



UML

Les méthodes de modélisation

Rédigé par

David ROUMANET
Professeur BTS SIO

Changement

Date	Révision
Août 2020	Création
24/09/2020	Séparation des méthodologies étudiées (Merise+MEA et RUP+UML)

Sommaire

A Introduction à la modélisation.....	1
A.1 Problématique.....	1
A.2 Définition de modélisation.....	1
A.3 Objectifs du cours.....	1
B Les méthodes de modélisation.....	2
B.1 Les méthodes non-étudiées.....	2
B.1.1 IDEF/SADT.....	2
B.1.2 ORM/NIAM.....	3
B.1.3 FAST.....	4
B.2 Les méthodes étudiées.....	5
B.2.1 Merise.....	5
B.2.1.a MCD.....	6
B.2.1.b MOD.....	7
B.2.1.c MLD.....	7
B.2.1.d MPD.....	7
B.2.2 RUP.....	8
B.2.2.a Diagramme d'utilisation.....	10
B.2.2.b Diagramme de séquences.....	10
B.2.2.c Diagramme de classes.....	11
C Synthèse.....	12
D Annexes.....	13
D.1 Sources complémentaires.....	13
D.2 RUP.....	13
D.3 Cycle en cascade versus cycle itératif.....	13

A Introduction à la modélisation

A.1 Problématique

Lorsque vous travaillez sur un petit programme, il paraît relativement facile d'atteindre son objectif.

Les difficultés apparaissent lorsque vous décidez de respecter un ensemble de contrainte, et que le programme devient une application.

Le client, propose généralement un cahier des charges, avec de nombreuses informations et de multiples traitements. Programmer une solution ne peut – et ne doit plus – se faire en improvisant au fur-et-à-mesure.

Comment s'assurer que notre base de données sera correctement réalisée ? Contiendra-t-elle toutes les informations nécessaires ? N'y a-t-il pas d'oubli dans ma conception ?

En résumé, existe-t-il des outils pour analyser un cahier des charges ?

A.2 Définition de modélisation

La modélisation est une solution pour réussir un programme.

La définition de modélisation par Larousse est la suivante : Établissement de modèles, notamment des modèles utilisés en automatique, en informatique, en recherche opérationnelle et en économie.

Dans ce cours, je propose la définition suivante :

Modéliser signifie proposer une représentation, la plus précise possible, d'une réalité.

On passe du concret à l'abstrait.

A.3 Objectifs du cours

Les objectifs de ce cours – classés selon la taxonomie de Bloom – sont :

Mémoriser	Connaître les différentes méthodes de modélisation
Comprendre	Expliquer l'utilité de la modélisation Comprendre les différences entre Merise et UML
Appliquer	–
Analyser	–
Évaluer	–
Créer	–

B Les méthodes de modélisation

Ce paragraphe propose plusieurs méthodes de modélisation, sans rentrer dans les détails.

Nous allons distinguer les méthodes non étudiées en cours (IDEF/SADT, ORM, FAST...) puis nous aborderons les méthodes orientées objets (Merise et RUP).



Le contenu de ce paragraphe répond à la connaissance des différentes méthodes existantes. Les deux méthodes qui seront approfondies en cours sont Merise et RUP.

B.1 Les méthodes non-étudiées

B.1.1 IDEF/SADT

IDEF¹ est une famille de modélisation dans les sciences de l'informatique. Elle a été initiée par l'US Air Force dans les années 1970. Il y a jusqu'à 15 méthodes définies. En 1995, seuls les niveaux 0 à 4 étaient complètement définis :

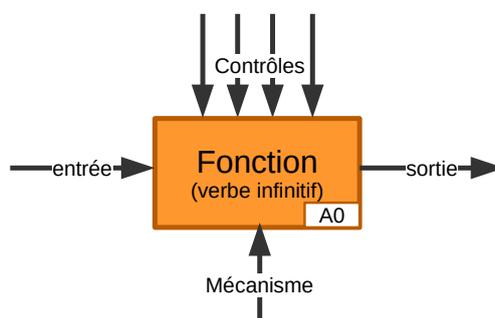
Première génération (ICAM)	IDEF0	Function modeling
	IDEF1	Information modeling
	IDEF2	Simulation model design
Deuxième génération (USAF)	IDEF1X	Data Modeling
Troisième génération (USAF)	IDEF3	Process description capture
	IDEF4	Object-oriented design
	IDEF5	Ontology description capture
En développement	IDEF6	Design rationale capture
<i>Envisagé</i>	IDEF7	<i>Information system auditing</i>
En développement	IDEF8	User interface modeling
En développement	IDEF9	Business constraint discovery
<i>Envisagé</i>	IDEF10	<i>Implementation architecture modeling</i>
<i>Envisagé</i>	IDEF11	<i>Information artifact modeling</i>
<i>Envisagé</i>	IDEF12	<i>Organisation modeling</i>
<i>Envisagé</i>	IDEF13	<i>Three schema mapping design</i>
En développement	IDEF14	Network Design

La méthode IDEF0 est en réalité dérivée de la méthode SADT². Nous verrons qu'UML ressemble à la famille IDEF.

1 IDEF : ICAM DEFinition, renommé en Integration DEFinition en 1999

2 SADT : structured analysis and design technique

Concrètement, la méthode IDEF0/SADT permet le découpage d'un système complexe en sous-blocs dont on peut expliquer les fonctionnalités. Il s'agit d'une méthode **hiérarchique**, car l'**analyse est descendante**.



Vous pouvez consulter la vidéo [La méthode SADT – SI – 1ère – Les bons profs](#) (3'10) pour en savoir plus sur cette méthode.

B.1.2 ORM/NIAM

La méthode ORM³ permet de modéliser des systèmes d'informations basés sur les bases de données relationnelles. Elle relie les objets par des relations. Contrairement aux autres modèles qui utilisent des techniques "entités-relations" (ER) et des approches "orientées objets", ORM n'utilise pas d'attribut explicite.

Autres méthodes	Méthode ORM
Personne <i>contient un attribut</i> PaysNaissance	Personne (être) né dans PaysNaissance

La méthode NIAM⁴ est une extension d'ORM. Elle provient d'un ingénieur chez Control Data Europe. Elle utilise une représentation graphique et un langage linéaire pour spécifier respectivement le schéma conceptuel de données et les contraintes.

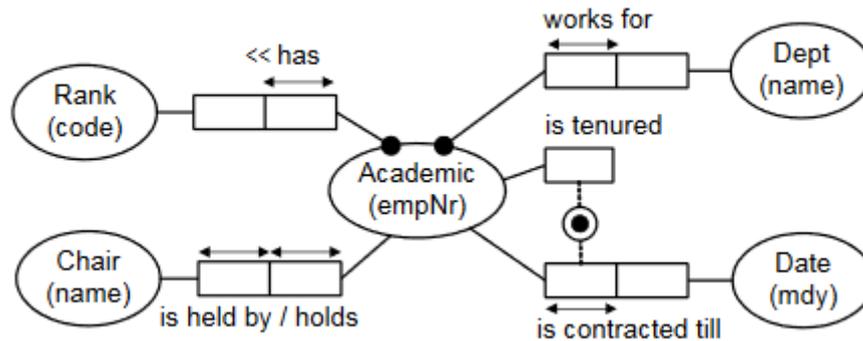
La [méthodologie](#) est la suivante :

Étape	Description
1	Transformer les exemples et informations en faits élémentaires. Appliquer des vérifications de qualités.
2	Dessiner le type des faits et appliquer une vérification sur une population de données
3	Tester les entités qui peuvent être combinées et noter les dérivations arithmétiques
4	Ajouter les contraintes uniques et vérifier l'arité des types de faits
5	Ajouter les contraintes de rôles nécessaires et noter les dérivations arithmétiques
6	Ajouter les valeurs, affecter les comparaisons et les contraintes de sous-types
7	Ajouter les autres contraintes et effectuer les vérifications finales

3 [ORM](#): Object-Role Modeling

4 [NIAM](#): Nijssen information analysis method

Le graphisme de la méthode ORM donne ceci :



La vidéo de Magic Monk, [ORM Lesson 4 – Writing elementary sentences](#) permet de décomposer la méthode ORM.



Attention, la vidéo fait partie d'un ensemble de leçons sur la méthode ORM. Bien que la méthode soit intéressante et puisse être réutilisée, Merise et UML fonctionnent différemment.

B.1.3 FAST

La méthode FAST⁵ analyse le pourquoi d'un système (à gauche) et le comment (à droite).

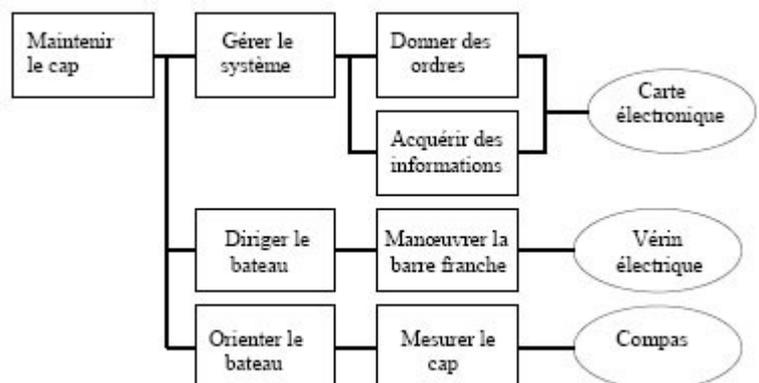
Les [étapes de construction](#) d'un diagramme FAST sont :

Étapes	Description
1	Rechercher toutes les fonctions qu'un produit doit proposer aux yeux du client
2	Chaque fonction doit être définie de la manière la plus générique possible
3	Rechercher les fonctions basiques et les fonctions secondaires (requisés, Inesthétique, Non-voulues)
4	Décrire comment chaque fonction est complétée (à droite de la fonction)

L'exemple ci-contre, montre la représentation d'un système de pilotage automatique de navire.

La réponse au "pourquoi" est "maintenir le cap".

La réponse au "comment" est "Mesurer le cap", "Manœuvrer la barre franche", etc.



Il s'agit à nouveau d'une sorte d'analyse descendante.

On trouve un peu plus d'informations sur [Wikipédia](#). Un autre site utile Canadien, [ValueAnalysis](#).

5 FAST : Function Analysis System Technique

Outre cette représentation graphique claire, la méthode pose les questions des **fonctions**, comme avec les méthodes hiérarchiques (SADT notamment).

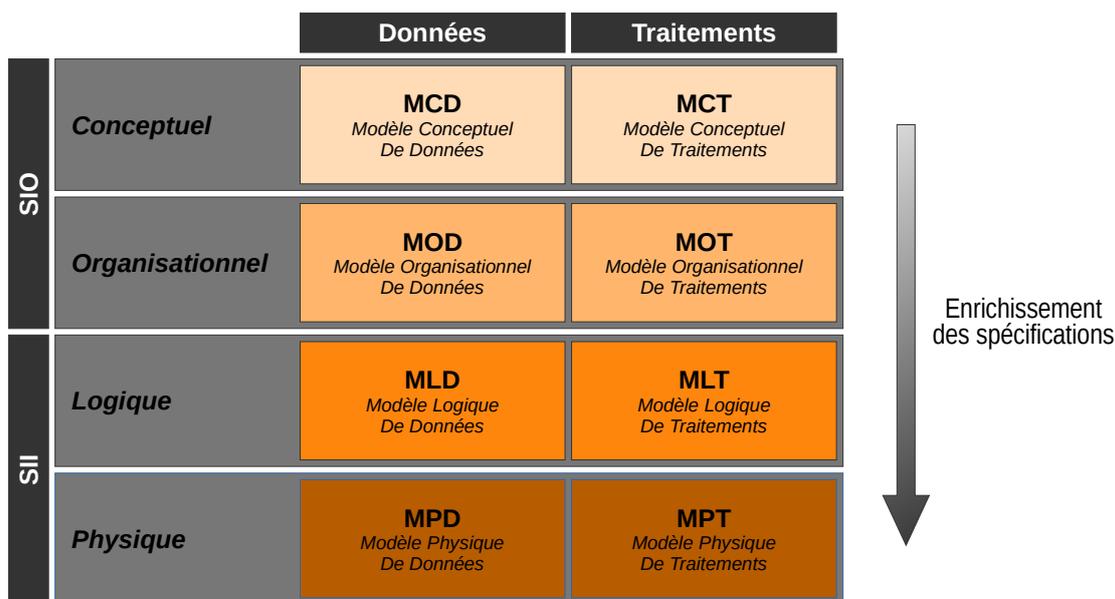
B.2 Les méthodes étudiées

En BTS SIO – et dans de nombreuses entreprises – les méthodes employées sont orientées objets (un peu comme ORM), cependant l'informatique n'ayant pas toujours été une science internationale, on trouvera la méthode française **Merise** dite **séquentielle** et une méthode plus récente, **RUP**, dite **itérative**.

B.2.1 Merise

Merise⁶ est une méthode fondée en France et dans les pays francophones dans les années 1970 à 1980. Issue de l'analyse systémique⁷, elle se veut généraliste et capable de modéliser tous les processus de conception.

Elle s'appuie sur les travaux d'Edgar Franck CODD (l'inventeur du modèle relationnel) et propose une modélisation graphique par entité-association (EA). Elle se rapproche des modélisations par **rôles**. Cette méthode gère les **données** et leurs **traitements**. Elle propose 4 niveaux d'analyse :



La partie "système d'information organisationnel" (SIO) est plutôt abstraite et ne se soucie pas des contraintes techniques.

La partie "système d'information informatisée" (SII) est liée aux moyens et aux ressources.

Merise est une méthode d'analyse qui propose son propre formalisme : MEA⁸. Cependant, il est possible d'utiliser un autre formalisme, comme UML.

6 Méthode d'Études et Réalisation Informatique des Systèmes Évolués : [https://fr.wikipedia.org/wiki/Merise_\(informatique\)](https://fr.wikipedia.org/wiki/Merise_(informatique))

7 Science des systèmes

8 MEA : Modèle Entité-Association

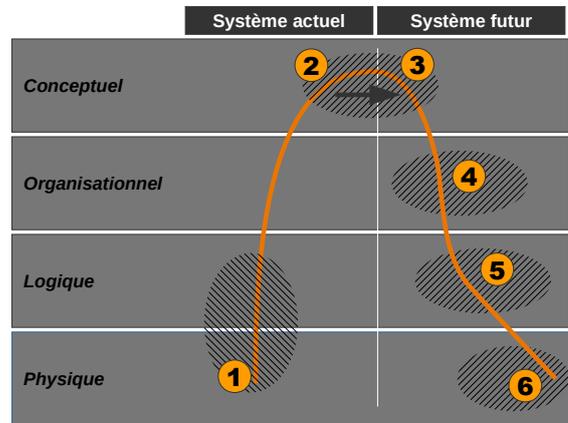
Merise met en évidence la notion d'état présent (le système physique d'origine) et l'état futur (le nouveau système physique) avec une progression de conception en forme de courbe du soleil.

Cette progression montre clairement qu'il faut observer le système physique, le comprendre pour pouvoir le conceptualiser.

Ensuite, il faut apporter des modifications sur le modèle conceptuel, y ajouter les contraintes organisationnelles.

Enfin, effectuer la transformation en modèle logique et l'implémentation sur le système physique.

Le schéma ci-contre décrit l'approche séquentielle.



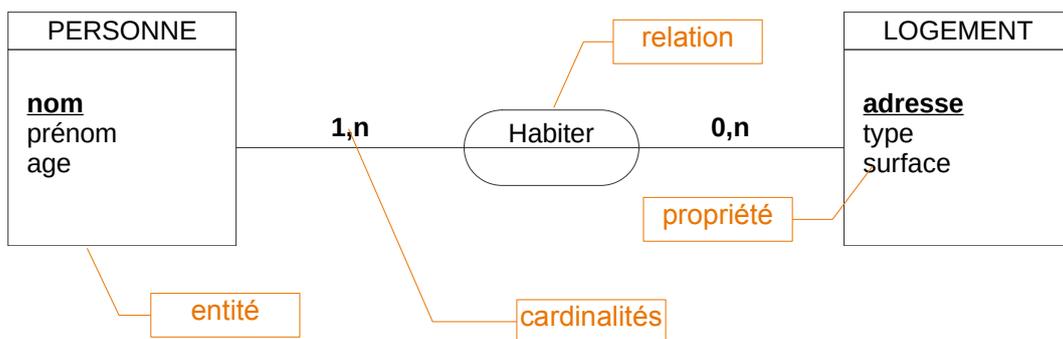
Merise étant une méthode permettant la conception d'un système dès le schéma directeur d'un projet, seule la partie de modélisation pour les bases de données relationnelles sera abordée en cours.

B.2.1.a MCD

Pour élaborer un MCD, Merise préconise l'usage de démarches déductives et/ou inductives :

- ◆ Démarche déductive : il faut constituer une liste d'informations
- ◆ démarche inductive : il est possible de construire le MCD à l'aide du formalisme Merise

Le formalisme de Merise est désigné par entité-relation (entity-relationship).



La relation entité-relation ci-dessus peut se lire dans les deux sens :

- une personne habite dans un ou plusieurs logement-s.
- un logement est habité par zéro à plusieurs personne-s.

Il est important de trouver les propriétés qui constituent l'entité : nous verrons dans un prochain cours, l'utilité d'une recherche pragmatique et les règles de distinguabilité.

B.2.1.b MOD

Le MOD est l'étape dans laquelle on spécifie des contraintes sur les données :

- durée de vie (transaction facture: 3 mois, client : 1 an, etc.)
- règles de calcul (montant total commande = somme des quantités * prix de vente)

B.2.1.c MLD

Le MLD est relativement facile à générer avec le MCD. L'écriture respecte le schéma d'une table :

```
PERSONNE(nom, prénom, age) ;  
LOGEMENT(adresse, type, surface) ;  
HABITER (#personne, #adresse, date).
```

On doit y trouver les contraintes d'unicité (en gras souligné) et les contraintes référentielles (liens sémantiques définis entre deux tables). Dans les bases de données relationnelles, on parle généralement de clés primaires et clés étrangères.

B.2.1.d MPD

La dernière étape est l'implémentation du MLD dans un SGBDR. L'exemple ci-dessous est la représentation SQL du MLD ci-dessus :

```
CREATE DATABASE maBase  
CREATE TABLE PERSONNE  
(  
    nom VARCHAR(100) PRIMARY KEY NOT NULL,  
    prenom VARCHAR(100),  
    age INTEGER  
)  
CREATE TABLE LOGEMENT  
(  
    adresse VARCHAR(255) PRIMARY KEY NOT NULL,  
    type VARCHAR(2),  
    surface INTEGER  
)  
CREATE TABLE HABITER  
(  
    nom VARCHAR(100) FOREIGN KEY REFERENCES PERSONNE (nom),  
    adresse VARCHAR(255) FOREIGN KEY REFERENCES LOGEMENT (adresse),  
    date DATE PRIMARY KEY  
)
```

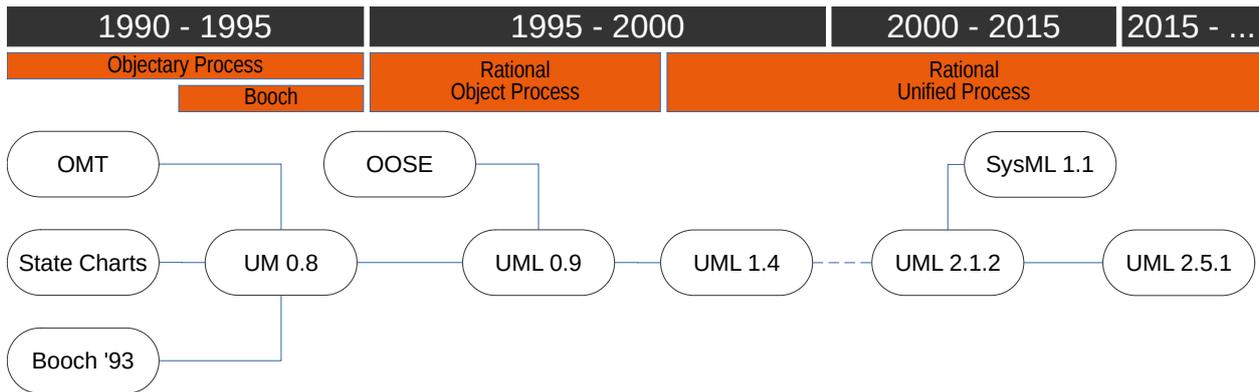


On trouve toutes les informations utiles et complètes sur la méthode Merise dans le livre de D. NANJI, B. ESPINASSE, B. COHEN, JC. ASSELBORN et H. HECKENROTH nommé [Ingénierie des systèmes d'information](#). Il est téléchargeable gratuitement en PDF.

B.2.2 RUP

Rational Unified Process est une méthodologie différente de Merise, car elle est itérative. Le projet informatique est construit progressivement et tient compte des commentaires dans chacune de ses phases.

La méthodologie RUP est une sérieuse concurrente à la méthodologie Merise et dispose d'une plus grande communauté internationale. Elle est le résultat de la fusion de plusieurs autres méthodes (et fusion de sociétés) et de l'itération successive de ses versions. Son évolution est indissociable de celle d'UML :

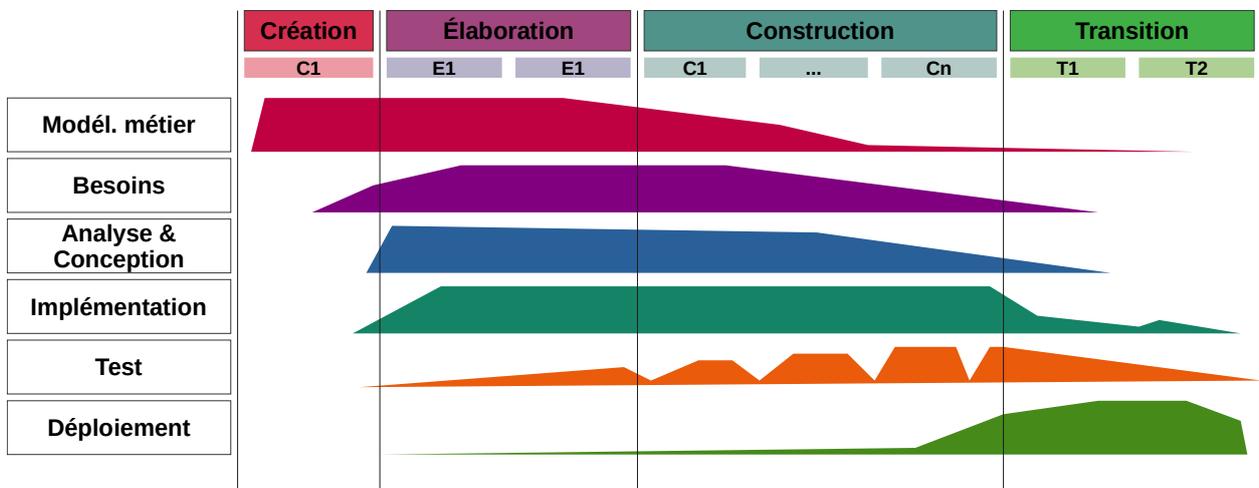


Les trois principaux fondateurs du langage UML proviennent de la division Rational Software d'IBM.

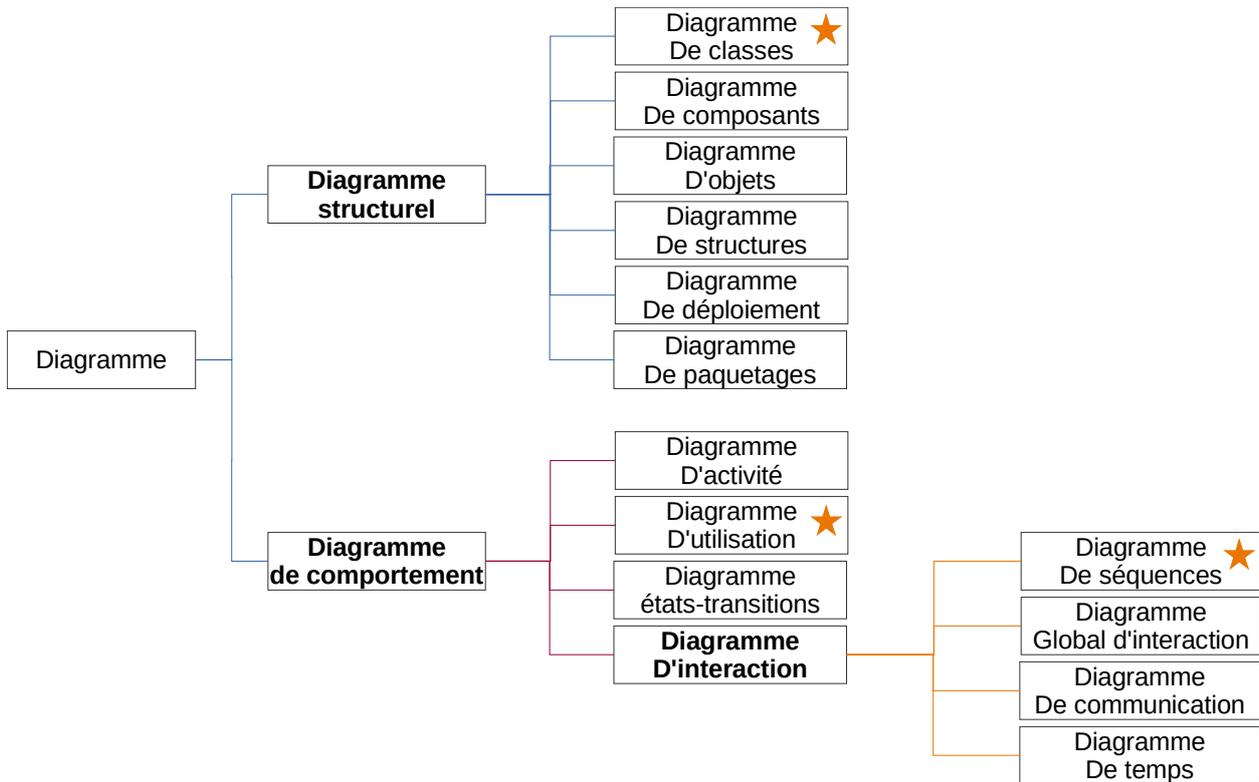
RUP se caractérise par :

- Un pilotage par les cas d'utilisation
- Une démarche centrée sur l'architecture
- Une approche basée sur les modèles (Model-based design)
- Une approche itérative et incrémentale (l'objectif est de réduire les incertitudes).

Le découpage du projet se fait sur quatre phases (Création, Élaboration, Construction et Transition) et porte sur l'enchaînement de plusieurs activités (Exigences, Analyse, Conception, Implémentation et Test) :



Comme IDEF, UML propose plusieurs modélisations, basées sur différentes vues.



Alors que Merise est une véritable méthode d'analyse, UML n'est un ensemble de diagrammes, qui cadrent les analyses dans chacun de leurs domaines.

La démarche UML est donc basée sur l'élaboration des modèles, plutôt qu'une séparation entre analyse et conception. Cependant, la force d'UML est de pouvoir modéliser les prototypes (notion de programmation objet) et les dépendances.

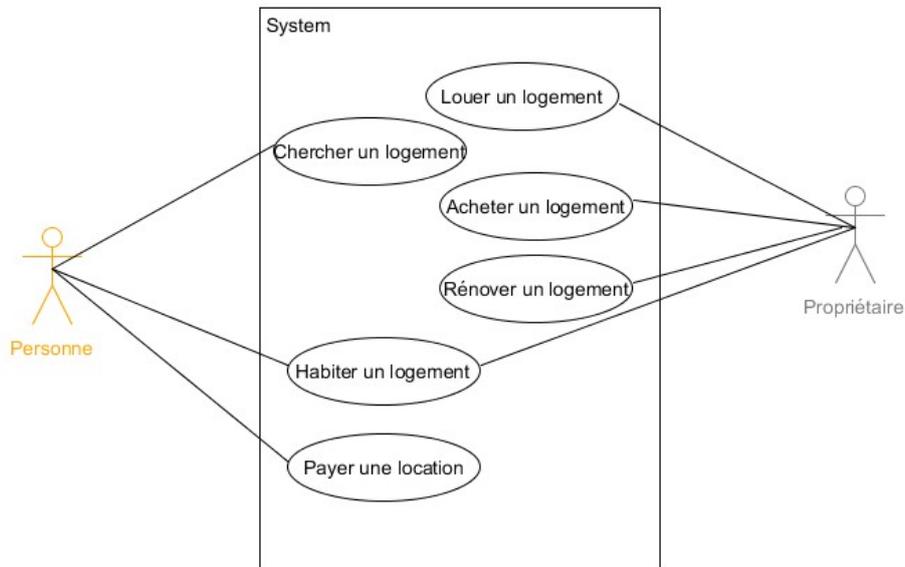
Les trois diagrammes importants que nous étudierons sont le diagramme d'utilisation, le diagramme de séquence et le diagramme de classe. Leur utilité dépend de la phase du projet étudié. Bien qu'il n'y ait pas de méthode UML, le tableau ci-dessous, démontre un certain niveau de structuration.

Les diagrammes en phase...		
...d'expression des besoins	...d'analyses et conceptions	...implémentation
<ul style="list-style-type: none"> - Diagrammes cas d'utilisation - Diagrammes de séquences 	<ul style="list-style-type: none"> - Diagrammes cas d'utilisation - Diagrammes de séquences - Diagrammes de classes - Diagrammes de composants - Diagrammes d'activités 	<ul style="list-style-type: none"> - Diagrammes de classes - packages et interfaces - Langages de programmation

Source : C. VIAL

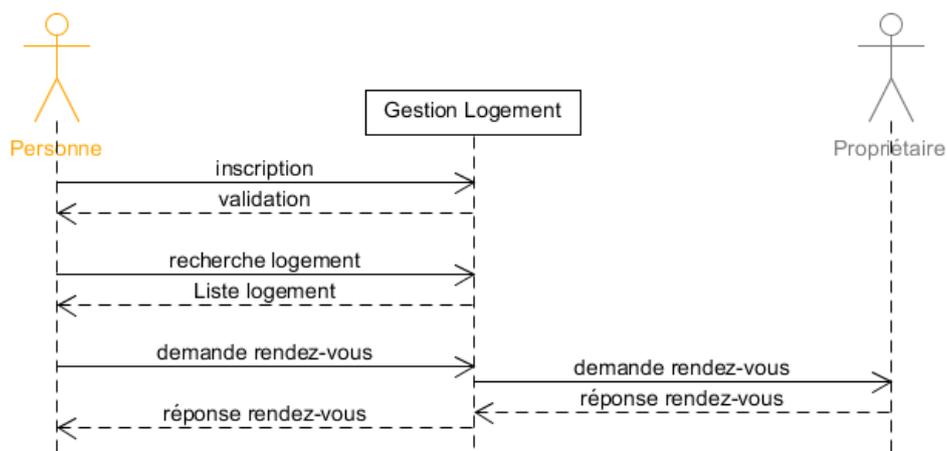
B.2.2.a Diagramme d'utilisation

Il s'agit du diagramme qui formalise la première étape : l'expression des besoins. Plutôt que de découper le projet en fonctions, UML propose une approche par **rôle** (les acteurs) et par **usage** (les cas d'utilisation). Ce diagramme permet de décrire précisément ce que le système doit faire, et peut servir de guide lors de l'implémentation, en utilisant les "stories" des méthodes de **projets agiles**.



B.2.2.b Diagramme de séquences

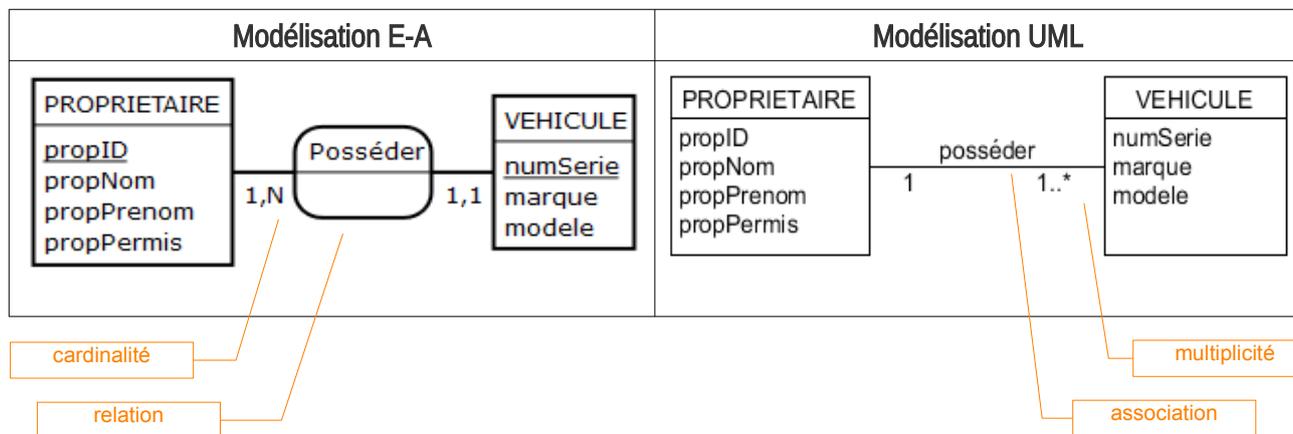
Le diagramme de séquences est une vue permettant de déterminer les réactions d'un système en fonction des sollicitations des acteurs. Cette vue peut servir lors de l'expression des besoins, parce qu'elle propose une perspective temporelle (les échanges dans le temps). Elle décrit les étapes des cas d'utilisation.



B.2.2.c Diagramme de classes

C'est le diagramme le plus proche du formalisme MEA de la méthode Merise. Les différences sont :

- la relation devient une association (disparition du symbole spécifique entre les entités)
- les cardinalités sont inversées (et se notent différemment en UML)



C Synthèse

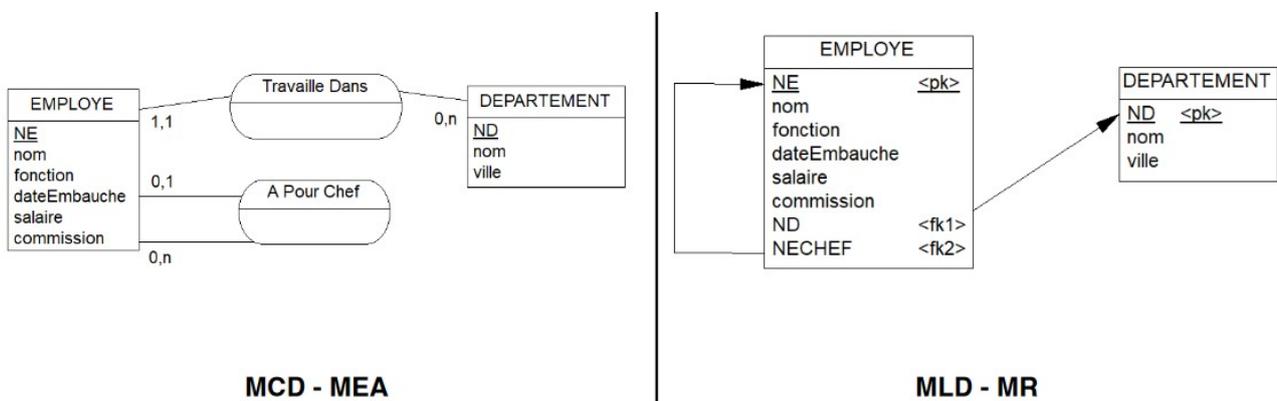
Finalement, il existe beaucoup d'outils et de méthodes pour analyser un système d'informations et le faire évoluer. Il reste difficile de trier les différentes méthodes de modélisation et de conception.

La particularité de **MEA** et de **UML** est que ce sont des modèles sagittaux.



Certains spécialistes parlent de classes de modèles :

- les **modèles "sagittaux"**⁹ (MS) qui n'utilisent pas de clés étrangères mais des représentations de relations qui suggèrent les attributs possibles dans les entités. MEA, UML sont dans cette classe.
- les **modèles relationnels** (MR) qui utilisent des clés étrangères. MLD, SQL font partie de cette classe. Ces modèles peuvent avoir des représentations graphiques (ci-dessous) ou textuelles.



Voir le cours de Bernard Liaudet : [Modélisation MR - MEA](#)

Merise propose le formalisme MEA, tandis que RUP intègre la modélisation UML.

Or, la difficulté de concrétiser un projet (un changement pour le client) reste importante. Ne rien oublier, penser à tout, définir un schéma de base de données qui garantira un usage pérenne à long terme... tout cela implique de savoir utiliser les outils présentés dans ce support.

Le prochain cours sera plus orienté sur les bonnes pratiques de la modélisation. Nous y verrons comment traduire les demandes en représentations facilitant le codage.

9 Sagitta signifie flèche en latin.

D Annexes

D.1 Sources complémentaires

Vidéos - ORM Lessons : https://www.youtube.com/watch?v=7vym87e0bfY&list=PLxumuDj9hbvrLM_GMPFC8TZdTtcJQyFtB

D.2 RUP

Wikipedia : https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9

NutCache : <https://www.nutcache.com/fr/blog/methode-rup/>

D.3 Cycle en cascade versus cycle itératif

