International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 10 October - 2025, Pages: 27-39

Comparative study between MERISE and UML as tools for analyzing and designing information systems.

[1]Charles KAKUNGA LUSHIMBA, [2]Trésor BANANTAMBA KALUNDE
Sciences informatiques, Institut Supérieur Pédagogique de Lwiza, ISP/Lwiza Kananga, République Démocratique du Congo
Olivierc287@gmail.com
Sciences informatiques, Université de Kananga, UNIKAN, Kananga, République Démocratique du Congo
banantambatresor02@gmail.com

Abstract: This scientific work explains a comparative study between MERISE and UML, which are two modeling tools used in the design of information systems. MERISE is a classic French method, which is distinguished by a structured approach in three levels (conceptual, logical, physical) and a clear separation between data and processing. It is especially suited to management systems and relational databases. Meanwhile, UML (Unified Modeling Language) is an object-oriented modeling language, generally adopted internationally. It allows to represent both the structure and the behavior of systems using many types of diagrams (use cases, classes, sequence, etc.), thus offering great flexibility. The comparison highlights that MERISE remains imperatively crucial in traditional contexts where design and data modeling are priorities, while UML presents itself as a modern tool, suitable for large-scale, complex, object-oriented projects and software development. The choice between the two depends mainly on the type of project, the design objectives and the skills of the team. On this, we dare to believe that in certain cases, a concatenation of the two approaches may be desirable, allowing to exploit the complementary capabilities of MERISE and UML.

Keywords **-** Unified Modeling Language, MERISE, Information System, analysis tools, design

Résumé

Cette œuvre scientifique explique une étude comparative entre MERISE et UML, qui sont deux outils de modélisation utilisés dans la conception des systèmes d'information. MERISE est une méthode française classique, qui se distingue par une approche structurée en trois niveaux (conceptuel, logique, physique) et une séparation nette entre données et traitement. Elle est surtout adaptée aux systèmes de gestion et aux bases de données relationnelles. Parallèlement, UML (Unified Modeling Language) est un langage de modélisation orienté objet, généralement adopté à l'échelle internationale. Il permet de représenter aussi bien la structure que le comportement des systèmes à l'aide de nombreux types de diagrammes (cas d'utilisation, classes, séquence, etc.), offrant ainsi une grande flexibilité.

La comparaison met en relief que MERISE reste impérativement crucial dans des contextes traditionnels où la conception et la modélisation des données sont prioritaires, tandis qu'UML se présente comme un outil moderne, adapté aux projets complexes de grandes envergures, orientés objets et au développement logiciel. Le choix entre les deux dépend principalement du type de projet, des objectifs de conception et des compétences de l'équipe. Sur ce, nous osons croire que dans certains cas, une concaténation des deux approches peut être souhaitable, permettant d'exploiter les capacités complémentaires de MERISE et d'UML.

0. INTRODUCTION

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé *analyse*. Il existe plusieurs méthodes d'analyse, cette étude fera l'objet d'une comparaison entre le langage UML et la méthode Merise.

L'évolution des technologies de l'information a conduit au développement de divers outils et méthodologies pour la conception de systèmes d'information. Parmi ceux-ci, MERISE et UML se distinguent par leurs approches respectives et leurs applications variées. À l'opposé, UML (Unified Modeling Language), développé dans un contexte international, est un langage de modélisation orienté objet, largement adopté pour sa flexibilité et sa capacité à représenter des systèmes complexes.

Cette étude comparative vise à analyser les forces et les faiblesses de MERISE et UML dans la conception de systèmes d'information, en mettant en lumière leur pertinence, leur adaptabilité et leur efficacité dans des contextes variés. En examinant les principes fondamentaux de chaque méthodologie, ainsi que leurs applications concrètes, nous espérons offrir une compréhension approfondie qui aidera les praticiens et les chercheurs à choisir la méthode la plus adaptée à leurs besoins.

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 10 October - 2025, Pages: 27-39

1. Présentation de la technologie MERISE et UML

Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu UML (UnifiedModifiedLanguage « *langage de modélisation objet unifié* »), qui est issu de la fusion des méthodes Booch, OMT (Object Modelling Technique) et OOSE (Object Oriented Software Engineering). [1]

Issu du terrain et fruit d'un travail d'experts reconnus, UML est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté UML et participent à son développement. En l'espace d'une poignée d'années seulement, UML est devenu un standard incontournable. Certes, MERISE (Méthode de Recherche Informatique pour le Système d'Entreprise), d'originaire Française, met l'accent sur la modélisation conceptuelle et permet de structurer les informations à travers des modèles successifs. Ceci nous amène à nous questionner sur les apports réels d'UML dans la modélisation et la place des méthodes dites « t*raditionnelles* » telle que MERISE. [2]

2. Cadre théorique

2.1. Système d'information

Le terme Système d'informa**tion comprend deux concepts. Pour mieux** comprendre ce terme « *système d'information* », il convient de bien appréhender les concepts qui le composent dont dans la suite, nous traiterons d'abord ces concepts de manière séparée.

2.1.1. Systèmes

Il n'existe pas une définition classique du concept « *système* », mais des approches définitionnelles orientées vers un secteur d'activités. Dans le domaine de gestion, l'approche définitionnelle du concept « *système* » est orientée vers le point de vue de LEIBNIZ, complétée par BERTANLAFFY et enrichie par ROSNAY, partant de Jean-Louis LEMOIGNE. Joël de ROSNAY dit alors qu'un « *système est un ensemble d'éléments vivants en interaction mutuelle et poursuivant un but commun* ».

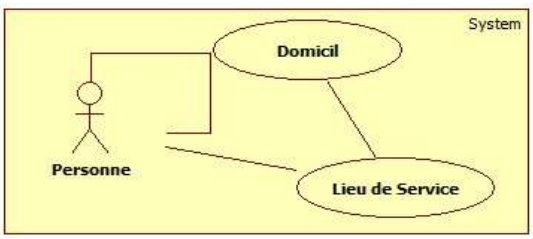


Figure 1. Présentation du système

2.1.2. Système d'Information

Un système d'information est un ensemble des moyens informatiques et de télécommunication qui permettent d'élaborer, traiter, stocker, acheminer, présenter ou détruire des données. Il est la partie informatique du système d'informations, composée des ressources matériels, humaines, logiciels et réseaux en interaction dynamique poursuivant un but commun, en dépit de ceci, nous considérons l'entreprise selon les catégories comme étant « système». [3]

2.2. Modélisation des données

Un modèle est une abstraction de la réalité, et l'abstraction est un des piliers de l'approche objet : il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise. Dans le même ordre d'idées, un modèle est une vue subjective mais pertinente de la réalité.

Un modèle définit une frontière entre la réalité et la perspective de l'observateur. Ce n'est pas "*la réalité*", mais une vue très subjective de la réalité. Par ailleurs nous disons, le caractère abstrait d'un modèle doit permettre, notamment : de faciliter la compréhension du système étudié, cet effet, il réduit la complexité du système étudié, de simuler le système étudié, en dépit de ceci, représente le système étudié et réduit ses comportements.

La modélisation des données est une approche intuitive pour concevoir une Base de Données. Elle est adaptée dès qu'on doit structurer un ensemble de données un peu complexe. C'est à chacun de juger quand un ensemble de données atteint son seuil, en adaptant à la théorie de la normalisation qui est plus formelle, plus rigoureuse et qui offre des moyens d'analyse très fins pour régler des problèmes sûr des sous-ensembles précis des données. [4]

2.3. Modélisation des processus

Le travail du concepteur de Système d'information est analogue au travail de l'architecte qui conçoit et dessine les plans d'un immeuble. Comme lui, le concepteur de Système d'information par sa spécification, à la fois, définit le futur Système d'information, décide de la solution technologique et prépare les tâches de réalisation.

Il est désormais professionnellement admis que le processus de modélisation d'un Système d'information s'organise globalement en deux phases. Première phase de « *conception* » qui aboutit à la spécification du Système d'information sous la forme d'un schéma, et une grande phase de « *réalisation* », qui conduit à la fabrication effective du Système d'information [5], comme l'illustre la figure ci-après.

Fig. 2 – Développement d'un Système d'information

2.4. Principes fondamentaux de MERISE

La Méthode MERISE d'analyse et de conception propose une démarche articulée simultanément selon 3 axes pour hiérarchiser les préoccupations et les questions auxquelles on répondra lors de la conduite d'un projet :

- **Cycle de vie :** phase de conception, de réalisation, de maintenance puis nouveau cycle de projet ;

- **Cycle de décision :** de grands choix (GO-NO GO : Etude préalable), la définition du projet (étude détaillé) jusqu'aux petites décisions, des détails de la réalisation et de la mise en œuvre du système d'information. Chaque étape est documentée et marquée par une prise de décision

- **Cycle d'abstraction :** niveaux conceptuels, d'organisation, logique et physique/opérationnel (du plus abstrait au plus concret). L'objectif du cycle de d'abstraction est de prendre d'abord les grandes décisions métiers, pour les principales activités (conceptuelles) sans rentrer dans le détail de questions d'ordre de l'organisation ou technique.

La méthode Merise très analytique (attention méthode systémique), distingue nettement les données et les traitements, même si les interactions entre les deux sont profondes et s'enrichissent mutuellement (validation des données par les traitements et réciproquement).

2.4.1. Diagrammes ou modèles utilisés (ex: MCD, MPD, etc.)

Les diagrammes ou modèle sur l'approche MERISE se basent sur *quatre niveaux organisationnels.*

Par ailleurs, chaque niveau d'abstraction comprend deux volets : données et traitement. Enfin, chaque modèle par un formalisme utilisant des concepts adaptés et qui seront explicités plus tard. Le tableau suivant résume les étapes du raisonnement selon chaque niveau.

Raisonnement de niveau d'abstraction selon Merise		
Niveau Conceptuel : Ce niveau représente les contenues de la base en terme conceptuel, indépendant de toute considération informatique c'est-à-dire au niveau conceptuel, on conçoit le système d'information en faisant abstraction des toutes les contraintes techniques ou organisationnelles et cela tant au niveau des données que les traitements. Il correspond à la description des finalités de l'entreprise en expliquant sa raison d'être et traduit les objectifs et contraintes qui pèsent sur elle. C'est à ce niveau que l'on découvre ce qu'il faut faire en répondant à la question « QUOI ? »(Le quoi faire, avec quelles données) ; on fait le choix sur la gestion.	Modèle conceptuel des données (MCD)	Modèle conceptuel de traitement (MCT)
Niveau organisationnel	Modèle organisationnel des données (MOD)	Modèle organisationnel de traitement (MOT)
Système d'information informatisé		

Niveau logique : Ce niveau permet de décrire les outils à utiliser pour la mise en place du système. Il est indépendant du matériel informatique, des langages des programmations ou de gestion des données. C'est la réponse à la question « AVEC QUOI ? » ou « AVEC QUEL OUTIL ?» ; On fait le choix sur le logiciel.	Modèle physique de données (MLD)	Modèle logique d traitement (MLT)
Niveau physique : Le niveau physique permet de définir l'organisation réelle physique des données. Il apporte les questions techniques en faisant le choix matériel ou technique pour le système d'information. On répond à la question « COMMENT ?»	Modèle physique des données (MPD)	Modèle physique de traitement (MPT)

Tableau 1. Raisonnement de niveau d'abstraction selon Merise

2.4.2. Les étapes de développement d'un système d'information

Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire l'objet d'un traitement automatisé grâce aux outils informatiques. Pour assurer la cohérence du système d'information, la méthode Merise propose une démarche d'informatisation comportant comme étapes : [6]

	DECISION
Schéma Directeur	**Approbation et mise en application,** dont le rôle est de définir, de manière globale, la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier et à développer.
	Choix d'une solution nouvelle ou arrêt. qui doit aboutir à une présentation générale du futur système de gestion en indiquant les principales novations par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans coût – avantage. Cette étude est réalisée en 4 phases :
Etude préalable par domaine	✚ Une phase de recueil qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel. ✚ Une phase de conception qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble. ✚ Une phase d'organisation dont l'objectif est de définir le système futur au niveau organisationnel : qui fait quoi ? ✚ Une phase d'appréciation dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation.
Etude détaillée par projet	**Accord des utilisateurs qui** consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion… A l'issue de cette étude, il est possible de définir le cahier des charges 2utilisateurs qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs.
Réalisation	dont l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.
Mise en œuvre	qui se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs.
Maintenance	qui consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques

Tableau 2. **Les étapes de développement d'un système d'information**

2.5. Principes fondamentaux d'UML

En effet, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche du processus dans le développement et ce dernier qui devrait favoriser la réussite d'un projet, ainsi libellé :

1. Une démarche itérative et incrémentale

Pour modéliser (comprendre et représenter) un système complexe, il vaut mieux s'y prendre en plusieurs fois, en affinant son analyse par étapes. Cette démarche doit aussi s'appliquer au cycle de développement dans son ensemble, en favorisant le prototypage. Le but est de mieux maîtriser la part d'inconnu et d'incertitudes qui caractérisent les systèmes complexes.

2. Une démarche pilotée par les besoins des utilisateurs

Avec UML, ce sont les utilisateurs qui guident la définition des modèles ; Le périmètre du système à modéliser est défini par les besoins des utilisateurs (les utilisateurs définissent ce que doit être le système). Le but du système à modéliser est de répondre aux besoins de ses utilisateurs (les utilisateurs sont les clients du système).

Les besoins des utilisateurs servent aussi de fil rouge, tout au long du cycle de développement (itératif et incrémental) :

• à chaque itération de la phase d'analyse, on clarifie, affine et valide les besoins des utilisateurs.

• à chaque itération de la phase de conception et de réalisation, on veille à la prise en compte des besoins des utilisateurs.

• à chaque itération de la phase de test, on vérifie que les besoins des utilisateurs sont satisfaits.

3. Une démarche centrée sur l'architecture

Une architecture adaptée est la clé de voûte du succès d'un développement. Elle décrit des choix stratégiques qui déterminent en grande partie les qualités du logiciel (adaptabilité, performances, fiabilité...).

Ph. Kruchten propose différentes perspectives, indépendantes et complémentaires, qui permettent de définir un modèle d'architecture (publication IEEE, 1995). Ph. Kruchten défend l'idée que l'architecture logicielle doit être une discipline à part entière.

Il propose que plusieurs perspectives concourent à l'expression de l'architecture d'un système et il explique qu'il est nécessaire de garantir la séparation et l'indépendance de ces différentes perspectives. L'évolution de l'une des perspectives ne doit pas avoir d'impact (sinon limité) sur les autres. La relation entre les différentes perspectives a été représentée par ph. Kruchten dans le schéma suivant, dit « schéma 4+1 vues ».

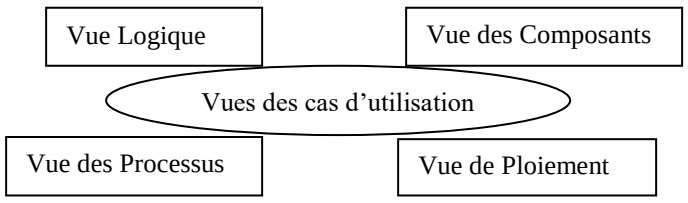


Figure 3. **Démarche centrée sur l'architecture du Schéma 4+1 vues**

a) La vue logique

Cette vue concerne « *l'intégrité de conception* », Appelée de haut niveau, elle se concentre sur l'abstraction et l'encapsulation ; elle modélise les éléments et mécanismes principaux du système, identifie les éléments du domaine, ainsi que les relations et interactions entre ces éléments « notions de classes et de relations »

b) La vue des composants

Cette vue concerne « *l'intégrité de gestion* ». Elle exprime la perspective physique de l'organisation du code en termes de modules, de composants et surtout des concepts du langage ou de l'environnement d'implémentation.

c) La vue des processus

Cette vue concerne « l'intégrité d'exécution ». Elle est très importante dans les environnements multitâches ; elle exprime la perspective sur les activités concurrentes et parallèles. Elle montre ainsi : la décomposition du système en termes de processus (tâches). Les interactions entre les processus (leur communication). La synchronisation et la communication des activités parallèles (threads).

d) La vue de déploiement

Cette vue concerne « l'intégrité de performance ». Elle exprime la répartition du système à travers un réseau de calculateurs et de nœuds logiques de traitements. Cette vue est particulièrement utile pour décrire la distribution d'un système réparti.

e) La vue des cas d'utilisation

Cette vue est particulière en ce sens qu'elle guide toutes les autres. Cette vue permet de trouver le « bon » modèle. Les cas d'utilisation permettent de guider la modélisation. L'utilisation des scénarios et des cas d'utilisation s'avère plus rigoureuse et plus systématique que les entretiens et l'analyse des documents pour découvrir les abstractions du domaine.

2.5.1. Diagrammes ou modèles utilisés

Dans le même ordre d'idée, le modèle utilisé en UML se base sur deux types de vues du système qui comporte chacune leurs propres diagrammes :

- **Les vues statiques**

Le but de la conceptualisation est de comprendre et structurer les besoins du client. : Il ne faut pas chercher l'exhaustivité, mais clarifier, filtrer et organiser les besoins.

- ***Diagrammes de cas d'utilisation*** : Les use cases permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs ils partent du principe que les objectifs du système sont tous motivés;
- **Diagrammes d'objets** : Le diagramme d'objets permet de mettre en évidence des liens entre les objets. Les objets, instances de classes, sont reliés par des liens, instances d'associations. A l'exception de la multiplicité, qui est explicitement indiquée, le diagramme d'objets utilise les mêmes concepts que le diagramme de classes;
- **Diagrammes de classes** : Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'information;
- **Diagrammes de composants** : Les diagrammes de composants décrivent les composants et leurs dépendances dans l'environnement de réalisation. En général, ils ne sont utilisés que pour des systèmes complexes;
- **Diagramme de ploiement** : Les diagrammes de déploiement montrent la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des instances de composants, processus et objets qui « vivent » sur ces matériels.

➢ **Les vues dynamiques**

- **Diagrammes de collaboration :** Le diagramme de collaboration permet de mettre en évidence les interactions entre les différents objets du système;
- **Diagrammes de séquence :** Le diagramme de séquence est une variante du diagramme de collaboration. Par opposition aux diagrammes de collaboration, les diagrammes de séquence possèdent intrinsèquement une dimension temporelle mais ne représente pas explicitement les liens entre les objets. ;
- **Diagrammes d'états-transitions :** Ils ont pour rôle de représenter les traitements (opérations) qui vont gérer le domaine étudié. Ils définissent l'enchaînement des états de classe et font donc apparaître l'ordonnancement des travaux.
- **Diagrammes d'activités :** Le diagramme d'activité est attaché à une catégorie de classe et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle".

2.5.2. Les éléments graphiques de base d'UML

Il existe quatre types d'éléments dans UML :

- Les éléments structurels : classe, interface, collaboration, cas d'utilisation, classe active, composant et nœud ;
- Les éléments comportementaux : messages et états ;
- Les éléments de regroupement : paquetages ;
- Les éléments d'annotation : commentaires.

Ces éléments sont liés par quatre types de relations :

- La relation de dépendance ;
- La relation d'association ;
- La relation de généralisation ;
- La relation de réalisation.

2.6. Comparaison entre Merise et UML

2.6.1. Méthodologie d'approche

2.6.1.1. Approche orientée données (Merise)

Pour avoir bien la maitrise de ces deux tâches, la méthode Merise dite Méthode d'analyse, de conception et de réalisation de systèmes d'informations informatisées, propose un ensemble de formalismes et de règles destinées à modéliser de manière indépendante les données et le traitement du système. Cette méthode n'est qu'une base de réflexion pour le concepteur et un moyen de communication entre les divers acteurs du système d'information. Seule la validation de l'ensemble se fera en commun.

Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire objet d'un traitement automatisé grâce aux outils informatiques. Pour assurer la cohérence du système d'information, la méthode Merise propose une démarche d'informatisation comportant les étapes suivantes :

☀ **Le Schéma directeur :** dont le rôle est de définir de manière globale la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier ou soit à développer. Pour rendre contrôlable et modulable ce développement, il est nécessaire de découper le système d'information en sous-ensemble appelé « domaine ». Par exemple, on peut le « domaine d'approvisionnement », le domaine « personnel », les résultats attendus à la fin de cette étape sont une définition précise du domaine, une planification du développement de chaque domaine et un plan détaillé, des applications qui doivent être réalisées. [7]

☀ **L'étude préalable par domaine :** qui doit aboutir à une présentation générale du futur système de gestion (modèle des données et des traitements) en indiquant les principales novations par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans coût – avantage. Cette étude est réalisée en 4 phases :

☀ Une phase de recueil qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel.

☀ Une phase de conception qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble.

☀ Une phase d'organisation dont l'objectif est de définir le système futur au niveau organisationnel : qui fait quoi ?

☀ Une phase d'appréciation dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation.

A cet effet, un découpage en projet est effectué.

1. l'étude détaillée par projet qui consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion… A l'issue de cette étude, il est possible de définir le cahier des charges utilisateurs qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs. Le fonctionnement détaillé du futur système, du point de vue de l'utilisateur, y est entièrement spécifié.
2. la réalisation dont l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.
3. la mise en œuvre qui se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs. Après une période d'exploitation de quelques mois, la recette définitive de l'application est prononcée. - la maintenance qui consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques. Le schéma suivant, extrait de l'ouvrage « La méthode Merise » reprend les étapes décrites ci-dessus.

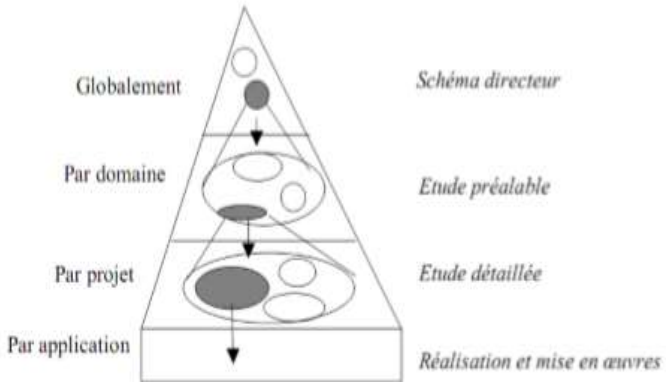


Figure. 4 : Etape de conception du système d'information (MERISE)

2.6.1.2. Approche orienté Objet UML

Un (en anglais Unified Modeling Language) ou « langage de modélisation unifié ». Est un langage de modélisation graphique à base de pictogramme. Il est apparu dans le monde du génie logiciel, il peut être appliqué à toutes sortes de système ne se limitant pas au domaine informatique.

Nous rappelons que « UML » reste un accomplissement de la fusion de précèdent langage de modélisation objet. Booch, OMT, OOSE. Principalement issu de travaux de Grady Booch, Jame Rumbaugh et Ivar Jackbson, Uml est à pr »sent un standard d »fini par l'objet management group (OMG). La dernière version diffusée par l'OMG est UML2, 3 depuis mai 210.

UML est avant tout un support de communication performant qui facilite la représentation et la compréhension de solution objet :

- ✓ Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évolution de solutions ;
- ✓ L'objet formel de sa notation, limite les ambiguïtés et les incompréhensions ;
- ✓ Son indépendance par rapport aux langages de programmation, aux domaines d'applications et aux processus, en font un langage universel.

La notation d'UML n'est que le support du langage. La véritable force d'UML, c'est qu'il repose sur un méta-modèle.

En d'autres termes, la puissance et l'intérêt d'UML, c'est qu'il normalise sémantiquement des concepts qu'ils véhiculent.

Qu'une association d'héritage entre deux classes soit représentée par une flèche terminée par un triangle ou un cercle, n'a que peu d'importance par rapport au sens que cela donne à votre modèle. La notation graphique est essentiellement guidée par des considérations esthétiques. [8]

Par contre, utiliser une relation d'héritage, reflète l'intention de donner à votre modèle un sens particulier, un bon langage de modélisation doit permettre à

n'importe qui de » déchiffrer cette intention de manière non équivoque, il est donc primordial de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les représenter. Le méta-modèle UML apporte une solution à ce problème fondamental.

UML est donc bien plus qu'un outil qui permet de 'dessiner » des représentations mentales…il permet de parler un langage commun, normaliser mais accessible.

- UML comme cadre d'un langage objet : une autre caractéristique importante d'UML, est qu'il cadre l'analyse. UML permet de représenter un système selon différentes vues complémentaires : les diagrammes ; un diagramme est une représentation graphique qui regroupe les éléments de modélisation. Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système).

Une caractéristique importante du diagramme UML, est qu'il supporte l'abstraction. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet. [9]

Ainsi, UML opte en effet pour l'élaboration des modèles, plutôt que pour une approche qui impose une barrière strictes entre analyse et conception. Les modèles d'analyse et de conception ne se différent que par leur niveau de détail, il n'y a pas de différences dans les concepts utilisés.

UML n'introduit pas d'éléments de modélisation propre à une activité (analyse, conception…).

UML favorise donc le pro-typage et c'est là l'une de ses forces. En effet, modéliser une application n'est une activité linéaire. Il s'agit d'une tâche très complexe, qui nécessite une approche interactive, car il est plus efficace de construire et valider par étapes, ce qui est difficile à cerner et maitriser.

UML permet non seulement de présenter et de manipuler les concepts objet, il sous-entend une démarche d'analyse qui permet de concevoir une solution objet de manière interactive, grâce aux diagrammes qui supportent l'abstraction.

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. UML est une sorte de boite à outils qui permet d'améliorer progressivement vos méthodes de travail, tout en préservant vos modèles de fonctionnement.

Intégrer UML par étapes dans un processus, de manière pragmatique, est tout à fait possible. La faculté d'UML se forge dans le processus courant, tout en véhiculant une démarche méthodologique.

2.6.1.3. Dualités des données-traitement

L'approche Merise considéré le système réel selon deux points de vue : une vue statique (les données) et une vue dynamique (le traitement). Il s'agit d'avoir une vision duale du système réel pour bénéficier de l'impression de relief qui en résulte et donc consolider et valider le système final.

L'approche UML basée sur les objets, associe les informations et traitement. De cette façon, elle assure un certain niveau de cohérence.

2.6.1.4. Tableau comparatif MERISE-UML

MERISE	UML
Méthode d'analyse et réalisation Informatique pour le Système d'Entreprise	Langage de modélisation Unifié
Méthode de conception de données et traitement relationnel	Système de notation orienté objet
Relationnel	Objet
Franco-français	International
Schéma directeur, étude préalable, étude détaillée et la réalisation.	Langage de modélisation des systèmes standard qui utilise des diagrammes pour représenter chaque aspect d'un système : statique et dynamique en s'appuyant sur la notion objet.
Plus adapté à une approche théorique. Du « battom up » de la base de données vers le code.	Plus orientée vers la conception. Du « top down » du modèle vers la base de données.
- La démarche de développement d'un système d'information de Merise doit être conduite suivant trois axes appelés cycle : ☀ Cycle de vie ; ☀ Cycle de décision ; ☀ Cycle d'abstraction : découpage en ensemble homogènes de préoccupations :	Est un métalangage et n'est pas une méthode, donc ne présente aucune démarche. Cependant les auteurs d'UML proposent d'utiliser une démarche pour

✓ Le niveau organisationnel qui détermine les choix de l'organisation. ✓ Le niveau technique : la réalisation d'un système d'information est conduite au travers d'un projet décomposé en étapes s'appuyant sur le trois cycles précédents.	favoriser la réussite du projet : - Une démarche interactive et incrémentale ; - Une démarche centrée sur l'architecture d'un logiciel (elle décrit des choix stratégique qui déterminent en grande partie les qualités du logiciel.
Découpage plus au travers de ses phases d'analyse métier et l'architecture logicielle.	Fonctionne sur un principe dite ration qui ne s'oppose pas aux phases définies dans Merise

Tableau 3 : comparaison de MERISE ET UML

2.7. Cas pratique

Dans le domaine de l'ingénierie logiciel et de la modélisation des systèmes d'information, deux méthodologies se démarquent par leur approche et leurs outils : Merise et UML.

Ce cas pratique vise à comparer ces deux méthodologies en termes de concepts, diagrammes, et d'application concrètes. Nous analysons les forces et les limites de chacune, tout en illustrant leur utilisation à travers un exemple concret de modélisation du système d'information**.**

Section 2.7.1. Etude de cas pratique d'utilisation MERISE

2.7.1.1. Description du projet

En voici une description concrète d'un projet développé avec la méthode Merise. « *La création d'un système de gestion de bibliothèque* »

- Contexte du projet, la bibliothèque de l'ISP LUIZA souhaite modéliser son système de gestion afin d'améliorer l'expérience utilisateur et d'optimiser la gestion des ressources.
- Objectif du système : permettre l'enregistrement des emprunts et retours de livres, gérer les abonnements des membres, et faciliter la recherche de livres par titre, auteur ou catégorie.

Eu égard à ce qui précède, le processus du développement de système d'information avec la merise se base sur l'entreprise en analysant le système au niveau du schéma directeur, à l'étude préalable jusqu'à sa mise en œuvre. A cet effet, nous allons regrouper les informations métier circulant au sein de l'ISP dans de modèles qu'on appelle « Entité » comme un objet du monde réel. Pour commencer, il est demandé de rechercher les différentes entités, attributs ou propriétés, relations ou associations et cardinalités pour assurer l'intégrité de données. [10]

A. Recherche des entités et attributs

- Livre (ID_Auteur, Nom,PostNom,Prenom,Sexe,Specialiste)
- Emprunteur {ID_Emprunteur,Nom,Prenom,Adresseligne,CodePostal,Commune,Téléphone,Email,Code barre emprunteur}
- Exemplaire{Code_Exemplaire, Date d'acquision,Durée de vie}
- Emprunter Exemplaire{Exemplaire,Date de etour au plus tard,ID_Emprunteur,Code_Exemplaire}
- Ouvrage{IdentifiantLibraire}
- Rayon{Salle,Royonanage,Etagère,Section}

B. Construction du Modèle Conceptuel de Données

A cette section on utilise WinDesign Comme outil de modélisation qui est orienté métier. En voici son Interface :

Figure 5: Environnement WinDesign

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 10 October - 2025, Pages: 27-39

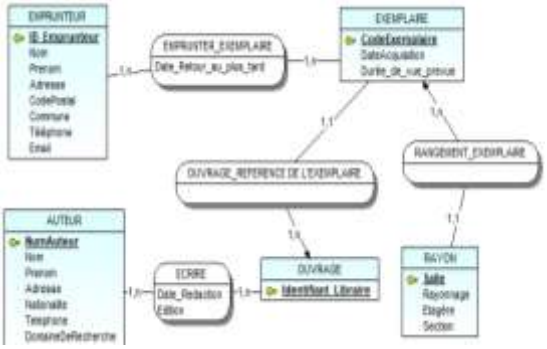


Figure 6 : Modèle Conceptuel de Données

C. Modèle Conceptuel de Traitement de l'Emprunt

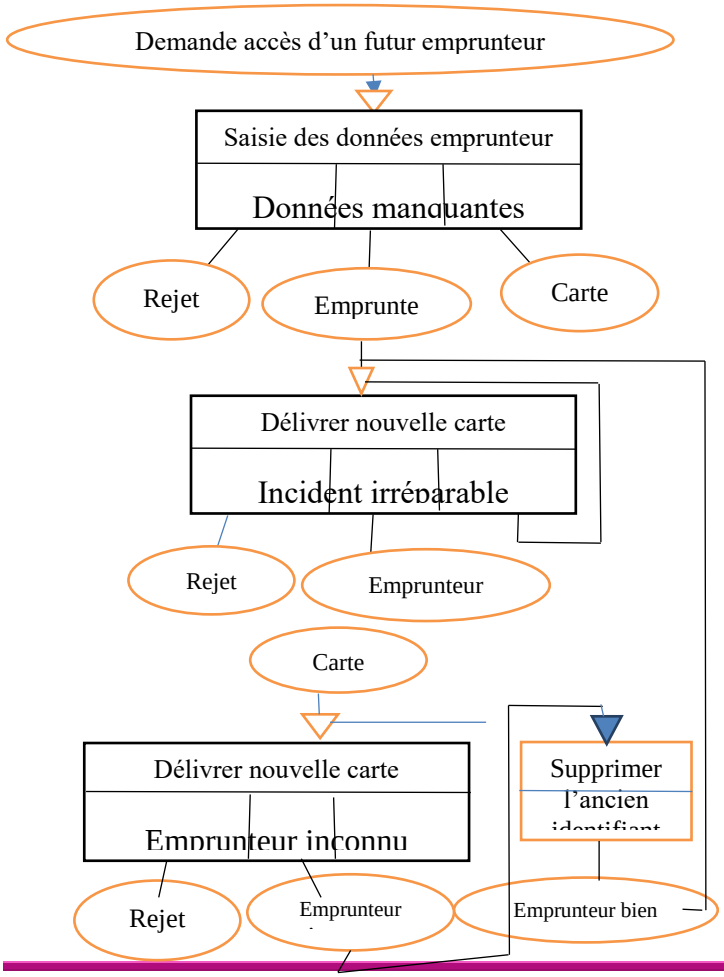


Figure 7 : modèle Conceptuel de Traitement

D. Règles de passage du MCD au MLD

Règle 1 : Chaque entité avec au moins un attribut non identifiant donne lieu à un schéma relationnel (table), les identifiants deviennent les clés.

Règle 2 : Les associations binaires de type 1: n donnent lieu à l'ajout de l'identifiant côté 1 vers le côté n, en tant qu'attribut non-clé (ou étrangère). Cette clé étrangère ne peut recevoir la valeur vide si la cardinalité est 1, 1.

Règle 3 : Les associations binaires de type n:m ou les associations non binaires donnent lieu à la création de nouveaux schémas relationnels supplémentaires (tables de jointure).

– Les identifiants des entités liées deviennent des clés.

– Les propriétés de l'association deviennent des attributs simples.

Règle 4 : Les associations binaires de type 1:1 sont traduites comme des associations binaires de type 1: n, sauf que la clé se voit imposer une contrainte d'unicité (sans doublon) en plus d'une éventuelle contrainte de non vacuité.

Si d'un côté, on a la cardinalité 0, 1. C'est dans la table côté opposé que devra aller la clé étrangère. Si les deux côtés sont de cardinalité 0, 1 alors la clé étrangère peut être placée indifféremment dans l'une de deux tables. Une alternative consiste à voir une association binaire de type 1 : 1 comme une association bina ire de type n : m particulière, il suffit pour cela d'ajouter une contrainte d'unicité sur chacune des clés étrangères de la table de jointure supplémentaire.

D. Construction du Modèle Logique de Données (MLD)

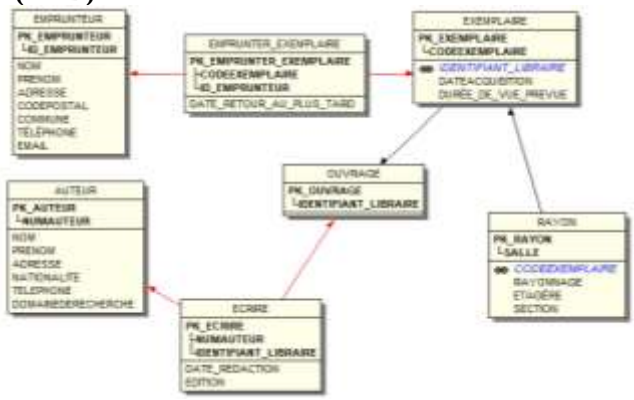


Figure 8 : Modèle Logique de données

A cette section, la Méthode Merise montre comment enchainer lorsque qu'on veut concevoir un modèle abstrait du monde réel afin de mémoriser les données

circulant dans l'entreprise, surtout en rapport avec la gestion de Bibliothèque de l'ISP/Luiza. Ici voyez qu'il y a un seul métier à informatiser. Raison pour laquelle nous disons que Merise Vise à un informatiser un métier. Il se concentre d'avantage sur la modélisation des données et des relations dans un format plus traditionnel et tabulaire. Dans les lignes qui suivent, nous verrons comment UML modélise et informatise.

Section 2.7.2. Etude de cas d'UML

2.7.2.1. Description du projet

Confère la description du projet approche MERISE, à cet effet, UML représente les données dans des diagrammes, à cette illustration nous allons utiliser entre autres le diagramme de Cas d'utilisation, de classe, et de séquence.

Dans le même ordre d'idée, UML c'est un langage de modélisation unifier orienté objet, montre la manipulation d'objets et des méthodes dans une classe qui est comme un ensemble d'objet (la couleur, Nom) et des méthodes, les opérations sur les objets (Supprimer un objet, affecter un objet, imprimer un état, etc.). Certes, pour modéliser, il y a les outils du marché dans cette optique nous utiliserons StartUml.

2.7.2.1.1. Diagramme de cas d'utilisation

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. [11]

C'est la description du modèle vu par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI). Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est dire, toutes les fonctionnalités que doit fournir le système.

Un cas d'utilisation est une unité cohérente d'une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

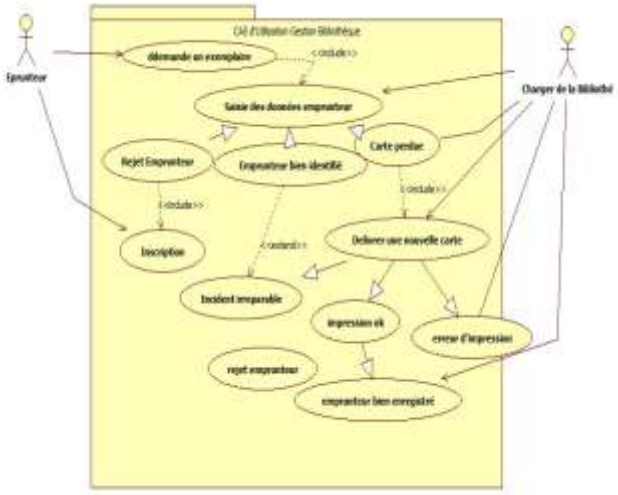


Figure 9 : Diagramme de cas d'utilisation

2.7.2.1.2. Diagramme de classe

Le diagramme de classe représente les classes intervenant dans le système.

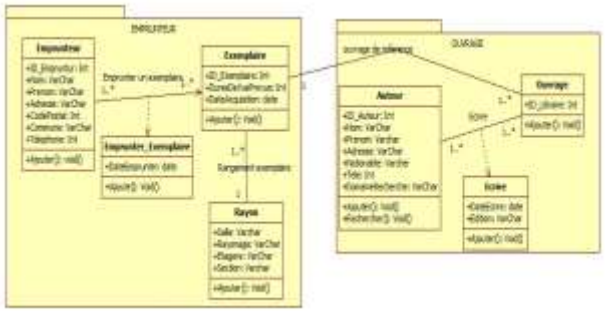


Figure 10 : Diagramme de Classe

2.7.2.1.3. Diagramme de séquence (Sequence Diagram)

C'est une représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

A cet effet, nous bouclons en disant qu'UML offre une approche orientée objet et permet de modéliser le comportement et interaction du système. Le diagramme ci-dessous montre le déroulement de l'Emprunteur au sein du Système.

Figure 11 : Diagramme de Séquence

CONCLUSION GENERALE

L'analyse comparative entre UML et MERISE révèle des atouts et des limites propres à chacune de ces méthodologies de conception de systèmes d'information.

A l'Approche et flexibilité, UML, avec sa nature orientée objet, offre une approche flexible adaptée aux systèmes complexes, permettant une modélisation variée à travers ses nombreux diagrammes (cas d'utilisation, classes, séquence, etc.). Cela la rend particulièrement appropriée pour le développement applicatif moderne. En revanche, MERISE se concentre sur une approche procédurale et se distingue par une séparation claire entre conception conceptuelle, logique et physique, ce qui peut faciliter la compréhension des données et des traitements, notamment dans des environnements moins complexes.

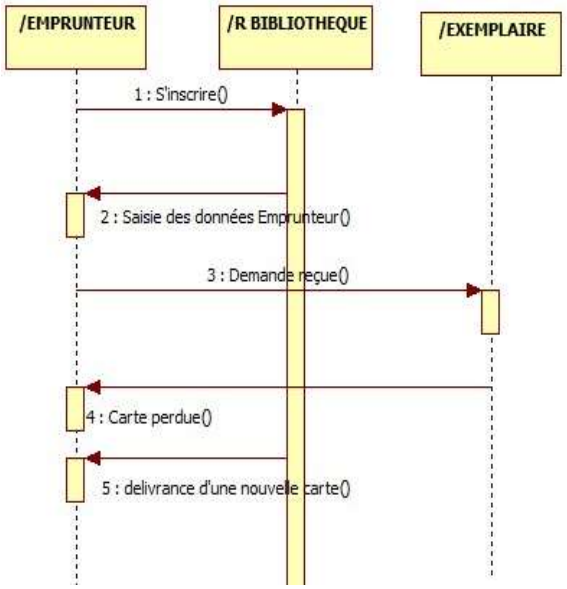
A la documentation et standardisation, UML est largement adopté à l'échelle internationale, jouissant d'une reconnaissance qui facilite la communication entre les équipes de développement, surtout dans des projets d'envergure. Sa standardisation permet une homogénéité des modèles dans divers projets. MERISE, bien qu'étant moins répandue en dehors des pays francophones, fournit une documentation riche qui peut aider à la gestion des savoirs dans des organisations plus traditionnelles.

A l'utilisation et adoption, La tendance actuelle dans le développement logiciel penche davantage vers UML, particulièrement dans les contextes agiles où la souplesse et l'adaptabilité sont essentielles. MERISE, quant à elle, est souvent utilisée dans des contextes où les systèmes d'information sont statiques ou évoluent lentement, notamment dans les administrations ou les grandes entreprises.

A l'issu des Outils et supports, UML bénéficie d'un large éventail d'outils logiciels, tels que Rational Rose ou Enterprise Architect, qui soutiennent la modélisation et la génération de code. MERISE a également des outils, mais ceux-ci sont souvent moins répandus et intégrés dans les solutions modernes de développement.

Certes, à propos de la perspectives d'évolution, UML continue d'évoluer avec les besoins du marché, intégrant des pratiques agiles et DevOps, MERISE pourrait bénéficier d'une mise à jour pour intégrer des concepts modernes de gestion des données et de modélisation orientée services. Toutefois, sa méthodologie reste pertinente dans des environnements spécifiques où une approche rigoureuse et structurée est requise.

En résumé, le choix entre UML et MERISE



dépend des spécificités du projet, des compétences de l'équipe, et des exigences de l'organisation. UML est plus adapté aux projets modernes complexes nécessitant flexibilité et adaptabilité, tandis que MERISE peut convenir à des systèmes d'information plus traditionnels et moins dynamiques. En définitive, une combinaison des deux approches pourrait s'avérer bénéfique pour tirer parti des forces de chaque méthode dans des contextes varié.

Cette conclusion permet de présenter une synthèse claire des points clés de l'étude comparative, tout en fournissant une vision des contextes dans lesquels chaque méthode peut être la plus efficace.

Références

[1] C.Bernard, *Introduction à UML*, paris, iut de Lyon, 2018, p.30.
[2] M.YassirBouhaddaoui, Comparatif UML – Merise, Ecole Nationale supérieur d'Ingénieur de CAEN, 2011, p.10.
[3] P. VALDURIEZ, « Objets complexes dans les systèmes de bases de données relationnels », Techniques et science informatique, Vol. 6, N° 5, 1987.
[4]https://www.manager-go.com (Organisation-entreprise/système d'information.htm
[5] Rapport introductif Modèles de structure de données dans les systèmes d'information, Séminaire international, Namur 1974.
[6] Piere Gérard, Modélisation de système d'information : DUT informatique 2ème année 2004/2005
[7] J. Rumbaugh, I, Jacobson et G. Booch, the Unified Modeling Language Reference Manual, Addisons-wesley, 1999. Op.cit.
[8] https://www.rational.com/uml: site de Rational Software corporation.
[9] UML en Français site web :https://uml.freefr/.

[10] R. ELMASRI, S. NAVATHE, *Conception et architecture des bases de données*, Pearson Education, 2004.
[11] Laurent AUDIBERT, *UML 2.0*, Institut Universitaire de Technologie de Villetaneuse – Département Informatique, 1ère Année, 2010-2011.