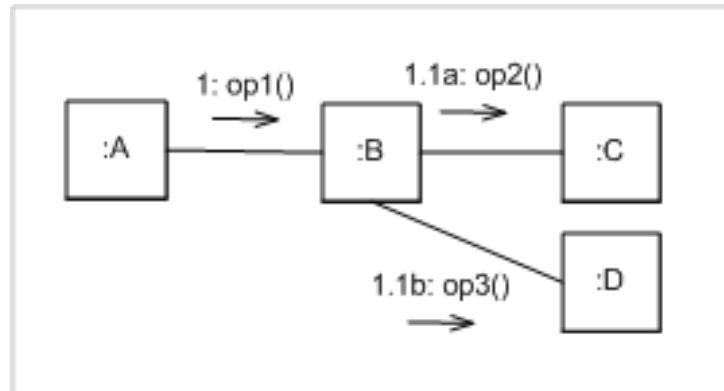


Diagramme de communication

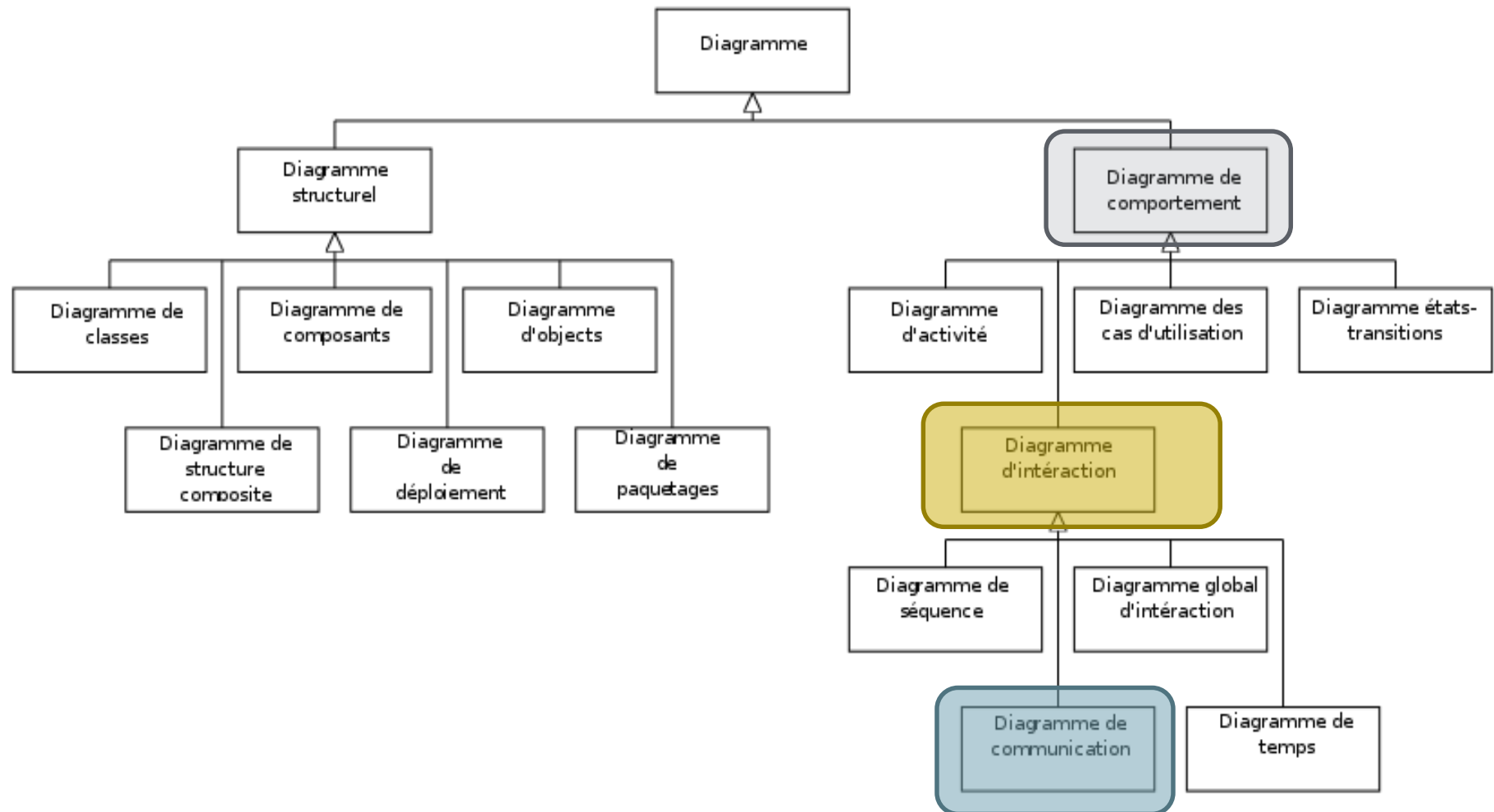
Mme CHALOUAH Anissa

I.S.E.T Bizerte

Département Technologies de l'informatique



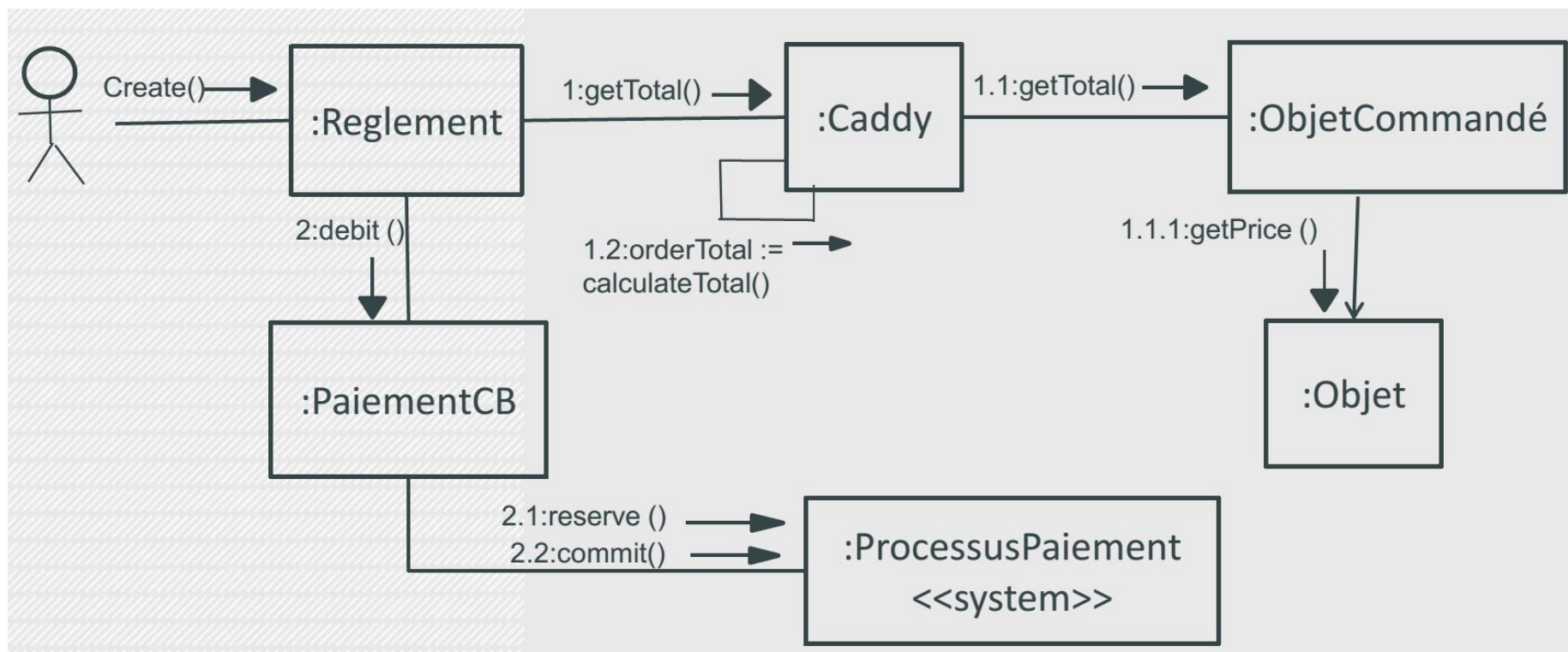
Positionnement



Positionnement

Nommage

- Diagramme de **collaboration** (UML 1.X)
Diagramme de **communication** (UML 2.X)
- Les diagrammes de **communication** et les diagrammes de **séquences** sont deux types de diagramme **d'interaction** représentant le **comportement** du système.

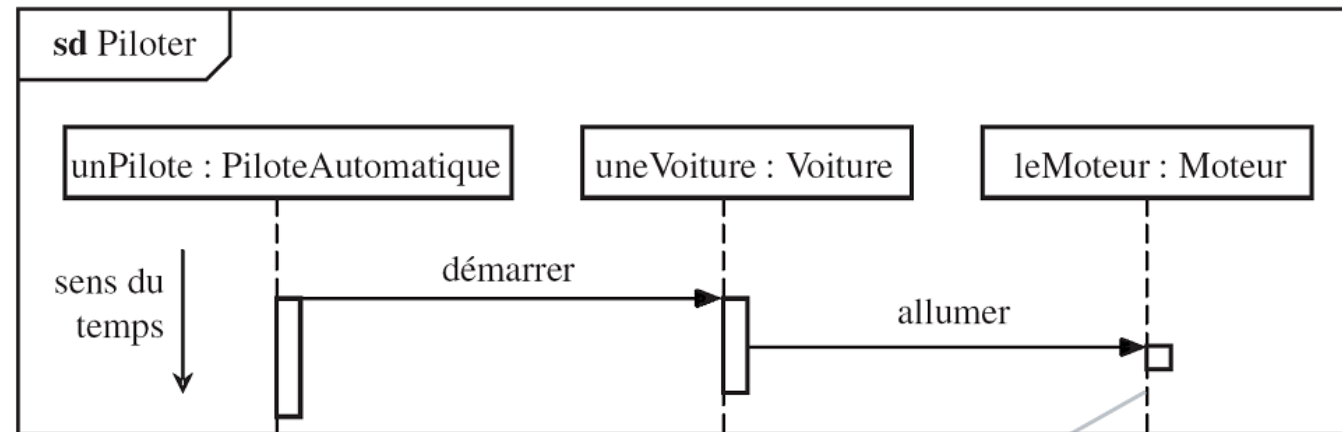
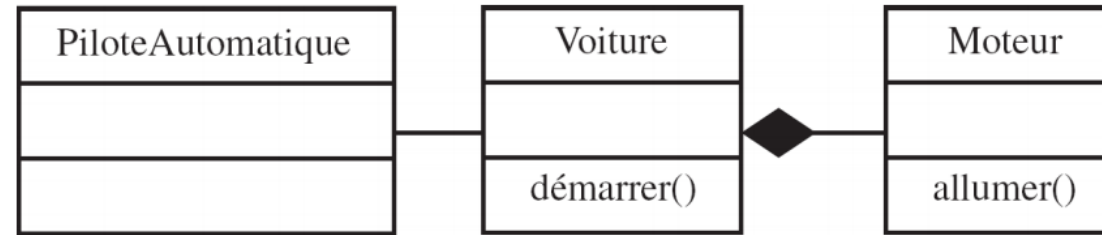


Présentation

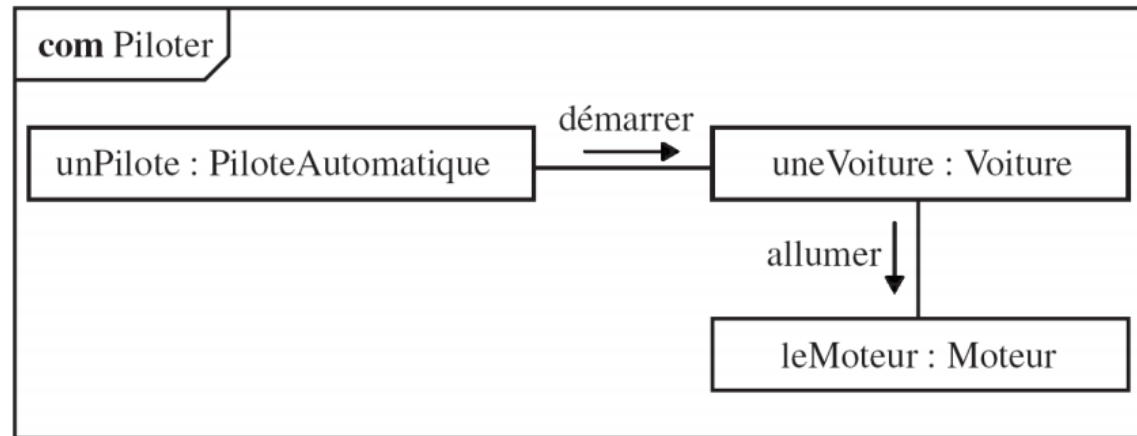
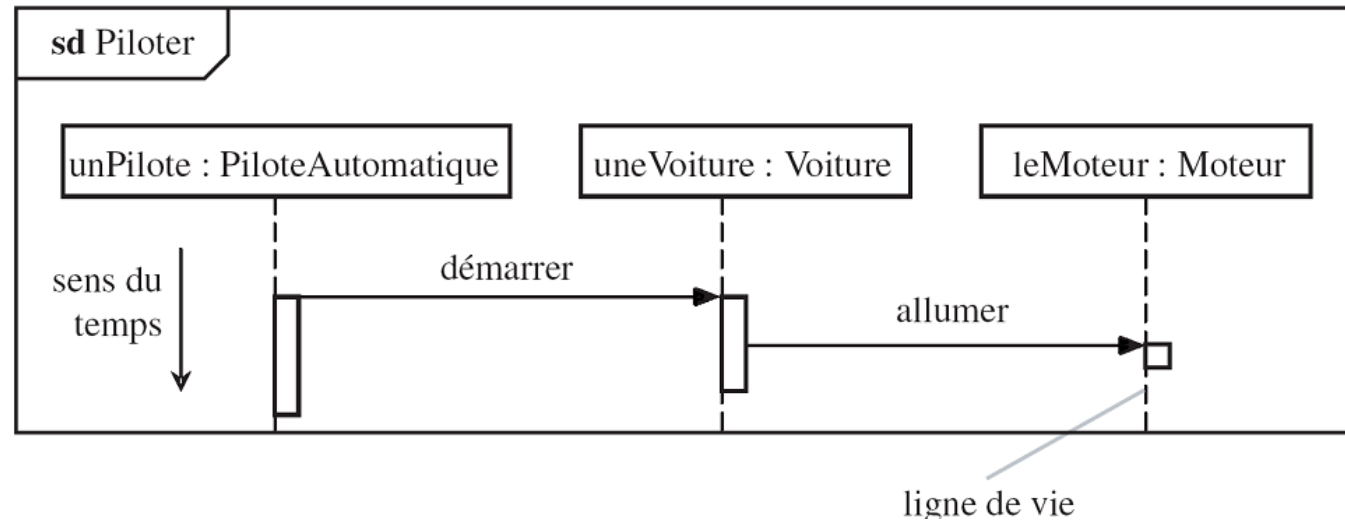
- Un diagramme de **séquence** montre des interactions sous un angle **temporel**, en mettant l'emphasis sur le séquençement temporel de messages échangés entre des objets,
- Un diagramme de **communication** montre une représentation **spatiale** des objets.

Ils représentent la même chose, mais sous des formes différentes.

Exemple



Exemple




- 
- Un diagramme de communication est associé à un cas d'utilisation.
 - A un cas d'utilisation on peut associer plusieurs diagrammes de communication. Chaque diagramme de communication correspondant à une exécution particulière.

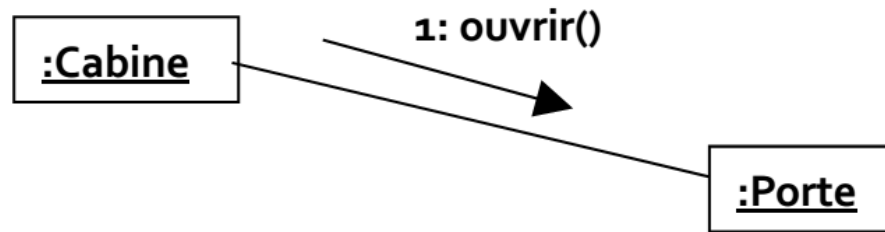


Diagramme de communication : Concepts

Trois concepts fondamentaux :

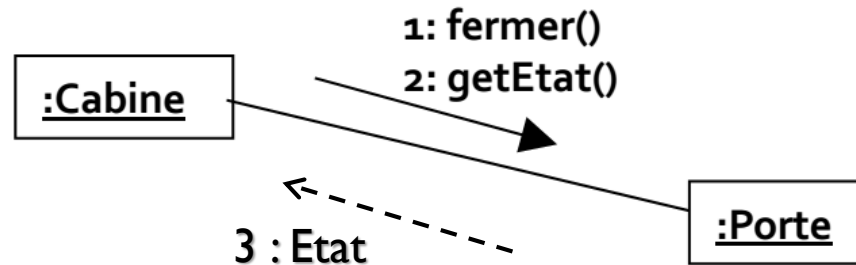
- Les Objets
- Les liens d'interactions
- Les Messages

Notion de lien(1)



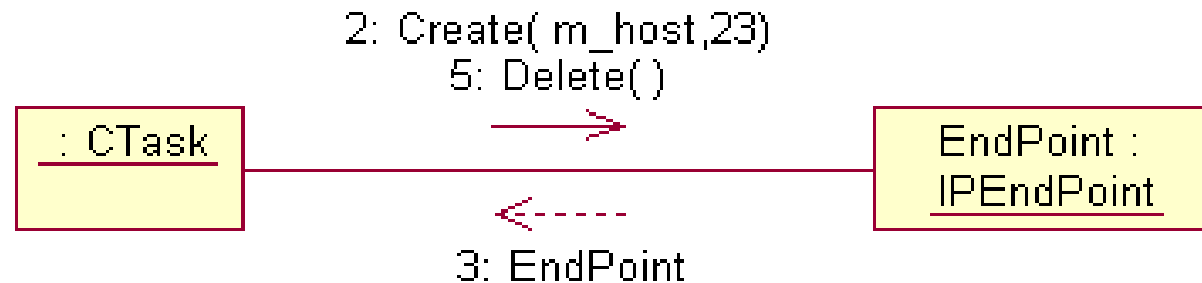
- Un lien est une connexion entre deux objets, qui indique qu'une forme de navigation et de visibilité entre eux est possible.
- Autrement dit, un lien permet d'acheminer des messages dans un sens ou dans l'autre.

Notion de lien(2)



- Plusieurs messages, et ce dans les deux sens, peuvent circuler sur le même lien.
- Il n'y a pas un lien par message : tous les messages empruntent le même lien.

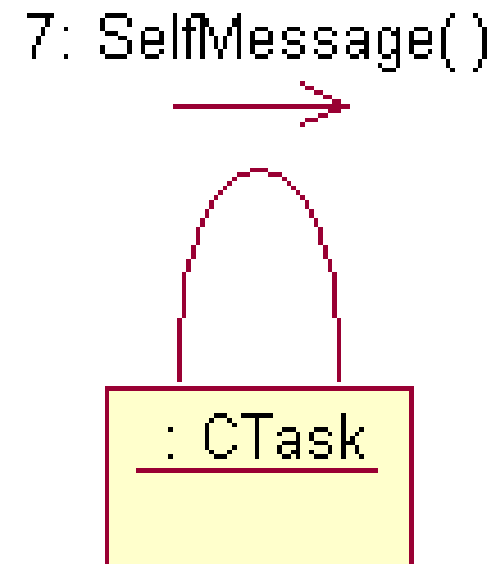
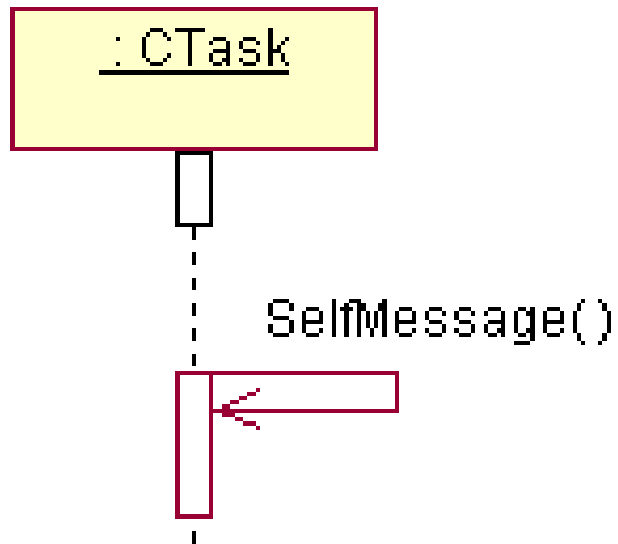
Message



- Chaque message entre objets est représenté par une **expression**, une **flèche** indiquant sa direction, et un **numéro** indiquant sa place dans la séquence.

Message réflexif

- Exemple d'un message d'un objet envoyé à lui-même



Synchronisation des messages

- UML permet de spécifier de manière très précise l'ordre et les conditions d'envoi des messages sur un diagramme dynamique.
- Pour chaque message, il est possible d'indiquer :
 - les clauses qui conditionnent son envoi,
 - son rang (son numéro d'ordre par rapport aux autres messages),
 - sa récurrence,
 - ses arguments.

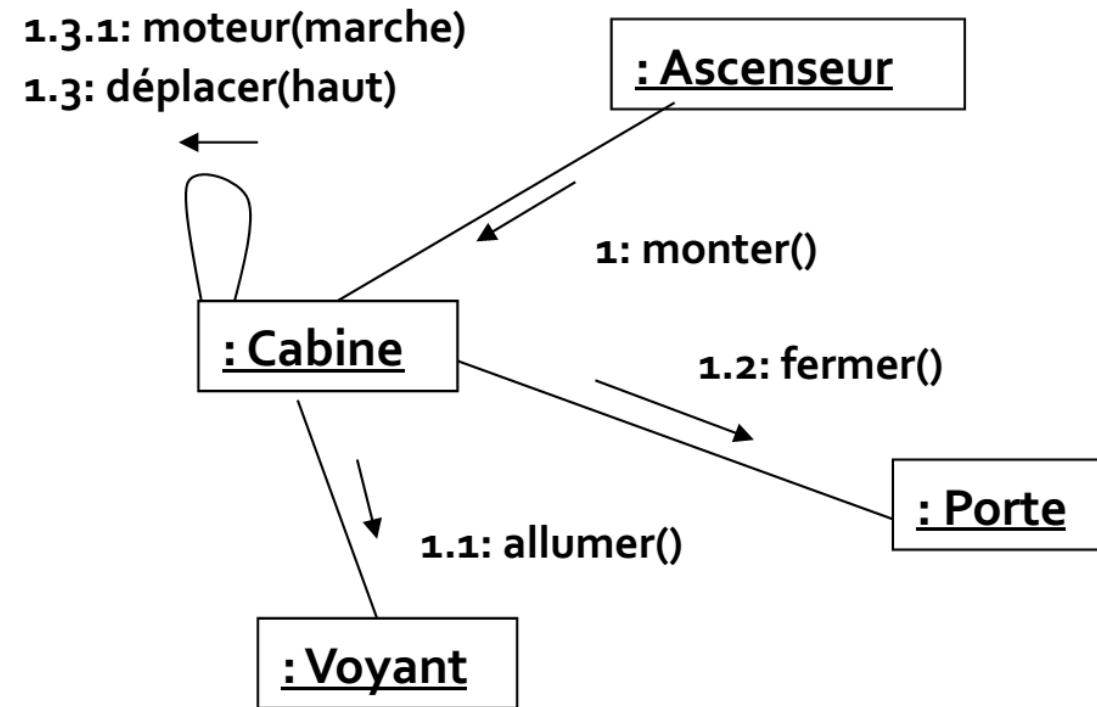
Numérotation des messages

- Indique le **rang** du message, c'est-à-dire son **numéro d'ordre** par rapport aux autres messages.
- Les messages sont numérotés à la façon de chapitres dans un document, à l'aide de chiffres séparés par des points.
- Ainsi, il est possible de représenter le niveau d'emboîtement des messages et leur précedence.
 - Exemple : l'envoi du message 1.3.5 suit immédiatement celui du message 1.3.4 et ces deux messages font partie du flot (de la famille de messages) 1.3.

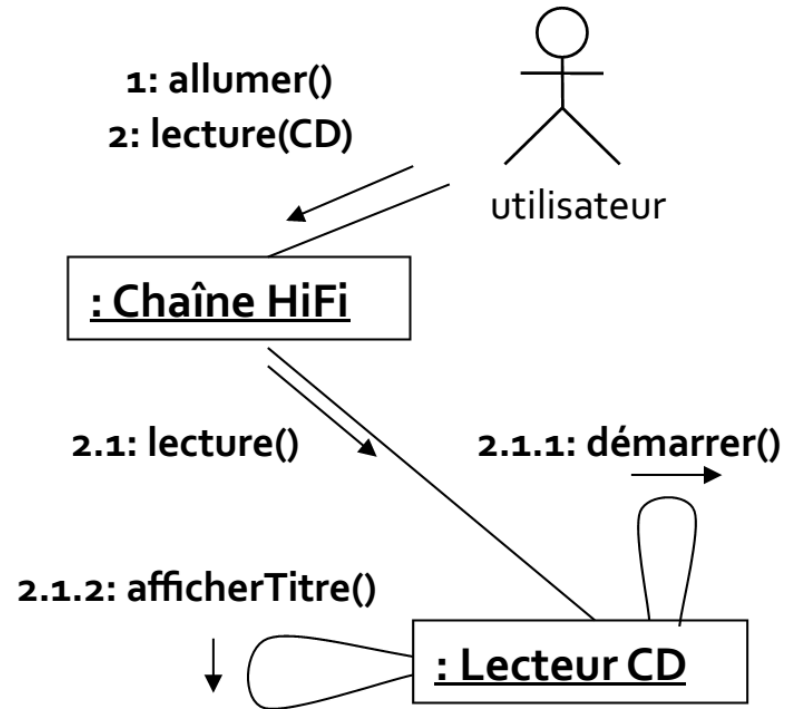
Messages simultanés

- Pour représenter l'envoi **simultané** (en **parallèle**) de deux messages, il suffit de les indexer par une **lettre**.
 - Exemple : l'envoi des messages 1.3.a et 1.3.b est simultané.

Exemple 1 :



Exemple 2 :



syntaxe d'un message (1)

pre/ [cond] seq * || [iter] r:= msg(par)

- **Pré/** : **prédécesseurs** (liste de numéros de séquence de messages séparés par une virgule). Indique que le message courant ne sera envoyé que lorsque tous ses prédécesseurs le seront aussi (permet de synchroniser l'envoi de messages).
- **cond** : **garde, expression booléenne**. Permet de conditionner l'envoi du message.
- **seq** : **numéro de séquence du message**.

syntaxe d'un message (2)

pre/ [cond] seq * || [iter] r:= msg(par)

- **Iter** : récurrence du message.
Permet de spécifier l'envoi séquentiel (ou en parallèle, avec "||") de messages. Il est possible de spécifier qu'un message est récurrent en omettant la clause d'itération (en n'utilisant que "*" ou "*||").
- **r** : valeur de retour du message.
Permet d'affecter la valeur de retour d'un message, pour par exemple la retransmettre dans un autre message, en tant que paramètre.
- **msg** : nom du message.
- **par** : paramètres du message

Exemples de messages (1)

- 3 : bonjour()

Ce message a pour numéro de séquence "3".

- [heure = midi] 1 : manger()

Ce message n'est envoyé que s'il est midi.

- 1.3.6 * : ouvrir()

Ce message est envoyé de manière séquentielle un certain nombre de fois.

- 3 / * || [i := 1..5] : fermer()

Représente l'envoi en parallèle de 5 messages. Ces messages ne seront envoyés qu'après l'envoi du message 3.

Exemples de messages (2)

- 1.3,2.1 / [t < 10s] 2.5 : age := demanderAge(nom,prenom)

Ce message (numéro 2.5) ne sera envoyé qu'après les messages 1.3 et 2.1, et que si "t < 10s".

- 1.3 / [disk full] 1.7.a * : deleteTempFiles()
1.3 / [disk full] 1.7.b : reduceSwapFile(20%)

Ces messages ne seront envoyés qu'après l'envoi du message 1.3 et si la condition "disk full" est réalisée. Si cela est le cas, les messages 1.7.a et 1.7.b seront envoyés simultanément. Plusieurs messages 1.7.a peuvent être envoyés.



DIAGRAMME DE COMMUNICATION VS DIAGRAMME DE SÉQUENCE

Choix du diagramme

- Chaque type de diagramme a ses points forts : il n'y a pas de choix « correct » dans l'absolu et chaque modélisateur aura ses propres préférences.
- Néanmoins, les outils UML privilégient les **diagrammes de séquence**, en raison de leur grande richesses notationnelle.

Points forts et points faibles : Spécification UML

- La spécification UML est **plus centrée** sur les **diagrammes de séquences** que sur les **diagrammes de communication**.
 - Davantage de réflexion et d'efforts ayant été consacrés à leur notation et leur sémantique.
 - En conséquence, les outils les prennent mieux en charge et les options de notation sont plus nombreuses.

Points forts et points faibles :

Lecture du flot d'appels

- Les **diagrammes de séquence** permettent une meilleure visualisation des flots d'appels, parce qu'on les **lit simplement de haut en bas**.
- Les **diagrammes de séquence** sont excellents pour les besoins de la documentation ou pour suivre facilement le flot d'appels généré automatiquement par la rétro ingénierie.
- Les **diagrammes de communication** ne permettent quant à eux de retrouver la séquence des appels qu'en se **référant à leur numérotation**, comme « 1 », « 2 », ..

Points forts et points faibles :

Mode esquisse

- Les **diagrammes de communication** sont intéressants lorsqu'on applique UML « en mode esquisse » en modélisant au tableau parce qu'ils permettent beaucoup mieux d'exploiter l'espace.
 - Possibilité d'ajouter/supprimer des boîtes n'importe où sur l'axe horizontal et vertical.
 - Important vu la **fréquence des changements** intervenant lors de la conception.
- Au contraire, dans les **diagrammes de séquence**, les nouveaux objets doivent être ajoutés à droite, ce qui impose des limites car le bord droit de la page (ou du tableau) est rapidement épuisé (tandis que de l'espace vertical est disponible).

Points forts et points faibles :

Synthèse

Type	Forces 😊	Faiblesses ☹️
Séquence	Indique clairement la séquence et l'ordonnancement des messages. Grande richesse de la notation.	Ajout de nouveaux objets s'effectuant obligatoirement vers la droite; consomme trop d'espace horizontal.
Communication	Économique en terme d'espace, permet d'ajouter des objets dans les deux dimensions.	Rend plus difficile la lecture des séquences de messages. Moins d'options de notation.

Bibliographie

- Introduction à UML 2
Modélisation Orientée Objet de Systèmes Logiciels
- Pierre Gérard -
- <http://uml.free.fr/cours>