

Exercice 04

Objets connectés



1. ENONCÉ 2
2. Correction 3

1. Enoncé

Un objet connecté est défini par une adresse IP sur laquelle il est possible de faire une requête d'interrogation de données, appelées "datas" sous la forme d'une liste de couples (key, valeur) et de lui donner un ordre d'exécuter une action sous la forme d'un couple (nom de l'action, arguments de l'action).

L'action est dynamique et "téléchargée" sur l'objet connecté.

Un objet connecté est vu comme une JVM distante. La relation entre cette JVM et les capteurs physiques de l'objet connecté n'est pas traité ici. Il faut juste savoir que le rôle de ces capteurs est de renseigner certaines valeurs des "datas".

L'objectif est de faire les classes et interfaces qui permettent à des programmes extérieurs de :

- **configurer** ces objets connectés : créer les "datas" et charger le code permettant de réaliser l'action.
- **interroger** les "datas" en faisant du "pull" sur ces "datas".

En utilisant les Designs Patterns vus en cours faire le diagramme de classe de ce package.

NB : on n'utilise pas le protocole RMI mais le protocole de base Socket pour réaliser cette exercice (ex: en java, ServerSocket, Socket).

2. Conception préliminaire

L'Objet Connecté (OC) contient un serveur de socket permettant de traiter les requêtes provenant de l'extérieur. On va utiliser le concept de "skelton" et de "stub" pour réaliser la communication entre l'OC et l'extérieur.

Le rôle de ce stub est de se connecter à l'OC, puis d'écrire les requêtes sur le socket.

Le rôle de ce skelton est de lire le socket pour traiter les requêtes reçues.

On définit une interface d'utilisation de l'OC. L'interface est implémentée par le stub et utilisée par le skelton. Cette interface définit les traitements suivants :

- création des "datas"
- mise à jour de la valeur d'une des "datas"
- récupérer la valeur d'une des "datas"
- donner le code de l'action à exécuter par l'OC
- donner l'ordre de réaliser l'action

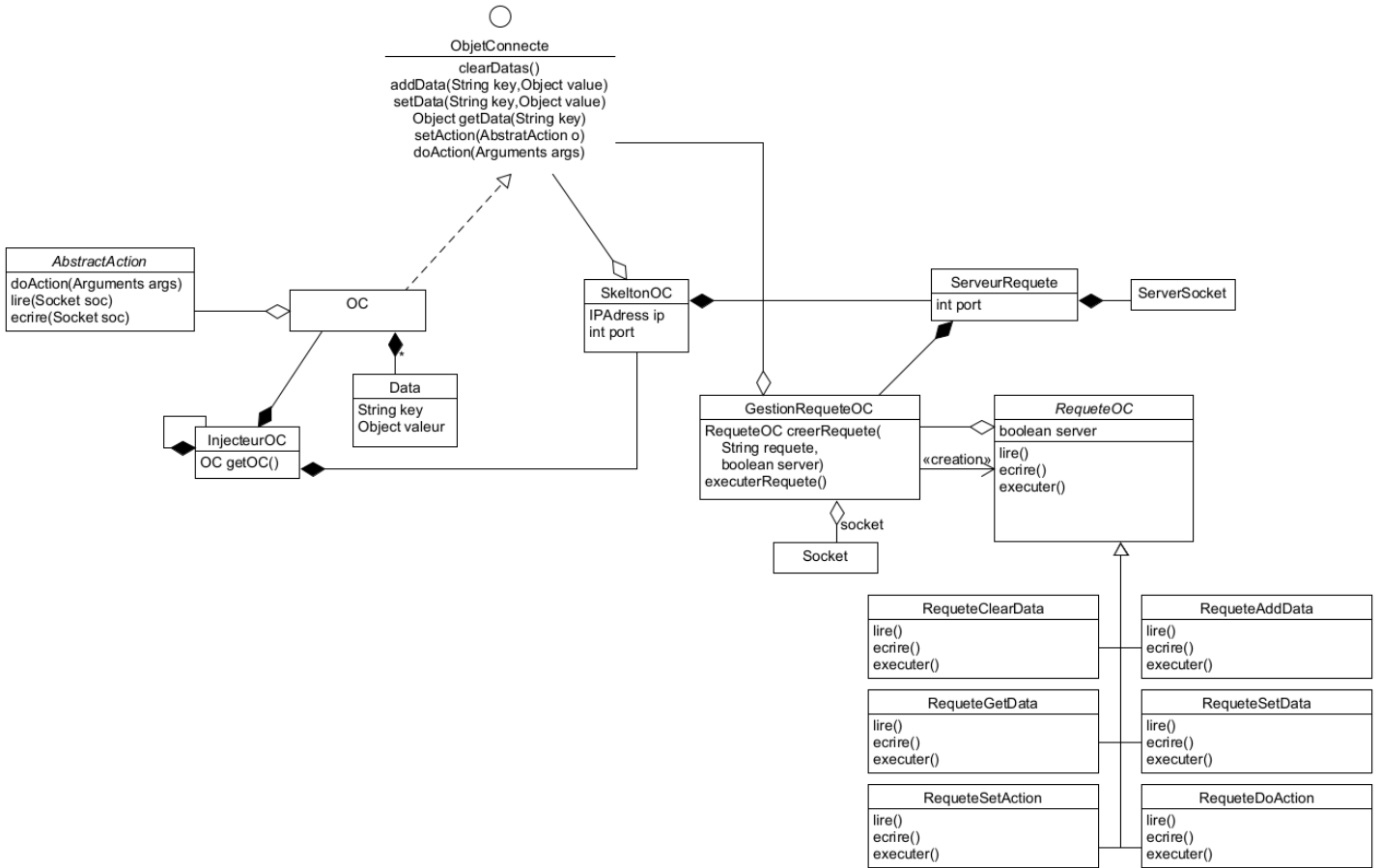
Le rôle du stub est de convertir l'appel d'une méthode de cette interface en une écriture sur un socket et lire la réponse.

Le rôle du skelton est de lire les requêtes et les convertir en un appel aux méthodes de l'OC et écrire la réponse.

Le code de l'action est transmis à l'OC sous la forme d'un objet qui est injecté dynamiquement dans le code de l'OC.

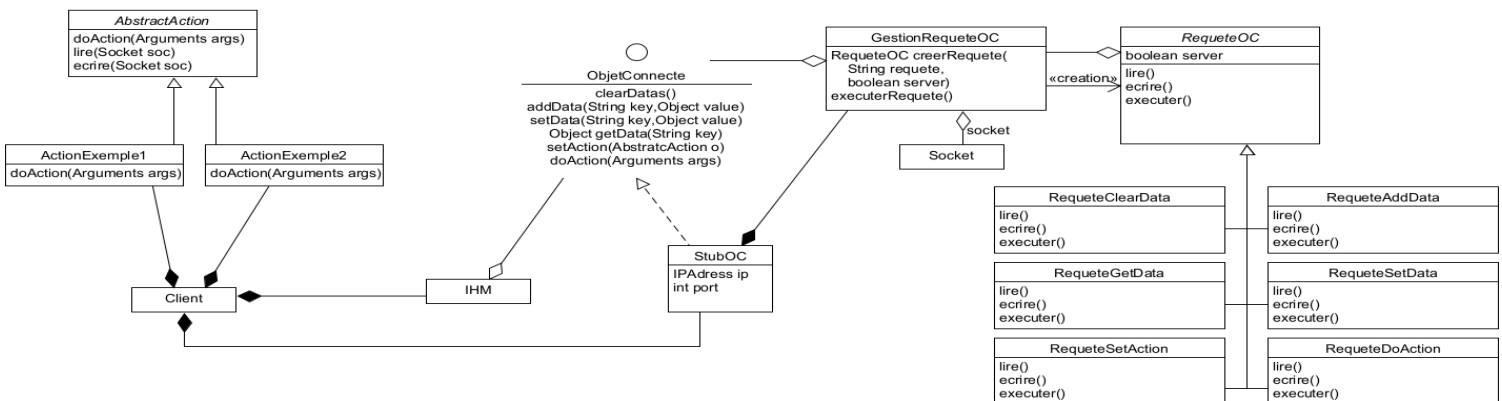
3. Correction

L'Objet Connecté :



Commentaires en cours

Un client qui utilise l'Objet Connecté :



Commentaires en cours