UPMC
SORBONNE UNIVERSITÉS

SRCS : Systèmes Répartis Client/Serveur

TD 06 – Service de partage en CORBA

Janvier 2016

Exercice 1 : Espace de données partagées

Le but de cet exercice est de définir un serveur CORBA de données partagées (`ServDP`) dans lequel des clients viennent déposer et retirer des données. Un tel service peut servir par exemple dans un environnement distribué, à ce que des stations de travail déposent des travaux à effectuer, et qu'une ou plusieurs autres stations, en fonction de leurs disponibilités, récupèrent ces travaux et les exécutent éventuellement de façon concurrente. Ce service s'inspire du système Linda qui implémente un tel espace de données sous forme d'une mémoire partagée répartie. Il s'agit ici d'en réaliser une implantation sous forme d'objets CORBA.

Question 1

Un serveur `ServDP` stocke des couples `(cle,valeur)`, où clé est une chaîne de caractères et valeur une séquence d'octets. Le stockage de chaque valeur est ainsi identifié par une clé unique. Son interface offre les services suivants :

deposer : qui permet à un client de demander l'enregistrement d'un couple `(cle,valeur)`

retirer : qui permet à un client de récupérer la `valeur` et
le 1er paramètre est la clé à retirer. La valeur associée à la clé est fournie au client par un 2ème paramètre.

lire : idem retirer sauf que le couple clé – valeur reste stocké.

Elle comprend de plus l'accès à un attribut de type entier contenant le nombre de couples clé – valeur stockés.

Définir l'interface CORBA IDL correspondant à cette spécification. Si l'hypothèse sur l'unicité des clés stockées dans l'espace est relâchée (i.e. on considère que plusieurs clés identiques peuvent être stockées simultanément), quelles en sont les conséquences sur l'interface IDL ?

Question 2

En se replaçant dans l'hypothèse d'une clé unique, définir des exceptions pouvant survenir pour chacune des méthodes de l'interface.

Question 3

Proposer une autre signature pour les méthodes `retirer` et `lire` qui, à l'aide de la valeur de retour, indique si la clé est présente ou non. Discuter les avantages et les inconvénients des deux approches. On suppose maintenant que certaines valeurs de clé peuvent être rejetées car contenant des caractères non valides. Comment prendre en compte cette situation ?

Question 4

En ne considérant pas les exceptions, quelles méthodes de l'interface définie en 1 pourraient être déclarées `oneway` ? Rappeler les caractéristiques des appels de méthodes `oneway`.

Question 5

On considère maintenant que la valeur peut être soit une valeur nulle, soit une séquence d'octets. Définir le profil des méthodes prenant en compte ce changement.

Question 6

Jusqu'à présent, les clients qui ont besoin de lire ou de retirer des données n'ont d'autres alternatives, lorsque la donnée est absente, que d'interroger périodiquement le serveur. Cette interrogation réalisée à distance, est potentiellement coûteuse. On souhaite mettre en place un mode plus interactif dans lequel, en cas d'absence de données, les clients sont prévenus dès que cette donnée devient disponible. Mettre en place les éléments IDL qui permettent de faire cela.

Question 7

On se place maintenant dans le cas où une station maître, ayant déposé dans l'espace de données une tâche à effectuer, veut être renseignée sur l'état d'activité de la station esclave ayant pris en charge la tâche : essentiellement, on veut savoir si la station esclave est plantée ou si elle est toujours en marche. Un tel mécanisme de détection permet, dans les systèmes tolérant les fautes, de pouvoir réagir (typiquement re-affectation de la tâche à une nouvelle station esclave) en cas de défaillance. Proposer deux mécanismes permettant de réaliser cette détection et définir les interfaces IDL les spécifiant.

Question 8

Donner l'implantation en Java ou en pseudo-code des deux solutions définies à la question précédente.

