**RAPPORT - Résolution du PCC-FT**

**Présentation du problème :**

Le problème ce présente sous la forme d'un algorithme dynamique de résolution de plus court chemin avec fenêtre de temps, et ce à implémenter en C++.

Il faudra aussi implémenter diverse variantes de l'algorithme pour faire des tests statistiques pour comparer les performances des diverses variantes.

Le jeu de graphe lu par l'algo étant lui aussi varié.

**Choix liés à la programmation :**

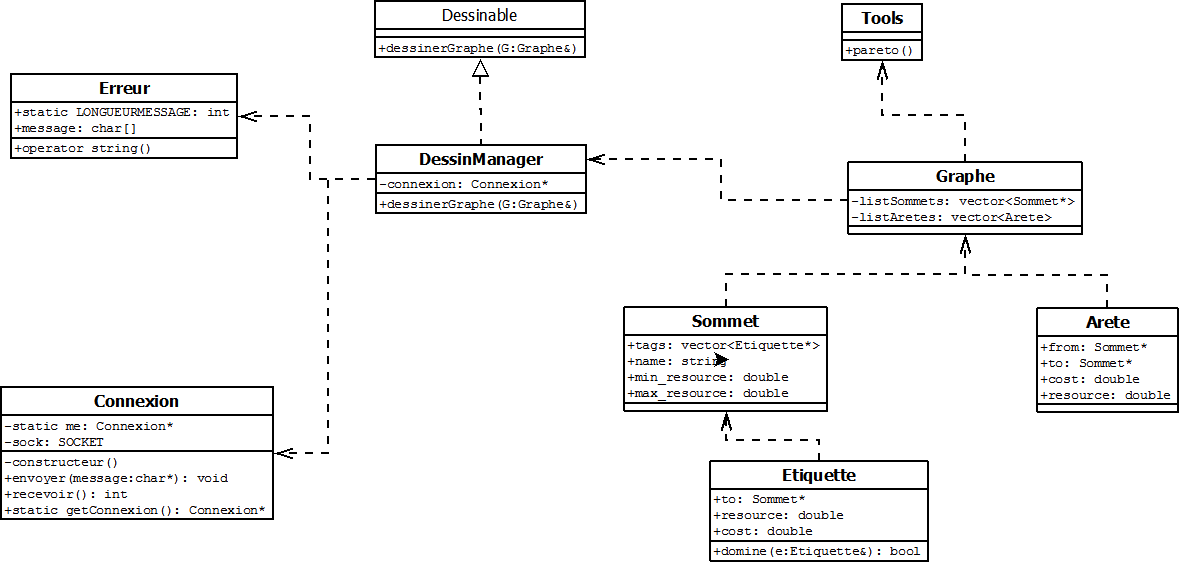
Nous avons décider de modéliser sous forme d'objets le graphe à parcourir. Cela sera détaillé dans la partie "structures de données utilisées".

Nous avons permis la lecture et la sauvegarde d'un graphe depuis/vers un fichier texte.

Nous avons également ajouter un serveur de dessin Java permettant de modéliser le graphe saisi/lu.

Après utilisation de l'algo sur le graphe, le chemin optimale de sortie est retourné à l'utilisateur dans la console, et affiché vers le serveur de dessin.

Les variantes se font sur le front et le meilleurs cout.

**Diagrammes de classes :**

**Structures des données utilisées :**

Les structures de données utilisées ont étaient modélisées par un objet Graphe composé lui-même par une liste de pointeur sur des Sommets, ainsi qu'une liste d'Arêtes.

Les objets sommets contiennent une liste d'Étiquettes représentant un couple de réel (coût / ressource) rattaché à un sommet.

**Information divers :**

La phase de compilation a été faite sous g++ avec l'option -lws2\_32 et -std=c++1y (ou -std=c++1y). Du fais que nous utilisons la bibliothèque WinSock2 de Windows pour le réseau, ainsi que des fonctionnalités C++ 11 et supérieur.

Nous avons ajouter au projet la fonctionnalité permettant de dessiner un graphe sur une application serveur basée sur Java.

Le serveur est fourni avec le projet, il suffit de lancer le fichier "serveur \_dessin.jar" dans une invite de commande avec "java -jar serveur\_dessin.jar " ; pour que le serveur se mette en écoute et attende de dessiner ce qu'on lui envoie. Pour l'envoie du graphe via l'application, il suffit de paramétrer le port et l'ip du serveur pour créer la connexion, l'envoie du graphe se fait via l'objet DessinManager qui possède une méthode prenant un Graphe en paramètre.

De plus, il est possible de passer en paramètre directement à l'exécutable produit, le chemin vers le graphe à charger. Et ainsi, le programme exécutera directement l'algo. Si jamais, le serveur de dessin java n'est pas lancé à l'exécution du programme, il ne dessinera tout simplement rien. Et si jamais aucun chemin n'est donné en paramètre à l'exe, alors il vous demandera de saisir le fichier à charger.

**Tests statistiques :**

