

## I. INTRODUCTION

La réalisation d'un site web doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un site afin de rendre ce développement plus fidèle aux besoins du client. La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités et La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation. Dans ce chapitre nous présentons une brève introduction au langage de modélisation UML, ainsi qu'une vue générale sur la méthode Processus Unifié UP.

## II. DEFINITION D'UML

Unified Modeling Language(UML) est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un formalisme graphique issu de notations employées dans différentes méthodes objets.

UML unifie également les notations et les concepts orientés objet (voir d'UML sur la figureII.1).Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage. Dans ce cadre, un concept appartenant aux exigences des utilisateurs projette sa réalité dans le modèle de conception et dans le codage.

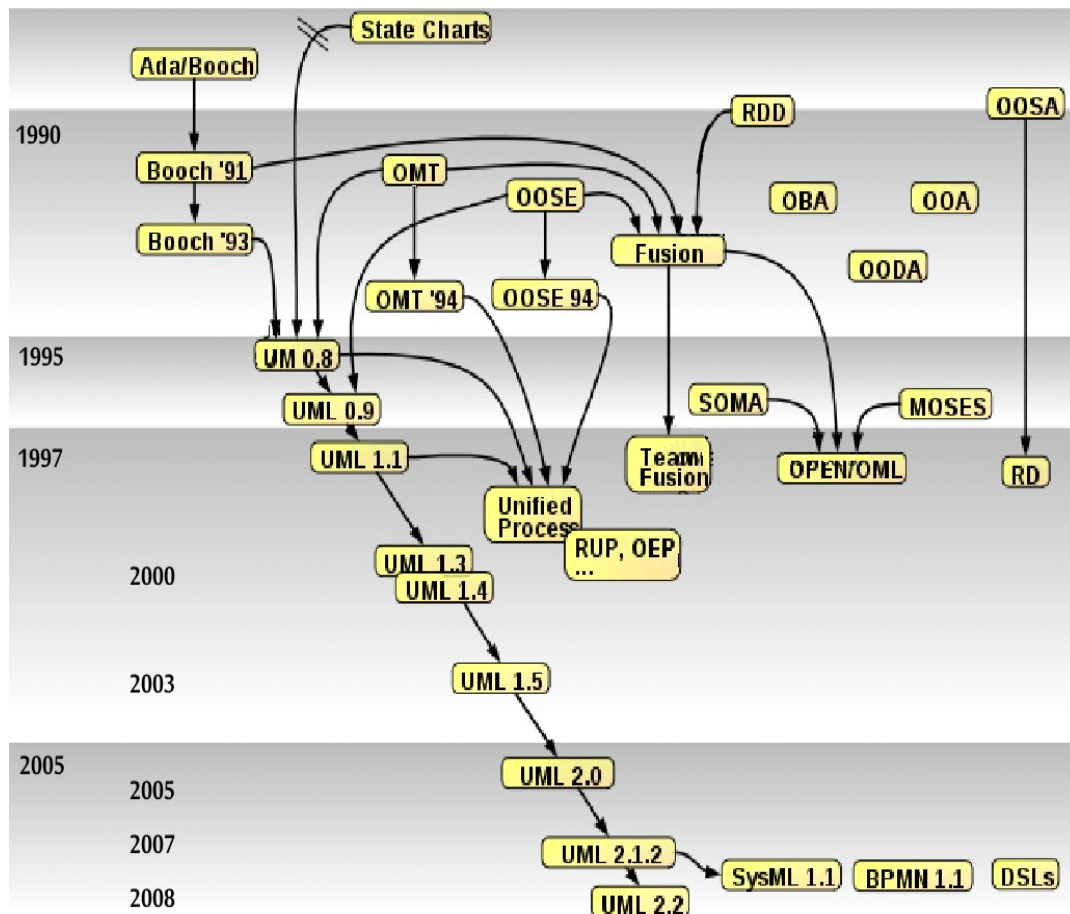


Figure II.1 : Historique d' UML

### III. LE FORMULAIRE D'UML

UML 2.3 propose 14 types de diagrammes (9 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML ; des méthodologies, telles que l'Unified Process, axent l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (Use Case). De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

UML se décompose en plusieurs sous-ensembles :

- **Les vues** : Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.

- **Les diagrammes :** Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- **Les modèles d'élément :** Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes. Exemple d'élément : cas d'utilisation, classe, association, etc

### III.1. Mise en œuvre d'une démarche à l'aide d'UML

#### III.1.1. les vues

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système :

Vue des cas d'utilisation : 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).

✓ **Vue logique :** c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).

✓ **Vue d'implémentation :** cette vue définit les dépendances entre les modules.

✓ **Vue des processus :** c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.

✓ **Vue de déploiement :** cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le OÙ).

#### III.1.2. Les diagrammes

Les 14 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

##### ➤ Diagrammes structurels ou statiques

Les diagrammes structurels ou statiques rassemblent :

✓ **Diagramme de classes :** il représente les classes intervenant dans le système.

✓ **Diagramme d'objets :** il sert à représenter les instances de classes (objets) utilisées dans le système.

✓ **Diagramme de composants :** il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)

✓ **Diagramme de *HYPERLINK*** : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.

✓ **Diagramme des *paquetages***: un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le diagramme de paquetage sert à représenter les dépendances entre paquetages, c'est-à-dire les dépendances entre ensembles de définitions.

✓ **Diagramme de *structure composite*** : depuis UML 2.x, permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

✓ **Diagramme de *profils***: depuis UML 2.2, permet de spécialiser, de personnaliser pour un domaine particulier un Meta-modèle de référence d'UML.

### ➤ **Diagrammes comportementaux**

Les diagrammes comportementaux rassemblent :

✓ **Diagramme des *cas d'utilisation***: il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

✓ **Diagramme *états-transitions***: permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.

✓ **Diagramme d'*activité***: permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

✓ **Diagramme de *séquence***: représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

✓ **Diagramme de *communication*** : depuis UML 2.x, représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.

✓ **Diagramme *global d'interaction***: depuis UML 2.x, permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences.

✓ **Diagramme de *temps*** : depuis UML 2.3, permet de décrire les variations d'une donnée au cours du temps.

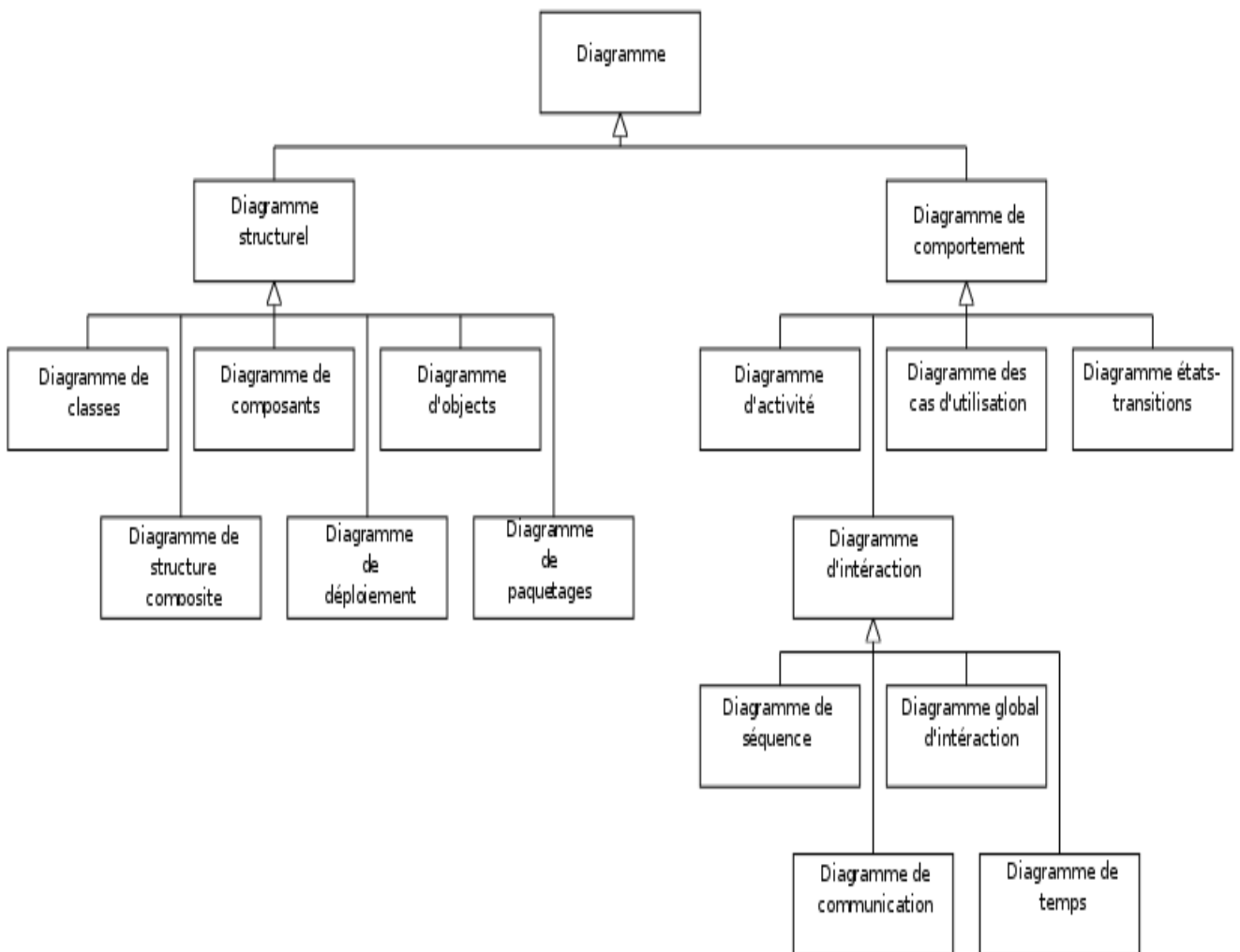


Figure II.2 : La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme

#### IV. QUE CE QU'UN DIAGRAMME UML

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle.

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).

Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système. Les diagrammes permettent donc d'inspecter un modèle selon différentes perspectives et guident l'utilisation des éléments de modélisation (les concepts objet), car ils possèdent une structure.

Une caractéristique importante des diagrammes UML, est qu'ils supportent l'abstraction. Cela permet de mieux contrôler la complexité dans l'expression et l'élaboration des solutions objet.

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

UML n'introduit pas d'éléments de modélisation propres à une activité (analyse, conception.) le langage reste le même à tous les niveaux d'abstraction.

Cette approche simplificatrice facilite le passage entre les niveaux d'abstraction. L'élaboration encourage une approche non linéaire, les "retours en arrière" entre niveaux d'abstraction différents sont facilités et la traçabilité entre modèles de niveaux différents est assurée par l'unicité du langage. Il s'agit d'une tâche très complexe, qui nécessite une approche itérative, car il est plus efficace de construire et valider par étapes, ce qui est difficile à cerner et maîtriser.

C'est donc avec beaucoup d'intérêt que nous avons pris connaissance de cette proposition de notre encadreur à utiliser comme langage de modélisation UML.

## V. LES AVANTAGES ET LES INCONVENIENTS d'UML

### V.1. Les points forts d'UML

UML est un langage formel et normalisé : Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.

UML est un support de communication performant :Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse en font un langage universel.

### V.2. Les points faibles d'UML

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

Même si l'Espéranto est une utopie, la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet. Or, l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue. Les auteurs d'UML sont tout à fait conscients de l'importance du

processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse.

## VI. LES METHODES ORIENTES OBJET

On propose ici la méthode que nous utilisons dans notre projet

### VI.1. Le Processus Unifié (UP)

#### VI.1.1. Définition

Pour définir le processus unifié, nous allons simplement définir les deux termes qui le composent :

##### ➤ **Processus**

Suite continue d'opérations constituant la manière de fabriquer. En d'autres termes, c'est une succession de tâches dans le but d'accomplir un travail, un projet.

##### ➤ **Unifié**

Etre amené à l'unité, se fondre en un tout. En fait, les méthodes d'analyse et de conception orientées objet, étaient variées jusqu'à ce que Rumbaugh, Jacobson et Booch eut l'idée de les unifier.

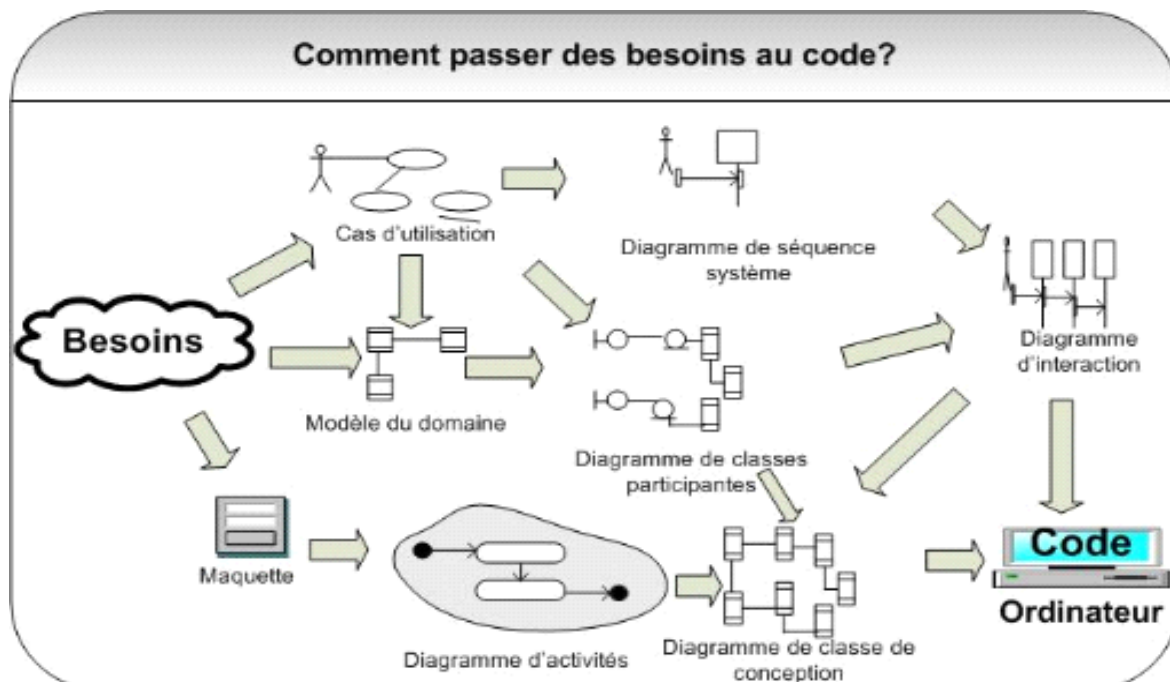


Figure II .3: la démarche utilisée

**VI.2. Caractéristiques**

D'après les auteurs d'UML, un processus de développement qui possède ces qualités devrait favoriser la réussite d'un projet. Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche :

- ✓ Itérative et incrémentale.
- ✓ Guidée par les besoins des utilisateurs du système, -Centrée sur l'architecture logicielle.

**VI.3. Une démarche itérative et incrémentale**

L'idée est simple : pour modéliser (comprendre et représenter) un système complexe, il faut mieux s'y prendre en plusieurs fois, en affinant son analyse par étapes. Cette démarche devrait aussi s'appliquer au cycle de développement dans Le but est de mieux maîtriser la part d'inconnu et d'incertitudes qui caractérisent les systèmes complexes.

**VII. CONCLUSION**

Dans ce chapitre Nous avons présenté quelques notions sur la modélisation avec le langage UML et la méthode UP, ces deux concepts sont utilisés dans la conception et la réalisation de notre projet.