



Cours 5 : UML diagramme de comportements (activités & séquences)

Modèles dynamiques

- ❑ **Diagramme de cas d'utilisation (user case diagram)**
 - Interaction entre les acteurs et le système
- ❑ **Diagramme d'activités (activity diagram)**
 - Flux d'activités ou enchaînement d'activités
 - Extension des **réseaux de pétri**
- ❑ **Diagramme de séquences (sequence diagram)**
 - Déroulement séquentielle des traitements et des interactions entre les éléments du système et/ou de ses acteurs
 - ❑ Utilise le diagramme de classes

Diagramme d'activités

Diagramme d'activités
<http://www.uml-sysml.org>

Les diagrammes d'activités

❑ Pourquoi faire ?

- Mettre l'accent sur les traitements
 - ❑ Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données.
- Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation

❑ Utilisation courante

- adaptés à la description des cas d'utilisation
 - ❑ Ils viennent illustrer et consolider la description textuelle des cas d'utilisation
 - ❑ Décrire des activités séquentiels
 - une utilisation d'UML comme langage de programmation visuelle

Les composants des diagrammes d'activités

□ Action

- Le plus petit traitement qui puisse être exprimé
- Une action a une incidence sur l'état du système
- Graphique : rectangle ou rectangle arrondi

□ Transition

- Le passage d'une activité vers une autre est matérialisé par une transition.
- Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles



Les composants des diagrammes d'activités

□ **Nœud initial** (*initial node*)

- Le flot débute lorsque l'activité enveloppante est invoquée.
- Une activité peut avoir plusieurs nœuds initiaux.
- Un nœud initial possède un arc sortant et pas d'arc entrant.
- Graphiquement : un petit cercle plein ●

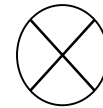
□ **Nœud de fin d'activité** (*final node*)

- Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant.
- Graphiquement : un cercle vide contenant un petit cercle plein ○●

Les composants des diagrammes d'activités

□ Nœud de **fin de flot** (*flow final*)

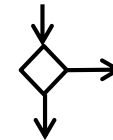
- lorsqu'un flot d'exécution atteint un nœud de fin de flot, le flot en question est terminé,
- Pas d'incidence sur les autres flots actifs de l'activité enveloppante.
- Graphiquement : un cercle vide barré d'un X :



Les composants des diagrammes d'activités

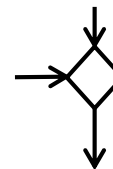
□ Nœud de décision (*decision node*)

- Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants.
- un arc entrant et plusieurs arcs sortants accompagnés de conditions **[if]** **[else]** (condition de garde)
- Graphiquement : un losange



□ Nœud de fusion (*merge node*)

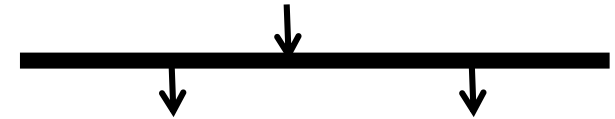
- Un nœud de fusion qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant.
- accepte un flot parmi plusieurs
- Graphiquement : un losange



Les composants des diagrammes d'activités

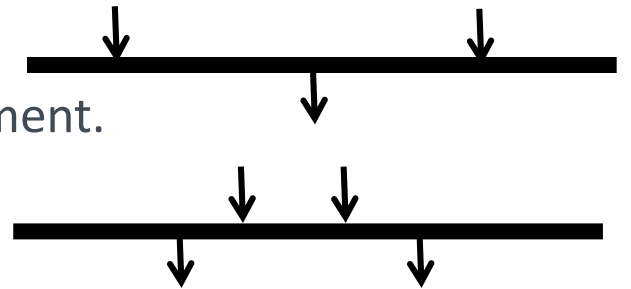
❑ nœud de **bifurcation** (*fork node*)

- sépare un flot en plusieurs flots concurrents
- possède donc un arc entrant et plusieurs arcs sortants.
- Graphiquement, un trait plein
- apparié généralement avec un nœud d'union



❑ nœud **d'union** (*join node*)

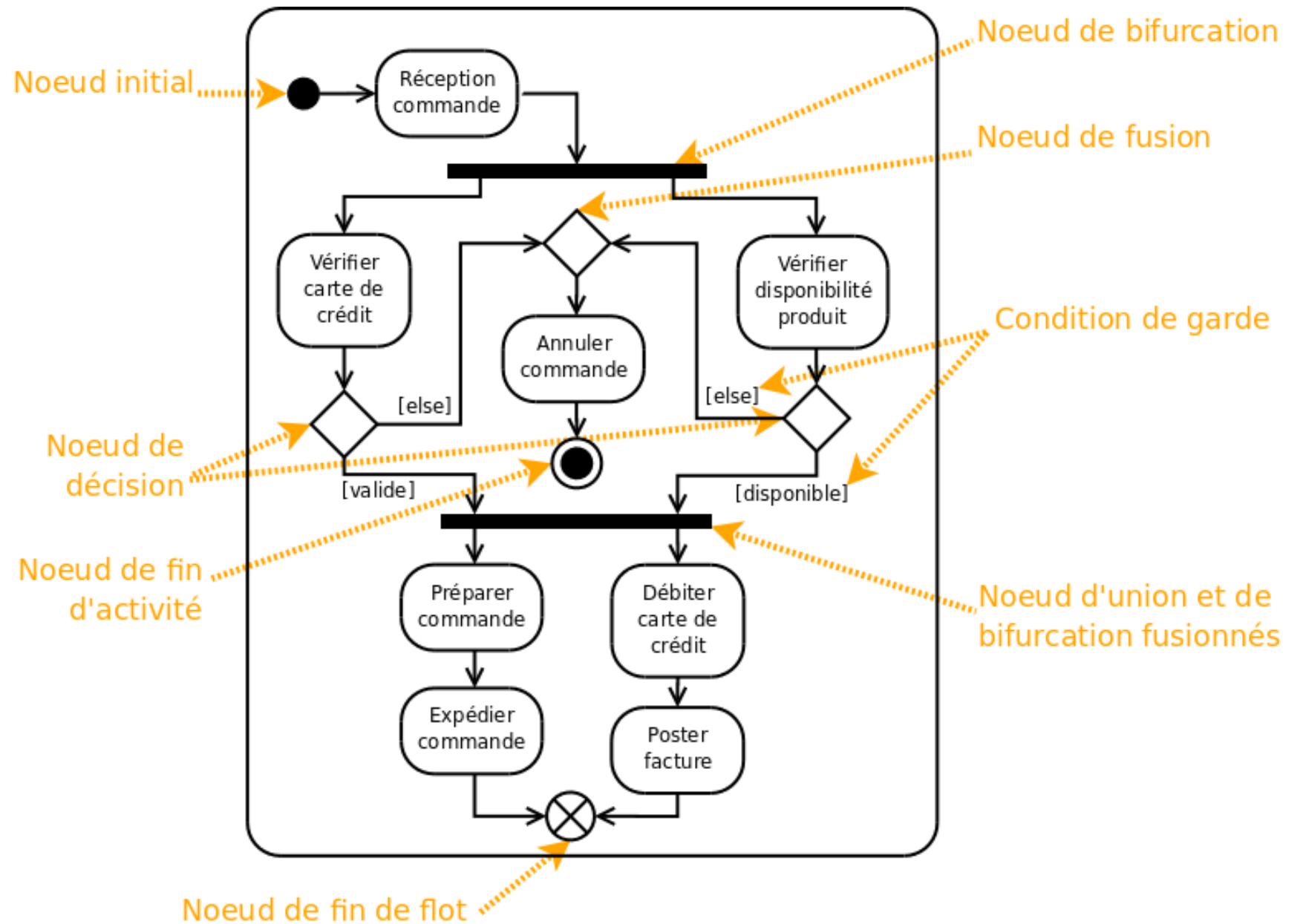
- synchronise des flots multiples.
- possède donc plusieurs arcs entrants et un seul arc sortant.
- Lorsque tous les arcs entrants sont activés, l'arc sortant l'est également.
- Graphiquement, un trait plein



❑ Remarque

- il est possible de fusionner un nœud de bifurcation et un nœud d'union, et donc d'avoir un trait plein possédant plusieurs arcs entrants et sortants

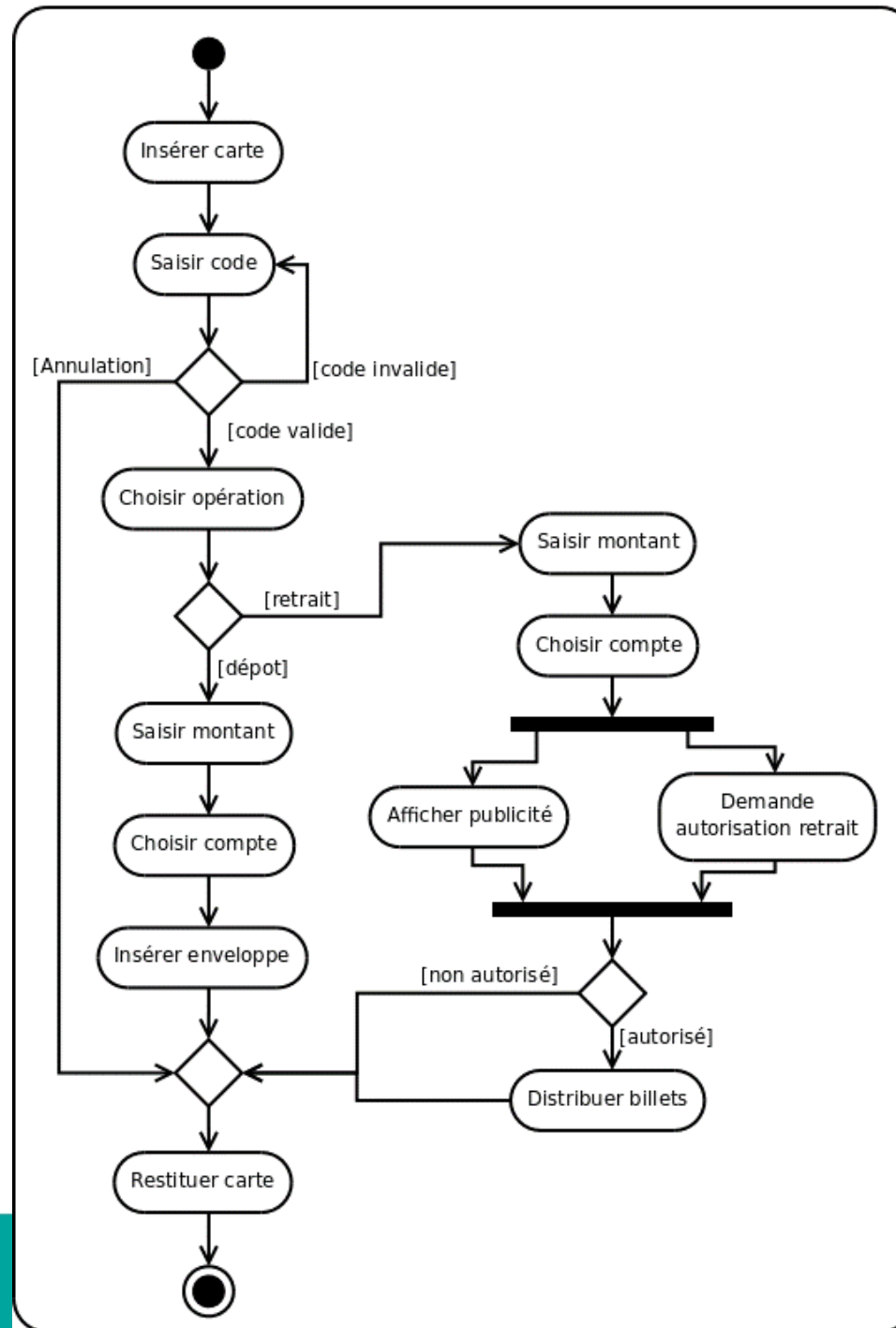
Premier exemple



Deuxième exemple

❑ Questions

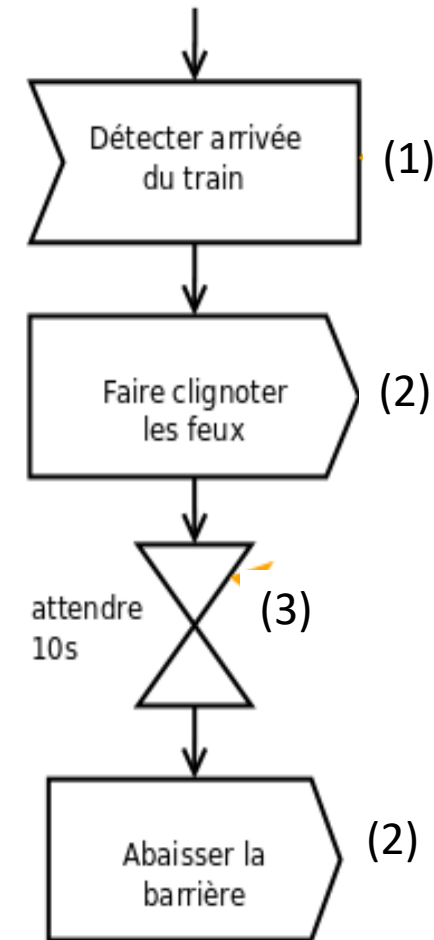
- Y a-t-il la possibilité d'enchaîner plusieurs opérations ?
- Y a-t-il un nombre maximum d'essais pour le code ?



Diagrammes d'activités : actions spécifiques

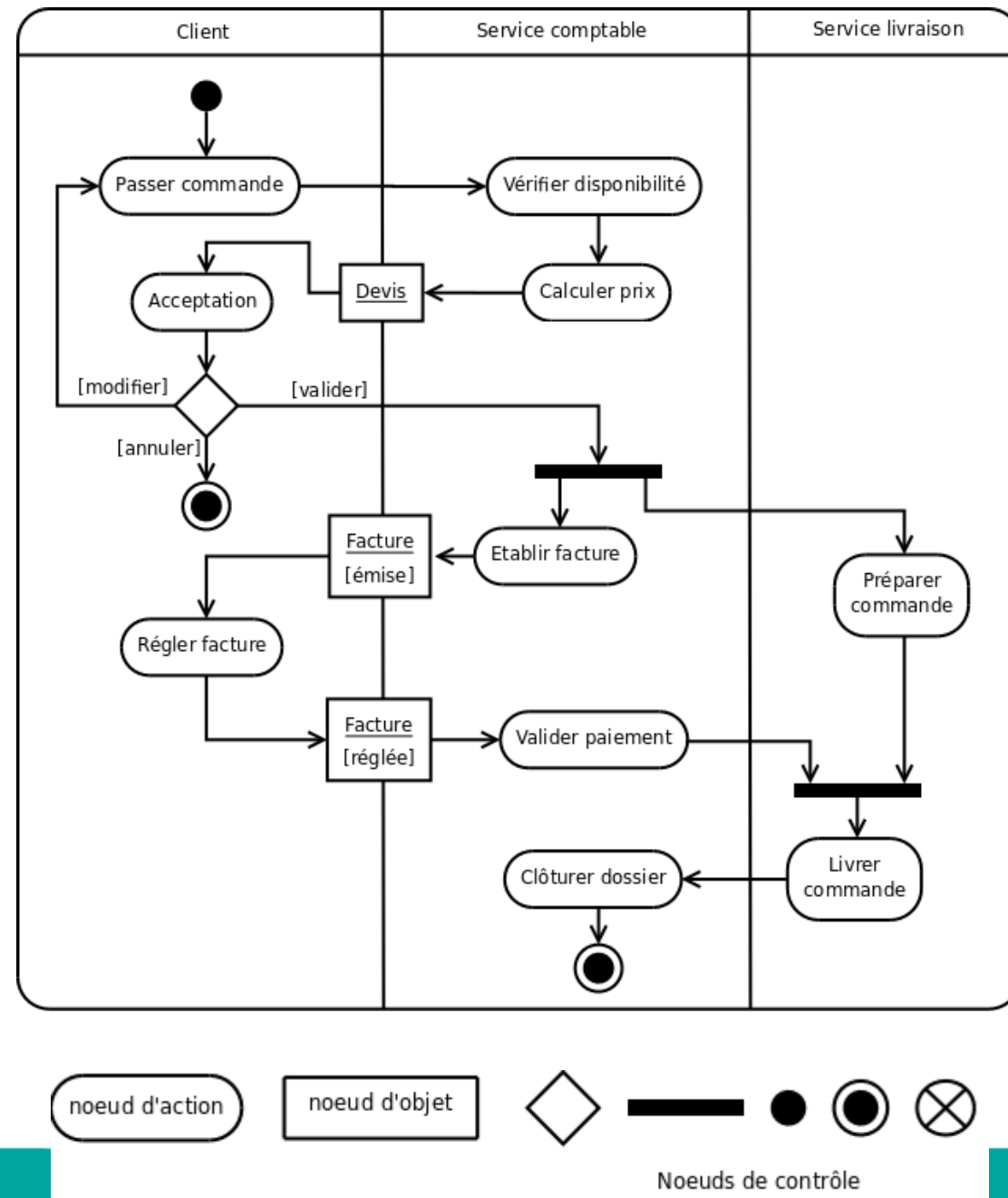
□ Action de type Signaux

- capteur
 - Action d'envoi de signal (1)
- Actionneur
 - Action de réception de signal (2)
- Timmer
 - Action temporelle (3)



Diagrammes d'activités :

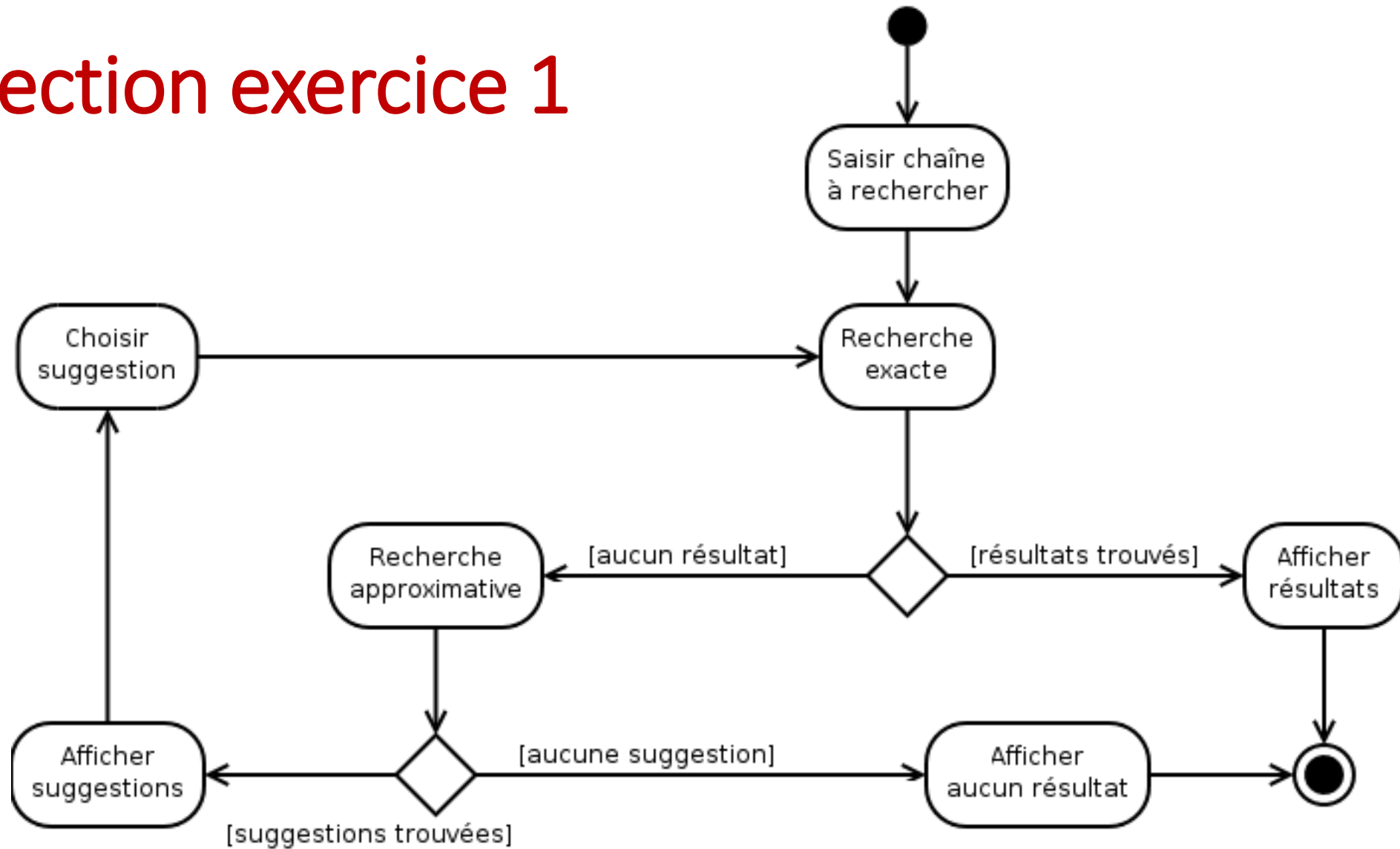
- ❑ Segmentation par système
- ❑ Passage d'objets



Exercice 1 : Rechercher une chaîne de caractères

- ❑ L'utilisateur recherche sur un moteur une chaîne de caractères
- ❑ Si la chaîne exacte est trouvée, alors les résultats possédant cette chaîne sont affichés, sinon des suggestions sont proposées à l'utilisateur, il peut alors choisir parmi les suggestions, pour définir une nouvelle recherche
- ❑ Si aucune suggestion n'est trouvée alors un message signifiant qu'il n'y a pas de résultat est affiché

Correction exercice 1



Exercice 2 : Café

□ Pour faire du café, vous devez :

- Aller chercher du café,
- Chercher de l'eau
- Chercher une tasse
- Mettre un filtre
- Remplir le filtre de café,
- Remplir le réservoir d'eau
- Mettre le récipient dans la cafetière
- Allumer la cafetière
- Attendre que la cafetière s'éteigne
- Servir le café dans la tasse

□ Définissez le diagramme d'activités associé.

Diagramme de séquences

A solid teal horizontal bar spanning the width of the slide, featuring a central V-shaped notch.

Diagramme de séquences

- ❑ Il permet de montrer les interactions d'objets dans le cadre d'un scénario
- ❑ Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets
 - Les objets et acteurs arrivent-il à communiquer les bons messages au bons moments ?
 - L'ensemble des messages (des méthodes) sont-ils disponibles ?

Diagramme de séquences

❑ Quand l'utiliser ?

- Pour affiner un **diagramme de cas**
 - ❑ Diagramme de séquences peu détaillé
- Pour décrire la partie dynamique associée à un **diagramme de classes**
 - ❑ Diagramme de séquences détaillé
 - ❑ Que modéliser ?
 - Les événements principaux qui modifient l'état du système?
 - Exemples : Création d'un nouveau client, Annulation d'une commande
 - Flot de données classiques

Diagramme de séquences

□ Buts

- Détailler le fonctionnement dynamique
- Montrer les interactions entre
 - les acteurs et le système
 - les composants du système
- Fournir des exemples de fonctionnement
- Illustrer la dynamique d'enchaînement des traitements d'une application à travers les messages échangés entre objets
- Simuler sur papier le fonctionnement de l'application

diagramme de séquences

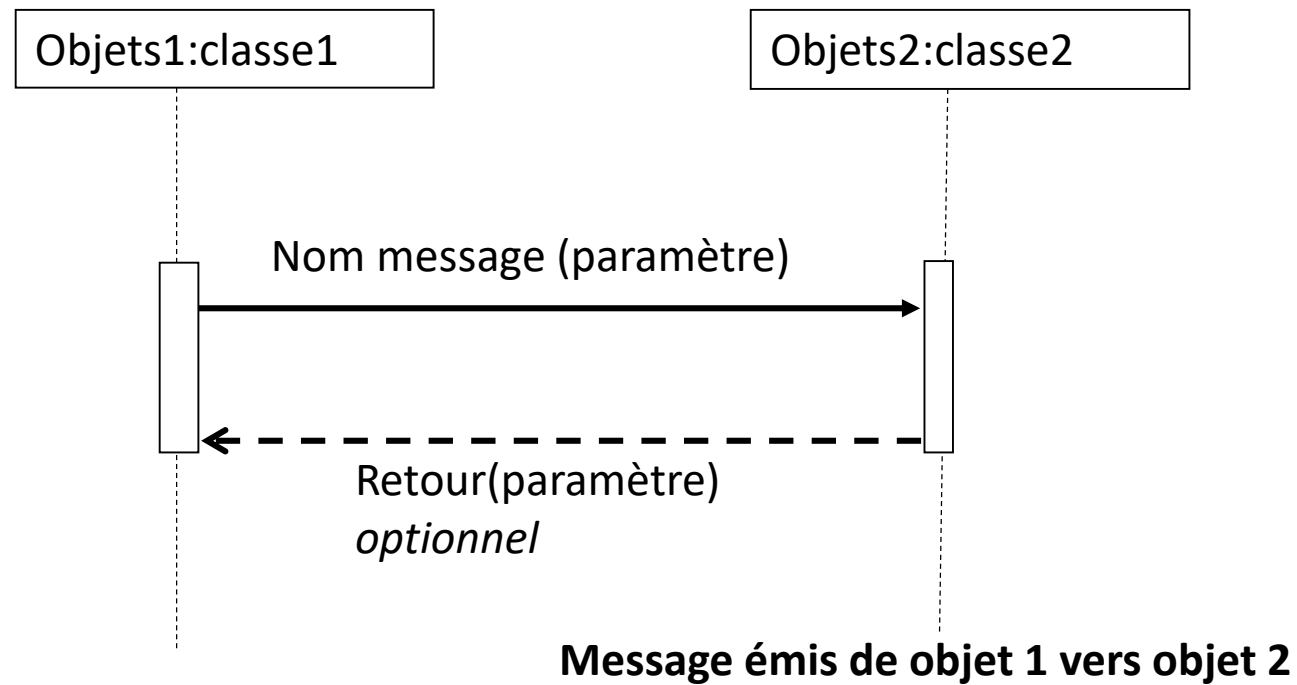
- ❑ **Un diagramme de séquences est un dessin qui illustre pour un scénario particulier d'un cas d'utilisation :**
 - les événements produits par les acteurs externes
 - leur ordre
 - les événements intersystème
- ❑ **Le diagramme de séquences du système doit être fait pour**
 - le scénario principal du cas d'utilisation
 - pour les scénarii alternatifs les plus intéressants
- ❑ **Depuis UML 2.0 possible de définir des scénarii alternatifs.**

Éléments d'un diagramme de séquences

- ❑ **Les éléments constitutifs d'un scénario sont**
 - un acteur (humain ou automate)
 - un ensemble d'objets et leurs états
 - la chronologie des échanges entre les objets de cet ensemble
 - ❑ messages (méthodes avec paramètres d'entrée et valeur de retour)
 - ❑ des signaux
 - ❑ des contraintes de temps

Formalisme

□ Objets et messages



Ligne de vie et activation

□ **Ligne de vie** (-----)

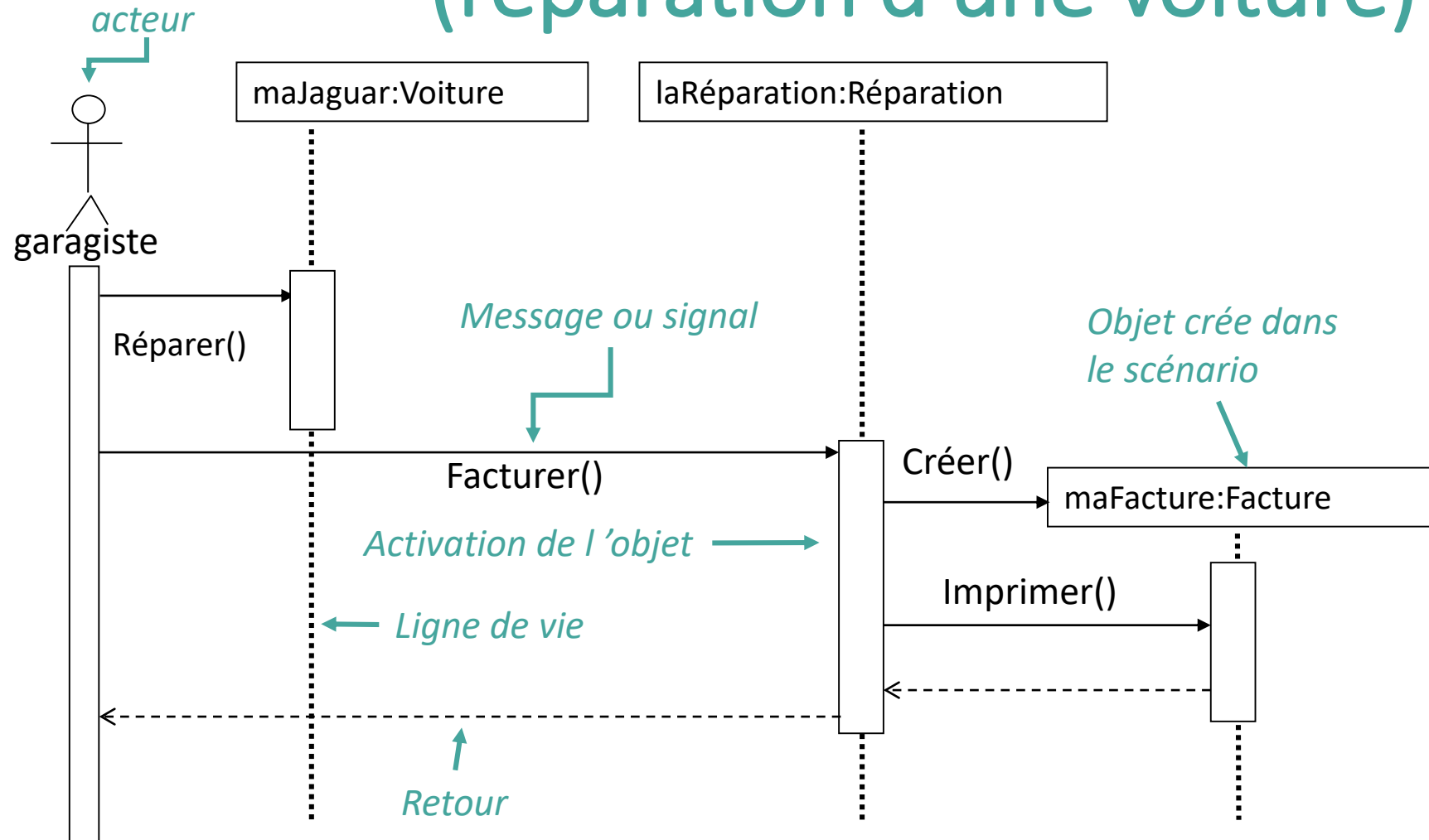
- existence d'un objet à un instant t
 - début : création
 - fin : destruction

□ **L'activation** ()

- période durant laquelle l'objet exécute une action lui-même ou via une autre procédure

□ **Un scénario se lit de haut en bas**

Exemple garagiste (réparation d'une voiture)



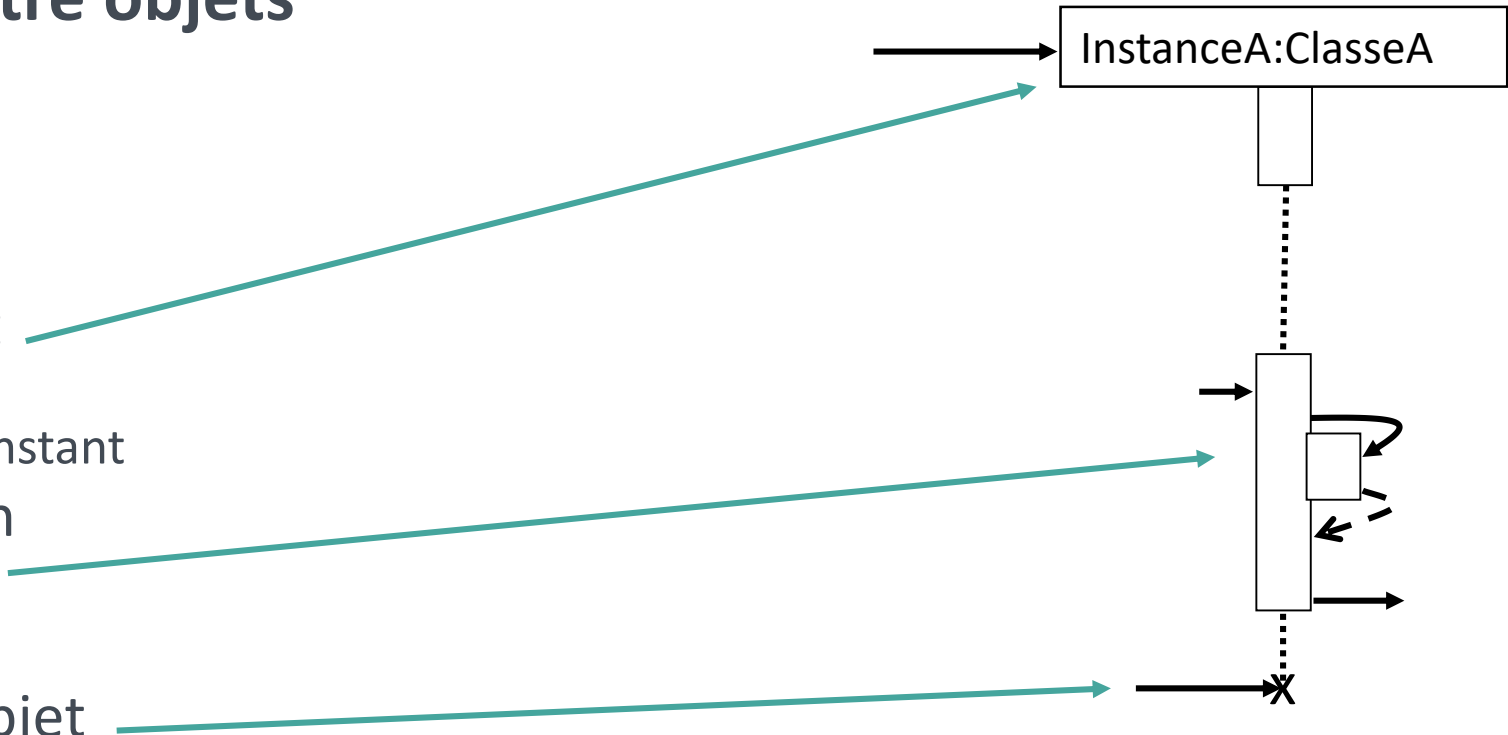
Messages

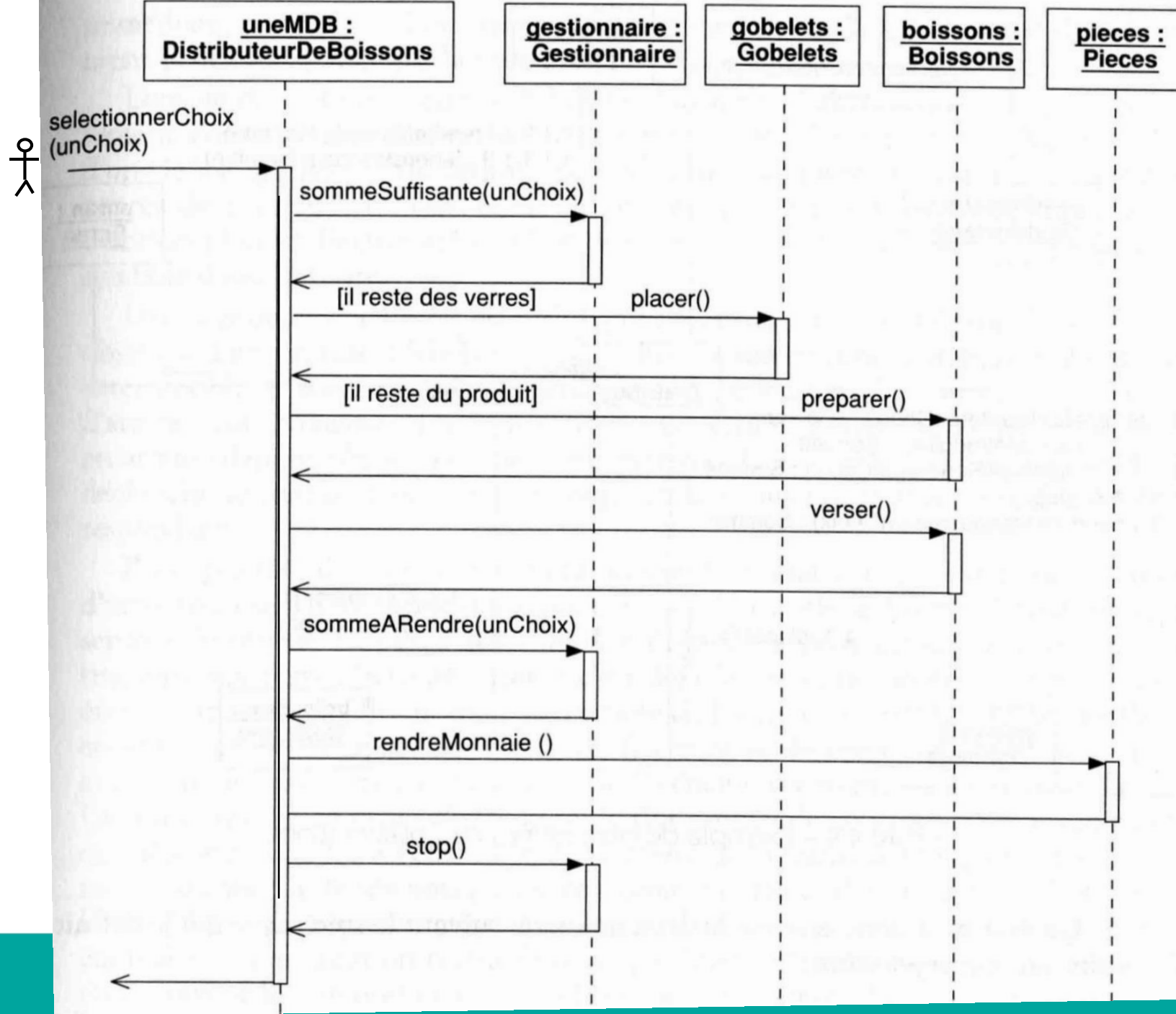
❑ Communication entre objets

- des paramètres
- un retour

❑ Cas particulier

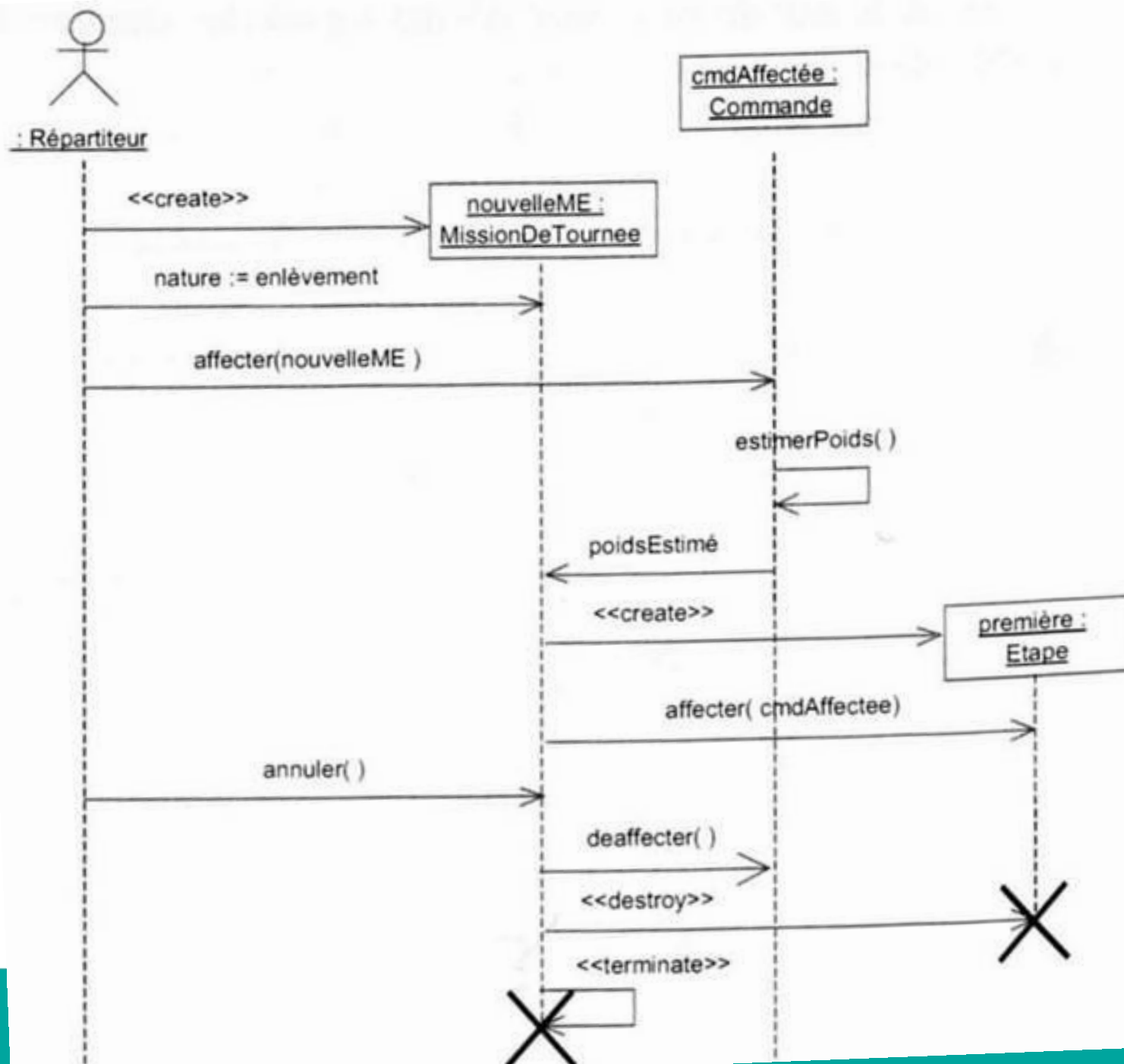
- création d'un objet
 - ❑ la ligne de vie commence à cet instant
- objet s'envoyant un message récursif
 - ❑ deuxième activité
- destruction d'un objet
 - ❑ une croix et fin de la ligne de vie





Distributeur de boissons

livraison

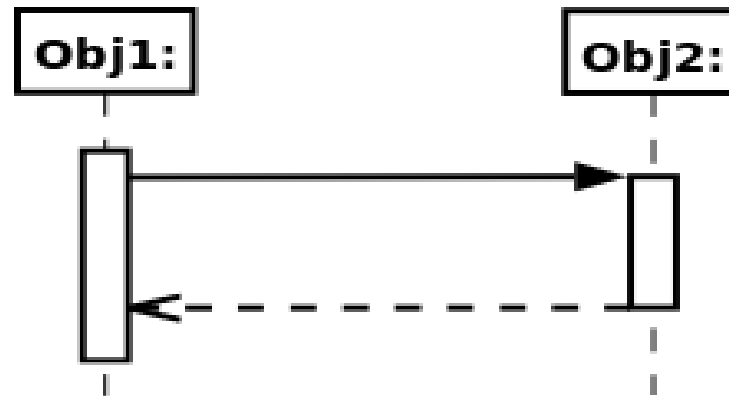


Synchronisation des messages

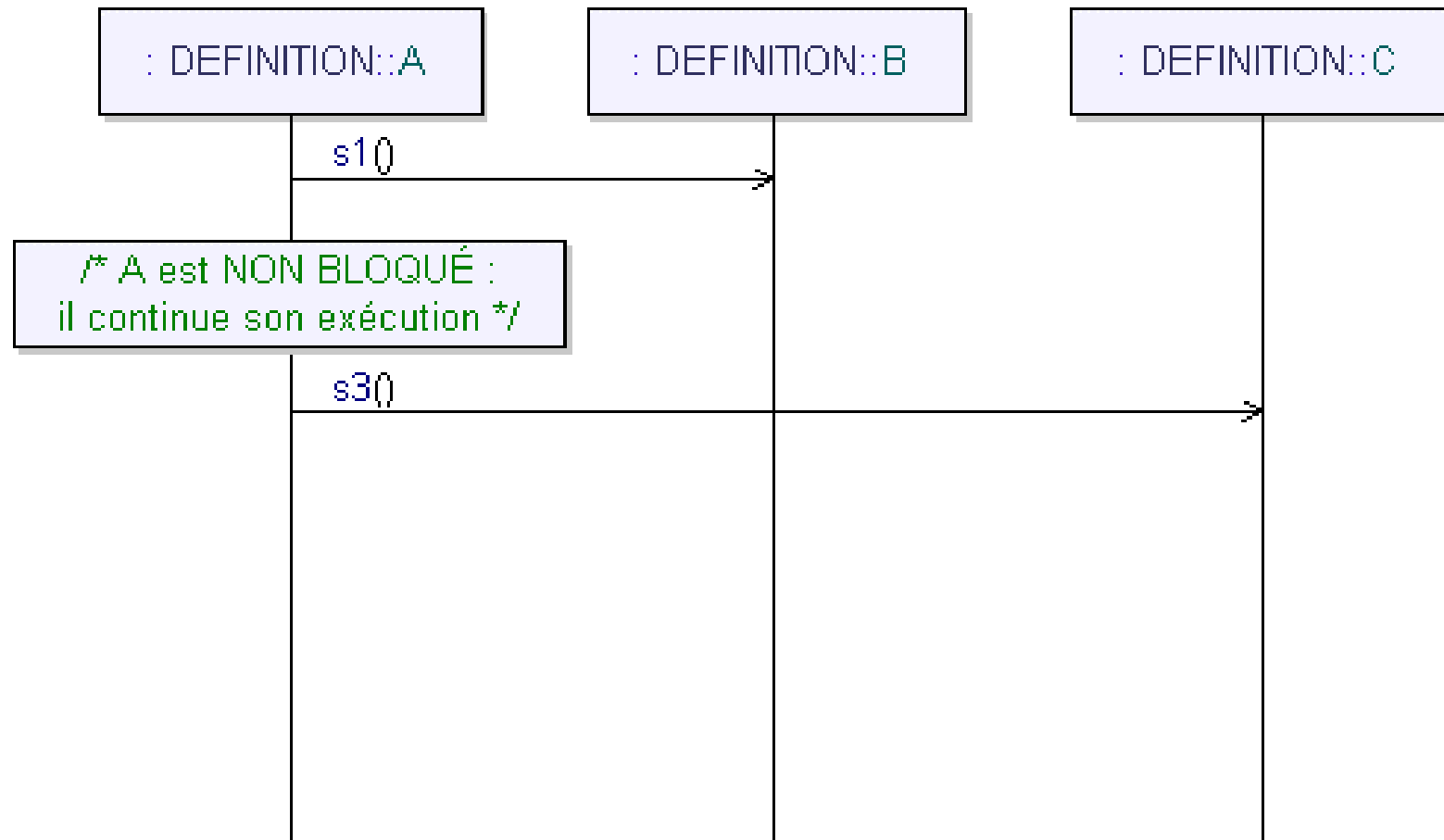
❑ Messages asynchrones



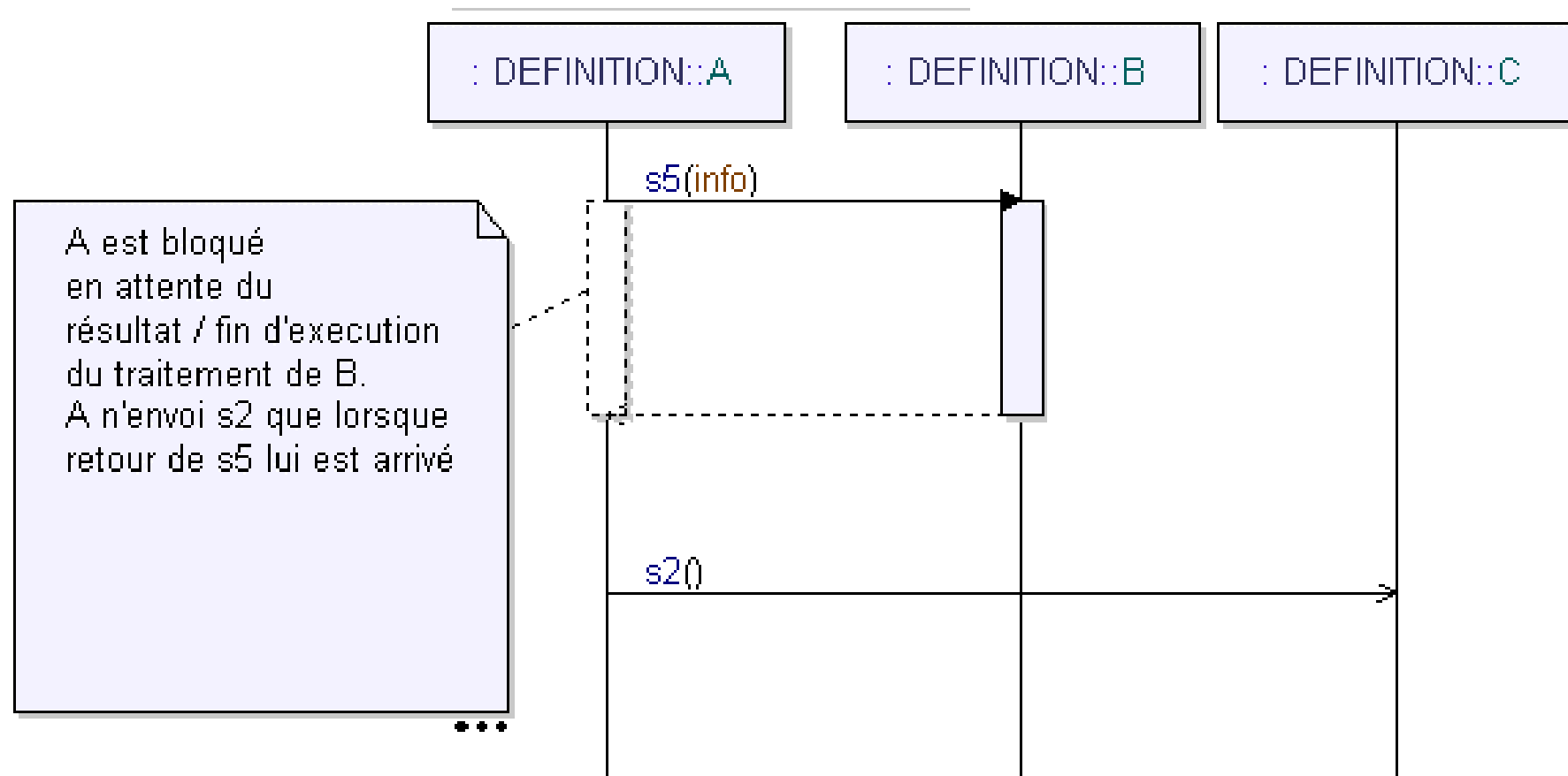
❑ Messages synchrones



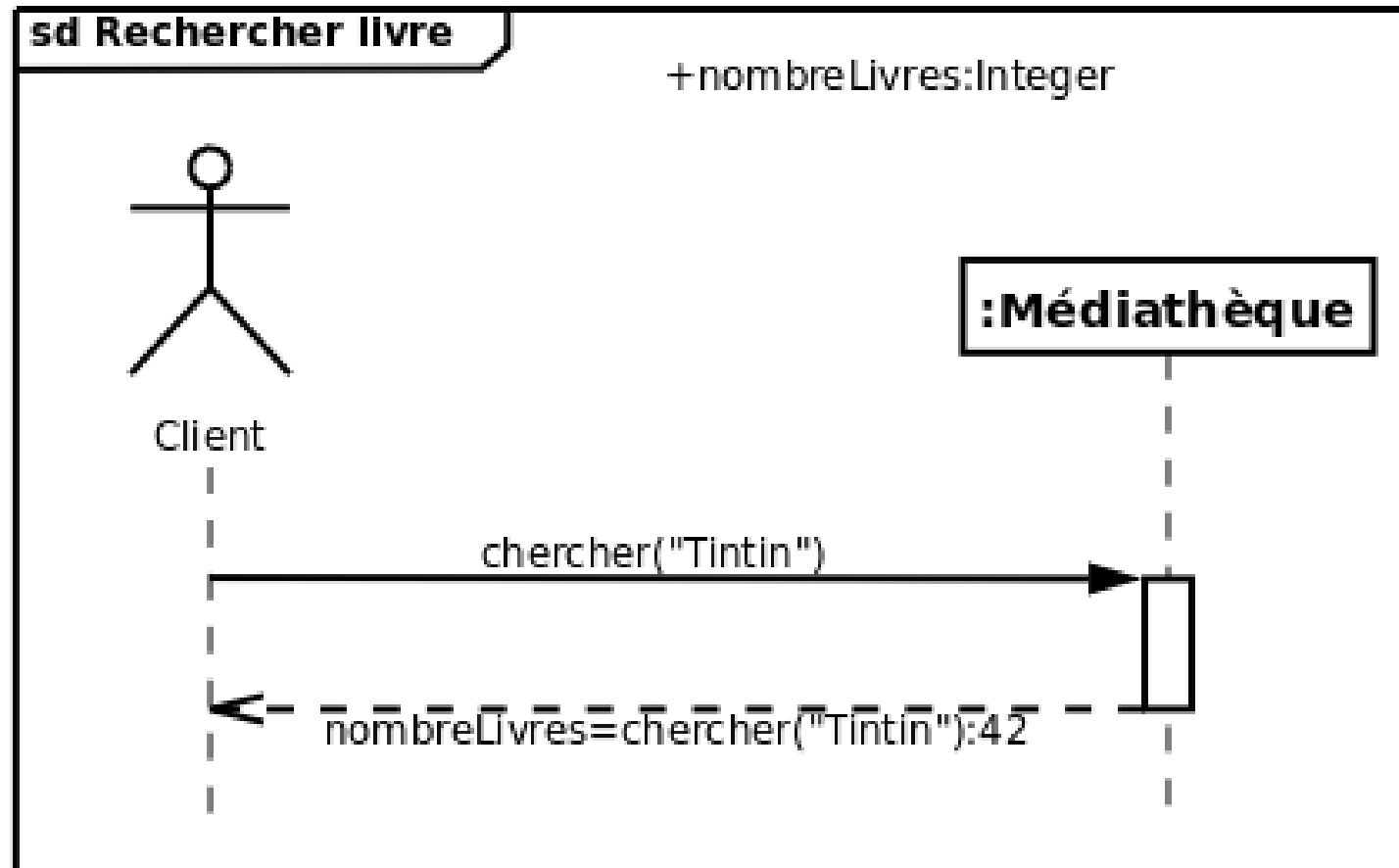
Message asynchrone



Message synchrone



Message synchrone



Fragments

□ Les différents de type de fragments

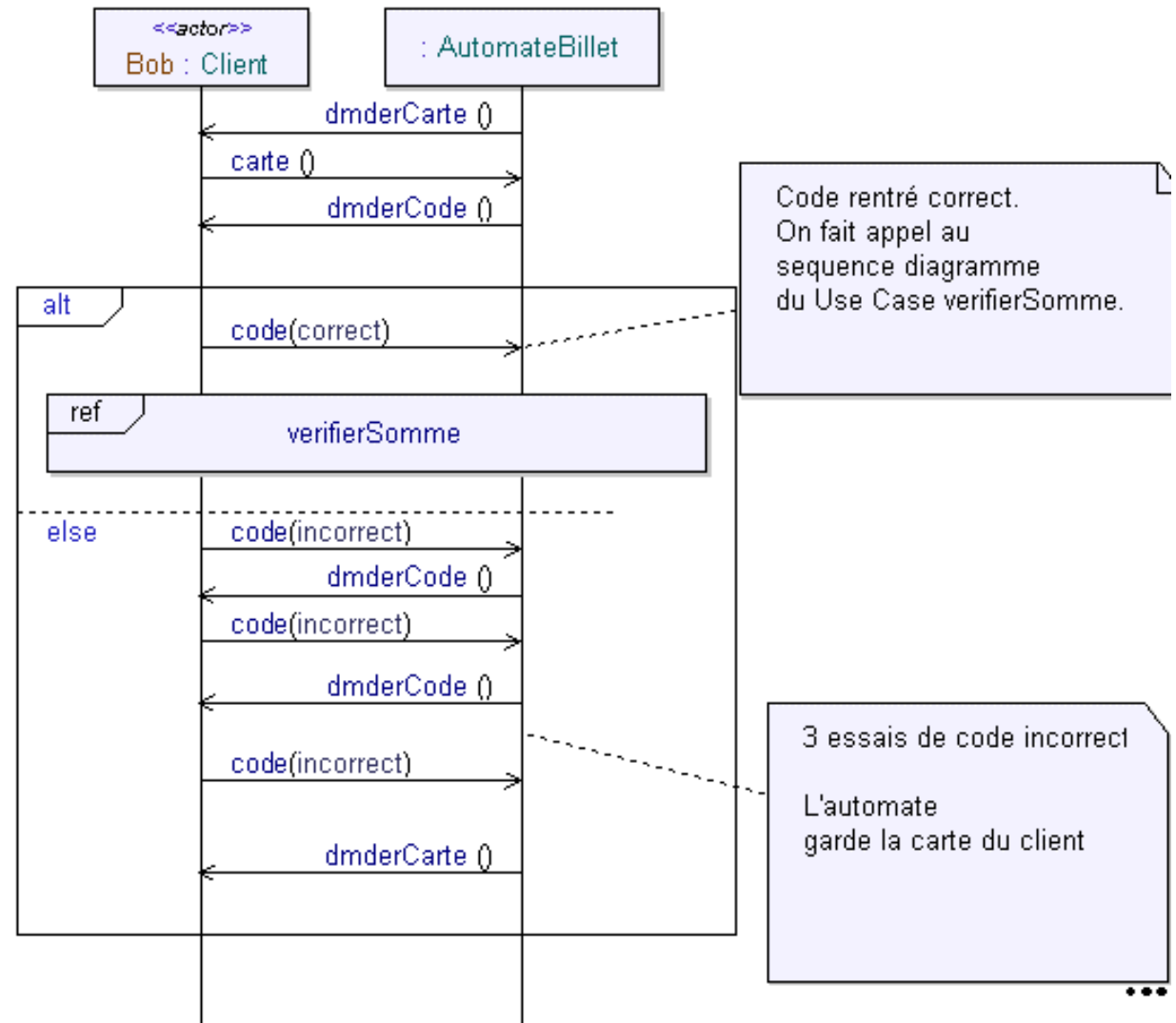
- **alt** : fragments multiple alternatifs (si alors sinon)
- **opt** : fragment optionnel
- **par** : fragment parallèle (traitements concurrents)
- **loop** : le fragment s'exécute plusieurs fois
- **region** : région critique (un seul thread à la fois)
- **neg** : une interaction non valable
- **break** : représente des scenarii d'exception
- **ref** : référence à une interaction dans un autre diagramme
- **sd** : fragment du diagramme de séquences en entier (inclus l'ensemble du diagramme de séquence)

Les fragments combinés

- ❑ **Les fragments combinés (combined fragment, inline frame)**
 - Un fragment combiné représente des articulations d'interactions.
 - Il est défini par
 - ❑ un opérateur
 - conditionne la signification du fragment combiné.
 - 10 opérateurs définis dans la notation **UML2.0**
 - ❑ des opérandes.
 - Permet de représenter plusieurs scénarii sur un même diagramme

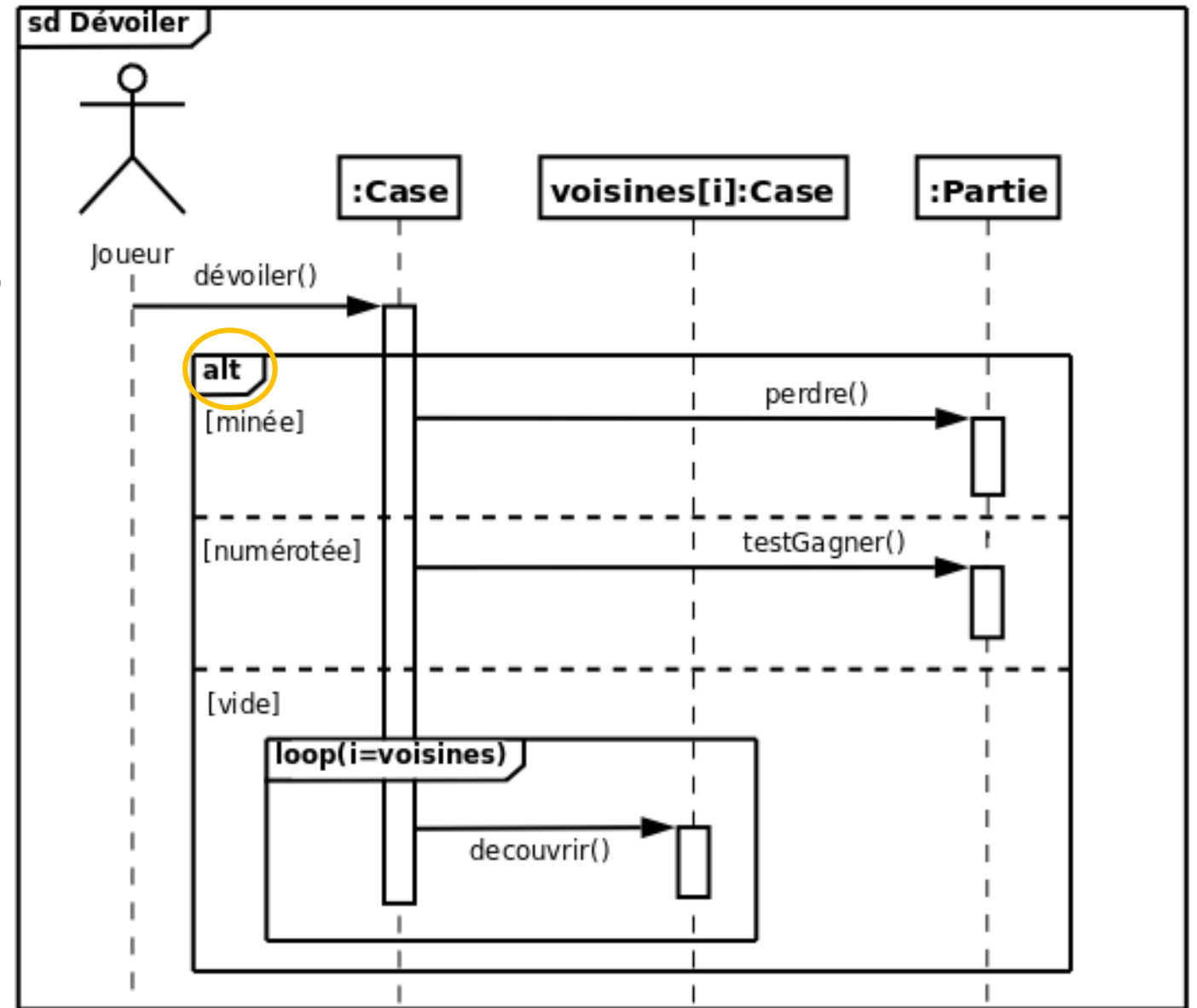
Les fragments combinés

- **Alternative : alt**
 - représente deux comportements possibles :
 - **SI...ALORS... SINON**
 - une seule des deux branches sera réalisée dans un scénario donné.



Opérateur alt

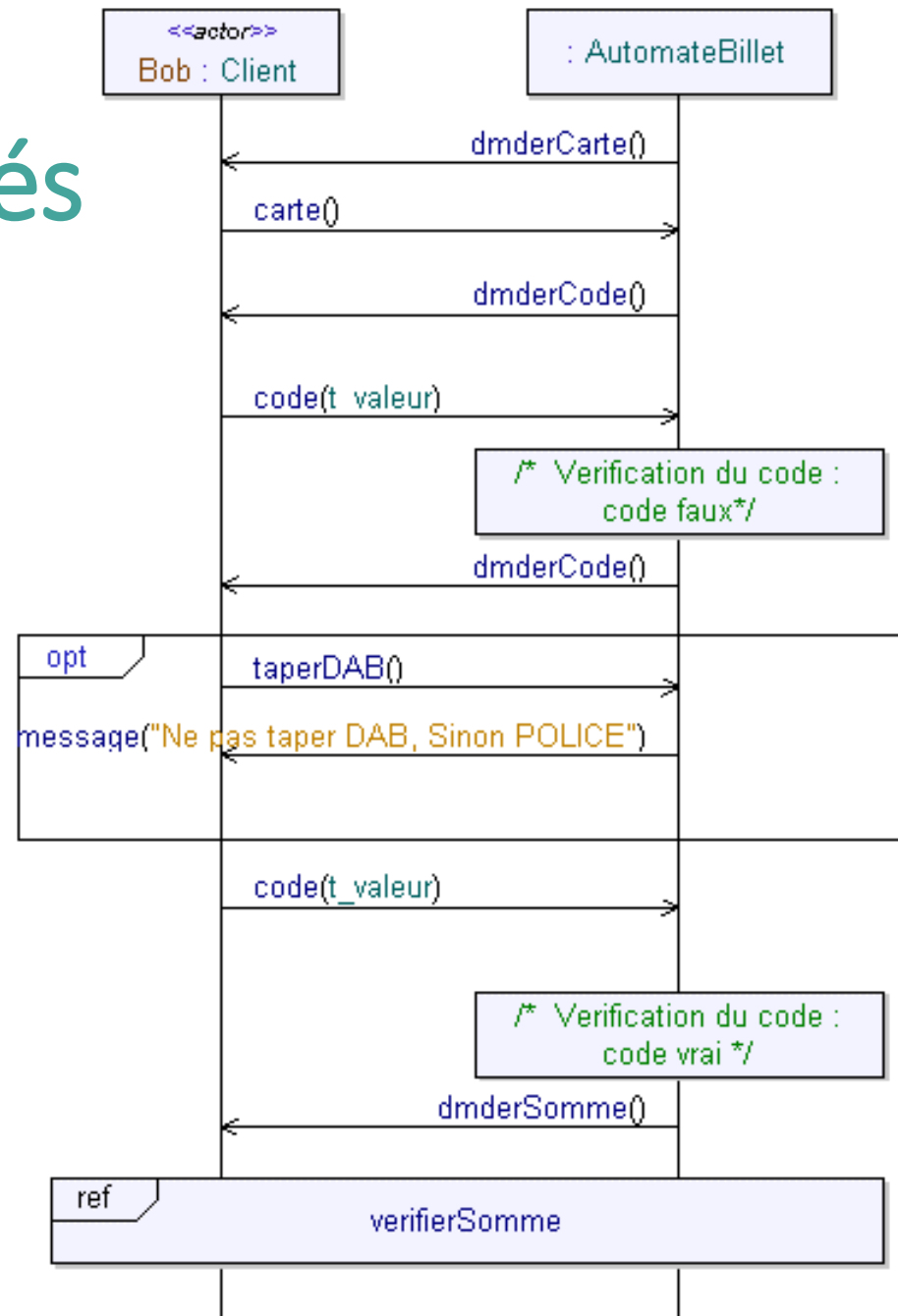
□ Exemple jeu du démineur



Les fragments combinés

❑ Option : **opt**

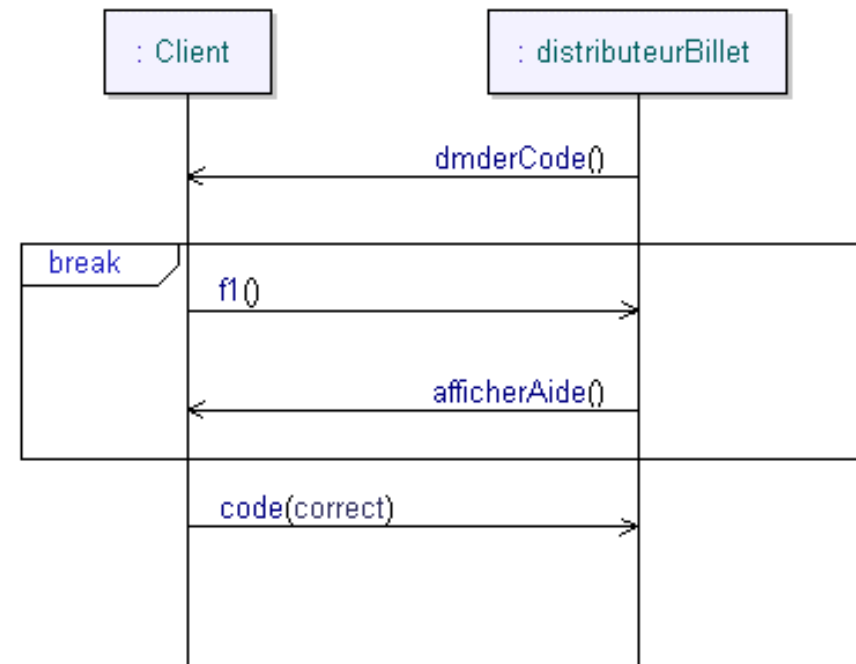
- fragment combiné optionnel
- Alt sans else
- Exemple *opérateur opt*
- *L'utilisateur, si il est mécontent, peut se défouler sur le distributeur de billets.*



Les fragments combinés

□ Break

- L'opérateur **break** est utilisé dans les fragments combinés qui représentent des scénarii **d'exception**
- Il y a donc une notion d'interruption du flot "normal" des interactions.



Les fragments combinés

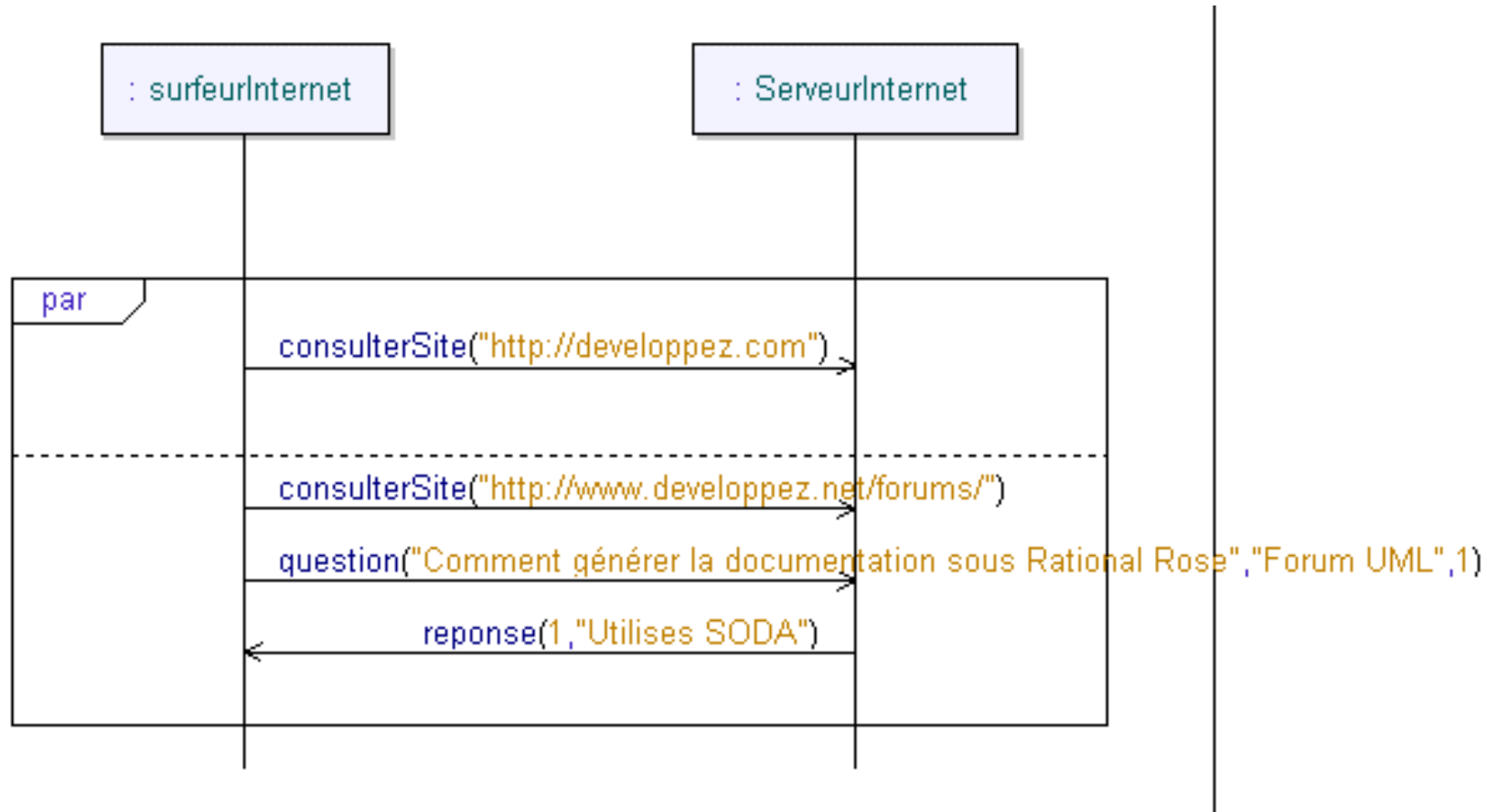
❑ Opérateur Parallèle : **par**

- représenter des interactions ayant lieu en parallèle.
 - ❑ les deux branches peuvent donc
 - se mélanger,
 - s'intercaler, dans la mesure où l'ordre imposé dans chaque opérande est respecté.

❑ Exemple

- Un développeur ayant accès à Internet peut consulter en parallèle, soit
 - ❑ <http://www.developpez.com>
 - ❑ <http://www.developpez.net/forums/>
- sans préférence d'ordre (il peut commencer par consulter les forums puis les cours, soit l'inverse).

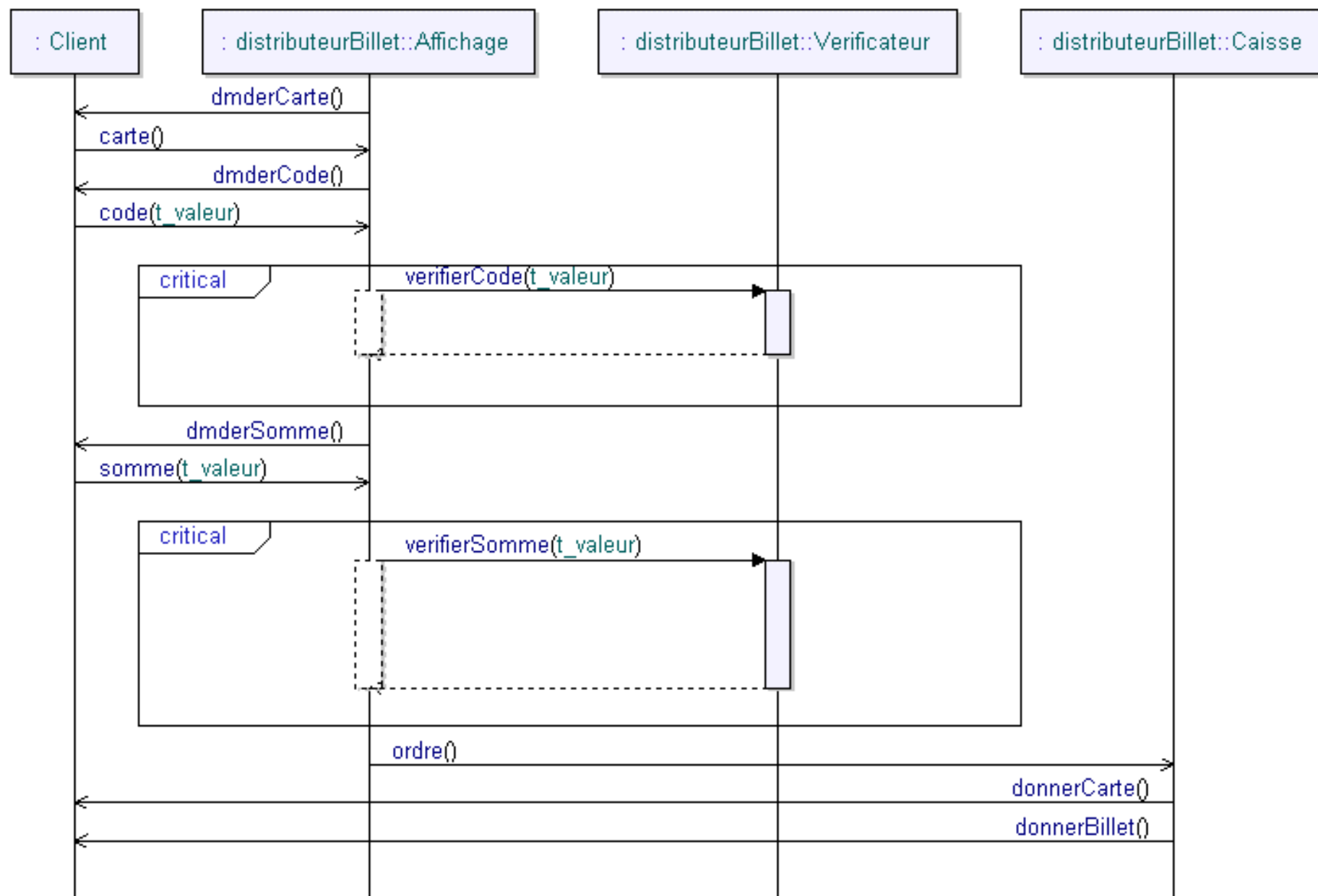
Les fragments combinés



Les fragments combinés

❑ Opérateur **Critical**

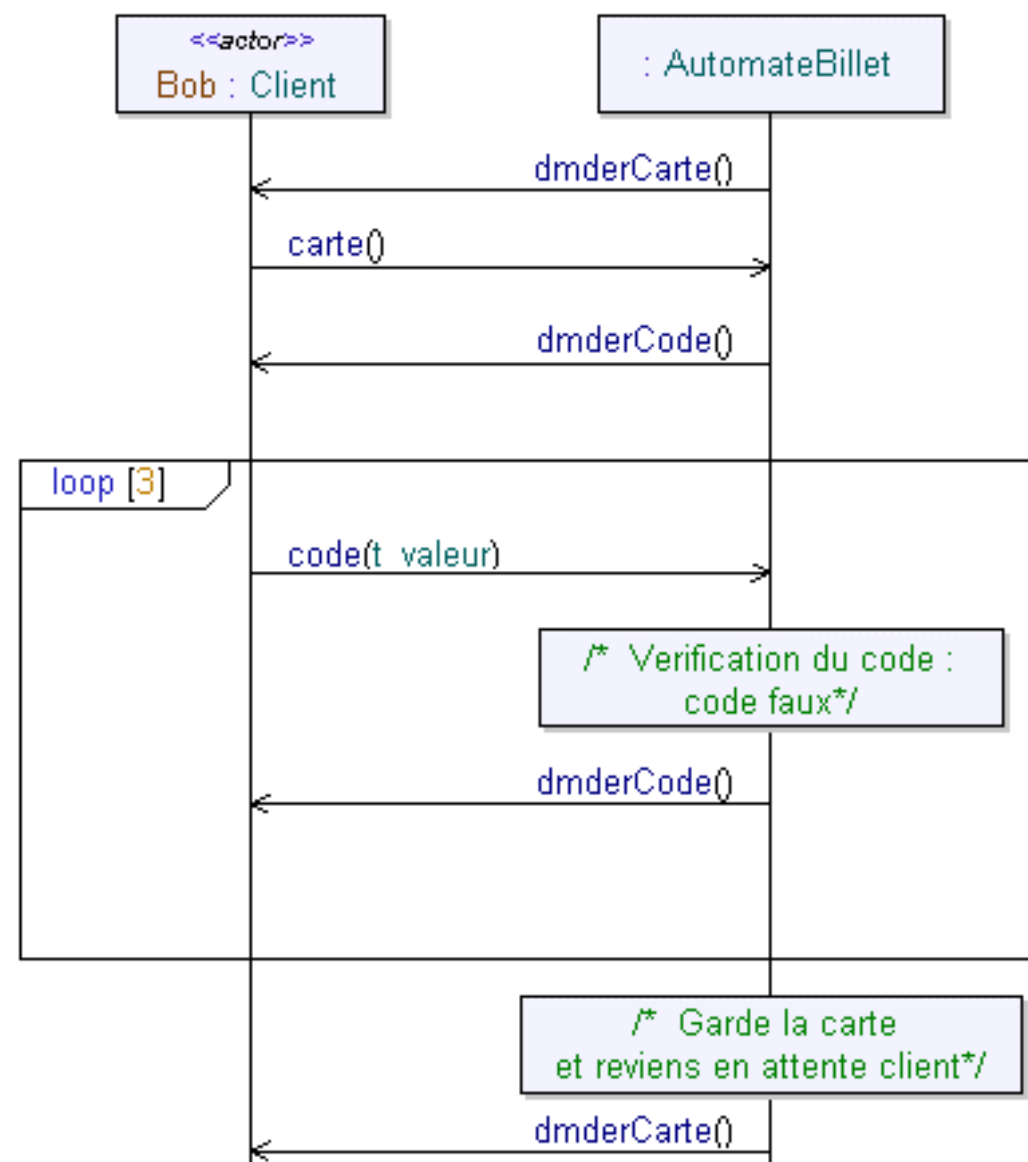
- désigne une section critique.
- les interactions décrites dans cet opérateur ne peuvent pas être interrompues par d'autres interactions décrites dans le diagramme.
- opérateur impose un traitement atomique des interactions qu'il contient.
- *Exemple*
 - ❑ *On ne souhaite pas que l'utilisateur puisse obtenir des billets avec un code erroné et une somme demandée incorrecte.*



Les fragments combinés

❑ Opérateur boucle : **loop**

- Ensemble d'interactions qui s'exécutent **en boucle**.
- une contrainte appelée **garde** indique le nombre de répétitions (minimum et maximum) ou bien une condition booléenne à respecter.
- *Exemple :*
 - ❑ *Le diagramme de séquences indique que lorsque l'utilisateur se trompe trois fois de code, la carte est gardée et le distributeur se remet en mode d'attente d'une carte.*



Les fragments combinés

❑ **Notion d'UML2**

- Plusieurs Scénarii
- Avant uniquement un seul scénarii

❑ **Attention**

- Ne pas représenter des scénarii trop différents
- Il vaut mieux faire plusieurs diagrammes de séquences

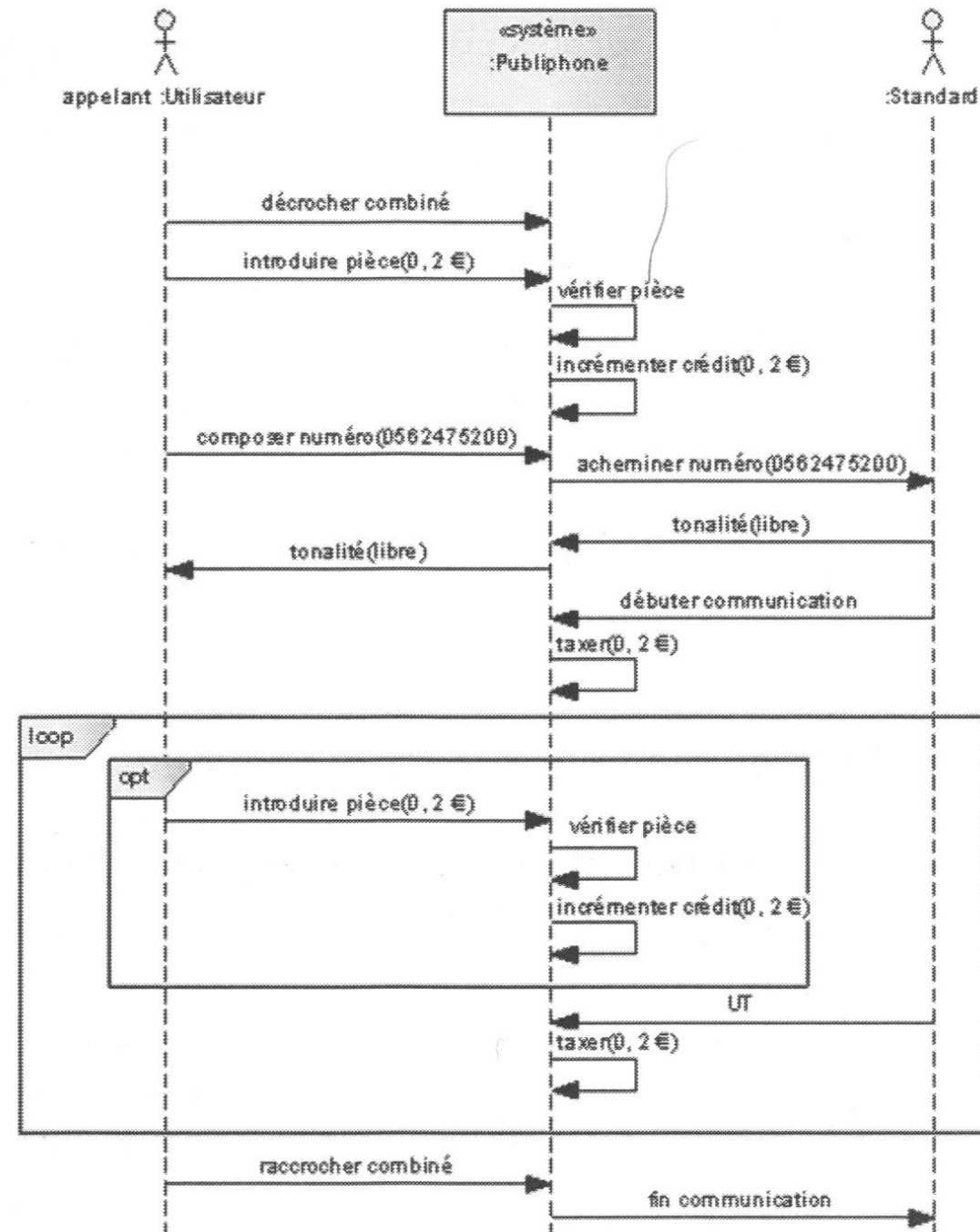
Exercice : Etude du cas d'un publiphone à pièces

- ❑ Cette étude de cas concerne un système simplifié de Publiphone à pièces.
 1. Le prix minimal d'une communication inter urbaine est de 0,2 euros.
 2. Après l'introduction de la monnaie, l'utilisateur a 2 minutes pour composer son numéro (ce délai est décompté par le standard (ou réseau)).
 3. La ligne peut être libre ou occupée.
 4. Le correspondant peut raccrocher le premier.
 5. Le Publiphone consomme de l'argent dès que l'appelé décroche et à chaque unité de temps (UT) générée par le standard.
 6. On peut ajouter des pièces à tout moment.
 7. Lors du raccrochage, le solde de monnaie est rendu.
- ❑ À partir de ces phrases, nous allons progressivement :
 - Identifier les acteurs et les cas d'utilisation.
 - Construire un diagramme de séquence



Diagramme de séquences

□ Sans la conversation



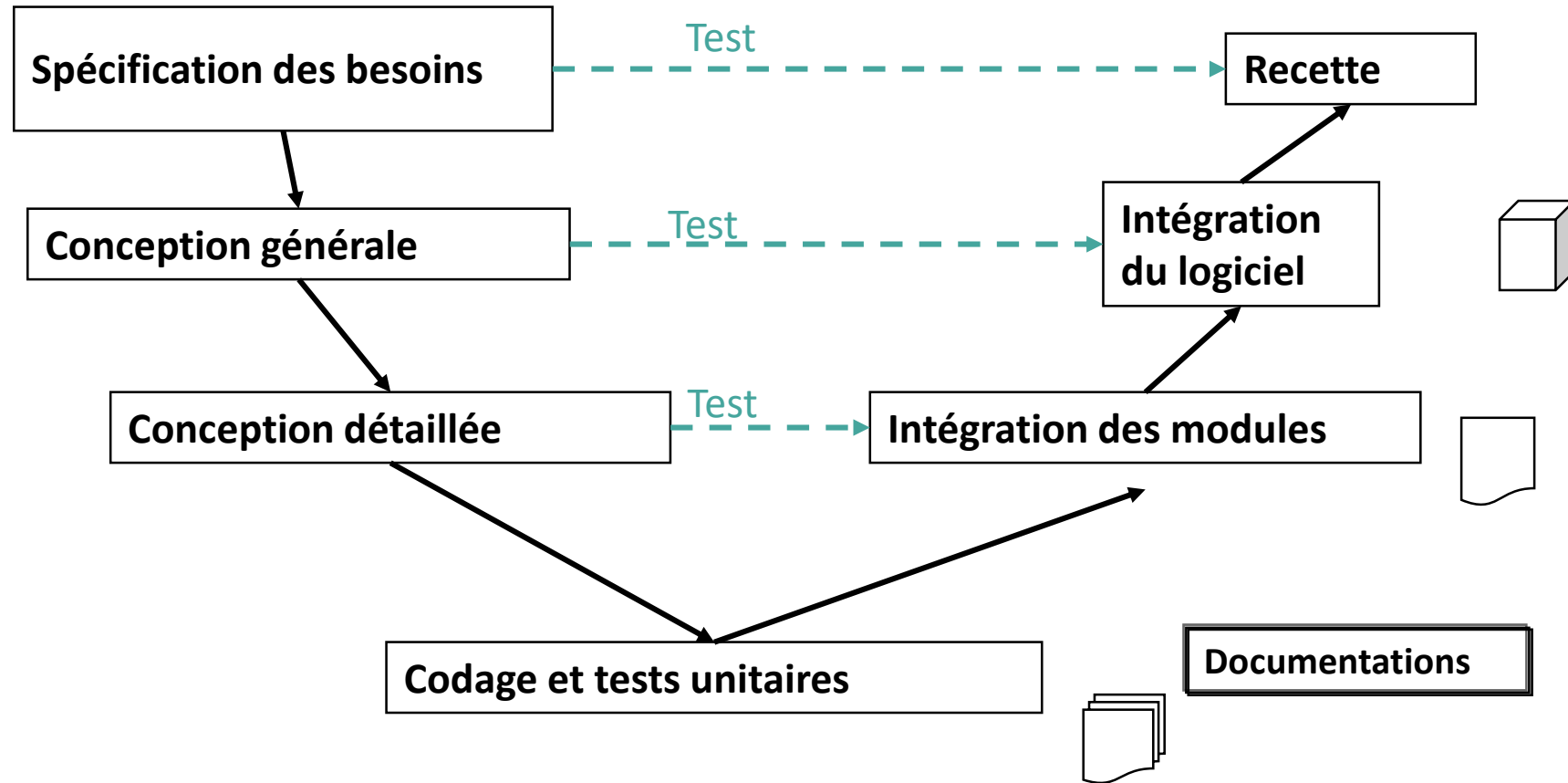
Conclusion UML

❑ Une bonne modélisation

- demande du temps
- est itérative
 - ❑ Modèle simple dans un premier temps
 - ❑ Ajout de détails (fonctionnels et techniques par la suite)
- fait gagner du temps
 - ❑ Moins d'erreurs
 - ❑ Meilleure compréhension du système (amélioration de la collaboration avec le maître d'œuvre)

❑ Il existe un grand nombre de modèles

Rappel : Cycle de vie en V



Diagrammes UML et Cycle en V

□ Diagramme de cas d'utilisation

- Très utile en phase de spécification de besoin

□ Diagramme de classes

- Très utile en phase de conception générale
 - Description simple des classes (pas de méthodes, de typage ni de cardinalité)
- Très utile en phase de conception détaillée
 - Description détaillée

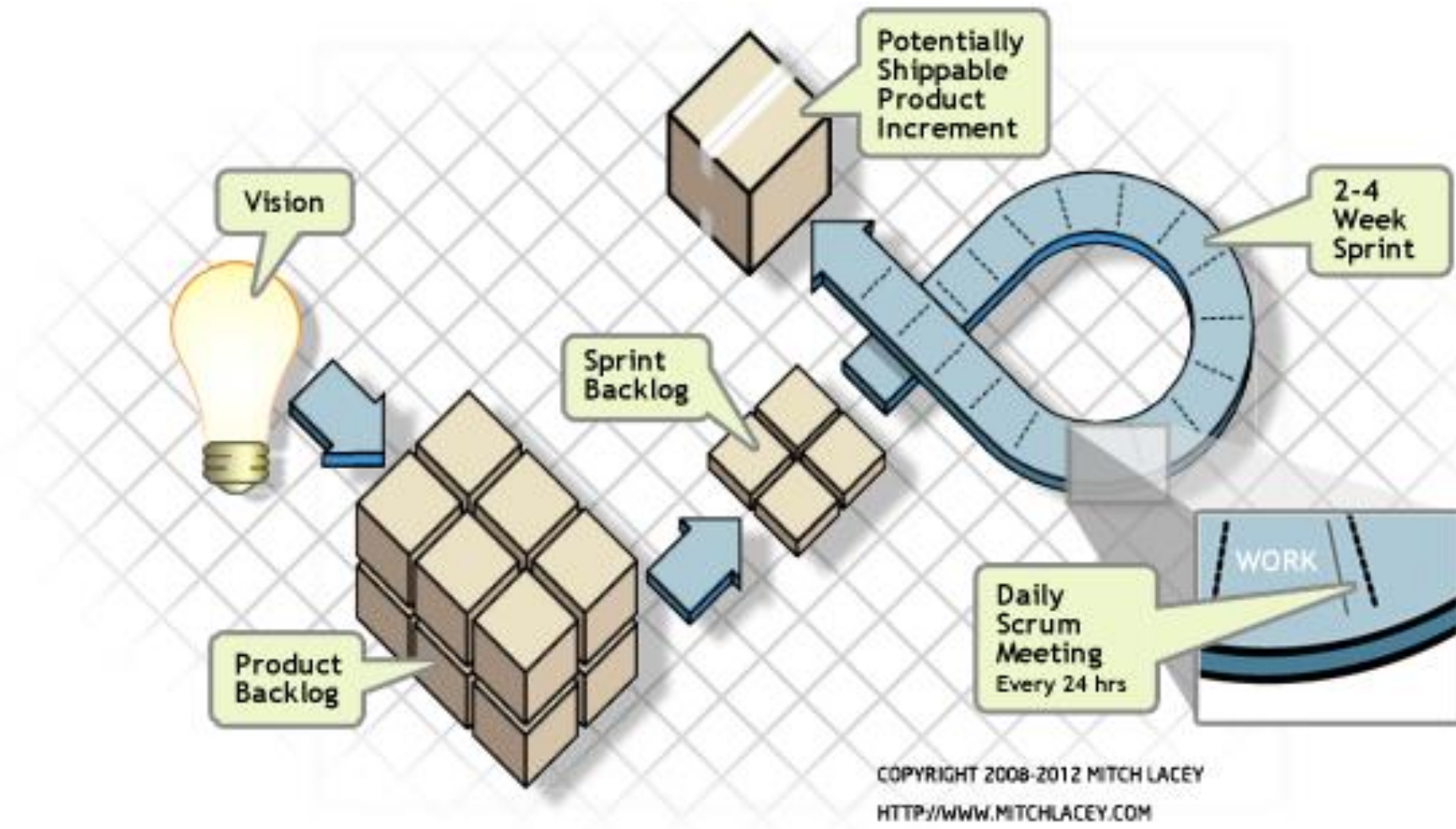
□ Diagramme d'activités

- Très utile en phase de conception générale
 - Pour décrire les flux d'informations et les interactions entre le(s) système(s) et le(s) utilisateur(s)

□ Diagramme de séquences

- Très utile en phase de conception détaillée
 - Description détaillée des flux sur des scénarii très utilisé ou des cas complexes à mettre en place

Rappel : Méthode SCRUM (AGILE)



Diagrammes UML et Scrum

□ Scrum

■ Sprint 0 :

- Diagramme de cas d'utilisation

■ Autres sprints

- Diagrammes de classes

- Raffinage uniquement des classes impliquées par le Sprint

- Diagramme de séquences

- Pour les actions du sprint
 - Cas classiques
 - Cas complexes

Diagrammes UML et modèle Agile

❑ Avec Agile

- Plus de concepteur qui décrit TOUT puis un programmeur qui code TOUT

❑ Avant le premier sprint

- Faire des modélisation simple du domaine (diagramme de cas)

❑ Au début de chaque sprint

- séance de modélisation en groupe de ce qui est nécessaire pour ce sprint
- coller les modèles résultant sur les murs
- Repartir des anciens modèles et ajouter les nouveaux concepts (cas d'utilisation, classes, activité)