

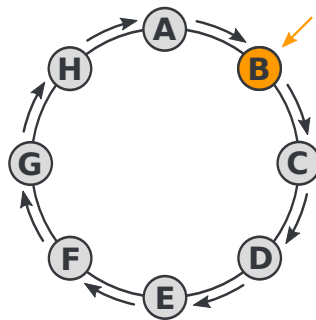
## SRCS : Systèmes Répartis Client/Serveur

### TD 07 – Service d'élection en CORBA

Janvier 2016

#### Exercice 1 : Election sur un anneau d'objets CORBA

On considère un nombre quelconque d'objets CORBA reliés selon une topologie logique en anneau. Chaque objet connaît un voisin droit et un voisin gauche avec qui il est capable de communiquer. Chaque objet gère donc deux variables (voisinDroit et voisinGauche) qui sont des références d'objets CORBA. Un identifiant unique (un entier) est attribué à chaque objet. L'élection est un mécanisme qui consiste à élire l'objet d'identifiant le plus élevé (pour par exemple, lui attribuer un privilège). L'élection est réalisée à l'aide d'un message qui « fait le tour » de l'anneau. Elle peut être déclenchée par n'importe quel objet de l'anneau qui dans ce cas, s'appelle un initiateur. Deux types d'élection sont possibles : en tournant dans le sens des aiguilles d'une montre ou dans le sens inverse. A la fin de l'élection, tous les objets de l'anneau doivent connaître l'identifiant de l'élu.



#### Question 1

Quel est le test d'arrêt de l'élection i.e. du « tour de l'anneau » ? Quelle information faut-il inclure dans le message qui fait le « tour de l'anneau » pour pouvoir réaliser ce test ?

#### Question 2

Représenter sur un diagramme les échanges de messages engendrés par l'exécution d'une élection avec l'objet 2 comme initiateur.

#### Question 3

Quel est le test de décision de l'élu? Quelle information faut-il inclure dans le message qui fait le « tour de l'anneau » pour pouvoir réaliser ce test?

#### Question 4

Prendre le schéma de l'anneau avec les cinq objets donnés ci-dessus et indiquer sur chaque message la valeur de ces deux informations.

#### Question 5

A la fin d'un tour, quel(s) objet(s) connaît(ssent) l'identifiant de l'élu? Pourquoi?

#### Question 6

Proposer deux solutions, une à base de message synchrone, une à base de message asynchrone, pour que tous les objets de l'anneau connaissent l'identifiant de l'élu.

#### Question 7

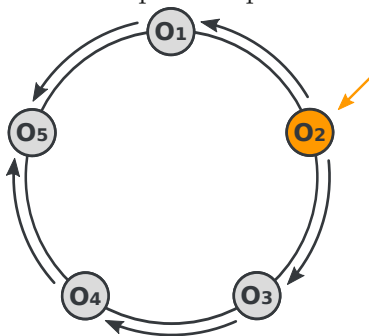
Sur chaque objet, le « tour de l'anneau » correspond à une méthode election prenant en entrée les informations définies aux questions 1 et 2. Pour chacune des solutions proposées à la question précédente, définir l'interface IDL CORBA correspondant à un objet de l'anneau.

#### Question 8

Pour chacune des solutions proposées à la question précédente, donner en Java ou en pseudo-code l'implantation de la méthode election.

### Exercice 2 : Election contrarotative sur un anneau d'objets CORBA

On considère un nombre quelconque d'objets CORBA reliés selon une topologie logique en anneau. L'élection consiste à désigner un site particulier au sein de cet anneau. On souhaite étendre l'algorithme d'élection de l'exercice précédent. On appelle « vague » le mécanisme de propagation d'objet en objet qui permet de réaliser l'élection. **L'élection n'est maintenant plus réalisée par une seule vague comme précédemment, mais par 2 vagues « qui tournent » en sens inverse (voir figure)**. Ainsi, l'objet initiateur d'une élection propage simultanément une vague vers la droite et une vague vers la gauche. Les objets intermédiaires propagent les vagues dans le sens où ils les reçoivent. Lorsque les deux vagues se rencontrent, elles refluent (via le message de retour associé à l'invocation de méthode) chacune de leur côté vers l'initiateur. Sauf pour la question 8, on suppose qu'il n'y a simultanément qu'un seul objet initiateur sur l'anneau.



Dans cet exemple,  $O_2$  est initiateur. Il propage 2 vagues :

1. une à gauche vers  $O_1$
2. une à droite vers  $O_3$

Lorsque les vagues se rencontrent, elles refluent vers  $O_2$ . Chaque objet visité par une vague, choisit un nombre aléatoire. L'élu est l'objet qui a tiré le plus grand nombre. A la fin, l'initiateur proclame les résultats (identité de l'élu et valeur tirée).

#### Question 1

Représenter sur un diagramme les échanges de messages engendrés par l'exécution d'une élection avec l'objet  $O_2$  comme initiateur.

## Question 2

Définir à l'aide du langage IDL de CORBA, l'interface des objets membres de l'anneau.

## Question 3

Cette interface doit-elle obligatoire être différente de celle définie pour l'élection avec une seule vague ou peut-on réutiliser la même interface ? Justifier votre réponse.

## Question 4

On s'intéresse maintenant à un **test d'arrêt de la propagation**. On va donc étudier le mécanisme à mettre en place pour décider si un objet doit continuer à propager une vague ou si la vague doit être retournée.

On appelle antipodes, l'objet (dans le cas d'anneaux contenant un nombre pair d'objets) ou les objets (dans le cas d'anneaux contenant un nombre impair d'objets) positionnés sur l'anneau de façon symétrique par rapport à un objet donné. Par exemple, sur un anneau à 4 objets ( $O_1, O_2, O_3, O_4$ ), l'objet aux antipodes de  $O_2$  est  $O_4$ . De même, sur un anneau à 5 objets ( $O_1, O_2, O_3, O_4, O_5$ ), les objets aux antipodes de  $O_2$  sont  $O_4$  et  $O_5$ .

Peut-on supposer que les deux vagues se rencontrent toujours sur des objets situés aux antipodes de l'initiateur ? Pourquoi ?

## Question 5

Expliquer comment le test à mettre en place pour décider si un objet doit continuer à propager une vague ou si la vague doit être retournée peut être réalisé. Il est conseillé d'illustrer cette explication par un schéma.

## Question 6

On souhaite maintenant **comparer les performances** de cette nouvelle version de l'élection avec la version en une seule vague. On s'intéresse aux performances en nombre d'invocations de méthodes (1 invocation = 1 message d'appel + 1 message de retour) et en temps. On suppose que la version en une seule vague utilise  $(n - 1)$  invocations sur un anneau comprenant  $n$  objets : l'objet situé « avant l'initiateur » détecte la terminaison de la propagation et retourne l'invocation.

Par rapport à la version en une seule vague, la nouvelle version diminue-t-elle ou augmente-t-elle le nombre d'invocations de méthodes nécessaire à une élection ? De combien d'unités ? Il est conseillé de détailler explicitement le nombre d'invocations, éventuellement à l'aide d'un exemple.

## Question 7

Par rapport à la version en une seule vague, la nouvelle version améliore-t-elle ou détériore-t-elle le temps nécessaire à une élection ? Sans pour autant donner un compte exact, donner une estimation du gain ou de la perte.

## Question 8

En supposant, qu'il n'y a qu'un seul objet initiateur simultanément, donner en Java ou en pseudo-code, la classe d'implantation d'un objet de l'anneau.

## Question 9

On suppose maintenant que plusieurs objets peuvent initier simultanément une élection. Y a-t-il des problèmes nouveaux à gérer ? Si oui, le(s)quel(s) et expliquer alors (sans forcément donner un nouveau code), comment votre algorithme de la question précédente peut être modifié.