



DIAGRAMMES D'INTERACTION: SÉQUENCES COMMUNICATION

Mme Sfaxi Lilia
Mlle Zoubeir Najet

MCOO-Chapitre 4

L2ARS/SIL - 2011/2012

Diagrammes d'interaction

- Diagrammes dynamiques
- Aspect commun : **les messages** : sorte de communication entre deux entités (objet, acteur, sous-système)
- Différents diagrammes :
 - ✓ **Diagramme de Séquences**
 - ✓ **Diagramme de Communication**
 - ✓ Diagramme global d'Interaction
 - ✓ Diagramme de Temps

Diagramme de Séquences

- Le diagramme d'interaction le plus commun
- Représentation temporelle de l'échange des messages entre les objets: séquencement des messages
- Représentation d'un seul scénario (avec la possibilité de combiner plusieurs)
- Écoulement du temps du haut vers le bas

Diagramme de Séquences : Contenu

- Lignes de vie
 - ✓ Rectangle + ligne pointillée
 - ✓ Etiquette [objet]:[classe]
- Messages
 - ✓ Communication entre les lignes de vie
 - ✓ Peuvent être :
 - Envoi de signal
 - Invocation d'une opération
 - Création ou destruction d'une instance

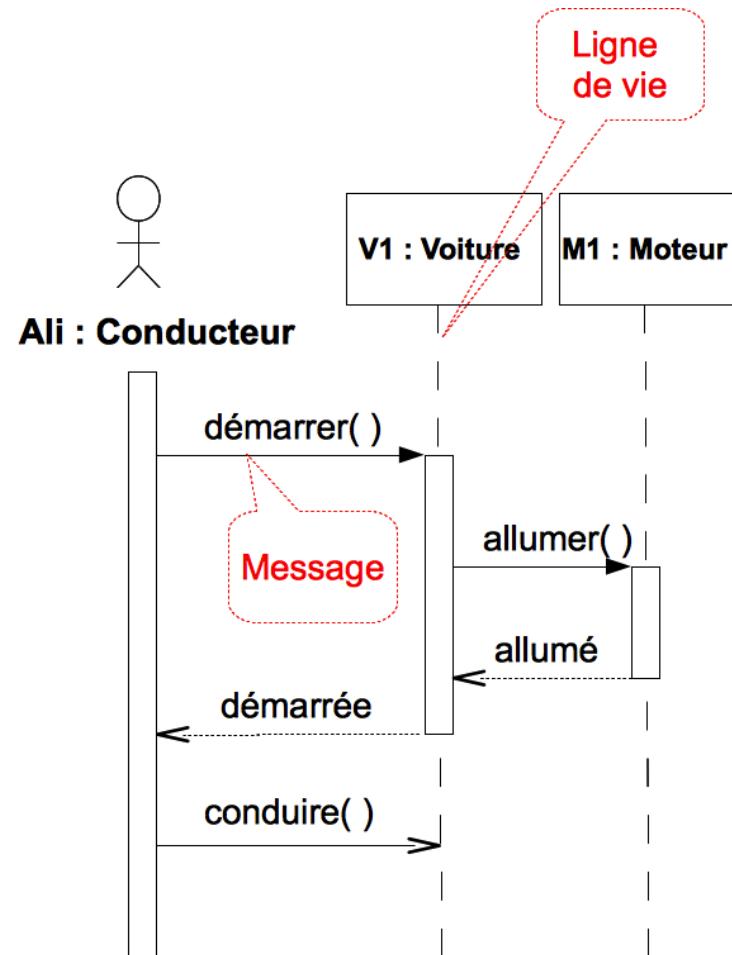


Diagramme de Séquences : Messages (1)

- Message Asynchrone
 - ✓ N'attend pas de réponse
 - ✓ Ne bloque pas l'émetteur
 - ✓ **Exemple** : signal (interruption, évènement)
 - ✓ **Représentation** : Flèche en traits pleins et à l'extrémité ouverte
- Message Synchrone
 - ✓ Emetteur bloqué jusqu'à la réponse du récepteur
 - ✓ **Exemple** : invocation d'une opération
 - ✓ **Représentation** : Flèche en traits plats à l'extrémité pleine; suivie d'une flèche en pointillés

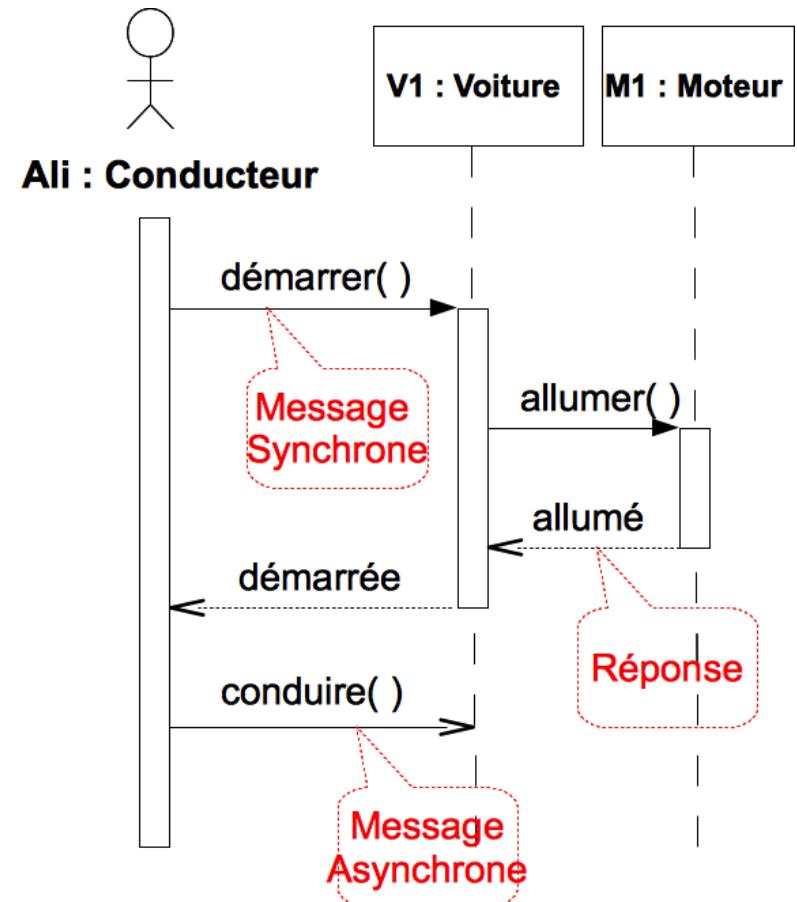


Diagramme de Séquences : Messages (2)

- Création d'instance
 - ✓ Création d'un objet qui n'existe pas
 - ✓ **Représentation** : flèche qui pointe sur le sommet d'une ligne de vie
- Destruction d'instance
 - ✓ Destruction d'un objet qui n'existera plus
 - ✓ N'est pas toujours provoquée par un message
 - ✓ **Représentation** : une croix qui marque la fin de la ligne de vie de l'objet détruit

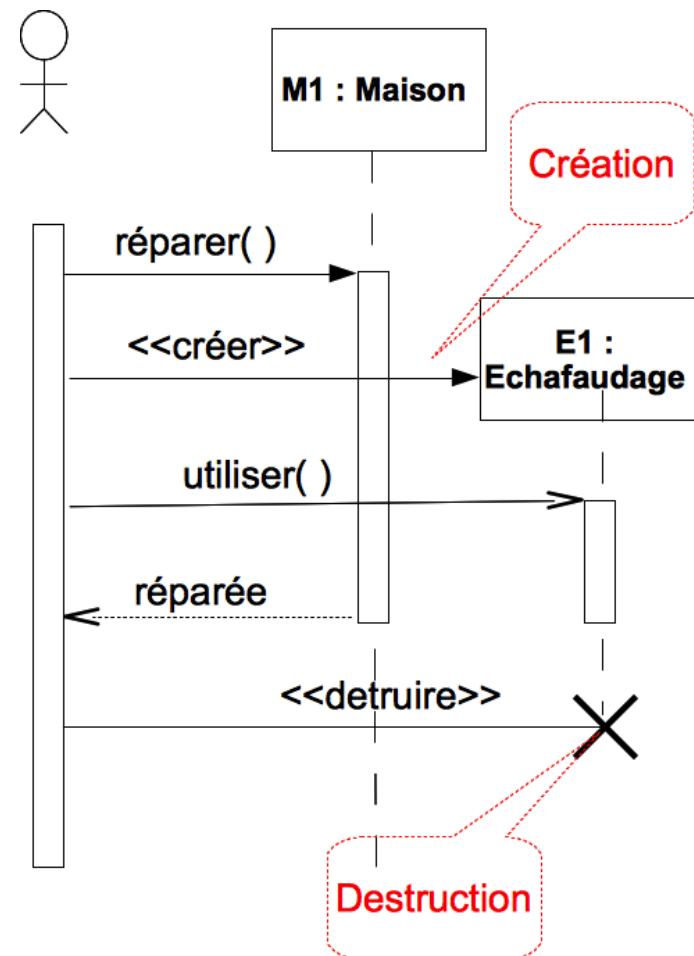


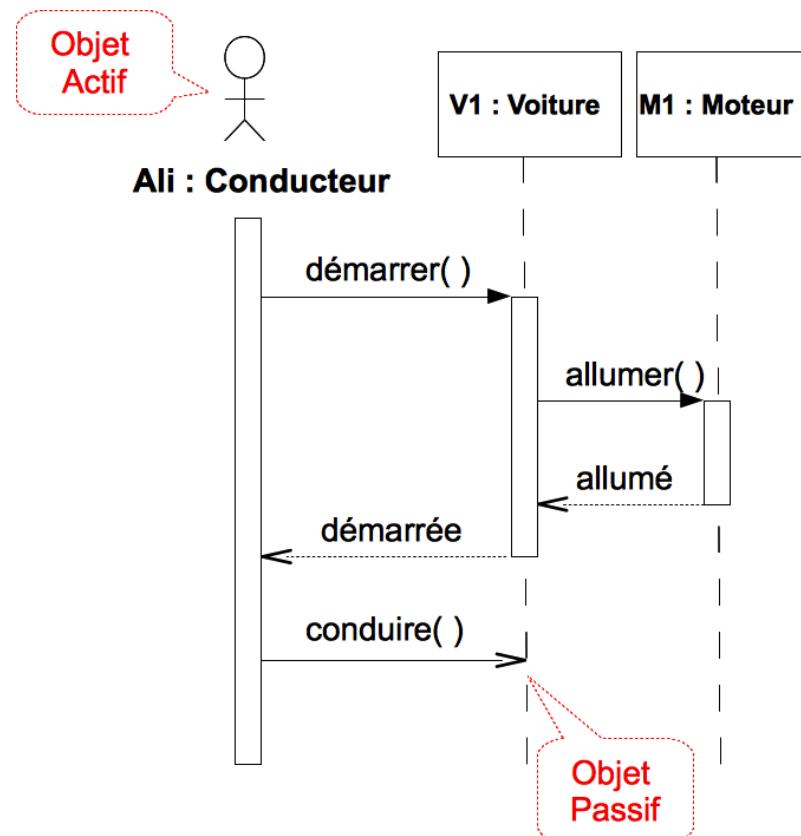
Diagramme de Séquences : Objets Actif et Passifs

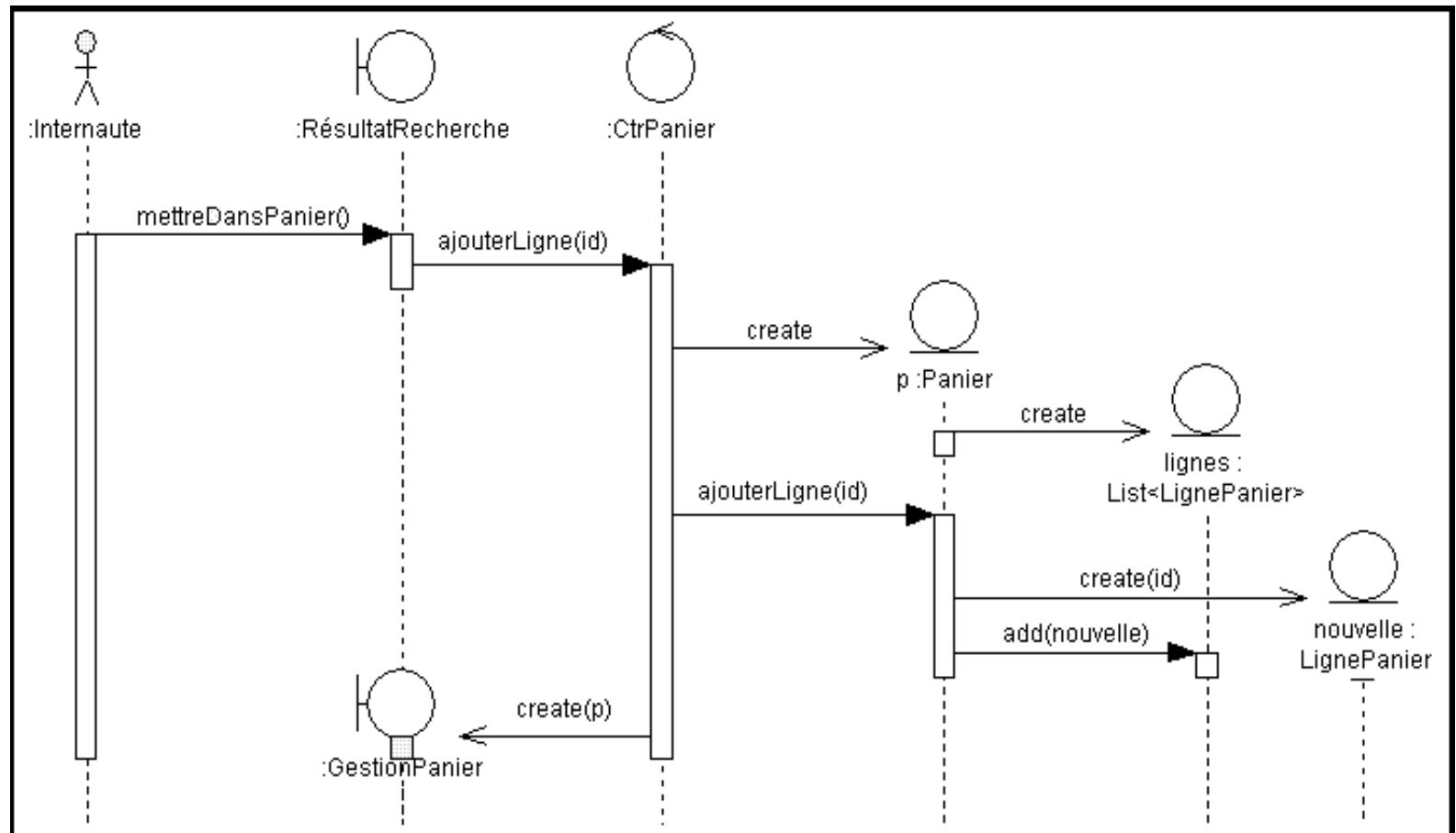
■ Objet actif

- ✓ Initie et contrôle le flux d'activités
- ✓ **Représentation** : un rectangle à la place de la ligne de vie verticale

■ Objet passif

- ✓ A besoin d'un flux d'activités pour pouvoir exécuter une méthode
- ✓ À l'exécution d'une méthode, un rectangle blanc est placé sur la ligne en pointillés





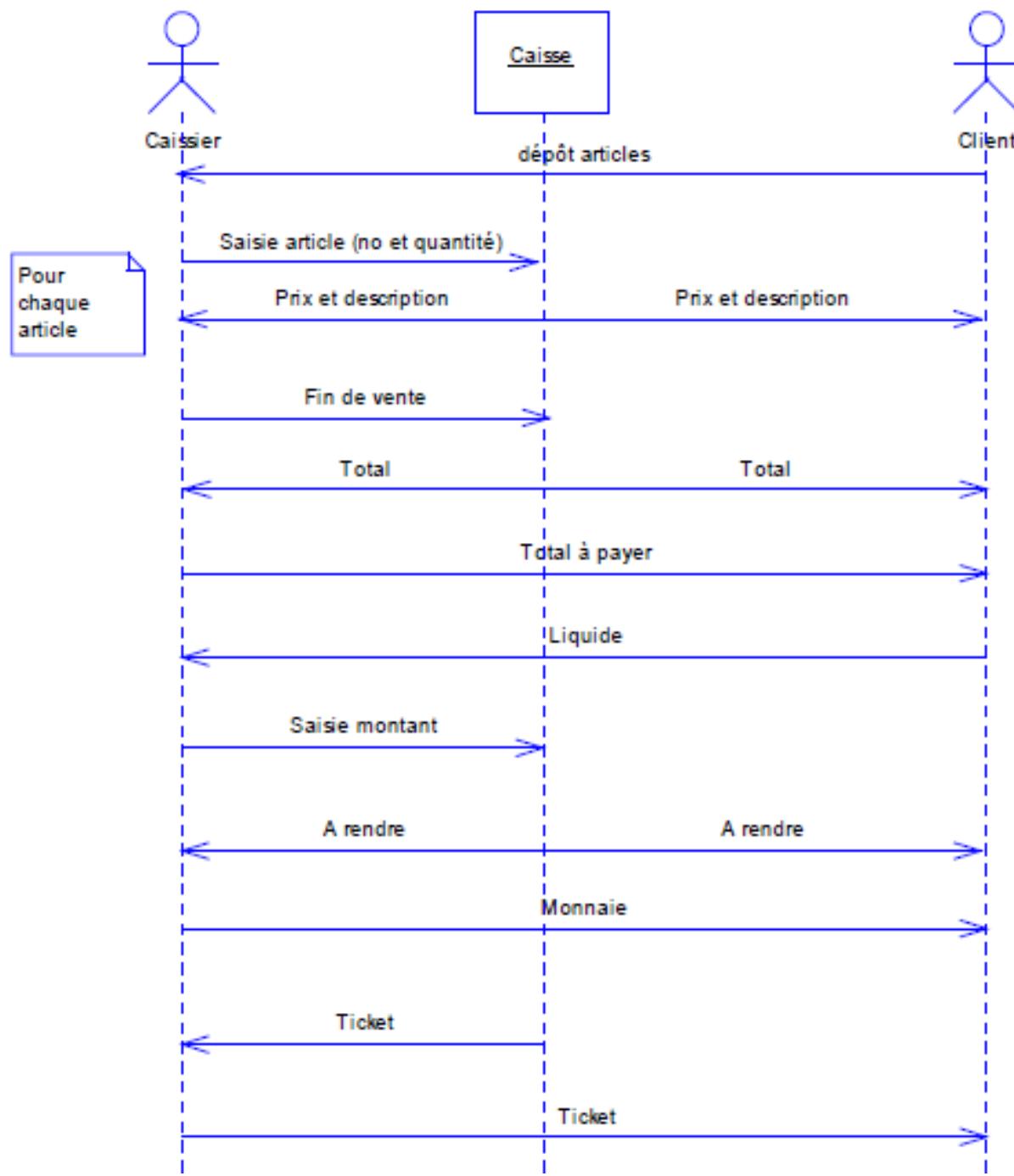
Exemple d'un Diagramme de Séquences

Exercice D'application

Le déroulement normal d'utilisation d'une caisse de supermarché est le suivant :

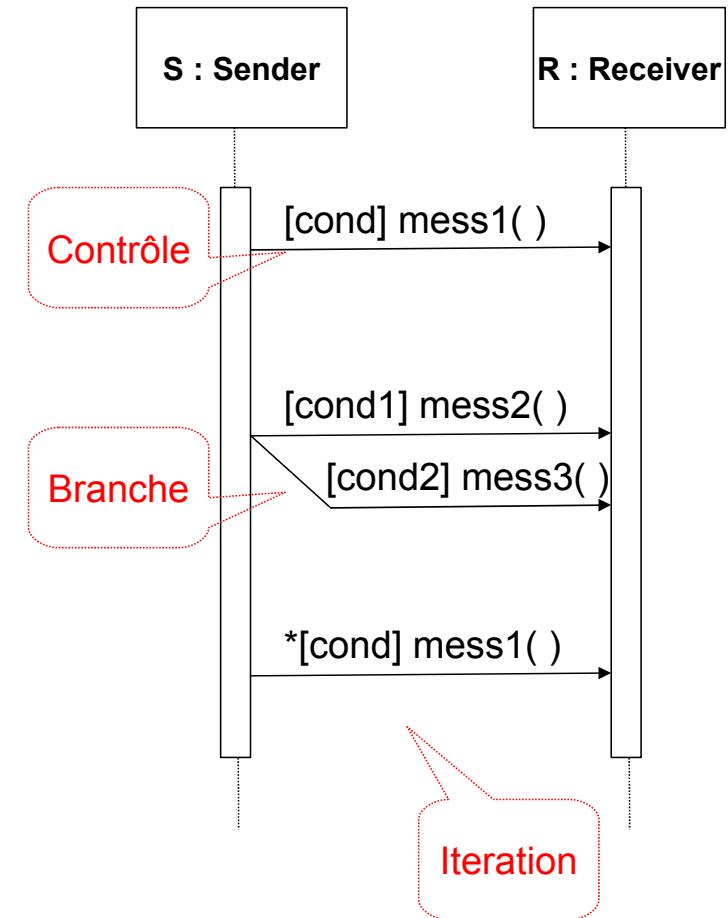
- un client arrive à la caisse avec ses articles à payer
- le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si elle est supérieure à 1
- la caisse affiche le prix de chaque article et son libellé
- lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente
- la caisse affiche le total des achats
- le caissier annonce au client le montant total à payer
- le client choisit son mode de paiement
 - ✓ liquide : le caissier encaisse l'argent, la caisse indique le montant à rendre au client
 - ✓ chèque : le caissier note le numéro de pièce d'identité du client
 - ✓ carte de crédit : la demande d'autorisation est envoyée avant la saisie
- la caisse enregistre la vente et l'imprime
- le caissier donne le ticket de caisse au client

Modéliser cette situation à l'aide d'un diagramme de séquence en ne prenant en compte que le cas du paiement en liquide.



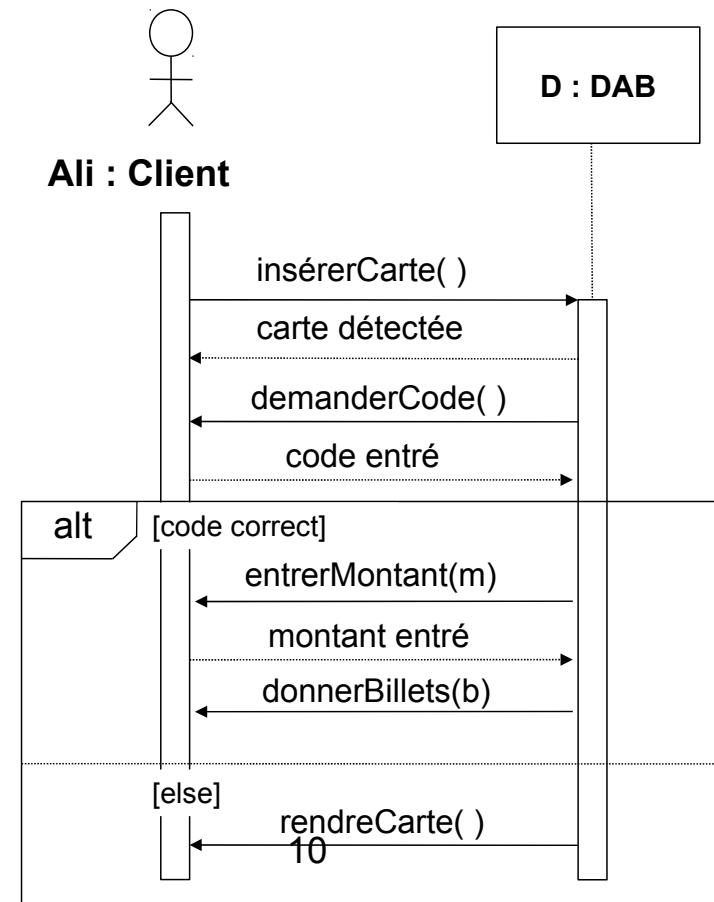
Structures de Contrôle

- Structure de contrôle
 - ✓ mess1 envoyéssi la condition de garde [entre crochets] est respectée
- Branche
 - ✓ On envoie soit mess2 soit mess3, selon les conditions
- Itération
 - ✓ Le mess4 est envoyé tant que la condition est vraie



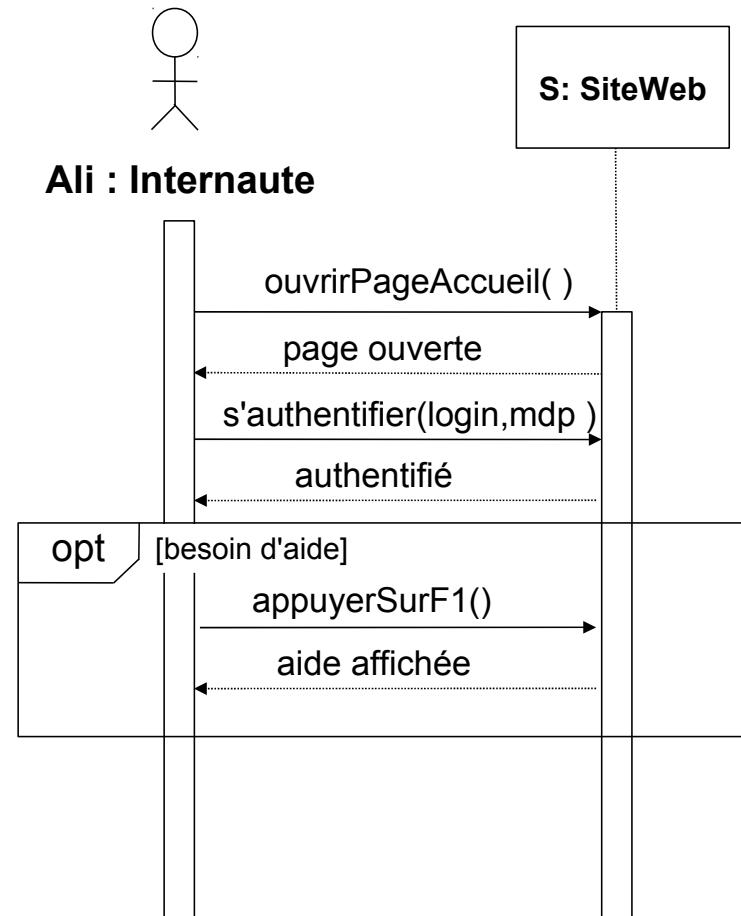
Opérateur « Alternative »

- Alternative (ou *alt*)
 - ✓ Opérateur conditionnel
 - Équivalent d'une exécution à choix multiples (*switch*)
 - ✓ Peut posséder plusieurs opérandes, chacune détient une condition de garde
 - ✓ Absence de condition de garde: condition vraie
 - ✓ Condition *else*: vraie si aucune autre condition n'est vraie



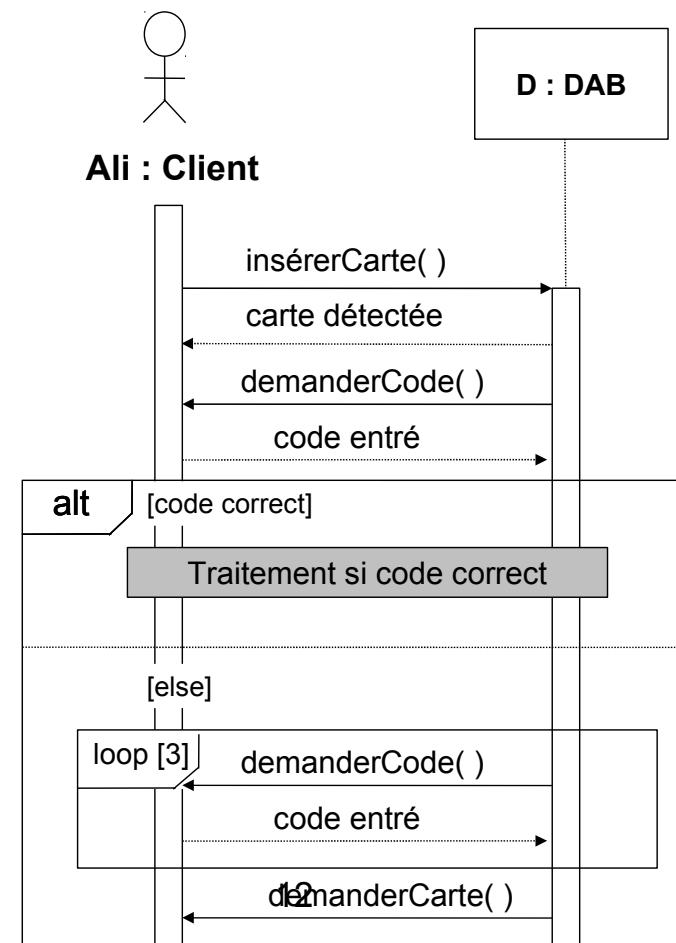
Opérateur « Option »

- Option (ou *opt*)
 - ✓ Représente un comportement qui peut se produire ou pas.
 - ✓ Équivalent à un *alt* à une seule branche et sans *else*



Opérateur « Loop »

- Loop
 - ✓ Équivalent d'une boucle *for*
 - ✓ Décrit des interactions qui s'exécutent en boucle
 - ✓ La condition (garde) indique le nombre de répétitions (min et max) ou une condition booléenne à respecter



Opérateur « Parallèle »

- Parallèle (ou *par*)
 - ✓ A au moins 2 sous-fragments exécutés simultanément
 - ✓ Simule une exécution parallèle

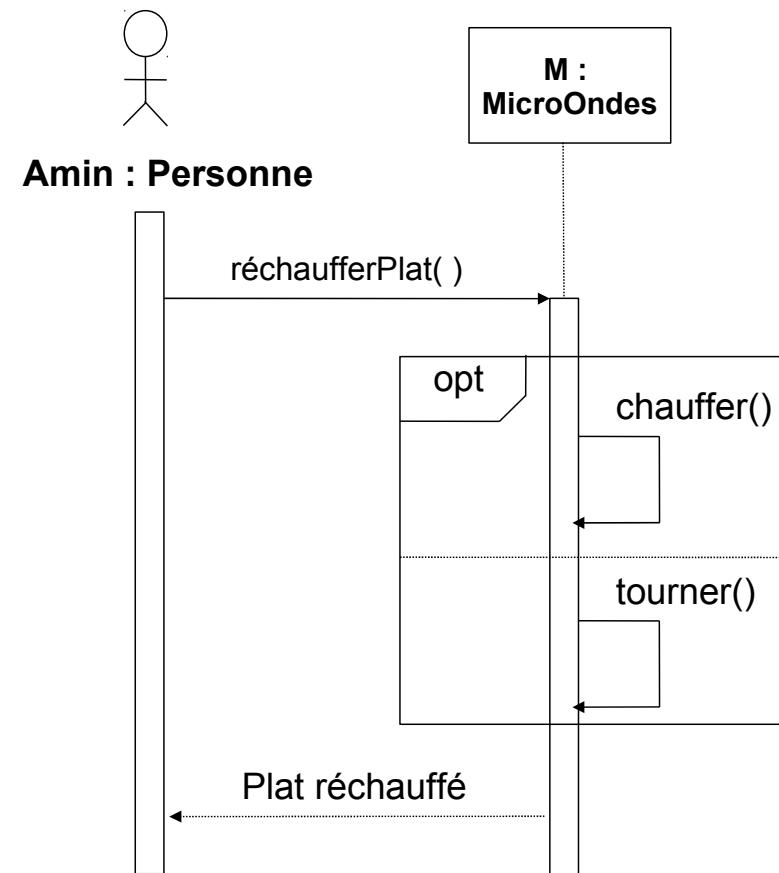
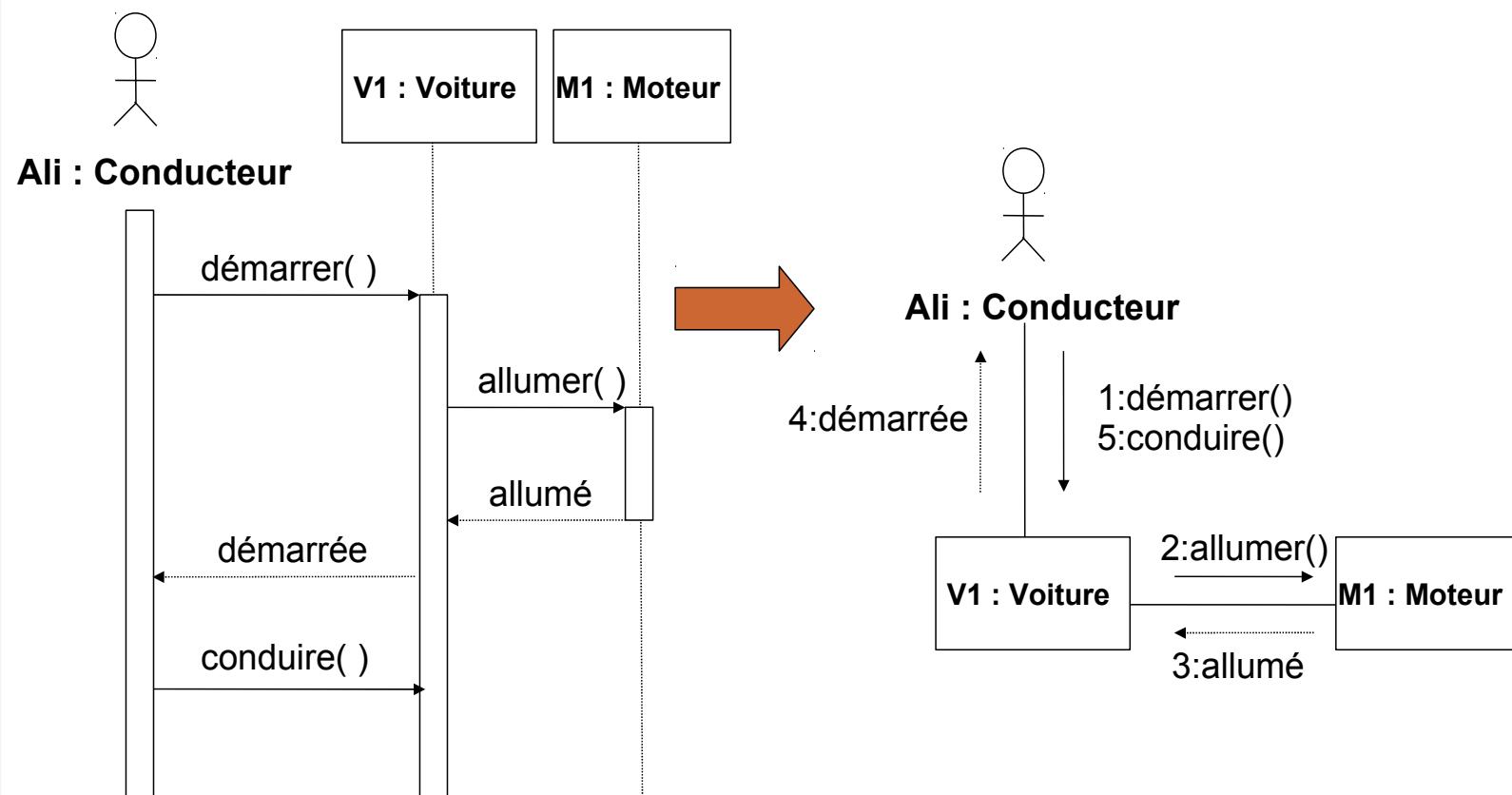


Diagramme de Communication

- Appelé diagramme de *collaboration* avant UML2
- Montre les interactions entre objets
- Insiste sur la structure spatiale pour mettre en collaboration un groupe d'objets
 - ✓ Messages : liens reliant les objets
 - ✓ Temps : représenté implicitement par une numérotation des messages

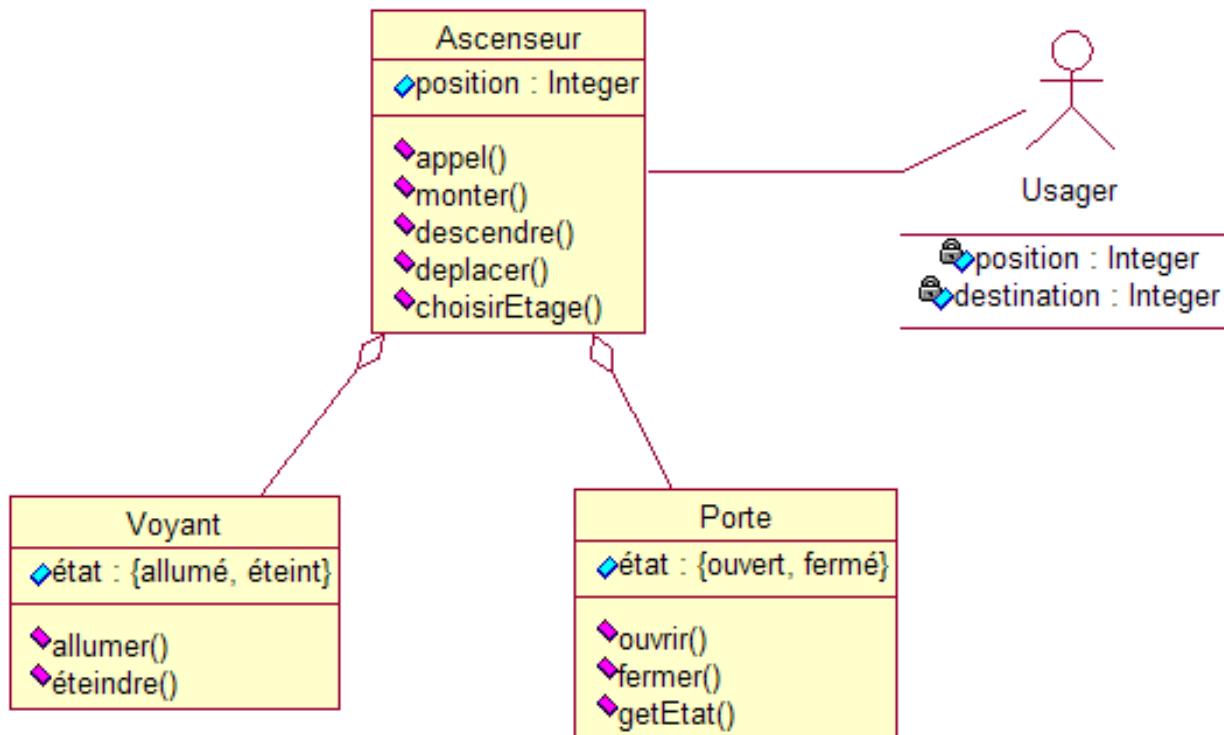
Exemple de Diagramme de Communication

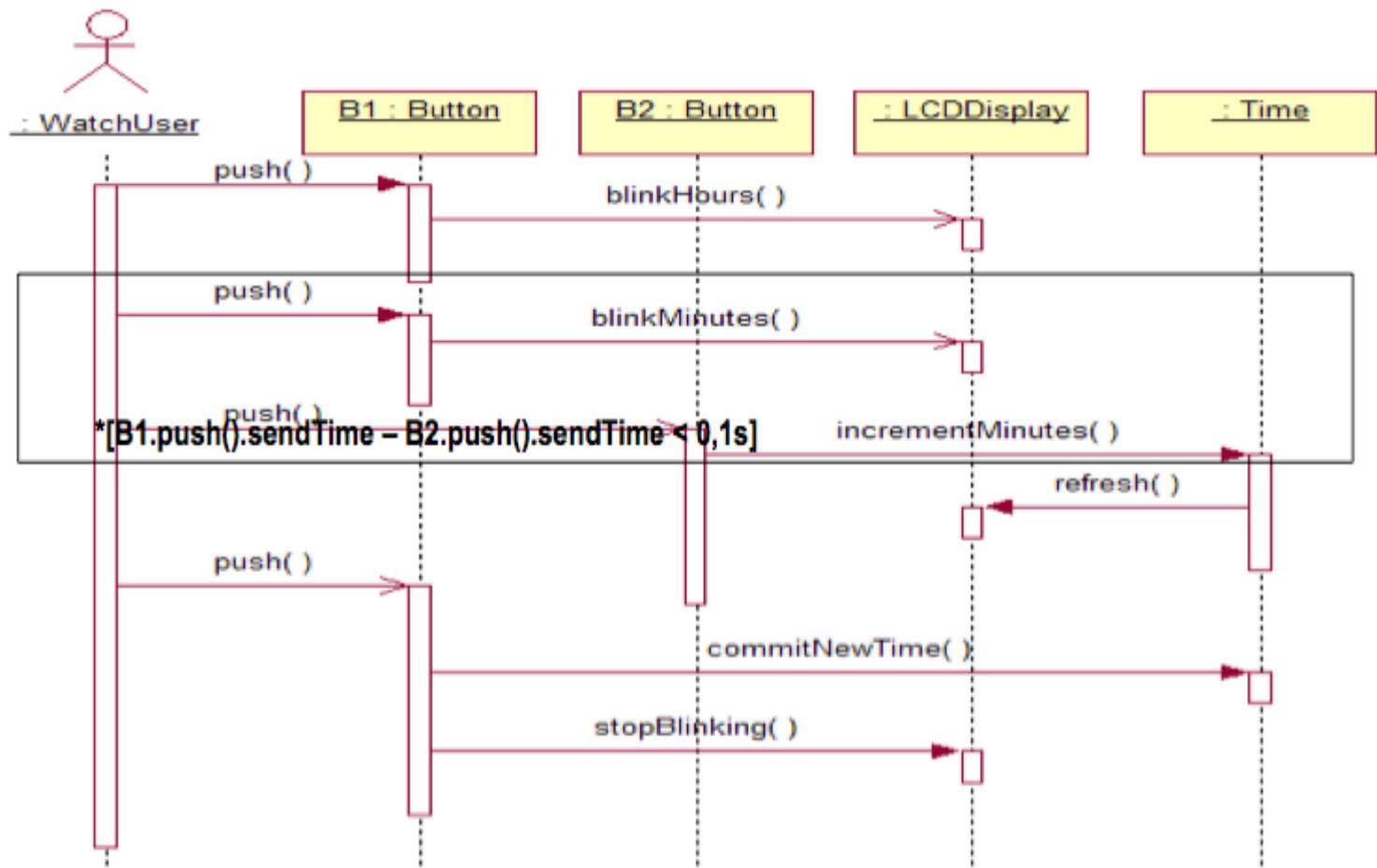


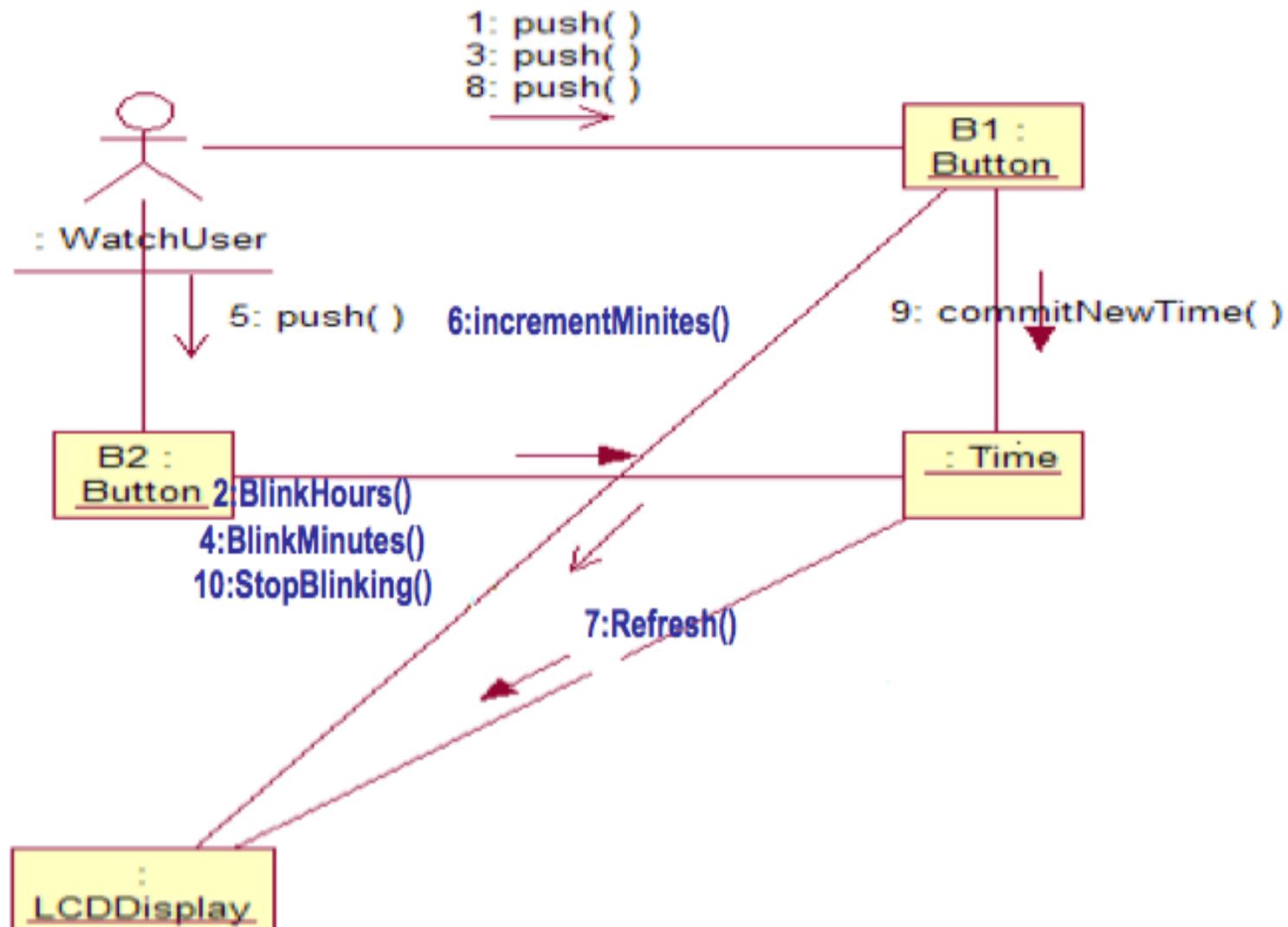
Application: Ascenseur

- D'après le diagramme de classes :

- ✓ Rédiger le diagramme de séquences pour modéliser un scénario où un usager veut monter en utilisant l'ascenseur
- ✓ En déduire le diagramme de communication







Application : Montre

- A partir du diagramme de classes suivant :
 - ✓ Rédiger un diagramme de séquences pour modéliser un scénario où un utilisateur voudrait régler les minutes sur sa montre :
 - En appuyant 2 fois sur le bouton 1, il accède au réglage des minutes (l'heure clignote puis la minute clignote). Ensuite, avec le bouton 2, il incrémente les minutes d'une valeur, le LCD display est rafraîchi après chaque pression. En appuyant sur les deux boutons à la fois, l'heure est enregistrée et l'affichage s'arrête de clignoter
 - ✓ En déduire le diagramme de communication

