**COA – TP**

Marc DOUCHEMENT / Romain LE HO

ISTIC – Université de Rennes1

Février 2013

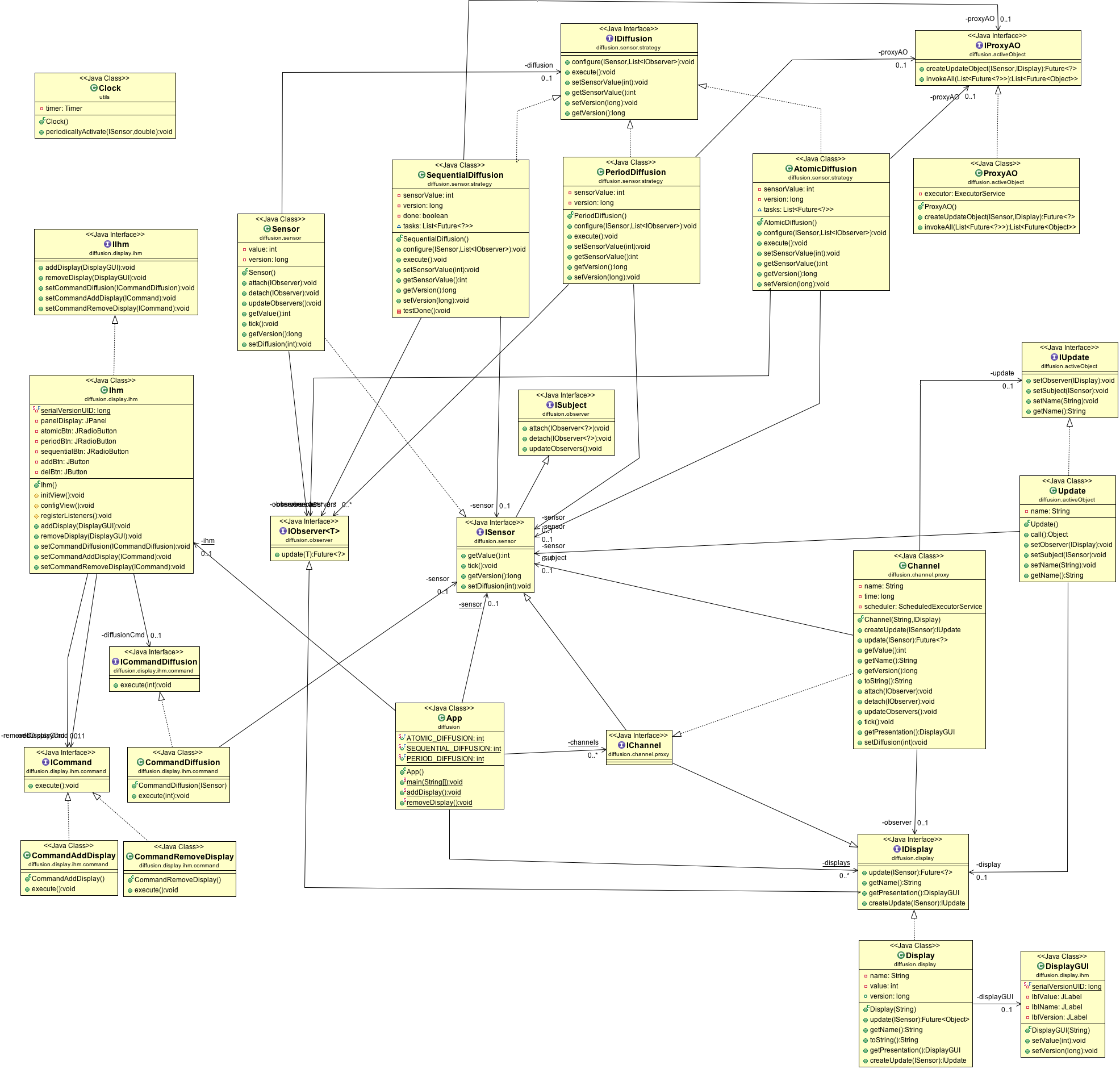
Rapport

# Introduction

Le TP de COA a pour but de gérer l’asynchronisme en Java. Pour cela, nous utilisons les classes Java permettant d’implémenter le patron de conception Active Object. Nous allons gérer la transmission d’une donnée d’un capteur vers des affichages, par le biais de canaux introduisant un délai de transmission.

Dans un premier temps, nous allons présenter le diagramme de classe de l’application, les différentes stratégies de diffusion qui peuvent être utilisées, et pour finir les diagrammes de séquences permettant de voir comment les affichages se mettent à jour.

# Diagramme de classe



# Les stratégies de diffusion

//TODO endroit où sont implémentés les stratégies

## Diffusion périodique

Cette stratégie de diffusion permet aux affichages de mettre à jour leur valeur uniquement si la valeur récupérée est plus fraiche que la précédente. Il n’y a pas de vérification quant à la synchronisation entre tous les lecteurs (affichages).

## Diffusion atomique

Ici, il faut que tous les affichages se soient mis à jour avant d’obtenir une nouvelle valeur. Cela signifie que le capteur est bloqué : il ne met plus à jour sa valeur tant que tous les affichages ne sont pas à jour.

## Diffusion séquentielle

La diffusion séquentielle suit le même principe que la diffusion atomique, la différence étant que le capteur peut quand même se mettre à jour même si tous les affichages n’ont pas tous récupérés une nouvelle valeur.

# Diagrammes de séquences

