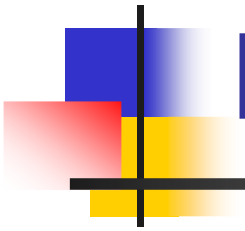
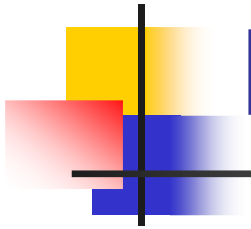


Le langage UML : Les diagrammes de séquence

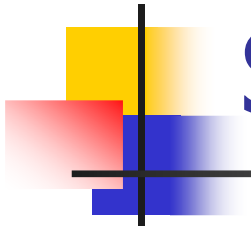
Lydie du Bousquet
Lydie.du-bousquet@imag.fr



Modélisation des interactions

- Les objets d'un système ont un comportement
- Ils interagissent entre eux
- Dynamique générale du système

- 2 diagrammes UML pour représenter les interactions entre les objets
  - Diagramme de séquence
  - Diagramme de communication



Séquence vs communication

- Diagramme de séquences
  - Focalisé sur les aspects temporels
- Diagramme de communication
  - Focalisé sur la représentation spatiale
  - (Diagramme de collaboration en UML 1)

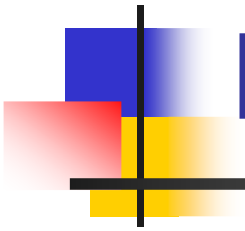
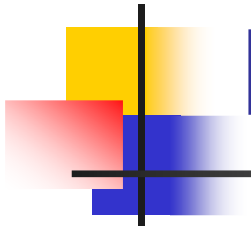


Diagramme de séquence

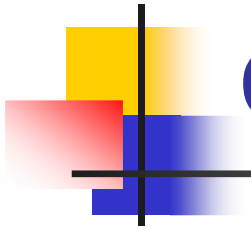
Lire un livre UML pour une explication
détaillée des diagrammes de communication



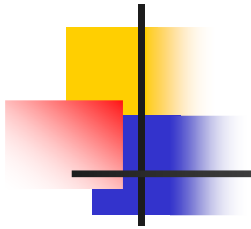
Plan

1. Définition
2. Utilisation
3. Objets
4. Ligne de vie
5. Messages
6. Structures de contrôle

1. Diagramme de séquence – définition

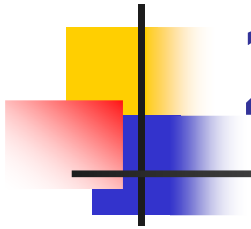


- Décrit la dynamique du système
- Plusieurs diagrammes nécessaire pour capturer « toute » la dynamique d'un système
  - Un / plusieurs diagramme(s) lié(s) à chaque sous-fonction
- Diagramme de séquence
  - Montre de façon séquentielle
  - Les envois de messages entre objets
  - Éventuellement les flux de données



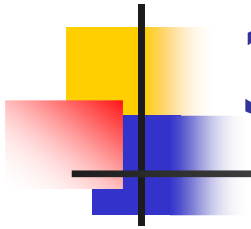
1. Définition: interactions

- Les objets s'envoient des messages entre eux pour **interagir**
- Une interaction se traduit par un envoi de message entre objets
- Envoi de message = appel de méthode
- A la réception d'un message,
  - Un objet devient actif
  - Exécute la méthode du même nom



2. Utilisation

- Documentation des cas d'utilisation :
  - description des interactions en des termes proches de l'usager,
  - les étiquettes des messages correspondent à des événements se produisant dans le système
- Représentation des interactions "informatiques" et répartition des flots de contrôle :
  - le concept de message unifie les formes de
  - communication entre objets (appel de procédure,
  - événement discret, signal, ...)

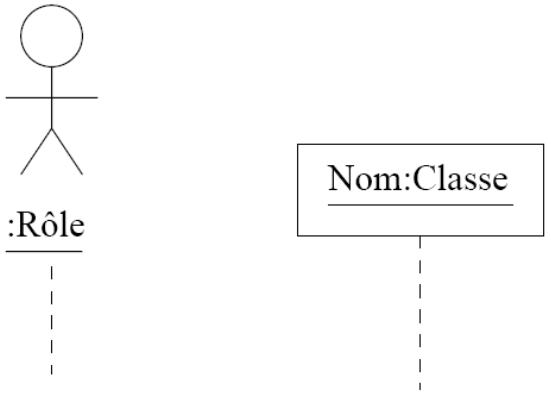


3. Objets

- Entités **appartenant au système** ou se trouvant à ses **limites**
- Ils représentent :
  - acteurs
  - concepts abstraits (cas d'utilisation)
  - soit des objets d'implantation (interactions "informatiques")
- Ils sont identifiés par l'intermédiaire
  - des cas d'utilisation
  - ou des diagrammes de classe.

3. Objets : représentation

- En UML, les objets sont représentés comme suit :



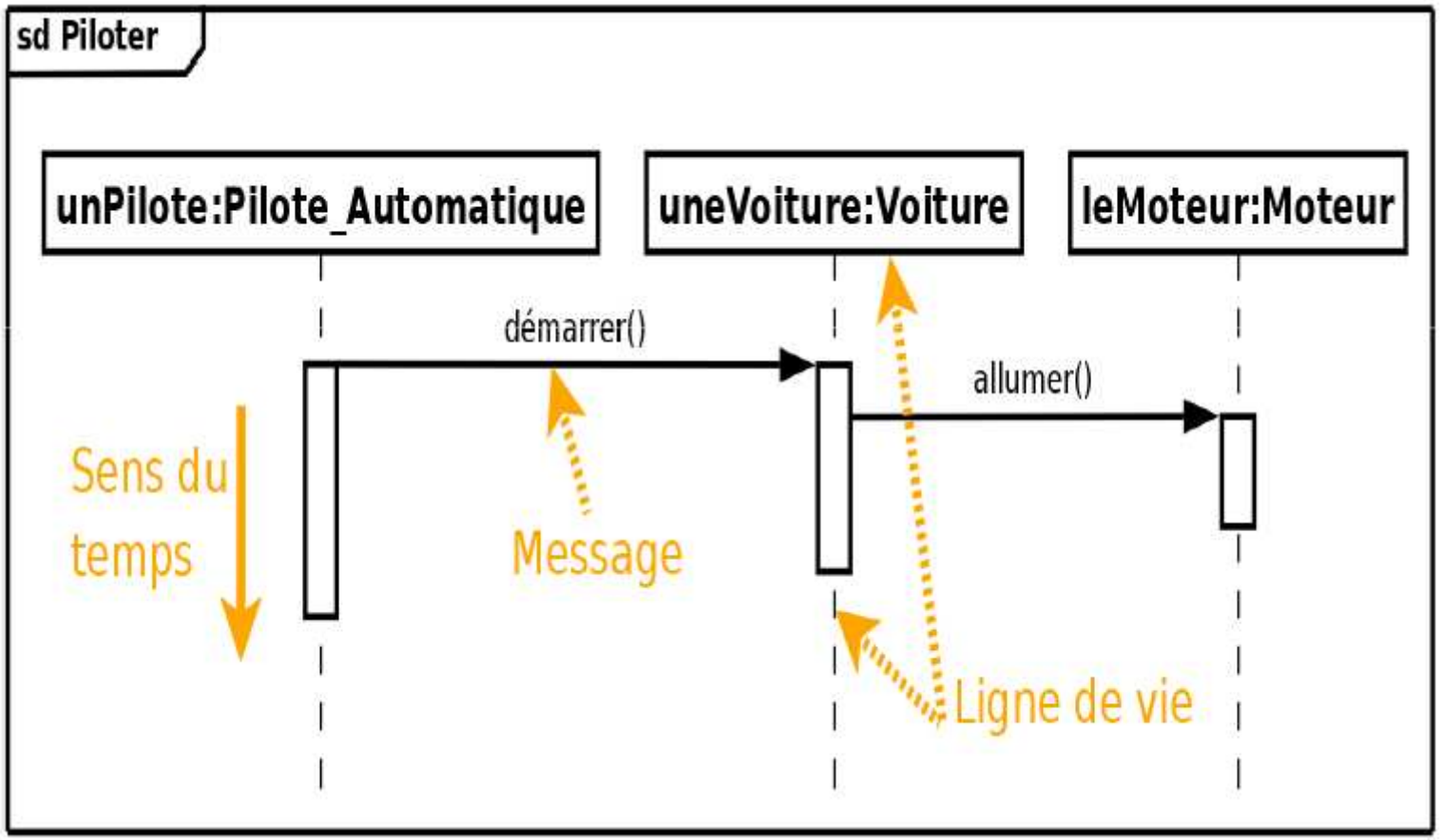
- Le nom de l'objet est composé de son rôle (ou nom)
- et/ou du nom de la classe instanciée (classe)
- Le nom est souligné pour indiquer qu'il s'agit d'une instance.

4. Ligne de vie d'un objet

- Le diagramme de séquence fait apparaître les **instances** des classes
- A chaque instance est associée une **ligne de vie**
  - Ligne verticale en dessous des objets qui montre
  - Période de temps durant laquelle l'objet "existe"

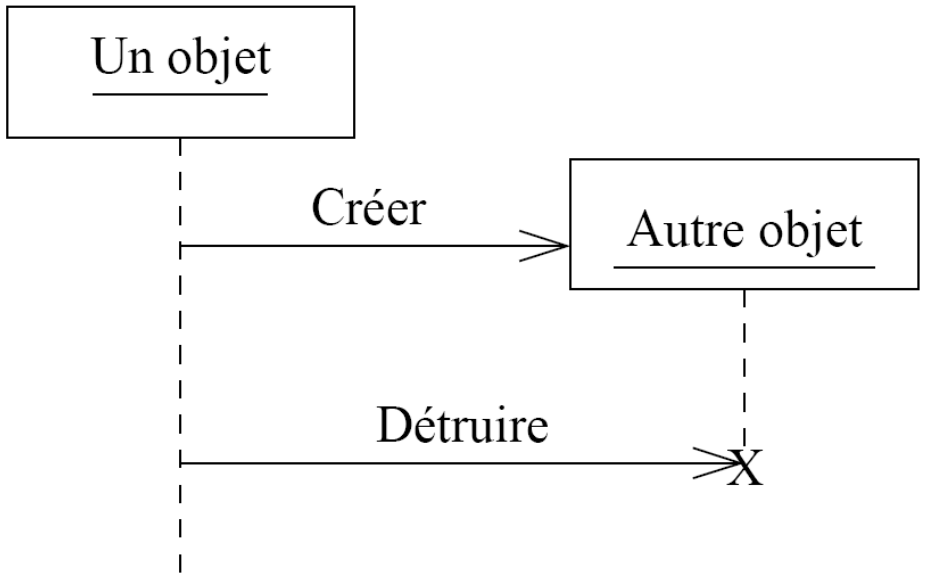
  - Les actions et réactions de l'objet
  - Ses périodes actives / inactives
  - (éventuellement) sa création et/ ou sa destruction

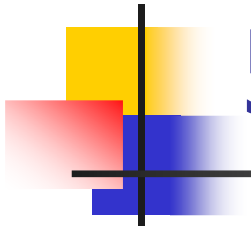
4. Ligne de vie



4. Ligne de vie

- **Création d'un objet**
  un message pointe sur le symbole de l'objet.
- **Destruction d'un objet**
  sa ligne de vie se termine par une croix en trait épais (x).



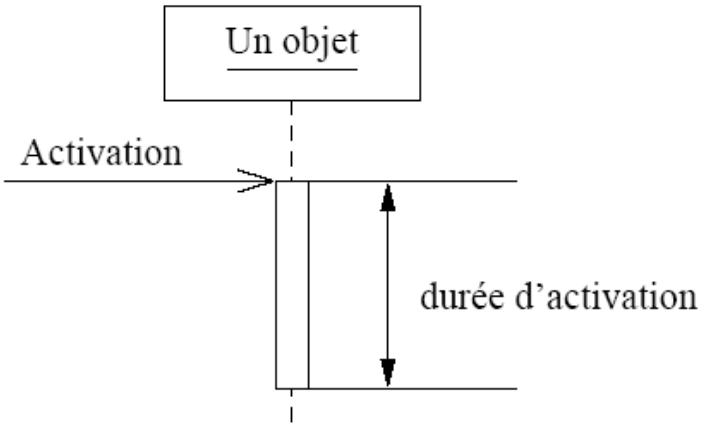


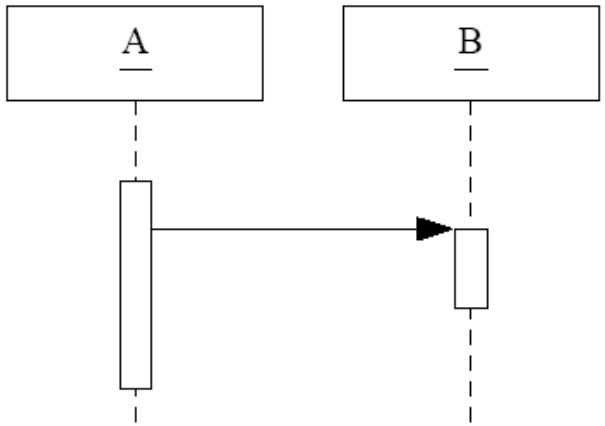
5. Messages

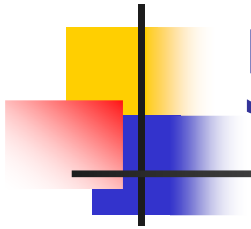
- Les objets communiquent en
  - échangeant des messages
  - représentés sous forme de flèches
- L'ordonnancement **horizontal** des messages n'a **aucune** signification
- Les messages sont étiquetés
  - par le nom de l'opération ou
  - du signal invoqué.

5. Message et activation

- Une période d'activité correspond au temps pendant lequel
- Un objet effectue une action directe ou indirecte
- Représentation : bande verticale le long de la ligne de vie de l'objet





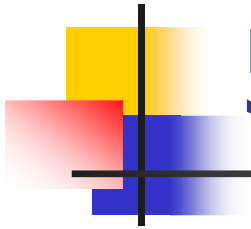


5. Différent types de message

- **message simple**
  Message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière

- **message minuté (timeout)**
  Bloque l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur
  L'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié

- **message synchrone**
  Bloque l'expéditeur jusqu'à prise en compte du message par le destinataire
  Le flot de contrôle passe de l'émetteur au récepteur (l'émetteur devient passif et le récepteur actif) à la prise en compte du message

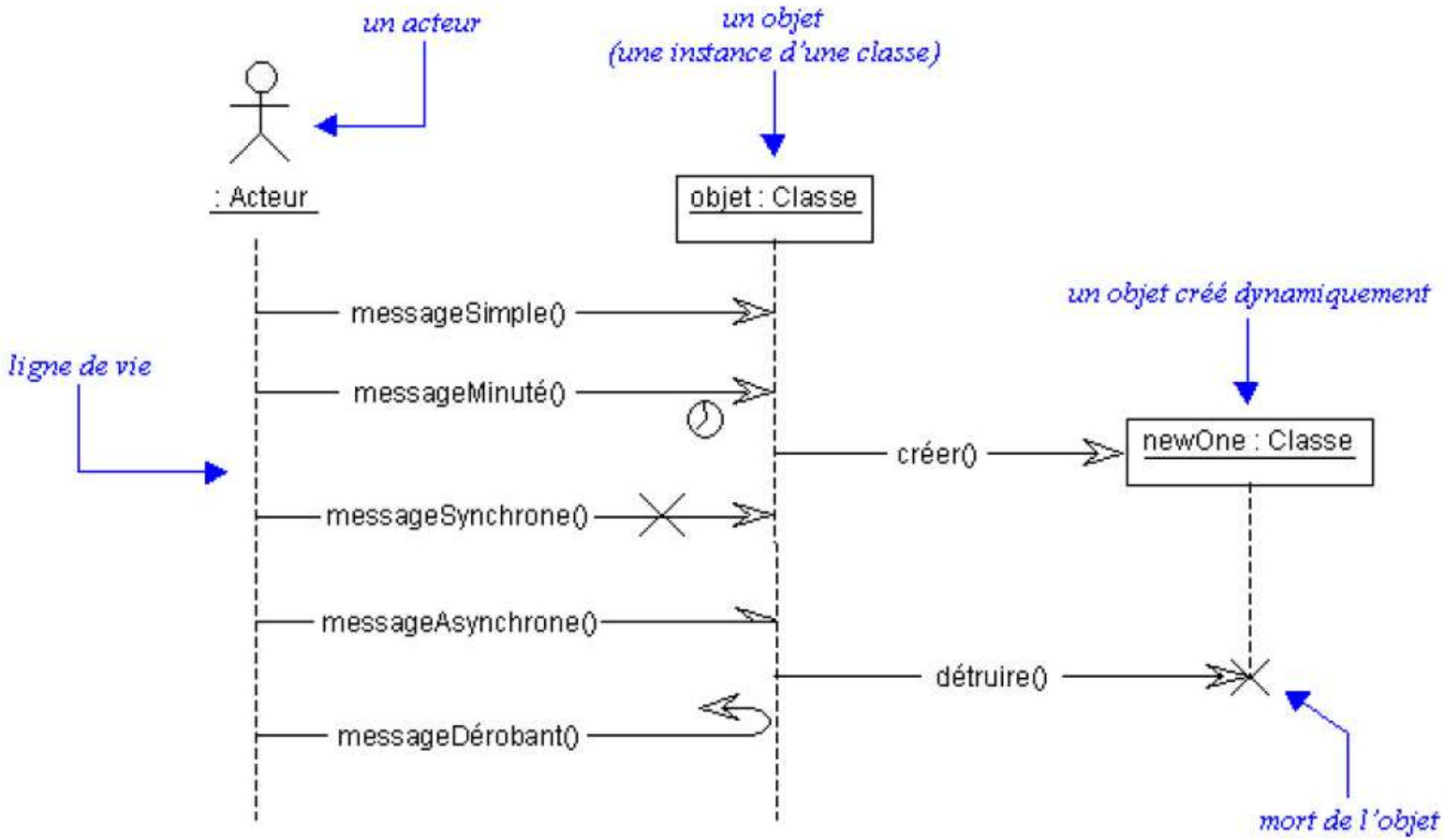


5. Différent types de message

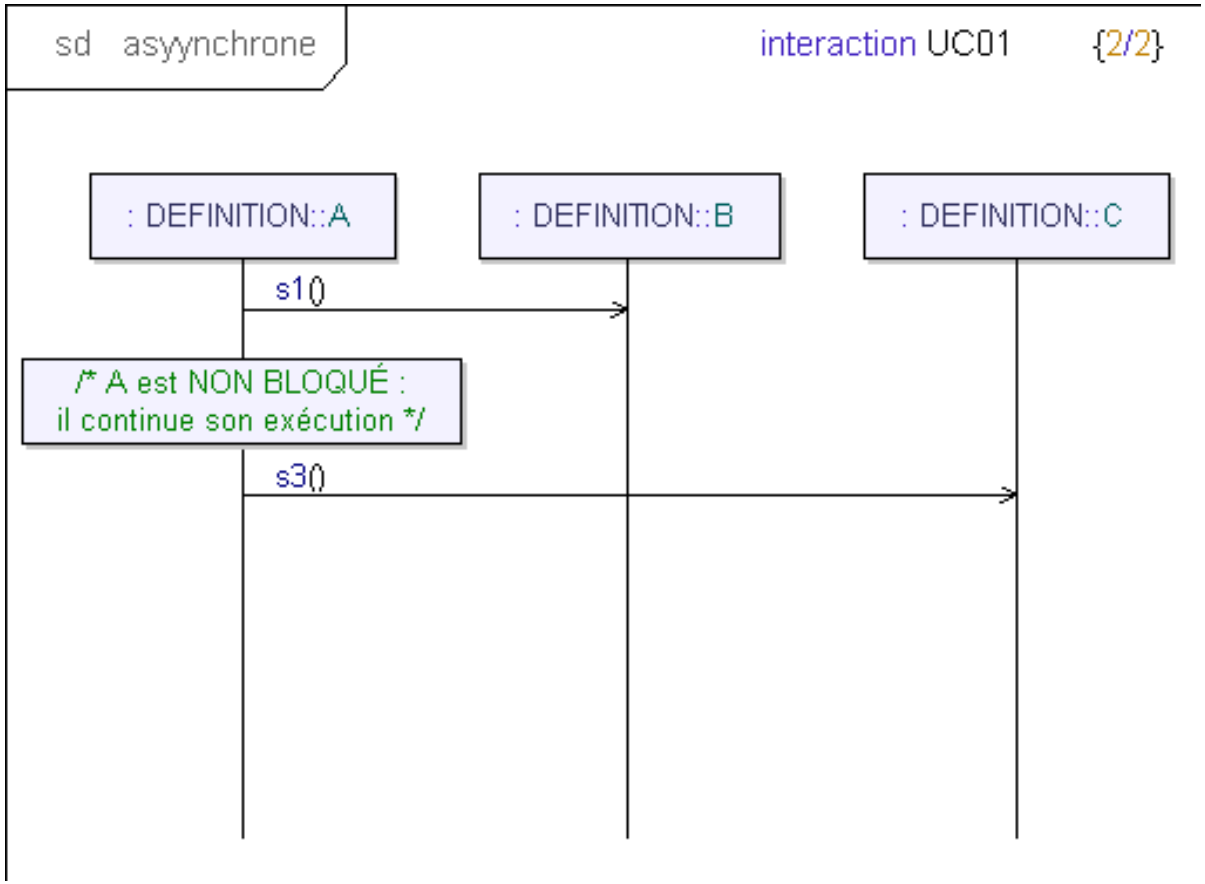
- **message asynchrone**
  N'interrompt pas l'exécution de l'expéditeur.
  Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité)

- **message dérobant**
  N'interrompt pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message

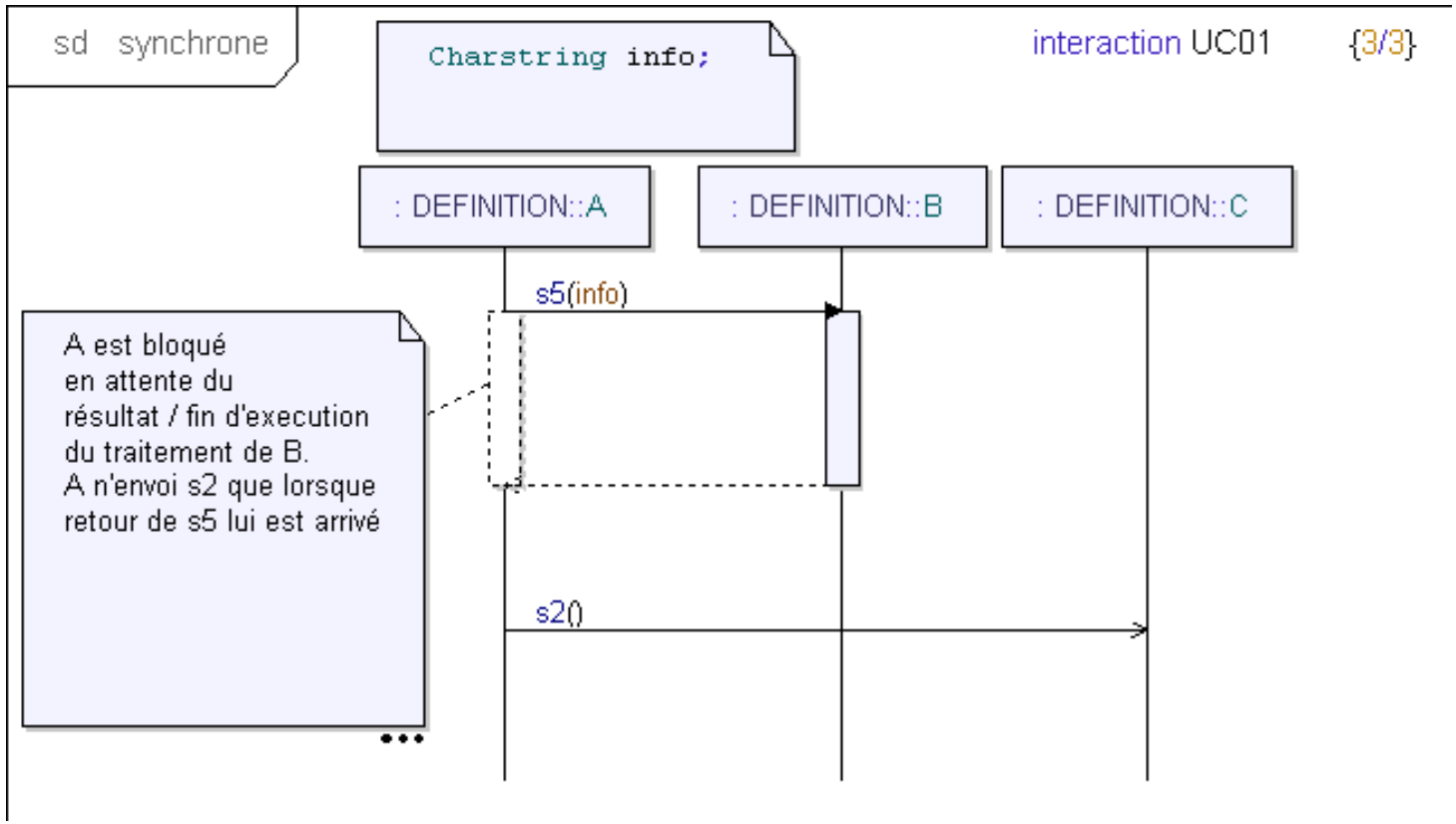
5. Différent types de message



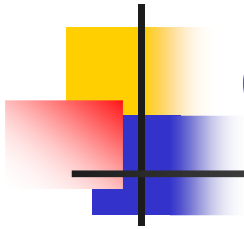
5. Message asynchrone



5. Message synchrone

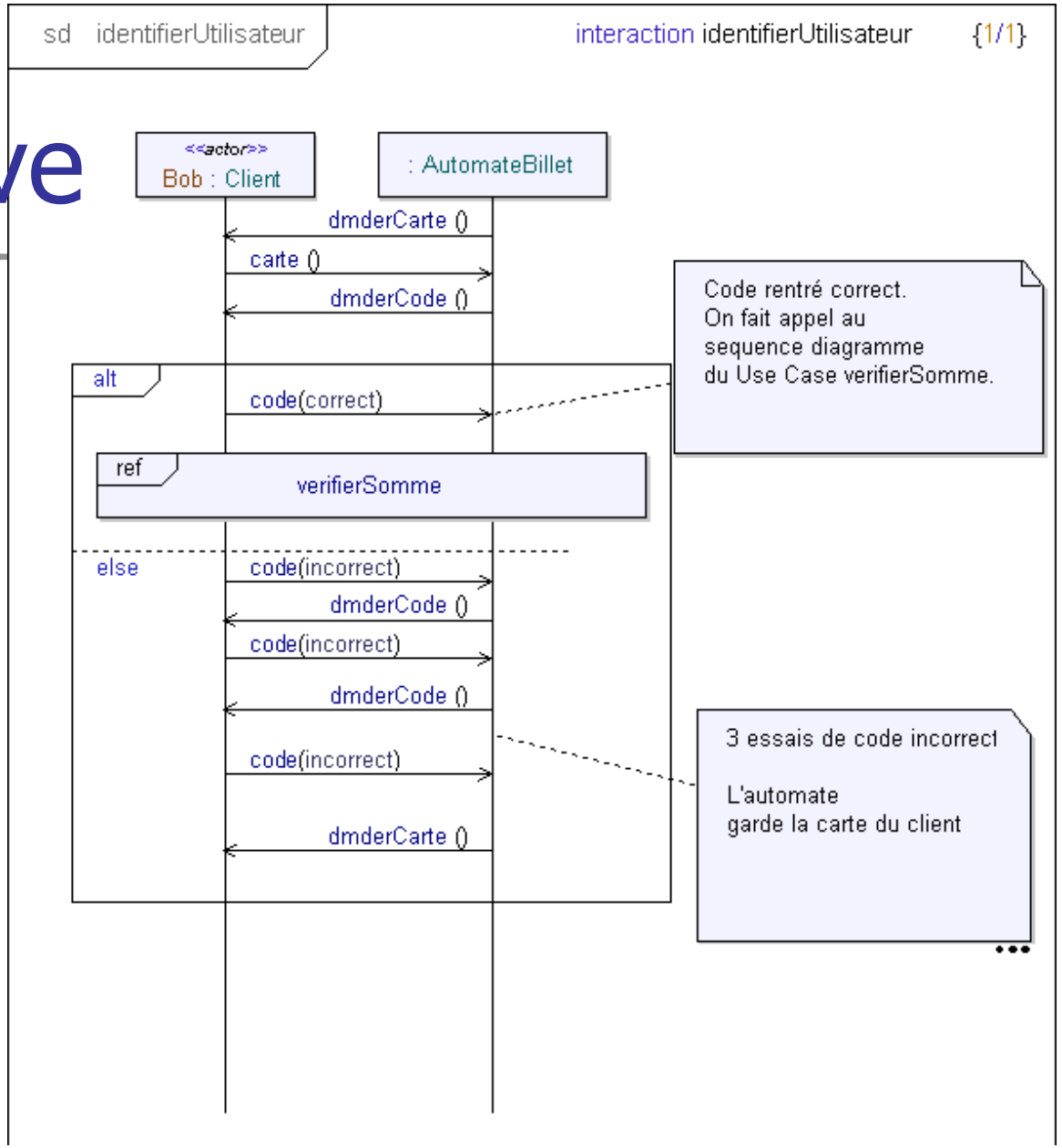


6. Structures de contrôle… des opérateurs

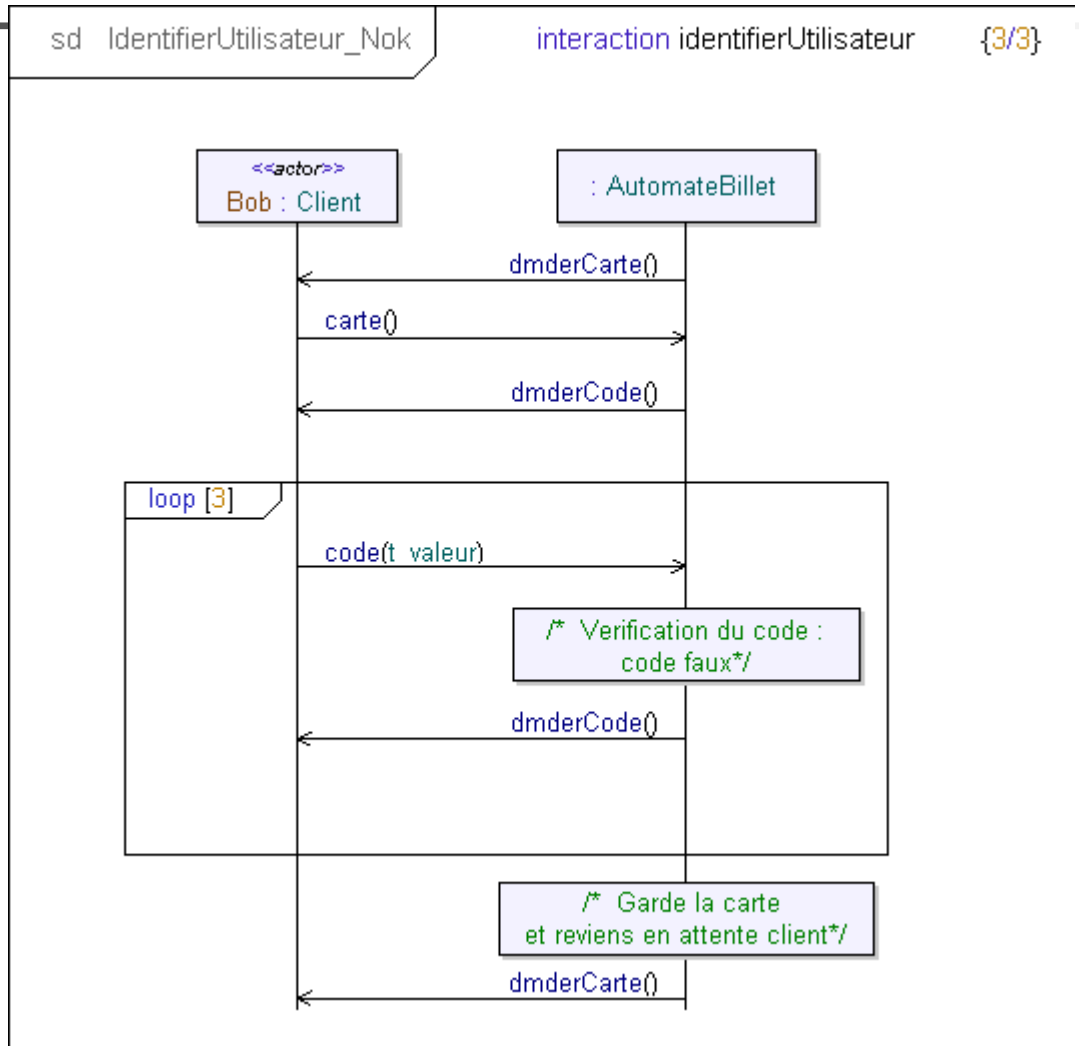


- De choix et de boucle :
  alternative, option, break et loop
- Contrôlant l'envoi en parallèle de messages :
  parallel et critical region
- Contrôlant l'envoi de messages :
  ignore, consider, assertion et negative
- Fixant l'ordre d'envoi des messages :
  weak sequencing , strict sequencing

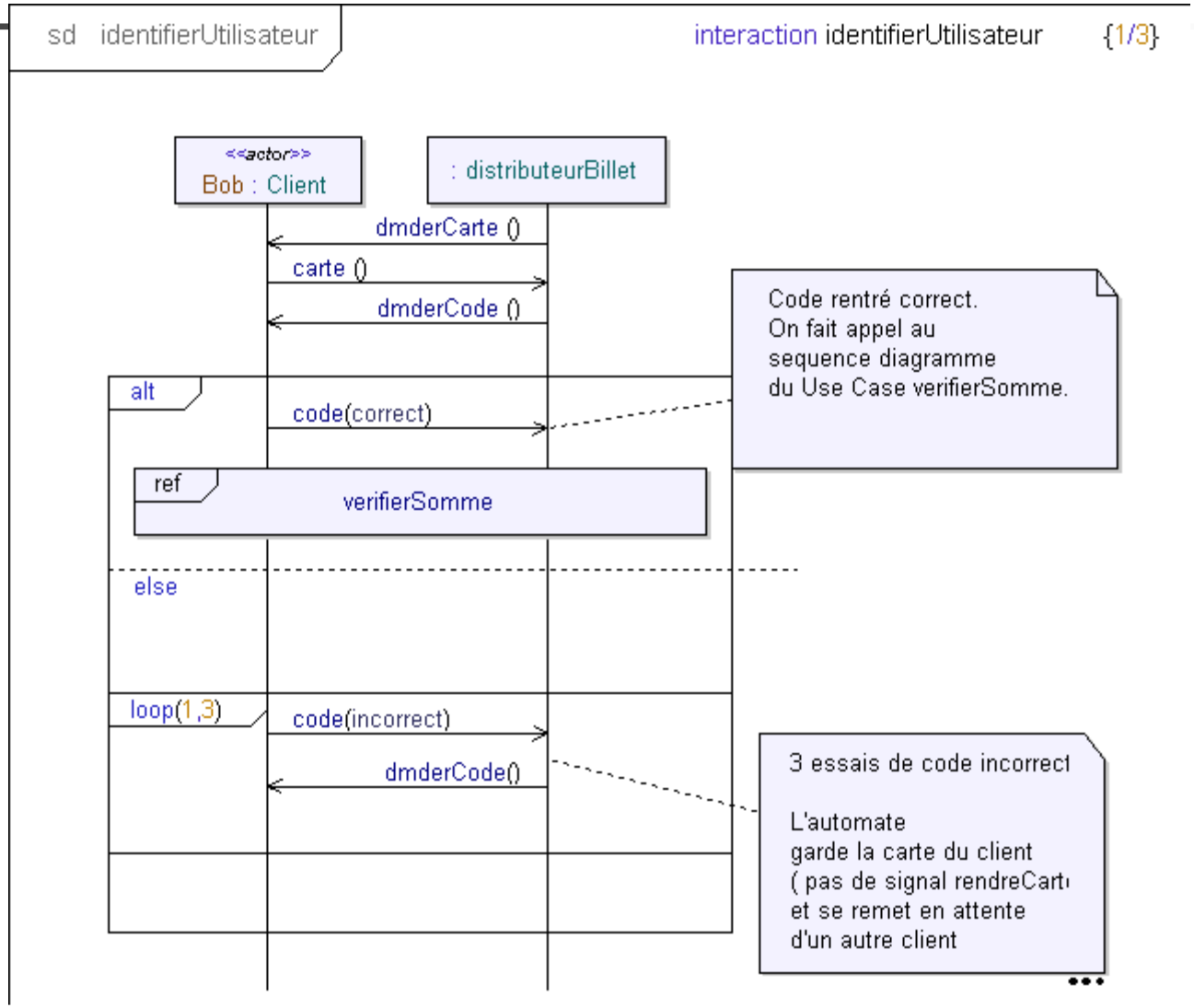
Alternative

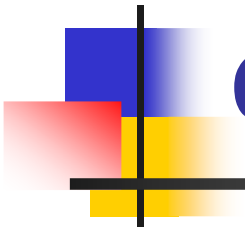


Loop


sd IdentifierUtilisateur_Nok
interaction identifierUtilisateur {3/3}
<<actor>>
Bob : Client
: AutomateBillet
dmderCarte()
carte()
dmderCode()
loop [3]
code(t valeur)
/* Verification du code :
code faux*/
dmderCode()
/* Garde la carte
et reviens en attente client*/
dmderCarte()

Exemple : DAB


sd identifierUtilisateur
interaction identifierUtilisateur {1/3}
<<actor>>
Bob : Client
: distributeurBillet
dmderCarte ()
carte ()
dmderCode ()
Code rentré correct.
On fait appel au
sequence diagramme
du Use Case verifierSomme.
alt
code(correct)
ref
verifierSomme
else
loop(1,3)
code(incorrect)
dmderCode()
3 essais de code incorrect
L'automate
garde la carte du client
(pas de signal rendreCart
et se remet en attente
d'un autre client



Exemple : outil de recherche de carte sur Internet

Diagramme de cas d'utilisation

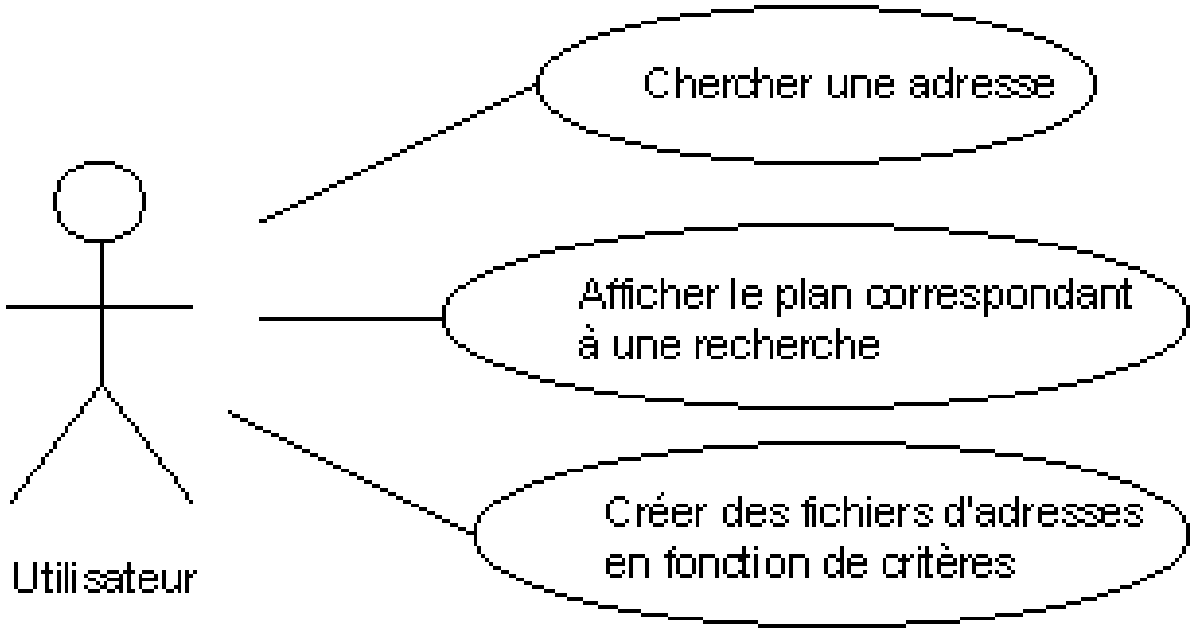
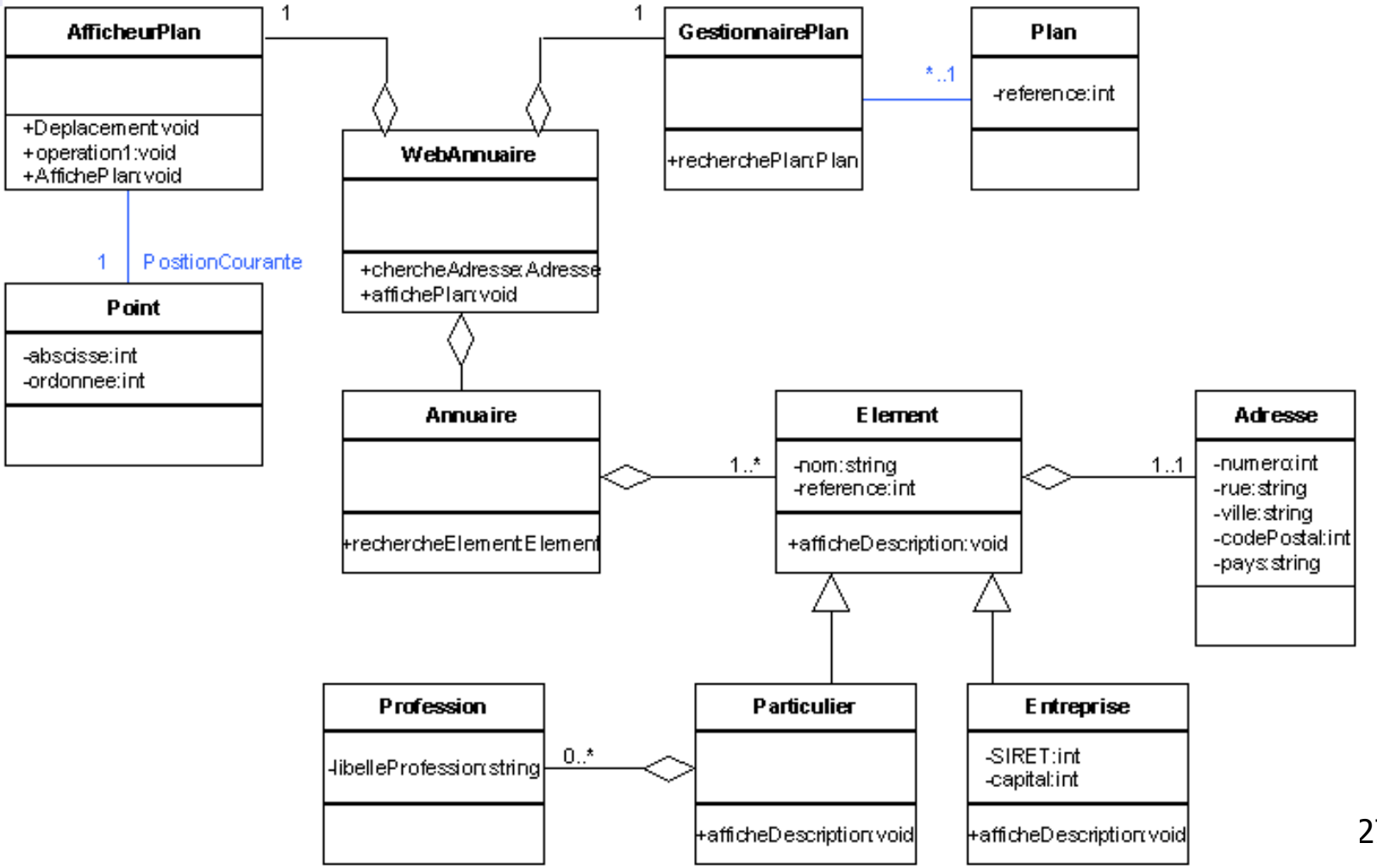
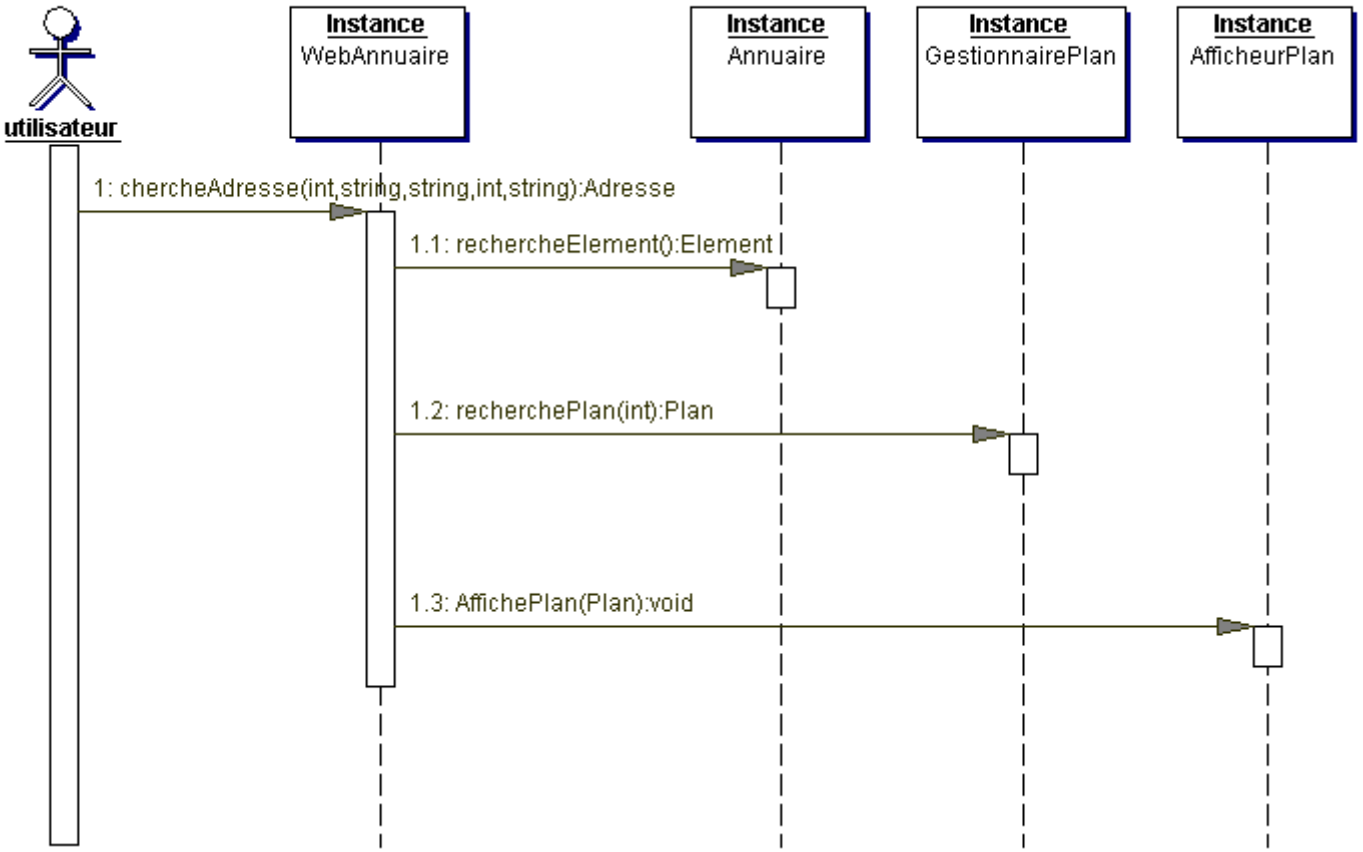
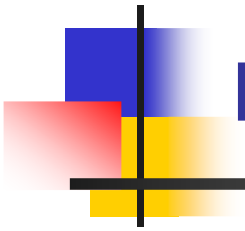


Diagramme de classe



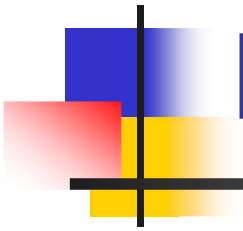
Chercher une adresse





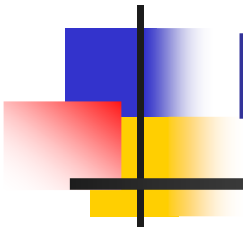
Exercice
Système de réservation de ressource

1. Traduire 2 scénarios en diagramme de séquence
2. Proposer un diagramme de classe
3. Détailler les 2 scénarios par rapport au diagramme de classe



Exercice

distributeur de boisson



Exercice

Achat sur un site marchand Internet