

Architectures Logicielles

Programmation par composants

Séance de TP n°1

Thème : Composant, Composite et Connecteurs

L'objectif de cette séance est de comprendre les notions de composant simple, de composant composite et de connecteur. Pour cela, nous vous proposons la progression suivante : définir un composant simple, définir un composant composite, puis définir divers connecteurs interposés entre les composants.

Exercice 1 – Composant Simple

Comme cela vous l'a été démontré en cours, définissez un composant simple offrant une seule interface dans un fichier SpeADL ainsi que l'interface Java qui permet de la typer. Définissez ensuite l'implantation de ce composant et réalisez un petit programme de test.

Exercice 2 – Composant composite

A la manière de ce qui a été fait en cours, définissez un composant composite qui contient 2 composants, l'un client, l'autre serveur. Comme l'usage le nécessite, définissez les interfaces, les implantations de composants et le programme de test.

Exercice 3 – Connecteurs

1. A partir de la configuration précédente (ou bien à partir d'une configuration équivalente), définissez un composant composite qui contient 2 composants reliés l'un à l'autre dont la fonctionnalité de cet assemblage est (bêtement) la suivante : lorsque son service est invoqué, le composant client génère une chaîne de caractères « Hello » qu'il transmet au serveur. Lorsque le serveur reçoit cette chaîne, il l'encapsule entre une chaîne caractérisant son identité et la chaîne « World ! », puis affiche le résultat à l'écran (ce qui donnerait par exemple « serveur1 : Hello World ! »).
2. L'objectif est maintenant d'intégrer du cryptage dans la communication entre les composants pour sécuriser l'application, bien sûr sans modifier les définitions et implantations des composants Client et Serveur. Définissez donc deux nouveaux composants responsables du chiffage (composant Crypte) et du déchiffage (composant Décrypte) et donnez un nouvel assemblage qui produit toujours le même résultat (affichage de « serveur1 : Hello World ! » à l'écran ;-))
3. A partir de l'assemblage de la question 1, on veut maintenant atteindre un autre objectif qui consiste à répliquer le serveur d'affichage. On cherche donc à obtenir un assemblage contenant 3 serveurs aux identités différentes. Proposez un nouvel assemblage de composants qui ne modifie pas client et serveur et permettent de diffuser le message du client aux serveurs (et donc d'afficher 3 messages à l'écran).

4. Généralisation et composition automatique : on veut maintenant être capable de combiner les modifications précédentes (Q2 et Q3). Etes-vous capable de réutiliser les composants précédemment définis sans modification pour définir un assemblage qui permette à la fois le chiffrement des communications et la réplication des serveurs ? N'y a-t-il qu'un seul assemblage qui permette cette composition des mécanismes de communications ? Est-il possible de généraliser votre solution à un nombre quelconque de serveurs et/ou de clients ? Est-il possible de généraliser votre solution entre des composants quelconques qui communiquent entre eux ?