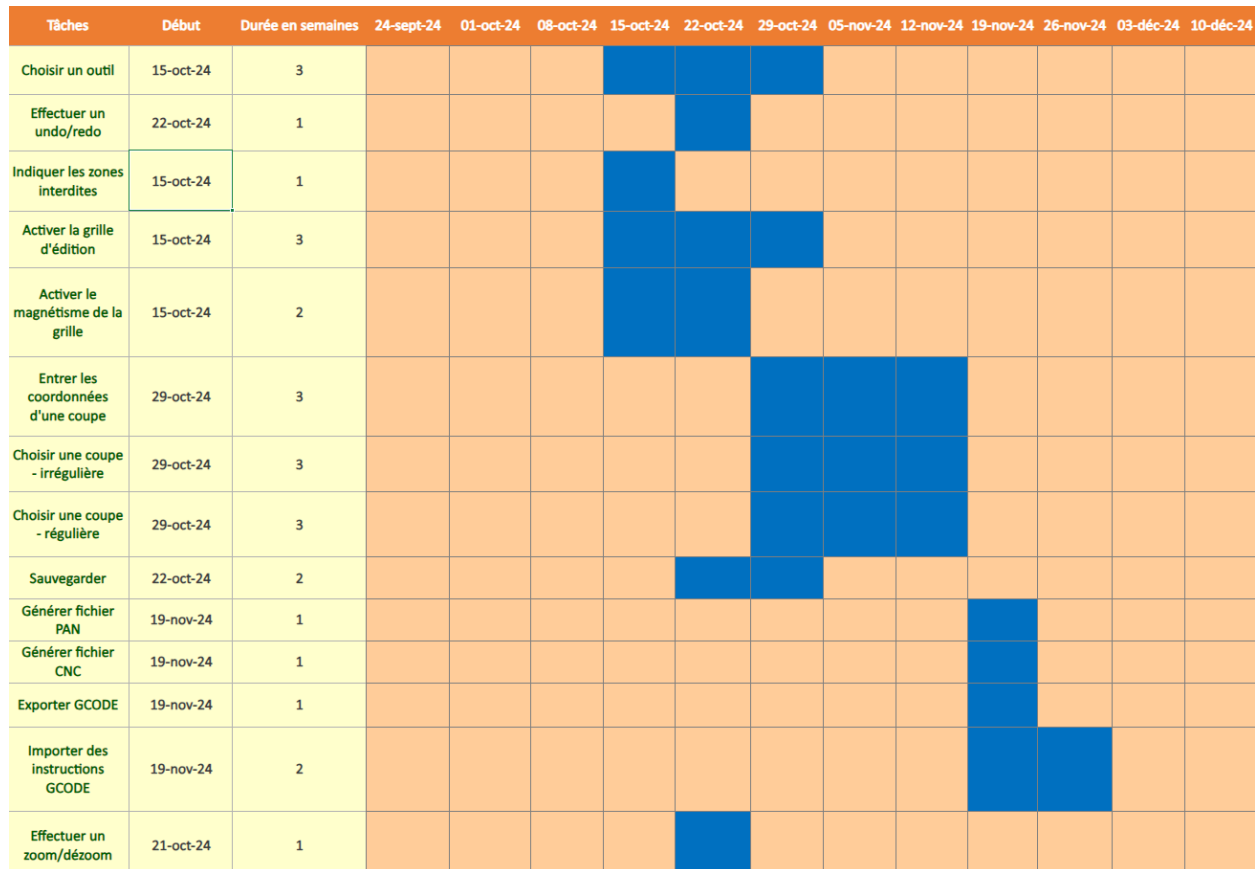


Figure 3:Diagramme de Gantt

## Contribution des membres de l'équipe

**Gabriel Rousseau : Diagramme de classe**

**Mathieu Roussel : Squelette d'interface / Squelette du code**

**Samuel Vézeau: DSC 3.2 et 3.3, architecture logique**

**William Bourque: Gantt, DSC**

**William Gingras: Diagramme de classe**