

# Analyse et modélisation

Séance 2.1: Modélisation des données dans le cycle de développement d'un logiciel \*

420-A56-GG

Par

Ange Adrienne Nyamen Tato

*\*inspiré du livre « Modéliser les données » de Céline Labbé.*

# Objectifs

- Maîtriser les cycles de développement d'un système
- Maîtriser le Processus de développement logiciel 2TUP
- Connaître l'importance de la modélisation et les types de modélisation

- Pourquoi est-il important de bien concevoir une base de données ?
- Utilisation de modèles

**Modèle** : Représentation abstraite de "quelque chose" de réel.

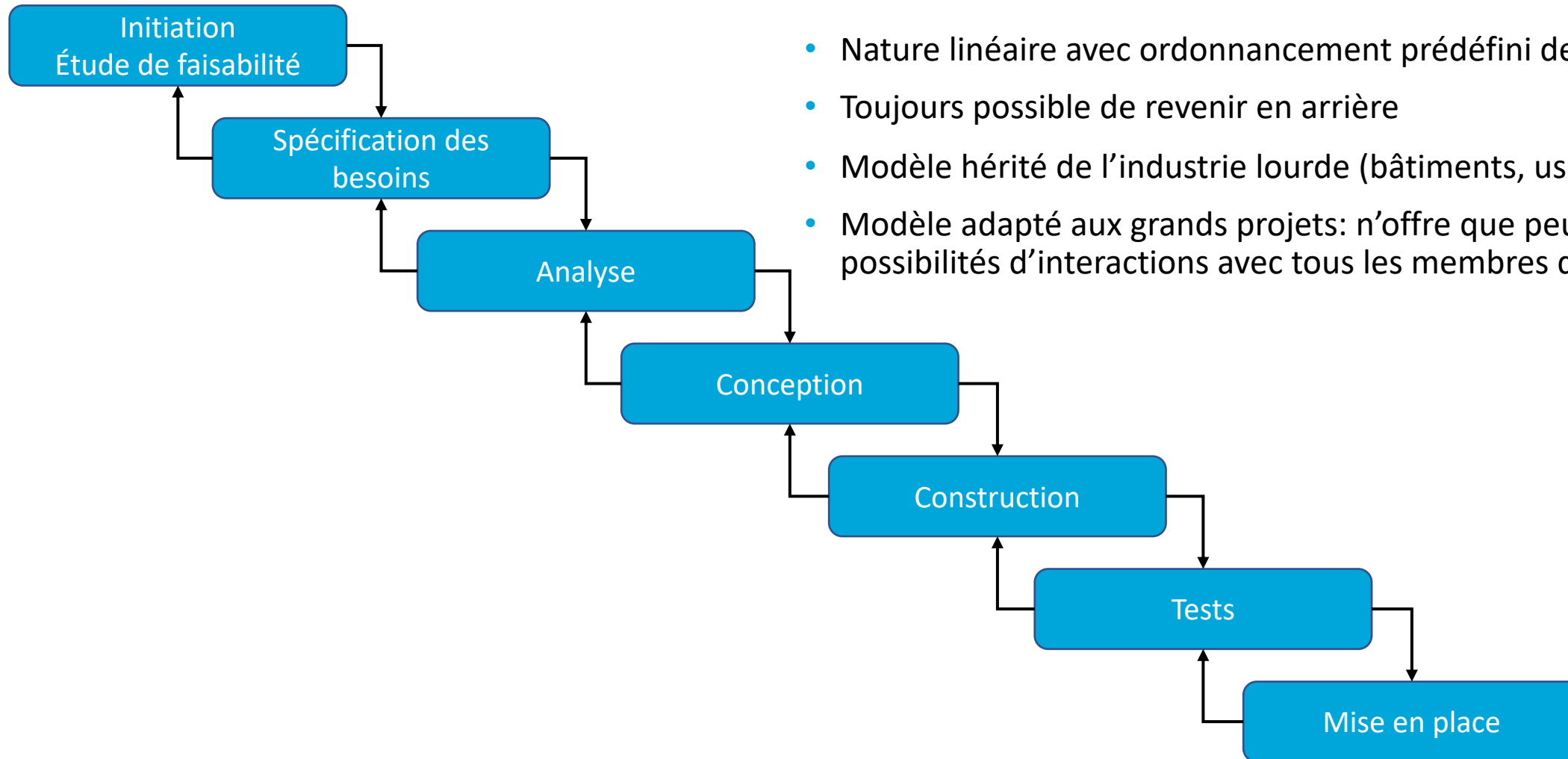
Exemples de modèles pour composantes du système:

- Flux de données
- Entité-relation
- Diagramme de cas d'utilisation
- Diagramme de classes

# Comprendre le cycle de développement d'un système

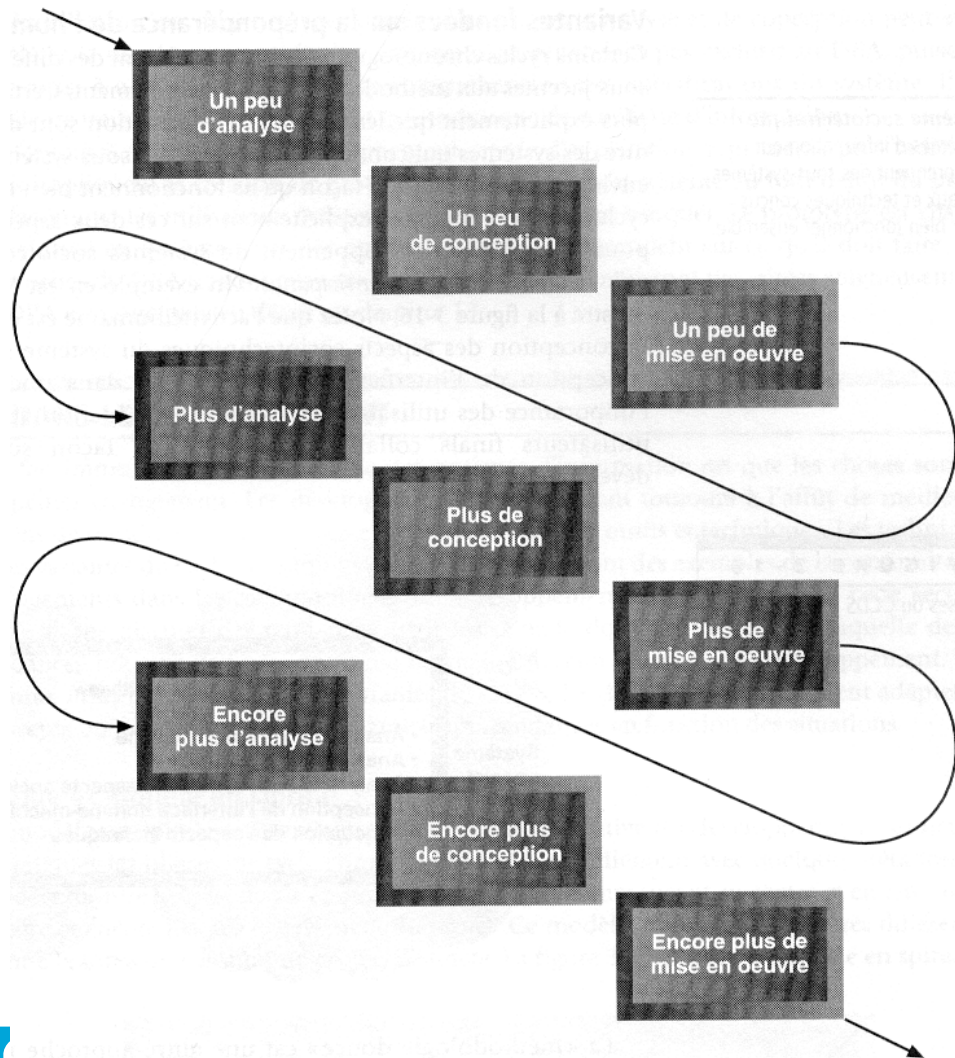
- Différents types de cycles de développement entrant dans la réalisation d'un système.
- Ces cycles prennent en compte toutes les étapes de développement d'un SI :
  - Spécification
  - Conception
  - Implémentation
  - Test
  - Installation
  - Maintenance

# Processus de développement en cascade



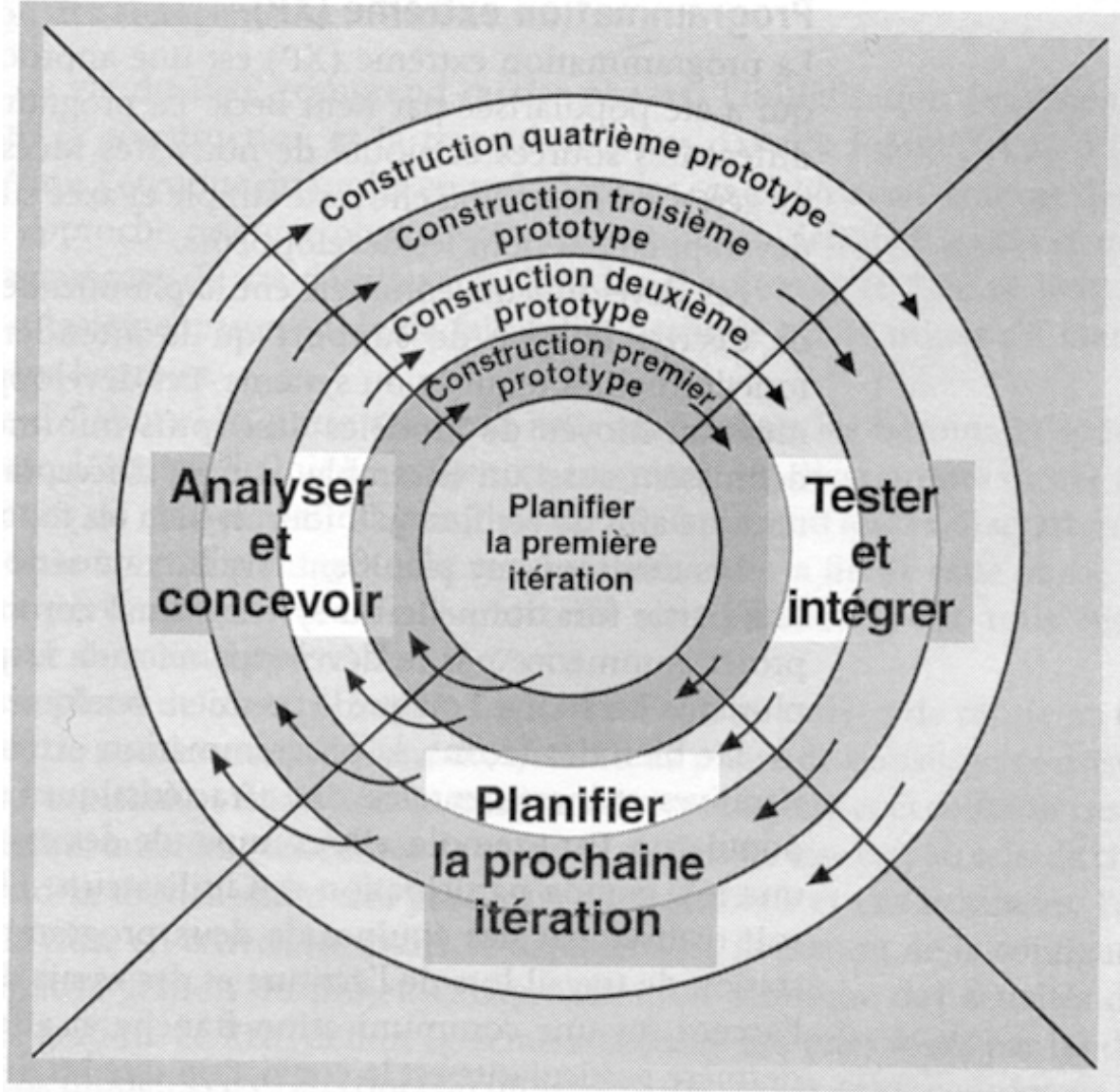
- Nature linéaire avec ordonnancement prédéfini des activités
- Toujours possible de revenir en arrière
- Modèle hérité de l'industrie lourde (bâtiments, usines, ...)
- Modèle adapté aux grands projets: n'offre que peu de possibilités d'interactions avec tous les membres de l'équipe.

# Processus de développement incrémentale



- Contrairement au cycle en cascade, ce cycle permet de programmer et tester précocement un système partiel selon des cycles répétitifs.
- On suppose que le développement commence **avant que tous les besoins** n'aient été définis en détail.
- Le feedback permettra de clarifier, d'améliorer et de faire évoluer les spécifications.
- Il s'appuie sur des étapes de développement **courtes** et **rapides**.

# Processus de développement en spirale



- Prévoit l'implémentation des versions successives : permet de mettre l'accent sur la gestion des risques
- Prévoit la livraison de prototypes ; des versions incomplètes du produit.
- Le prototype peut être une simple maquette, des sites partiellement fonctionnels.
- Avantageux pour les grands projets.



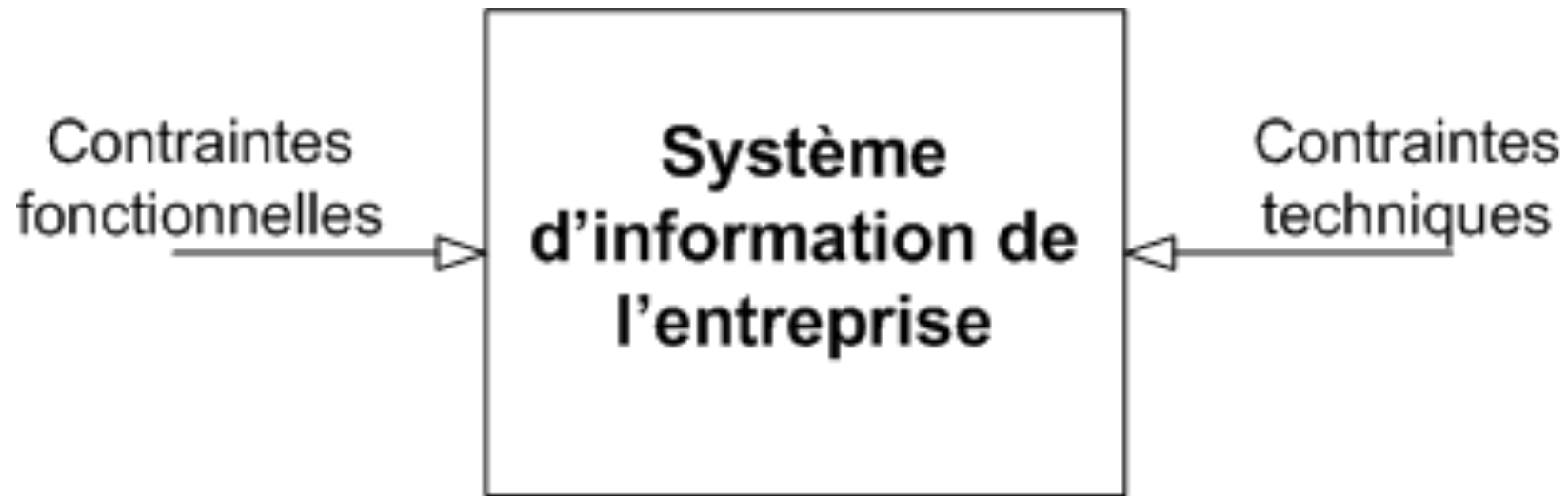
# Processus de développement logiciel 2TUP (2 track unified process)

- Processus définit une séquence d'étapes, en partie ordonnées, qui permet de produire ou de faire évoluer un système.
- Utilisation d'un processus dit Unifié (**UP: unified process**).
- Un **processus unifié** est un processus de développement logiciel construit sur **UML**, qui est :
  - **itératif et incrémental**: le projet est découpé en des itérations de courte durée. À chaque itération il est produit un exécutable.
  - **centré sur l'architecture**: système décomposé en modules pour des besoins de maintenabilité et d'évolutivité
  - **conduit par les cas d'utilisation**: met en avant les besoins et exigences des futures utilisateurs.
  - **pilotés par les risques**: identifie et écarte tout risque pouvant conduire à un échec du produit.



# Processus de développement logiciel 2TUP (2 track unified process)

2 track signifie littéralement que le processus suit deux chemins; le chemin « **fonctionnel** » et le chemin « **architecture technique** ».



# Bien distinguer les niveaux de modélisation

## Niveau conceptuel

- Ce que le système fait? Pourquoi? Pour qui?
- Quelles sont les données à conserver?
- Quelles sont les informations recherchées? ...

Accent mis sur la définition du système et sur le problème à résoudre

- Se préoccupe du QUOI et du POURQUOI
- Vocabulaire utilisé: entités (objets), associations(relations), attributs, identifiants, cardinalité
- Utilisation du modèle entité-association

# Partie statique d'un système : les données

## Le modèle de données:

Un modèle de données est une représentation des données utilisées par une entreprise et des associations entre ces données qui sont nécessaires pour répondre aux besoins en information de cette entreprise.

- Les données à modéliser sont définies par les besoins en informatique,
- Ces données sont emmagasinées dans un système d'information,
- **Ces données sont trouvées à partir des cas d'utilisation**

# Bien distinguer les niveaux de modélisation

Niveau fonctionnel - on définit le fonctionnement du système et la solution(informatisée)

- Quel est le comportement du système ?
- Quelles activités sont informatisées ?
- Quelles sont les façons d'accéder aux données?
- Où circulent-elles?
- Qui est responsable de quoi?

# Bien distinguer les niveaux de modélisation

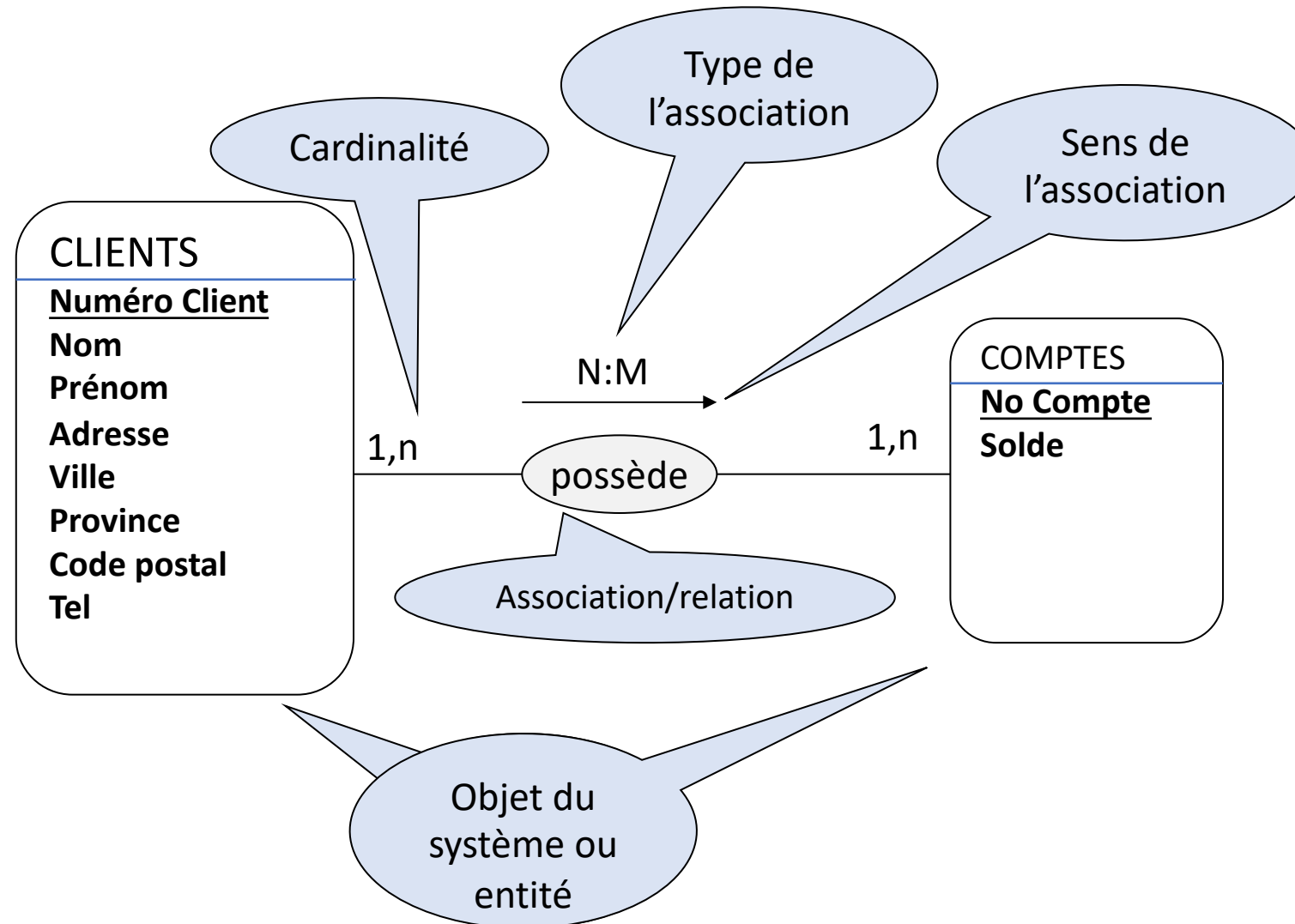
## Niveau physique

- Comment tel traitement est réalisé?
- Quels sont les équipement requis?
- Comment s'adapter aux contraintes clients?

# Pourquoi modéliser?

- La conception initiale du système a un impact énorme sur la performance finale.
- Si la construction a été laissée au hasard, les possibilités d'obtenir un système cohérent et intégré sont minimales.
- Plus les décisions importantes sont prises tôt, moins le système final est coûteux.
- Plus une erreur est trouvée tard, plus elle coûte cher !

# Exemple de modèle conceptuel de données (MCD)





# Modèle physique des données (MPD)

- Préciser les tables qui vont constituer la base de données
- Vocabulaire utilisé: table, colonne, clé primaire, clé étrangère
- Utilisation du modèle conceptuel de données (MCD)

