

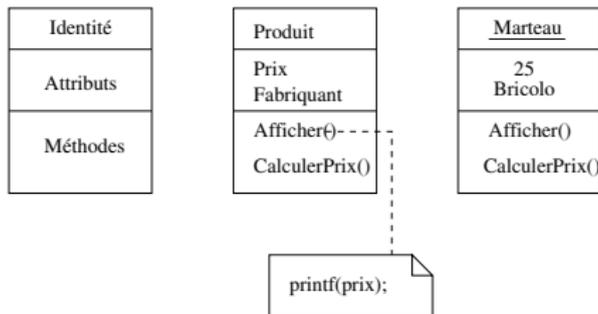
UML - Un outil de restitution de modèles

- Langage graphique, en rien une méthode..
- Constitué d'une série de diagrammes pour représenter différents aspects du programme : diagramme de classes, d'interactions, d'états ...
- Connaitre et respecter les conventions graphiques est indispensable.
- Certains outils peuvent générer automatiquement des diagrammes à partir du code ou inversement.
- ArgoUML permet de créer des diagrammes et génère le code dans différents langages objet.

Objets et Classe d'objets

Objet = Etat + Comportement + Identité

- Conventions graphiques de UML

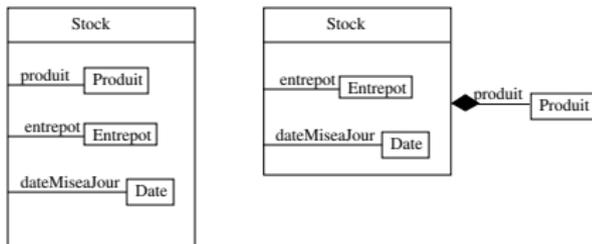


Les classes abstraites sont représentées uniquement par leur nom dans un rectangle. Il est possible de faire apparaître le mot `abstract` sous ce nom.

La composition d'objets

Les attributs d'un objet peuvent être eux-même des objets.

- **La composition par valeur** : la construction d'un objet physique implique la construction de ses attributs par valeur.



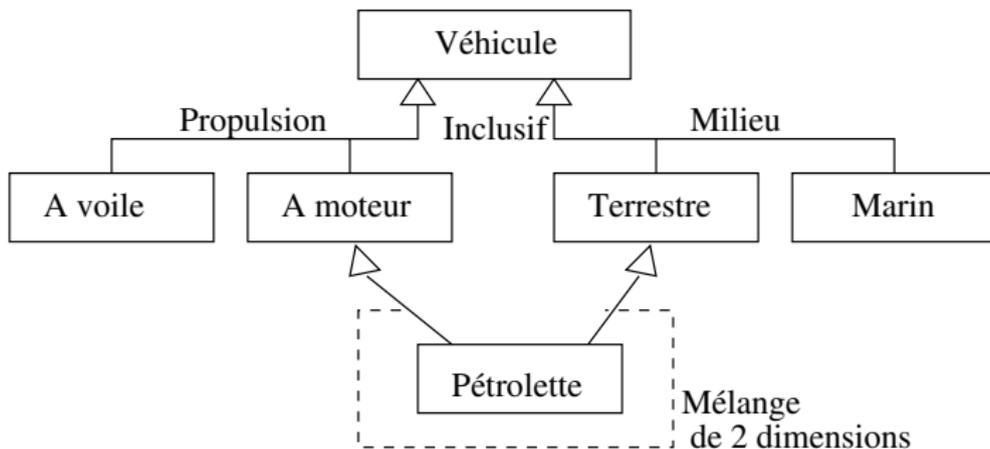
- **La composition par référence** : c'est un lien de référence qui peut être partagé par plusieurs objets.
- La construction du conteneur n'implique pas la construction de l'objet référencé.



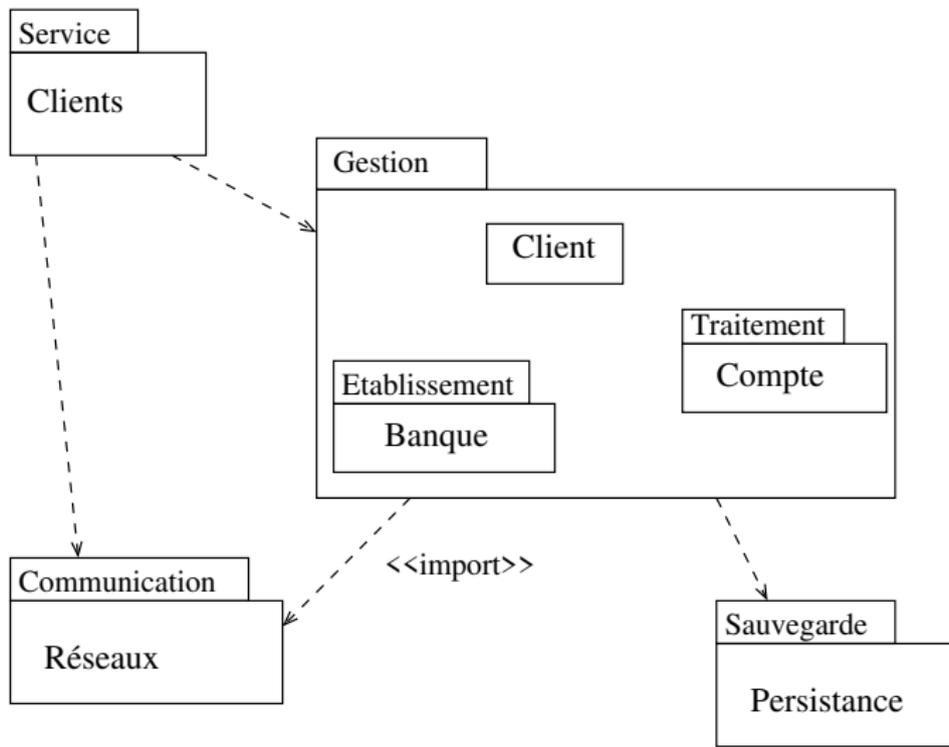
La visibilité des attributs et des méthodes

- **Publique** : un attribut ou une méthode publique est spécifiée avec le signe +.
- **Privée** : un attribut ou une méthode privée est spécifiée avec le signe -.
- **Protected** : un attribut ou une méthode protégée est spécifiée avec le signe #.

Héritage

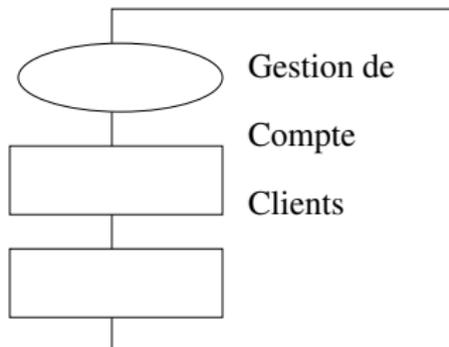


Représentation des Paquetages



Représentation de modules

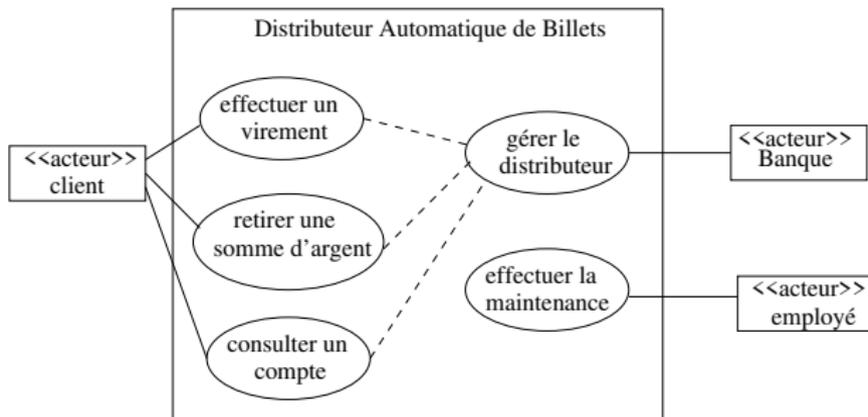
Les modules sont des unités de compilation. Certains langages de programmation n'ont aucune correspondance avec ce concept.



Cas d'utilisation¹

Problème :

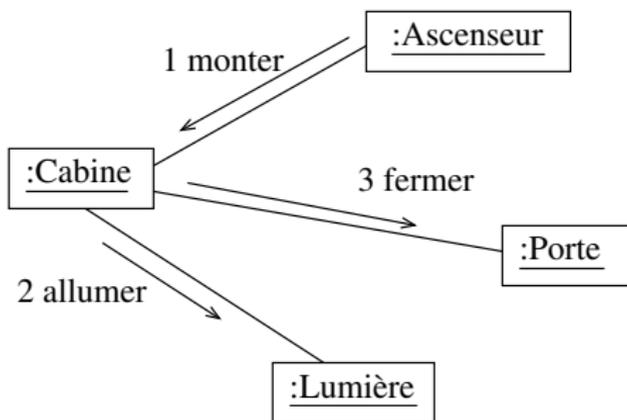
- Les besoins d'un système (cf cahier des charges) sont souvent exprimés de manière non structurée, sans forte cohérence (imprécision, oublis, contradictions).



¹Ont été formalisés par I. Jacobson *Object-Oriented Software Engineering, A Use Case Driven Approach*, Addison-Wesley, 1992.

Diagramme de collaboration

- Le diagramme de collaboration montre simultanément les interactions entre les objets et les relations structurelles qui permettent ces interactions.
- La numérotation donne l'ordre d'envoi des messages.
- Le temps n'est pas représenté.

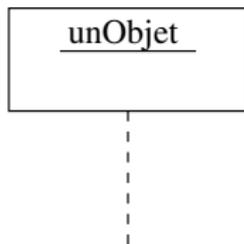


Les diagrammes de collaboration

- Rôle :
 - Ils montrent les interactions entre les objets.
 - Ils expriment le contexte d'un groupe d'objets.
 - Ils sont des extensions du diagrammes d'objets.
- Une interaction est réalisée par un groupe d'objets qui collaborent en échangeant des messages.
- Ces messages sont représentés le long des liens qui relient les objets avec des flêches orientées vers le destinataire du message.

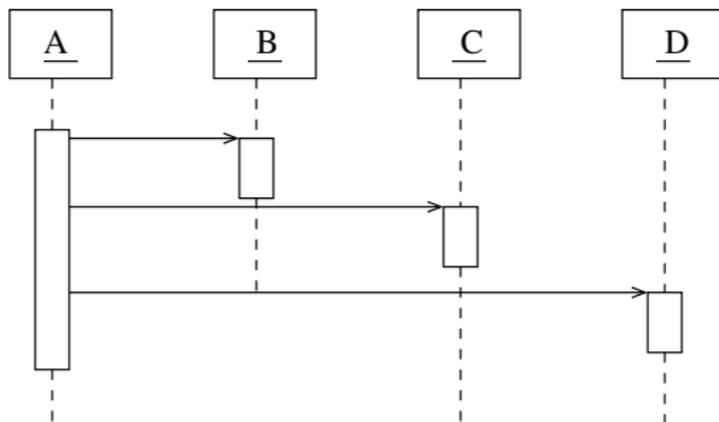
Diagramme de séquence

- Montrent des interactions entre objets selon un point de vue temporel.
- Pas de représentation explicite du contexte des objets.
- Notation² :
 - Un objet est matérialisé par un rectangle est une barre verticale appelée ligne de vie



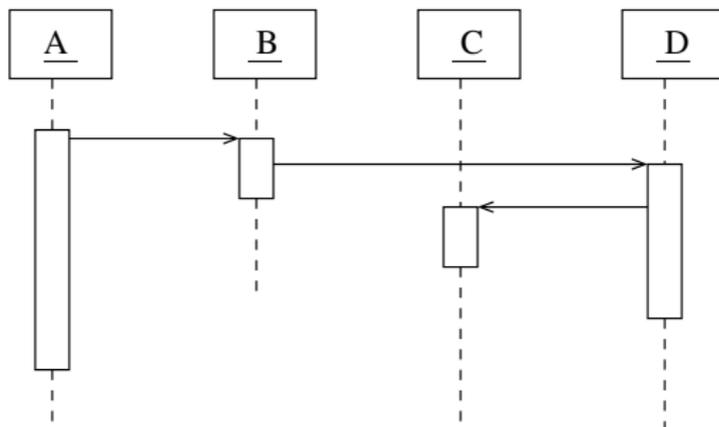
Mode centralisé - Mode décentralisé

- Les diagrammes de séquences reflètent le choix des structures de contrôle.

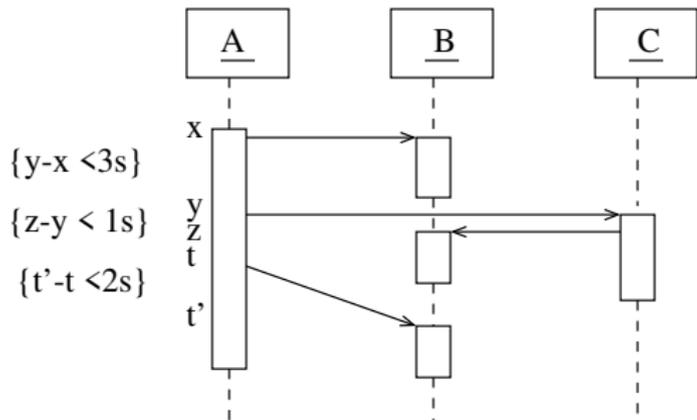


Mode centralisé - Mode décentralisé

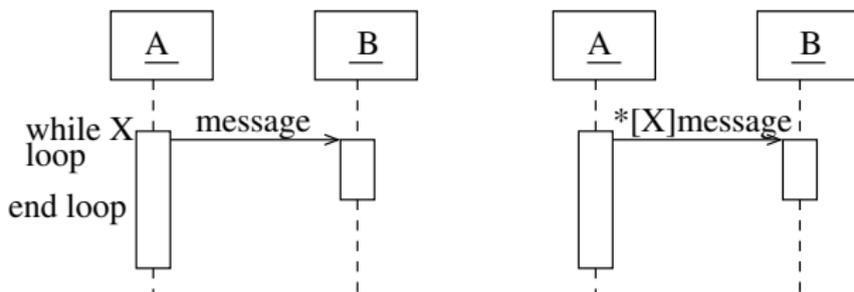
- Envoi décentralisé de messages.



Expression des contraintes temporelles

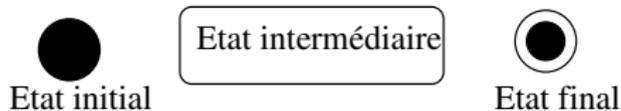


Boucles et branchements



Les diagrammes d'état

- Un objet est à tout moment dans un état donné.
- L'état d'un objet est constitué des valeurs instantanées de ses attributs.



Les événements

- La syntaxe d'un événement dans un diagramme est la suivante :
 - `Nom_événement (Nom_paramètre : type, ...)[condition]`
condition est la garde qui valide ou non le déclenchement d'une transition quand l'événement s'est produit.
- On peut associer à chaque transition une **action** à exécuter lors du franchissement dû à un événement. Les spécifications de l'action sont contenues dans l'objet destinataire.
- **Exemple** : l'événement `il fait trop chaud` entraîne la climatisation ou l'ouverture des fenêtres selon la saison.

