

# DEA ISC

## Construction des Applications Réparties

(mars 1998)

Des réponses précises et concises sont demandées.

L'examen porte sur l'analyse de systèmes de type *bus à événement* (appelé aussi *bus d'information*, ou encore *bus applicatif*) pour la mise en œuvre d'applications réparties.

Le support de l'examen est constitué des trois documents suivants :

1. L'article intitulé « *Publish or Perish* », paru en septembre 1997 dans la revue BYTE, présente de façon générale la problématique d'une approche de la construction des applications réparties fondée sur un *bus à événements*. Cette approche est ensuite illustrée par deux réalisations particulières, décrites dans les deux autres documents (pages 1 à 6).
2. Système *TIB/Rendezvous* (pages 7 à 18).
3. Système *ActiveWeb* (pages 19 à 29)

Dans les questions qui suivent on sera amené à faire référence à certains éléments de ces documents (ils seront alors identifiés par le numéro de page correspondante).

L'examen est structuré en trois parties :

- des questions d'ordre général sur les architectures de type *Publish and Subscribe*
- des questions portant sur le système *TIB/rendezvous*
- des questions portant sur le système *ActiveWeb*

### **Partie I – Etude de l'architecture P&S**

#### **Question 1.1**

Résumez brièvement les propriétés respectives des architectures de type *client-serveur (C&S)* et de type *Publish-Subscribe (P&S)*. On prendra plus particulièrement en compte les critères d'évaluation suivants :

- Construction d'applications réparties
  - Modèle de programmation et outils de développement pour les concepteurs d'applications réparties
  - Intégration des applications existantes
- Mise en œuvre du système
  - Structure du système sur les machines concernées,
  - Capacité de croissance dynamique (introduction de nouveaux utilisateurs du service)

## Question 1.2

Le document N°1 fait référence en page 5 à trois manières de mettre en œuvre une architecture *P&S*. Ces trois possibilités sont désignées sous les termes :

- *Bus*
- *Hub and Spoke*,
- *Snowflake*

Décrire le principe de mise en œuvre de chacune de ces trois approches. On prendra en compte les éléments de comparaison suivants :

- Implantation du système sur les machines
- Gestion des abonnements (opérations *Subscribe*)
- Gestion des événements (opérations *Publish*)
- Gestion des communications,

## Question 1.3

Le document N°1 fait référence en page 3 à deux modes de fonctionnement pour informer des processus consommateurs (*consumers*) de l'occurrence d'événements émis par des producteurs (*producers*) :

- Le mode *Pull* dans lequel les consommateurs interrogent périodiquement (*polling*) le service de gestion des événements (*Broker*) pour recevoir les messages émis qui les concernent.
- Le mode *Push* dans lequel le service de gestion des événements informe directement les consommateurs de l'occurrence d'un événement qui les concernent.

On s'intéresse ici à la mise en œuvre de ces deux modes de fonctionnement dans un système de type *client-serveur*. En d'autres termes, on veut « simuler » le fonctionnement d'un service de gestion d'événements selon les deux modes *Pull* et *Push* en utilisant un système *client-serveur*. On notera au passage que c'est la solution choisie par l'OMG (Object Management Group) pour fournir un service de type « communication par événement » sur une architecture de type *Corba*.

On considère donc un système constitué des éléments suivants :

- Des processus producteurs, notés *Pi*,
- Des processus consommateurs, notés *Cj*,
- Un service de gestion d'événements, noté *B*.

Les *Pi*, les *Cj* et *B* sont supposés être implantés sur des machines distinctes. Décrire le fonctionnement du service *B* et les interactions avec les composants *Pi* et *Cj* pour réaliser les modes *Pull* et *Push* (on notera que les *Pi* et les *Cj* sont « clients » du serveur *B*).

## Partie II – Etude du système TIB/rendezvous

### Question 2.1.

A quelle catégorie appartient le système *TIB/Rendezvous* (dans la classification de la question 1.2) ?

Question 2.2

En utilisant les différents schémas d'interaction offerts par le système *TIB/Rendezvous* (voir page 9), on désire réaliser un système de communication entre un professeur et ses étudiants, qui réponde aux spécifications suivantes :

- Le professeur, représenté par un processus noté  $P$ , soumet un certain nombre d'exercices (ou documents) à ses étudiants. Pour cela il stocke la description de l'exercice (ou du document) sur un serveur, noté  $S$ , connu du professeur et des étudiants. Il informe les étudiants de la disponibilité de l'exercice (ou du document) en publiant un événement significatif à l'attention des étudiants concernés.
- Sur réception d'un événement informant de la disponibilité d'un exercice (ou d'un document), les étudiants, représentés par un processus  $Ei$ , récupèrent l'exercice (ou le document) sur le serveur.
- Par la suite chaque étudiant résout l'exercice et stocke ultérieurement la réponse sur le serveur. Il informe ensuite le professeur de la disponibilité du résultat.
- Le professeur récupère les résultats sur le serveur pour les corriger.

On suppose que le processus  $P$  et les processus  $Ei$  n'ont pas de connaissance mutuelle directe.

### **Partie III – Système ActiveWeb**

#### **Question 3.1.**

A quelle catégorie appartient le système *ActiveWeb* (dans la classification de la question 1.2) ?

#### **Question 3.2.**

Caractériser, de façon générale, le rôle d'un « adaptateur » (*adapter*) ?

Décrire le principe de fonctionnement d'un adaptateur dont le rôle serait de transmettre un message (par courrier électronique) à un administrateur système sur détection d'une tentative d'intrusion d'un réseau Intranet par un système pare-feu (*firewall*).

#### **Question 3.3.**

Le système *ActiveWeb* gère des événements « éphémères » et des événements « persistants ».

- Caractériser simplement chaque catégorie.
- Décrire une méthode simple pour gérer des événements persistants ?

#### **Question 3.4.**

Expliquer l'objectif et le principe des opérations suivantes liées à la sécurité :

- *Authentication* d'un producteur/consommateur auprès du service de gestion des événements (*Broker*),
- *Autorisation* pour un producteur/consommateur d'effectuer certaines opérations,

Comment ces contrôles pourraient-ils être appliqués dans le cas d'un environnement client-serveur ?