



PlanIFTicateur

Travail pratique 2

GLO-2004 – Génie Logiciel Orienté Objet – Hiver 2015

Travail présenté à
M. Jonathan Gaudreault

Chayer, Philippe	Philippe.chayer.1@ulaval.ca IFT	PHCHA47
Khouma, Abdou	abdou.khouma.1@ulaval.ca GIF	ABKHO9
Gadoury, Gabriel	Gabriel.gadoury.1@ulaval.ca IFT	GAGAD1
Yeo, Clotioloman	Clotioloman.yeo.1@ulaval.ca GLO	CLYEO1

Table des matières

Introduction	3
Interfaces utilisateurs	4
Fenêtre principale	5
Fenêtre statistiques	6
Diagramme des classes de conception	7
Diagramme de package.....	9
Diagrammes de séquence.....	10
Indicateur du nombre de cours par jour.....	10
Affichage de la grille horaire	12
Sélection/Déplacement d'un cours déjà sur la grille horaire.....	Error! Bookmark not defined.
Sélection d'un cours dans la grille horaire avec l'aide de la souris / déplacement cours dans la grille horaire.....	Error! Bookmark not defined.
Annexe	17
Modèle du domaine.....	17
Modèle des cas d'utilisation	18
Glossaire.....	19
Gestion de projet	23

Introduction

La construction des horaires de session est une tâche difficile pour les directions de programme. À l'aide du logiciel dont nous avons le mandat, cette tâche deviendra visuelle et interactive, donc plus facile à réaliser. Ainsi, la direction économisera beaucoup de temps et, par le fait même, diminuera significativement le coût associé à la production d'un horaire optimale.

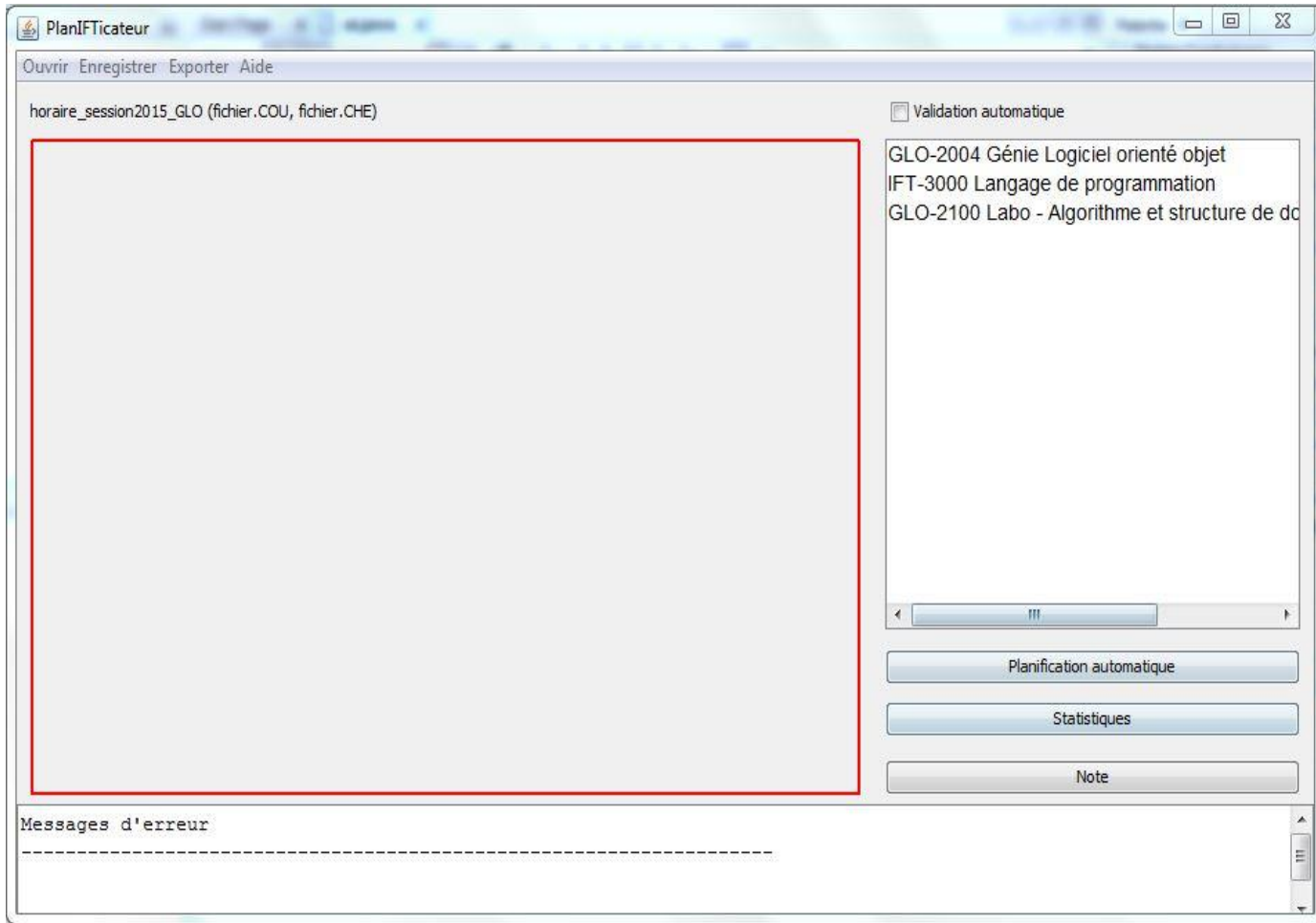
Ce rapport consiste à présenter le modèle de conception et l'architecture logicielle de planIFTicateur. L'analyse sera basée principalement sur les éléments préalablement réalisés.

La première section illustre les principales interfaces utilisateurs, qui permettent à l'utilisateur d'interagir avec le système. Par la suite, un diagramme de classes conceptuelles est présenté. Il s'agit de l'architecture qui sera utilisée afin d'implanter notre solution sous le langage Java. Ensuite, une section est consacrée à illustrer la communication entre les différents regroupements de paquets du programme. Nous poursuivrons en détaillant les diagrammes de séquence qui permettront de mieux visualiser les fonctionnements particuliers de PlanIFTicateu et une dernière section mettra à jour la gestion du projet ainsi que les éléments du rapport précédent.

Interfaces utilisateurs

Tout d'abord, une interface utilisateur doit miser sur l'aspect ergonomique. Tout en respectant les critères du client, nous avons réussi à rendre simple et intuitif l'interface graphique de notre programme. Les informations généralement recherchées par les utilisateurs sont concentrées au haut de la fenêtre, tandis que les autres sont positionnées de manière à respecter les contraintes du client. Une fenêtre pour la grille horaire encadrée par une couleur vivante et représentative de l'état du travail indiquera à l'utilisateur s'il y a un problème ou si tout va bien, un champ au bas communiquera avec précision les différentes erreurs au besoin ou encore des messages de confirmation et une barre de menu semblable à toutes les barres représentées dans les différents logiciels connus de tous. Voyez maintenant les différents croquis :

Fenêtre principale

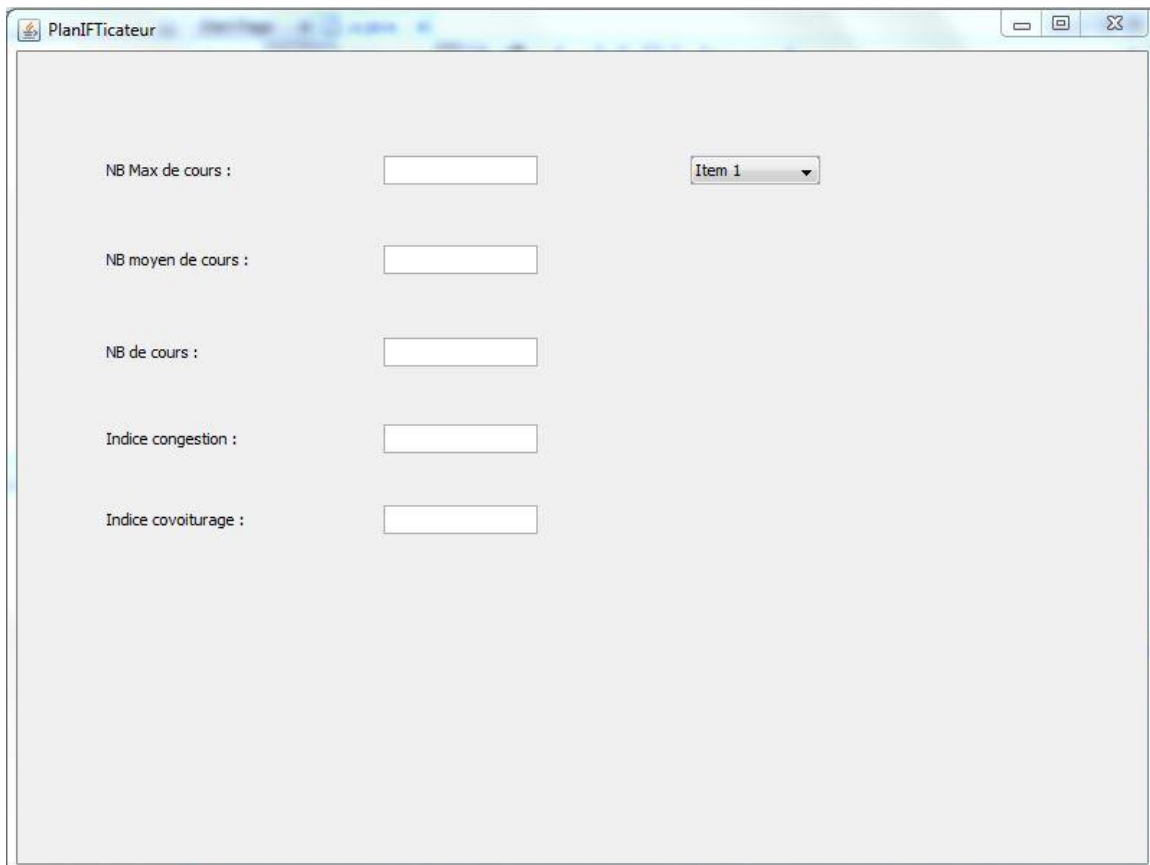


La fenêtre principale est composée des parties suivantes :

- Une barre de menu au haut de l'écran, permettant l'accès à des fonctionnalités triviales;
- Une grille horaire qui suivra la même logique, c'est-à-dire simple et intuitive (Notez qu'elle n'est pas dessinée ici, elle sera dans l'encadré rouge);
- Une liste des activités non attribuées (situé à droite);
- Un champ pour l'indication d'erreurs ou de confirmation (au bas);

- Un bouton *planification automatique* permettant de faire l'essai de la fonctionnalité de planification automatique de l'application;
- Un bouton *note* permettant à l'utilisateur de prendre des notes;
- Un bouton *statistiques* dont le contenu est détaillé ici-bas.

Fenêtre statistiques



NB Max de cours : Item 1 ▼

NB moyen de cours :

NB de cours :

Indice congestion :

Indice covoiturage :

La liste des statistiques est automatiquement affichée, et l'utilisateur peut changer le jour désiré à l'aide d'un menu déroulant.

Diagramme des classes de conception

Le diagramme qui suit sera notre bible pour la phase de création. Nous proposons une solution axée sur la clarté et la compréhension, ce qui facilitera la maintenance du logiciel. Le diagramme ci-dessous est considéré comme étant le diagramme de classes conceptuelles mais également comme étant le diagramme de packages. Nous avons jugé juste de représenter ces deux diagrammes en un seul.

(Schéma sur page suivante)

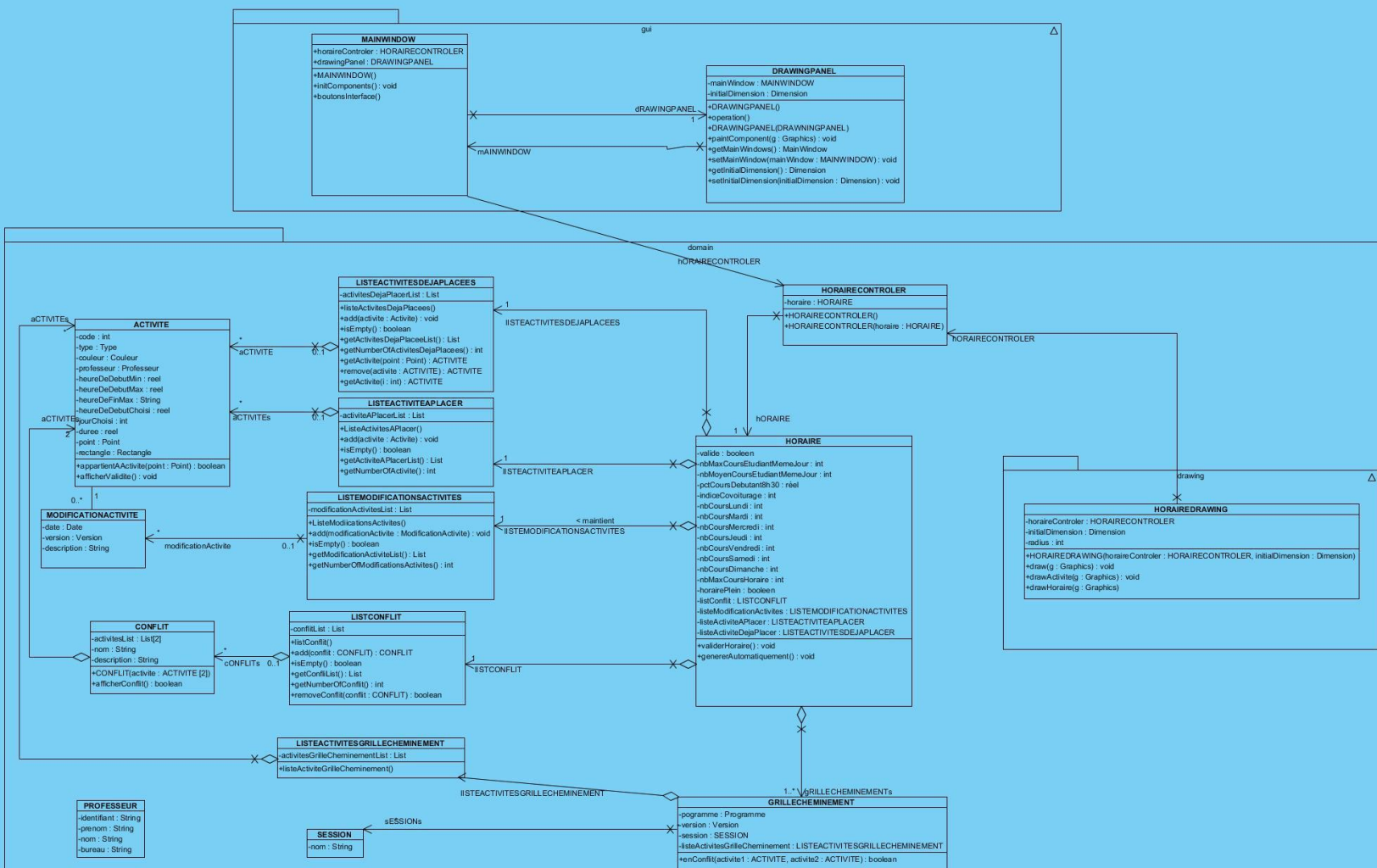


Diagramme de package

La communication entre l'interface graphique et la logique métier (le domaine) se fait par le billet du contrôleur de Larman. Ce dernier reçoit de l'interface graphique des objets primitifs de type *Int*. Il sert d'articulation entre l'interface et le domaine s'occupe de faire les bons appels dans le domaine.

L'utilisateur appelle le **paintComponent** du **drawingPanel**. Cette méthode est chargée d'initialiser les dimensions de la zone de la grille horaire. Cette classe appartient au paquet de l'interface graphique utilisateur (**GUI**).

Puis le **paintComponent** crée un objet de type **HoraireDrawing** qui possède un objet de l'**HoraireController**.

Par le billet de ce contrôleur qui a un objet de type Horaire alors le **HoraireDrawing** dispose de toute l'information nécessaire afin de dessiner un horaire.

Dans l'architecture, on pourra observer que plusieurs listes sont considérées en tant qu'objet. Ainsi, le contrôleur pourra accéder à ces listes avec facilité: **ListeActiviteAPlacer**, **ListeActivitesDejaPlacees**, **ListConflit**, **ListeModificationsActivites**, etc.

HoraireDrawing est dans le package de **drawing** mais appartient à la couche logique. Un objet graphique passe de l'objet **drawingCadrePanel** vers le **HoraireDrawing**.

Ce dît objet permet de dessiner les activités qui sont en fait des rectangles. La largeur de ce rectangle correspond à la durée de l'activité et la longueur (hauteur) reste la même pour toutes les activités. Les points d'intersection de la longueur et de la largeur (une droite) correspondent au début de l'activité. De même que, la hauteur des panneaux reste constante et permet de bien séparer chaque jour de la grille. Finalement, chaque journée contient plusieurs colonnes correspondant aux heures pertinentes pour le placement des activités.

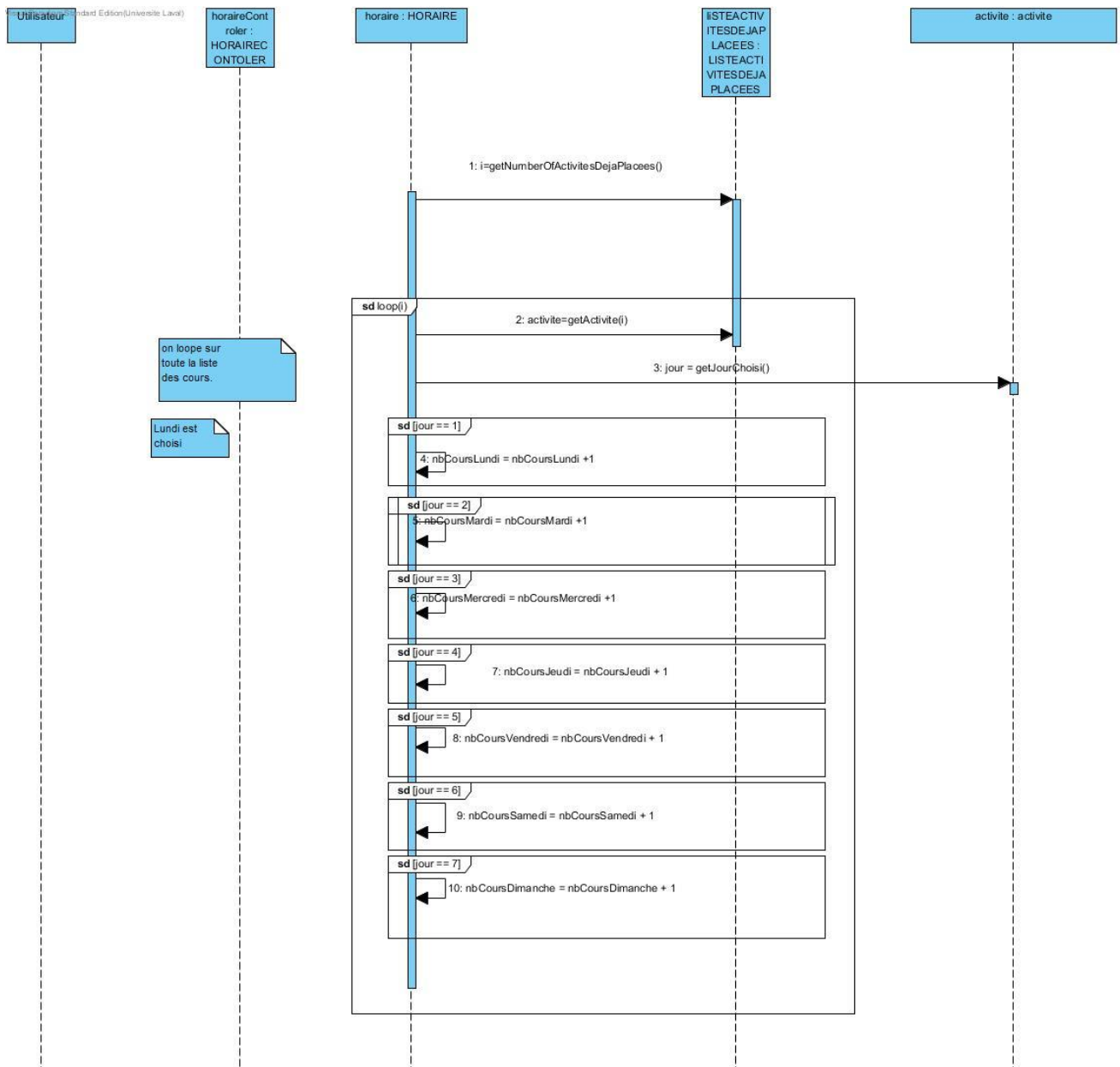
Diagrammes de séquence

Indicateur du nombre de cours par jour

Afin d'obtenir cet indicateur, la classe **Horaire** fait la demande à la classe **ListeActivitesDejaPlacees** du nombre de cours déjà placé à l'horaire et l'enregistre dans un tableau **i**. Une fois cette opération faite, Une boucle permet d'incrémenter la variable **nbCoursJournee** appropriée (voir ici **nbCoursLundi**, **nbCoursMardi**, etc.)

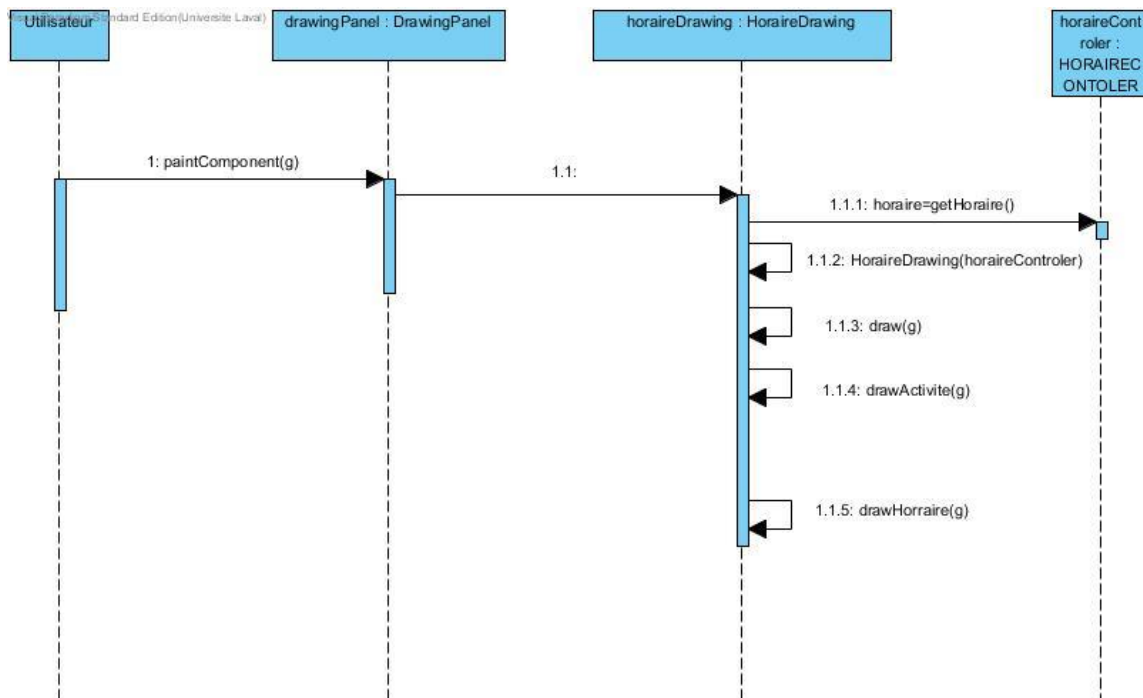
En effet, **Horaire** demande à la classe **ListeActivitesDejaPlacees** l'identification du premier cours déjà placé, fait une demande à la classe **Activite** pour connaître la journée attitrée à ce cours (sous forme de nombre entier) et l'enregistre sous la variable **jour**. Ensuite, comme mentionné ci-haut, **Horaire** incrémente la variable appropriée selon la variable **jour**. Une fois cette opération complétée, on incrémente l'index du tableau **i** et on répète la boucle jusqu'à ce que l'on ait traversé le tableau dans son entièreté.

Par la suite, la variable désirée pourra être cueillie avec une simple commande demandant la valeur de **nbCoursJournee** appropriée afin de l'intégrer aux statistiques ou autres.



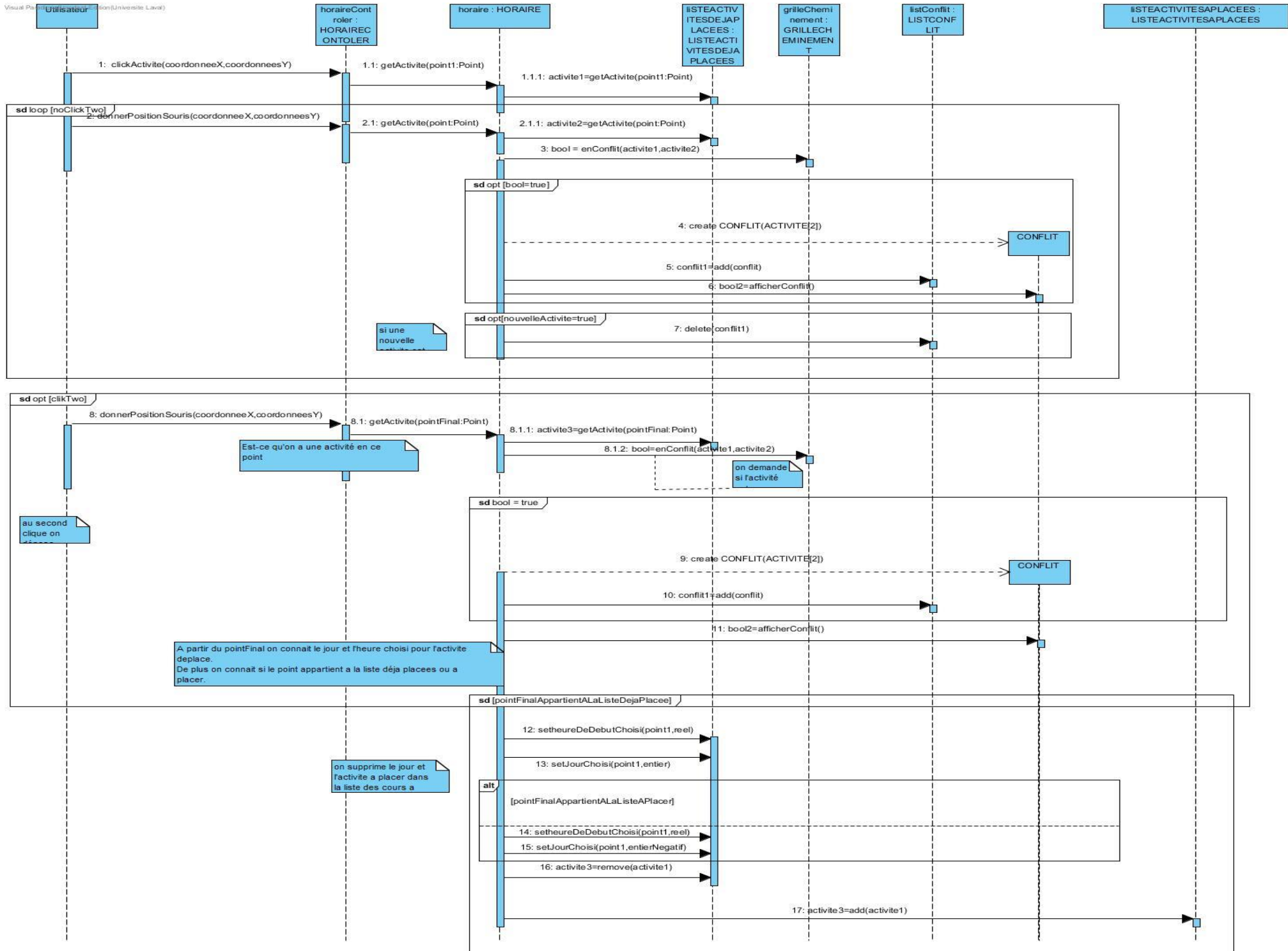
Affichage de la grille horaire

Lorsque la grille d'affichage est appelée par l'application, une demande est faite à la classe **DrawingPanel** pour générer une fenêtre avec les bonnes dimensions. Ensuite, une demande à la classe **HoraireDrawing** afin de créer les éléments graphiques nécessaires à la présentation visuelle de la grille horaire. Cette même classe demande à son tour au contrôleur (**HoraireController**) de lui indiquer l'objet horaire correspondant pour la suite. Notez bien que les éléments graphiques se généreront à la suite de cette dernière demande. Finalement, la classe **HoraireDrawing** pourra dessiner créer un visuel de l'horaire en cours de création et apposer ces éléments graphiques sur la grille générée.



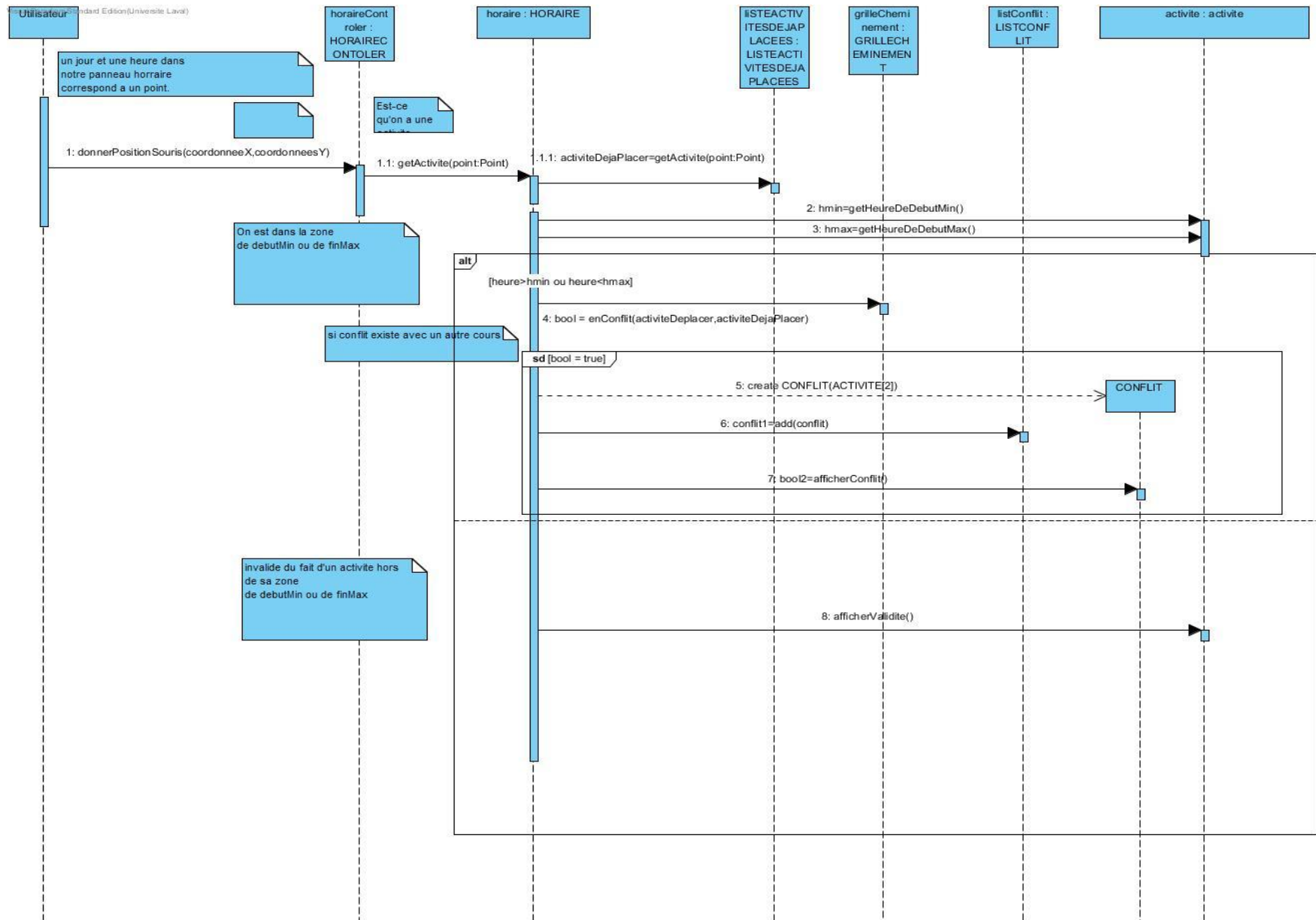
Sélection d'un cours dans la grille horaire avec l'aide de la souris / déplacement cours dans la grille horaire

L'utilisateur peut, en tout temps, cliquer sur une activité dans l'interface du système via la classe **HoraireController**. Cette dernière utilise la classe **Horaire** pour charger en RAM les informations de l'activité concernée, question de se préparer à un éventuel traitement. L'utilisateur peut ensuite choisir un emplacement pour l'activité dans la grille horaire. Le système, quant à lui, avec l'aide des classes auxiliaires, vérifie s'il y a un conflit ou une contrainte avec l'horaire actuel. Si ce n'est pas le cas, l'activité est déplacée.



Validation qu'un jour / heure de début est valide pour un cours donné

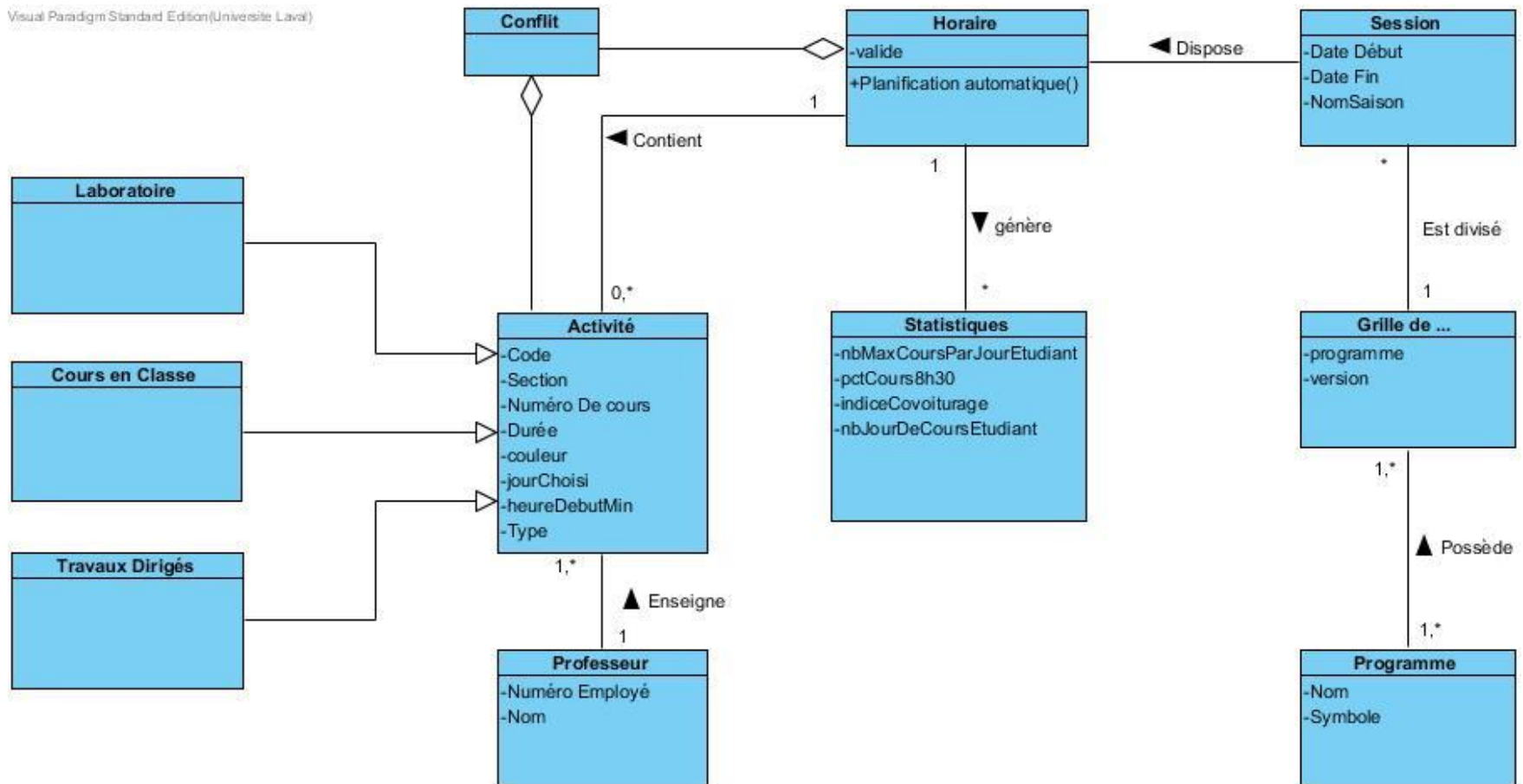
Lorsque l'utilisateur choisi une position pour une activité dans l'horaire, le système, via la classe **HoraireController**, doit étudier son choix avant de le valider. Pour cela, l'utilisateur fourni la date désirée. Le système vérifie d'abord par la classe **Horaire** si la plage horaire est libre en accédant à **listeActivitesDejaPlacees**. Si la plage est libre, il vérifie dans un second temps si le choix respecte la grille de cheminement courante. Lorsque toutes ces contraintes sont respectées, le nouveau positionnement de l'activité est enregistré.



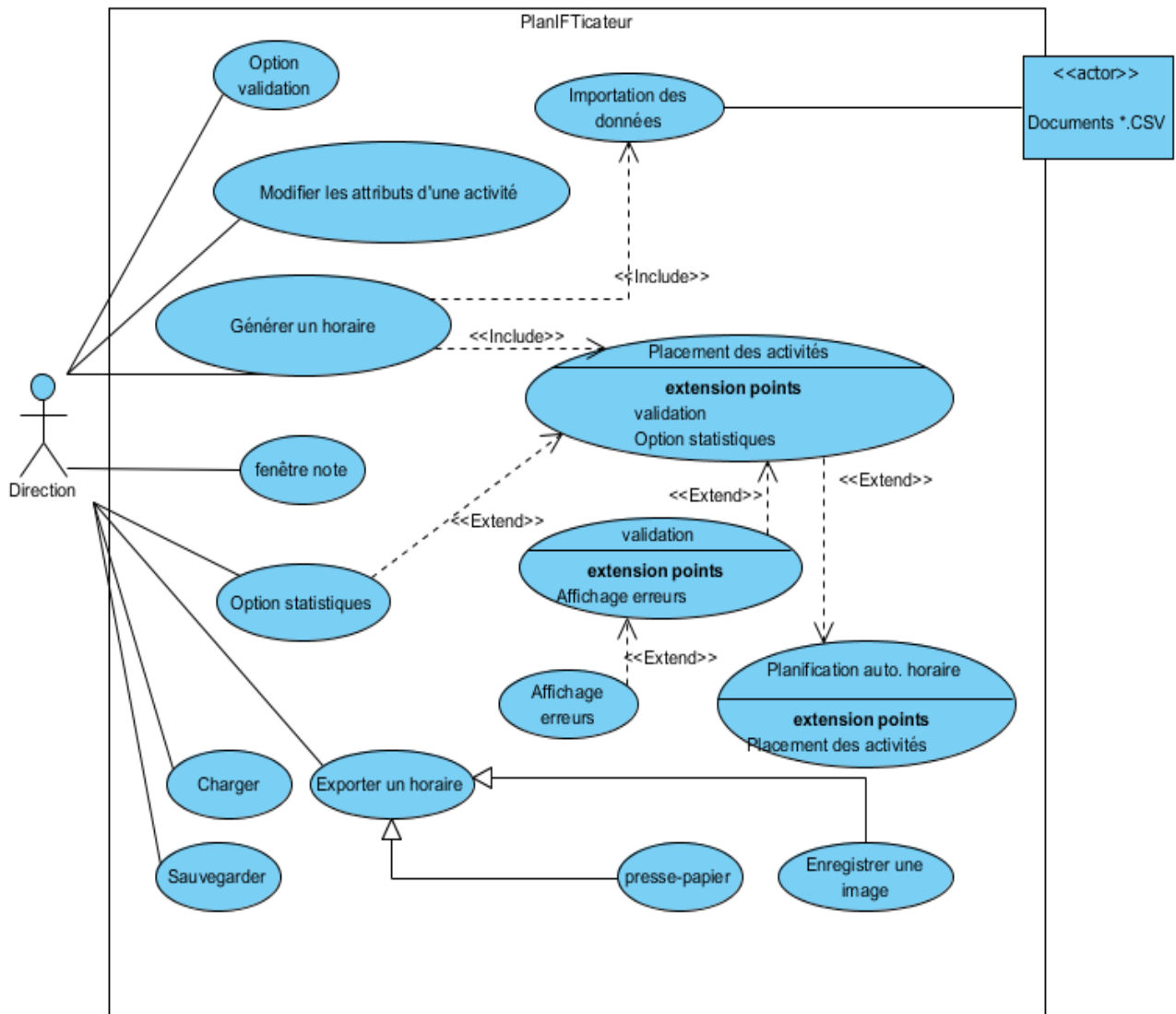
Annexe

Modèle du domaine

Visual Paradigm Standard Edition (Université Laval)



Modèle des cas d'utilisation



Glossaire

Algorithme

Suite de règles permettant de résoudre un problème.

Application

Programme ou ensemble de programmes visant à aider un utilisateur d'un ordinateur dans le traitement d'une tâche précise. *(Réf. : Larousse.fr)*

Cas d'utilisation

Relation établie entre une fonctionnalité et, soit un acteur ou une autre fonctionnalité.

Chemin

Emplacement précis sur un disque dur défini par la suite ordonnée des dossiers pour avoir accès au fichier recherché.

Diagramme de séquence système (DSS)

Représentation des interactions chronologique entre un système et ses acteurs.
(réf. : Wikipédia)

Domaine d'affaires

Référent pour l'ensemble des processus d'affaires d'un projet ou d'une organisation. Englobe les entités, les acteurs et autres participants de ces processus.

« Drag and drop »

(Terme anglais) En informatique, processus durant lequel un usager d'un ordinateur sélectionne un objet, au moyen d'une souris, en maintenant enfoncé le bouton gauche de cette dernière afin de déplacer le dît objet et de le relâcher à l'endroit voulu en relâchant également le bouton de sa souris.

Fenêtre

Zone d'affichage d'informations d'un programme.

Fonction

Bloc d'une séquence d'instructions visant un but, une fonctionnalité précise.

Logiciel

Ensemble d'instructions et de règles interprétable par un ordinateur.

Multiplateforme

Fonctionnant sur plusieurs plateformes, soit plusieurs ordinateurs/systèmes d'exploitation différents.

Multiutilisateur

Offrant la possibilité à plusieurs usagers d'interagir simultanément sur le même logiciel.

PlanIFTicateur

Nom du logiciel en développement pour le projet contenu dans ce document.

Plateforme PC

Liaison entre un ordinateur personnel munit d'un processeur spécifique et du système d'exploitation *Windows*.

Presse-papier

Fonction intégrée dans tous les systèmes d'exploitation stockant des données que l'on souhaite déplacer ou copier.

Programme

Succession d'instructions qu'un ordinateur peut exécuter afin d'accomplir des opérations.

Serveur dédié

Système informatique dont l'ensemble des ressources est dédié à un seul utilisateur.

Serveur web

Système informatique qui a pour fonction la publication de sites web à la demande d'un autre système.

Session

Période de 15 semaines durant laquelle un étudiant de l'université doit accomplir les objectifs de ses cours.

GUI

Signifie : **G**raphical **U**ser Interface, ce qui représente l'interface graphique utilisateur. C'est ce qui est affiché par le programme.

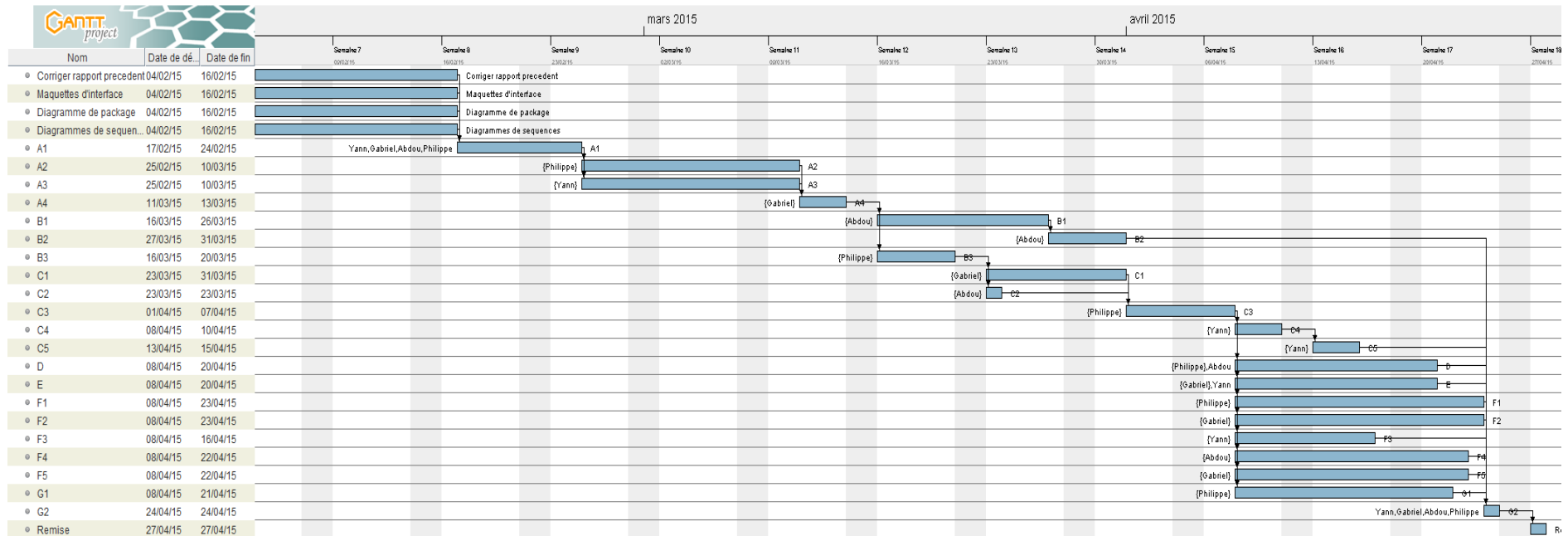
Diagramme de package

Permet de représenté l'architecture des différents groupements de classe en différentes couches logique.

Diagramme de séquence

Diagramme illustrant ce qui se produit lors de certaines actions. Permet de comprendre la communication entre les classes.

Gestion de projet



Voir diagrammeDeGant.gan