

GIS-IMA 3^{ème} année

Devoir Surveillé Architectures Logicielles

© Olivier Caron

Durée 2 heures

1 Programmation par composants Java Beans (8 points)

Le déroulement d'un projet se décline en une succession d'activités. Plusieurs activités peuvent être actives au même moment. Certaines activités nécessitent que d'autres soient terminées pour commencer. Pour un projet, une activité est donc caractérisée, par un *état* qui est, soit en cours d'exécution, soit en attente de la fin d'exécution d'autres activités, soit terminée. Selon le projet à réaliser, une activité se mettra *automatiquement* en cours d'exécution que lorsque les activités dépendantes seront terminées.

Ainsi, dans l'exemple de la figure 1, l'activité a3 se mettra en route lors de la fin des activités a1 et a2. Une activité peut être nécessaire à plusieurs autres activités, c'est le cas pour l'activité a1 qui est nécessaire aux activités a3, a4 et a6.

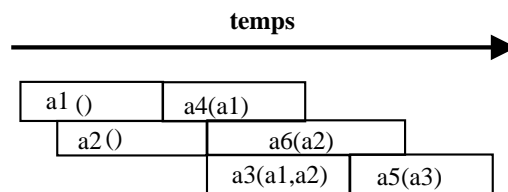


FIG. 1 – Exemple de séquençage par activités d'un projet

On veut réaliser une application à base de composants qui permet de gérer le suivi d'un projet. On ne se préoccupe pas de ce que fait exactement cette activité ; on veut surtout savoir essentiellement l'état des différentes activités participantes au projet.

Nous désirons donc programmer le composant Java Beans de nom `Activite` qui simule une activité. Cette programmation doit atteindre deux objectifs. Le premier est que ce composant puisse être réutilisé dans n'importe quel projet (avec des dépendances entre activités différentes). Le second est que la phase de mise en route des activités dépendantes d'autres activités se fasse automatiquement. Pour cela, ce composant dispose des propriétés suivantes :

- la propriété `designation` de type `String` qui décrit le traitement à réaliser
- la propriété entière `nbreDependances` qui indique, pour un projet particulier, le nombre d'activités dépendantes (exemple : 2 pour l'activité a3, 0 pour l'activité a2)
- la propriété entière `liee etat` qui peut prendre les valeurs constantes : `ACTIVE`, `EN_ATTENTE`, `TERMINEE`.

Question 1 : écrire le composant `Activite`.

Question 2 : écrire le code JAVA qui, pour des variables a1, a2, a3, a4, a5 et a6 de type `Activite` configure ces activités pour initialiser le projet décrit dans la figure 1

2 Architecture logicielle distribuée (12 points)

On désire concevoir une architecture client/serveur multi-sites qui permet de gérer l'achat de pièces automobiles. Cette architecture utilisera la technologie EJB et uniquement celle-ci (pas de composants web).

Les entreprises automobiles font appel à différents fournisseurs pour les différentes pièces composantes d'une voiture. Ces sociétés peuvent fournir plusieurs entreprises. La codification des pièces a fait l'objet d'une standardisation, ce qui permet d'identifier n'importe quel pièce à l'aide d'un simple code.

Sociétés automobiles et fournisseurs sont bien évidemment situés à des emplacements physiques différents (cf figure 2).

L'architecture logicielle distribuée à mettre en place doit assurer les points suivants :

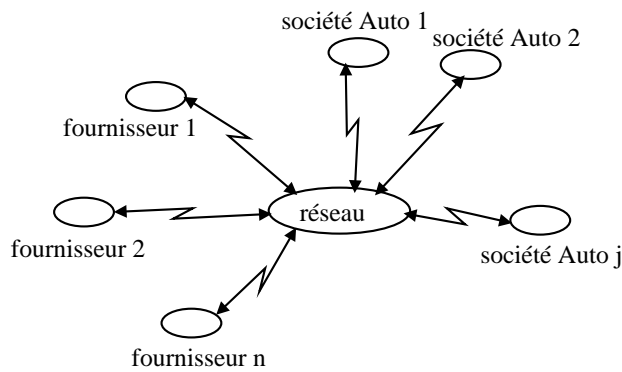


FIG. 2 – Répartition distribuée des sites

- Chaque fournisseur doit fournir une description de l'ensemble des pièces gérées, la quantité disponible pour chaque pièce et, bien sûr, proposer l'achat de ces pièces.
- L'une des contraintes, est que les sociétés automobiles n'ont pas à connaître l'ensemble des adresses réseaux des différents fournisseurs. L'idéal serait pour ces sociétés, de disposer, à partir d'un point d'entrée unique, des fonctionnalités suivantes :
 - de connaître les fournisseurs qui peuvent fournir une quantité donnée d'une pièce donnée.
 - de connaître le ou les fournisseurs qui propose le prix le moins cher pour une pièce donnée
 - d'acheter un nombre donné d'une pièce donnée.

Question 3 : Proposez une architecture logicielle qui répond à ce problème. Décrire le rôle de chaque application et composant introduits. Détaillez comment ces composants s'articulent entr'eux (vous pouvez donner des exemples). Décrire également par quel moyen, il est possible d'accéder à tel composant. Décrire pour chaque composant introduit, les éléments accessibles (fabrique et interface). Utilisez la norme EJB pour spécifier ces interfaces.

Question 4 : Détaillez la fonctionnalité *connaître le ou les fournisseurs qui propose le prix le moins cher pour une pièce donnée* à l'aide d'un scénario. Dans ce scénario, il faut spécifier et décrire les différents objets concernés, décrire également comment accéder à ces objets.

Remarque : vous pouvez utiliser le formalisme des diagrammes de séquences UML qui permet de décrire des scénarios. Un exemple est donné dans la figure 3. Cet exemple montre qu'un objet 1 appelle la méthode m1 sur l'objet 3 puis appelle la méthode m2 sur l'objet 2, ce qui déclenche dans cette méthode l'exécution de la méthode m21 qui retourne le résultat 0.

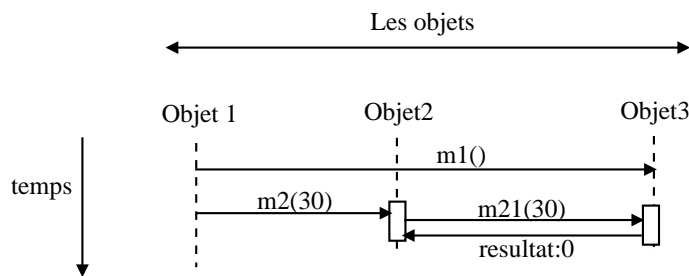


FIG. 3 – Exemple de diagramme de séquence