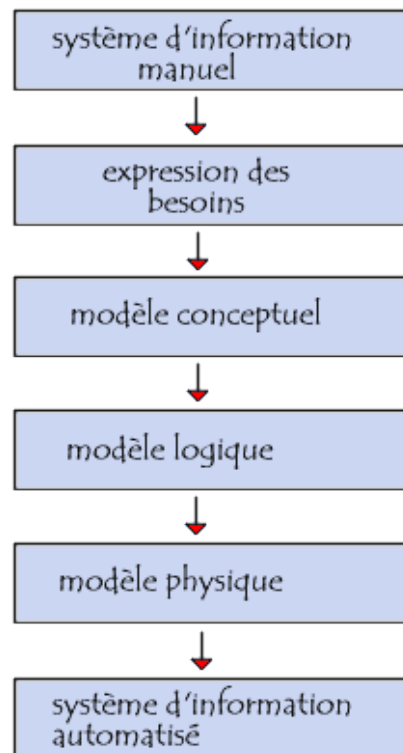


ANALYSE ET CONCEPTION DE BASE DE DONNEES

Présenté par :
Mor Anta SENE
Mohameth LAYE Mbengue

APPROCHE DE LA DECOMPOSITION FONCTIONNELLE : METHODE MERISE

DEMARCHE



AVANTAGES

Les avantages de la méthode Merise sont nombreux :

1. Elle permet une **modélisation** claire et précise du système, en utilisant des diagrammes et des représentations graphiques.
2. Elle facilite la **communication** entre les parties prenantes, en utilisant un langage commun pour décrire les besoins et les fonctionnalités attendues.
3. Elle permet une **conception modulaire** et **évolutive** du système, en divisant le système en sous-systèmes fonctionnels.
4. Elle permet une **validation** précoce des choix de conception, en identifiant les erreurs et les incohérences dès les premières étapes.

INCONVENIENTS

Cependant, la méthode Merise présente également certains inconvénients :

1. Elle peut être **coûteuse** et **chronophage**, en raison du grand nombre d'étapes et de documents requis pour la conception complète d'un système.
2. Elle peut être **rigide** et **difficile à adapter à des changements de besoins**, car elle repose sur une analyse préalable exhaustive.
3. Elle peut nécessiter une expertise **technique** et **fonctionnelle** importante pour être mise en œuvre efficacement.

APPROCHE ORIENTÉE OBJET : UML

DEMARCHE

La démarche de l'approche orientée objet avec UML comprend les étapes suivantes :

1. **Analyse des besoins** : Collecte des exigences du système et identification des objets clés impliqués.
2. **Modélisation des classes** : Définition des classes qui représentent les objets et leur structure interne, y compris les attributs et les méthodes.
3. **Modélisation des relations** : Définition des relations entre les classes, telles que l'héritage, l'agrégation, l'association, etc.
4. **Modélisation du comportement** : Définition des interactions et du flux de contrôle entre les objets en utilisant des diagrammes de séquence, des diagrammes d'activité, etc.
5. **Modélisation de l'architecture** : Définition de l'architecture logicielle globale du système en utilisant des diagrammes de composants, des diagrammes de déploiement, etc.

AVANTAGES

1. Abstraction et modularité : Permet de modéliser des systèmes complexes en utilisant des objets abstraits et en les regroupant en modules réutilisables.
2. Réutilisabilité : Les classes et les objets peuvent être réutilisés dans différents projets, ce qui accélère le développement logiciel.
3. Encapsulation : Les objets encapsulent à la fois les données et les méthodes, ce qui permet de maintenir l'intégrité des données et de cacher les détails internes.
4. Flexibilité et extensibilité : Permet des modifications et des ajouts faciles grâce à l'héritage, la polymorphisme et la composition.
5. Communication efficace : UML fournit des diagrammes graphiques qui facilitent la communication entre les parties prenantes techniques et non techniques.

INCONVÉNIENTS

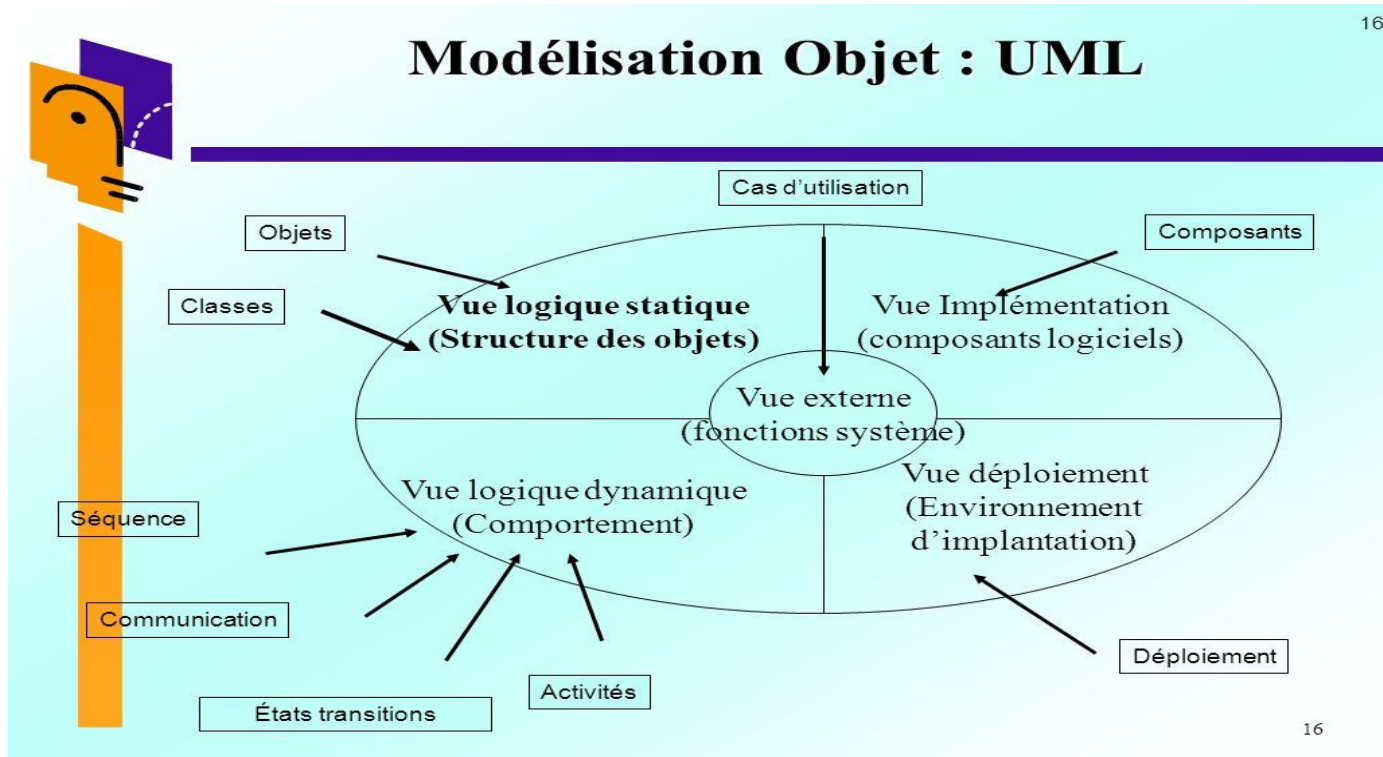
1. Complexité : L'approche orientée objet et UML peuvent être complexes à comprendre et à maîtriser, surtout pour les débutants.
2. Surdimensionnement : Pour les petits projets, l'utilisation complète de l'approche orientée objet et UML peut être excessive et entraîner un surdimensionnement.
3. Temps et coût : La modélisation complète d'un système avec UML peut nécessiter du temps et des ressources considérables.
4. Surcharge de documentation : La méthode UML encourage la création de nombreux documents et diagrammes, ce qui peut entraîner une surcharge de documentation.

ETUDE COMPARATIVE

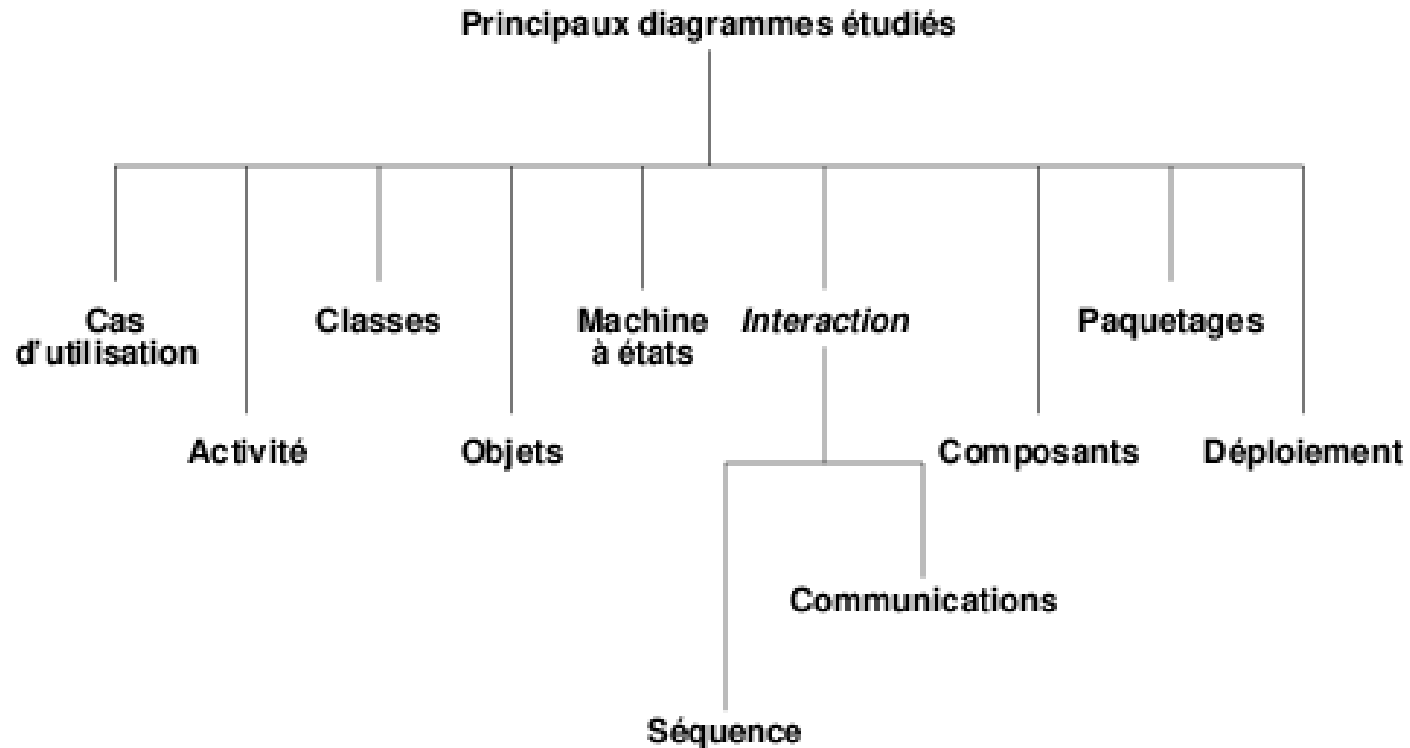
COMPARAISON MERISE vs UML

	UML	MERISE
Abstraction	Permet la modélisation de plusieurs aspects y compris les classes, les objets, les interactions et les comportements	Permet principalement la modélisation des données et des processus métier.
Réutilisabilité	Favorable à la réutilisabilité des classes et des objets grâce à l'héritage et le polymorphisme	Met moins l'accent sur la réutilisabilité
Flexibilité	offre une plus grande flexibilité en permettant des changements plus faciles grâce à la modélisation orientée objet	peut être moins flexible en raison de son approche plus structurée
Documentation	peut générer une plus grande quantité de documentation en raison de sa variété de diagrammes	peut être plus concis dans sa documentation en se concentrant sur les aspects clés de la modélisation des données et des processus

UML : APPROCHE 4 VUES



LES 13 DIAGRAMMES UML



CLASSIFICATION DES DIAGRAMMES

CLASSIFICATION PAR ASPECT

En termes d'aspect, les diagrammes fonctionnels sont ceux qui modélisent les aspects du système qui sont liés aux processus métier, aux besoins des utilisateurs et aux interactions entre les acteurs. Les diagrammes d'architecture modélisent les aspects techniques du système, tels que la structure des classes, les composants logiciels, la configuration matérielle et logicielle, etc.

CLASSIFICATION PAR VUE

1. Vue logique

- Diagramme de cas d'utilisation
 - Diagramme de classes
 - Diagramme d'objets
 - Diagramme de séquence
 - Diagramme de collaboration
 - Diagramme d'états
 - Diagramme d'activité

2. Vue des processus

- Diagramme de cas d'utilisation
 - Diagramme de séquence
 - Diagramme de collaboration
 - Diagramme d'états
 - Diagramme d'activité
 - Diagramme de flux de données
 - Diagramme de flux de contrôle
 - Diagramme de flux de traitements

3. Vue de composants

- Diagramme de composants
 - Diagramme de déploiement
 - Diagramme de packages

4. Vue de déploiement

- Diagramme de déploiement

5. Vue des besoins

- Diagramme de cas d'utilisation
 - Diagramme de séquences d'utilisation
 - Diagramme d'activité d'utilisation
 - Diagramme de classes d'analyse