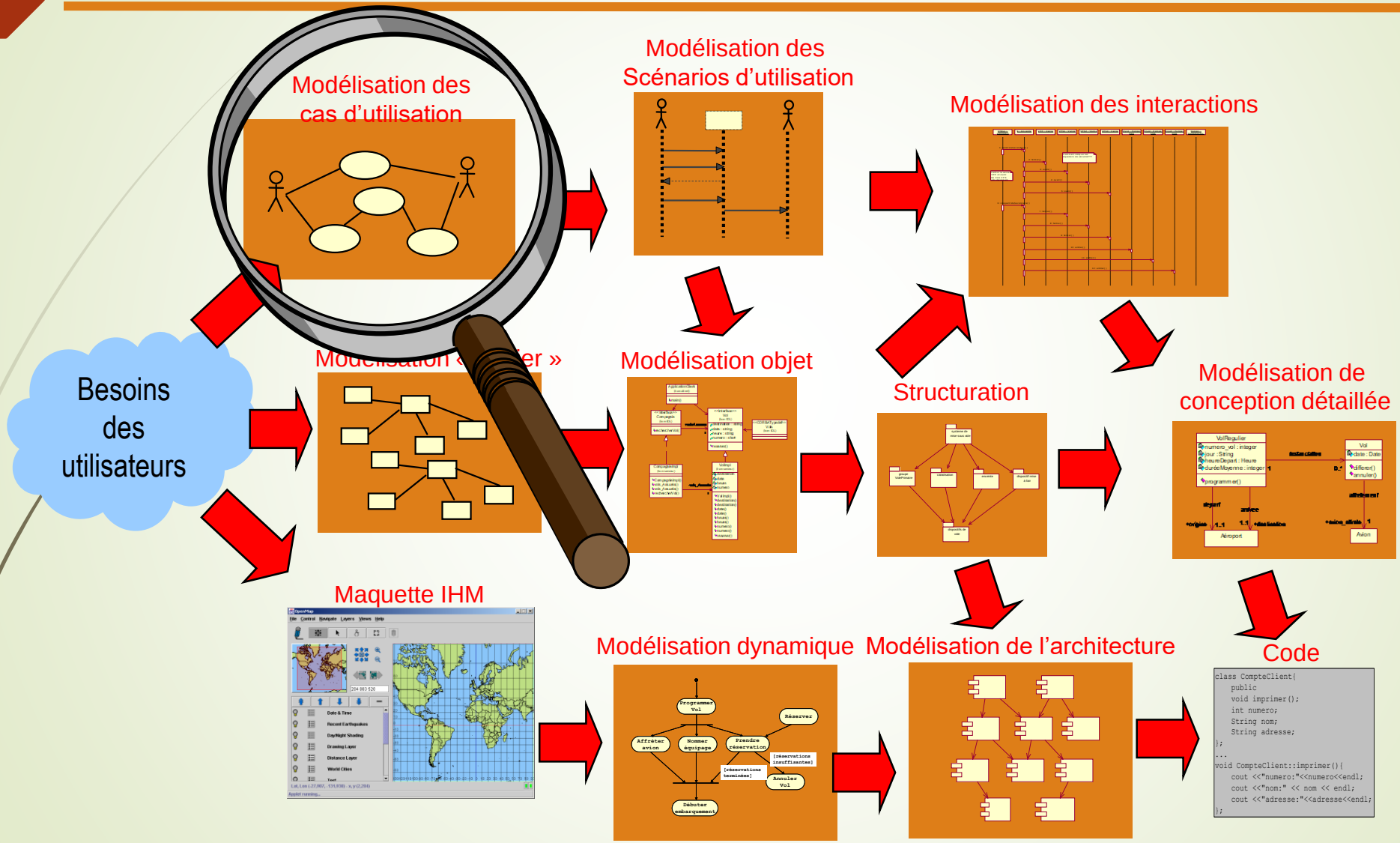
Chapitre II : Analyse, vues cas d'utilisation et processus

OBJECTIFS DU CHAPITRE:

- ✓ Identifier l'acteur d'un système;
- ✓ Identifier les cas d'utilisations d'un système;
- ✓ Représenter les relations entre acteurs;
- ✓ Représenter les relations entre cas d'utilisations;
- ✓ Donner la description détaillée d'un cas d'utilisation
- ✓ Présenter l'algorithme d'un cas d'utilisation à l'aide d'un diagramme d'activité.

Place dans le processus de développement



Cas d'utilisation: à quoi ça sert?

- un cas d'utilisation « **raconte** » sous forme de texte la façon dont un **acteur** va utiliser un **système** pour atteindre un **but** particulier.

  - Il correspond à une fonction du système
  - Il décrit les interactions entre le système et les utilisateurs
  - Il détermine un contrat à remplir par le système
  - Il induit des exigences fonctionnelles applicables au système et il peut être utilisé pour organiser la spécification
  - sa finalité est de détecter et de décrire les besoins fonctionnels, en précisant de quelle manière un système est utilisé pour permettre à un client -au sens large à un utilisateur- d'atteindre ses différents objectifs.

Cas d'utilisation: à quoi ça sert?

- Nous avons des **buts**, et nous voulons des systèmes informatiques qui nous aident à les atteindre.
- De brillants analystes ont inventé de nombreux moyens de capturer ces besoins et ces buts, mais les meilleurs sont **les plus simples et les plus courants**.
- Ils facilitent la participation des clients et utilisateurs à leur définition et à leur évaluation, ce qui **diminue le risque d'erreur**.

Cas d'utilisation: à quoi ça sert?

- L'absence d'implication des utilisateurs est l'une des principales raisons d'échec des projets logiciels, et tout ce qui peut les aider à demeurer motivés est éminemment souhaitable.
- Les cas d'utilisation constituent un procédé qui aide à rester **simple**, et qui permet aux experts du domaine et/ou aux utilisateurs de les **écrire eux-mêmes** (ou de participer à leur rédaction).
- Un autre intérêt des cas d'utilisation est qu'ils mettent l'accent sur les **points de vue** et les **buts de l'utilisateur.**

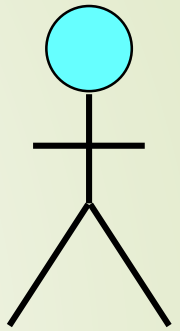
Cas d'utilisation: Exemple

- **Traiter une vente** :
  - Un client arrive à la caisse avec les articles qu'il souhaite acheter.
  - Pour enregistrer chaque article, la caissier utilise le système informatisé, lequel présente le détail des articles et le montant total des achats.
  - Le client fournit les informations nécessaires pour le règlement.
  - Le système valide et enregistre ces informations, puis met à jour les quantités en stock et imprime le ticket de caisse destiné au client.
  - La vente est terminée et le client peut quitter le magasin.

Concepts de base : Acteurs

Acteur

- représente un **rôle** joué par une personne ou une chose qui interagit avec le système mais qui lui est **extérieure**.
- est caractérisé par un nom qui exprime son rôle.
- une même personne physique peut être modélisée par plusieurs acteurs.



Etudiant

<< acteur >>

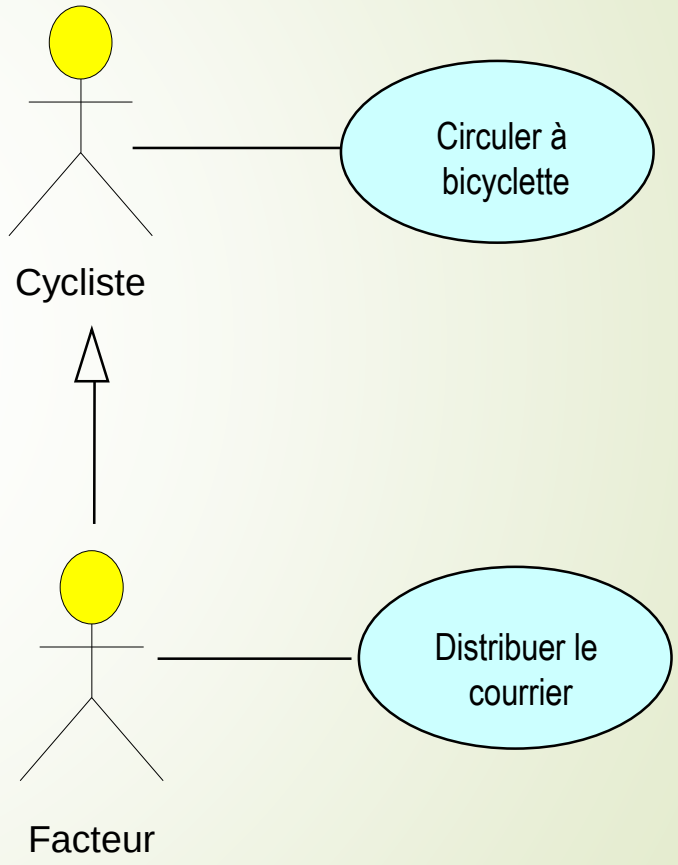
Administrateur

Concepts de base : Catégories d'Acteurs

- **On distingue 4 catégories d 'acteurs:**
  - les **acteurs principaux** (ex: usager, client, etc.)
  - les **acteurs secondaires** (ex: opérateur de maintenance, administrateur, etc.)
  - le **matériel externe** (capteurs, imprimantes, périphériques divers, etc.)
  - les **autres systèmes** (serveur central, service ou organisation, etc.)

Concepts de base : Relation entre acteurs

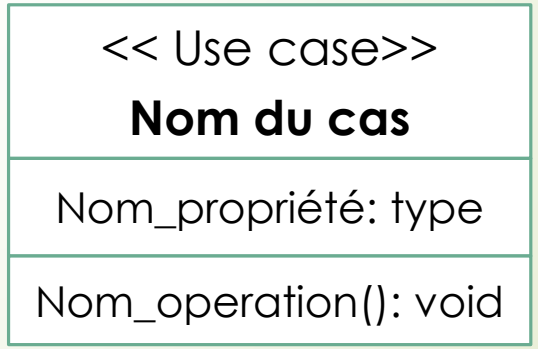
- *Un Acteur peut hériter d'autres Acteurs.*
- *Un Acteur peut posséder des Interfaces.*



Concepts de base : Cas d'utilisation

Cas d'utilisation

- **unité fonctionnelle** cohérente assurée par un système ou une classe
- correspond à un certain type d'interaction entre le système et les acteurs.
- doivent être vus comme des classes dont les instances sont des **scénarii**.

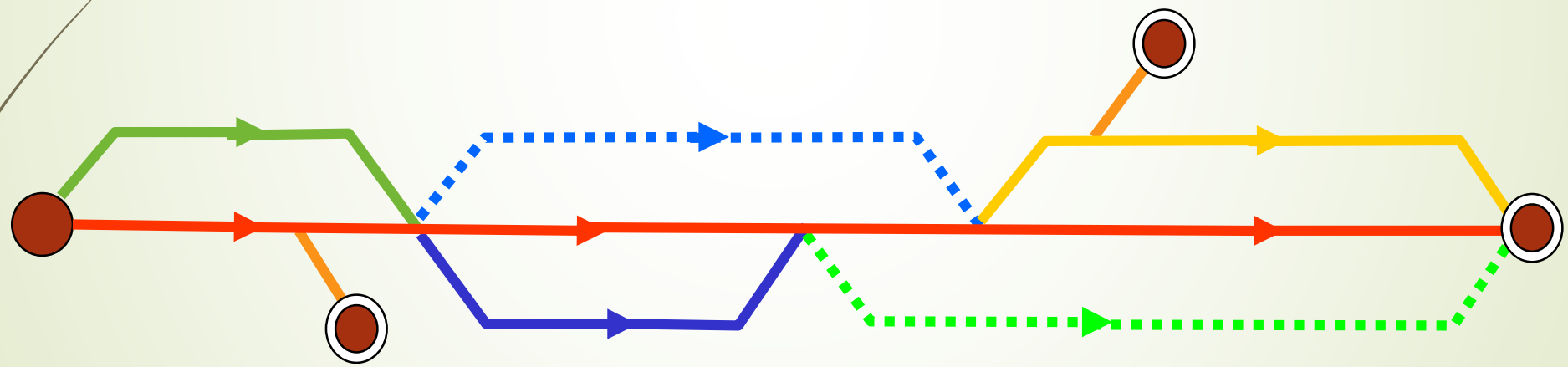


Concepts de base : scénario

- Un **scénario** est une suite spécifique d'actions et d'interactions entre un ou plusieurs acteurs et le système.
  - C'est une « histoire » particulière de la façon dont on utilise un système, ou l'un des cheminements possibles d'un cas d'utilisation.
  - Par exemple, le traitement d'une vente a plusieurs scénarios possibles : la vente est validée car le client règle en espèces ou elle est invalidée car la carte de crédit du client est refusée.

Concepts de base : scénario

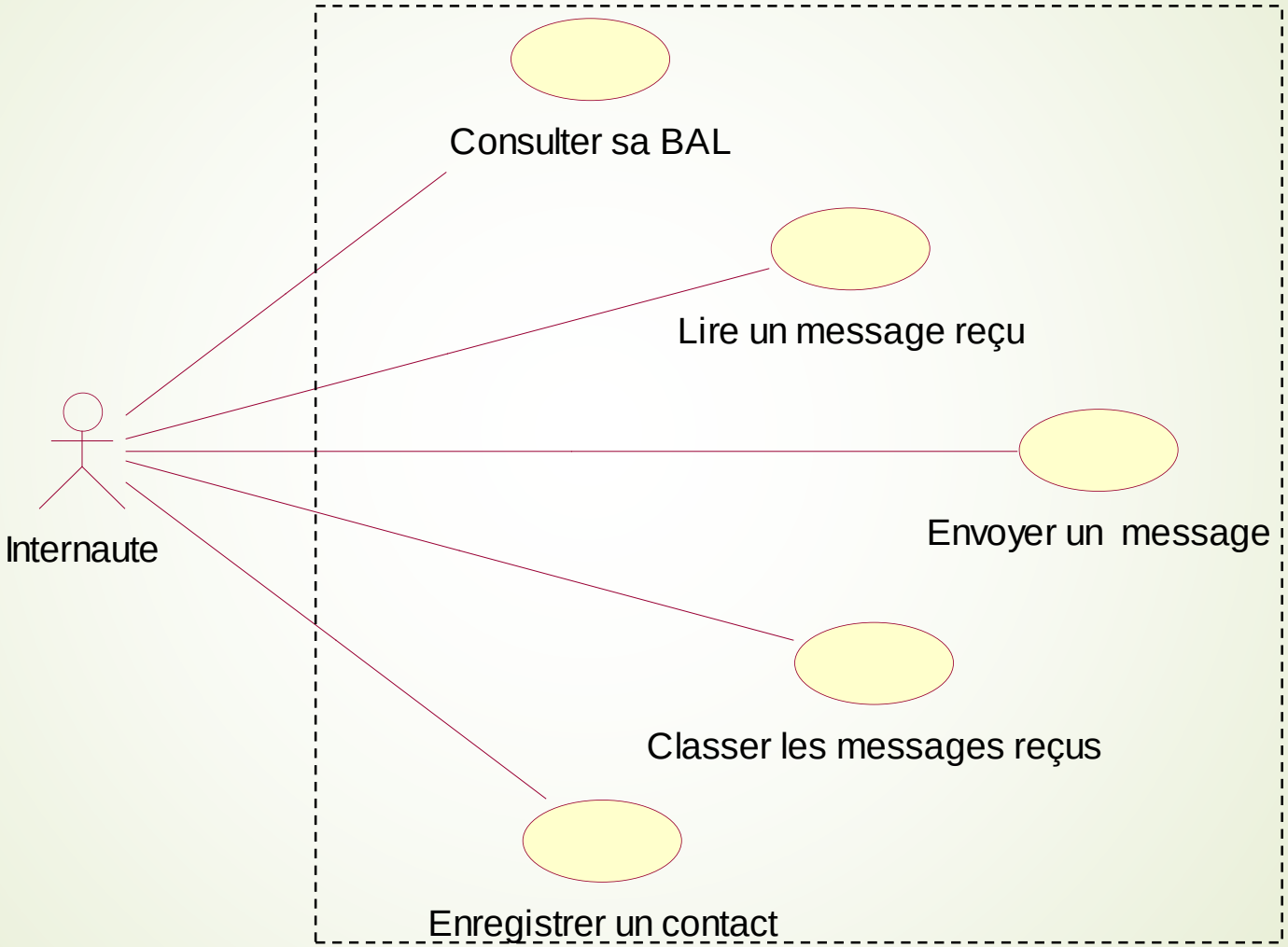
- Un **cas d'utilisation** peut être vu comme une collection de scénarii décrivant différentes façons d'utiliser le système pour atteindre un même but (avec ou sans succès).



Exemple : scénarios de cas d'utilisation

- **Traiter un retour**
  - **Scénario principal (succès)**
    - Un client arrive à la caisse avec les articles qu'il veut retourner. Le caissier utilise le système automatisé pour enregistrer chaque article retourné …
  - **Autres scénarios**
    - Si le client a payé à crédit d'avance et que la demande de remboursement sur son compte est rejetée, l'en informer et le rembourser en espèce.
    - Si le code de l'article n'est pas reconnu par le système, informer le caissier et lui suggérer de saisir le code manuellement (l'emballage peut être endommagé).
    - Si le système ne parvient pas à communiquer avec le système comptable externe…

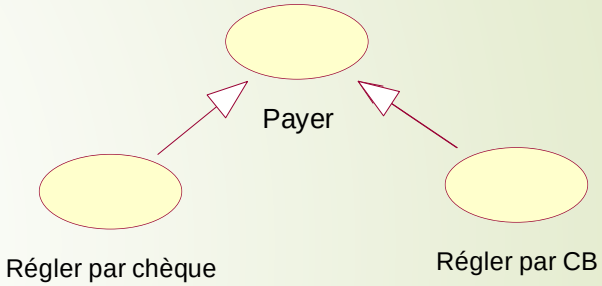
Exemple de diagramme de cas d'utilisation



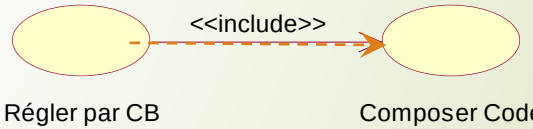
Relations entre cas d'utilisation

Communication – exprime le fait que l 'acteur participe à la réalisation d 'un cas d 'utilisation . C 'est la seule relation qui peut exister entre un acteur et un cas d 'utilisation.

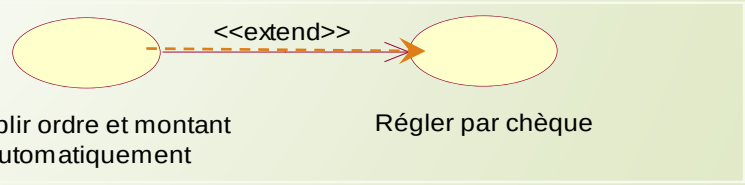
Généralisation - un cas d 'utilisation « enfant » hérite du comportement et de la sémantique du cas d 'utilisation parent



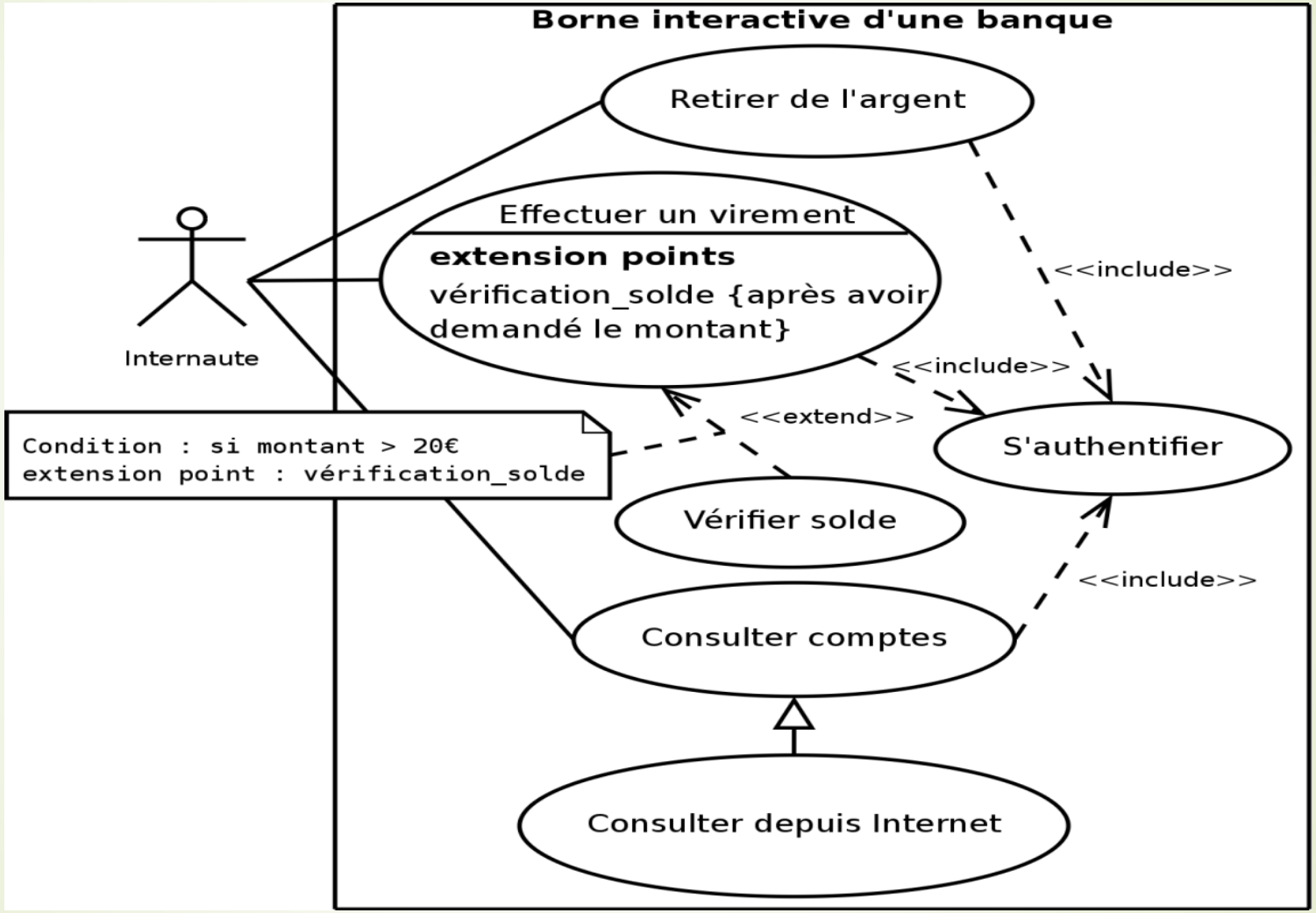
Relation « Include » – Une relation « include » du cas d 'utilisation A vers le cas d 'utilisation B signifie que le flot d 'événements de A contient une séquence d 'événements qui correspond à B.



Relation « Extend » – Une relation « extend » du cas d 'utilisation A vers le cas d 'utilisation B signifie que le flot d 'événements de A peut intervenir, de façon facultative, pendant le déroulement de B. B spécifie un comportement facultatif



Exemple de diagramme de cas d'utilisation



Les trois formats des cas d'utilisation

- **Format abrégé :**
  - Résumé succinct qui présente généralement le scénario de base (succès) dans un paragraphe.
  - C'est le cas de notre premier exemple 'traiter une vente'
  - S'utilise lors de la première étude des besoins, pour se faire rapidement une idée du sujet et de son périmètre.
  - Quelques minutes peuvent suffire à les créer.

Les trois formats des cas d'utilisation

- **Format informel :**
  - Les différents scénarios sont décrits dans plusieurs paragraphes.
  - C'est le cas de notre deuxième exemple, 'Traiter un retour'.
  - S'utilise comme le format abrégé.

Les trois formats des cas d'utilisation

- **Format détaillé :**
  - Toutes les étapes et les variantes sont indiquées en détail, de même que les préconditions et les postconditions (ou garantie de succès).
  - S'utilise lorsque de nombreux cas d'utilisation on été identifiés et rédigés au format abrégé.

CAS D 'UTILISATION Niveaux de description

- **Général**
  - Brève description
  - 3-5 phrases
- **Détaillé**
  - Description précise et structurée
  - Description des alternatives

Calculer un itinéraire

Titre: Calculer un itinéraire
***Acteur*:** Usager
***Description*:** Ce cas d'utilisation commence lorsque l'usager se connecte au système pour obtenir un itinéraire à suivre. Il précise son lieu de départ et son lieu d'arrivée ainsi que les paramètres de calcul. Le système lui fournit une chronologie des étapes à suivre pour atteindre la destination dans les conditions souhaitées.

Titre :	*Édition d'un plan de travail régional*	**Version :**	*1.0*
Auteur :	*P. Giroux*	**Date :**	*02/09/03*
Objet :	*Saisie des sites d'étape dans une région et détermination des itinéraires de liaisons entre sites.*		
Acteur(s) :	*Opérateur*		
Précondition(s) :	*Un fond cartographique est disponible (les cartes ont été numérisées)*		
Enchaînement nominal :			
Les acteurs	**Le système**		
1. Démarre l'application	*2. Ouvre la fenêtre d'édition et un formulaire pour définir la zone de travail*		
3. Définit la zone de travail	*4. Charge le fond cartographique*		
Pour tous les sites de la région			
5. Désigne sur la carte un site d'étape	*6. Affiche un symbole graphique et ouvre un formulaire de description du site*		
7. Complète le formulaire et le valide	*8. Enregistre le site dans la base de données*		
	9. Détermine automatiquement des routes de jonction avec les autres sites en fonction de la cartographie et calcule les distances entre sites en fonction de l'échelle de la carte		
11. Sort de l'application	*10. Enregistre les routes de jonction dans la base de données*		
	12. Ferme la fenêtre d'édition		
Exceptions :			
Jonction impossible	*Le système ne parvient pas à déterminer la route de jonction avec le site désigné*		
Base de données saturée	*Espace disque insuffisant pour enregistrer les données dans la base*		
Séquences alternatives :			
S'il existe déjà des sites définis pour la région	*4. Les sites existants sont lus en base de données et affichés avec la cartographie*		

CAS D 'UTILISATION :Description détaillée

- CANEVAS:
  - nom explicite (= label UML)
  - brève description (entre 3 et 10 lignes)
  - acteurs concernés (principaux , secondaires et autre)
  - Dates/responsable/version
  - pré-conditions
  - événement déclenchant et qui cause l 'arrêt
  - Résultats attendus / post-conditions
  - description du flot d 'événements principal
    - interactions avec les acteurs
    - échanges d 'informations (paramètres des interactions)
    - chronologie et origine des informations
    - répétitions de comportement
  - description des flots secondaires et des exceptions
  - contraintes et règles de gestion
  - exigences couvertes

Exemple : Inscription à une formation

- **Description** : le UC permet à l'administratif d'inscrire un candidat à la prochaine session d'une formation On doit pouvoir trouver ses informations s'il est déjà client, sinon on les demande. Si la prochaine session est complète, on peut l'inscrire dans une liste d'attente.
- **Intervenants** :
  - Analyste : T. Laura
  - Client : Iness Formation
- **Date de création** : 23 Novembre 2016
- **Mises à jour** : 24 novembre 2016, description simple
- **Acteurs** : enclenché par un Administratif
- **Effets** :
  - complète la liste des inscrits pour la session choisie.
  - Ajout dans la liste des clients si nouveau client

Exemple : Inscription à une formation

- **Fréquence d'utilisation** : apériodique
- **Technique de déploiement** : accessible via un PC dans le bureau des administratifs
- **Préconditions :** la liste des formations doit exister sinon ce UC n'a pas de sens.
- **Scénario normal** :
  - On présente la liste des formations. Après choix on affiche la date de la prochaine session.
  - Si pas de session, message. Si le candidat est d'accord, on demande son nom et on cherche s'il est déjà client.
  - Si oui, on récupère ses coordonnées, sinon on les encode.
- **Scénario alternatif** : .
  - Si la prochaine session est complète. On peut l'inscrire dans une liste d'attente.
  - Celle-ci sera utilisée pour créer la liste des inscrit de la prochaine session créée.
- **Scénario d'exception :**
  - …

Exemple : Inscription à une formation

- **Tests** : on testera avec un nouveau client, un client existant, une formation sans session, une session non complète et une session complète.
- **Informations nécessaires** : les coordonnées du client (nom, prénom, date de naissance, n° carte national, adresse, téléphone/GSM, [Email], [coordonnées employeur]
- **Contraintes** : un minimum d'interactions et d'encodage de la part de l'utilisateur.
- **Risques** :
  - connaissance du domaine : bonne
  - compétence designer : moyenne
  - compétence programmeurs : bonne
- **Importance du UC** : grande
- **Dictionnaire abstractions**: ListeFormations, Formation, Sessions, ListeClients, Inscription, ListeAttente, FicheInscriptionSession
- **Dictionnaire opérations** : consulterFormations, consulterSession, inscrireClient, rechercheClient, inscrireListeAttente

Exemple : Inscription à une formation

- **Tests** : on testera avec un nouveau client, un client existant, une formation sans session, une session non complète et une session complète.
- **Informations nécessaires** : les coordonnées du client (nom, prénom, date de naissance, n° carte national, adresse, téléphone/GSM, [Email], [coordonnées employeur]
- **Contraintes** : un minimum d'interactions et d'encodage de la part de l'utilisateur.
- **Risques** :
  - connaissance du domaine : bonne
  - compétence designer : moyenne
  - compétence programmeurs : bonne
- **Importance du UC** : grande
- **Dictionnaire abstractions**: ListeFormations, Formation, Sessions, ListeClients, Inscription, ListeAttente, FicheInscriptionSession
- **Dictionnaire opérations** : consulterFormations, consulterSession, inscrireClient, rechercheClient, inscrireListeAttente

Diagramme d'activités

- Introduction
- Objectifs
- Diagramme d'activités
- Etat action, état activité
- Sous-diagramme d'activités
- Couloirs d'un diagramme d'activités

Diagramme d'activités

- Apport en grande partie de la méthode OMT (Rumbaugh)
- Alors que les diagrammes d'interaction modélisent le flot de contrôle **entre objets**, le diagramme d'activités est utilisé pour modéliser le flot de contrôle **entre activités**
- Diagramme **très peu utilisé** car d'un formalisme peu adapté à la description de **gros** algorithmes
- Diagramme gagnant à être connu : possibilité de l'utiliser pour décrire des processus métier de haut niveau (= équivalent du MOT MERISE)

Diagramme d'activités

Objectifs

- Déterminer les activités induites par un **flot de contrôle transverse** au système

  La modélisation objet incite à utiliser la délégation entre objets et de ce fait rend difficile la lecture des flots de contrôle au travers du système

- Définir **avec précision les traitements** qui ont cours au sein du système

  Certains algorithmes ou calculs nécessitent de la part du modélisateur une description poussée

- Représenter les **interactions synchrones** au sein du système

  Certains types d'application développent de lourds algorithmes ou calculs séquentiels et n'ont pas recours à des communications asynchrones entre objets

Diagrammes d'activités (Définition)

Le diagramme d'activité est un **diagramme états-transitions simplifié** pour lequel les états se réduisent à de simples actions ou activités et dont les transitions se déclenchent **automatiquement** avec éventuellement des gardes

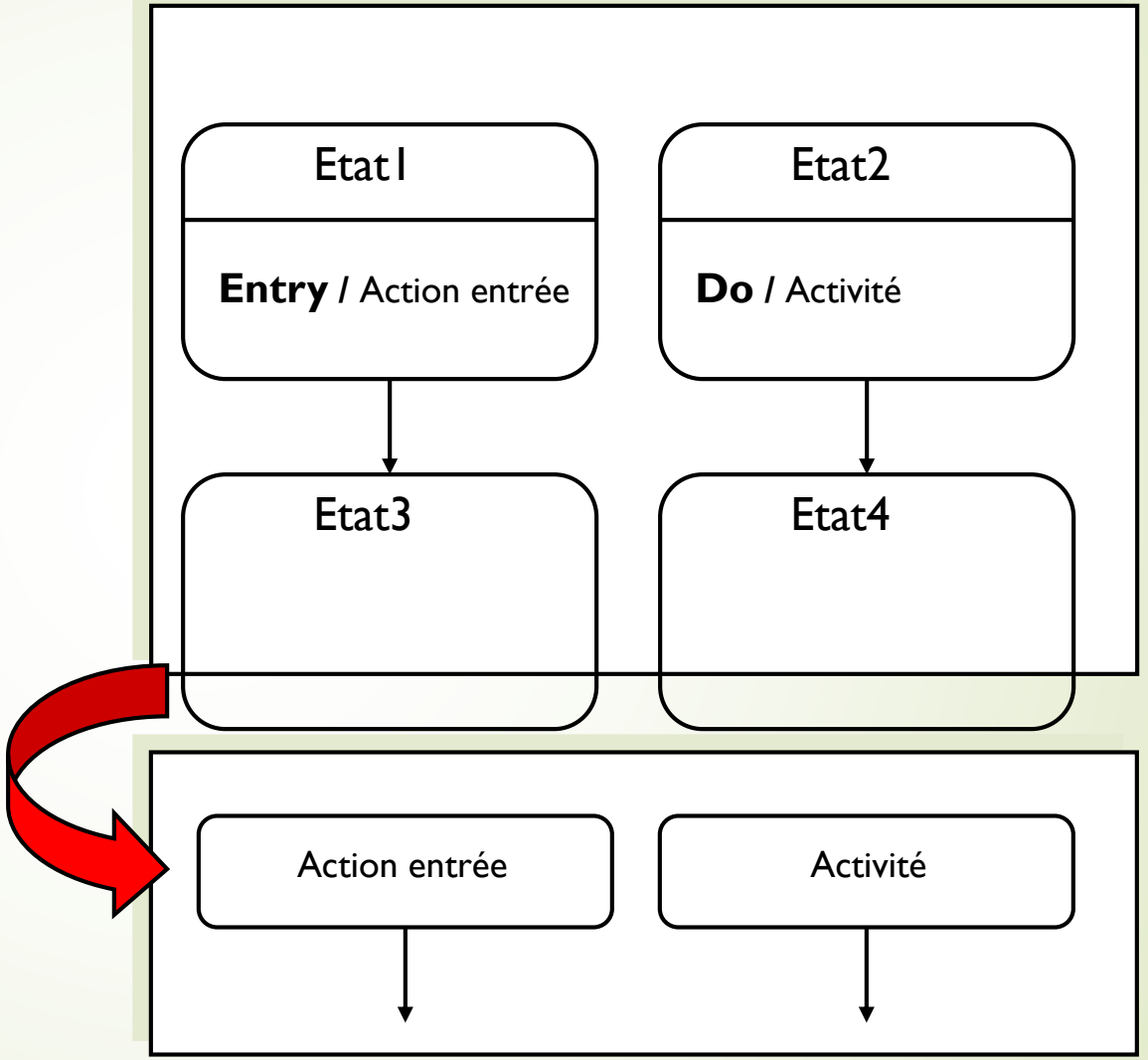
- Le diagramme d'activité est composé de deux sortes d'états :
  - Les **états d'action** ne contenant qu'une action en entrée
  - Les **états d'activité** ne contenant qu'une activité en leur sein
- Les notions d'action et d'activité dont il est question ici sont identiques à celles utilisés par les diagrammes états-transitions

Diagramme d'activités

- Le diagramme d'activités simplifie l'écriture des diagrammes états-transitions
- Un **état action** est étiqueté par le nom de l'action en entrée

  Il est **atomique** et **non redécomposable**
- Un **état activité** est étiqueté par le nom de l'activité

  Il n'est **pas atomique** et peut se **redécomposer** en un autre diagramme d'activités



Diagrammes d'activités

- Un diagramme d'activités peut être utilisé pour décrire une fonctionnalité induisant un flot de contrôle traversant le système

  En particulier, il est une alternative aux diagrammes d'interaction pour la **description d'un cas d'utilisation**

- Un diagramme d'activités peut être utilisé pour décrire avec précision le **contenu d'une opération d'une classe**

- Un diagramme d'activités peut être utilisé pour décrire avec précision une **activité** incluse dans un diagramme états-transitions

Diagramme d'activités Exemple

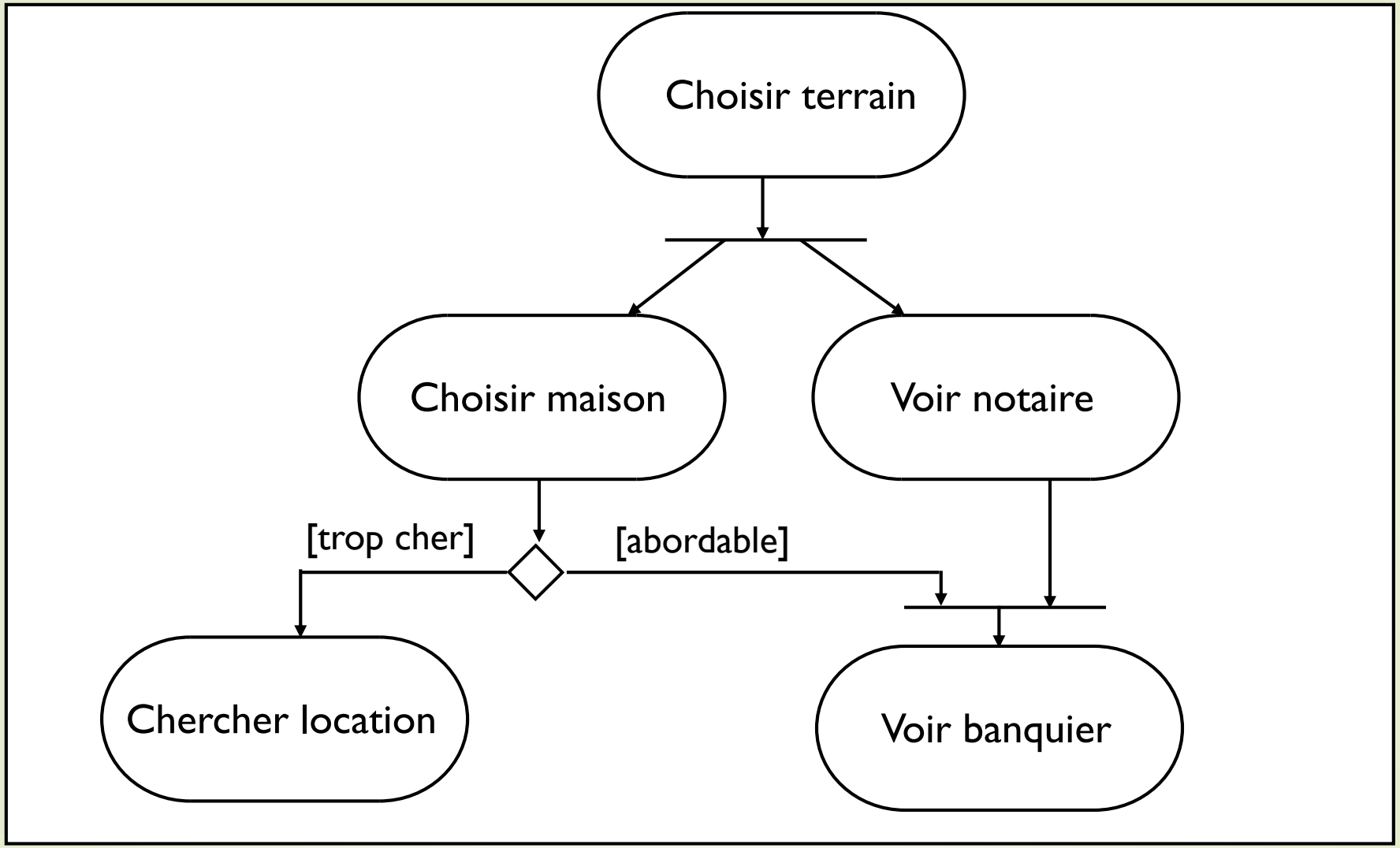
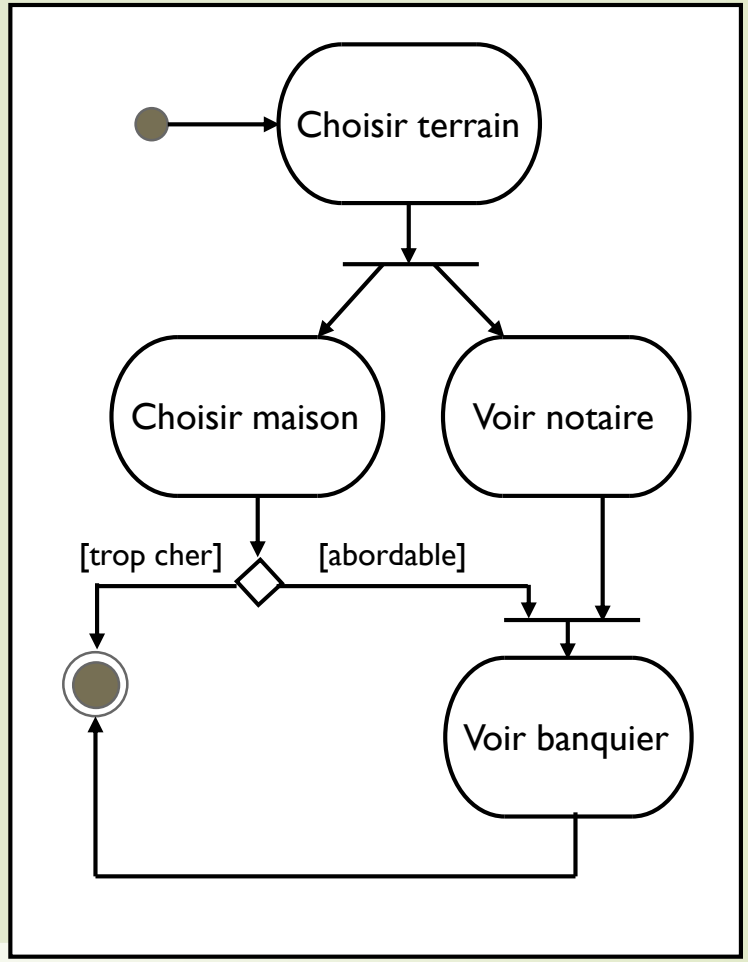


Diagramme d'activités

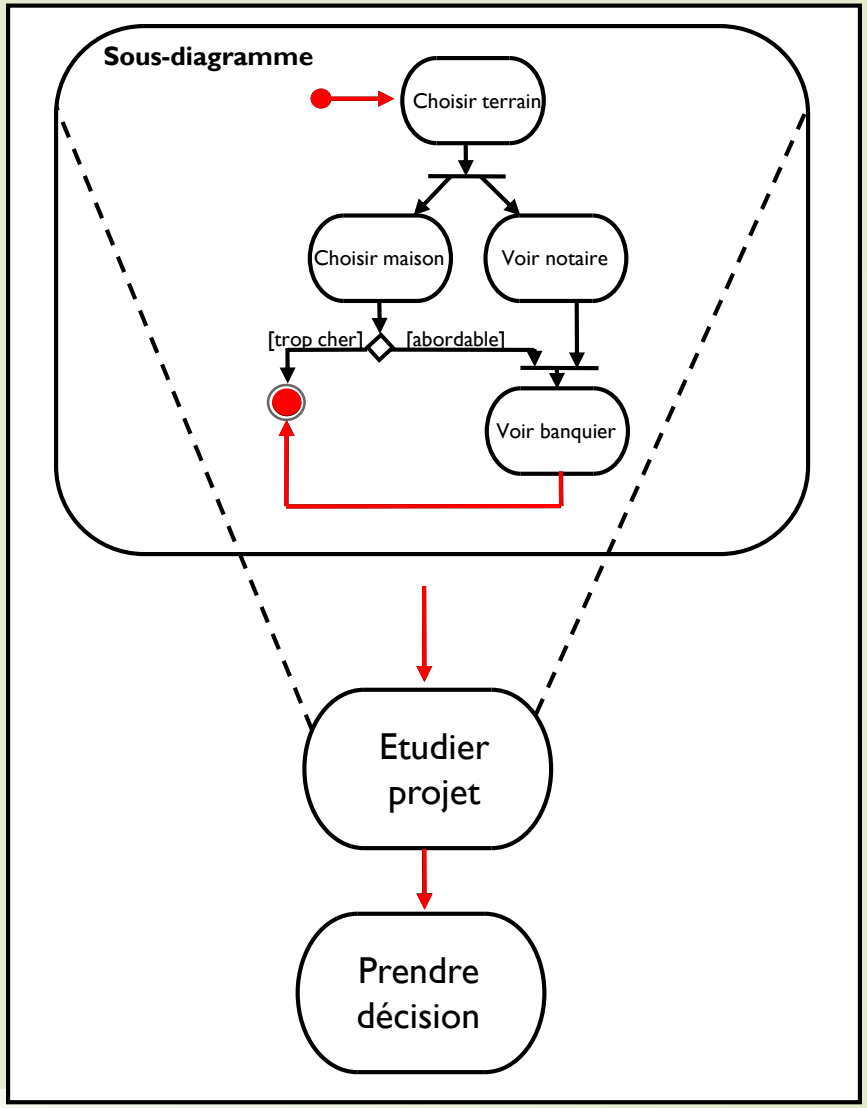
- Une **transition** sur un diagramme d'activités est représentée par une flèche éventuellement étiquetée par une garde
- Un **branchement conditionnel** est représenté par un losange d'où partent toutes les alternatives **obligatoirement exclusives**
- On utilise des **fourches** et des **jonctions** pour synchroniser les activités entre-elles
- **Etat initial** et **final** peuvent être représentés sur le diagramme



Sous-diagramme d'activités

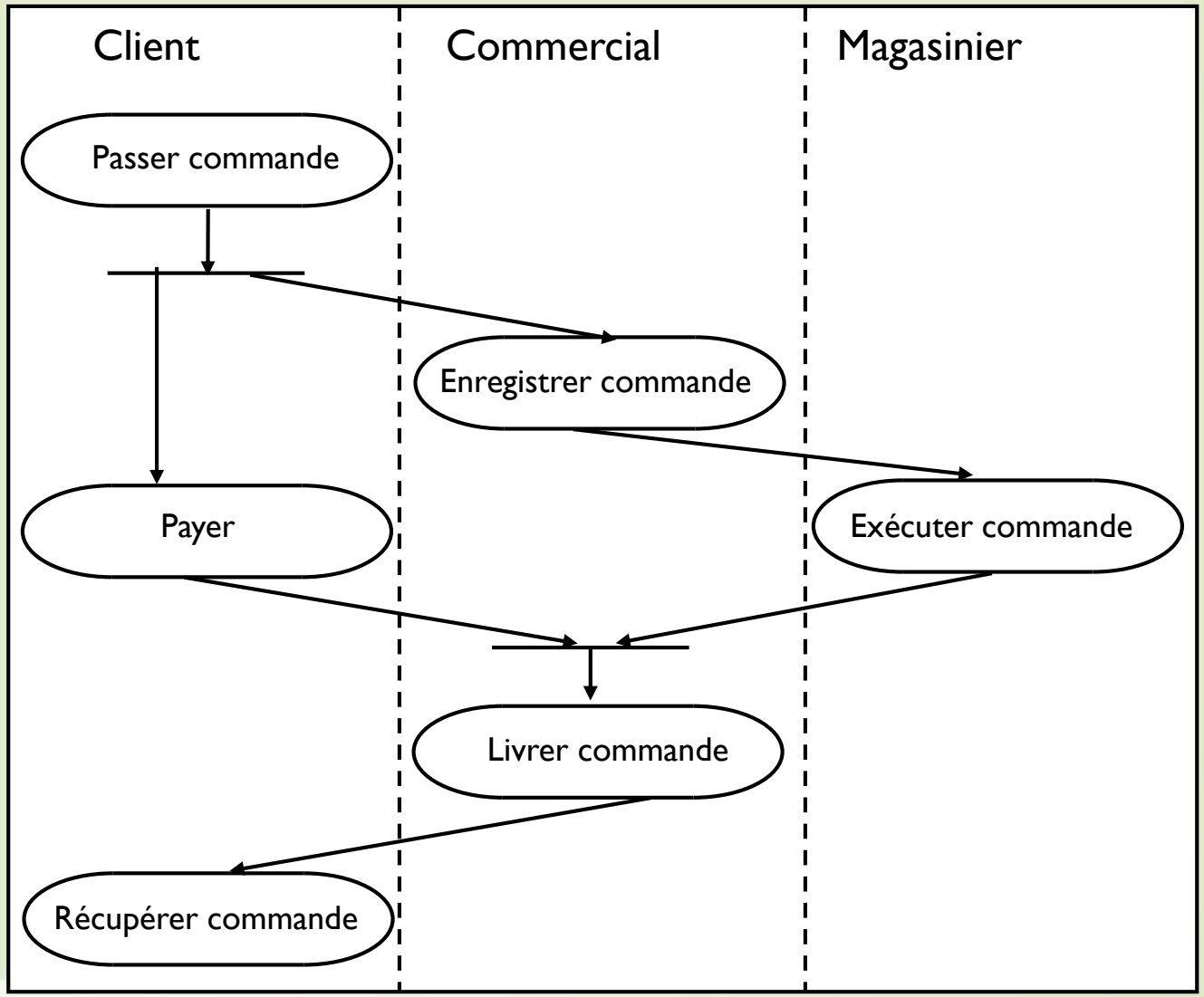
- Un état activité figurant sur un diagramme d'activité peut être redécomposé dans un sous-diagramme d'activité

- La terminaison du sous-diagramme entraîne le déclenchement de la transition en sortie de l'activité décomposée



Couloirs d'un diagramme d'activités

- Chaque couloir possède un nom
- Il n'est pas obligatoire que ce nom ait une sémantique particulière
- En général, un couloir correspond à une **classe** du système



Diagrammes d'activités Recommandations)

- Utiliser le diagramme d'activité pour décrire un processus métier de haut niveau (= équivalent du MOT MERISE)
- Ne pas utiliser le diagramme d'activités si l'on souhaite modéliser des **interactions asynchrones** entre objets
- Préférer le diagramme d'activités à un diagramme d'interaction pour décrire un cas d'utilisation **purement algorithmique (cas des batchs)**
- Privilégier l'utilisation d'un **pseudo-code** pour décrire les algorithmes trop imposants

Le but du diagramme d'activité

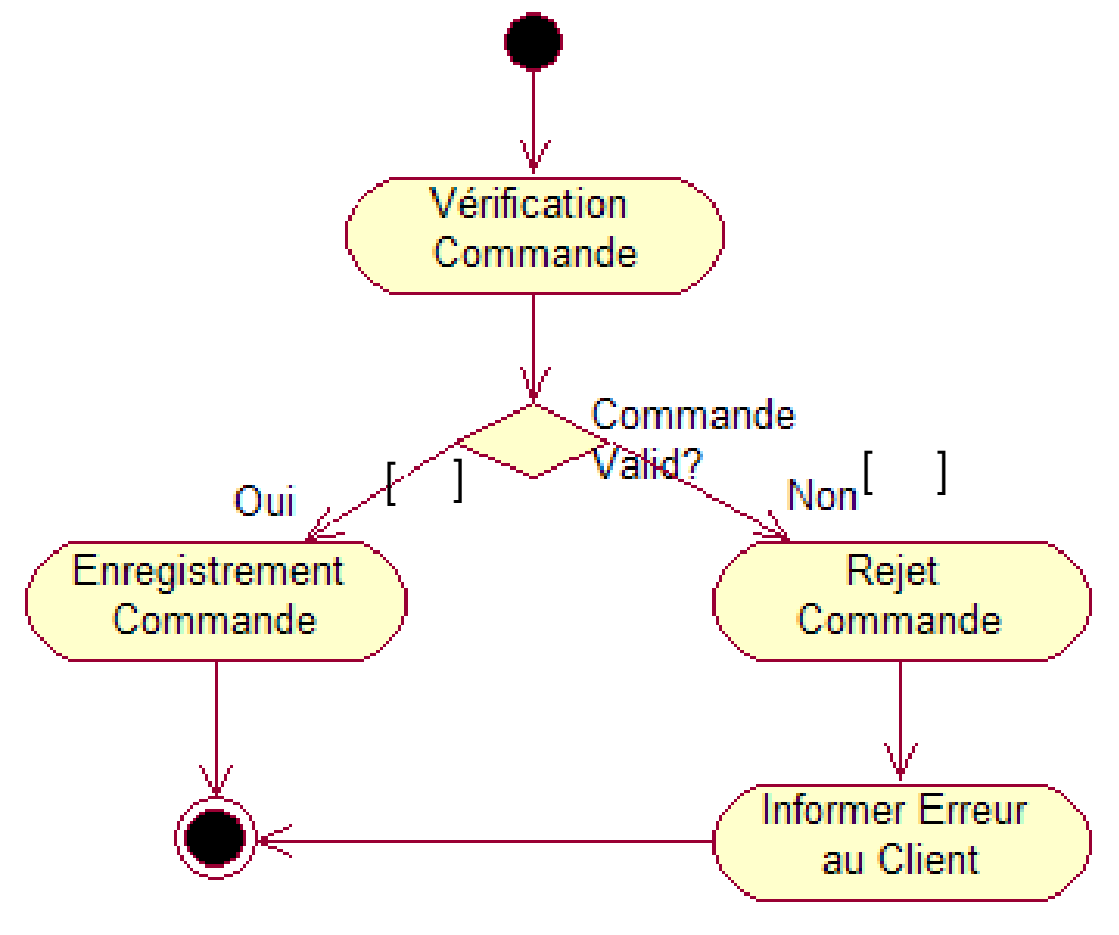
- Diagramme d'activité est utilisé pour:
  - Modéliser un workflow dans un use case ou entre plusieurs use cases.
  - Spécifier une opération (décrire la logique d'une opération)
- Le diagramme d'activité est le plus approprié pour modéliser la dynamique d'une une tâche, d'un use case lorsque le diagramme de classe n'est pas encore stabilisé.

Notion du diagramme d'activité

Diagramme d'activité =

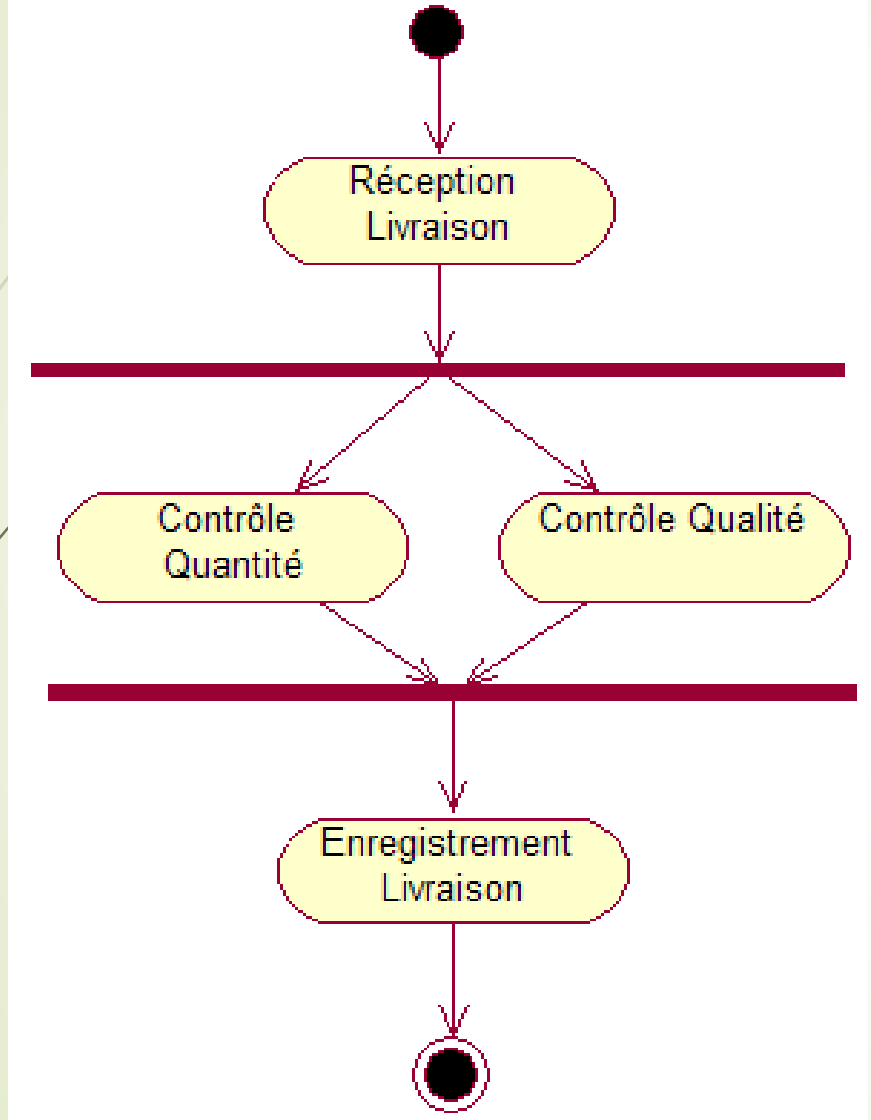
- ensemble d'activités liés par:
  - *Transition (sequentielle)*
  - *Transitions alternatives (conditionnelle)*
  - *Synchronisation (disjonction et conjonctions d'activités)*
  - *Itération*
- + 2 états: état de *départ* et état de *terminaison*
- *Swimlanes*: represente le lieu, le responsable des activités.

Notion du diagramme d'activité



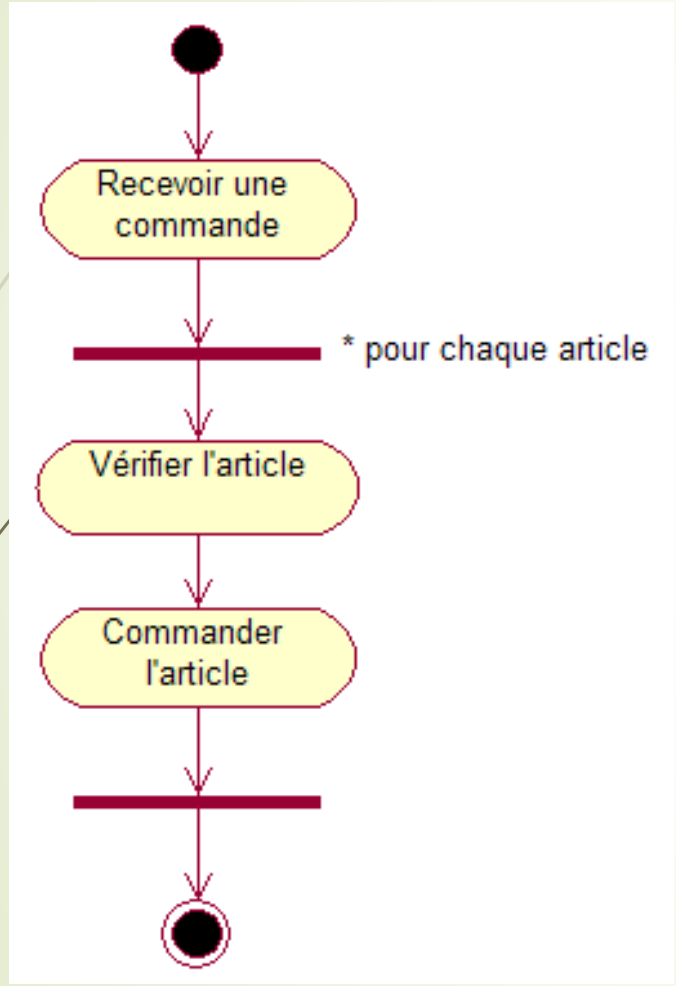
- Etat de départ
- Etat de terminaison
- Transition
- Transition Alternative

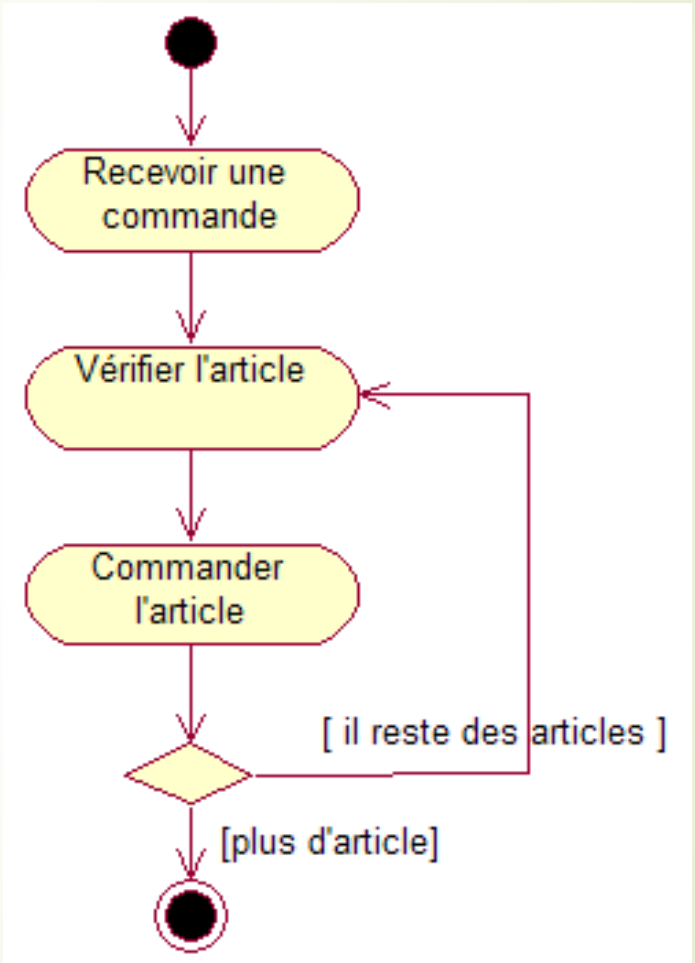
Notion du diagramme d'activité



Synchronisation disjonctive et conjonctive

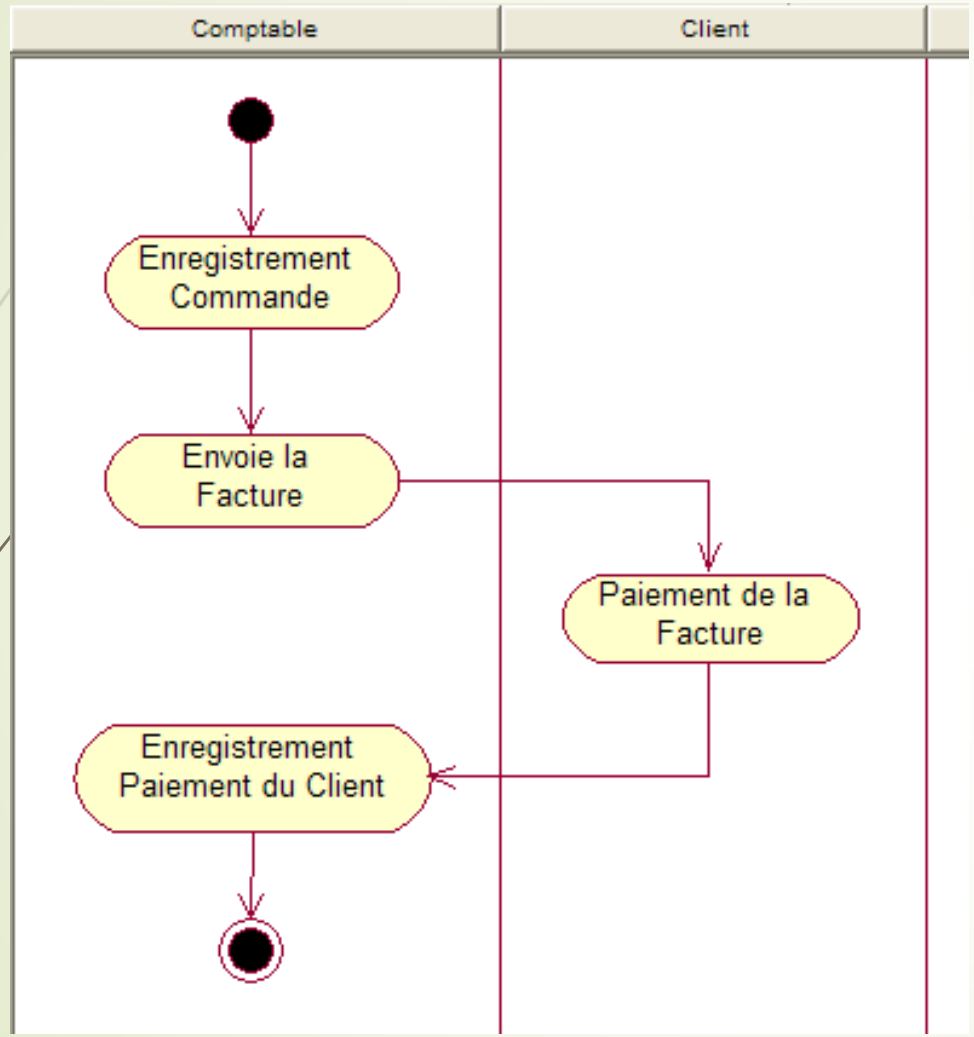
Notion du diagramme d'activité





Itération

Notion du diagramme d'activité



Swimlanes

Construction un diagramme d'activité

1. **Identifiez la portée (« scope ») du diagramme d'activité**
   Commencez en identifiant ce que vous allez modéliser. Un seul use case? Une partie d'un use case ? Un « workflow » qui inclut plusieurs use cases ? Une méthode de classe ?
2. **Ajouter l'état de *départ* et de *terminaison***
3. **Ajouter les activités**
   Si vous modélisez un use case, introduisez une activité pour chaque use case principal. Si vous modélisez un « workflow », introduisez une activité pour chaque processus principal, souvent un use case. Enfin, si vous modélisez une méthode, il est souvent nécessaire d'avoir une activité pour chaque grand étape de la méthode.
4. **Ajouter des transitions (séquentielles), des transitions alternatives (conditionelles), des synchronisations entre des activités, des itérations.**
5. **Identifier des swimlanes et répartir des activités identifiées dans ces swimlanes.**