Modèle M3: dans un univers sans appel asynchrone d'opération, 1 thread par client, n threads scheduler: application de Active Object pour Capteur GetValue «interface» «interface» <<Interface>> Callable **Future** ?? Capteur + attach(o:ObserverDeCapteur) + call() + detach(o:ObserverDeCapteur) + getValue():Integer MethodInvocation Future + tick() ConcreteMethodInvocation ScheduledExecutorService Servant + field: type _Scheduler :ActiveObject + schedule(c:Callable) CapteurImpl + field: type + attach(o:ObserverDeCapteur) + detach(o:ObserverDeCapteur) + getValue():Integer <<Interface>> + tick() **ObserverDeCapteur** Service ConcreteClient + update(s:Capteur) Context :Strategy **Afficheur** + field: type <<Interface>> Strategy CapteurAsync + update(s:Capteur) + getValue(?): Future(Integer) <<Interface>> AlgoDiffusion + configure() + execute() ConcreteStrategy Canal ?? DiffusionSequentielle **DiffusionAtomique DiffusionCausal** + update(s:Capteur): Future + getValue(?):Future(Integer) + configure() + configure() + configure() + execute() + execute() + execute()

Diagramme de classe