

Sujet d'examen Session de septembre 2002

Conception d'Applications Réparties

GMI 3^{ème} année

12 septembre 2002

Objets répartis

- 1 Qu'est-ce qu'une invocation de méthode semi-synchrone ? En quoi cela permet-il d'introduire de la concurrence dans les programmes ?
- 2 Dans le cadre de communications distantes, qu'est-ce qu'une souche ? Quel(s) type(s) de souche rencontre-t-on ?
- 3 Les protocoles client/serveur vus en cours (IIOP, RMI, HTTP, RPC DCE) ont en commun deux messages fondamentaux. A quoi servent-ils (il n'est pas nécessaire de donner explicitement les noms pris par ces messages dans chacun des protocoles cités) ?

CORBA

- 4 Qu'est-ce que le référentiel d'interfaces dans CORBA ?
- 5 Qu'est-ce qu'un mapping dans CORBA ?
- 6 Que permet de faire le langage IDL de CORBA ? Quel est son intérêt ?
- 7 Quelles sont les fonctionnalités de l'adaptateur d'objets dans un environnement CORBA ?
- 8 Une interface IDL CORBA peut-elle être implantée par plusieurs classes Java ?
- 9 Une classe Java peut-elle implanter plusieurs interfaces IDL CORBA ?
- 10 Que permet de faire un serveur de noms dans un environnement comme CORBA ?
- 11 Donner les éléments d'une référence d'objet CORBA (IOR) permettant d'acheminer une requête jusqu'à son destinataire.

Service CORBA de gestion d'un document partagé

On souhaite concevoir un service CORBA permettant à des clients de travailler simultanément sur un document partagé.

On considère que le document est géré par un serveur centralisé. Il s'agit de concevoir en IDL CORBA l'interface de ce service. Le cahier des charges spécifie que cette interface comporte les 4 méthodes suivantes :

- une méthode `openSession` prenant en paramètre d'entrée une référence de client (objet CORBA), un nom de document (chaîne de caractères) et retournant un identifiant de session (entier). Cette méthode permet au serveur de tenir à jour la liste des clients travaillant sur le document.
- une méthode `readDocument` prenant en entrée un identifiant de session et retournant le contenu du document sous forme d'un tableau d'octets.

- une méthode `writeDocument` prenant en entrée un identifiant de session et un tableau d'octets représentant le nouveau contenu du document.
- une méthode `closeSession` prenant en entrée un identifiant de session. Cette méthode permet d'indiquer au serveur que le client a terminé ses manipulations sur le document.

12 Donner en IDL cette interface.

13 En vous inspirant de ce qui se passe pour les API d'accès à un système de fichiers, décrire en français et en IDL au moins 2 exceptions pouvant être ajoutées à cette interface.

14 En supposant que l'on veuille que chaque client accède à une version à jour et cohérente du document tout en gardant un certain degré de concurrence dans le traitement des requêtes, proposer en français, en pseudo-code ou en Java une implantation pour les méthodes de cette interface.