

Conception des systèmes d'information

Chapitre 1 : introduction à la conception orientée objet

Dr. Mariem MAHFOUDH

mariem.mahfoudh@gmail.com

2 LSI-ADBD, ISIMS, 2023-2024



University of Sfax

Objectifs

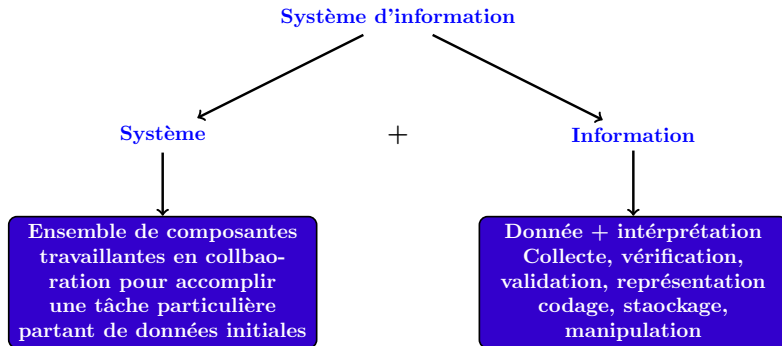
Ce cours vise à permettre aux étudiants de :

- ▶ assimiler et maîtriser les concepts de base de l'approche orientée objet ;
- ▶ concevoir des systèmes d'information en se basant sur le langage de modélisation UML (Unified Modeling Language) ;
- ▶ profiter de l'apport de cette approche au niveau programmation Objet.

Ce cours a été construit en se basant sur les références suivantes :

- ▶ Livre "UML 2 pour les bases données", Chritian Soutou
- ▶ Livre "UML 2 pour la pratique", Pascal Roques
- ▶ Livre " UML 2 pour les développeurs", Xavier Blanc et Isabelle Mounier
- ▶ Livre "UML 2 de l'apprentissage à la pratique", Laurent Audibert
- ▶ Livre "Modélisation Objet avec UML", Pr. Pierre Alain Muller
- ▶ Cours "Conception des systèmes d'information", Pr. Faiez Gargouri
- ▶ Cours "Conception Orientée Objets", Frédéric Mallet

- ① Les systèmes d'information
- ② La modélisation
 - C'est quoi un modèle, une modélisation ?
 - Pourquoi modéliser ?
 - Historique des méthodes d'analyse et de conception
- ③ L'approche orientée objet
 - L'objet
 - La classe
 - L'encapsulation
 - L'héritage
 - Le polymorphisme
- ④ Le langage UML
 - Introduction à UML
 - Les diagrammes
 - Les diagrammes



De point d'un système de type entreprise, l'information est un symbole qui fournit pour les acteurs d'une entreprise une connaissance utile à l'accomplissement de leur travail.

Définition

Le Système d'Information (SI) est l'ensemble des moyens humains et matériels, et les méthodes se rapportant au traitement des informations associées aux activités d'une organisation.

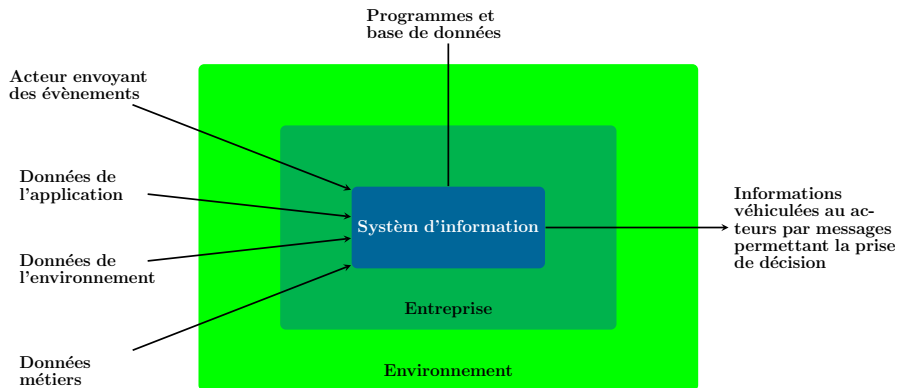
Définition

Le Système d'Information (SI) est un ensemble organisé de ressources (matériel, logiciel, personnel, données, procédures...) permettant d'acquérir, de stocker, de traiter, de communiquer des informations de toutes formes dans une organisation.

Exemples de SI

Une entreprise de transport, une école, une banque, etc.

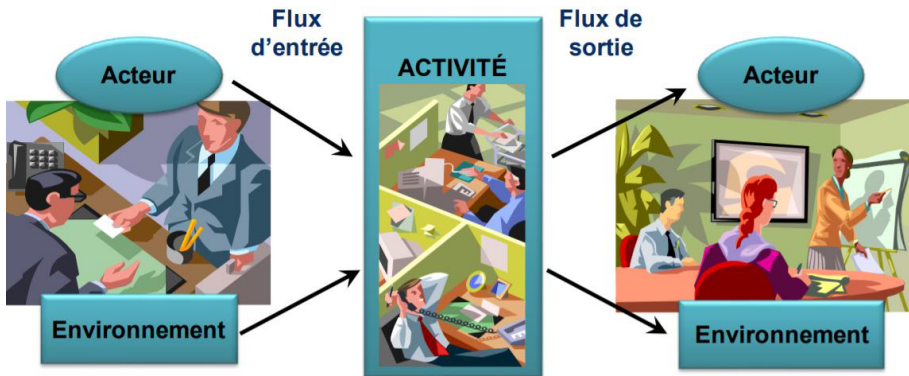
Système d'information



Rôle d'un système d'information

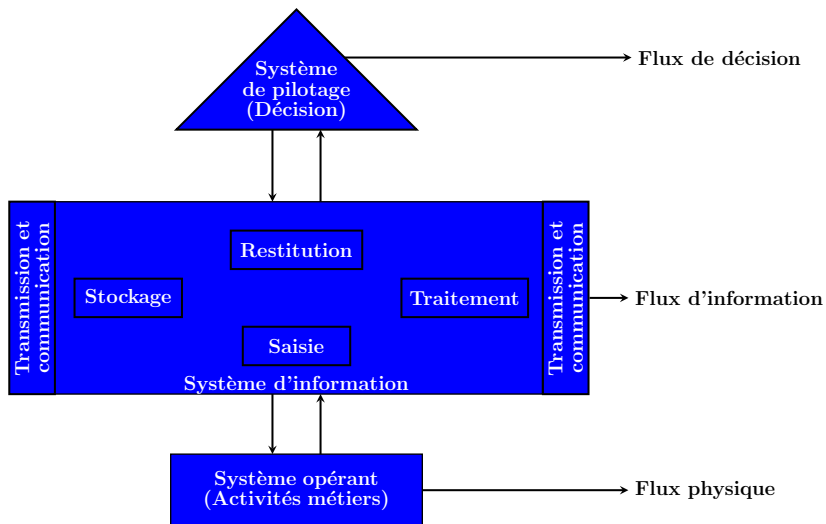
- ▶ Aider les utilisateurs dans leurs activités : stocker et restaurer l'information, faire des calculs, permettre une communication.
- ▶ Collecte, stockage, traitement et diffusion de l'information.

Système d'information d'une organisation



Collecte, saisie \Rightarrow Traitement, stockage \Rightarrow Diffusion

Système d'information d'une organisation



Système opérant

- ▶ Responsable de l'exécution des tâches
- ▶ Permettre de transformer les matières premières en des produits finaux

Système d'information

- ▶ Sous-système de l'organisation qui s'occupe de récolter l'information, de la stocker, de la traiter et de la diffuser dans le système opérant et dans le système de pilotage

Système de pilotage

- ▶ L'administration : fixer des objectifs, contrôler leur réalisation, gérer, planifier, etc.

Système d'information d'une organisation

Exemple : Système d'information d'une école

Flux physique

- ▶ Les élèves, les concours, les cours, les examens, les résultats, les enseignants, les départements, etc.

Flux d'information

- ▶ Nom, prénom, adresse des étudiants, des parents des enseignants, date de naissance, num de l'étudiant, etc.

Flux de décision

- ▶ Nombre d'élèves qui seront admis, la définition des programmes des études, la définition des modalités d'examen, les développements futures de l'école tels que la création de nouvelles filières, les investissements en matériel, etc.

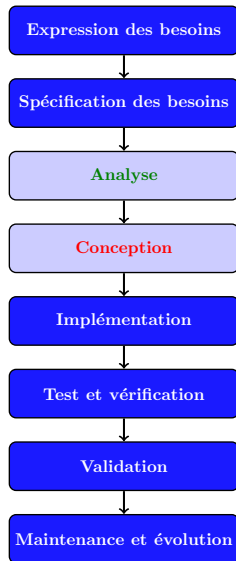
C'est quoi un modèle ?

- ▶ Un modèle est une représentation abstraite et simplifiée d'un système ou d'un processus : une simplification de la réalité.



La modélisation

- ▶ La modélisation est la conception d'un modèle.
- ▶ En informatique, on parle de modélisation des données pour désigner une étape de construction d'un système d'information.



Etapes de réalisation d'un SI

- ▶ La modélisation consiste tout d'abord à **comprendre et décrire un problème**, puis à **trouver et décrire la solution de ce problème**.
→ Ces activités s'appellent respectivement **l'analyse** et **la conception**.
- ▶ L'**analyse** est le processus d'examen de l'existant. Trois axes à définir :
 - Savoir-faire de l'objet : axe fonctionnel
 - Structure de l'objet : axe statique
 - Cycle de vie de l'objet : axe dynamique
- ▶ La **conception** est le processus de définition de la future application informatique :
 - Apport de solutions techniques : architecture, performance et optimisation
 - Définition des structures et des algos.

Pourquoi modéliser ?

Objectifs de la modélisation

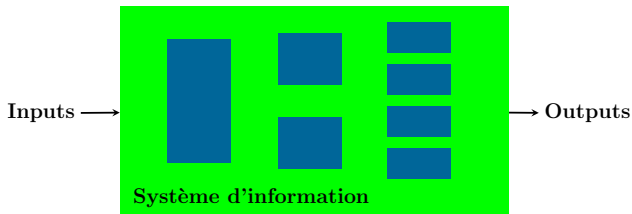
- ▶ Aider à visualiser le système.
- ▶ Spécifier la structure et le comportement.
- ▶ Servir de plan pour la construction effective.
- ▶ Permettre de documenter les choix.

Avantages de la modélisation

- ▶ Abstraction : diviser pour régner.
- ▶ Compréhension : mise au point avec le client.
- ▶ L'énergie déployée pour modéliser révèle les difficultés.
- ▶ Les erreurs sur les modèles coûtent bien moins cher.

Approches fonctionnelles

- ▶ Première génération des méthodes d'analyse et de conception de SI (à partir des années 70).
- ▶ Approche traditionnelle utilisant des procédures et des fonctions : identification des fonctions nécessaires à l'obtention du résultat.
- ▶ Les grands programmes sont décomposés en sous programmes.



Approches fonctionnelles : points forts

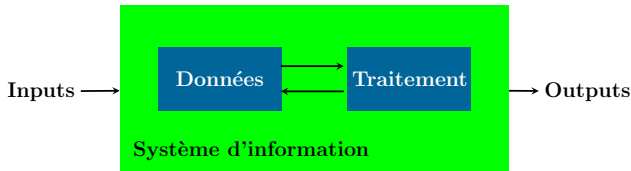
- ▶ Simplicité de processus de conception.
- ▶ Affinage progressif.
- ▶ Répond plus vite aux besoins ponctuels des utilisateurs (applications simples).

Approches fonctionnelles : points faibles

- ▶ Concentration sur le traitement.
- ▶ Redondance possible de données.
- ▶ Problème de structuration des données.

Approches systémiques

- ▶ Deuxième génération des méthodes d'analyses et de conception de SI (à partir des années 80).
- ▶ Traiter le système un ensemble d'entités communicant entre elles et avec l'extérieur.
- ▶ Une double approche de modélisation : des données et des traitements.
- ▶ Exemple : Merise, SSADM, Axial, etc.



Approches systémiques : points forts

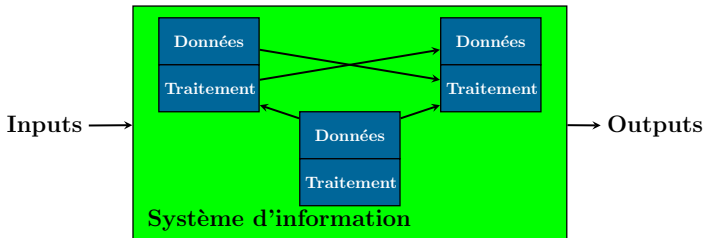
- ▶ Approche globale : données et traitements.
- ▶ Introduction des niveaux d'abstraction dans le processus de conception : conceptuel, logique et physique.
- ▶ Bonne adaptation à la modélisation des données et la conception des base de données.

Approches systémiques : points faibles

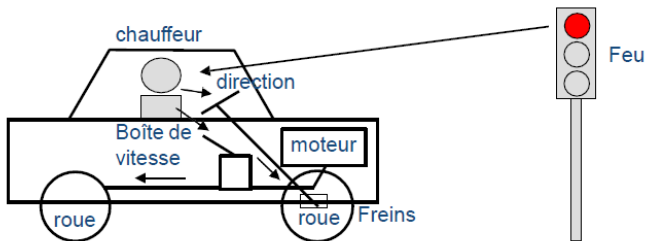
- ▶ Redondance possible de données.
- ▶ Séparation entre les données et les traitements.

Approches orientées objets

- ▶ Dernière génération des méthodes d'analyse et de conception des SI (à partir des années 90).
- ▶ Émergence de la philosophie Objet.
- ▶ Inclure les concepts objets dans la démarche d'analyse et de conception.
- ▶ Exemple : UML, O*, OMT , etc.



Le monde est composé d'entités qui "collaborent".



→ L'approche orientée objet consiste à résoudre un problème en termes d'objets qui collaborent.

Approches orientées objets : points forts

- ▶ Reflète plus finement les objets du monde réel.
- ▶ Du code (plus) facile à maintenir.
- ▶ Plus stable : un changement s'applique à un sous-système facile à identifier et isoler du reste du système.
- ▶ Réduire la distance sémantique entre le langage des utilisateurs et le langage des concepteurs : meilleure communication entre utilisateurs et concepteurs.

Approches orientées objets : points faibles

- ▶ Parfois moins intuitive que l'approche fonctionnelle.
- ▶ Pas de fils conducteur, nécessite une expérience pour être mise en place.

Les concepts de l'orientée objet

- ▶ L'objet
- ▶ La classe
- ▶ L'encapsulation
- ▶ L'héritage
- ▶ Le polymorphisme

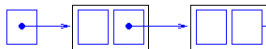
L'objet

- ▶ représente une entité (physique, logicielle, ...) du monde réel

- ▶ Entité physique : ex. une voiture X



- ▶ Entité logicielle : ex. une liste chaînée



L'objet

- ▶ possède sa propre identité : une identité intrinsèque (deux objets sont distincts même si toutes les valeurs de leurs attributs sont identiques)

L'objet

- ▶ L'objet a une frontière bien définie, une identité : état et comportement.
- ▶ Un ensemble d'attributs caractérise l'état de l'objet.
- ▶ Un ensemble d'opérations (méthodes) définissent le comportement.
- ▶ L'état d'un objet peut changer dans le temps (durant son cycle de vie)



Une voiture	
	Bleu
	979 kg
	12 CV

La classe

- ▶ regroupe un ensemble des objets semblables :
 - les mêmes propriétés structurelles (attributs) ;
 - le même comportement (opérations, méthodes)
 - les mêmes liens avec les autres objets d'autres. classes
- ▶ Un objet est une instance d'une classe.

Class Voiture

\\Attributs

```
String marque;  
String couleur;  
String immatriculation;
```

\\Opérations

```
void démarrer( ){ }  
void conduire( ){ }  
void arrêter( ){ }
```

Voiture
+Immatriculation +Couleur +Marque
+Démarrer() +Conduire() +Arrêter()

Instanciation

```
Voiture V1= new Voiture ();
```

L'encapsulation

- ▶ consiste à cacher les données et les détails d'implémentation et laisser visible une interface (signature des opérations publiques des classes)

Class Compte

\\Attributs

```
int numéro ;  
float solde;  
... ;
```

\\Opérations

```
void créer () {...};  
float calculerIntérêt (float x)  
{...};
```

Programme utilisateur

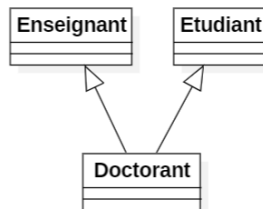
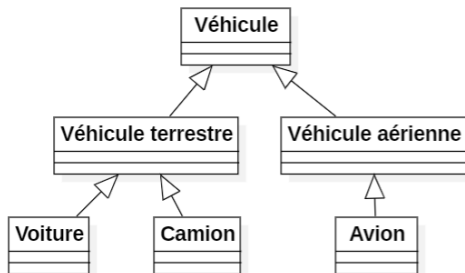
```
a= lire(intérêt) ;  
b= calculerIntérêt(a) ;  
print (b);
```

Partie invisible à l'utilisateur

```
float calculerIntérêt (float x)  
{ return x*solde ; }
```

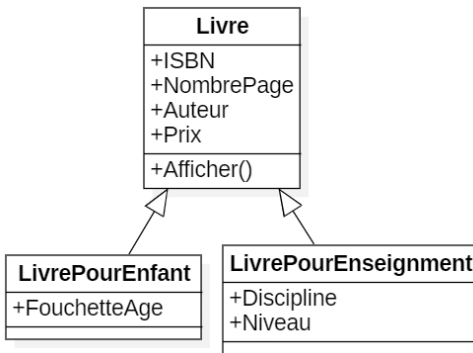
L'héritage

- ▶ L'héritage est un mécanisme de transmission des caractéristiques (attributs, méthodes) d'une classe à une sous classe
- ▶ L'héritage peut être simple ou multiple



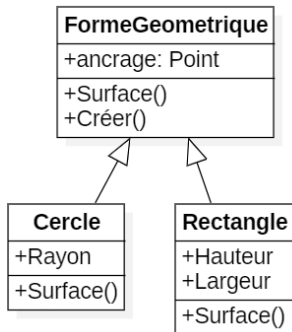
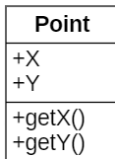
L'héritage

- ▶ Une sous-classe hérite les attributs, les opérations et les références de ses parents.
- ▶ Une sous-classe peut :
 - ajouter des attributs, des opérations, des références ;
 - redéfinir des opérations héritées.



Le polymorphisme

- C'est un mécanisme qui permet à une sous classe de redéfinir une méthode dont elle a hérité tout en gardant la même signature de la méthode héritée.



- ▶ Durant les dernières décennies plus que cinquante méthodes objets ont été proposées (OOD, OOSE, OMT, O*, etc.)
- ▶ Suite à cette multitude de propositions :
 - divers concepts sont utilisés ;
 - divers modèles proposés ;
 - diverses démarches suivies ;
 - diverses notations graphiques supportées.
- ▶ En 1994, les industriels ont commencé à demander une notation standard

D'où le besoin d'unification



Proposition du langage



Unified Modeling Language (UML)



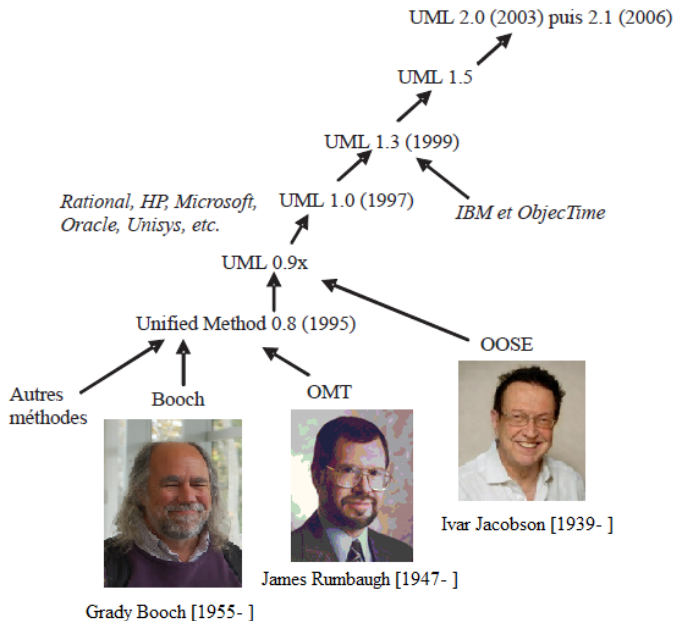
UML

- ▶ Le résultat de la fusion des trois méthodes qui ont influencé la modélisation objet : OMT, Booch et OOSE.
- ▶ Un compromis qui a été trouvé par une équipe d'experts : Grady Booch, James Rumbaugh et Ivar Jacobson.
- ▶ Un standard défini par l'OMG (Object Management Group).
- ▶ Un standard des notations graphiques et du vocabulaire.
- ▶ N'est pas une méthode de conception mais un ensemble de notation unifié.
- ▶ UML n'est pas une méthode mais un langage.



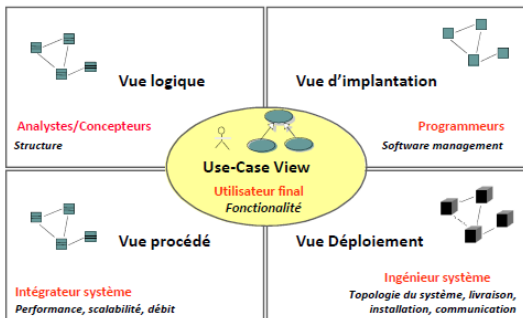
OBJECT MANAGEMENT GROUP

Historique

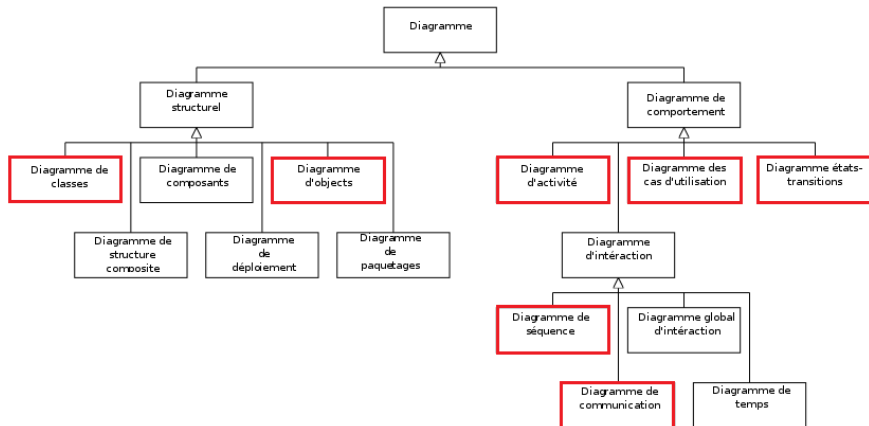


Les diagrammes

- ▶ Le diagramme est un élément fondamental de modélisation UML.
- ▶ Un diagramme est une représentation graphique d'objets et de relations entre eux.
- ▶ Il en existe plusieurs sortes de diagramme : un seul modèle ne suffit pas !
- ▶ Les modèles sont regardés et manipulés par les utilisateurs au moyen des vues : à chaque vue correspond un ou ++ diagrammes.



- Les différents types de diagrammes UML combinés offrent une vue complète des aspects statiques et dynamique d'un système.



- ▶ L'ordre de représentation des diagrammes n'est pas obligatoirement l'ordre de leur utilisation.
- ▶ Chaque type de diagramme UML :
 - possède une structure : les types des éléments de ;
modélisation (concepts) qui le composent sont prédéfinies
 - véhicule une sémantique précise.
- ▶ La structure des diagrammes UML et la notation graphique des éléments de modélisation sont normalisées.
- ▶ La sémantique des éléments de modélisation et leur utilisation est définie par le métamodèle UML (www.omg.org).

Chapitre 2 : Diagrammes de cas d'utilisation