INP ENSEEIHT

Examen d'Intergiciels

Documents autorisés : tout document mis à disposition

Deuxième année Informatique et Réseaux

Durée : 1H45

11 avril 2012

1 Processus communicants et intergiciels (12 points)

Les question suivantes abordent des points généraux sur les processus communicants et les intergiciels.

Questions (1,5 points par question)

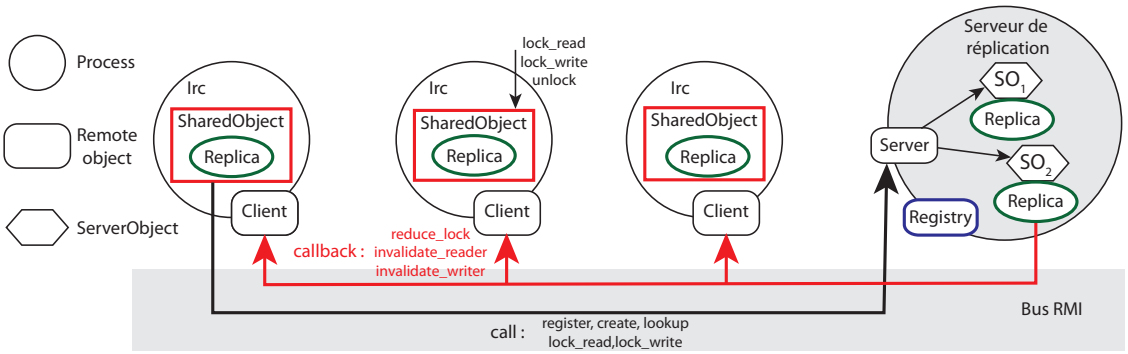
1. Quel service de base est nécessaire dans tous les intergiciels pour permettre la désignation des entités à distance : procédures, méthodes, files, thèmes, etc ?
2. Pour quelle raison essentielle distingue-t-on plusieurs types de sémantiques pour l'appel de procédure à distance ?
3. Comment la transparence d'accès à un objet distant est implantée en Java ? Expliquer le schéma d'implantation en l'occurrence que référence le client ? que fait le serveur ?
4. Quel problème pose les objets accessibles à distance vis-à-vis d'un ramasse-miettes ?
5. Enoncer d'une part les points communs, d'autre part, les différences essentielles entre les modèles d'exécution proposés par le modèle Linda et la spécification JMS
6. Comment rend-on persistant les thèmes (topics) ou files (queues) dans le modèle JMS ?
7. Les intergiciels implantent différents modèles de calculs répartis. Parmi les familles d'intergiciels suivantes : CORBA, Linda, MOM-JMS et Java RMI, vues en cours, précisez laquelle ou lesquelles répondent aux propriétés suivantes :
   - Modèle orienté objet accessible à distance ;
   - Modèle offrant le protocole de communication de type publication/abonnement (publish/subscribe) ;
   - Modèle traitant l'hétérogénéité.
8. Quel rôle joue le protocole IIOP dans la spécification de l'environnement Corba ?

2 Bus à objets répliqués : conception revisitée (8 points)

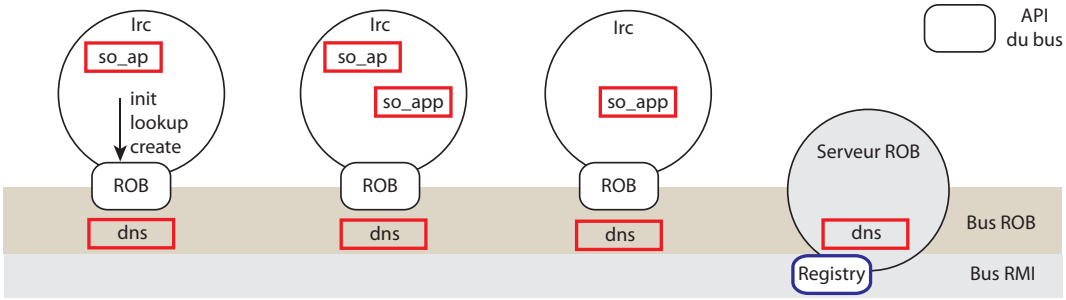
Le projet réalisé consistait à implanter un bus à objets répliqués en utilisant comme support un bus à objets accessibles à distance via le protocole Java RMI. L'objectif est ici de revisiter cette conception et de concevoir une solution un peu différente.

La solution proposée comporte en effet un serveur assimilable à un serveur de noms indispensable dans la plupart des intergiciels. Ce serveur assure en particulier la correspondance entre un nom externe d'objet répliqué et son moniteur de synchronisation (classe ServerObject).

La figure suivante rappelle l'architecture générale de cette conception :



Conception revisitée Nous avons pu constater que le service de nommage des intergiciels implantant l'appel de méthode à distance est lui-même considéré comme un objet accessible à distance (Registry pour Java RMI et NamingContextExt pour Corba par exemple). Par analogie, on peut donc envisager d'implanter le serveur de noms du bus à objets répliqués comme, lui-même, **un objet répliqué**. La figure ci-dessous illustre la démarche :



La classe ROB (Replicated Object Bus) implante l'API d'usage du bus. Cette API devrait rester très similaire à celle de la classe Client avec les méthodes statiques : init, lookup et create. L'objet répliqué dns de classe NameServer assure donc, quant à lui, un service de nommage entre tout nom externe d'objet répliqué et son moniteur de synchronisation. On propose (sans obligation) les interfaces suivantes :

```
public class ROB {
  public static void init() ;
  public static SharedObject lookup(String name) ;
  public static SharedObject create(String name, Object o) ;
}
public class NameServer {
  ServerObject lookup(String name) ;
  void register(String name, ServerObject monitor) ;
}
```

Question (8 points) : Vous devez décrire la conception générale de cette nouvelle implantation. Votre réponse est donc assez « ouverte » mais devra comporter les éléments de réponse suivants :

- la faisabilité de cette approche : soulève-t-elle des difficultés ou simplifie-t-elle au contraire l'implantation d'une solution ? Préciser, en particulier, qui crée l'objet répliqué « serveur de noms » et comment peut-il être repéré au niveau de la gestion du bus à objets répliqués ?
- la localisation du moniteur de synchronisation (ServerObject)de chaque objet répliqué applicatif créé.
- l'architecture générale de cette nouvelle solution. Décrire les interfaces et classes réutilisées, modifiées et/ou nouvelles de votre solution et préciser quelles classes sont accessibles à distance. Vous pouvez vous appuyer sur le schéma proposé en y apportant toute information supplémentaire jugée utile.
- les avantages et inconvénients d'une telle approche.