Dossier Conception

Groupe 1_3

2ème Itération

Pour cette itération nous avions commencer à utiliser le patron commande pour dessiner les formes demandées en gardant l'utilisation du patron visiteur qui était faite au tout début du projet.

Nous avons aussi ajouter la fonctionnalité de faire un CTRL-Z pour revenir en arrière au cas d'une fausse manipulation.

3ème Itération

Sur cette itération nous avons réussi à intégrer le Cube comme nouvelle forme à dessiner en utilisant le jar fourni 'shape-1.0.1.jar', en gérant son export XML/JSon, ensuite nous avons réussi à faire un déplacement des formes fonctionnel (pour le cube il faut bien le tirer du coté en bas à droite sinon il ne bouge pas).

Nous avons aussi changé totalement l'architecture du code pour une meilleure implémentation qui respecte le patron command.

Nous avions bien pensé à adapter notre retour à l'arrière (CTRL-Z) pour prendre en compte le déplacement des objets et maintenant vous pouvez créer et déplacer dans l'ordre que vous voulez et notre retour en arrière pourra tout gérer (annulation des déplacements et des dessins).

Malheureusement, nous n'avons pas pu intégrer le regroupement à notre projet.

4ème Itération

Comme invoqué sur le retour de la dernière itération le bouton undo a été supprimé et comme pour cette itération vous n'avez pas demandé de se rattraper sur le groupement donc nous avons décidé de partir sur une base simple et fonctionnelle et d'implémenter que l'import qui est obligatoire.

Pour la nouvelle fonction d'import un nouveau bouton a été rajouter à la barre en haut qui nous permet sur click de choisir le fichier d'où on fera l'import, il suffit que le fichier supporte la même structure de données xml pour le que le programme le lise et le dessine sur la fenêtre.

Les scénarios qui doivent marcher ont été testés et tout marche bien.

On pourra ainsi jouer le scénario complet :

- On lance l'application
- On crée des figures et des groupes de figures, on les déplace
- On sauvegarde le dessin (en XML ou JSon)
- On ferme l'application
- On l'ouvre à nouveau
- On importe le fichier précédemment créé
- On complète le dessin
- On l'exporte
- On ferme l'application
- On réimporte le dessin
- …

Afin de pouvoir exécuter le projet il est obligatoire d'exécuter la commande suivante :

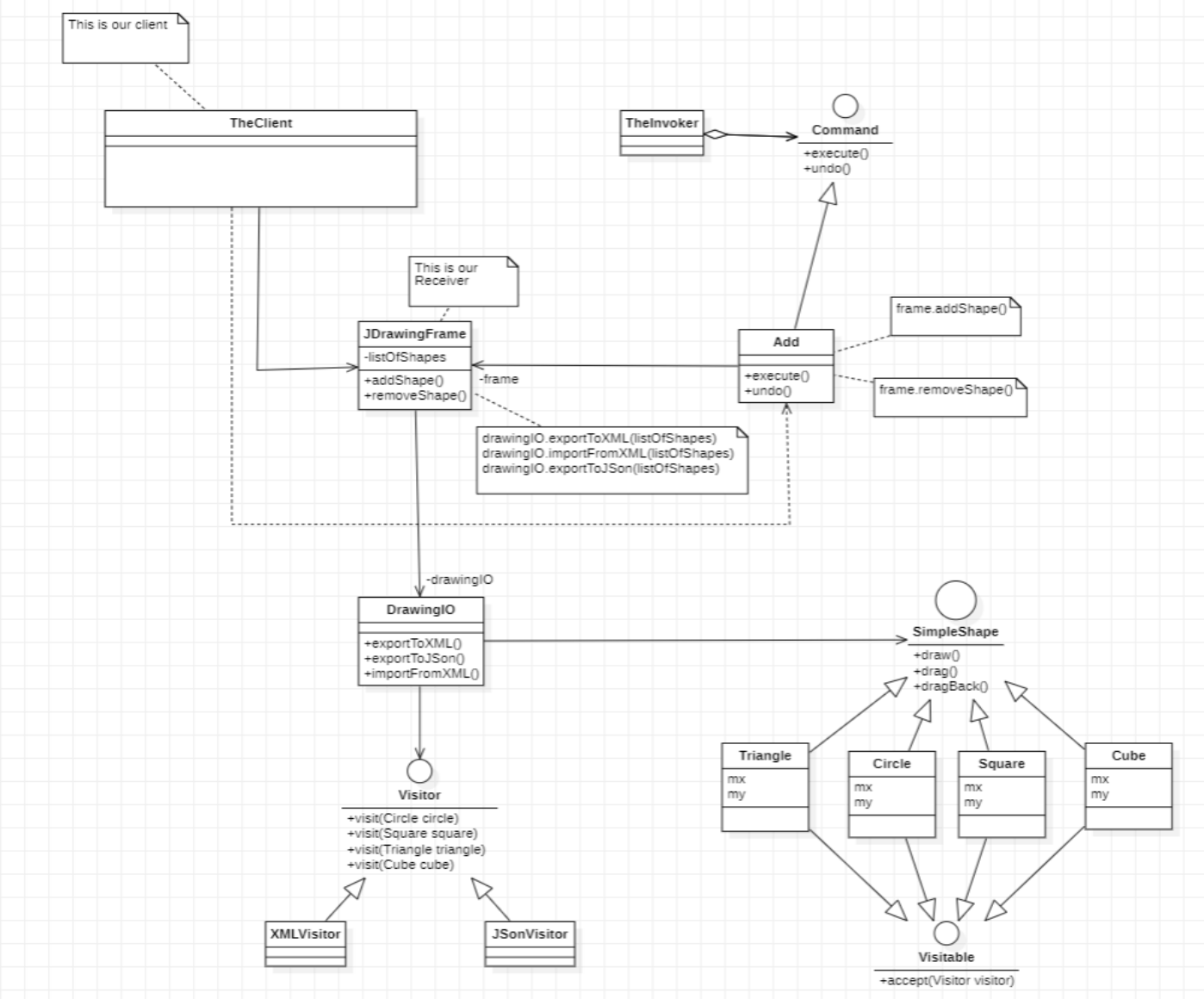
mvn install:install-file -Dfile="lib/anasat.jar" -DgroupId="com.anasat" -DartifactId="anasat" -Dversion="1.0-SNAPSHOT" -Dpackaging="jar" -DgeneratePom=true

Difficultés rencontrées et potentielles améliorations:

- Nous voulions intégrer la possibilité de faire un retour vers l'arrière si nous avons effectué un import par erreur mais malheureusement nous n'avons pas pu le faire à temps en respectant aussi le patron commande.
- Nous n'avons pas pu tester la totalité du code en utilisant JUnit et donc nous avons fait quelques tests des différents scénarios à la main.
    - Quelques scénarios testés à la main :
        - **1 :**
            - Importer fichier xml
            - Déplacer shape
            - Exporter le dessin
            - Fermer l'application
            - Re-importer le fichier xml
            - **Vérifier que le shape déplacé est dans dans le bon endroit**
        - **2 :**
            - Ouvrir l'application
            - Dessiner shape
            - Importer fichier xml
            - **Vérifier que le dessin contient le shape dessiné et le dessin importé aussi**
                - On a fait ça pour avoir la possibilité d'importer des dessins prédéfinis, par exemple un dessin d'un visage en utilisant les cercles, et on pourra ajouter ce petit visage à chaque fois on veut dans notre dessin courant.
        - **3 :**
            - Ouvrir l'application
            - Importer un dessin
            - Importer encore une fois
            - Rien ajouter au dessin
            - Exporter le dessin en XML
            - **Vérifier que le fichier xml exporté ne contient pas les shapes en double**
        - **4 :**
            - Ouvrir l'application
            - Importer un dessin
            - Ajouter un shape quelque part
            - Exporter le dessin en XML
            - **Vérifier que le fichier xml exporté contient les shapes de l'import + le shape ajouté et sans les doublants des shapes du dessin importé au début**

Diagramme UML :

Ce diagramme englobe l'architecture générale du code (patron visiteur + patron commande).



Diagrammes Séquence :

Diagramme de séquence de l'import :

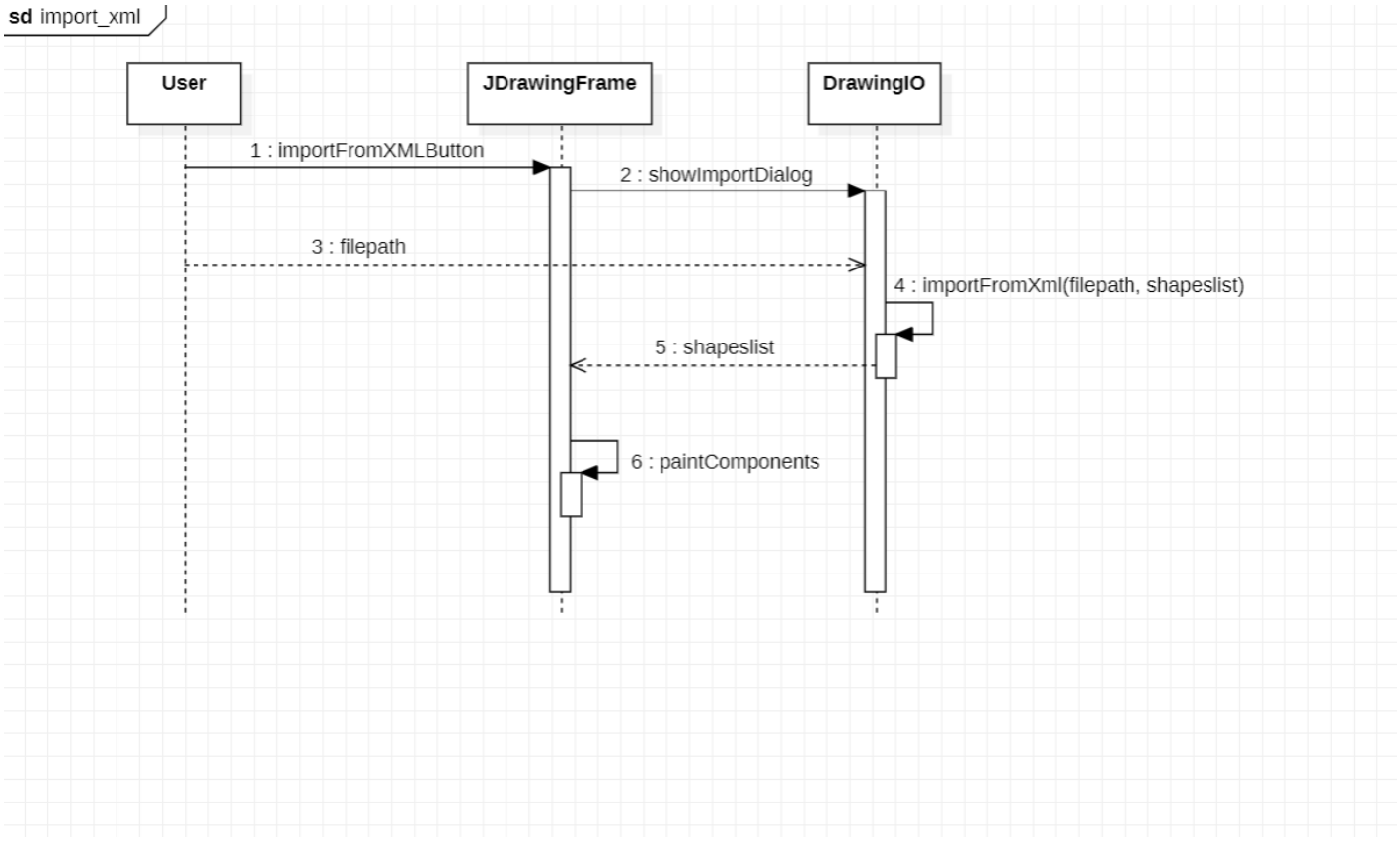


Diagramme de séquence de l'export :

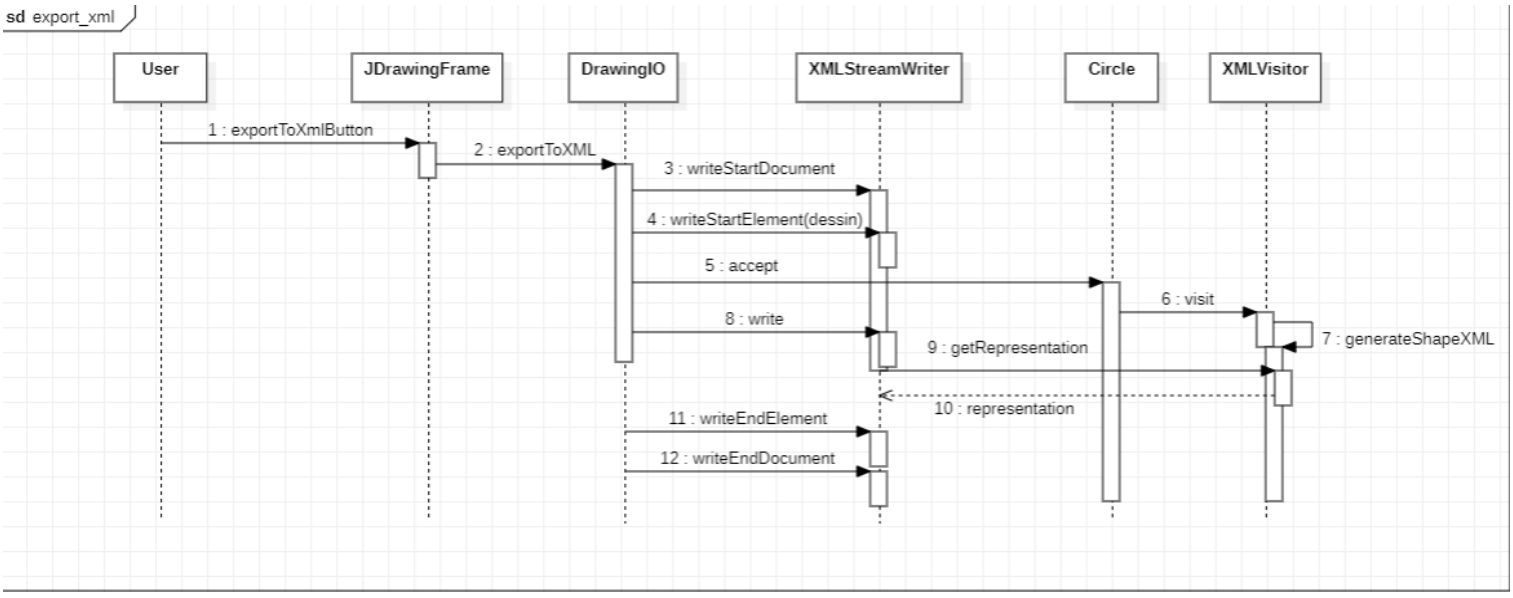


Diagramme de séquence de la commande ADD :

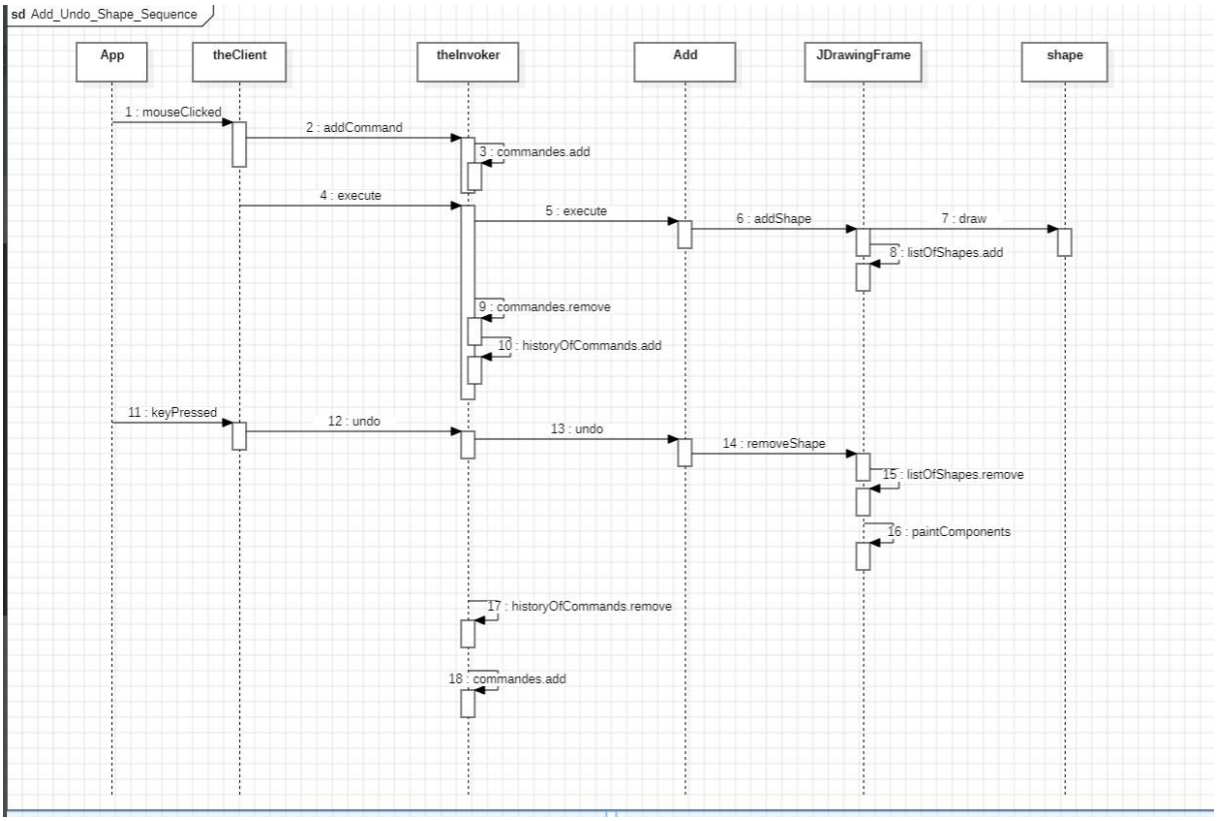
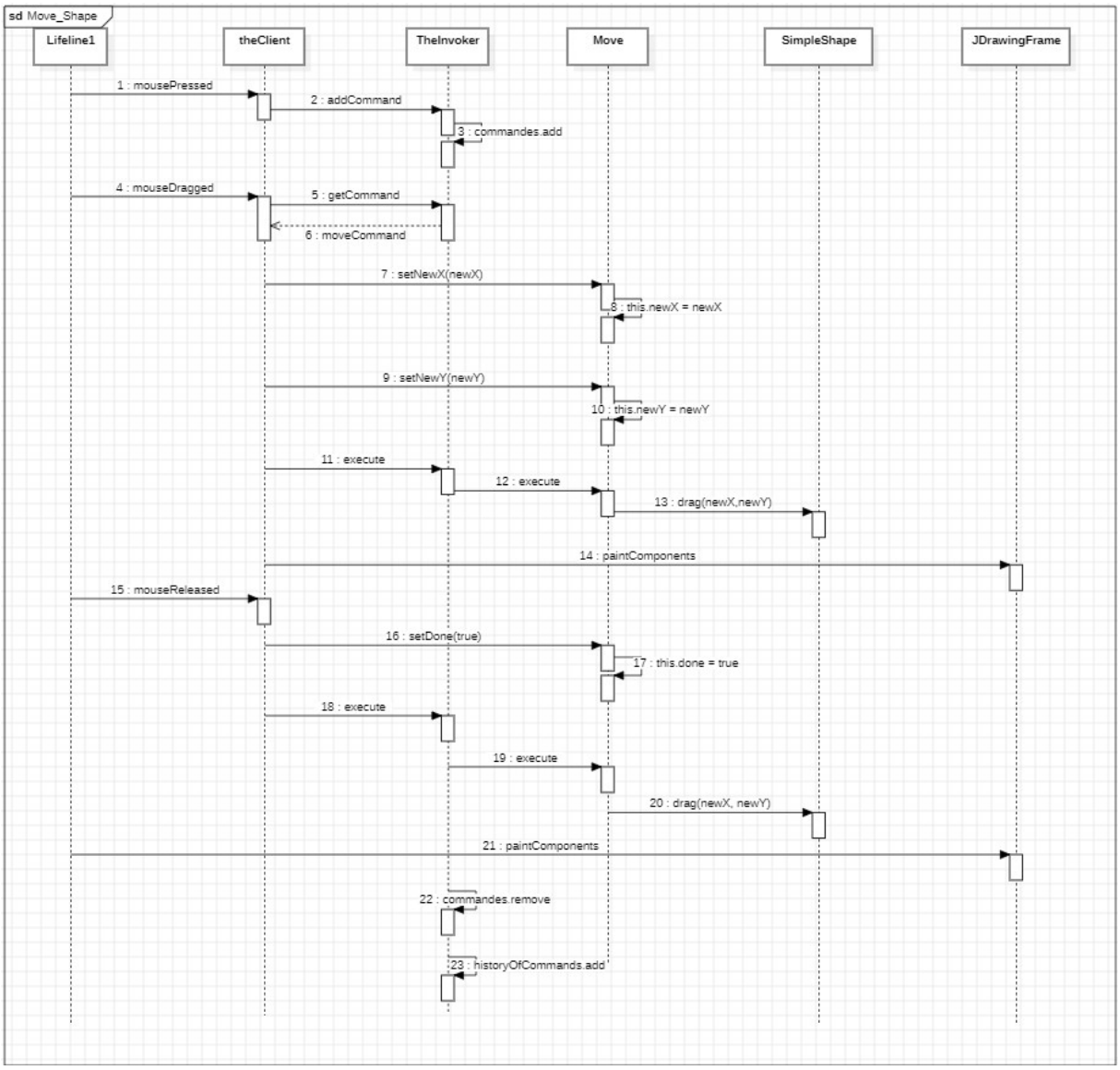


Diagramme de séquence de la commande MOVE :



Diagrammes Composants :

Diagramme de composants architecture :

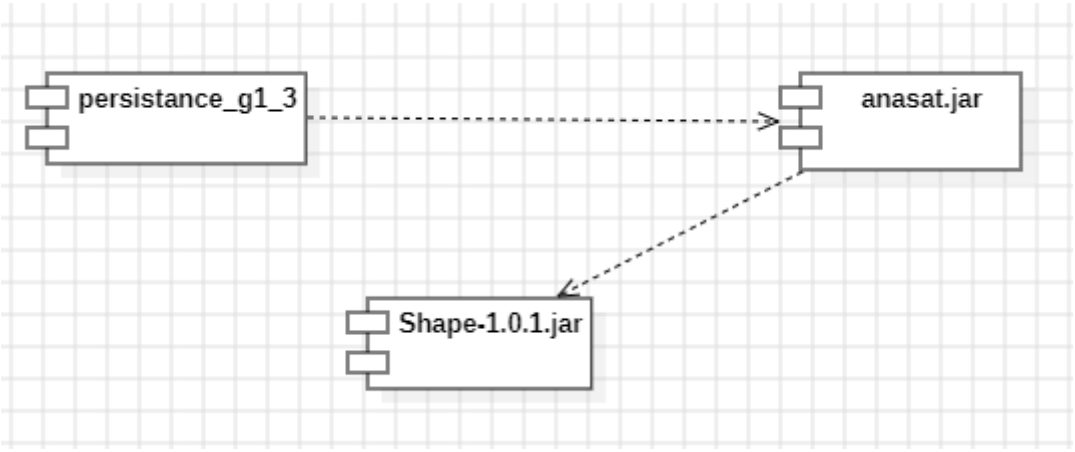
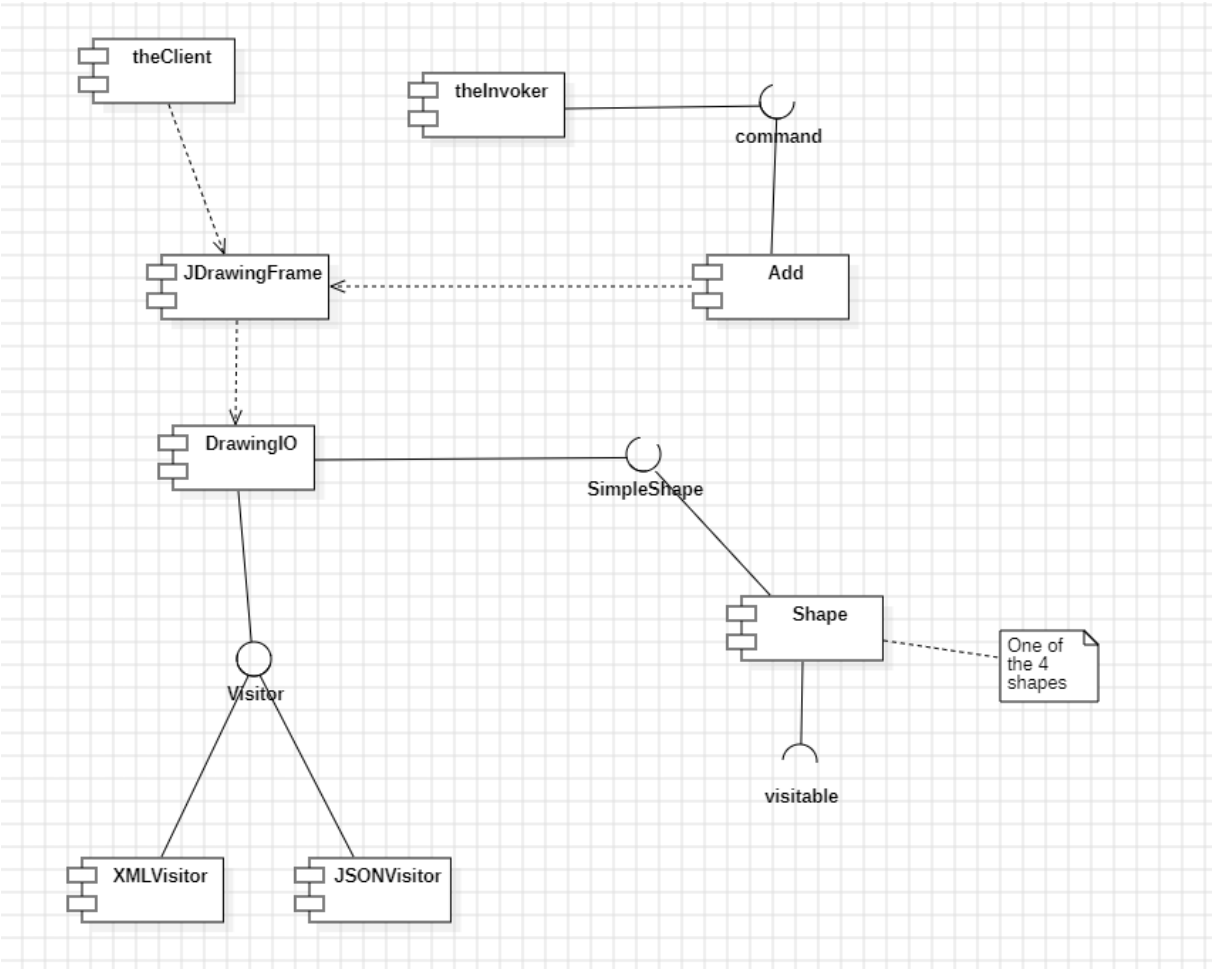


Diagramme de composants entre les classes:



Authors : *EL BOUCHRIFI Anas – BENABBOU Anas*