Le langage de modélisation UML

Objectifs de ce cours de modélisation orientée objet

- Introduire la modélisation orientée objet
- Introduire la modélisation à base de graphiques des systèmes informatiques
- Introduire la notation UML
  - Différents types de diagrammes avec leurs notations
  - Rôles complémentaires des types de diagrammes
  - Cohérence entre diagrammes de même type ou de types différents
- Présenter des éléments méthodologiques d'utilisation des différents types de diagrammes dans un processus de développement
  - Présentation dans le cours d'une première étude de cas
  - Mise en pratique lors des bureaux d'étude avec deux autres études de cas
  - Évaluation de l'acquisition lors d'un examen sur table avec une quatrième étude de cas

Généralités sur la modélisation orienté objet et sur UML

- Principes de la modélisation

- Pourquoi et comment modéliser en orienté objet

- Unified Modelling Language (UML)

- Cinq façons de voir un système informatique : les 4+1 vues de Kruchten

- Phases de la modélisation, cycle en V

- Rôle de l'expression des besoins

- Rôle de l'analyse

- Rôle de la conception

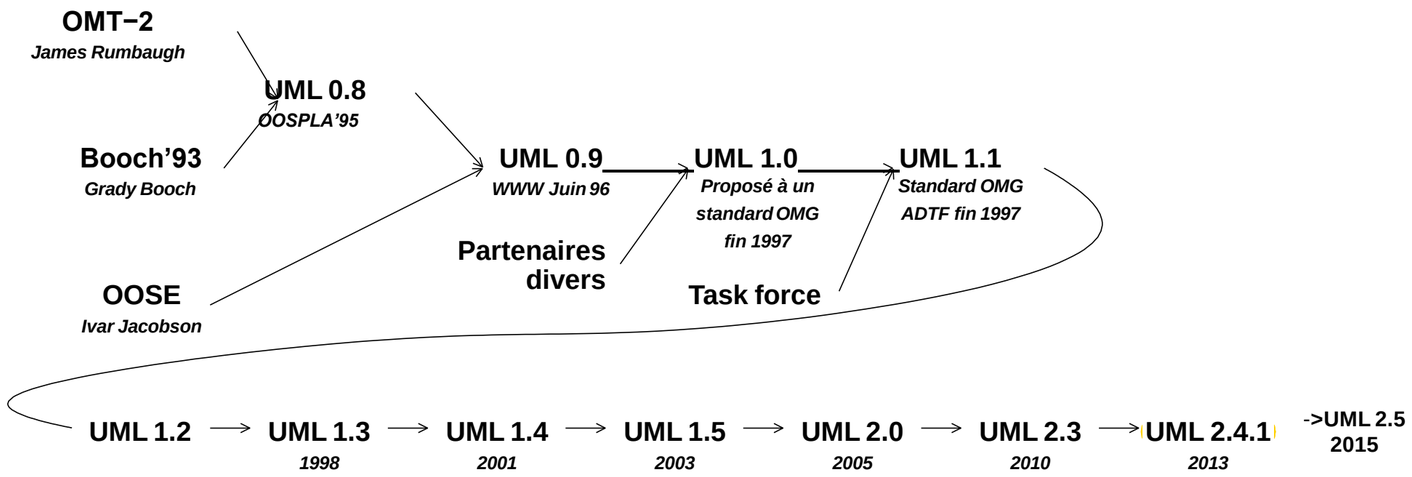
Principes de la modélisation

- Objectif principal de la modélisation = maîtriser la complexité
- Modéliser = abstraire la réalité pour mieux comprendre le système à réaliser / réalisé
- Le modèle doit être relié au monde réel
  - Par exemple : l'existant avant les travaux, le réalisé, le restant à réaliser
- Un modèle peut être exprimé avec différents niveaux d'abstraction / raffinement
  - Par analogie : répartition électrique de l'immeuble, de la cage d'escalier, de l'appartement, de la pièce
- Une seule « vue » du système n'est pas suffisante
  - Les intervenants multiples du projet informatique possèdent des préoccupations multiples
    - Par analogie : plan de masse, vues de face et de côté, schéma électrique, plan de plomberie, plan de calculs de construction

Pourquoi et comment modéliser en orienté objet

- Relier le modèle au monde réel par la notion d'objet
- Orienté objet = abstraire et décomposer le système informatique en objets
  - Le monde réel est constitué d'objets physiques ou immatériels
  - Tracer les objets virtuels de modélisation depuis les objets du monde réel
    - Relier les objets (réels) du problème et les objets (virtuels) de la solution
  - Favoriser les abstractions naturelles du monde réel utilisables en modélisation
    - Objets vus comme des « boîtes noires » : seules les propriétés visibles de l'extérieur intéressent
    - Objets possédant un nom, qualifiables, classables, polymorphes, dé-/composables, interagissants avec d'autres objets, etc.
- Objectifs supplémentaire lors de la modélisation orientée objet
  - Meilleure indépendance du modèle par rapport aux fonctions demandées
  - Meilleure capacité d'adaptation et d'évolution du modèle lorsque des fonctionnalités sont modifiées ou ajoutées

Unified Modelling Language (UML)

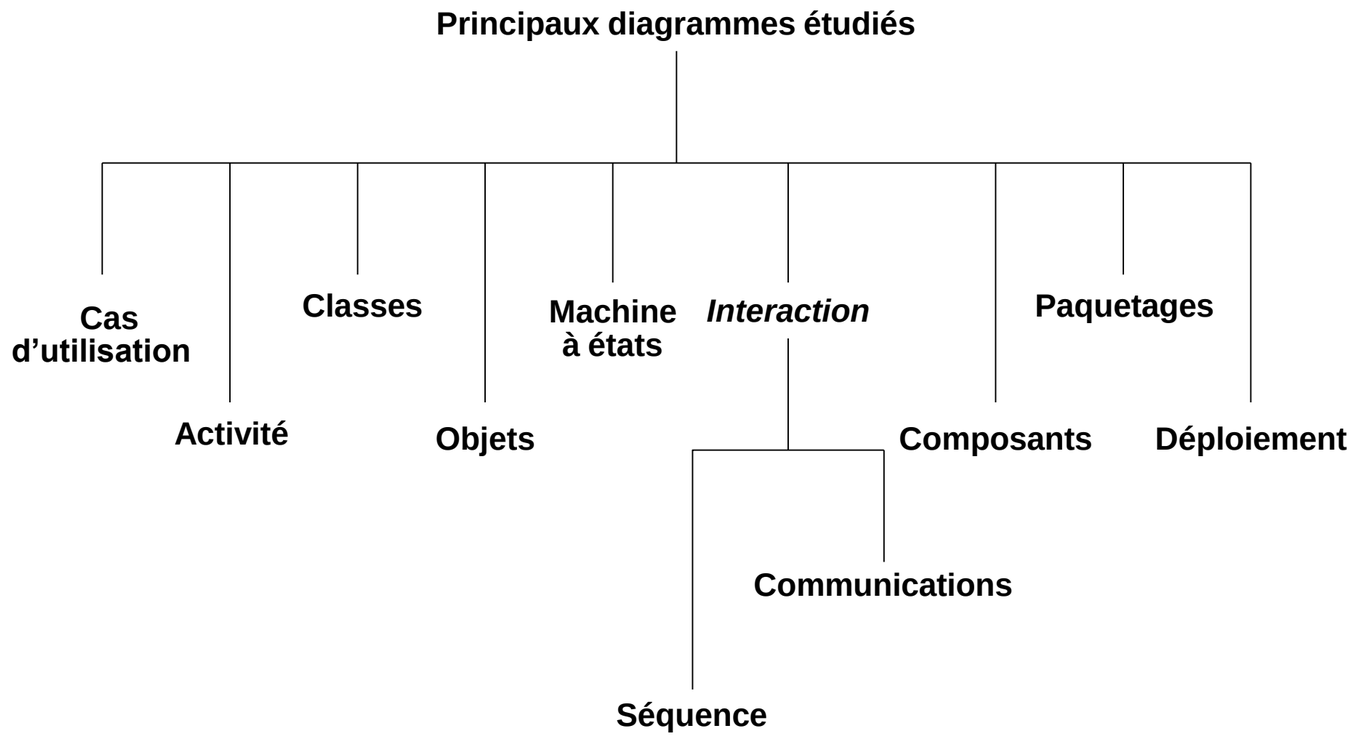


- Systèmes d'informations des entreprises, Banques et services financiers, Télécommunications, Transports, Défense et aérospatiale, Calculs scientifiques, Applications réparties en réseau, services Web Internet, etc.

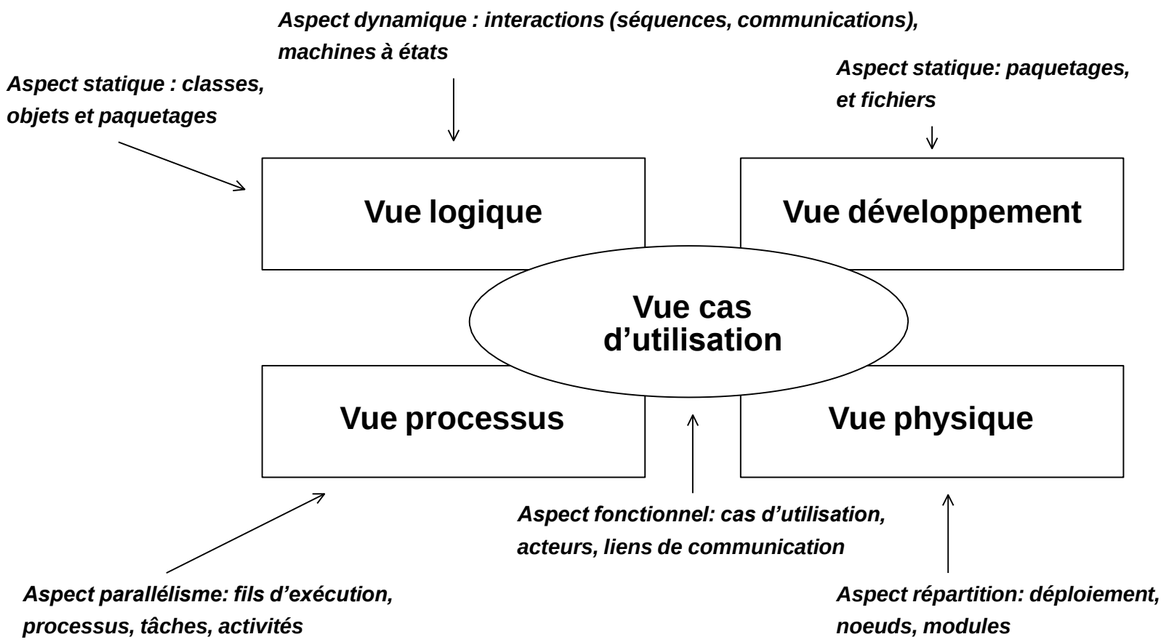
UML

- UML est un langage de modélisation orientée objet
- UML n'est pas une méthode
- UML a été adopté par toutes les méthodes orientées objet
- UML est dans le domaine public ; c'est un standard
- UML est un langage pour :
  - Visualiser
    - Chaque symbole graphique possède une sémantique
  - Spécifier
    - De manière précise et complète, sans ambiguïté
  - Construire
    - Une partie du code des classes peut être généré automatiquement
  - Documenter
    - Les différents diagrammes, notes, contraintes, exigences sont conservés dans un document

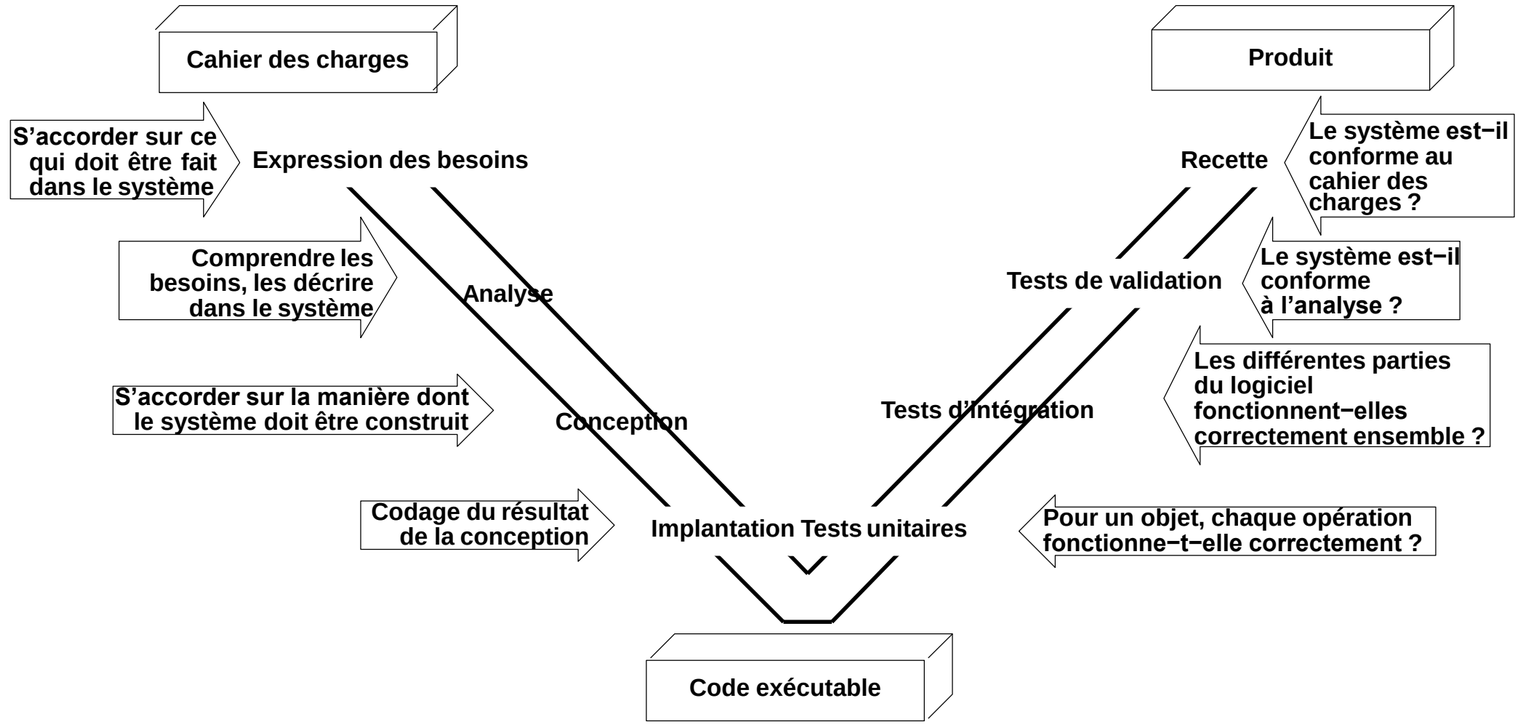
Les 10 principaux diagrammes UML



Cinq façons de voir un système informatique



Phases de la modélisation, cycle en V



Rôle de l'expression des besoins

- Permettre une meilleure compréhension du système
- Comprendre et structurer les besoins du client
  - Clarifier, filtrer et organiser les besoins, ne pas chercher l'exhaustivité
- Une fois identifiés et structurés, ces besoins :
  - Définissent le contour du système à modéliser
    - Précisent le but à atteindre
  - Permettent d'identifier les fonctionnalités principales du système

Exemple de cahier des charges : Studs

- Studs : Application d'aide à la planification de réunions et à la prise de décision.
- L'objectif est de permettre d'exprimer des préférences parmi plusieurs choix.
- Les choix sont de deux types : *(1)* des plages horaires (dates avec heures de début et de fin) et *(2)* des votes concernant une liste de choix.
- L'organisatrice, crée un scrutin, renseigne les plages horaires possibles (type 1) et ajoute les participants à la réunion.
- Les participants peuvent exprimer dans un bulletin leurs préférences en indiquant pour chaque plage horaire s'ils votent « pour » (ils sont disponibles et annoncent leur intention de participer) ou « contre ».
- L'organisatrice récupère les résultats du scrutin à la fin du vote et annonce la plage horaire choisie.
- La procédure est similaire pour le type 2, c.-à-d. pour les scrutins concernant des propositions (chaînes de caractères quelconques) à choisir.

Règles de gestion et contraintes de l'application Studs

- Toutes les personnes peuvent être organisatrices.
- L'organisatrice est *de facto* une participante au scrutin.
- Seule l'organisatrice est autorisée à gérer un scrutin.
- Seuls les participants enregistrés peuvent participer au scrutin et consulter les résultats.
- Pour que les participants puissent voter, il faut que le scrutin soit ouvert (dateDuJour ≥ dateDebutScrutin).
- La durée d'ouverture du scrutin est limitée.
- L'organisatrice doit indiquer la date de destruction automatique du scrutin.
- Toutes ces dates permettent de gérer de manière automatique le cycle de vie d'un scrutin.
- Les transitions du cycle de vie peuvent aussi être effectuées à la demande de l'organisatrice.

Rôle de l'analyse

- Le but de l'analyse est de traduire dans un langage qui se rapproche doucement de celui des informaticiens les modèles exprimés dans l'expression des besoins
- Cependant, pour rester compréhensible par les clients ou utilisateurs, elle ne prend en considération que des entités du domaine (métier)
- Elle sert d'interface, avec l'expression des besoins, aux dialogues avec les clients et les utilisateurs
- L'analyse doit servir de support pour la conception, l'implantation et la maintenance
- Le modèle de l'analyse décrit le problème (ce que doit faire le système et comment il le fait tel que vu d'un point de vue métier) sans spécifier la solution technique (avec les canevas logiciels)
  - ♦ Analyse = LE-QUOI

Rôle de la conception

- Le but de la conception est de fixer les choix techniques et de préparer l'implantation
- Le modèle de la conception décrit la solution (comment le problème est résolu)
  - ♦ Conception = LE-COMMENT
- La conception doit servir de support pour l'implantation et la maintenance
- Le plus souvent, le modèle de la conception n'est pas destiné à être compréhensible par les utilisateurs mais par les développeurs

Un peu de réflexion !

1. Le code source d'une application est-il un modèle de l'application ?

2. UML est-il un processus de développement ?

3. Un client demandant une informatisation est-il censé comprendre les diagrammes UML ?

4. Un développeur / programmeur est-il censé comprendre les diagrammes UML ?

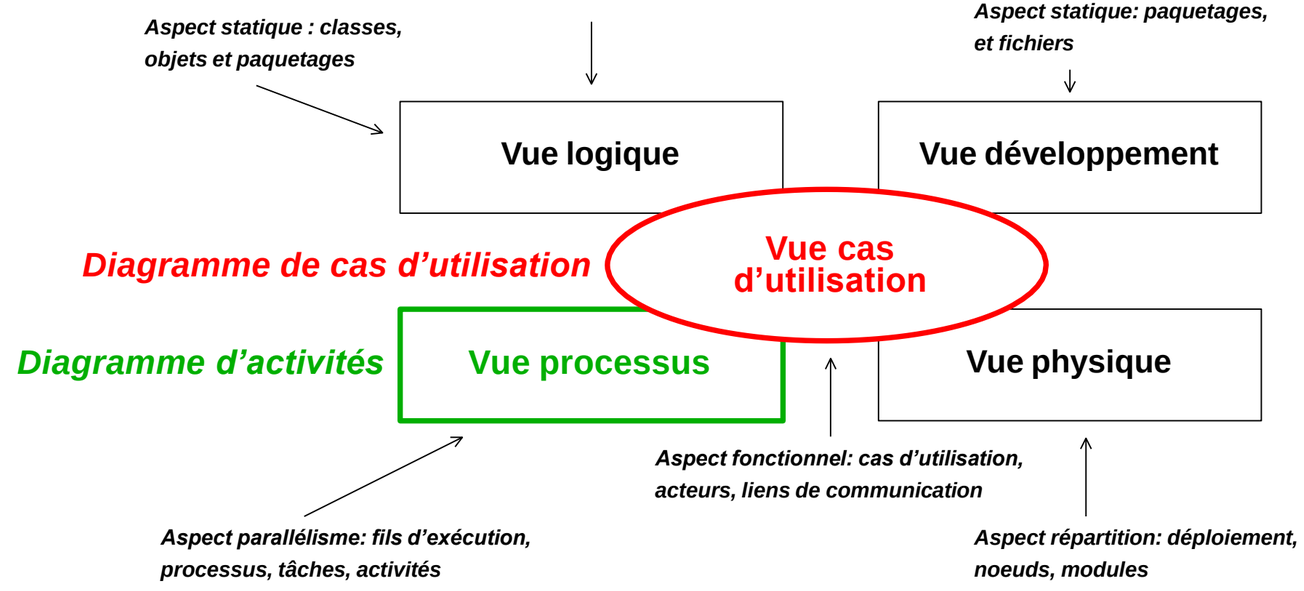
5. La phase d'expression des besoins est-elle étudiée dans ce module ?

3 Analyse, vues cas d'utilisation et processus

Modèle de l'analyse

- Vision de l'extérieur du système, le problème :
  - Vue cas d'utilisation = ce que fait le système, les fonctionnalités
  - Vue processus = ce que fait le système, les règles de gestion

Aspect dynamique : interactions (séquences, communications), machine à états

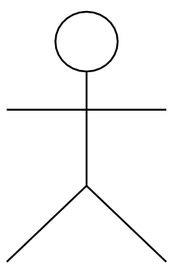


Introduction au Diagrammes de cas d'utilisation

- Les fonctionnalités sont modélisées par des cas d'utilisation
- Un diagramme de cas d'utilisation définit :
  - Le système
  - Les acteurs
  - Les cas d'utilisation (fonctionnalités)
  - Les liens entre acteurs et cas d'utilisation
    - Quel acteur accède à quel cas d'utilisation ?
- Un modèle de cas d'utilisation se définit par :
  - Des diagrammes de cas d'utilisation
  - Une description textuelle des scénarios d'utilisation

Acteur

- Un acteur représente une personne, un périphérique ou un autre système qui joue un rôle (interagit) avec le système
  - « L'organisatrice, crée un scrutin, renseigne les plages horaires possibles et ajoute les participants à la réunion. »
  - « Les participants peuvent exprimer leurs préférences en indiquant pour chaque plage horaire s'ils votent "pour" (ils sont disponibles et annoncent leur intention de participer) ou "contre". »

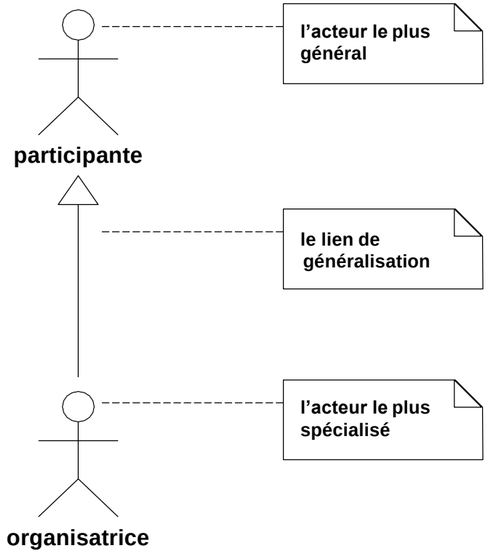


organisatrice

<< acteur >>
organisatrice

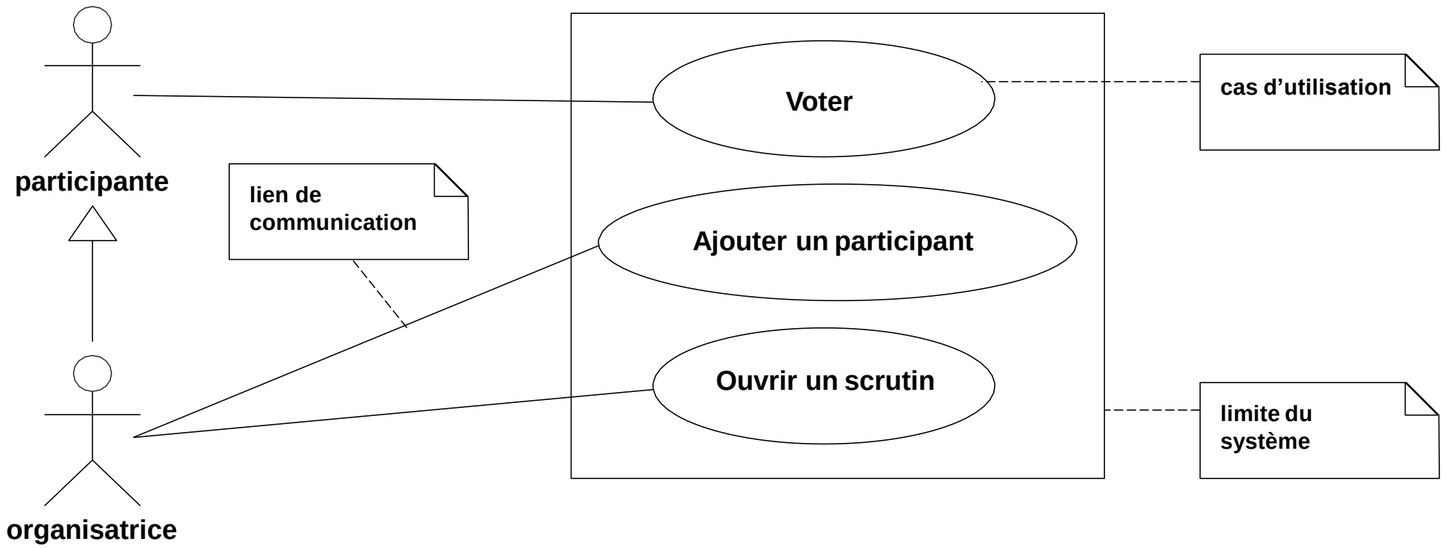
Relation de généralisation spécialisation entre acteurs

- La généralisation spécialisation est aussi appelée héritage (EST-UN)
  - « L'organisatrice participe au scrutin qu'elle gère. »
    - Une organisatrice est une participante

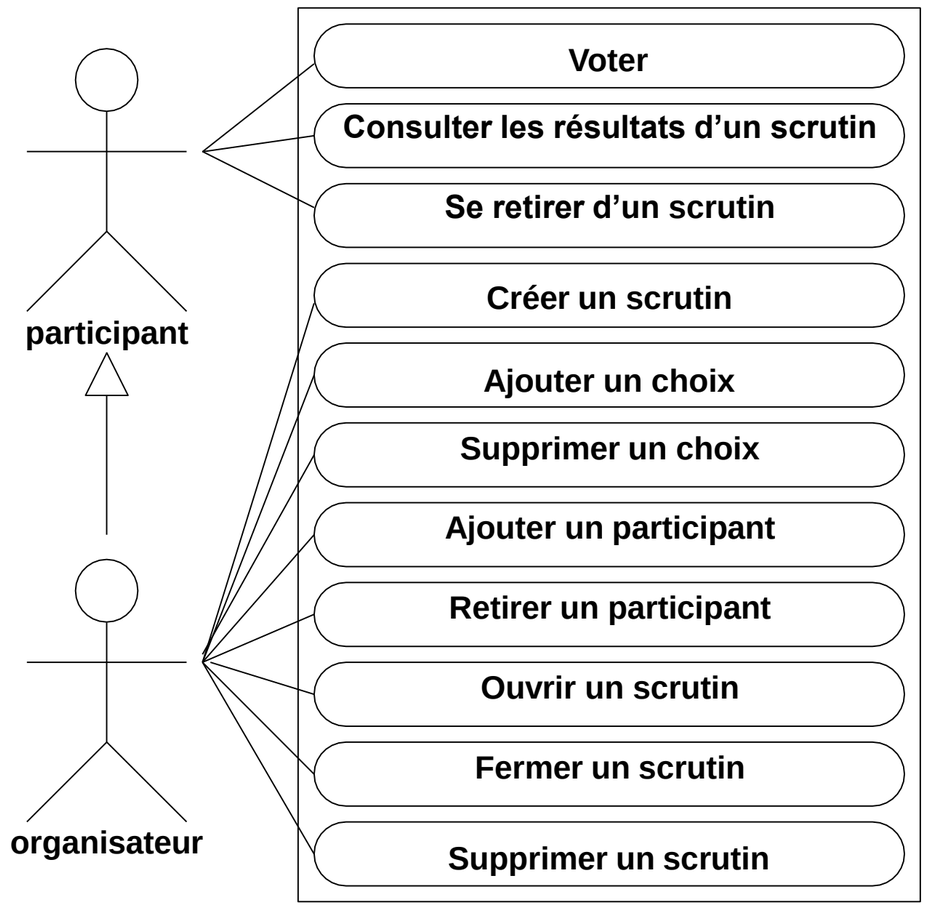


Cas d’utilisation, lien de communication et système

- Un cas d’utilisation est un moyen de représenter les différentes possibilités d’un système
- Il correspond à une suite d’interactions entre un acteur et le système
- Il définit une fonctionnalité utilisable par un acteur

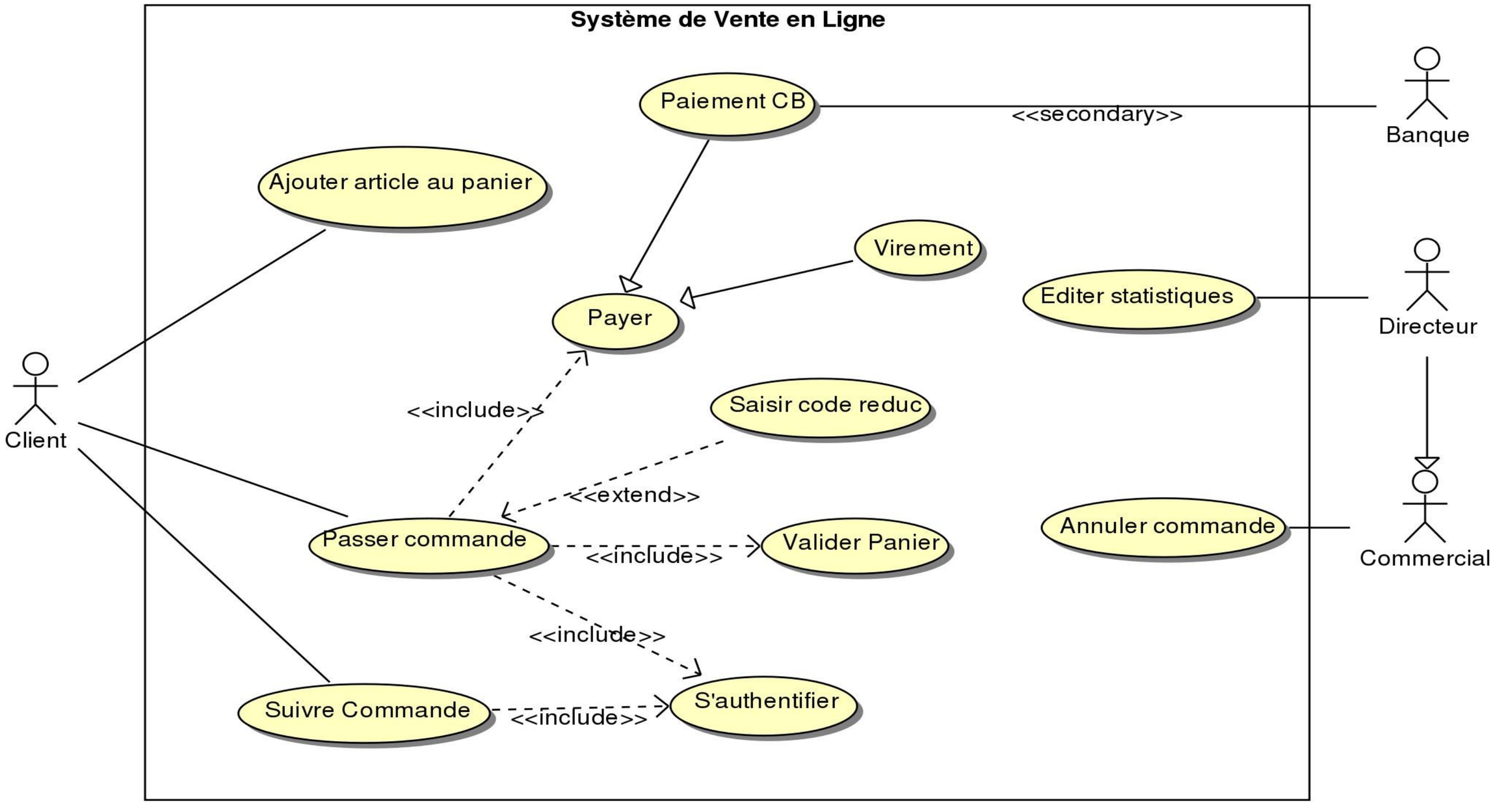


Exemple de diagramme de cas d'utilisation



Éléments de méthodologie

- Identifier les acteurs qui utilisent, gèrent et exécutent des fonctionnalités spécifiques
- Organiser les acteurs par relation de généralisation spécialisation si c'est pertinent
- Pour chaque acteur, rechercher les cas d'utilisation du système


Système de Vente en Ligne
Paiement CB
<<secondary>>
Banque
Ajouter article au panier
Virement
Editer statistiques
Directeur
Payer
<<include>>
Saisir code reduc
Client
<<extend>>
Passer commande
Valider Panier
<<include>>
Annuler commande
Commercial
<<include>>
Suivre Commande
<<include>>
S'authentifier

Description des cas d'utilisation

- Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation.

- Un simple nom est tout à fait insuffisant pour décrire un cas d'utilisation.

- Chaque cas d'utilisation doit être documenté pour qu'il n'y ait aucune ambiguïté concernant son déroulement et ce qu'il recouvre précisément.

Description textuelle

Identification :

- Nom du cas : Payer CB
- Objectif : Détailler les étapes permettant à un client de payer par carte bancaire
- Acteurs : Client, Banque (secondaire)
- Date : 18/09
- Responsables : Toto Version : 1.0

Description textuelle

Séquencements :

Le cas d'utilisation commence lorsqu'un client demande le paiement par carte bancaire

Pré-conditions

Le client a validé sa commande

Enchaînement nominal

1) Le client saisit les informations de sa carte bancaire Le système véri e que le numéro de CB est correct.
2) Le système véri e la carte auprès du système bancaire
3) Le système demande au système bancaire de débiter le client
4) Le système noti e le client du bon déroulement de la transaction

Enchaînements alternatifs

En (2) : si le numéro est incorrect, le client est averti de l'erreur, et invité à recommencer
En (3) : si les informations sont erronées, elles sont re-demandées au client

Post-conditions

- La commande est validée
- Le compte de l'entreprise est crédité

Description textuelle

Rubriques optionnelles

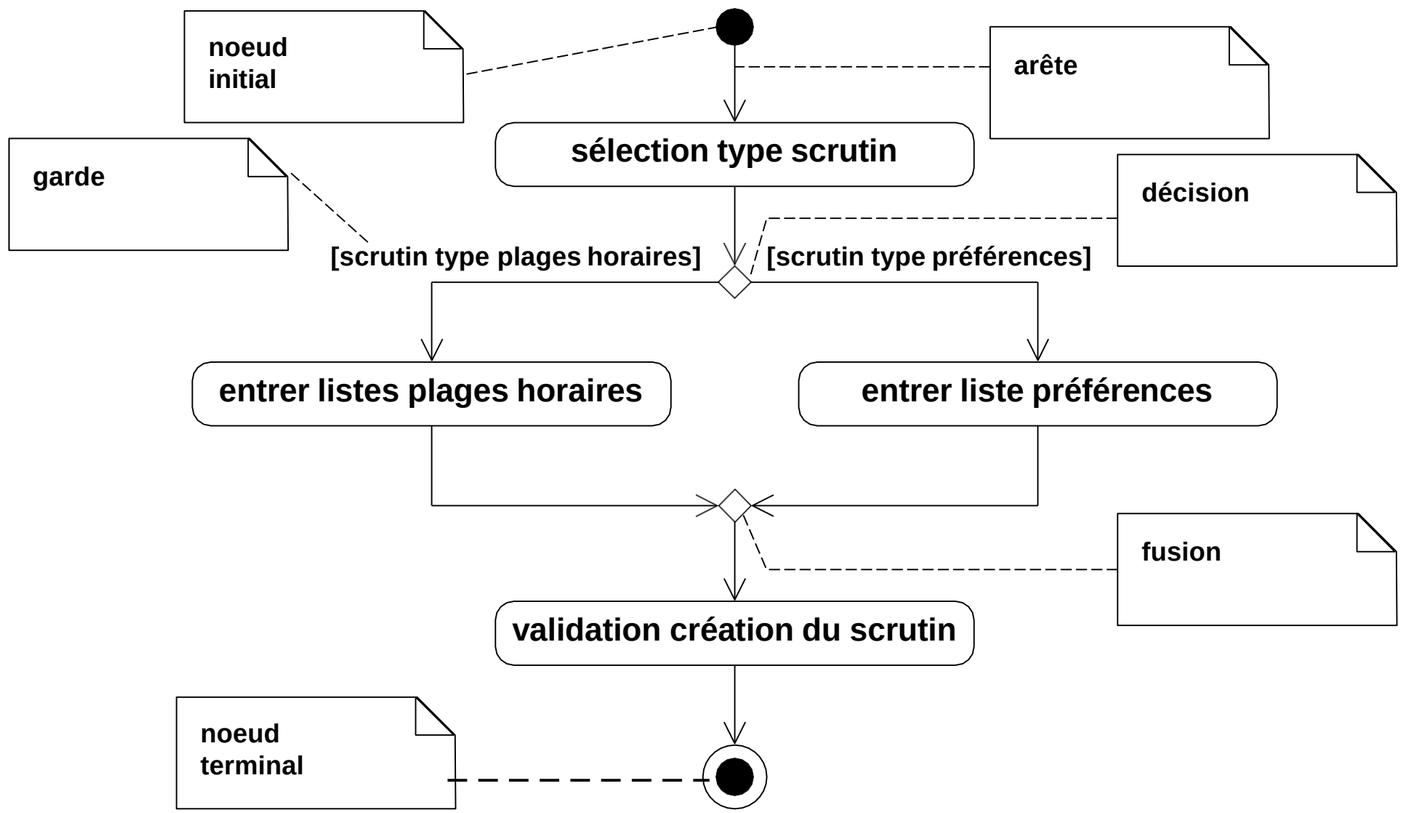
* Contraintes non fonctionnelles :
  - Fiabilité : les accès doivent être sécurisés
  Confidentialité : les informations concernant le client ne doivent pas être divulgués

* Contraintes liées à l'interface homme-machine :
  - Toujours demander la validation des opérations bancaires

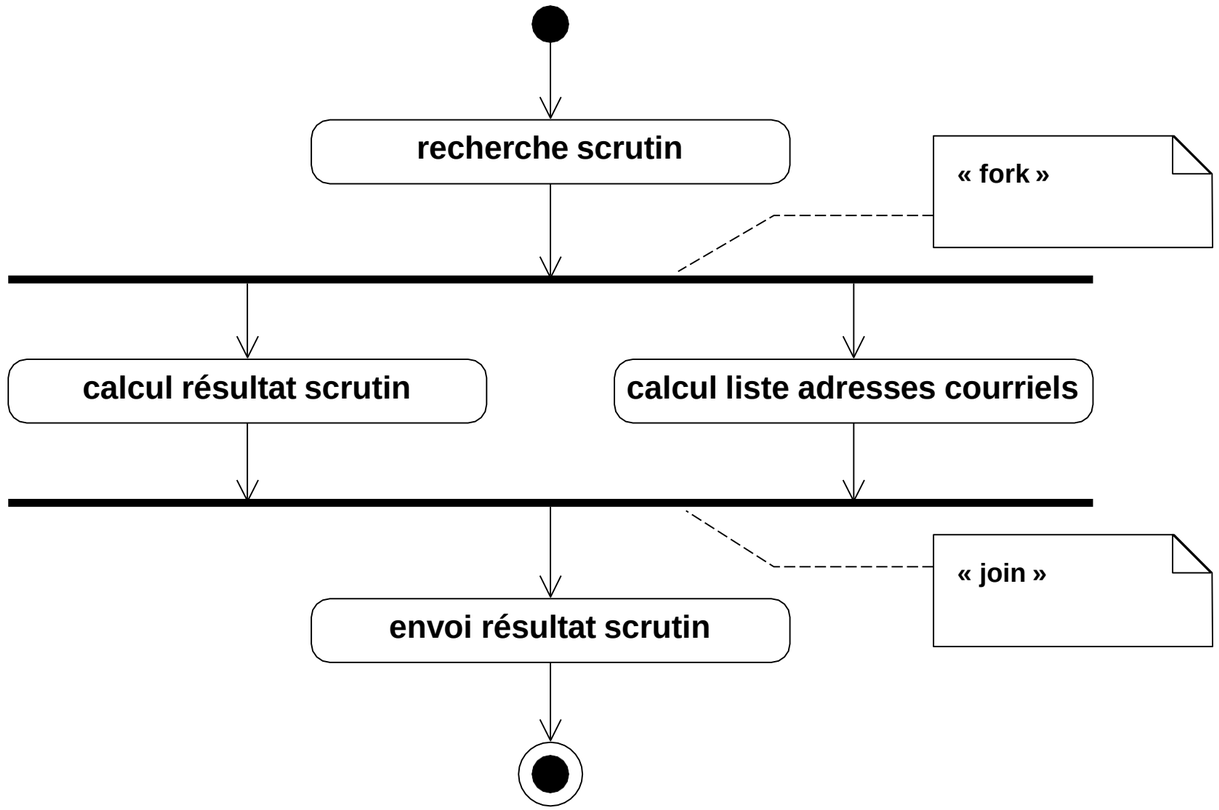
Scénarios d'un cas d'utilisation

- La description d'un cas d'utilisation se fait par des scénarios qui définissent la suite logique des actions qui constituent ce cas
- Il est possible de définir des scénarios simples ou des scénarios plus détaillés faisant intervenir les variantes, les cas d'erreurs, etc.
- La description du scénario précise ce que fait l'acteur et ce que fait le système
- En plus de descriptions textuelles, le scénario peut être représenté en utilisant un diagramme d'activité
- Le diagramme d'activité permet de :
  - Présenter « l'algorithme », c'est-à-dire les étapes de la fonctionnalité
  - Visualiser l'aspect temporel des interactions
  - Connaître le sens des interactions (acteur vers système ou inverse)
  - Distinguer le cas nominal des variantes (p.ex. avec traitement des erreurs)

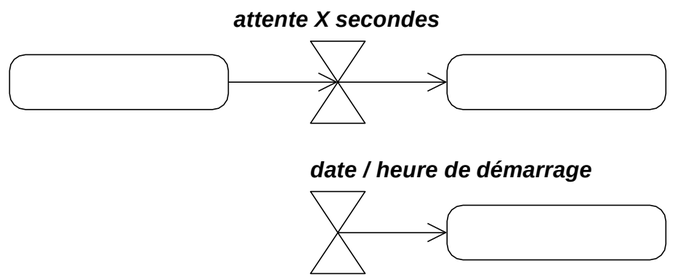
Actions et choix

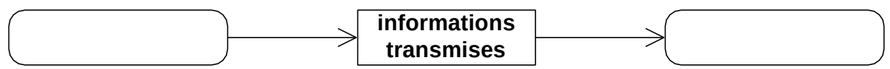


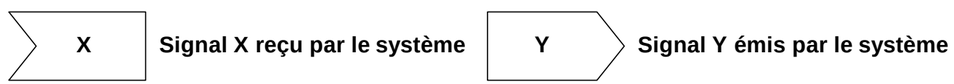
Concurrence



Autres notations du diagramme d’activité







Un peu de réflexion !

1. Dans un diagramme de cas d'utilisation, est-il pertinent d'ajouter des cas d'utilisation ne faisant pas partie du système ?

2. Quelles entités peuvent être dessinées dans un diagramme de cas d'utilisation ?
   (a) système
   (b) action
   (c) acteur
   (d) généralisation spécialisation

3. Dans une généralisation spécialisation entre acteurs, l'acteur spécialisé a-t-il accès à toutes les fonctionnalités de l'acteur généralisé ?