

Systemes et Applications Répartis (SAR)

TD 3 : Communication avec RPC/RMI

Exercice 1 (Réalisation d'un système réparti avec RMI)

Nous voulons réaliser une application répartie entre **trois partenaires P1, P2 et P3**. Chaque partenaire est **composé d'un objet distant (RMI)** ainsi qu'un **processus** (appelé souvent processus métier). Les objets distants représentent les services offerts par chaque partenaire. Le rôle du processus est d'**appliquer le protocole d'interaction** c.à.d d'invoquer les méthodes des partenaires selon le protocole décrit par le diagramme de séquences de la figure 1.

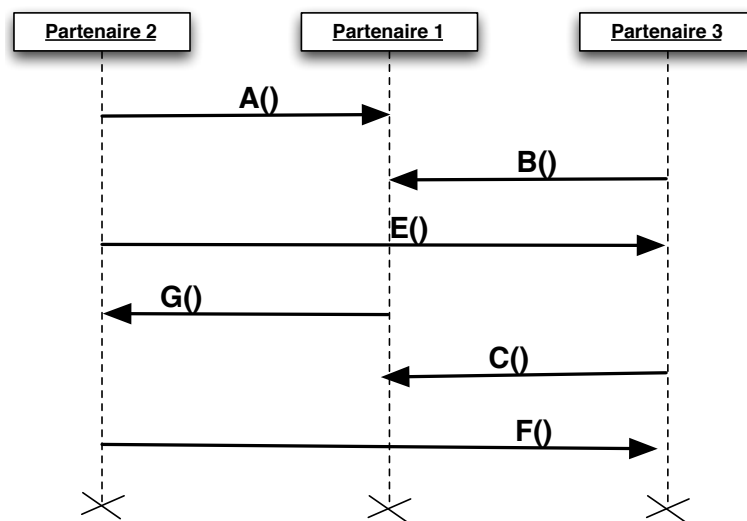


FIGURE 1 – Le diagramme de séquences de l'application répartie.

Partenaire 1 (P1) Le partenaire P1 offre un objet distant avec trois méthodes distantes A(), B() et C().

Le comportement du processus du partenaire P1 est le suivant :

1. Attendre que la méthode A() soit invoquée ;

2. Attendre que la méthode B() soit invoquée ;
3. Attendre que la méthode E() de P3 soit invoquée ;
4. Invoquer la méthode G() de P2 ;
5. Attendre que la méthode C() soit invoquée ;
6. Attendre que la méthode F() de P3 soit invoquée ;
7. Se terminer.

Partenaire 2 (P2) Le partenaire P2 offre un objet distant avec une seule méthode distante G().

Le comportement du processus du partenaire P2 est le suivant :

1. Invoquer la méthode A() de P1 ;
2. Attendre que la méthode B() de P1 soit invoquée ;
3. Invoquer la méthode E() de P3 ;
4. Attendre que la méthode G() soit invoquée ;
5. Attendre que la méthode C() de P1 soit invoquée ;
6. Invoquer la méthode F() de P3 ;
7. Se terminer.

Partenaire 3 (P3) Le partenaire P3 offre un objet distant avec deux méthodes distantes E() et F().

Le comportement du processus du partenaire P3 est le suivant :

1. Attendre que la méthode A() de P1 soit invoquée ;
2. Invoquer la méthode B() de P1 ;
3. Attendre que la méthode E() soit invoquée ;
4. Attendre que la méthode G() de P2 soit invoquée ;
5. Invoquer la méthode C() de P1 ;
6. Attendre que la méthode F() soit invoquée ;
7. Se terminer.

Question 1. Commencez par développer les trois types d'objets distants (interfaces et implémentations).

Appelez les interfaces OD1I, OD2I, OD3I et les implémentations OD1Impl, OD2Impl, OD3Impl.

Question 2. Proposez une solution afin de réaliser les protocoles de chaque partenaire c.à.d coordonner leurs interactions de telle manière que seul l'ordre d'interaction précisé par le diagramme de séquences soit possible.

Inspirez vous de la solution de l'exercice 1 du TD1.

Question 3. Réalisez votre solution et écrivez les processus de chaque partenaire.

Distribution via RMI

Question 4. Testez votre solution par groupe de trois partenaires.

Notes q3 :

RMI c'est du RPC, orienté objet

Principe

Etape 1 : interface service, serveur; décrit par un contrat -> exprimé par une interface

on doit les adapter car dans notre td, on ne peut pas les utiliser comme ceci. On doit ajouter des méthodes et exceptions sur les interfaces : extends Remote. On ajoute également, throw RemoteException;

Etape 2 : SOit on est coté client soit, coté serveur. QUELQUES soit le partenaire que l'on veut implémenter, on doit jouer le rôle serveur pour l'objet puis le client pour les autres serveurs. On commence par le serveur pour le client.

pour le serveur : on implémente le contrat

pour p1, j'offre que l'objet o1 (partie faite dans la q1). Plus qu'à faire pour RMI : extends UnicastRemoteObject pour les interfaces. Dans P1, on ajoute également les exceptions.

Vérifier que le constructeur est compatible avec le constructeur de classe mère. car cette dernière lève des exceptions.

Dans ce corrigé, on ne corrige que le P1. Les autres, P2 et P3. C'est la même procédure. dans le cas où je fais l'exo avec Mathis et Arthur, je ne devrais appeler qu'un P dans mon main.

Le processus ne change pas, il faut cependant gérer les exceptions des méthodes.

Etape suivante : Création du main