

Examen

Composants répartis

Documents autorisés. 1 heure 30.

1 Exercice Composants CORBA

La définition OMG IDL suivante décrit un type d'événements, un type de composants publiant des événements via le port *source* et un type de composants consommant des événements via le port *puît* :

```
eventtype Evenement { ... };  
component Emetteur { publishes Evenement source; };  
component Recepteur { consumes Evenement puit; };
```

- 1.1 Spécifier le descripteur d'assemblage de composants (Component Assembly Descriptor) permettant d'instancier deux composants *Emetteur* et trois composants *Recepteur* et de les interconnecter afin que les composants *Recepteur* reçoivent les événements émis par tous les composants *Emetteur*.

Un descripteur d'assemblage contient explicitement la description de toutes les interconnexions entre tous les composants. Ainsi si l'application est formée de E composants *Emetteur* et R composants *Recepteur*, il est nécessaire de décrire les E*R interconnexions entre les composants.

- 1.2 Spécifier un nouveau type de composant CORBA permettant de minimaliser le nombre d'interconnexions à décrire pour assembler des composants *Emetteur* et *Recepteur*.
- 1.3 Donner le nouveau nombre d'interconnexions à spécifier dans le descripteur d'assemblage de E composants *Emetteur* et R composants *Recepteur* lorsque le nouveau composant est utilisé.
- 1.4 Décrire l'implantation de ce nouveau composant.

Durant l'exécution de l'application, de nouveaux composants *Emetteur* et *Recepteur* peuvent être créés dynamiquement. On désire que ces nouveaux composants soient interconnectés aux composants existants au préalable.

- 1.5 Décrire en Java une fonction permettant d'interconnecter un nouveau composant *Emetteur* aux composants *Recepteur* existants. Attention, cette fonction doit utiliser les opérations générées par la projection OMG IDL 3 vers OMG IDL.

La spécification du modèle de composants CORBA n'impose aucune contrainte sur la technologie utilisée pour le transport des événements. Par défaut, le conteneur des composants met en œuvre le service de notification. Toutefois les événements pourraient très bien être transportés via toute technologie de communication asynchrone comme par exemple un Message Oriented Middleware (MOM), la messagerie Internet, un protocole multicast, etc.

- 1.6 Proposer une extension du modèle de composants CORBA permettant de décrire et configurer les mécanismes de transport d'événements à mettre en œuvre à l'exécution. Identifier clairement l'impact de cette extension sur les descripteurs XML, le processus de déploiement et les conteneurs.

2 Exercice Fractal

2.1 Proposer une définition pour les notions d'interfaces métier et d'interfaces de contrôle présentes dans Fractal.

On souhaite mettre en œuvre une application Fractal qui permette de détecter la panne de sites distants organisés selon une topologie réseau quelconque. Cette détection de panne doit se faire par *heartbeat* : périodiquement, chaque site envoie à tous ses pairs un message `alive()` pour signaler qu'il est en vie. Si au bout d'un délai fixé à l'avance, un site n'a pas reçu le message `alive()` d'un de ses pairs, il le soupçonne d'être en panne.

On doit pouvoir interroger un site pour :

- lui demander la liste de ses pairs en panne,
- arrêter/démarrer le détecteur de panne sur ce site.

Chaque site doit aussi gérer les messages `alive()`.

2.2 Définir un ou plusieurs composants Fractal permettant de mettre en œuvre cette application. Pour chaque composant, préciser ses interfaces (définir en quelques mots la fonction de chaque interface). Pour chaque interface, préciser ses méthodes (définir en quelques mots la fonction de chaque méthode) et ses propriétés (client/serveur, métier/contrôle, obligatoire/facultative, interne/externe, singleton/multiple).

On souhaite pouvoir configurer de façon uniforme l'ensemble des détecteurs : par exemple définir la même valeur entre 2 envois successifs de `alive()` sur tous les sites.

On précise que les interfaces de contrôle Fractal ne sont pas accessibles à distance.

2.3 Spécifier (i.e. décrire en français) les éléments (interfaces, propriétés, méthodes, pseudo-code) à mettre en place pour implanter cette fonctionnalité. La spécification devra être suffisamment précise pour pouvoir être programmé par un développeur Fractal.