

UML – Unified Modeling Language

3/4 : diagrammes dynamiques et d'interaction

Yannick Prié

UFR Informatique – Université Claude Bernard Lyon 1

M1 MIAGE – SIMA / M1 Informatique MIF17

2008-2009

Objectifs de ce cours

- Apprendre la syntaxe et la sémantique des diagrammes dynamiques et d'interaction les plus importants
- Améliorer au passage la compréhension de différents principes objets

Plan

- **Diagrammes de cas d'utilisation**
- Diagrammes d'interaction
 - diagrammes de séquences
 - diagrammes de communication
- Diagrammes d'activité
- Diagrammes de machines d'états
- Autres diagrammes UML
- Autres diagrammes non UML

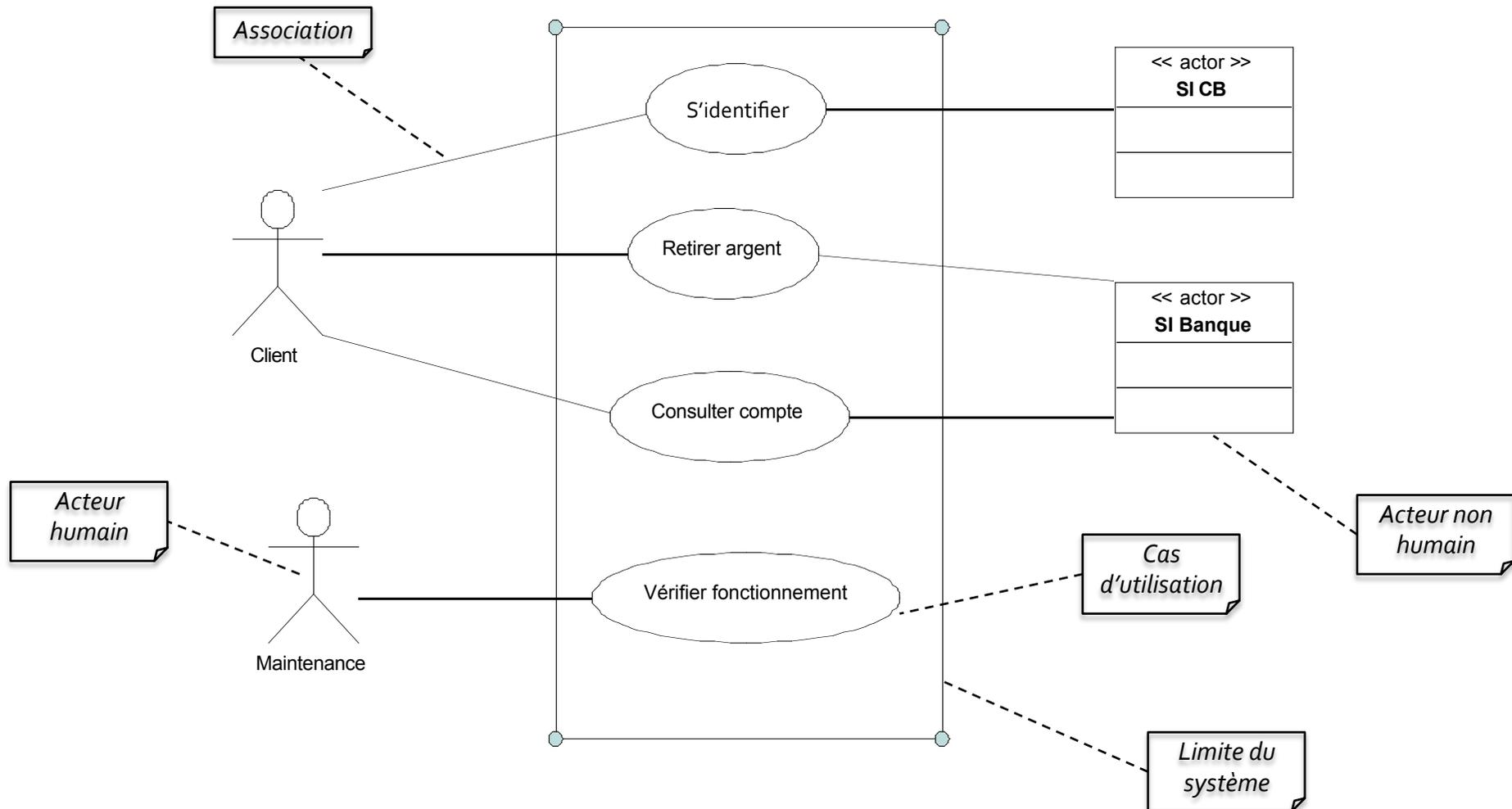
Cas d'utilisation : avertissement

- Dans ce cours sur UML
 - présentation « standard » des cas d'utilisation et des diagrammes de cas d'utilisation
 - ce qu'on trouve en général, y compris dans la norme
- Dans le cours spécifique « Cas d'utilisation »
 - présentation plus ciblée et plus précise
 - d'après le livre de Alistair Cockburn qui fait référence, au delà de la norme
 - à utiliser en TD

Cas d'utilisation

- Technique pour capturer les exigences fonctionnelles d'un système
 - déterminer ses limites
 - déterminer ce qu'il devra faire
 - mais pas comment il devra le faire
 - point de vue de l'utilisateur
- Pour cela
 - déterminer les acteurs qui interagissent avec le système
 - rôles
 - déterminer les grandes catégories d'utilisation
 - cas d'utilisation
 - décrire textuellement des interactions
 - scénarios

Diagramme de cas d'utilisation



Petit exercice à faire en classe

- Quels sont les acteurs et les cas d'utilisation d'un système d'information pour l'UFR informatique ?

L'importance vitale des utilisateurs

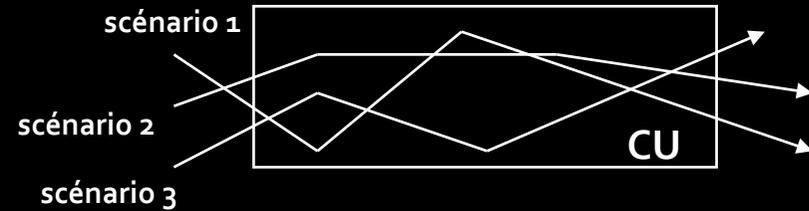
- Les CU sont issus du dialogue entre concepteurs (informaticiens) et futurs utilisateurs (non-informaticiens) pour
 - passer du flou du cahier des charges à des fonctionnalités exprimées dans le langage du domaine, donc des utilisateurs
 - exprimer complètement les besoins, tout au long du processus de conception de système d'information
- Attention : les CU doivent être validés par les futur utilisateurs
 - les diagrammes de CU ne disent pas grand chose, sont difficilement lisibles → **peu importants**
 - les descriptions textuelles des scénarios sont lisibles → **très importantes**
 - racontent l'histoire du futur utilisateur avec le futur système

Acteur

Client : personne qui se connecte au distributeur bancaire à l'aide de sa carte.
Peut avoir ou non un compte dans la banque qui possède le distributeur.

- Entité (humain ou machine) située hors du système qui permet d'en déterminer les limites
- Un acteur joue un rôle par rapport au système
 - soit déclenche un stimulus entraînant une réaction du système
 - soit est sollicité par le système au cours d'un scénario
- Un acteur est décrit précisément en quelques lignes
- 4 grandes catégories
 - acteurs principaux (fonctions principales du système)
 - acteurs secondaires (administration / maintenance)
 - matériel externe
 - autres systèmes

Scénario



- Séquence particulière d'étape dans la réalisation d'un CU
- Séquence particulière de messages dans le CU pendant une interaction particulière
 - « chemin » dans le cas d'utilisation
- Tous les scénarios d'un CU sont issus du même acteur et ont le même objectif
- Description du CU
 - ensemble de scénarios couvrant le CU
 - documents avec flot d'événements
 - détaille ce qui se passe entre utilisateur et le système quand le CU est exécuté
 - flot nominal des événements (80 %)
 - flots d'événements alternatifs
 - flots d'exceptions (terminaison incorrecte)
 - serviront de base pour les jeux d'essais

Cas d'utilisation

- Ensemble de séquences d'actions réalisées par le système, produisant un résultat observable pour un acteur particulier
 - ex. s'identifier, retirer du liquide
- Un cas d'utilisation
 - définit un ensemble de scénarios d'exécution impliquant le même acteur (déclencheur) avec le même objectif utilisateur
 - recense les informations échangées et les étapes dans la manière d'utiliser le système, les différents points d'extension et tous les cas d'erreur

Documentation d'un CU

- Fiche textuelle
 - ensemble de champs de description
 - nom, préconditions...
 - lisible et informelle
 - français simple, phrases descriptives
 - pas trop long (personne ne lit 10 pages)
 - décrivant
 - un scénario nominal
 - suite d'étapes avec objectifs de l'acteur bien identifiés et menés à bien
 - des points d'extension et étapes d'extensions
 - des points d'échec
 - des liens vers d'autres scénarios s'il y a trop d'étapes

Niveaux pour les cas d'utilisation

- Plusieurs niveaux de description

- abrégé, informel, détaillé

- Plusieurs niveaux d'objectif

- objectif stratégique (cerf-volant au dessus du niveau de la mer)

- fonction du SI dans organisation
- on se rapproche des processus métier



- objectif utilisateur (surface de la mer)

- fonction du SI pour l'utilisateur



- objectif sous-fonction (poisson en dessous du niveau de la mer)

- fonction interne au système, utile pour l'informaticien



- Plusieurs portées de conception

- organisation (boîte blanche ou noire)



- système (boîte blanche ou noire)



- composant



Description d'un CU

- Nom
- Contexte d'utilisation
- Portée
- Niveau
- Acteur principal
- Intervenants et intérêts
- Préconditions
- Garanties minimales
- Garanties en cas de succès
- Déclencheur
- Scénario nominal
 - étapes
- Extensions
 - étapes d'extension
- Variantes de techno-logie ou de données
- Informations connexes

Exemple scénarios pour CU



CU : Retirer de l'argent

Portée : système DAB

Niveau : objectif utilisateur

Acteur principal : Client

Intervenants et intérêts : Banque, Client

Préconditions : compte approvisionné

Garanties minimales : rien ne se passe

Garanties en cas de succès : de l'argent est retiré, le compte est débité de la même somme

...

Exemple scénarios pour CU



...

Scénario nominal :

1. Le Client introduit sa carte dans le lecteur.
2. Le DAB décrypte l'identifiant de la banque, le numéro de compte et le code secret de la carte, valide de la banque et le numéro de compte auprès du système principal.
3. Le client saisit son code secret. Le DAB valide par rapport au code secret crypté lu sur la carte.
4. Le client sélectionne retrait, et un montant multiple de 10 € (min 20 €)
5. Le DAB soumet au principal système de la banque le compte client et le montant demandé, et reçoit en retour une confirmation et le nouveau solde du compte
6. Le DAB délivre la carte, l'argent et un reçu montrant le nouveau solde
7. Le DAB consigne la transaction

...

Exemple scénarios pour CU



...

Extensions :

*a. Panne générale.

*a1. Le DAB annule la transaction, signale l'annulation, et rend la carte.

2a. Carte volée.

2a1. Le DAB confisque la carte volée

Inclusion autre scénario

4a. Plus de billets de 10 €

4a1. Le DAB arrondit la somme demandée à un multiple de 20 €.

4a2. Le Client valide la nouvelle somme demandée.

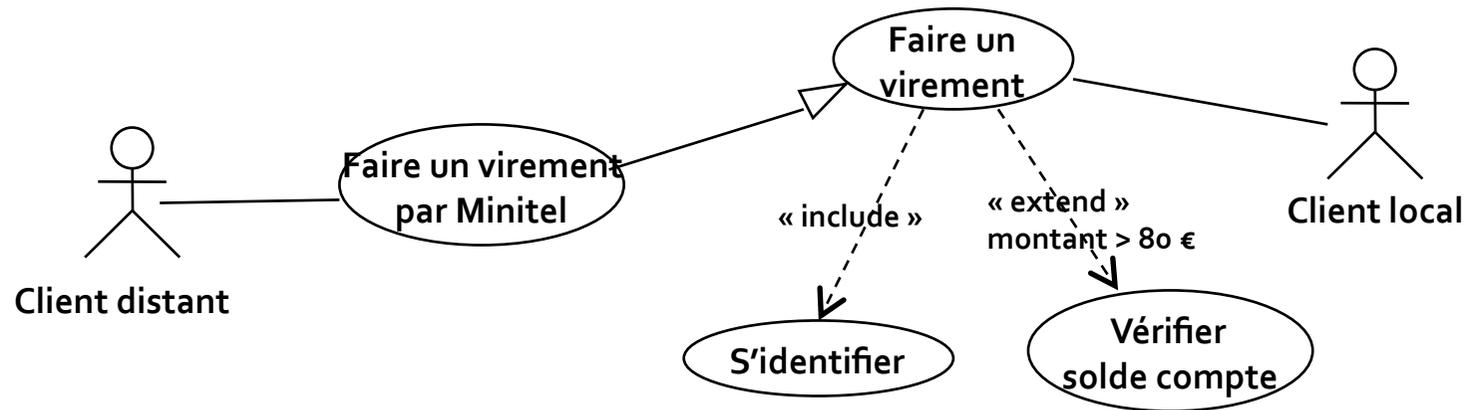
5a. Solde insuffisant.

5a1. Le DAB signale que la somme demandée est trop élevée et rend la carte.

Complément de description de CU

- Tout ce qui permet de mieux expliquer
 - modèle du domaine
 - diagrammes de séquence système,
 - diagramme d'activité, de machines d'états
 - dessin ou maquette d'interface
 - ...

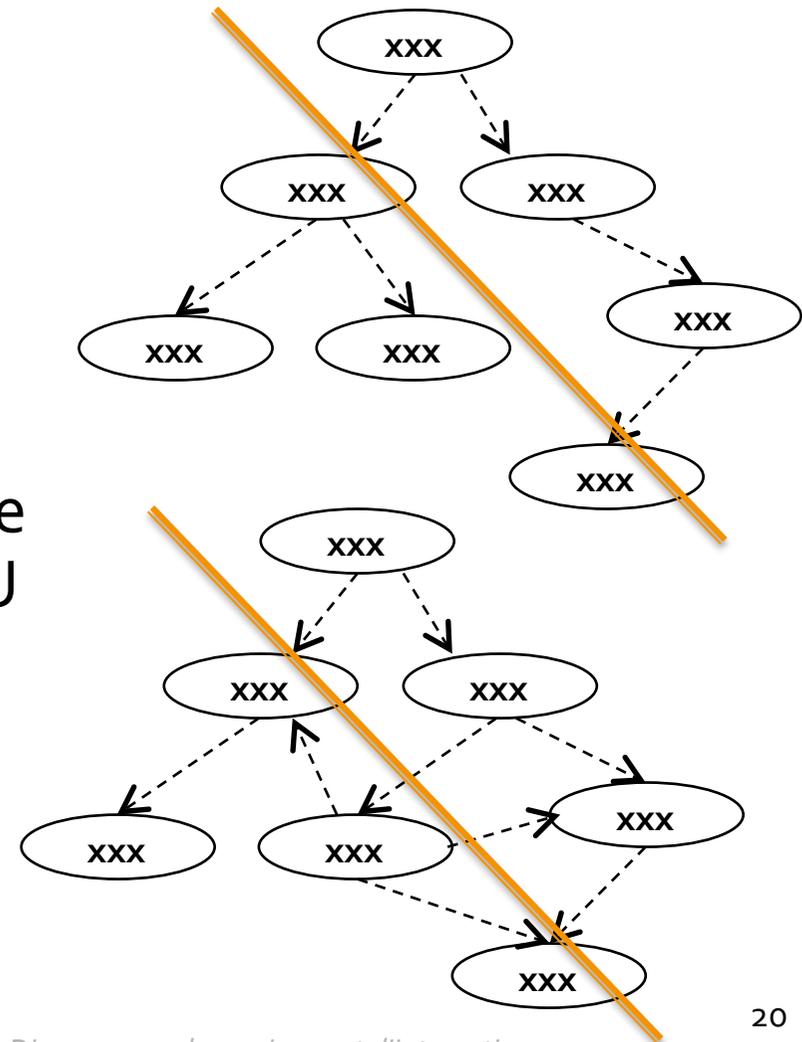
UML : relations entre CU



- « include »
 - la réalisation d'un CU nécessite la réalisation d'un autre, sans condition, à un point d'extension (**le seul important**)
- « extend »
 - entre deux instances de CU : le comportement de CU₁ peut être complété par le comportement de CU₂ (option avec condition et point d'extension)
 - conseil : **ne pas utiliser**, ou seulement si on ne peut toucher à CU₁
- « generalize »
 - héritage. (conseil : **ne pas utiliser**)
- Bref
 - le diagramme de CU est une bonne **table des matières**, pas plus

Quelques conseils

- Pas plus de 20 CU pour un système
- Pas de diagramme de CU hormis pour une table des matières
 - pas de définition fonctionnelle en utilisant les relations de CU
 - pas de description illisible du fonctionnement du système
- Retour sur les CU dans un cours dédié, un TD dédié

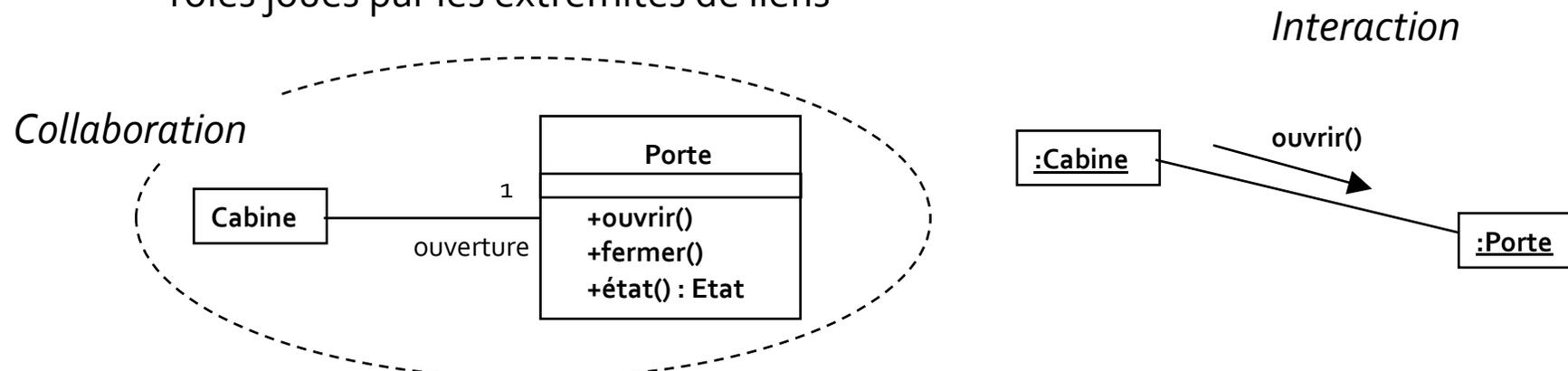


Plan

- Diagrammes de cas d'utilisation
- **Diagrammes d'interaction**
 - diagrammes de séquences
 - diagrammes de communication
- Diagrammes d'activité
- Diagrammes de machines d'états
- Autres diagrammes UML
- Autres diagrammes non UML

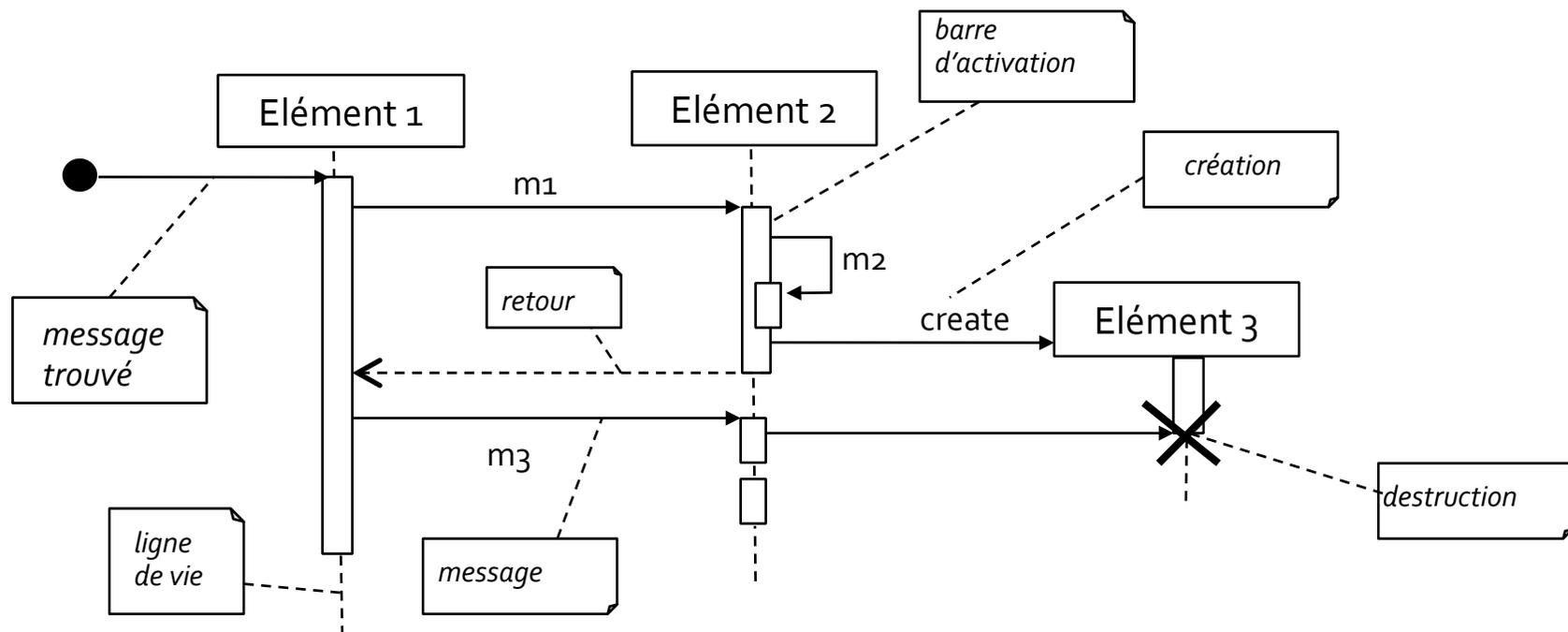
Collaborations et interactions

- Collaboration
 - ensemble de rôles joués par des classes fournissant un contexte d'interaction
- Interaction
 - communication entre instances des éléments d'une collaboration
 - ensemble partiellement ordonné de messages
 - plusieurs interactions possibles pour une même collaboration
- Éléments d'une interaction
 - participants (UML1 : objets, UML2 : souvent objets)
 - liens (supports de messages)
 - messages (déclenchant des opérations)
 - rôles joués par les extrémités de liens



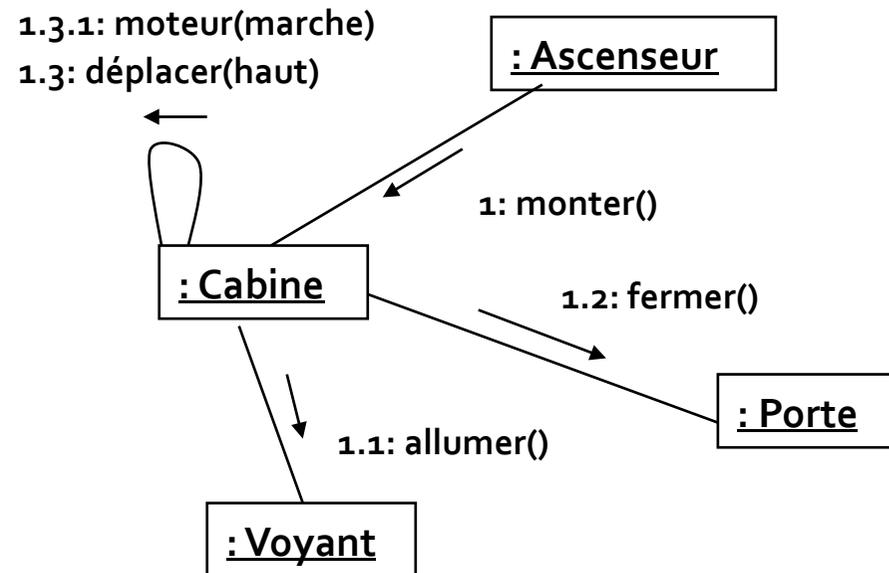
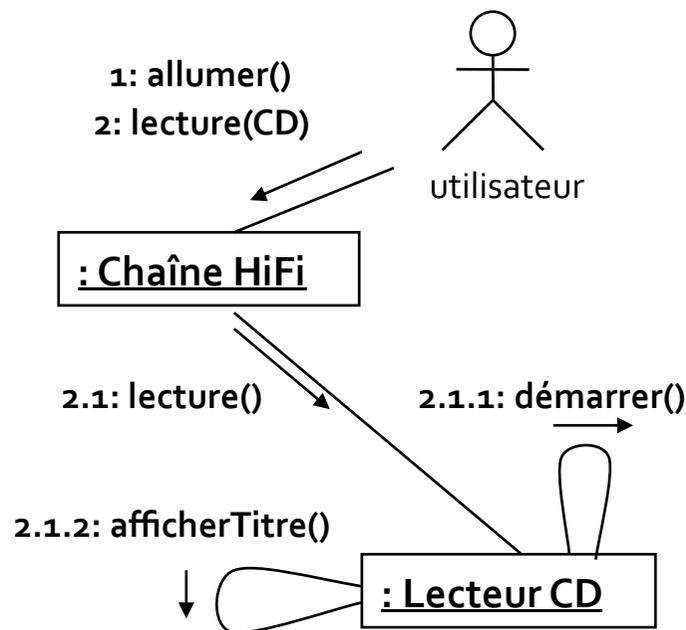
Diagrammes de séquences

- Interactions entre éléments dans une séquence temporelle
 - aspect chronologique ne rendant pas compte explicitement du contexte
 - permet de bien montrer qui fait quoi dans une interaction
- Description de scénarios typiques et des exceptions



Diagrammes de communication (UML1 : diagrammes de collaboration)

- Diagramme d'objets rendant compte de la dynamique
 - structure spatiale permet la collaboration d'objets
 - dimension temporelle : ordre des messages
 - numérotation pointée



Petit exercice à faire en classe

- Dessiner un diagramme de communication impliquant le passage de la balle entre deux tortues d'équipes différentes.

Utilisation des interactions

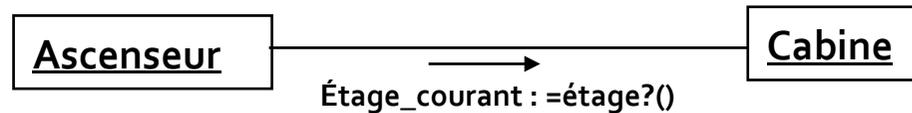
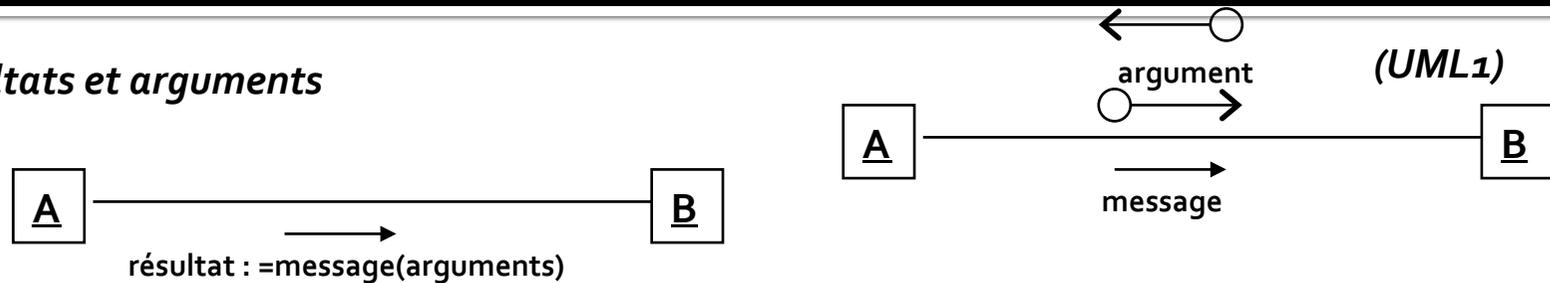
- Etudier/spécifier le comportement
 - du système dans sa globalité au sein d'un cas d'utilisation
 - se concentrer sur les événements du système considéré comme boîte noire
 - diagramme de séquence système (exemple plus loin)
 - de plusieurs objets au sein d'un cas d'utilisation
 - réalisations de CU comme des interactions dans une société d'objets
 - conseil : concevoir diagrammes de classes et d'interaction en même temps
- Illustrer/étudier un fonctionnement
 - diagramme qui traverse les couches : de l'IHM aux données
 - rétro-ingénierie

Messages

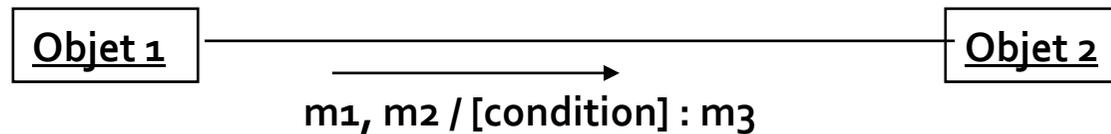
- Matérialisation d'une communication avec transmission d'information entre
 - émetteur (source)
 - récepteur (destination)
- Un message déclenche
 - une opération,
 - l'émission d'un signal
 - la création/destruction d'un objet
- Deux types principaux
 - appel de procédure ou flot de contrôle emboîté (appel standard de fonction)
 - déplacer()
 - flot de contrôle asynchrone (threads)
 - démarrer()
 - autres : à plat, dérobant (réception si attente), minuté (message actif pendant Δt)

Messages dans les diagrammes de communication

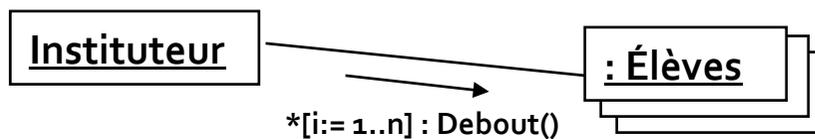
Résultats et arguments



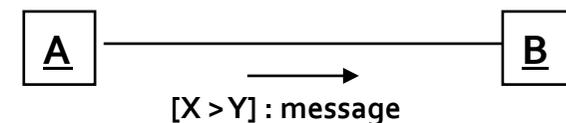
Synchronisation



Itérateurs



Garde



Messages dans les diagrammes de séquences

- Notation résultat = message(arguments)

- Echange de messages

- flèches d'appel standard

- blocage de l'émetteur en attendant la réponse

- flèche d'appel asynchrone

- pas d'attente du retour, poursuite de la tâche

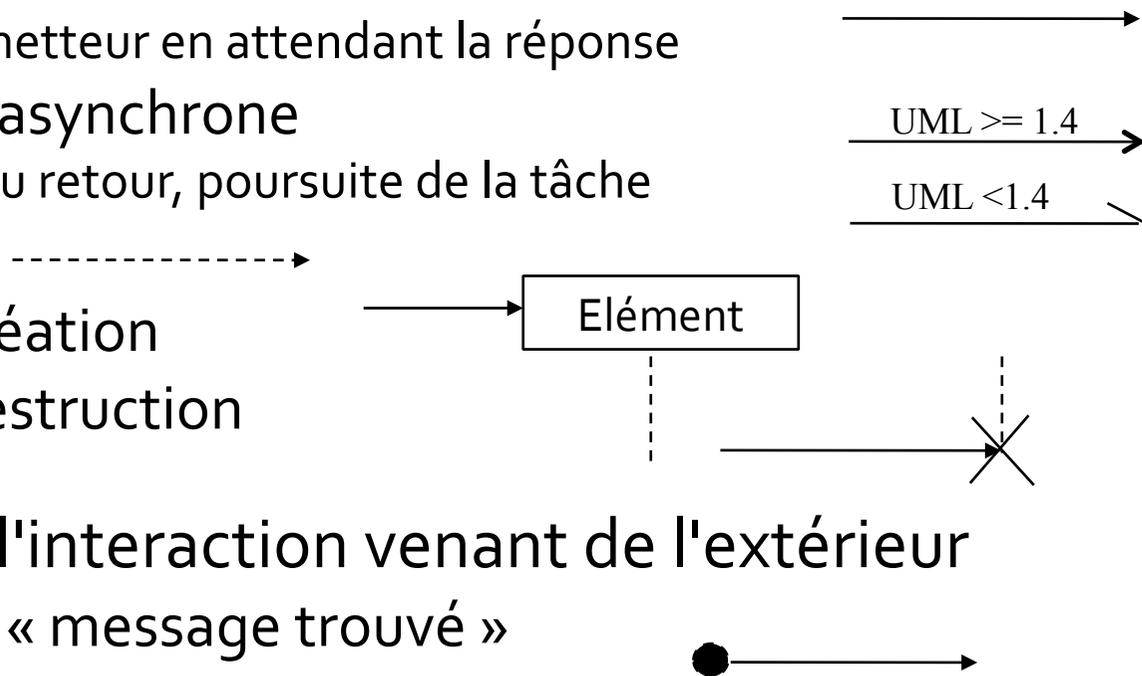
- Retour

- Message de création

- Message de destruction

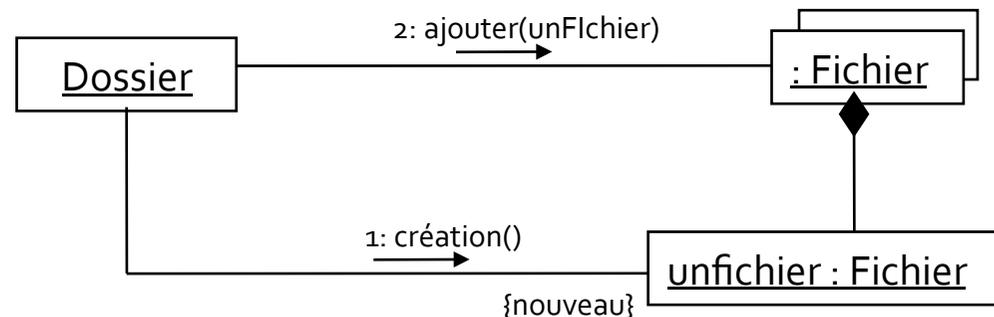
- Lancement de l'interaction venant de l'extérieur

- 1er message = « message trouvé »

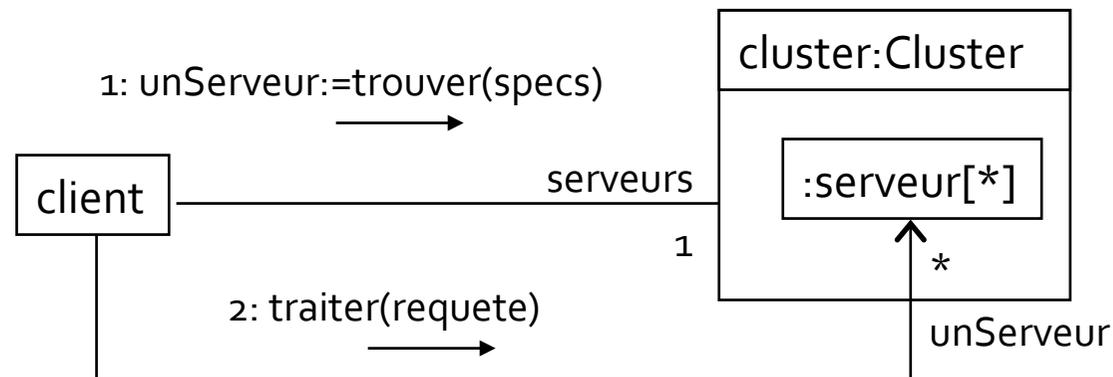


Gestion de collections

- Créer et ajouter (UML1 : multi-objet)



- Récupérer et utiliser (UML2 : structure composite)

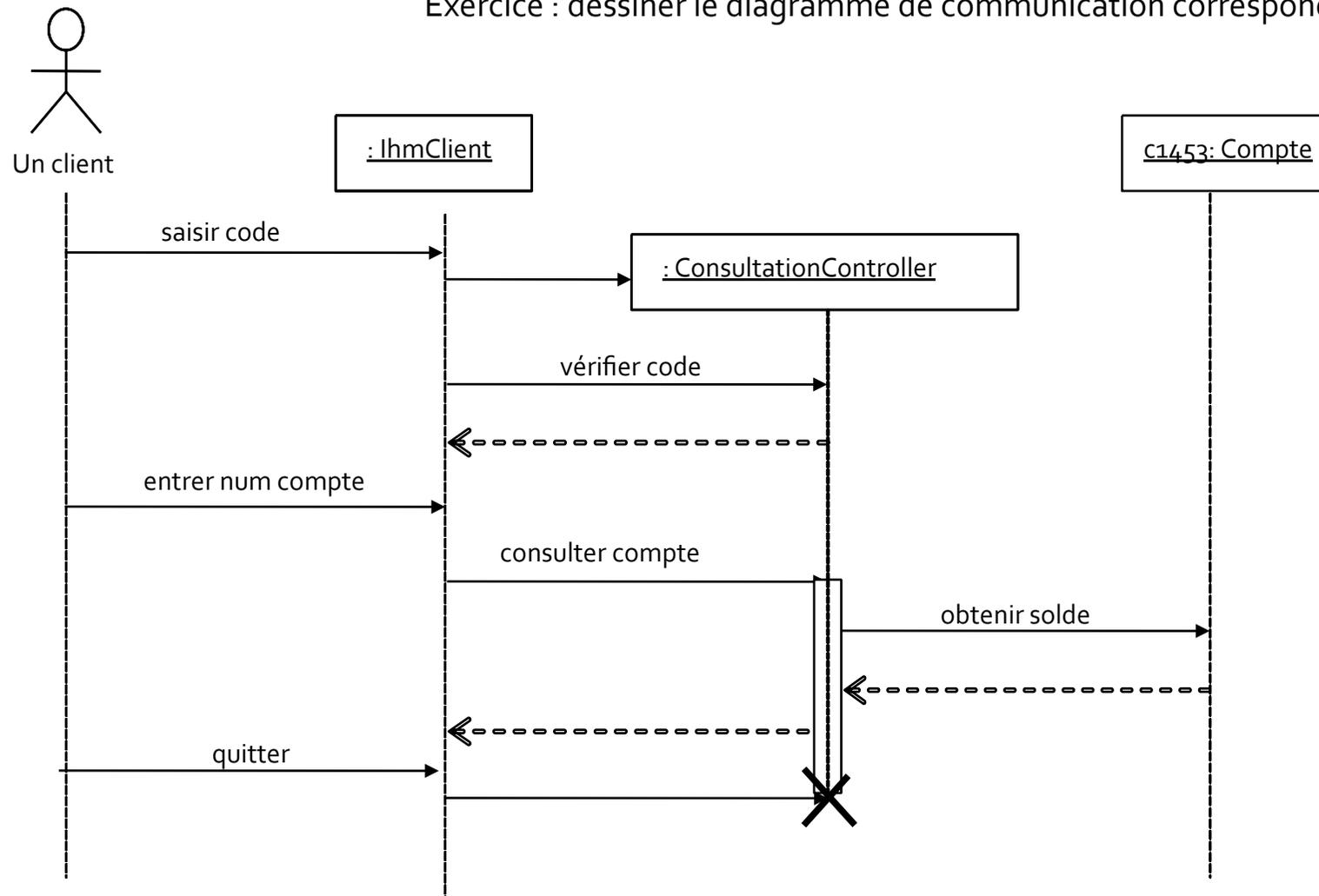


Diagrammes de séquences dans UML2

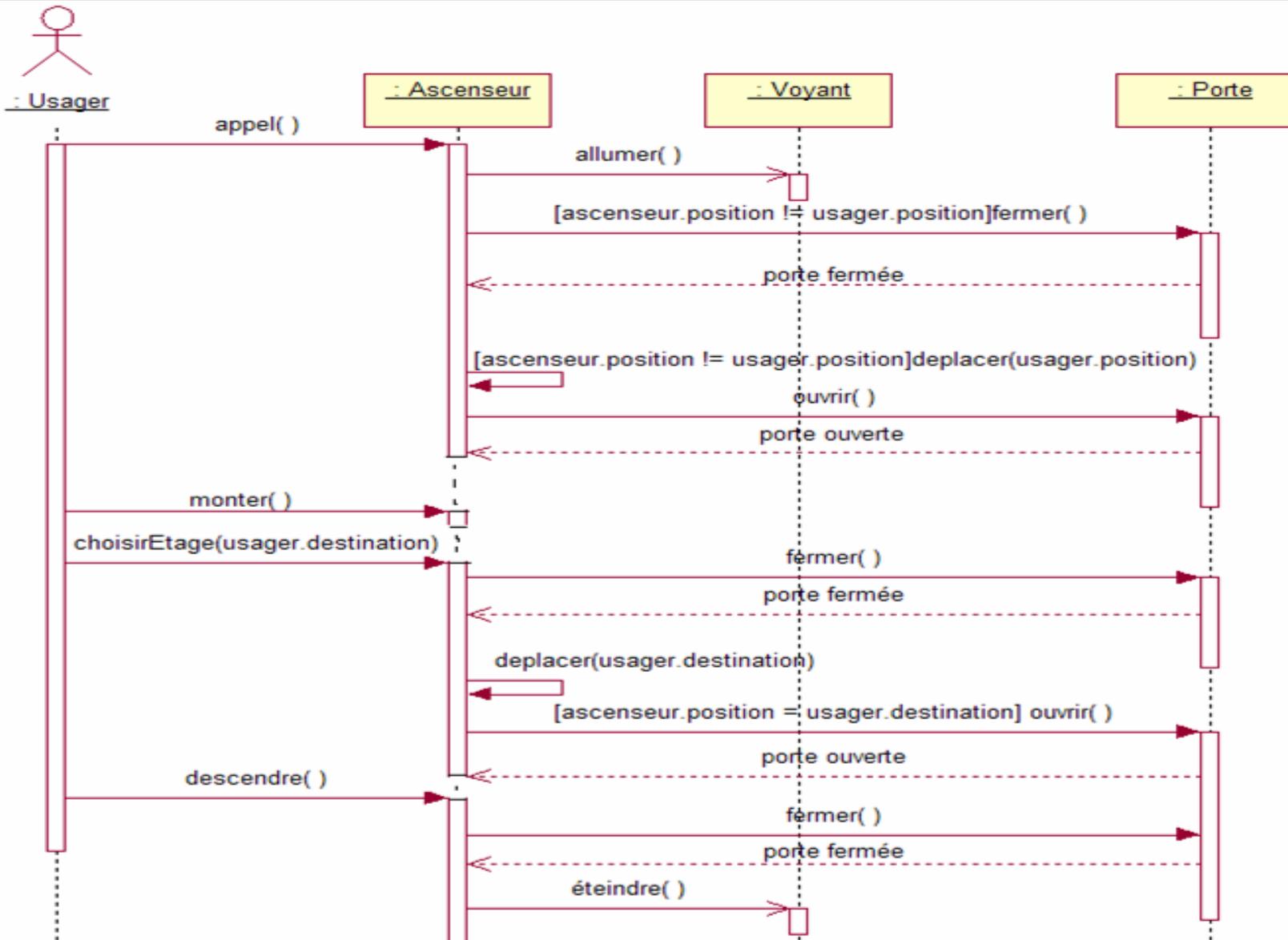
- En plus de objets participants (UML1), on ajoute
 - interfaces
 - pour spécifier quelle interface participe à l'interaction (classe peu importante)
 - classes
 - pour utiliser une méthode de classe
- Représentation polymorphisme / classe abstraite

Equivalence entre diagrammes de séquence et de communication

Exercice : dessiner le diagramme de communication correspondant



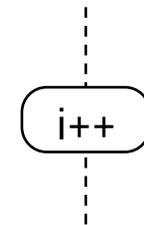
Autre exemple



Cadre d'interaction



- Cadre nommé par un opérateur qui entoure un fragment critique du DS
 - alt
 - fragment alternatif, conditions dans les gardes
 - loop
 - fragment à répéter tant que la condition de garde est vrai
 - notion de boîte d'action avec itérateur
 - opt
 - fragment optionnel exécuté si la garde est vraie
 - par
 - fragments qui s'exécutent en parallèle
 - region
 - region critique dans laquelle un seul thread doit s'exécuter
 - ref
 - passage à un autre diagramme de séquence
- Attention
 - ne pas représenter des algorithmes : trop compliqué

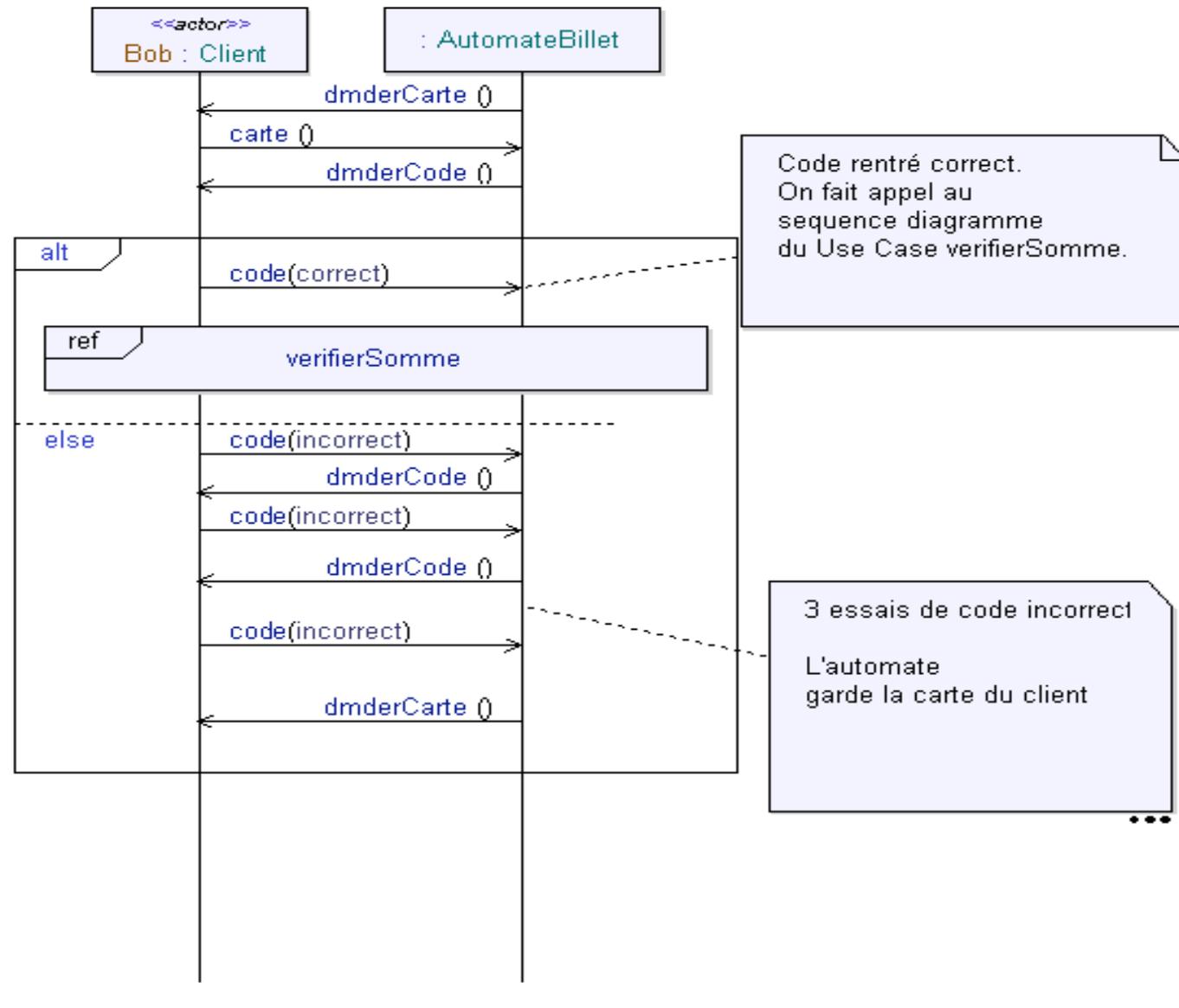


alt

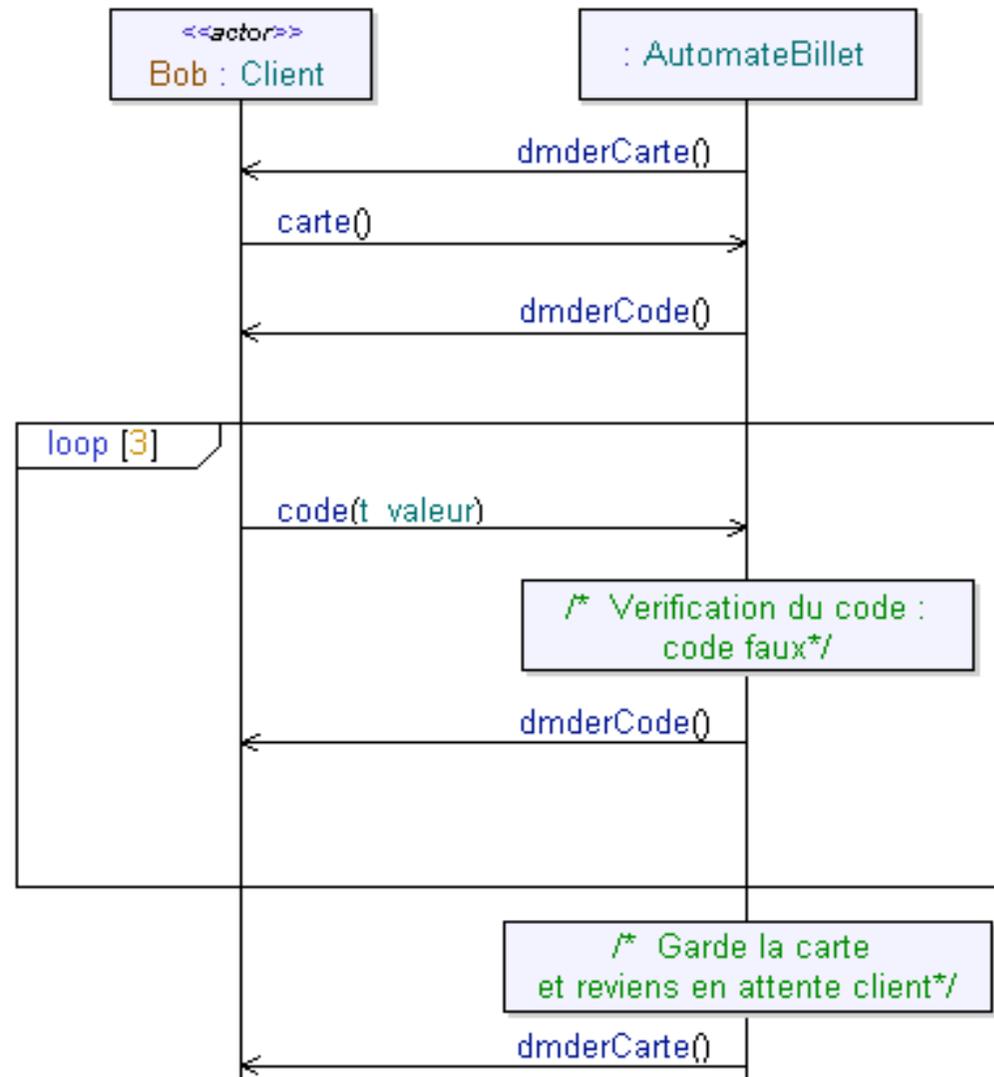
sd identifierUtilisateur

interaction identifierUtilisateur

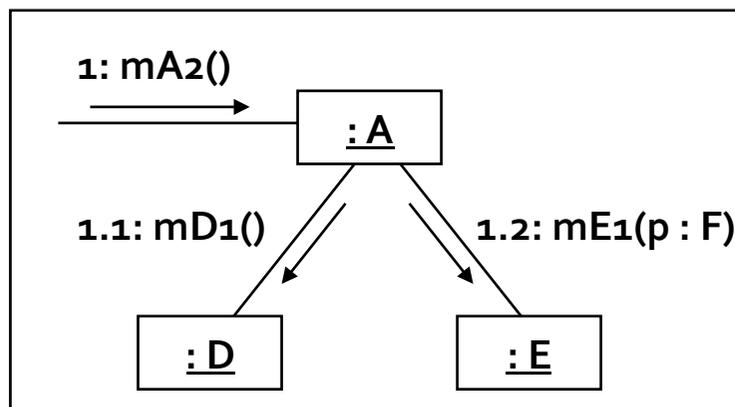
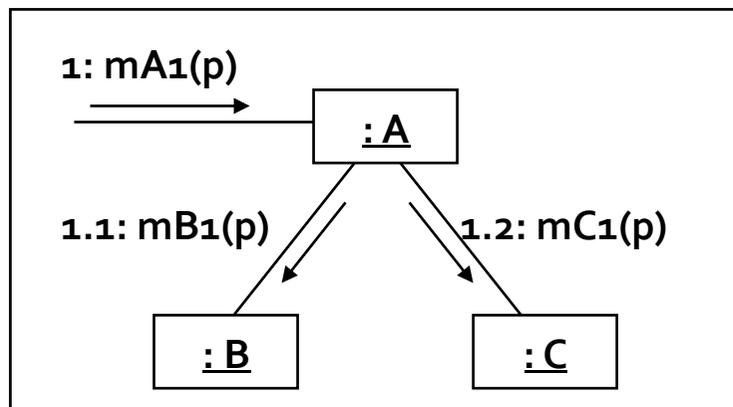
{1/1}



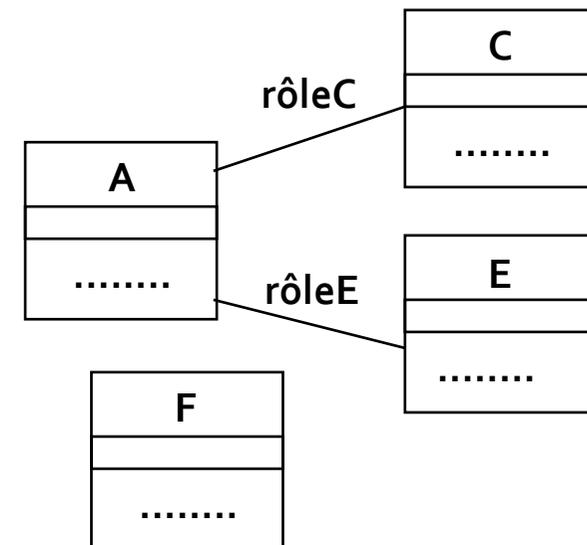
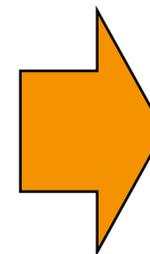
loop



Déduire structure et responsabilité des diagrammes d'interaction

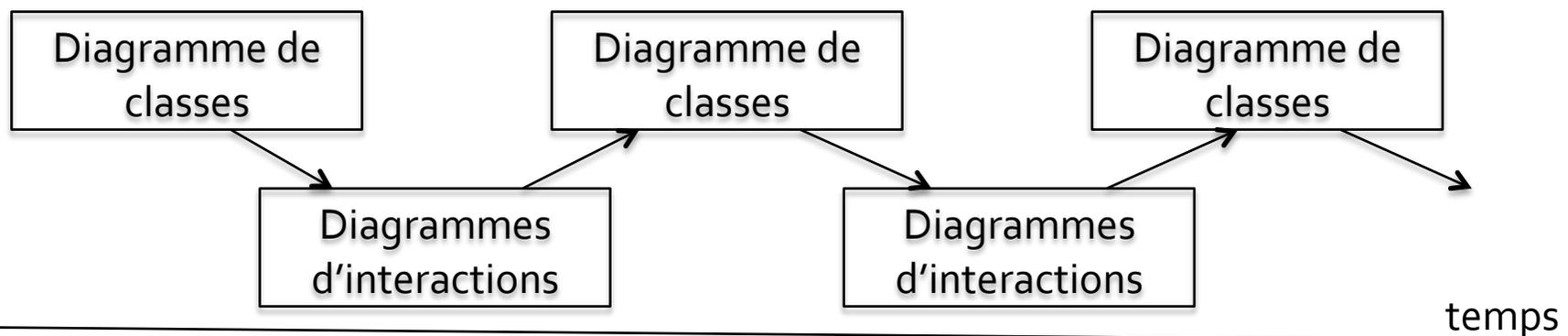


- Liens
 - associations
- Messages
 - opérations
 - dépendance



Co-conception des classes et des interactions

- Les objets utilisés dans les interactions pour réaliser les scénarios proviennent
 - des classes déjà décrites dans le diagramme de classes
 - des besoins nouveaux en objets pour l'interaction spécifique
- A partir des diagrammes d'interaction, on complète le diagramme de classes
 - précisions (attribut, méthodes)
 - nouvelles classes
- etc.
- On essaye de réaliser tous les scénarios en convergeant vers un diagramme de classes stables

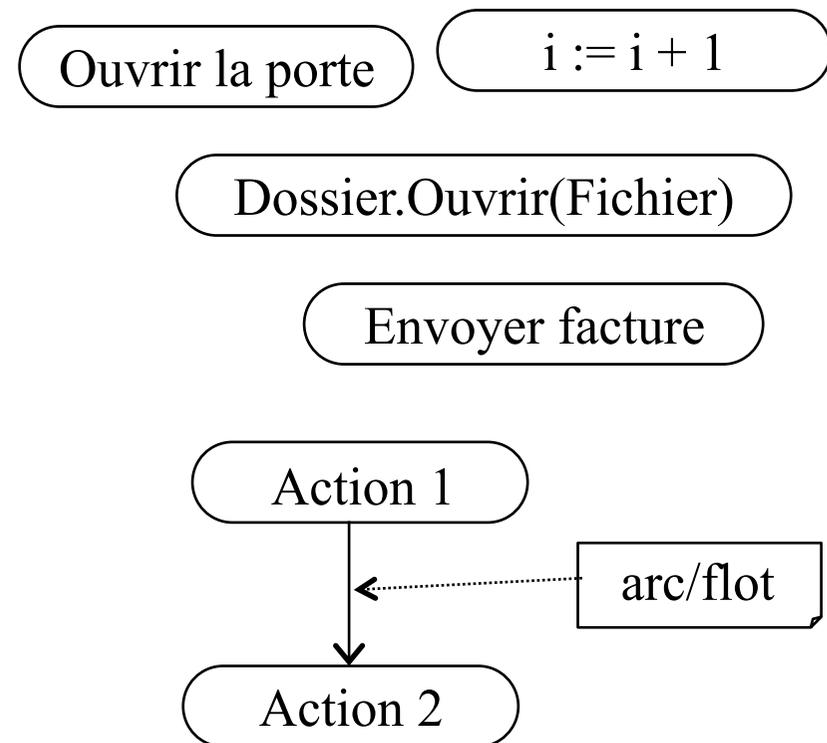


Plan

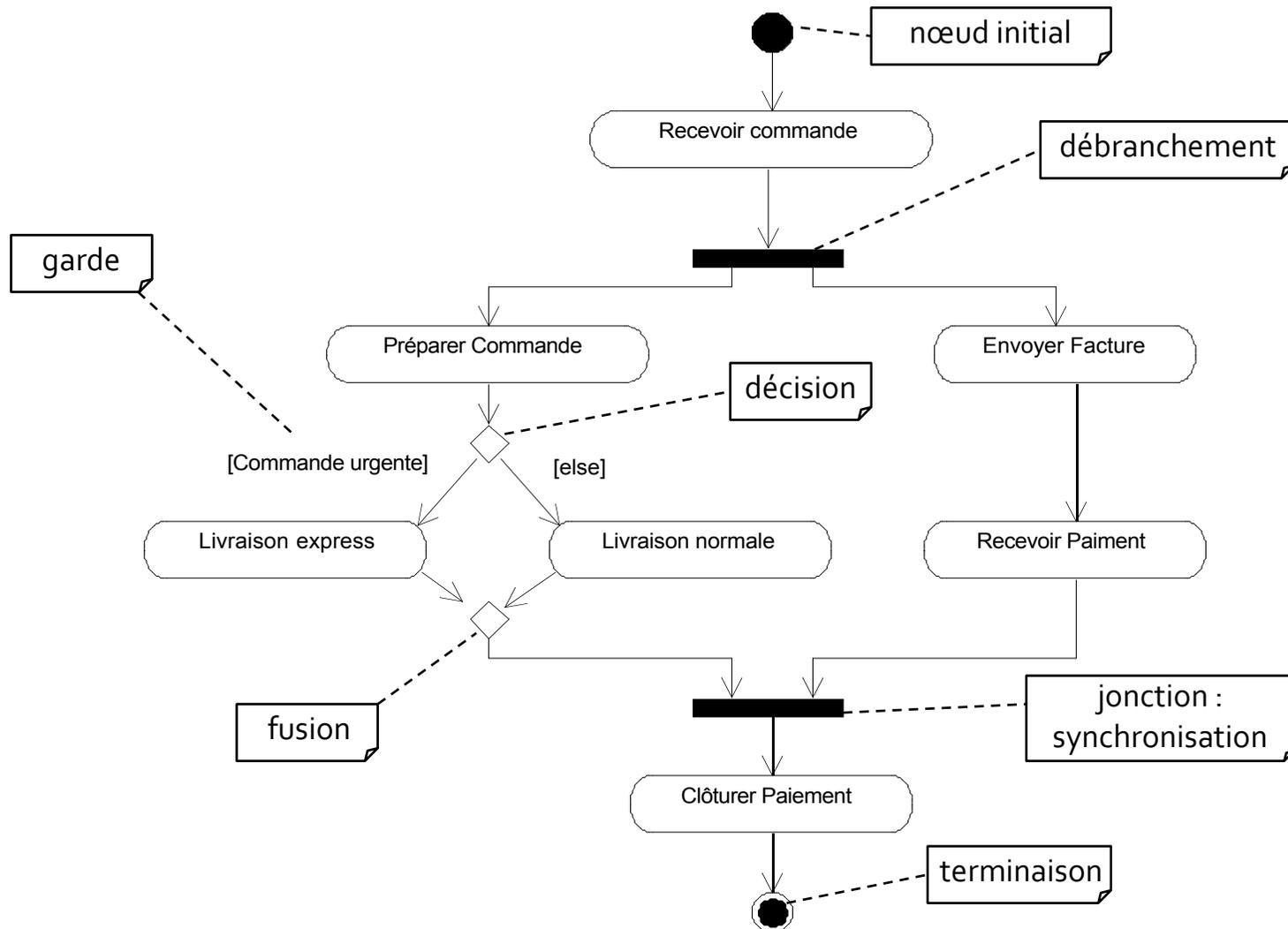
- Diagrammes de cas d'utilisation
- Diagrammes d'interaction
 - diagrammes de séquences
 - diagrammes de communication
- **Diagrammes d'activité**
- Diagrammes de machines d'états
- Autres diagrammes UML
- Autres diagrammes non UML

Diagrammes d'activité

- Objectif
 - présenter les activités séquentielles d'un processus
 - activité = suite d'actions
- Action
 - travail à réaliser
 - nœud du graphe
- Transition
 - contrainte d'enchaînement
 - relation du graphe
- Raffinements
 - débranchements / jointures
 - décisions / fusions
 - entrée / terminaison
 - ressources utilisées (objets)



Exemple



Petit exercice à faire en classe

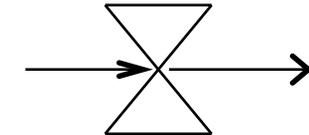
- Modéliser les activités autour d'un enseignement de l'UFR informatique.

Diagrammes d'activité pour modéliser ...

- Les processus métier de l'organisation
 - qui fait quoi, où
 - les enchaînement d'activité (workflow)
- Les flots de données
 - DFD (Data Flow Diagram) en UML
- La logique procédurale
 - algorithmes complexes, parallèles
 - organisation séquentielle globale des activités de plusieurs objets
 - vs. diag. machines d'états : un objet

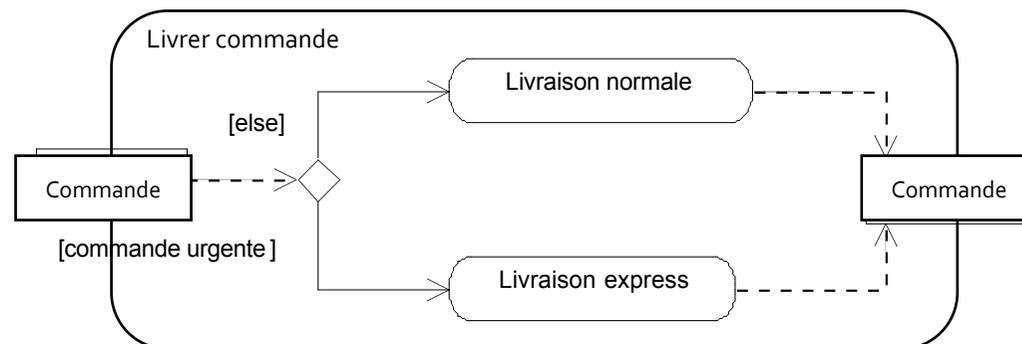
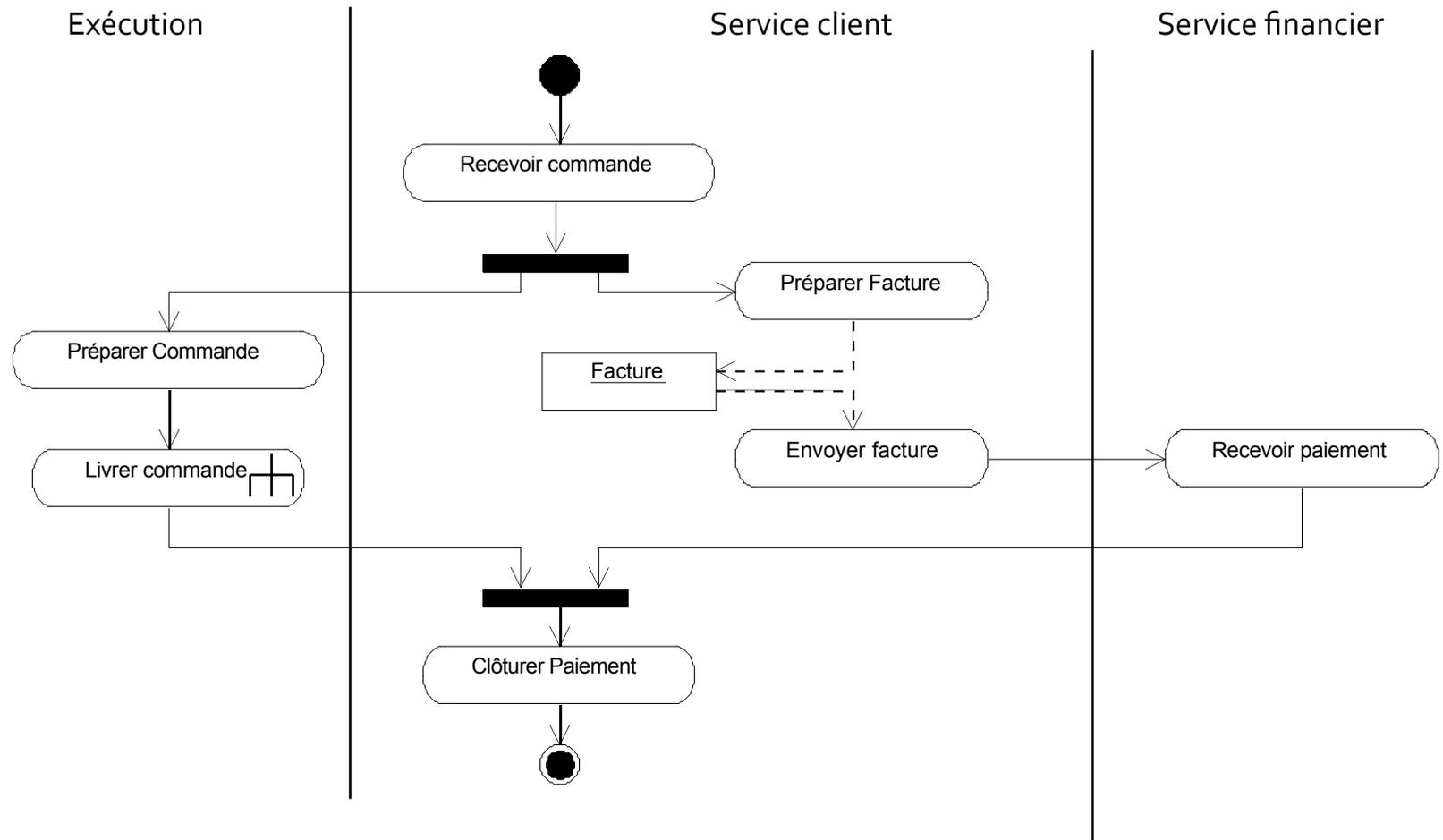
Diagrammes avancés

- Actions liées à des signaux
 - délai
 - envoi / réception
- Utilisation d'objets
 - en entrée ou sortie d'action
- Partitions (UML1 : swimlanes, travées)
 - montrer les responsabilités au sein du mécanisme ou d'une organisation
- Décomposition des actions
 - appeler une sous-activité (un autre diagramme d'activité) dans une action



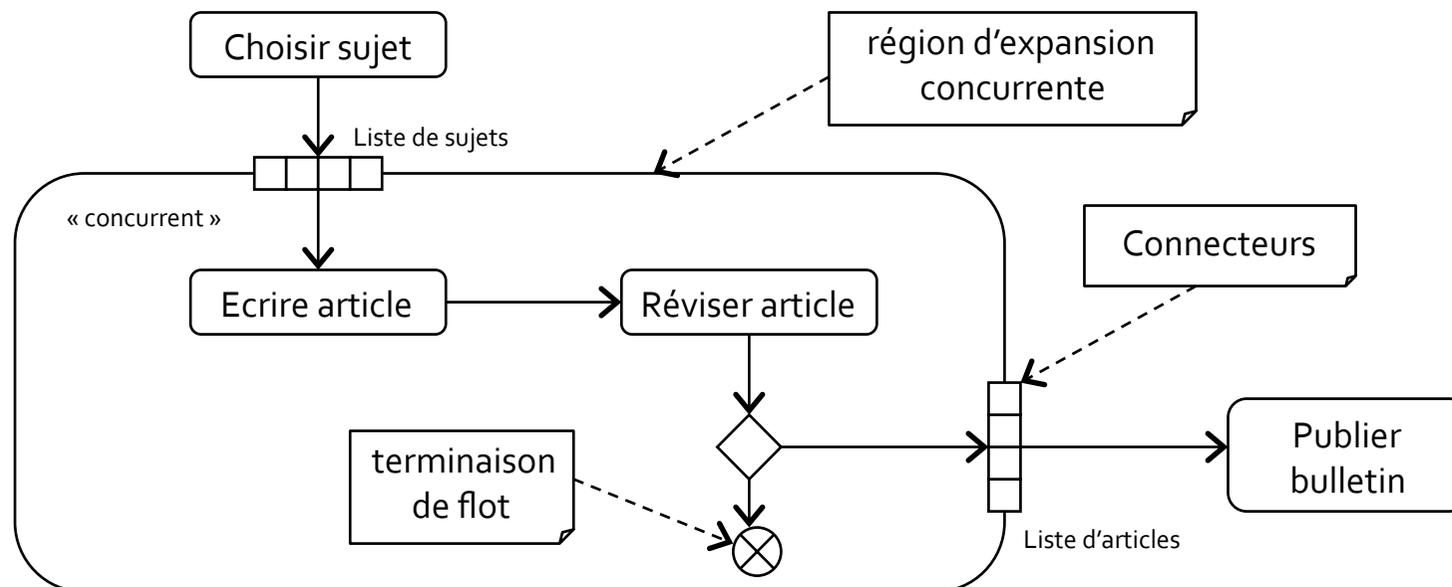
attendre 2 heures





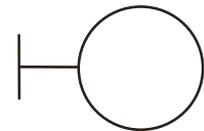
Connecteurs, régions d'expansion, terminaison de flots (UML2)

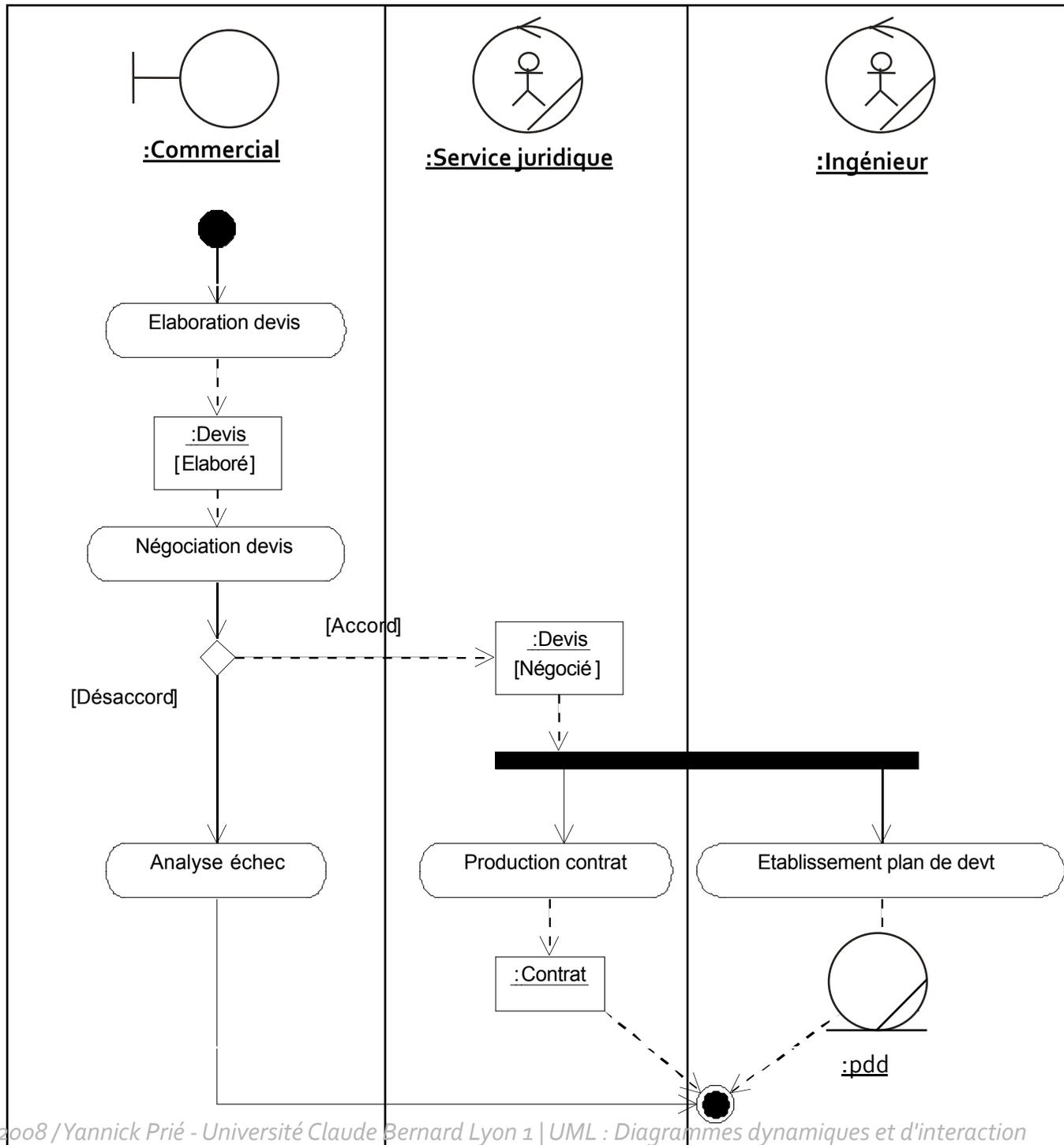
- Connecteurs
 - cf. objets paramètres entrée/sortie actions
- Régions d'expansion
 - actions qui se passent pour plusieurs éléments de même type (itératif ou concurrent)



Modélisation de processus métier

- Modéliser le fonctionnement d'une organisation
- Trois stéréotypes d'objets
 - *Case worker*
 - interaction avec l'ext. de l'entreprise
 - *Internal worker*
 - travail à l'intérieur de l'entreprise
 - *Entity*
 - objet passif : stockage d'informations
- Partitions / couloirs d'activité



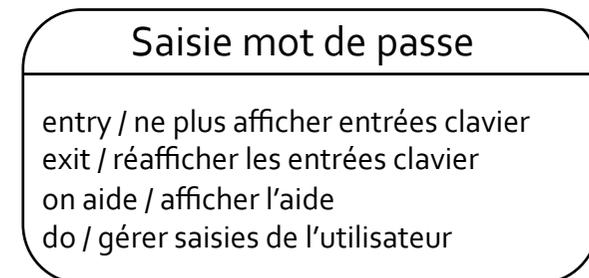


Plan

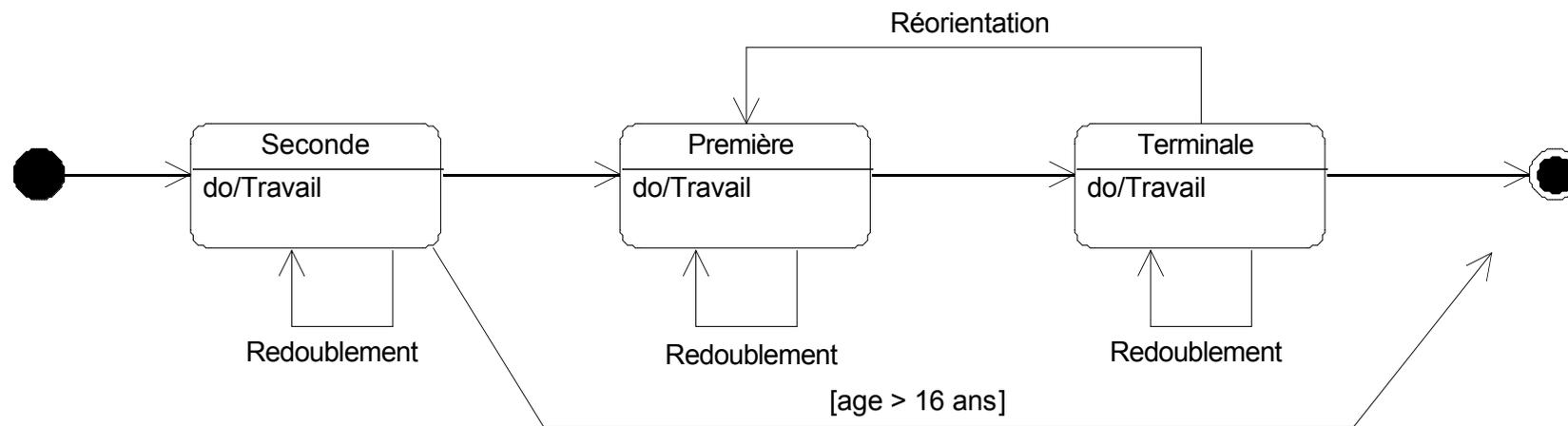
- Diagrammes de cas d'utilisation
- Diagrammes d'interaction
 - diagrammes de séquences
 - diagrammes de communication
- Diagrammes d'activité
- **Diagrammes de machines d'états**
- Autres diagrammes UML
- Autres diagrammes non UML

Diagrammes de machines d'états

- Abstraction des comportements possibles pour une classe
 - automate à états finis décrivant les chemins possibles dans le cycle de vie d'un objet
- Etat d'un objet
 - situation nommée d'un objet qui répond à certaines conditions de durée et de stabilité
 - Ex. : activité continue (sonnerie), tâche de fond (pagination), attente, suite d'actions...
- Transition entre états
 - réponse de l'objet dans un certain état à l'occurrence d'un événement
 - passage d'un état à un autre sur événement + condition respectée,
 - action à exécuter



Exemple de diagramme de machines d'états



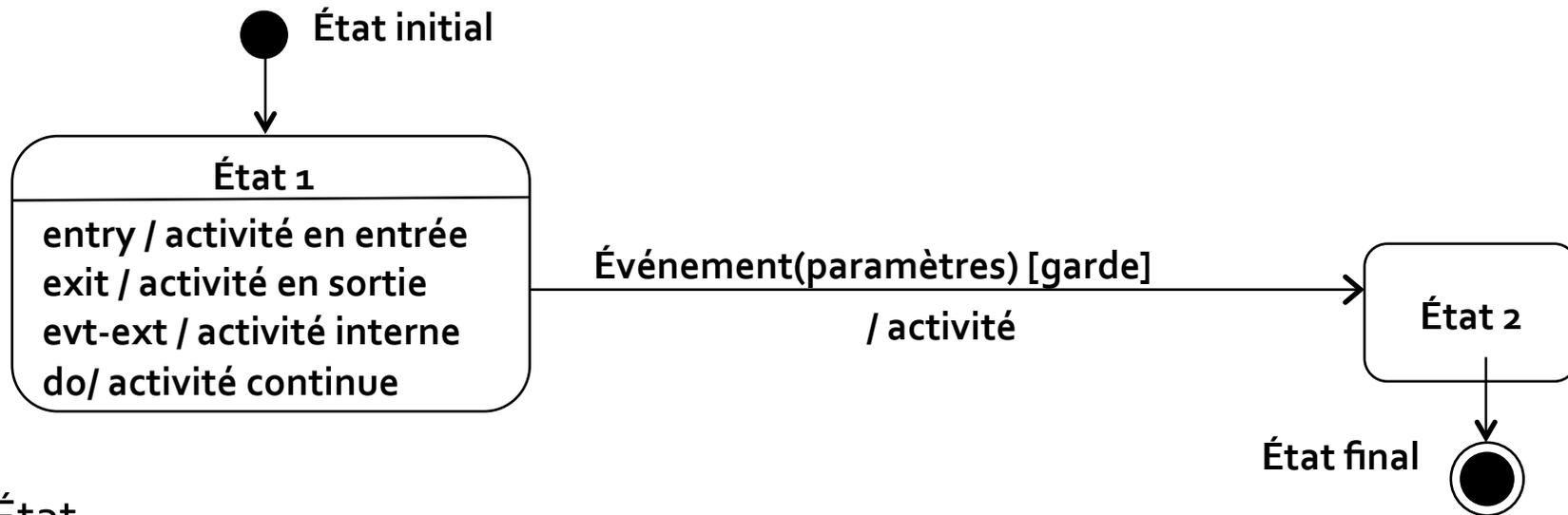
Petit exercice à faire en classe

- Tracer un diagramme de machines d'états pour un objet « TP-Etudiant ».

Utilisation

- Pour se concentrer sur le fonctionnement d'une classe
 - décrire / fixer le comportement concret de la vie d'une objet lié à un ou plusieurs scénarios
- Pour les classes complexes
 - objets réactifs complexes (objets métier...)
 - protocole et séquences légales (sessions...)
 - en général pas plus de 10% des classes d'une application
 - plus en télécommunication / moins en informatique de gestion
- Larman
 - navigation dans un site web, IHM
 - enchaînement de pages/fenêtres

Syntaxe générale

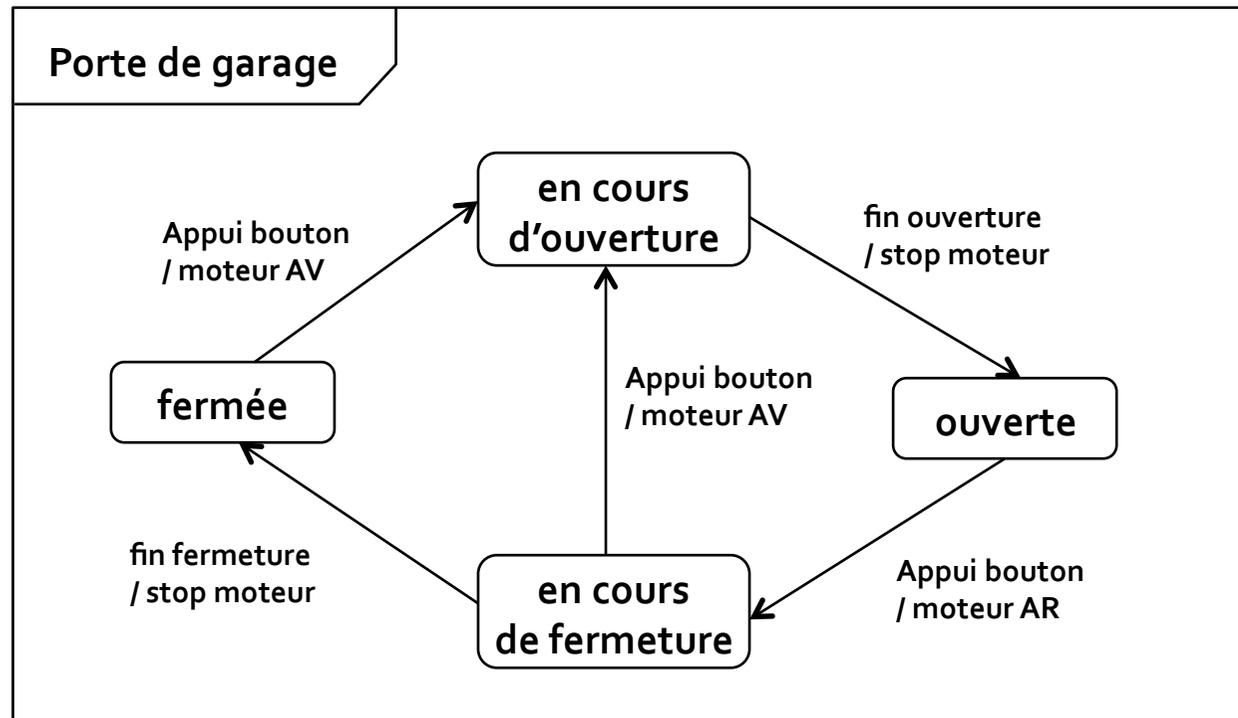
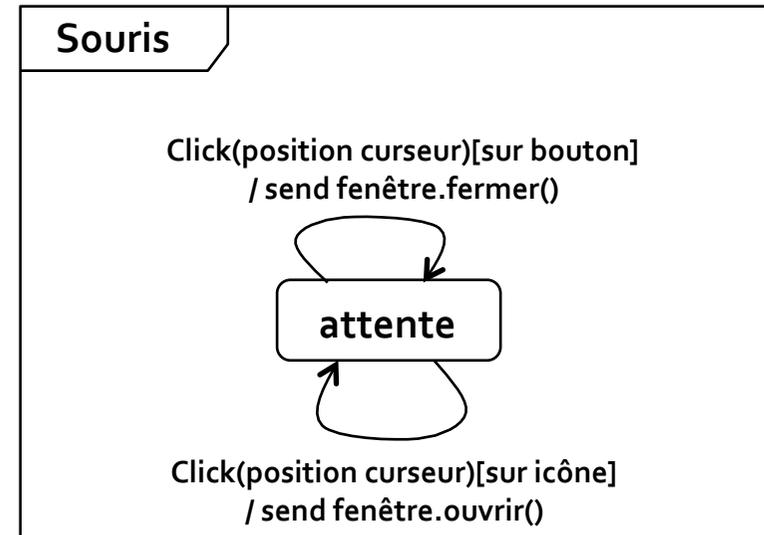
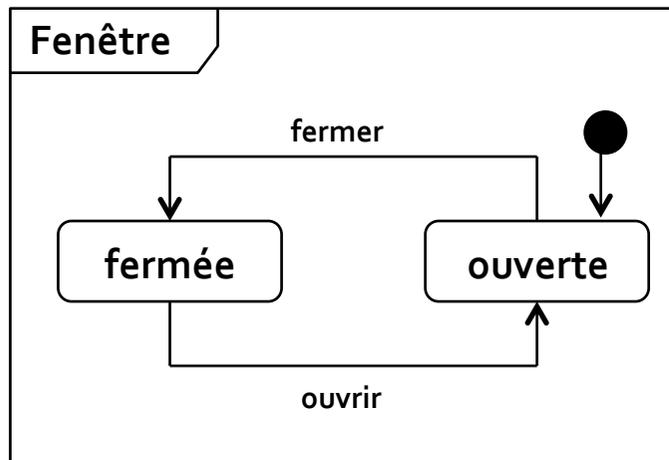


■ État

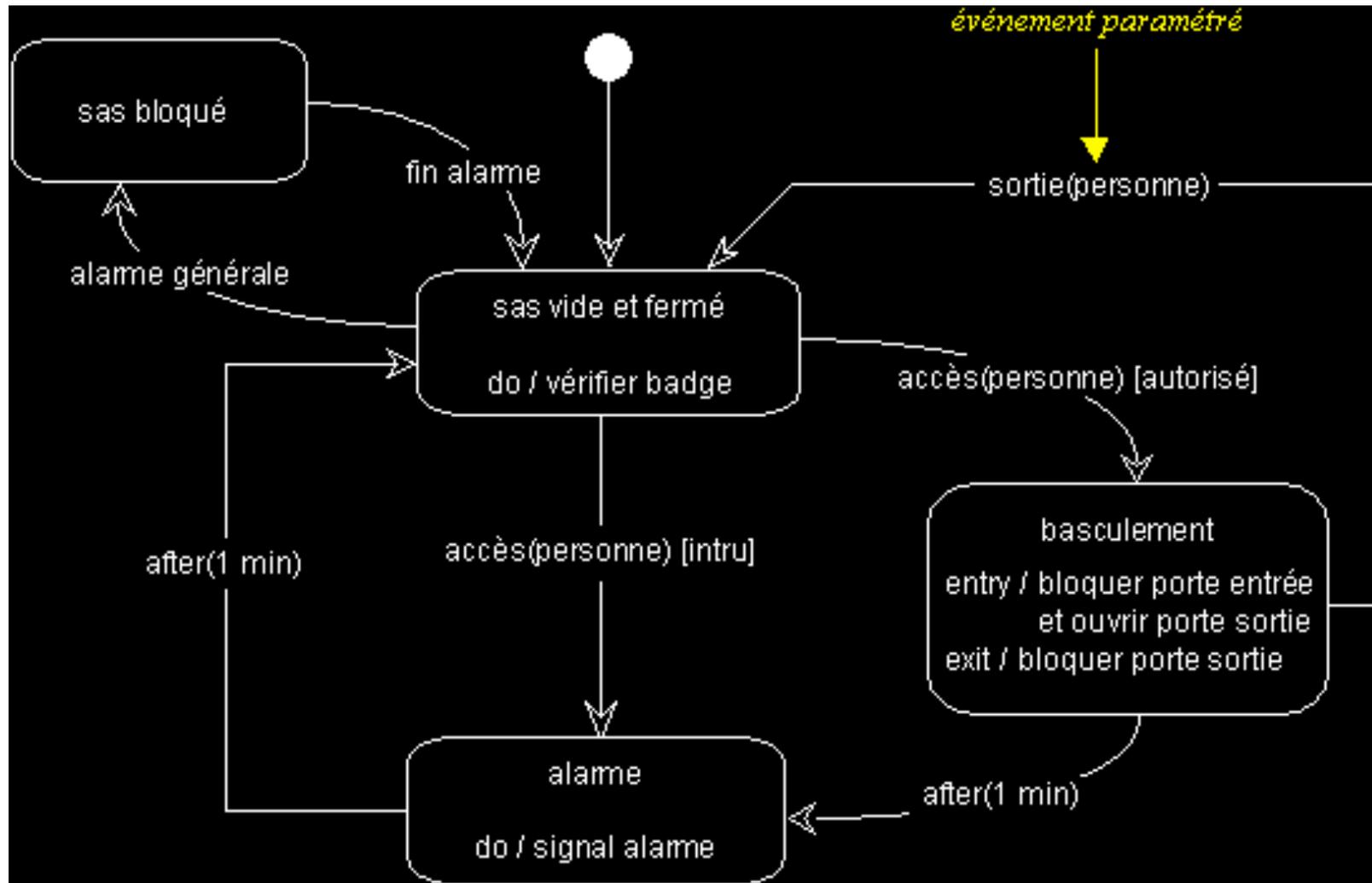
- activité continue : peut être interrompue
 - do / activité
- activités internes instantanées
 - auto-transition sur événement extérieur, instantanée, qui ne nécessite pas changement d'état
 - deux activités spéciales : sur entrée et sortie

■ Transition

- tout est facultatif mais absence d'événement rare
- événements
 - résultants de messages entre objets
 - internes : when(maximum atteint)
 - temporels : after(3 jours)
- activité classique : envoyer un message à une cible
 - send cible.message(arguments)

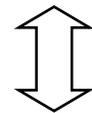
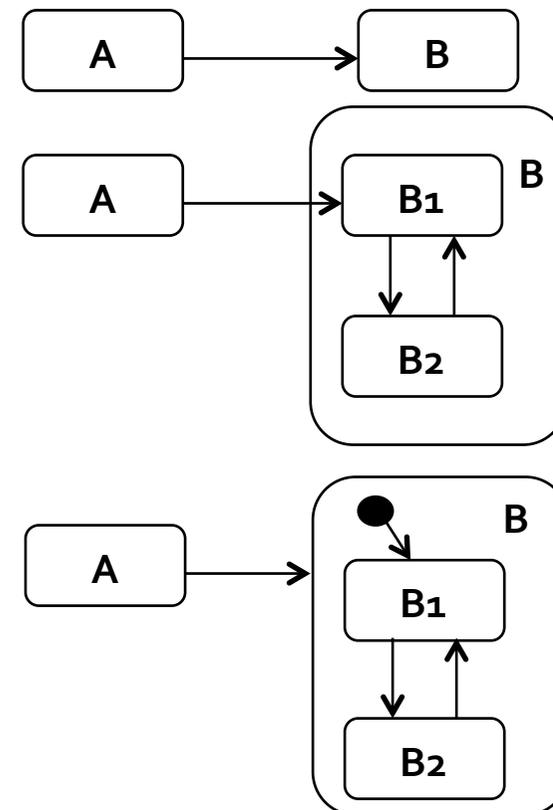
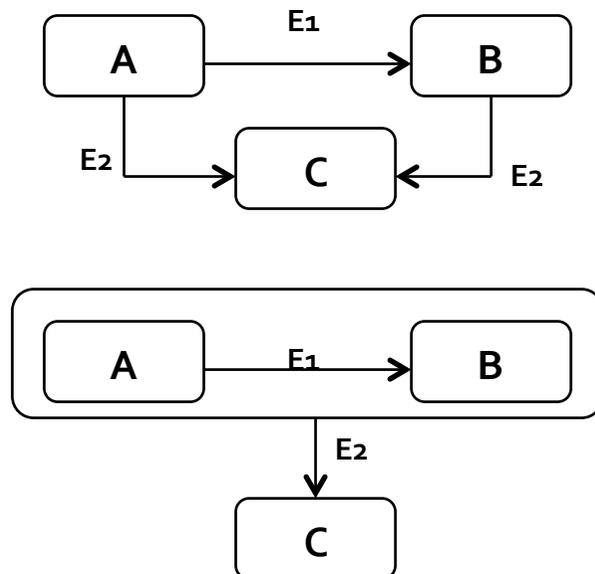


Autre exemple

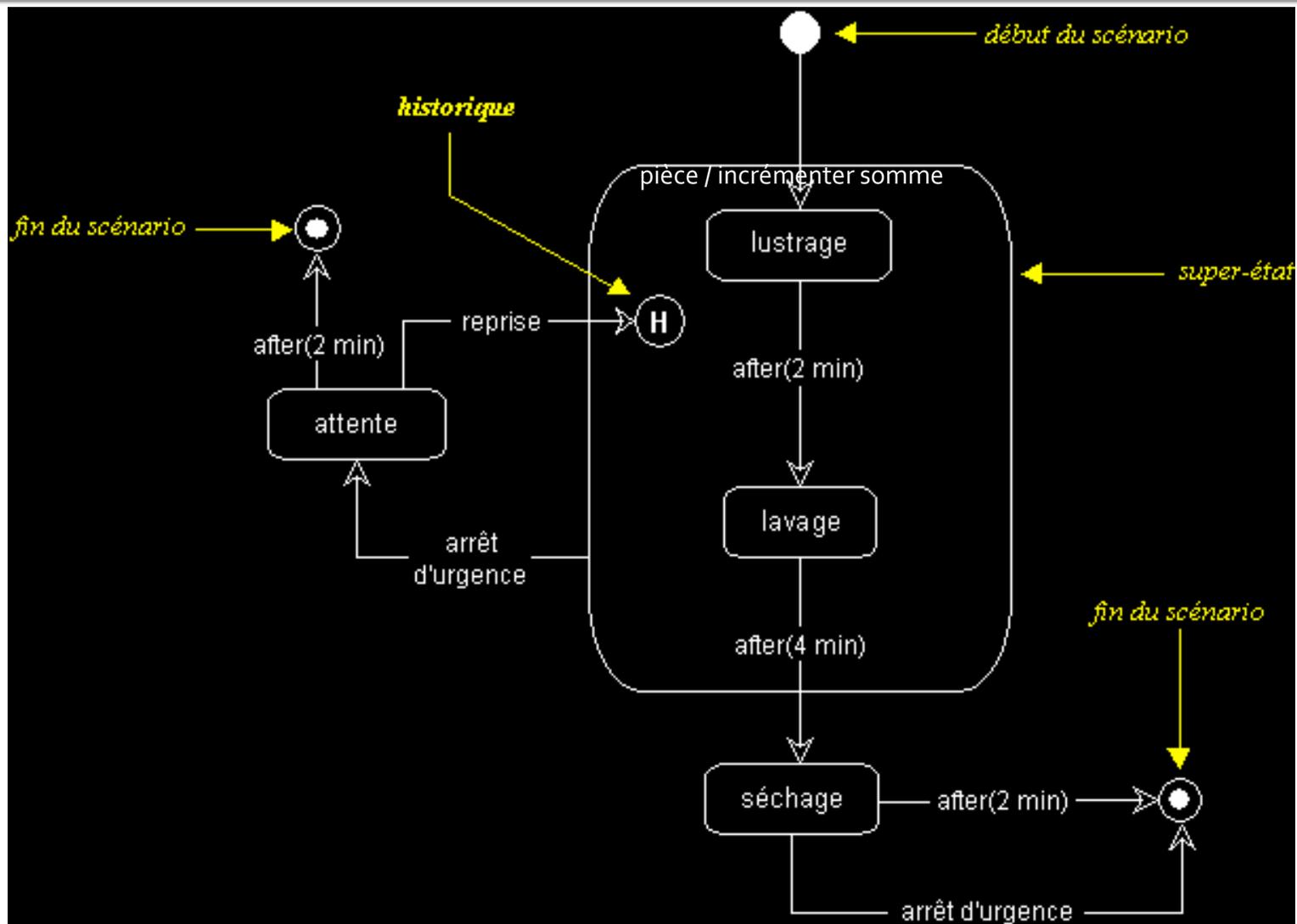


Super-états / états composites

- Pour factoriser un comportement
 - transitions déclenchées par le même événement, conduisant au même état
- Transition interne
 - couple événement / activité sans effet sur l'état courant
 - permet de ne pas « réinitialiser » l'état en revenant à l'état de départ
 - se note en haut du super état

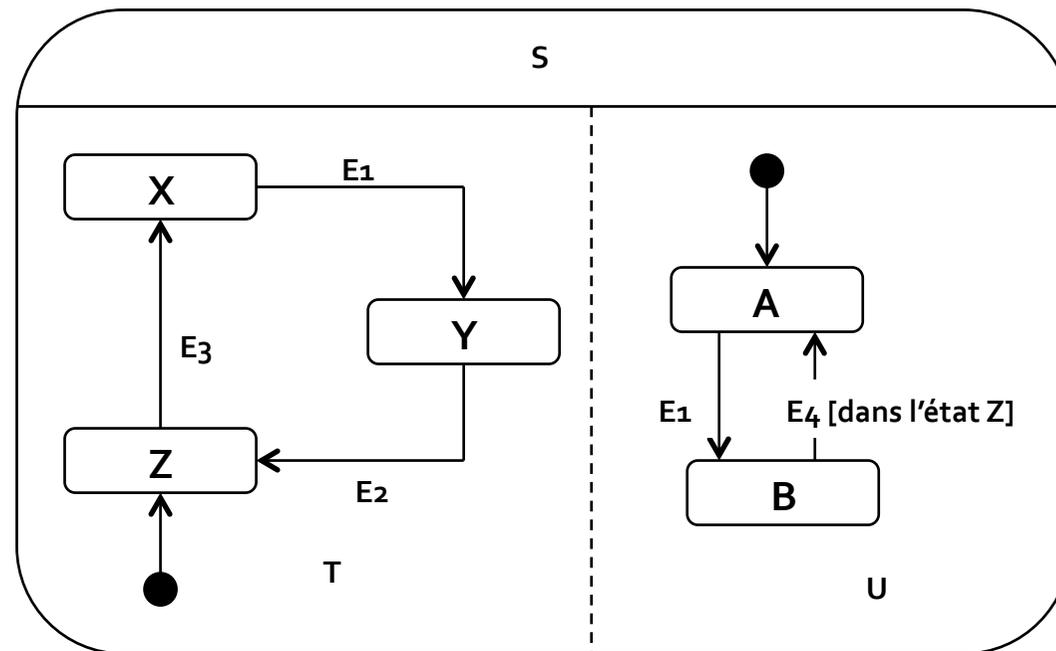


Exemple super-état



États concurrents

- Pour décomposer des états complexes
- Exercice : trouver le diagramme d'état « à plat » équivalent



Implémentation des états

- Utilisation de case ... switch
 - déconseillé
- Utilisation du pattern état
 - arborescence de classes états
 - délégation de la gestion de l'état
- Tables d'états
 - représentation tabulaire du diagramme
 - état source, cible, événement, garde, procédure à exécuter
 - permet de « paramétrer » le comportement de la classe

Plan

- Diagrammes de cas d'utilisation
- Diagrammes d'interaction
 - diagrammes de séquences
 - diagrammes de communication
- Diagrammes d'activité
- Diagrammes de machines d'états
- **Autres diagrammes UML**
- Autres diagrammes non UML

Vue d'ensemble des interactions

- Mixte diagramme activité / diagrammes de séquences
- les actions sont remplacées par des diagrammes de séquence
- Utilisé ?

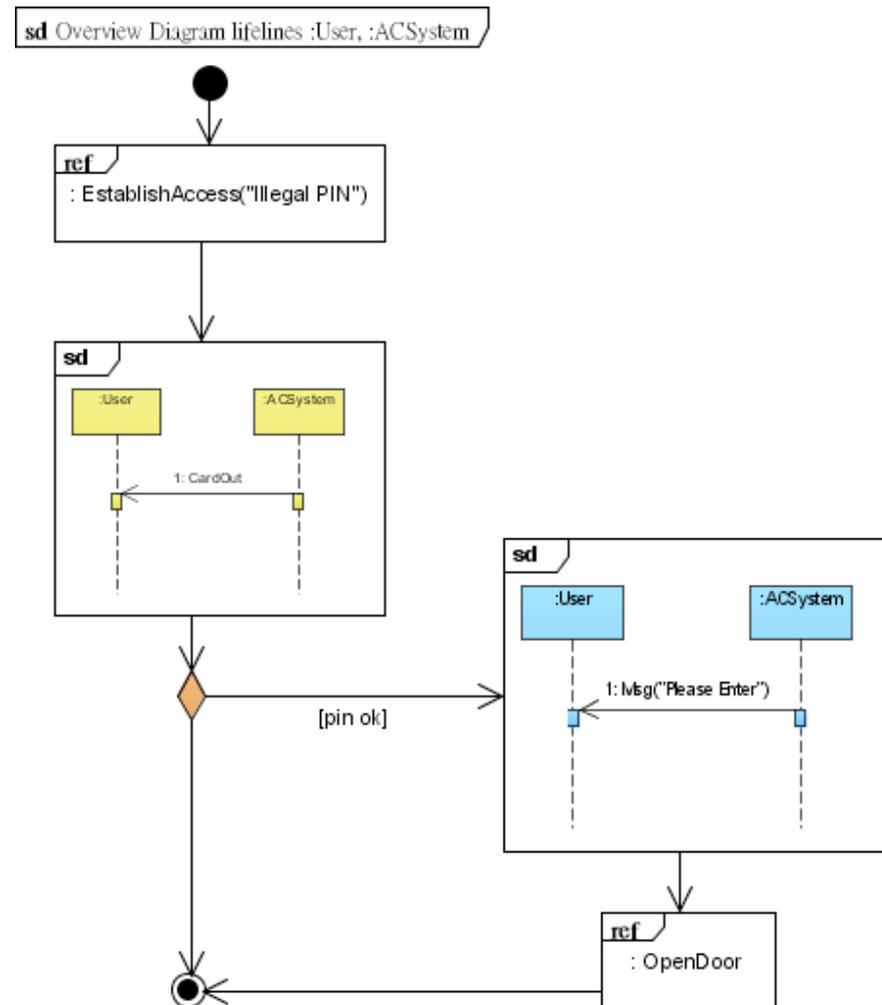
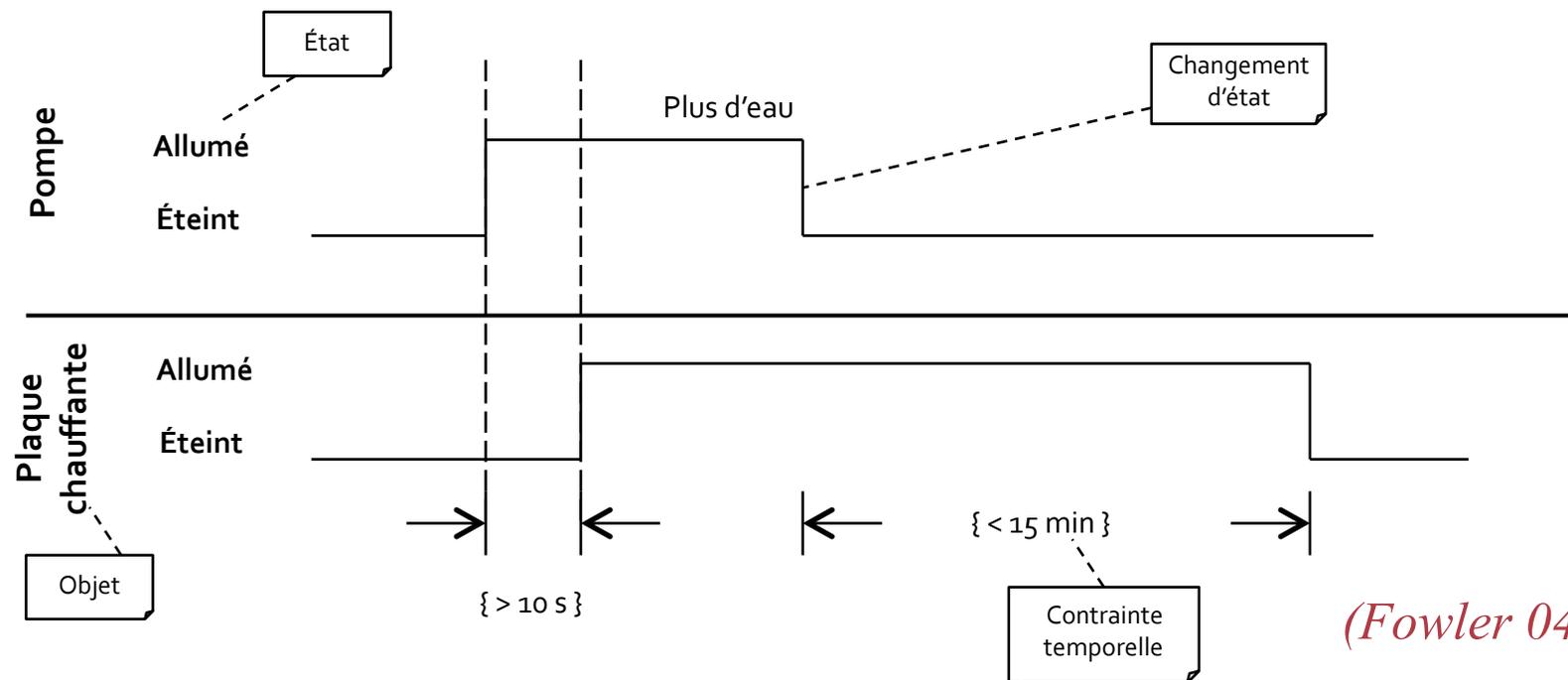


Diagramme de timing

- Interactions avec focus sur les changements d'états d'objets et les contraintes temporelles associées
 - ligne de vie horizontale
- Utilisé surtout dans les applications temps réel



Diagrammes de collaborations

- Non officiels dans UML2, se rapprochent des diagrammes de structure composite
- Permettent de présenter les éléments impliqués dans une collaboration, et le rôle qu'ils y jouent
 - fixer les éléments et les rôles pour les diagrammes d'interaction
- En théorie utilisés pour représenter des patterns

Plan

- Diagrammes de cas d'utilisation
- Diagrammes d'interaction
 - diagrammes de séquences
 - diagrammes de communication
- Diagrammes d'activité
- Diagrammes de machines d'états
- Autres diagrammes UML
- **Autres diagrammes non UML**

Diagrammes de contexte (Roques, 2004)

- Diagramme de contexte statique
 - diagramme de classe
 - une classe système
 - tous les acteurs autour
- Diagramme de contexte dynamique
 - diagramme de communication qui résume les messages entre système et acteurs (pas de numérotation)
- Diagramme de contexte statique étendu
 - diagramme de contexte statique avec
 - Attributs et opérations de haut niveau pour le système et les acteurs non humains
- Remarque
 - « diagrammes de classes avec messages »
 - diagramme de classe avec résumé des messages entre classes

Diagramme de flux d'écrans informel

(Fowler, 2004)

- Un rectangle par écran
- Des flèches pour la navigation
 - éventuellement un nom signifiant le lien

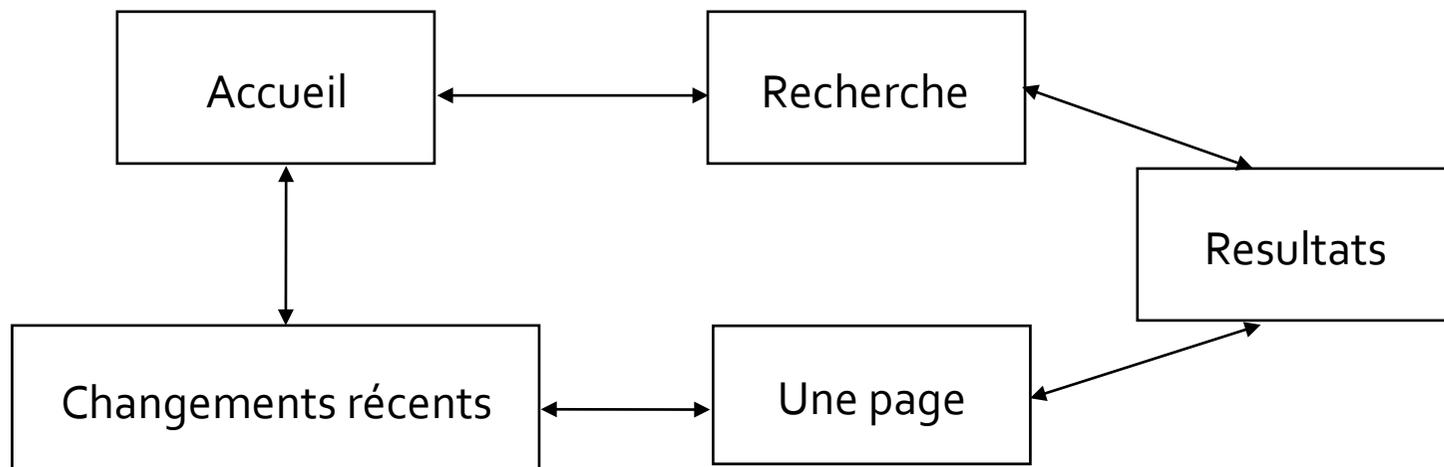


Table de décision (Fowler)

- Pour représenter des conditions logiques complexes
- Deux parties
 - conditions
 - conséquences

Client privilégié	X	X	O	O	N	N
Commande prioritaire	O	N	O	N	O	N
International	O	O	N	N	N	N
Prix	150	100	70	50	80	60
Alerte	oui	oui	oui			

Cartes CRC

- Classes - Responsabilités - Collaborateurs
 - à la base inventé pour l'enseignement
- Jouer des scénarios avec des cartes
 - 5-6 participants
- Une carte
 - nom de classe
 - tableau à deux colonnes
 - responsabilité de la classe : quelque chose d'un objet doit faire
 - collaborateurs : classes avec lesquelles il faut collaborer pour assurer la responsabilité
- Jeu
 - déterminer des classes de départ avec responsabilités évidentes
 - jouer les scénarios, ajouter les responsabilités, créer de nouvelles classes, etc.
- Voir par exemple
 - http://www.csc.calpoly.edu/~dbutler/tutorials/winterg6/crc_b/

Classe	
Responsabilite1	Coll1,2

A suivre

- UML 4/4 : concepts avancés