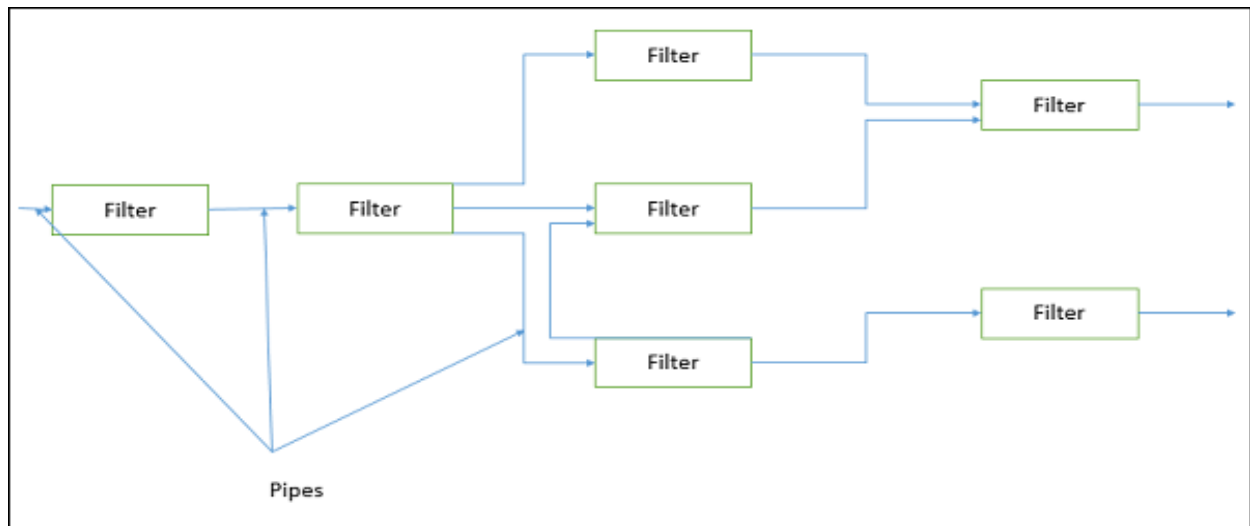


DESIGN PATTERN

PIPES AND FILTERS PATTERN



Niveau : **TIC4**

Présenté par : **NKOUNKOU Clerc**
YEBAS David

Professeur :

TABLE DE LA MATIERE

I. INTRODUCTION.....	3
II. EXPLICITATION DU CONCEPT	4
1. Un exemple de non-programmation	4
2. Un exemple de programmation	4
III. PROBLEMATIQUE	6
IV. Solution	7
V. IMPLEMENTATION	10
• Diagramme de classe	10
• Code java.....	10
• Phase d'exécution Eclipse	10
VI. CONCLUSION.....	11

I. INTRODUCTION

En architecture informatique, un système en style à **filtres et tubes** (anglais *pipes and filters*) est composé d'un réseau d'unités autonomes (les filtres) qui effectuent des transformations sur des informations transmises par un réseau de connexions (les tubes).

Dans cette construction, qui imite celle d'une usine chimique, chaque filtre comporte des entrées et des sorties auxquelles sont branchés les tubes. Le filtre lit les informations en entrée, effectue une transformation, puis envoie le résultat en sortie. Le traitement est effectué petit à petit, et le filtre produit des résultats en sortie avant d'avoir consommé toutes les informations en entrée. Les filtres communiquent entre eux uniquement à travers les tubes, et aucun filtre ne connaît l'identité du filtre qui est branché à l'autre bout du tube.

II. EXPLICITATION DU CONCEPT

1. Un exemple de non-programmation

Imaginons la chaîne de montage d'une usine automobile. La fabrication d'une voiture comporte de nombreuses tâches, mais n'envisageons que des tâches très générales:

- l'installation du châssis et de la chaîne de transmission
- le boulonnage de la carrosserie sur le châssis
- le montage du moteur
- la pose des sièges et des roues
- Envoie à la peinture et le test final

2. Un exemple de programmation

Imaginons un système d'analyseur de requête pour un moteur de recherche très complexe pour rendre le résultat plus riche et concret celle-ci sera découper en forme de procédure :

- Réception du texte de la barre de recherche en mode text plain
- Filtrage de la requête et sécurisation retrait potentiel js.code...
- Analyse de la position du navigateur et toutes autres données géographiques
- Consultation des bases de données et toute autre correspondance
- Trie des résultats en fonction des critères choisie sur la page
- Renvoi des résultats vers la vue.

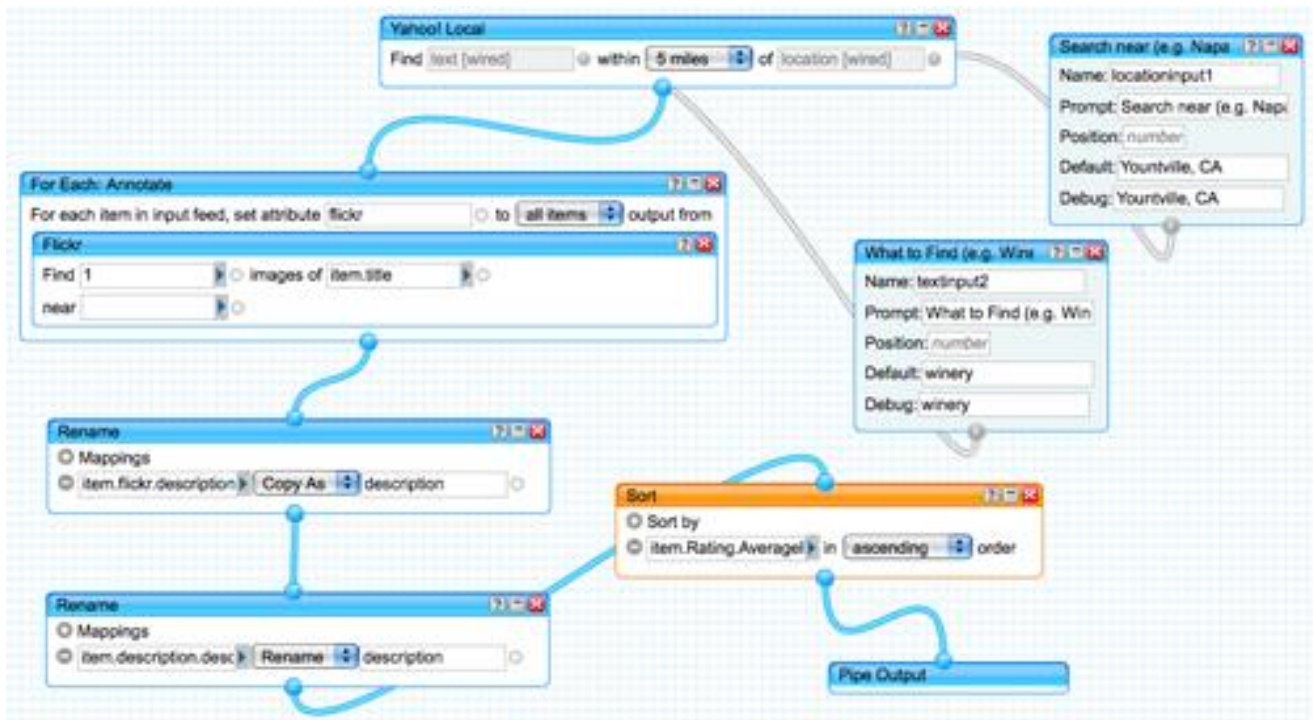


Figure 1. systèmes de recherche Yahoo.

III. PROBLEMATIQUE

L'application d'une série de calculs ordonnés mais indépendants est nécessaire, peut-être comme une série d'opérations pas à pas, sur des données ordonnées.

Conceptuellement, un seul objet de données est transformé. Si les calculs étaient effectués en série, le jeu de données de sortie de la première opération servirait d'entrée aux opérations lors de l'étape suivante, dont la sortie servirait à son tour d'entrée aux opérations par étapes suivantes.

En règle générale, la performance en tant que temps d'exécution est la caractéristique intéressante.

Les forces

Selon la description du problème et en considérant la granularité et l'équilibre de la charge comme autres forces importantes pour la conception de logiciels parallèles, les forces suivantes doivent être prises en compte pour une version parallèle du motif Tuyaux et filtres :

- **Maintenir l'ordre des opérations précis.**
- **Conservez l'ordre des données partagées entre toutes les opérations.**
- **Considérons l'introduction du parallélisme, dans lequel différentes opérations par étapes peuvent traiter différentes données en même temps.**
- **Répartissez le processus en quantités similaires entre toutes les étapes.**
- **L'amélioration des performances est atteinte lorsque le temps d'exécution diminue.**

IV. Solution

Le parallélisme est représenté par des opérations qui se chevauchent dans le temps. Les opérations produisent une sortie de données qui dépend des opérations précédentes de son entrée de données, en tant qu'étapes incrémentielles. Les données provenant de différentes étapes sont utilisées pour générer une modification de l'entrée au fil du temps. Le premier ensemble de composants commence à être calculé dès que les premières données sont disponibles, au cours du premier pas temporel. Lorsque son calcul est terminé, les données de résultat sont transmises à un autre ensemble de composants dans le deuxième pas de temps, en suivant l'ordre de l'algorithme. Alors, alors que ce calcul a lieu sur les données, le premier ensemble de composants est libre d'accepter plus de nouvelles données. Les résultats des composants du second pas de temps peuvent également être transmis, pour être exploités par un ensemble de composants dans une troisième étape, tandis que le premier pas de temps peut accepter plus de nouvelles données et le second pas de temps fonctionne sur le deuxième groupe de données, etc.

Structure

Ce modèle s'appelle Tuyaux et filtres, car les données sont transmises sous forme de flux d'une étape de calcul à une autre le long d'un pipeline de différents éléments de traitement. La principale caractéristique est que les résultats des données ne sont transmis que dans un sens de la structure. L'exécution parallèle complète se construit progressivement lorsque les données deviennent disponibles à chaque étape. Différents composants existent simultanément et se traitent pendant le temps d'exécution (Figure 2).

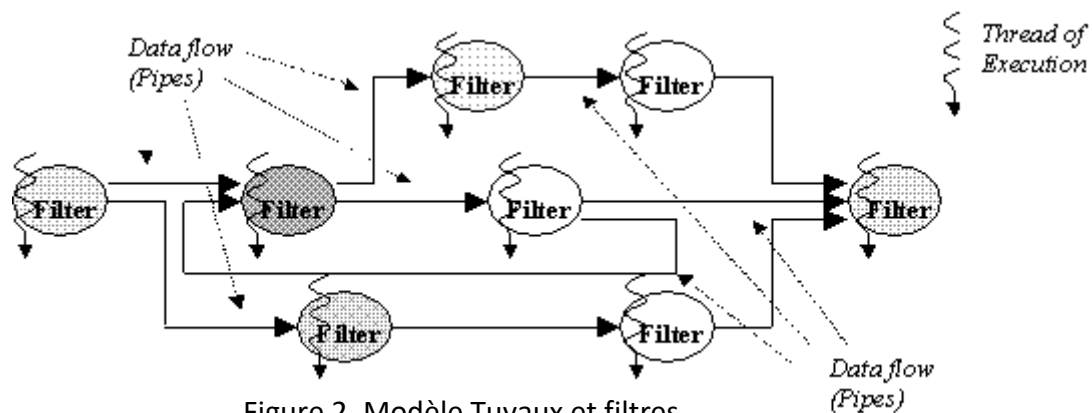


Figure 2. Modèle Tuyaux et filtres.

- **Filtre:** Les responsabilités d'un composant de filtre sont de générer des données ou d'obtenir des données d'entrée à partir d'un canal, d'effectuer une opération sur ses données locales et d'envoyer des données de résultat en sortie à un ou plusieurs canaux.
- **Tuyau:** Les responsabilités d'un composant de canal sont de transférer des données entre les filtres, parfois de les stocker en mémoire tampon ou de synchroniser l'activité entre les filtres voisins.

Dynamique

En raison de l'exécution parallèle des composants du modèle, le scénario typique suivant est proposé pour décrire son comportement d'exécution de base. Comme tous les filtres et tous les canaux sont actifs simultanément, ils acceptent les données, les exploitent dans le cas de filtres et les envoient à l'étape suivante. Les pipes synchronisent l'activité entre les filtres.

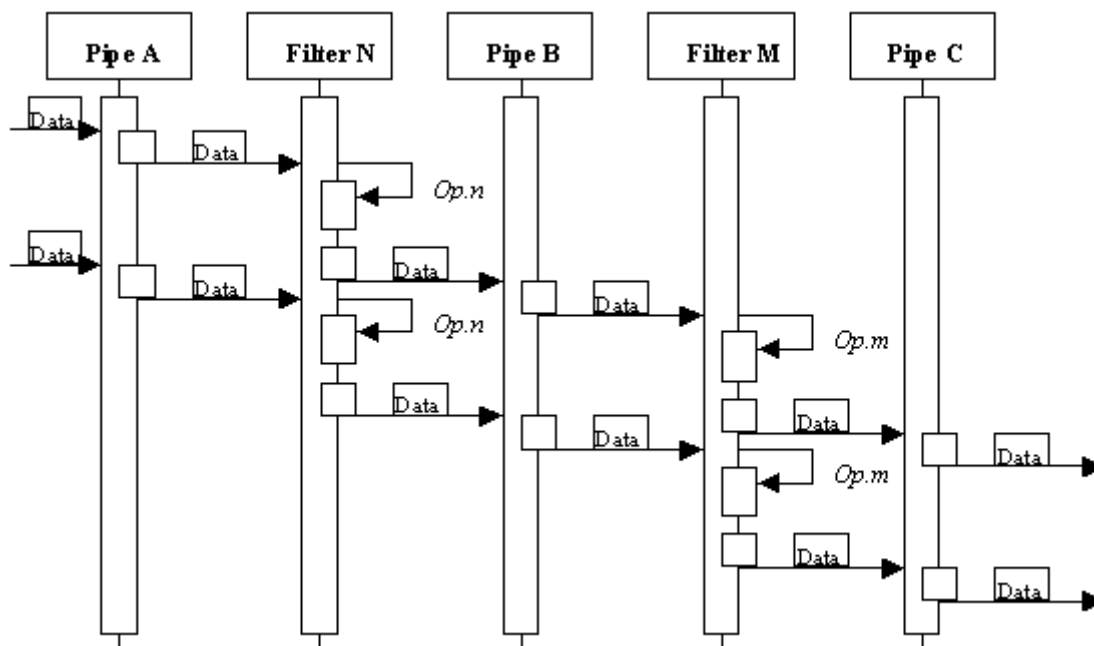


Figure 3. Scénario de modèle de canalisations et de filtres.

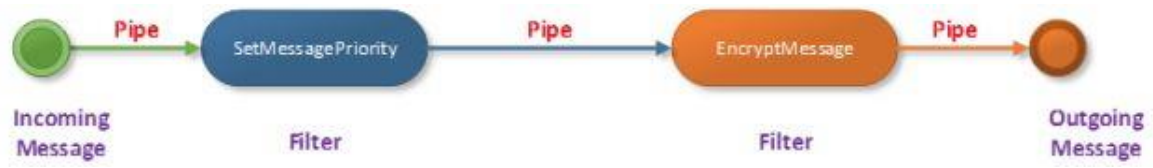
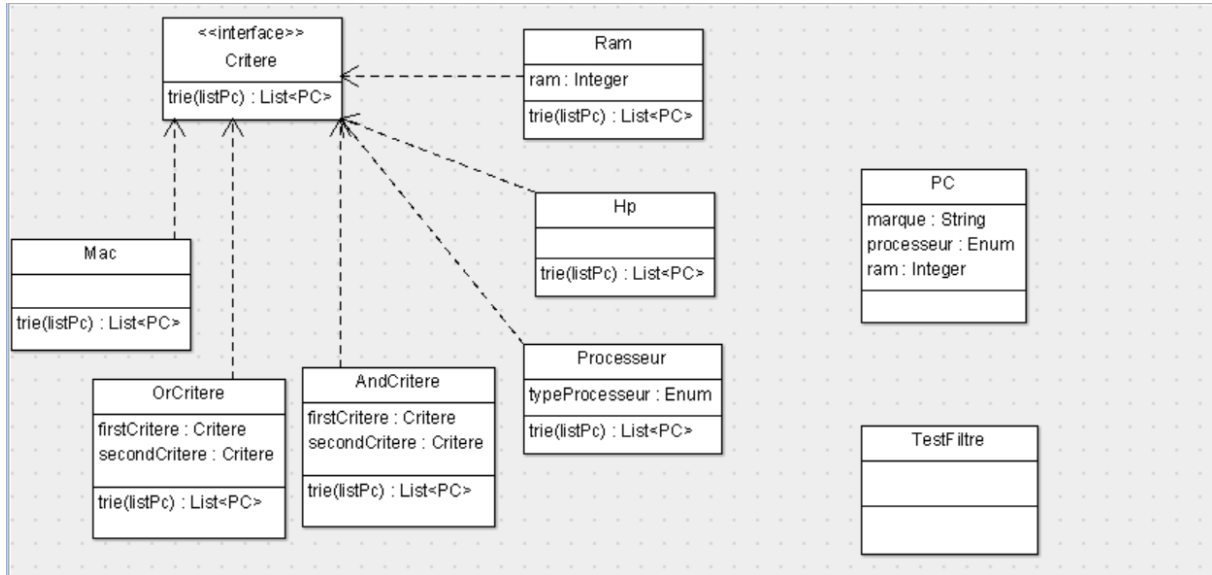


Figure 4. Scénario de cryptage de messages.

V. IMPLEMENTATION

- Diagramme de classe



- Code java
- Phase d'exécution Eclipse

La phase d'exécution consiste juste à tester le code implémenté sur l'IDE Eclipse.

Dans le souci de conservation du cours, le code et l'ensemble du document sont disponible sur GitHub.

Url : <https://github.com/ncn17/Pipes-and-Filters>

Language : Java

Version : 8

VI. CONCLUSION

En effet le Pattern Filtres and Pipers sont très utiles pour organiser notre code, de façon à ce que toute sorte de traitement de trie soit reparti en classe.