

# Projet AOC

MERZUOK Fahim, KADRI Nouredine

M2 Ingénierie Logicielle en alternance

ISTIC

Université Rennes 1

`fahim.merzouk@etudiant.univ-rennes1.fr`

`nouredine.kadri@etudiant.univ-rennes1.fr`

29 janvier 2019

**Année universitaire : 2018/2019**

**Module : AOC**



## Résumé

Le but de ce projet est la mise en place d'un programme de diffusion d'une valeur entière générée par un générateur sur des afficheurs à travers des canaux asynchrones.

# Table des matières

<b>1</b>	<b>Intrduction</b>	<b>3</b>
<b>2</b>	<b>Choix techniques</b>	<b>3</b>
2.1	Patron de conception Strategy . . . . .	3
2.2	GUI . . . . .	3
2.3	Capture d'écran GUI . . . . .	4
2.4	Patron de conception Active Object . . . . .	5
2.5	Diagramme de Séquence . . . . .	6
<b>3</b>	<b>Conclusion</b>	<b>7</b>

# 1 Introduction

Le projet consiste à mettre en place le design pattern Active Object vu en cours, pour la diffusion de valeurs entières générées à partir d'un générateur à travers des canaux vers des afficheurs. Le but étant de proposer une conception avec différentes stratégies de diffusion : Diffusion atomique, séquentielle et causale. Pour cela, nous avons utilisé d'autres patron de conception en plus d'Active Object comme Strategy et Observer. Nous avons aussi conçu une interface graphique avec java Swing qui permet de visualiser les différentes valeurs des afficheurs, générateur et permet de choisir le type de diffusion.

## 2 Choix techniques

### 2.1 Patron de conception Strategy

Notre programme permet de générer des valeurs et de les diffuser aux afficheurs souscrits via des canaux de diffusion, cependant, il est possible de choisir le mode de diffusion qu'on souhaite sur la GUI. Il existe trois modes de diffusion : **Atomique** (la mise à jour suivante est envoyée seulement si la première est reçue), **Causale et Séquentielle** (l'ordre d'arrivée des mises à jour n'est pas vérifié), pour donner ce choix à l'utilisateur, nous avons utilisé le PC Strategy qui répond parfaitement à ce genre de problématique de conception.

### 2.2 GUI

L'interface graphique de notre programme est réalisée avec **JavaSwing**

## 2.3 Capture d'écran GUI

L'utilisateur du programme peut choisir entre 3 stratégies de diffusion :

1. Diffusion Atomique
2. Diffusion Séquentielle
3. Diffusion Causale

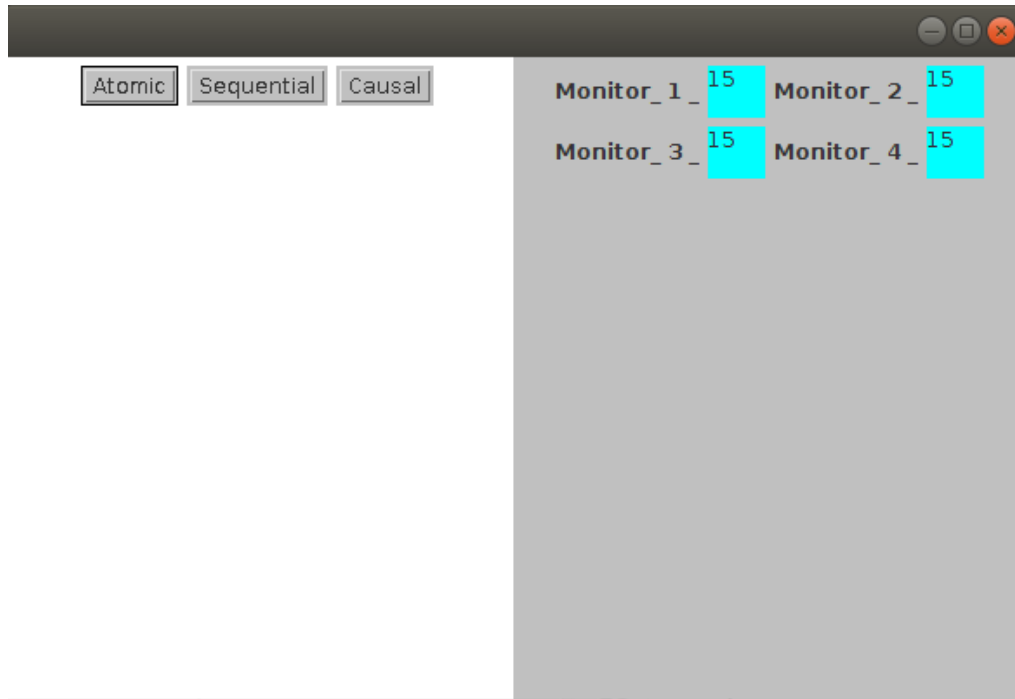


FIGURE 1 – Interface graphique du programme

## 2.4 Patron de conception Active Object

Nous avons mis en place deux fois le design pattern Active Object, parce qu'on a besoin d'appels asynchrones dans les deux sens. Voici un exemple pour Update.

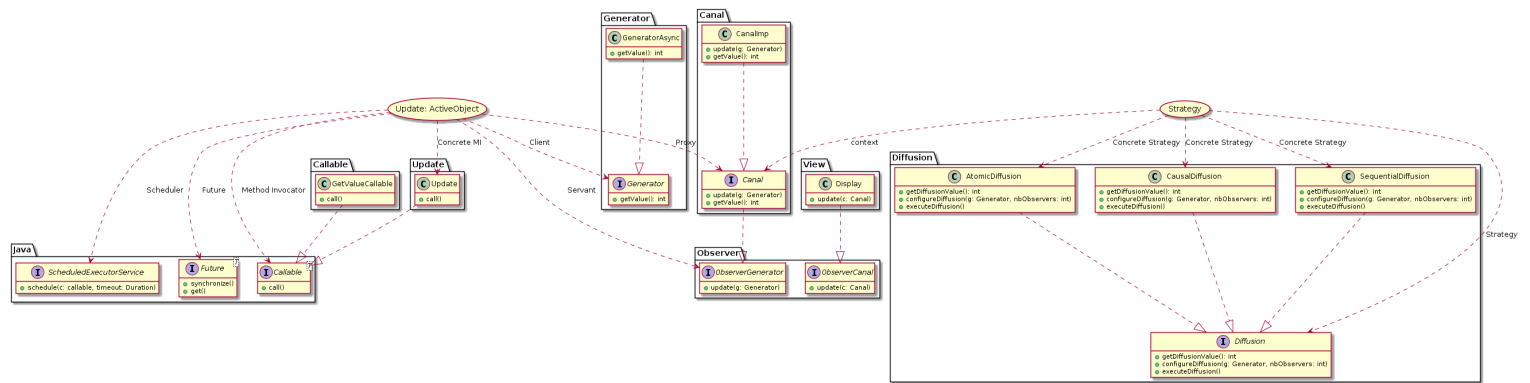


FIGURE 2 – Diagramme de classe de l'Update

## 2.5 Diagramme de Séquence

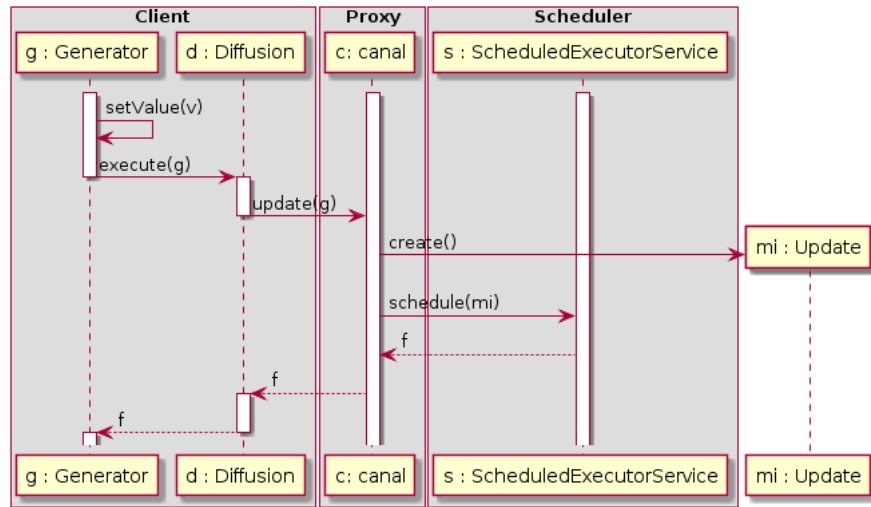


FIGURE 3 – Première partie du diagramme de séquence Update

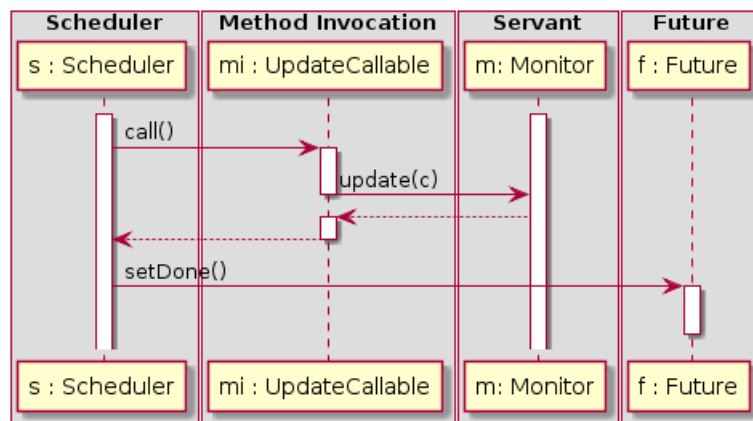


FIGURE 4 – Deuxième partie du diagramme de séquence Update

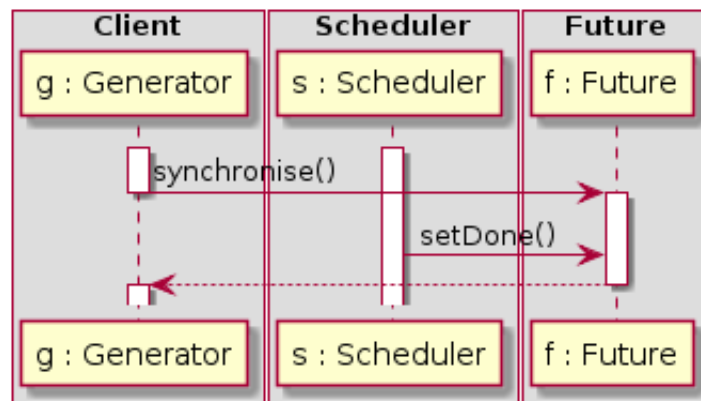


FIGURE 5 – Troisième partie du diagramme de séquence Update

### 3 Conclusion

A la fin de ce projet, nous nous sommes rendu compte qu'on a appris beaucoup de choses qui sont par exemple : la mise en place de plusieurs design pattern au même temps, l'utilisation de JavaSwing pour l'interface graphique et l'utilisation de Active Object. Nous avons rencontré des difficultés à comprendre tout l'ensemble, mais en partageant bout par bout, on estime qu'on a réussi l'implémentation.