



# Rapport Projet sur la conception d'une application des gestion de projets informatiques

Conception et élaboration d'un modèle suivant MERISE 2

HANFAOUI Karim<sup>1</sup>    KAFIF. Imane<sup>2</sup>

<sup>1</sup>EMSI, 3ème Année INFO G9 -  
Karim.Hanfaoui@emsi-edu.ma

<sup>2</sup>EMSI, 3ème Année INFO G9 -  
Imane.Kafif@emsi-edu.ma

EMSI Les Orangers, Décembre 2024

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# List of Figures

- Illustration des Événements dans le système
- Illustration des Opérations
- Illustration de la Synchronisation
- Illustration des Résultats
- MCD Réalisé
- MCT Réalisé
- MCTA Réalisé

## Abstract

*This report focuses on the design and development of a project management application tailored for IT teams. The study applies the MERISE 2 methodology to model and conceptualize the system effectively. Key contributions include the use of advanced modeling techniques (MCD, MCT, MLD, and MCTA) to optimize processes and ensure adaptability to modern organizational requirements. By comparing MERISE 1 and MERISE 2, the report highlights the advantages of a flexible, object-oriented approach suitable for evolving IT systems. This structured methodology provides a robust foundation for the design of complex, dynamic applications.*

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# C'est quoi notre Problématique

## Problématique

Comment concevoir et développer une application de gestion de projets informatiques capable de répondre efficacement aux besoins des équipes de développement en termes de suivi de l'avancement des projets, de gestion optimale des ressources, et de facilitation de la communication et de la collaboration, tout en s'adaptant aux contraintes organisationnelles et technologiques variées des entreprises ?

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# Acronymes et Terminologie

**MERISE** : Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise

**MCD** : Modèle Conceptuel de Données

**MLD** : Modèle Logique de Données

**MCT** : Modèle Conceptuel de Traitement

**MOT** : Modèle Organisationnel des Traitements

**MCTA** : Modèle Conceptuel des Traitements Automatisés

**MOTA** : Modèle Organisationnel des Traitements Automatisés

**UML** : Unified Modeling Language (Langage de Modélisation Unifié)

**USE CASE** : Cas d'Utilisation, outils de l'UML

**Réseau de Petri** : Modèle mathématique pour la modélisation et l'analyse des processus concurrents



# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours**
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# La Conception d'un Système d'Information (SI)

- **Nécessité stratégique** : Répondre aux besoins actuels and évoluer with l'organisation.
- **Objectifs** :
  - ▶ Optimiser les opérations courantes.
  - ▶ Préparer aux défis futurs.
- **Importance** : Garantir des systèmes efficaces et adaptés aux entreprises.

# Critères de Qualité d'un Logiciel

- **Utilité** : Répondre précisément aux besoins.
- **Utilisabilité** : Facilité d'apprentissage et d'interaction.
- **Fiabilité** : Fonctionnement sans défaillance.
- **Interopérabilité** : Interaction avec d'autres systèmes.
- **Performance** : Rapidité et efficacité en gestion des ressources.
- **Portabilité** : Adaptabilité à différents environnements.
- **Réutilisabilité** : Réutilisation des composants.
- **Facilité de maintenance** : Correction et mise à jour aisées.

# Cycle de Vie du Logiciel : Étapes

- ➊ **Analyse des besoins et des risques** : Étude de faisabilité.
- ➋ **Spécification** : Cahier des charges fonctionnelles et techniques.
- ➌ **Conception** : Définition de l'architecture et des interfaces.
- ➍ **Codage** : Traduction des spécifications en code source.
- ➎ **Tests** : Vérification et validation.
- ➏ **Livraison** : Déploiement et formation des utilisateurs.
- ➐ **Maintenance** : Correction, amélioration et adaptation.

# Modèles de Cycle de Vie

- **Modèle en cascade** : Processus séquentiel, validation étape par étape.
- **Modèle itératif et incrémental** : Développement par cycles avec ajout progressif de fonctionnalités.
- **Méthodes agiles** :
  - ▶ Exemples : Scrum, Kanban.
  - ▶ Itérations courtes avec adaptation rapide aux changements.

# Documents Produits dans le Cycle de Vie

- **Cahier des charges** : Définition des besoins fonctionnels et techniques.
- **Calendrier du projet** : Planification des étapes et livrables.
- **Plan de test** : Objectifs et stratégies de validation.
- **Plan d'assurance qualité** : Standards pour garantir la qualité.
- **Manuel utilisateur** : Guide pratique pour les utilisateurs.
- **Code source** : Instructions codées représentant le logiciel.
- **Rapport des tests** : Résultats des tests effectués.
- **Rapport des défauts** : Liste des anomalies et corrections.

# Modélisation : Définition et Rôle

- **Définition** : Représentation abstraite de la réalité pour concevoir des logiciels structurés.
- **Rôles du modèle** :
  - ▶ Document d'échange entre clients et développeurs.
  - ▶ Outil de conception.
  - ▶ Référence pour le développement et la maintenance.

- **MERISE :**

- ▶ MCD : Modèle Conceptuel de Données.
- ▶ MCT : Modèle Conceptuel des Traitements.
- ▶ MLD : Modèle Logique des Données.

- **UML (Unified Modeling Language) :**

- ▶ Diagrammes : Cas d'utilisation, classes, séquences.

- **Modélisation orientée objets :**

- ▶ Décomposition en objets autonomes et interactifs.



# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2**
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# Qu'est-ce que MERISE 1 ?

- MERISE 1 est une méthodologie française d'analyse et de conception des systèmes d'information.
- Développée dans les années 1970, elle est utilisée pour modéliser et réaliser des bases de données et des applications.

# Principes fondamentaux de MERISE 1

## ① Séparation des niveaux de modélisation :

- ▶ Modèle conceptuel (niveau conceptuel)
- ▶ Modèle logique (niveau organisationnel)
- ▶ Modèle physique (niveau technique)

## ② Cycle de vie du système d'information :

- ▶ Étude préalable, étude détaillée, réalisation, exploitation et maintenance.

## ③ Focus sur les données :

- ▶ Modèle Conceptuel des Données (MCD)
- ▶ Modèle Logique des Données (MLD)
- ▶ Modèle Physique des Données (MPD)

# Caractéristiques de MERISE 1

- Approche systématique, adaptée aux projets bien définis et peu évolutifs.
- Centré sur les bases de données relationnelles.
- Modélisation utilisant des diagrammes (MCD, MCT, DFD).

# Limites de MERISE 1

- Peu adapté aux systèmes modernes (orientés objet, dynamiques).
- Approche rigide, difficilement compatible avec des méthodologies agiles.

# Qu'est-ce que MERISE 2 ?

- MERISE 2 est une évolution de MERISE 1, adaptée aux systèmes d'information modernes.
- Introduite pour intégrer les technologies récentes comme les bases orientées objet et les architectures distribuées.
- Offre une flexibilité accrue grâce à des concepts enrichis et à une approche plus évolutive.

# Principes fondamentaux de MERISE 2

## ① Extension des niveaux de modélisation :

- ▶ Modèle conceptuel enrichi pour intégrer les cycles de vie des données.
- ▶ Modèle logique et physique adaptés aux bases orientées objet.

## ② Prise en compte des objets :

- ▶ Gestion des classes, objets, et relations dynamiques.
- ▶ Insistance sur les interactions entre objets et leurs états.

## ③ Cycle de développement souple :

- ▶ Permet des itérations fréquentes pour s'adapter aux évolutions du projet.

# Caractéristiques de MERISE 2

- Intègre les paradigmes orientés objet pour une modélisation plus moderne.
- Convient aux bases relationnelles et orientées objet.
- Approche plus souple, adaptée aux systèmes évolutifs et complexes.
- Utilisation des mêmes outils de modélisation (MCD, MLD, MPD) avec des extensions.



# Limites de MERISE 2

- Méthodologie complexe, nécessitant une expertise pour sa mise en œuvre.
- Moins populaire que les méthodologies modernes comme UML ou les approches agiles.
- Peut être lourde pour des projets de petite envergure.

# Comparaison entre MERISE 1 et MERISE 2

Critères	MERISE 1	MERISE 2
<b>Orientation</b>	Centrée sur les flux de données et traitements	Intègre une approche orientée objet
<b>Technologie ciblée</b>	Bases de données relationnelles classiques	Bases relationnelles et orientées objet
<b>Modélisation</b>	Modèles conceptuels, logiques et physiques classiques (MCD, MLD)	Modèles enrichis (cycles de vie des objets et interactions dynamiques)
<b>Flexibilité</b>	Rigide, séquentielle	Souple, itérative
<b>Adaptabilité</b>	Convient aux projets bien définis et peu évolutifs	Convient aux projets évolutifs et modernes
<b>Complexité</b>	Plus simple	Plus complexe, mais mieux adapté aux systèmes modernes
<b>Popularité actuelle</b>	Rarement utilisé aujourd'hui	Moins populaire que UML et les approches agiles

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique**
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références

# MCTA C'est quoi ?

**En modélisation informatique, notamment dans la méthode Merise,** le MCTA fait référence au **Modèle Conceptuel des Traitements et des Activités**. C'est une étape clé pour décrire comment les processus et les traitements fonctionnent dans un système d'information.

## Rôle du MCTA dans Merise :

- Le MCTA intervient après le Modèle Conceptuel des Données (MCD).
- Il se concentre sur les activités et traitements nécessaires pour manipuler les données.
- L'objectif est de décrire les flux, interactions et règles de gestion sans s'intéresser à l'aspect technique.

## Diagrammes associés :

- **Diagrammes d'activités** pour montrer les enchaînements.
- **Modèles conceptuels de flux (MCF)** pour représenter les flux de données entre traitements.

# Pourquoi MCTA et non pas autre modèle ?

## Différence avec le Modèle Logique des Traitements (MLT) :

- **MCTA** : Vision conceptuelle, indépendante de l'implémentation technique.
- **MLT** : Vision plus proche de l'implémentation informatique (scripts, programmes).

# Événements dans le MCTA

**Événements** : Eux qui déclenchent les activités /ou traitements dans le système.

- Correspondent aux stimuli externes ou internes qui démarrent un processus.
- Représenté par un ovale avec le nom et/ou description à l'intérieur
- **Exemples** :
  - ▶ Un client passe une commande.
  - ▶ Un paiement est reçu.
  - ▶ Une alerte est générée en cas de problème.

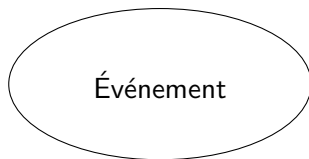


Figure: Illustration des Événements dans le système

# Synchronisation dans le MCTA

**Synchronisation** : Régule l'ordre d'exécution des traitements.

- Assure que les traitements se déroulent dans un ordre cohérent.
- **Exemples** :
  - ▶ Un paiement doit être validé avant l'expédition d'une commande.
  - ▶ Plusieurs traitements doivent s'exécuter simultanément.
  - ▶ Une activité attend la fin d'autres activités avant de se déclencher.

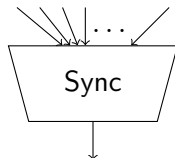


Figure: Illustration de la Synchronisation

# Synchronisation dans le MCTA

## Remarque :

Les règles représentant la synchronisation peuvent être composées et on obtient aisément la forme générale :

$$P = \sum_{i=1}^n \left( \prod_{j=1}^{m_i} L_{i,j} \right)$$

où:

- $P$  est le résultat de sortie.
- $L_{i,j}$  est un prédicat logique (un évènement arbitraire).
- $\sum_{i=1}^n m_i$  est le nombre total d'événements.



# Un exemple de Sync:

## Exemples

a ET b

Pour obtenir cette expression à partir de la forme générale on fixe les paramètres suivants :

\*  $n = 1$  (une seule clause) \*  $m_1 = 2$  (deux littéraux dans la clause) \*  $L_{1,1} = a$  \*  $L_{1,2} = b$

En substituant ces valeurs dans la forme générale, on obtient :

$$\sum_{i=1}^1 \left( \prod_{j=1}^2 L_{i,j} \right) = \prod_{j=1}^2 L_{1,j} = L_{1,1} \wedge L_{1,2} = a \wedge b$$

# Un autre exemple de Sync composé

## Exemples

a OU b ET c

Pour obtenir cette expression à partir de la forme générale

$$\sum_{i=1}^n \left( \prod_{j=1}^{m_i} L_{i,j} \right),$$

on fixe les paramètres suivants :

\*  $n = 2$  (deux clauses) \*  $m_1 = 1$  (un littéral dans la première clause) \*  $m_2 = 2$  (deux littéraux dans la deuxième clause) \*  $L_{1,1} = a$  \*  $L_{2,1} = b$  \*  $L_{2,2} = c$

En substituant ces valeurs dans la forme générale, on obtient :

$$\begin{aligned} \sum_{i=1}^2 \left( \prod_{j=1}^{m_i} L_{i,j} \right) &= \left( \prod_{j=1}^1 L_{1,j} \right) \vee \left( \prod_{j=1}^2 L_{2,j} \right) \\ &= L_{1,1} \vee (L_{2,1} \wedge L_{2,2}) \\ &= a \vee (b \wedge c) \\ &= a \vee (b \wedge c) \end{aligned}$$

# Opérations dans le MCTA

**Opérations** : Actions spécifiques réalisées par le système pour transformer les données.

- Correspondent aux unités élémentaires d'un traitement.
- **Exemples** :
  - ▶ Calculer le total d'une commande.
  - ▶ Mettre à jour le stock.
  - ▶ Valider une transaction.

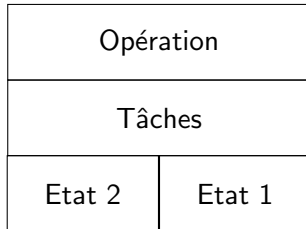


Figure: Illustration des Opérations

# Résultats dans le MCTA

**Résultats** : Les sorties des activités ou traitements.

- Correspondent aux effets observables ou données produites après une opération.
- Représenté par un ovale avec le nom et/ou description à l'intérieur
- **Exemples** :
  - ▶ Une facture générée pour un client.
  - ▶ Un email de confirmation envoyé.
  - ▶ Une mise à jour dans une base de données.



Figure: Illustration des Résultats

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2**
- 7 Conclusion
- 8 Bibliographies et Références

# Elaboration du MCD

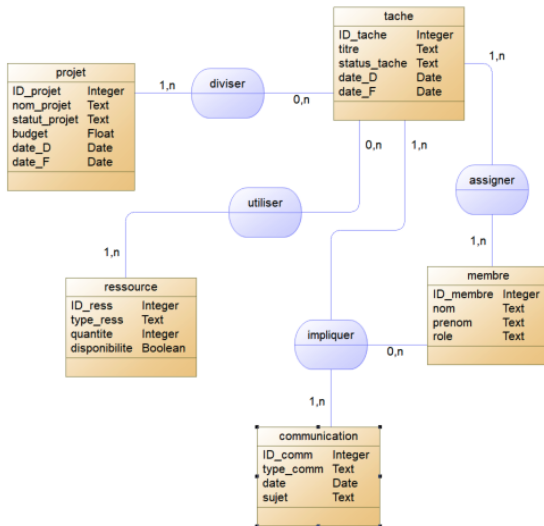


Fig. Schema du MCD

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion**
- 8 Bibliographies et Références

# Conclusion - Aperçu

## **À travers cette étude, nous avons exploré :**

- La conception d'une application de gestion de projets informatiques.
- L'application de la méthodologie MERISE 2.

## **Points clés :**

- Modélisation de systèmes complexes et évolutifs.
- Utilisation des modèles : MCD, MCT, MLD, MCTA.



# Points forts de l'étude

## **Approfondissements :**

- Concepts approfondis : événement, opération, synchronisation.
- Formalisation précise des processus métier grâce au MCTA.

## **Comparaison MERISE 1 vs MERISE 2 :**

- Avantages en termes de flexibilité et d'adaptation.
- Nécessité d'une expertise solide pour la mise en œuvre.

# Conclusion finale

## Résumé :

- MERISE 2, une méthodologie rigoureuse pour la gestion de projets.
- Facilite la compréhension des besoins et la communication.
- Offre un cadre structuré pour modéliser les processus.

## Impact :

- Pertinence démontrée pour la conception d'applications complexes.
- Une base solide pour les projets informatiques évolutifs.

# Table of Contents

- 1 Problématique
- 2 Acronymes et Terminologie
- 3 Résumé du cours
- 4 Comparaison entre Merise 1 et Merise 2
- 5 Présentation du modèle conceptuel du traitement analytique
- 6 Conception du projet selon les règles de Merise 2
- 7 Conclusion
- 8 Bibliographies et Références**

# Bibliographies



BTS CGO 2A P10 - *Organisation du Système d'Informations Fiche MCT, MCTA, MOT et MOTA*. Editions d'Organisation. [Lien]



Ben Salem Amir - *Conception des systèmes d'information La méthode Merise*. Academia. [Lien]



*Chap 4 MCT et MCTA* Calameo. [Lien]



G. PANET et R. LETOUCHE, *Merise/2 - Modèles et techniques MERISE avancés*. Les Éditions d'Organisation, 1994. [Lien]



Pr. S.EL MOUMNI - *Conception des systèmes d'information*. Cours mise à niveau. [Lien non disponible]

# Remerciements

Je tiens à remercier toutes les personnes suivantes pour leur soutien et leurs contributions :

- Mon professeur pour ses conseils et son accompagnement du projet.
- Mes collègues et camarades pour leur collaboration et leur soutien.
- Ma famille et mes amis pour leur encouragement constant.