

Rapport de projet

UML

Logiciel de dessin assisté

SOMMAIRE

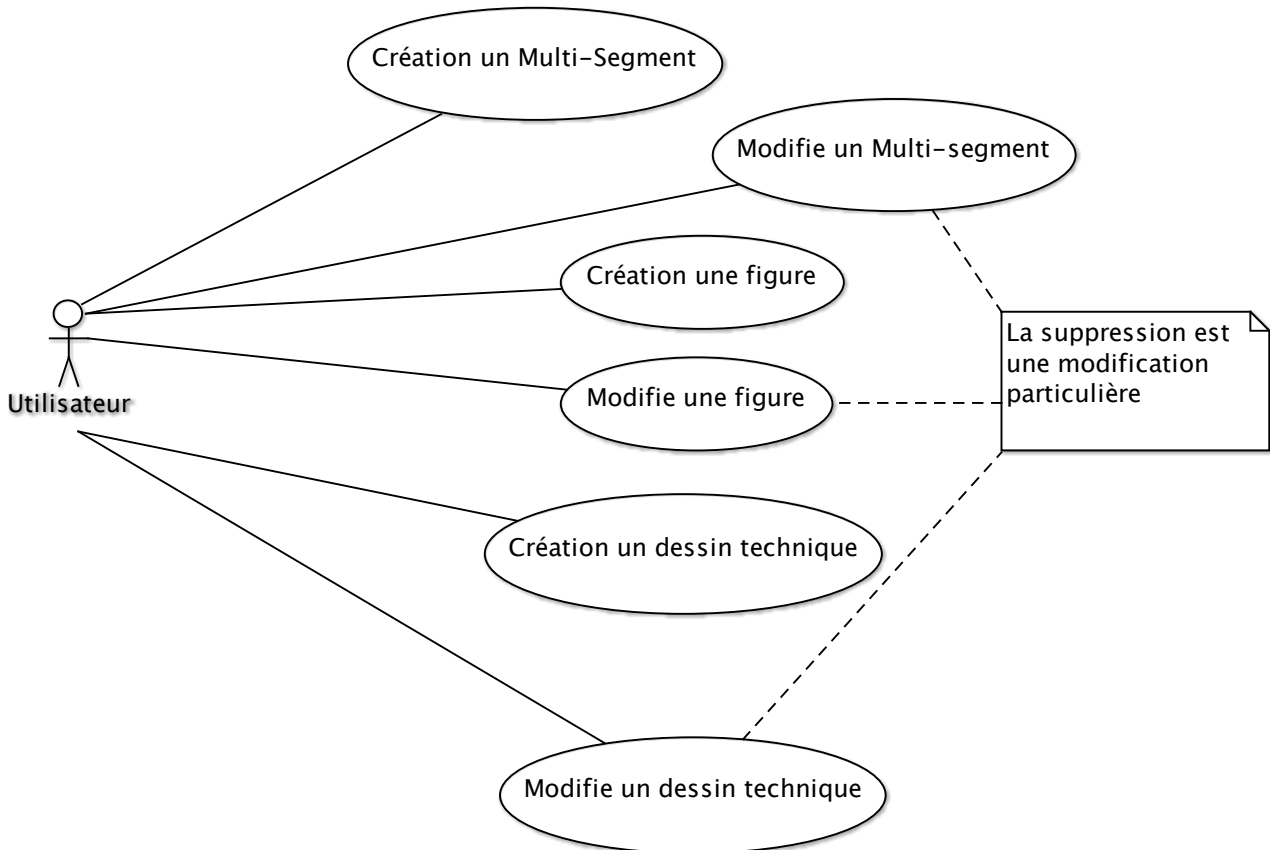
Introduction	4
II. Diagramme de cas d'utilisation	5
III. Diagramme de classe	6
IV. Diagramme d'objet	8
V. Diagrammes d'état transition	9
A) Création d'une figure	9
B) Création d'un multi-segment	10
C) Création d'un dessin technique	11
D) Modification d'une figure	12
E) Modification d'un multi-segment	13
F) Modification d'un dessin technique	14
VI. Diagrammes de séquence	15
A) La figure	15
B) Le multi segment	16
C) Le dessin technique	17
VI . Gestion Projet	18
VII. Conclusion	19

I. Introduction

Dans le cadre du projet UML/C++, nous devons modéliser des objets géométriques à deux dimensions. Ces objets sont répartis en deux catégories. Les objets dits de base tel que le triangle, le segment ou encore le losange. Et les objets dits composites, qui peuvent définir plusieurs objets de base manipulés comme une seule entité lors des actions d'affichage et de translation.

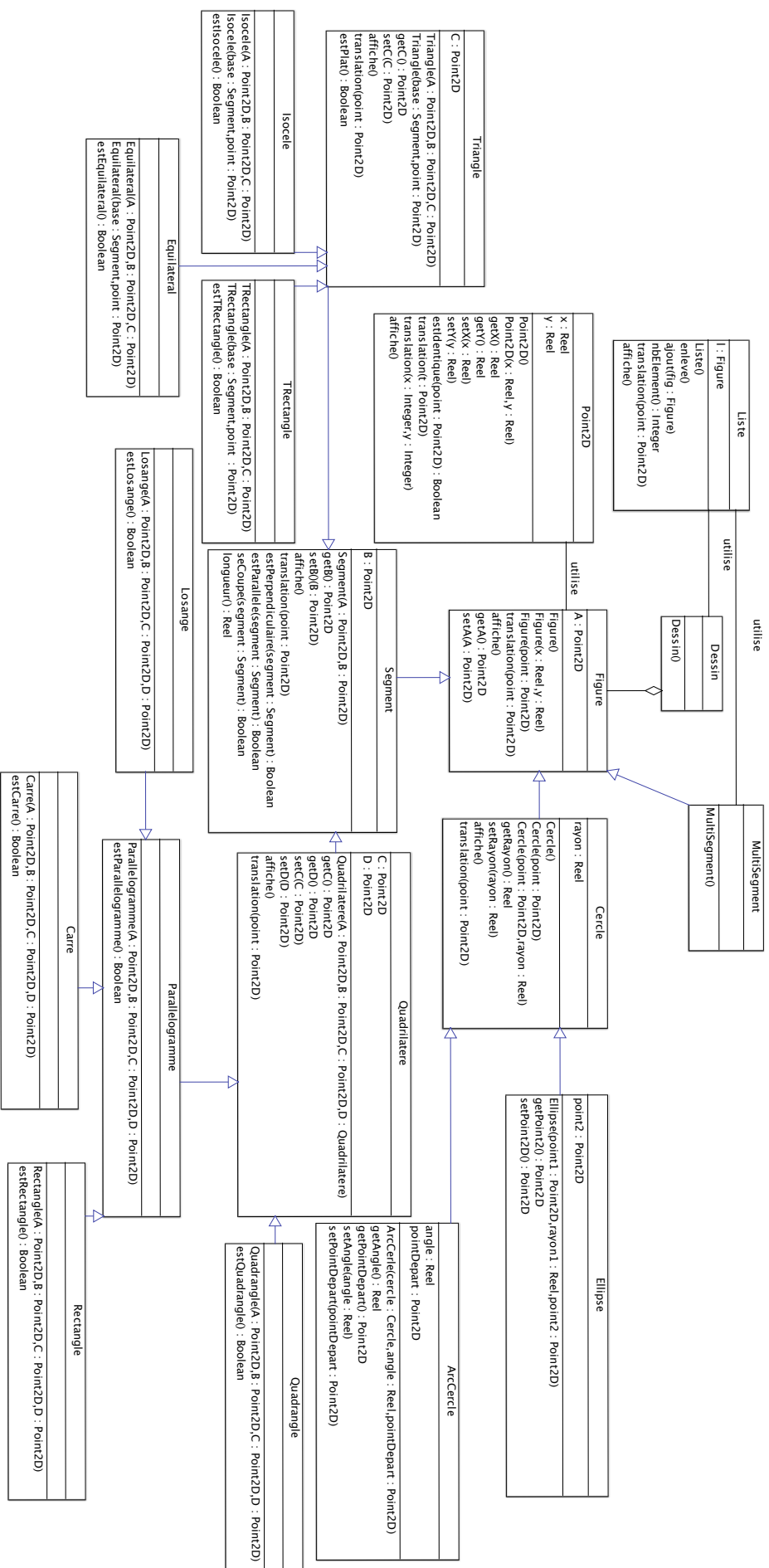
Dans ce rapport, nous tâcherons de montrer notre analyse du sujet et de justifier nos choix de la manière la plus rigoureuse possible. Pour cela, nous nous appuierons sur les différents diagrammes UML vu en cours.

II. Diagramme de cas d'utilisation



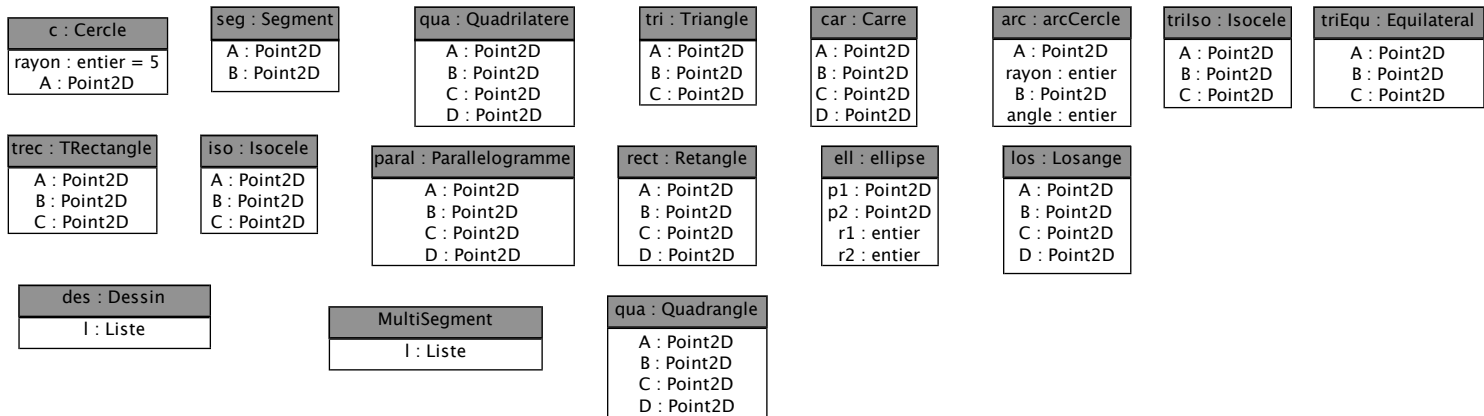
Dans notre analyse, nous avons déterminé six actions possibles. Ces actions interagissent avec les deux grands types d'élément de notre application. D'une part la figure, qui représente un élément simple comme le carré, le triangle ou encore le losange. D'autre part le dessin technique composé de différentes formes. Nous avons fait le choix de séparer le multi segment des figures, car celui-ci se compose d'un nombre indéfini de points. Pour chaque élément, nous avons deux actions distinctes. L'ajout et la modification. Nous sommes partis du principe que la suppression était en réalité une modification particulière. Ceci afin d'alléger le diagramme.

III. Diagramme de classe



A travers ce diagramme, nous pouvons voir toutes les classes que nous avons modéliser pour répondre aux contraintes du sujet. Nous remarquons la classe figure, cette classe représente la classe mère de tous les éléments. Elle regroupe notamment le point A qui correspond à l'origine de la figure. Cette classe abstraite nous permet de modéliser le dessin technique sous la forme d'une liste d'élément de type figure.

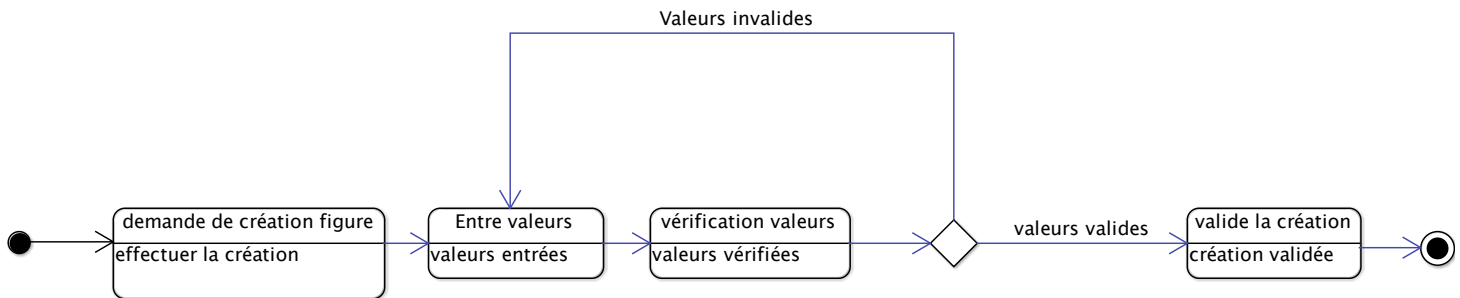
IV. Diagramme d'objet



Ici nous pouvons observer le diagramme d'objet que l'on a pu extraire du diagramme de classe. Nous y retrouvons tous les objets de notre application ainsi leurs attributs.

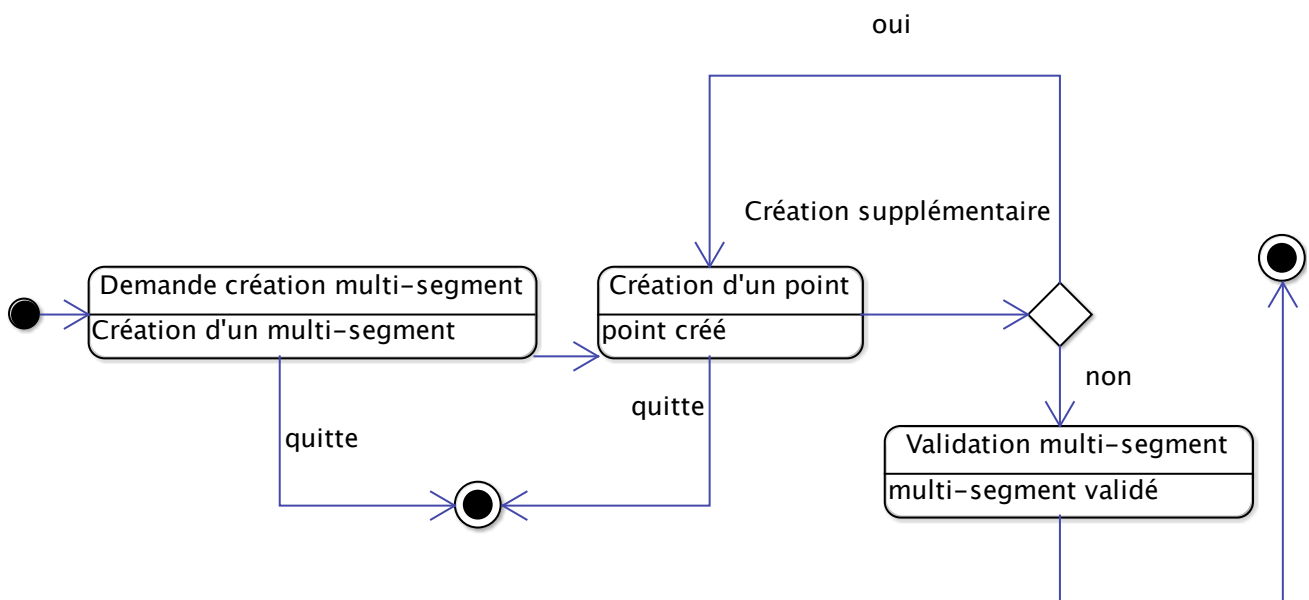
V. Diagrammes d'état transition

A) Création d'une figure



Ici, nous voyons le diagramme d'état transition pour la création d'une figure. La particularité de ce diagramme est la partie vérification des données entrées. En effet, nous vérifions la validité des données entrées par l'utilisateur, si il désire ajouter un triangle rectangle et que les valeurs renseignées caractérisent un triangle plat ou encore non rectangle, alors on notifie l'utilisateur et on invalide l'ajout de la figure.

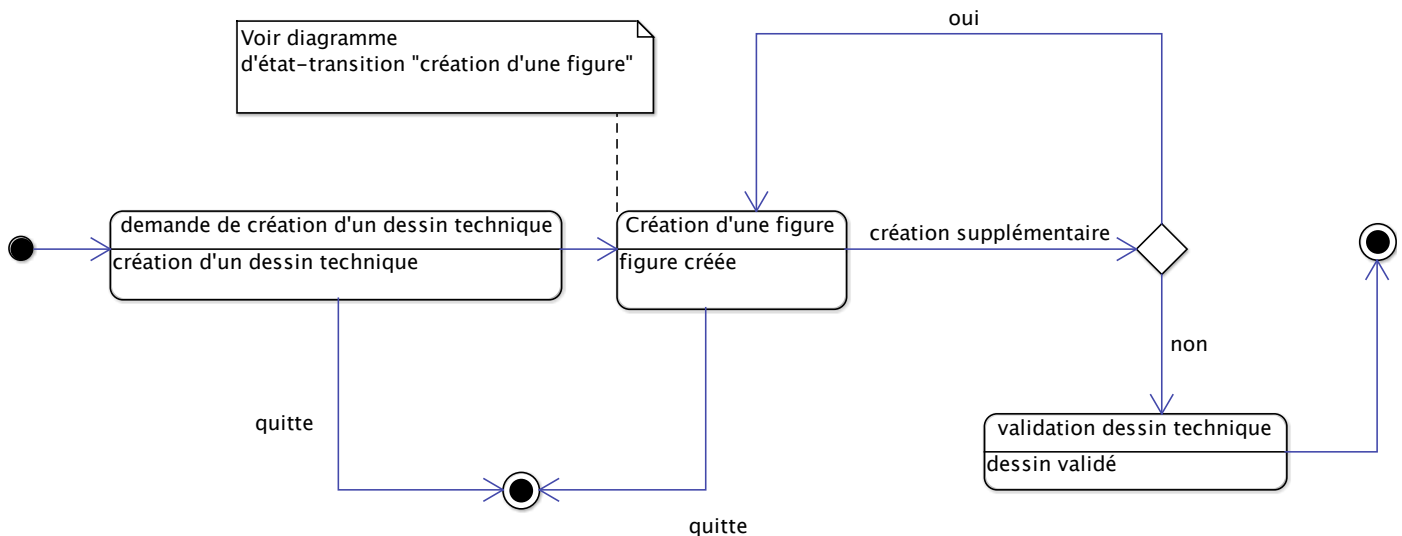
B) Création d'un multi-segment



Ce diagramme d'état transition représente la création d'un multi segment, l'utilisateur crée tout d'abord un point et ensuite il est libre de s'arrêter ou alors d'ajouter d'autres points.

En effet, le multi segment peut être composé de plusieurs points. Une boucle est mise en place pour pouvoir modéliser ce processus.

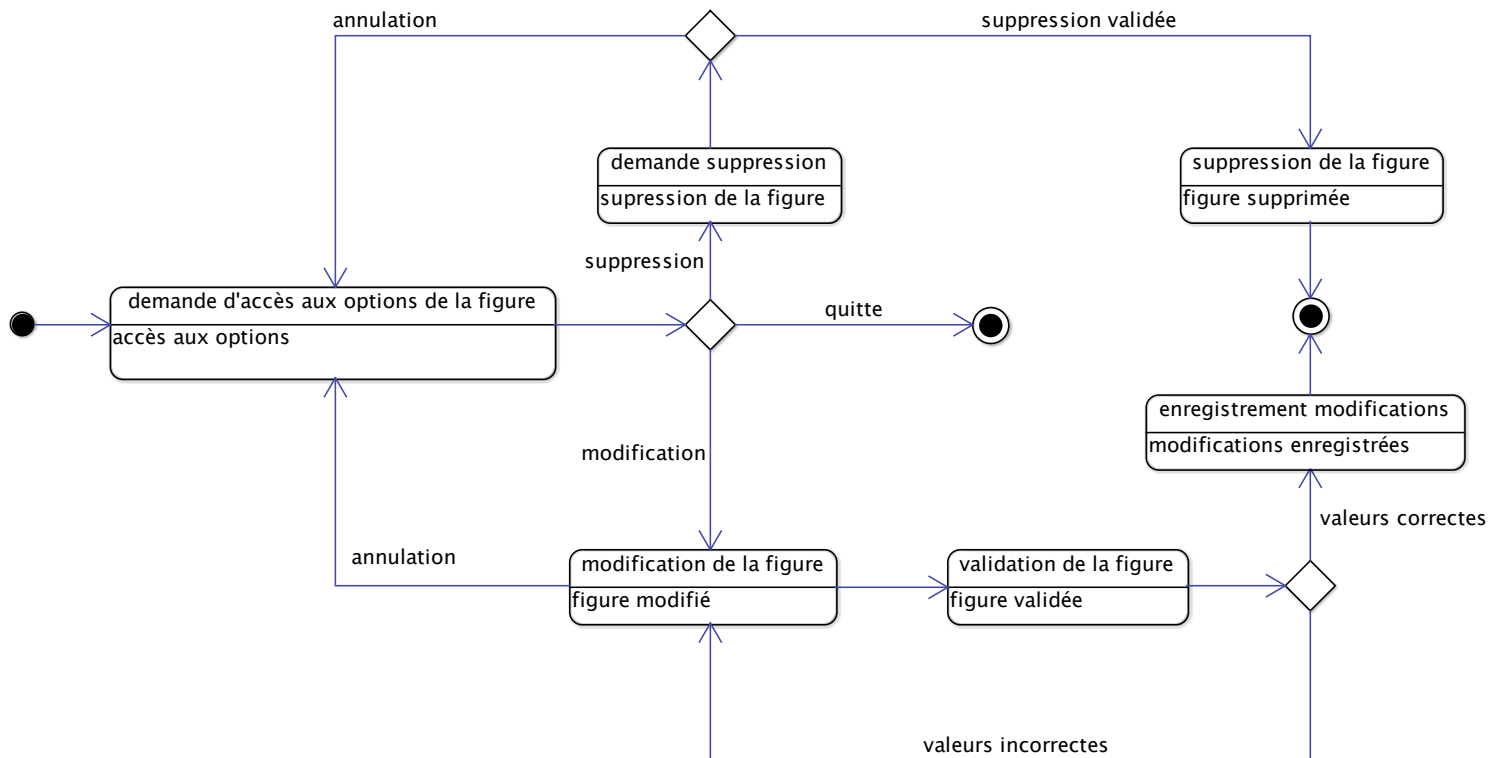
C) Création d'un dessin technique



Dans ce diagramme d'état transition représentant la création d'un dessin technique, nous avons deux particularités intéressantes. La première concerne la note que l'on peut lire sur ce diagramme. En effet, un dessin technique est constitué d'élément, l'ajout d'un élément n'est pas représenté dans ce diagramme en détail car cette action est déjà représenté par le diagramme d'état transition "Création d'une figure et Création d'un multi-segment".

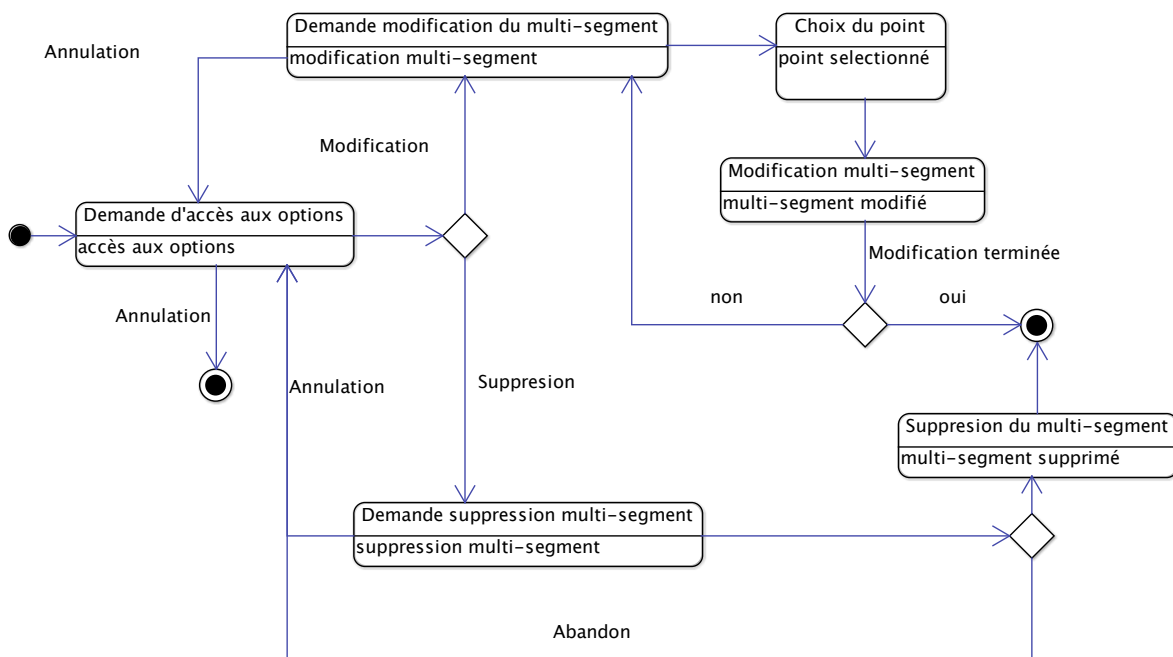
La seconde est la boucle que l'on observe sur l'ajout d'une figure. Un dessin technique peut être composé de plusieurs figures, leur nombre dépendant de l'utilisateur, une boucle est une manière judicieuse de représenter ce processus.

D) Modification d'une figure



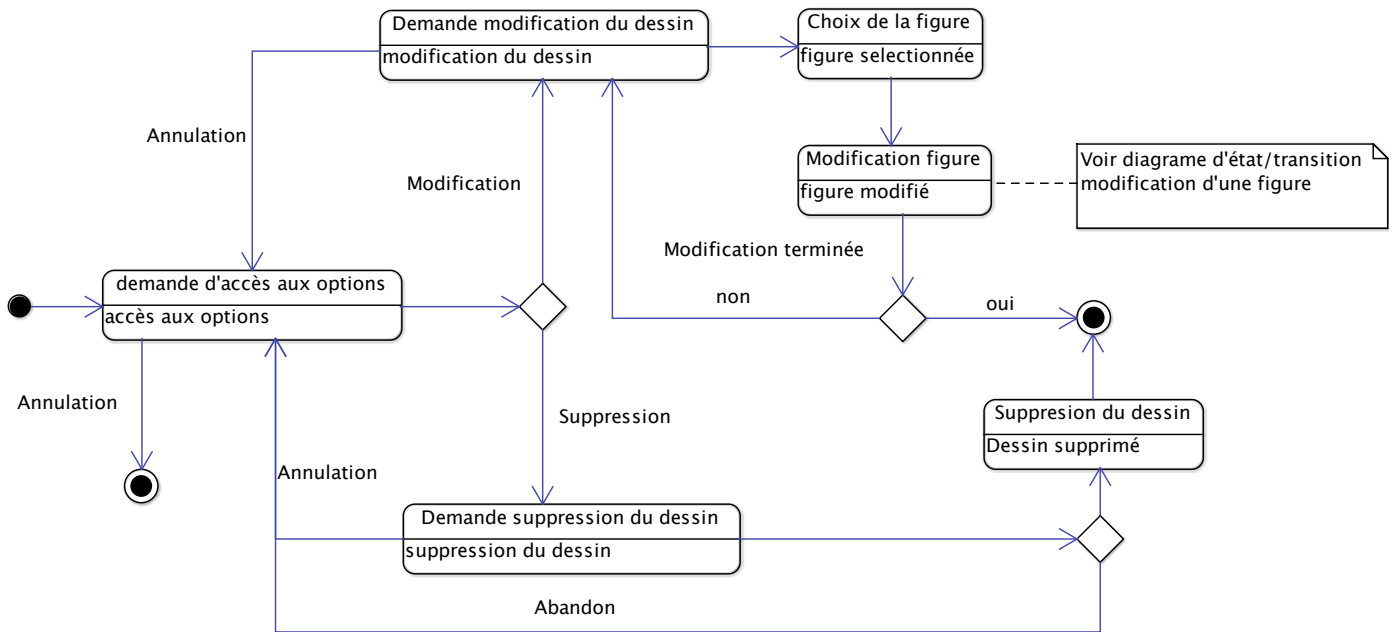
Ce diagramme d'état transition représente la modification d'une figure. Comme décrit lors de l'explication du diagramme de cas d'utilisation, nous assimilons la suppression comme une modification particulière. L'utilisateur demande dans un premier temps à modifier une figure. De là, deux choix s'offrent à lui, la suppression ou la modification. Si l'utilisateur choisit de supprimer la figure, une confirmation est requise. S'il décide au contraire de la modifier, nous retrouvons un schéma de traitement similaire à l'ajout avec la partie vérification des données.

E) Modification d'un multi-segment



Ce diagramme d'état transition de modification d'un multi segment commence par le choix de l'utilisateur. La suppression du multi segment suit le même schéma que celui de la suppression d'une figure, avec une confirmation pour éviter les suppressions par inadvertance. Une fois le point choisie et modifié la modification s'arrête.

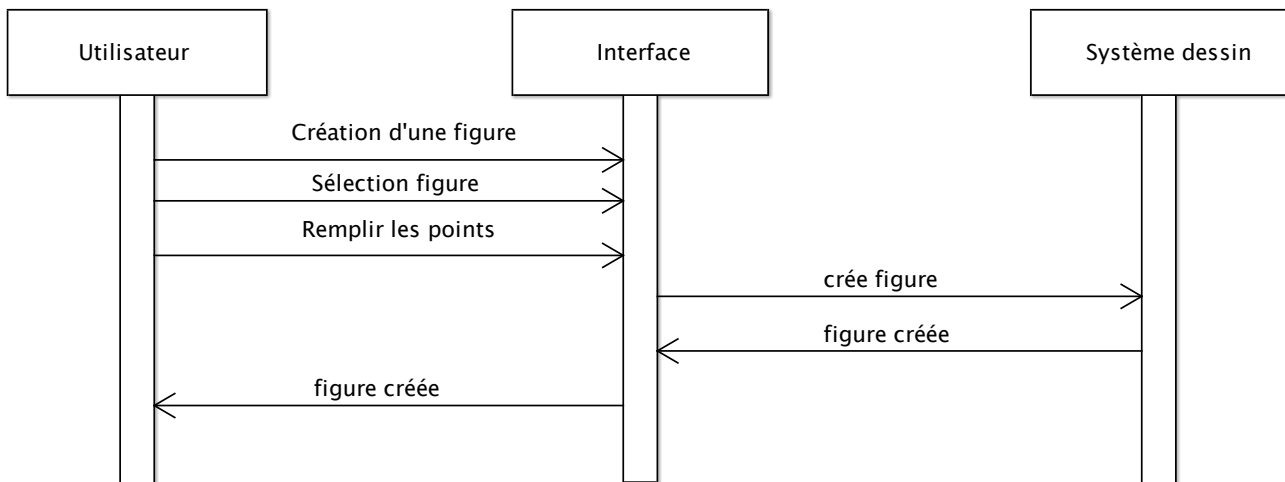
F) Modification d'un dessin technique



De même que pour le diagramme d'état transition de la modification d'une figure, celui de la modification de dessin commence par le choix de l'action voulue. La suppression du dessin suit le même schéma que celui de la suppression d'une figure, avec une confirmation pour éviter les suppressions par inadvertance. La modification reprend le principe de boucle comme pour la création d'un dessin technique. En effet, un dessin technique étant composé de plusieurs éléments, il est légitime que l'utilisateur puisse en une modification modifier plusieurs figures. La modification d'un élément particulier du dessin renvoie au diagramme d'état transition de modification de cet élément, c'est à dire le multi segment ou la figure.. Ceci afin de simplifier la lecture du diagramme et d'éviter les redondances.

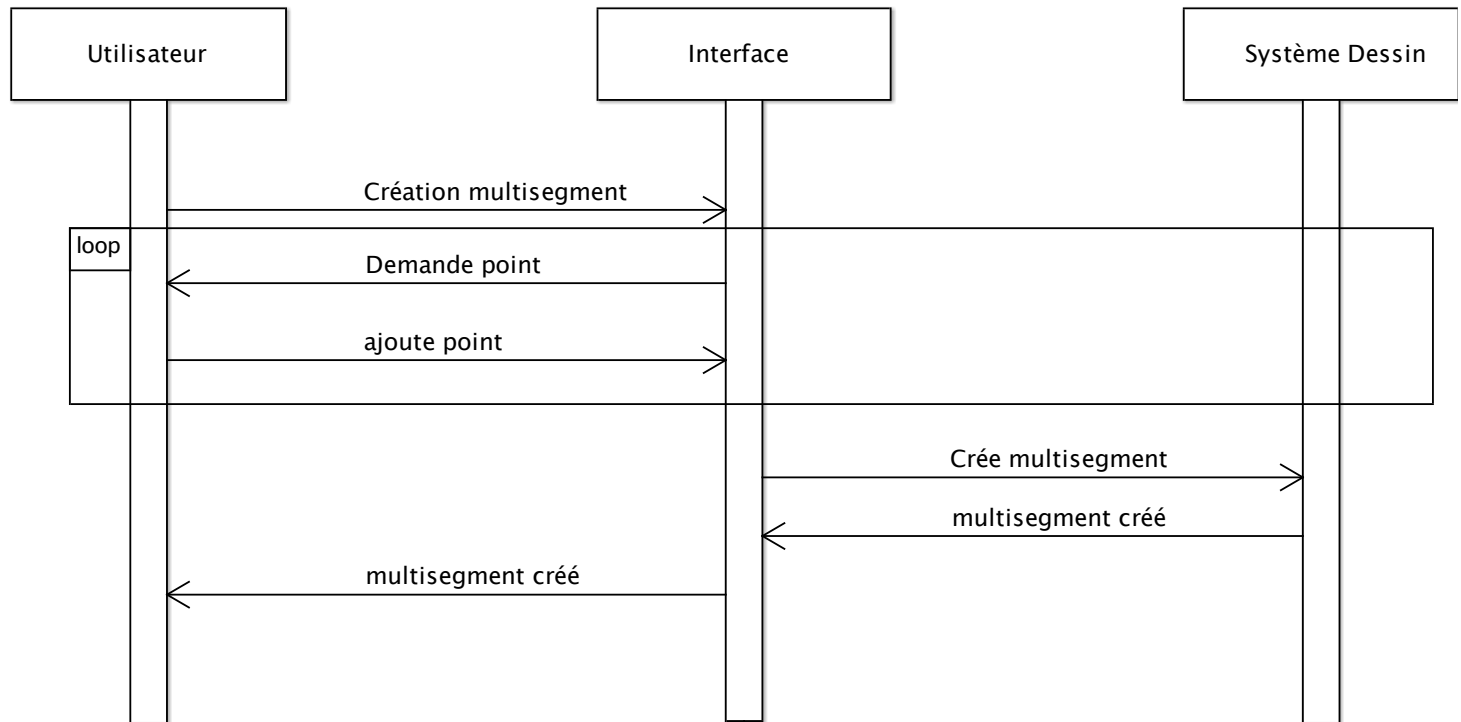
VI. Diagrammes de séquence

A) La figure



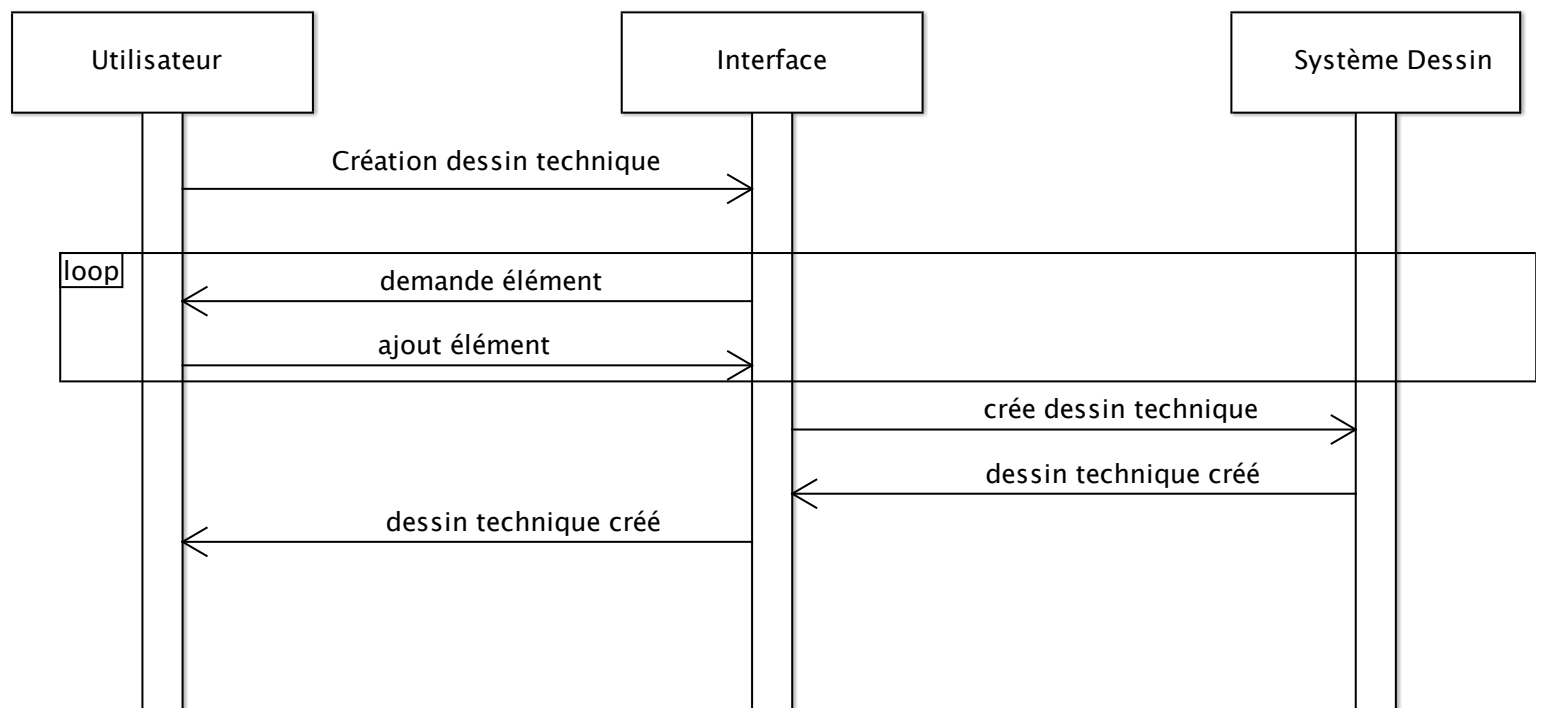
Dans ce premier diagramme de séquence, nous avons modélisé les interactions lors de la création d'une figure. Celui-ci ne prend pas en compte les cas alternatifs, tels que des données erronées. Etant déjà décrit dans le diagramme d'état-transition, nous avons évité de surcharger ce diagramme. La première partie concerne l'interaction de l'utilisateur avec l'interface graphique. Celui-ci choisit la figure puis remplit les valeurs correspondantes. Puis l'interface communique avec le système dessin pour générer l'objet correspondant.

B) Le multi segment



Le second diagramme de séquence modélisant la création d'un multi segment comporte également deux parties. La partie où l'utilisateur renseigne les données et une partie création. Cette première partie se caractérise par une boucle pour l'ajout des points. En effet, un multi segment est constitué d'un nombre indéfini de point.

C) Le dessin technique



Un dessin technique étant composé de plusieurs éléments pouvant être des figures ou des multi segments, nous retrouvons dans ce diagramme de séquence une boucle permettant d'ajouter autant d'éléments que désiré.

VI . Gestion Projet

Afin d'avancer dans notre projet de façon optimale, nous avons mis en place un planning de partage des tâches.

Ces tâches sont divisés en 4 :

- L'analyse
- L'UML
- La programmation de les classes
- La programmation de l'interface graphique

Voici un tableau de répartition des tâches qui récapitule la gestion de notre projet.

	S1	S2	S3	S4	S5	s6	S7	S8	S9	S10	S11	S12
Analyse												
UML												
Rapport UML												
C++												
C++/Qt												
Rapport C++												

VII. Conclusion

Au cours de ce rapport, nous avons essayé de faire preuve de la plus grande clarté et précision possibles pour expliquer et justifier nos choix de modélisation. Nous nous sommes appuyés sur les différents diagrammes de classes, de cas d'utilisation et de séquences afin de transmettre notre analyse. Le diagramme de cas d'utilisation nous a permis de bien définir les actions de notre application finale et le diagramme de classes de représenter les différentes entités en mémoire. Ceux d'état-transition et de séquence nous ont permis de mieux nous représenter le cheminement de l'application, du point de vue d'une action particulière puis relativement au temps.

La modélisation de ces différents diagrammes nous aura permis d'avoir une analyse plus structurée et surtout de pouvoir revenir sur nos différents choix pour les modifier et ainsi nous rapprocher de la solution finale choisie. La réalisation de ce premier projet à l'aide d'UML nous aura donc été très bénéfique. Nous avons pu nous rapprocher de méthodes d'analyse et de développement proches de ceux de l'entreprise et ainsi nous éloigner de cette mauvaise tendance que nous avons tous en commençant à programmer et ensuite réfléchir.