Informatisation d'une médiathèque à travers la norme UML

Boubker Sbihi

Juillet 2005

Informatisation d'une médiathèque à travers la norme UML

Boubker Sbihi
Ecole des Sciences de l'Information
BP 6204 Agdal, Rabat, MAROC
Bsbihi@esi.ac.ma

Résumé

L'objectif de cet article est de proposer une informatisation à travers une application décentralisée sur différents postes qui a pour but la gestion d'une médiathèque exploitée par un nombre important d'utilisateurs.

Ce travail a été fait à travers le langage de la modélisation unifié UML. Ce choix est justifié par le fait que l'UML présente la fusion de plus de 250 méthodes orientées objets et que c'est la norme incontournable dans le domaine de la modélisation orientée objet.

La modélisation en UML de cette médiathèque a été suivie d'une mise en œuvre d'une application qui tourne sur des machines en réseau et qui va desservir les différents utilisateurs de ce système.

Notre travail donc, consistera à modéliser le système médiathèque et à rendre l'application simple à manipuler de façon a permettre aux responsables la bonne gestion sur les niveaux : insertion, mise à jour, et la recherche des emprunts et des adhérents.

MOTS-CLÉS : Informatisation, Analyse, Conception, Médiathèque, UML.

1. Le Système médiathèque

La médiathèque est un endroit de base essentiel dans chaque établissement universitaire. Cet espace contient toutes sortes de médias qui aident les étudiants dans leurs travaux d'études. Dans ce contexte, une bonne gestion des ressources (livres, films, dvd, cd, ...), nécessaire pour faciliter aux utilisateurs l'exploitation de ces éléments **[9].**

L'informatisation du système médiathèque à pour objectif la recherche, le recueil, la saisie, le traitement, le stockage et la communication de l'information permettant ainsi l'automatisation de certaines fonctions telles que : l'indexation, la sauvegarde et la recherche multicritères de l'information.

Cette prise en charge informatisée permet un allégement considérable de la charge de travail et une plus grande souplesse par rapport au traitement manuel de l'information. Les principales tâches que réalise le système médiathèque sont présentées dans la figure suivante (Figure 1) :

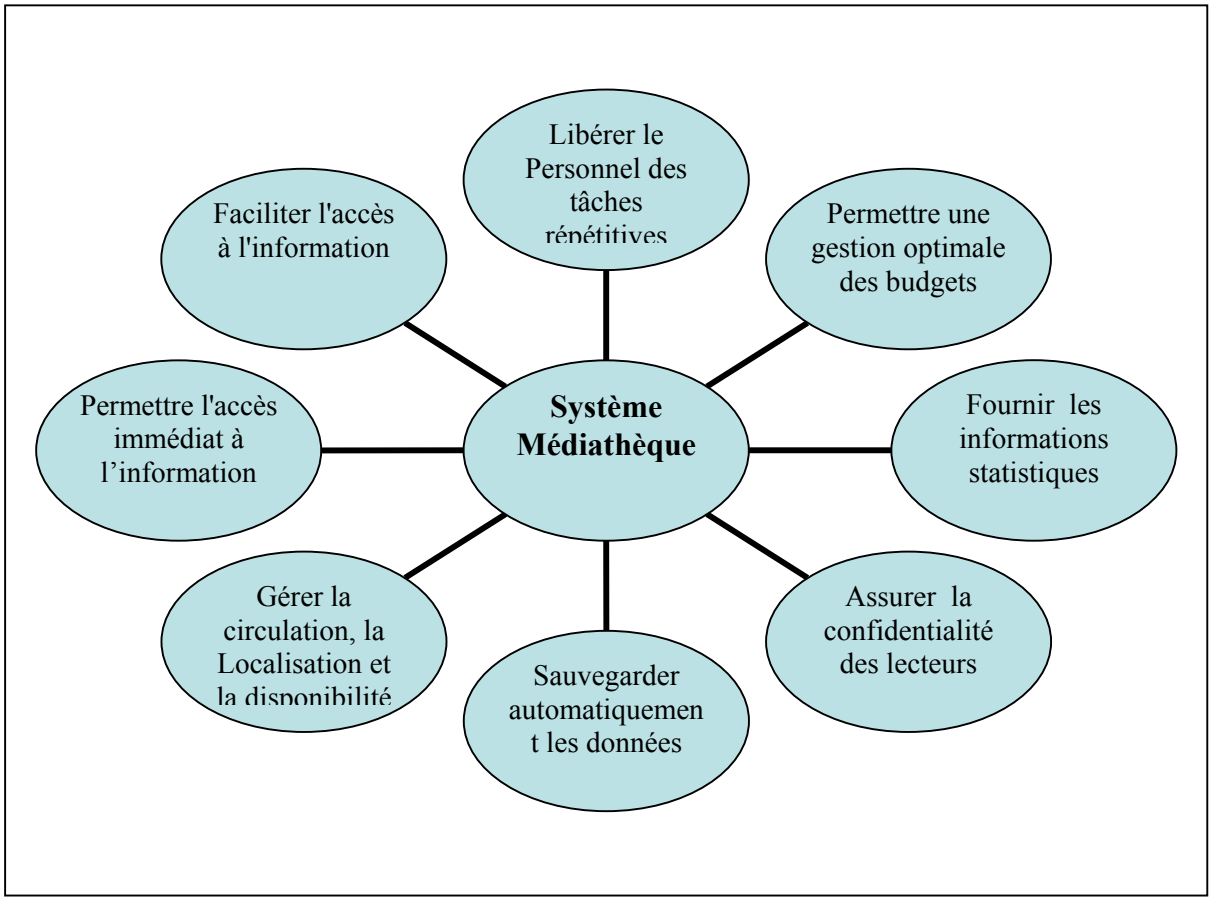

Figure 1: Les principales tâches du Système Médiathèque

Cette application doit permettre aux utilisateurs donc, d'insérer des données, de les consulter, de les modifier, de les mettre à jour, et de les rechercher dans la base de données selon des critères différents. Sur le plan technique, elle doit facilement être exploitable, évolutive et extensible.

En effet, le système informatique qu'on veut élaborer doit présenter un certain nombre de qualités qui garantiront l'utilisation optimale de ce dernier et permettront une facilité de gestion et d'exploitation. Les qualités visées par notre système sont présentées dans la figure suivante (Figure2):

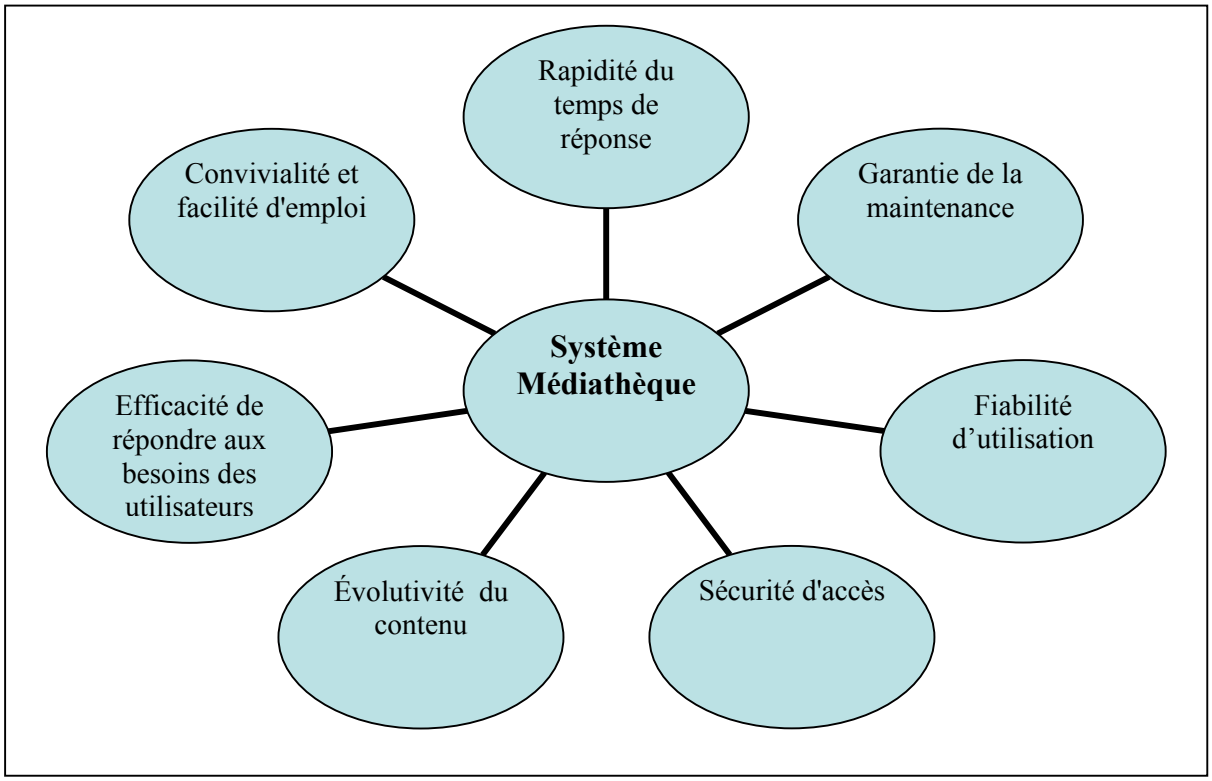

Figure 2 : Les principales qualités du système médiathèque

Afin de réaliser un système de qualité, une analyse et une conception normalisée selon la norme de modélisation universellement reconnue polyvalente et performante, s'avèrent nécessaires. En effet, Cette modélisation purement orientée objet permettra de représenter des concepts abstraits, de limiter les ambiguïtés en partant d'un langage commun, au vocabulaire précis, indépendant des langages orientés objet, et enfin nous facilitera l'analyse en simplifiant la comparaison et l'évaluation de solutions.

Il est tout à fait possible de produire un code syntaxiquement juste en adoptant une approche objet. En effet la programmation orientée objet implique une conception abstraite d'un modèle objet et en second plan l'implémentation à l'aide d'un langage orienté objet tel que Java et C++.

---

2. La présentation d'une médiathèque

Notre médiathèque contient un certain nombre de documents disponibles à la consultation ou à l'emprunt ; les personnes désirant emprunter des ouvrages pour une durée et à un tarif donnés doivent s'inscrire en tant que client **[9].**

L'inscription des clients consiste à remplir une fiche sur laquelle sont notées les informations concernant le client telles que son nom, son prénom ainsi que son adresse. Les clients peuvent choisir de payer à chaque emprunt effectué (type « à tarif normal » ou « à tarif réduit ») ou de régler une cotisation annuelle (type « abonné »). En plus du tarif, ce type conditionne les critères d'emprunt suivants : le nombre de documents empruntables et la durée de prêt. Dans le cas « tarif réduit », un justificatif est demandé à l'inscription, puis à chaque date anniversaire ; les cas à prévoir sont : étudiant/scolaire, carte vermeil et carte « à caractère social ».

Les documents disponibles sont des CD audio, des K7 vidéo ou des livres. Chaque document est repéré par un code unique et une localisation (salle/rayon) dans la médiathèque. Certains documents ne peuvent pas être empruntés, mais uniquement consultés sur place.

Les informations communes aux documents sont les suivantes : le titre, l'auteur (écrivain, groupe ou metteur en scène) et l'année d'édition. Enfin, afin de disposer de statistiques d'utilisation, on souhaite connaître le nombre d'emprunts effectués pour les différents types et genres de documents...

Les CD ont un genre musical (« classique », « variétés françaises », « variétés internationales », « compilation »...), une classification dans ce genre (par exemple, « opéra » pour le genre « classique », ou « rock » pour le genre « compilation »).

Les K7 ont un genre (« documentaire », « comédie »...), une durée d'émission et une mention légale de diffusion (restrictions d'usage) ; cette mention devra être disponible lors de l'emprunt de la K7 pour permettre un éventuel contrôle.

Les livres possèdent un genre (« roman », « policier »...) et un nombre de pages.

Les genres précisés sont libres ; ils sont donnés aux clients à titre indicatif pour aider au choix lors d'un emprunt.

Chaque sortie de documents entraîne la constitution d'une fiche d'emprunt. Sur cette fiche sont notés, le client emprunteur, la date de début du prêt et pour chaque document emprunté, la date limite de restitution. Les durées de prêt dépendent du type de document et de la catégorie du client.

Le système de gestion doit prévoir toute opération d'ajout et de suppression de clients et de documents. Les informations les concernant ne sont pas construites par le système (par exemple, la localisation des documents dans les locaux), mais supposées fournies lors de l'invocation de ces opérations. D'autre part, les formats de la plupart des informations sont libres (chaînes de caractères) ; le système devra toutefois veiller à la cohérence des

informations stockées (impossibilité d'avoir deux clients ou deux documents avec le même nom, emprunter deux fois le même document, etc.).

L'emprunt d'un document par un client obéit à certaines règles :

- Un client ne peut pas emprunter plus d'un certain nombre de documents fixé par son type (par exemple, 2 pour le type « à tarif réduit », 5 pour le type « à tarif normal » et 10 pour le type « abonné »). Dès que ce nombre maximal est atteint pour un client donné, tout nouveau prêt devra être impossible.
- Tout client qui n'a pas restitué un document avant sa date limite de restitution ne pourra plus faire de nouvel emprunt tant qu'il n'aura pas régularisé sa situation, ceci même si le nombre maximal d'emprunts n'est pas atteint. Pour ce faire, à chaque demande d'emprunt, on vérifie s'il est à jour dans ses restitutions ; si ce n'est pas le cas, l'emprunt n'est pas autorisé et le client est marqué de telle manière qu'il ne puisse plus emprunter de nouveaux documents. De plus, l'ensemble des fiches emprunt est parcouru chaque jour afin de repérer s'il existe des documents pour lesquels la date limite de restitution est dépassée ; pour chacune de ces fiches trouvées, le client concerné est marqué comme précédemment et la médiathèque envoie une lettre de rappel.
- Le tarif des emprunts dépend du document et du client. Le tarif du document est fixé par son type (par exemple, 0, 5 pour un livre, 1 pour un CD audio et 1, 5 pour une K7 vidéo). La règle pour les clients « à tarif normal » est de payer le montant fixé pour chaque document emprunté (indiqué auparavant). Le tarif appliqué aux clients « à tarif réduit » est la moitié du « tarif normal ». Les « abonnés » réglant une cotisation annuelle empruntent les documents gratuitement.
- La durée des emprunts dépend du document et du client. Chaque type de document est empruntable pour une durée dite nominale (par exemple, 2 semaines pour une K7 vidéo, 4 semaines pour un CD audio et 6 semaines pour un livre). Ensuite, la durée de prêt est modifiée selon le type de client (par exemple, aucun changement — la durée nominale — pour un client à tarif normal, la moitié de cette durée pour un client à tarif réduit et le double de cette durée pour un abonné).

---

3. Le langage de Modélisation Unifié UML

Durant la première moitié des années 90, il y'avait une cinquantaine de méthodes objet. Cette variance découle de la grande vitalité du paradigme objet et d'autre part de la multitude d'interprétations qu'il offre. Chaque méthode possédait ses propres avantages et ses caractéristiques. Dans le tableau qui suit (Tableau 1), nous présentons les éléments qui se sont dégagés de l'expérience de mise en oeuvre des différentes méthodes dans un effort de leur unification.

Méthode	Caractéristique
Booch	Sous-systèmes
Embley	Classes singletons et objets composites
Fusion	Description des opérations, numérotation des messages
Gamma	Frameworks, patterns et notes
Harel	Automates à états
OOSE	Cas d'utilisation (use cases)
Meyer	Pré- et post-conditions
Odell	Classification dynamique, éclairage sur les événements
OMT	Associations
Shlaer-Mellor	Cycle de vie des objets
Wirfs-Brock	Responsabilités (CRC)

Tableau 1: Les principaux éléments des célèbres méthodes orientées objets

Cependant; cette diversité méthodologique avait pour conséquence de créer une confusion et de limiter le progrès des méthodes en plaçant les utilisateurs dans une panoplie de choix.

En 1994, Jim Rumbaugh (OMT : Object Modeling Technique) et GradyBooch (BOOCH) décidèrent de proposer une méthode unique en unifiant leurs travaux, ce qui donna naissance à la méthode unifiée (*Unified method*) .

Ainsi, les deux variantes des méthodes BOOCH et OMT appelées respectivement Booch'93 et OMT-2, se sont rapprochées de façon considérable à tel point que les variations subsistantes sont devenues minimes **[4].**

Une année plus tard, ils ont été accompagné par Ivar Jacobson, le créateur des cas d'utilisation (*use cases) pour créer l'UML.* . La méthode unifiée se transforma donc en UML (Unified Modeling Language For Object Oriented Development)**[10][12].**

La version 1.1 soumise à l'OMG (object management group) en septembre 1997 fut acceptée à l'unanimité et devint de ce fait un standard. Des transformations continuelles furent effectuées pour supprimer les incohérences, apporter des améliorations et ajouter de nouveaux concepts.

En effet, la version 1.2 a apporté un remaniement de la forme par rapport à la version 1.1 mais aucun ajout d'ordre technique ne fut introduit.

La version 1.3 a apporté de nombreux changements, que ce soit en terme de correction ou d'ajout. Ainsi, il y a eu modification des associations entre cas d'utilisation, simplification des stéréotypes, changement d'éléments des graphes d'activités et des automates d'états....

La version 2.0 **[1] [2] [3]** a apporté des améliorations tout en ajoutant le dixième diagramme appelé diagramme d'architecture et l'ajout de la notion de profils. La figure suivante (Figure 3) montre l'historique de l'UML.

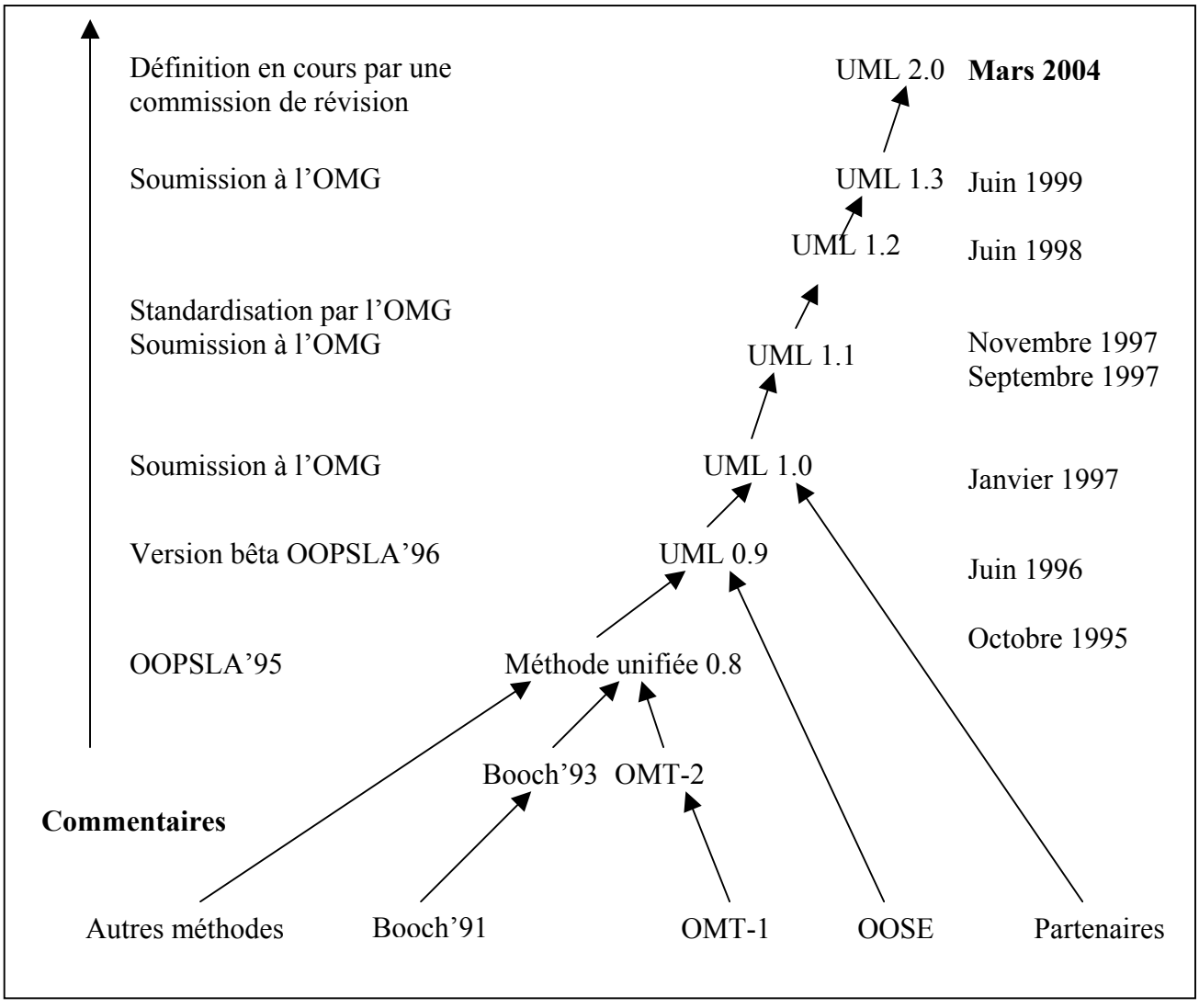

Figure 3 : L'évolution de la norme UML

UML, qui est le résultat de la fusion de plusieurs méthodes orientées objet est devenu sans conteste le langage de référence en terme de modélisation objet pour le développement de tout type de systèmes d'information base de données, e-business, application web, systèmes d'information d'entreprise, … .

Avec une seule représentation subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce à ces différents diagrammes. En effet, il couvre l'aspect statique et dynamique d'un système selon ses différents diagrammes.

Il définit pour cela dix diagrammes qui sont subdivisés en des vues statiques (qui représentent "physiquement" le système à modéliser au moyen de diagrammes d'objets, de classes, de cas d'utilisation, de composants, de déploiement et enfin d'architecture) et des vues dynamiques (qui montrent le fonctionnement du système au moyen de diagrammes de séquence, de collaboration, d'états transitions et d'activités).

La figure suivante (Figure 3) **[15]** montre les différents diagrammes de l'UML dans sa dernière version.

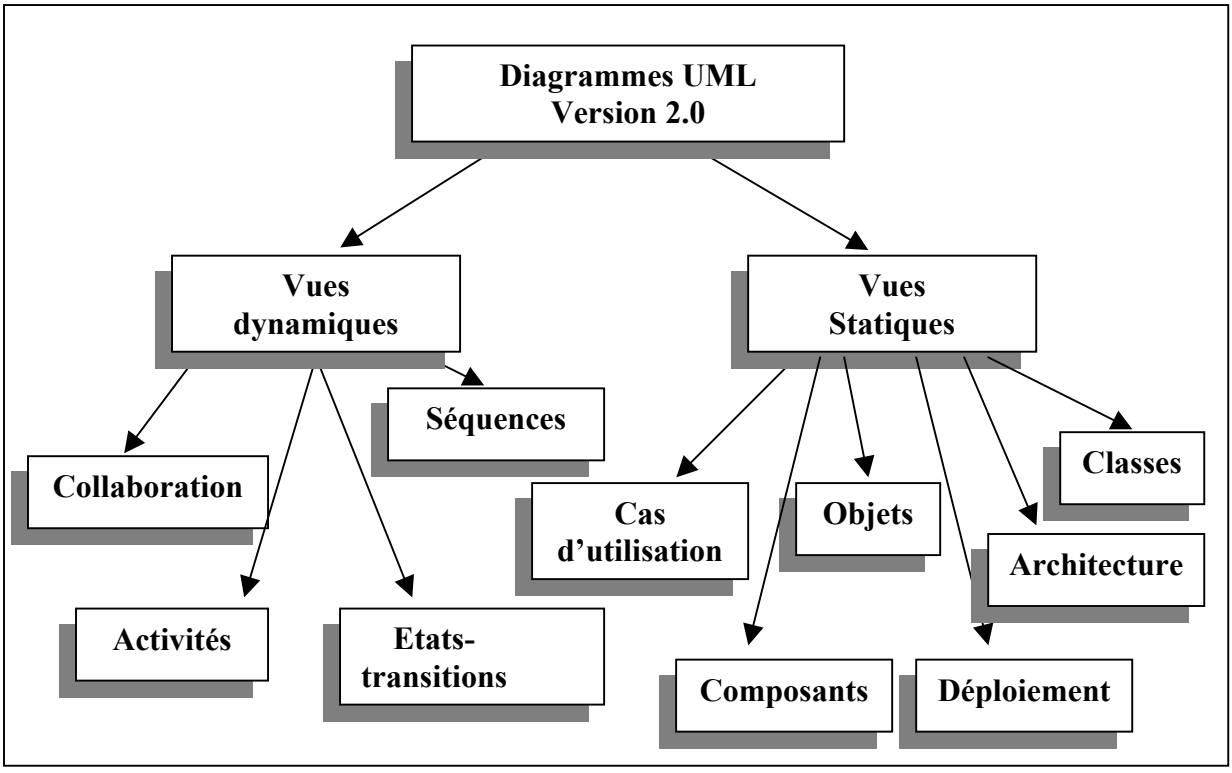

Figure 4 : Les différents diagrammes d'UML

L'avantage d'UML par rapport aux autres méthodes Booch, OMT, OOSE, c'est qu'il est plus expressif, plus propre et plus uniforme. Il supprime toutes les différences non nécessaires de notation et de terminologie qui obscurcissent les similarités de base de ces différentes approches.

UML définit ces diagrammes **[12]** pour représenter les différents points de vue de la modélisation. Ils permettent de spécifier, de construire, de documenter, de visualiser et de manipuler les systèmes informatiques, sachant bien que les diagrammes sont des " vues " qui permettent d'isoler certaines parties des modèles pour les rendre plus aisément compréhensibles. Les utilisations des diagrammes statiques de l'UML sont représentées dans le tableau suivant (Tableau 2):

Les diagrammes	Leur représentation
Les diagrammes de Cas d'utilisation	Représentent les fonctions du système
Les diagrammes de Classes	Représentent des classes et leurs relations
Les diagrammes d'Objets	Représentent les objets et leurs relations
Les diagrammes de Composants	Décrivent les composants physiques
Les diagrammes de Déploiement	Déploient les composants sur les matériels.
Les diagrammes d'architectures	Décrivent l'architecture du système a réaliser

Tableau 2 : Les six diagrammes de la modélisation statique de l'UML

Tandis que les utilisations des diagrammes dynamiques de l'UML sont représentées dans le tableau suivant (Tableau 3):

Les diagrammes	Leur représentation
Les diagrammes de Séquences	Représentent les objets et de leurs interactions
Les diagrammes de Collaborations	Représentent spatialement les objets et les liens
Les diagrammes d'États- Transitions	Modélisent le comportement d'une classe
Les diagrammes d'Activités	Expriment le comportement d'une opération

Tableau 3 : Les quatre diagrammes de la modélisation dynamique de l'UML

L'OMG a été doté de la nouvelle version de son langage de modélisation d'un modèle d'architecture. Le méta- modèle d'architecture de UML 2.0 **[1]** permet de définir les spécifications des composants, ainsi que l'architecture de l'application que l'on désire développer. En effet, UML 2.0 spécifie un composant comme étant une unité modulaire, réutilisable, qui interagit avec son environnement par l'intermédiaire de points d'interactions appelés ports. Les ports sont typés par les interfaces : celles-ci contiennent un ensemble d'opérations et de contraintes ; les ports peuvent être fournis ou requis.

Le standard UML permet aux informaticiens de concevoir des plans de systèmes afin de gérer et développer efficacement les solutions informatiques. Bon nombre d'outils atelier logiciel supportent aujourd'hui ce standard international de modélisation et de développement tel que la société Rational Software IBM Software Group, dont font partie les auteurs originaux d'UML. Il existe également les outils GDPro, ObjectTeam, Objecteering, OpenTool, Rational Rose, Rhapsody, STP, Visio, Visual Modeler,…..

Ces outils aideront à implémenter le diagramme de classes qui est le modèle de base pour la construction d'un logiciel du moment qu'il peut directement être traduit dans les différents langages de programmation tel que Java ou C++. Le modèle objet qui représente les briques du logiciel peut être directement déduit à partir du modèle de classes.

---

4. L'analyse des cas d'utilisations

La première étape consiste à bien comprendre le système à étudier pour mieux délimiter le système à étudier. La méthode générale **[5][6][7]**qu'on a adopté consiste à :

- Retrouver les acteurs qui interagissent avec le système.

- Rechercher les fonctionnalités du système par l'utilisation des « cas d'utilisation »

-Rechercher les différentes classes métier et leurs associations.

La figure suivante (Figure 5) montre la méthodologie adoptée et basée sur l'UML pour trouver les classes métier de notre système médiathèque.

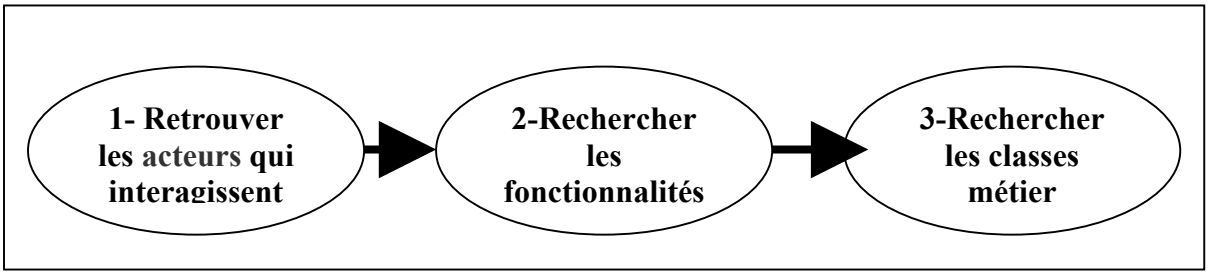

Figure 5 : La méthodologie adoptée basée sur l'UML

La médiathèque est un système qui permet la consultation ou l'emprunt d'un certain nombre de documents. Les adhérents à ce système qui veulent emprunter des ouvrages pour une durée et un tarif précis doivent s'inscrire en tant que client.

Dans notre étude des cas de ce système, on doit prévoir toutes les opérations entrantes dans les fonctionnalités des bibliothécaires puisqu'ils sont les seuls pouvant entraîner des modifications aux données du système.

On a procédé de façon à présenter généralement les cas d'utilisations du système et leurs acteurs, cette présentation fera l'objet du tableau suivant (Tableau 4).

Acteur	Rôles
Bibliothécaire	**Gestion des prêts**
	1. Enregistrer les prêts : - Entrer et vérifier ID -Vérifier le compte de l'adhérent s'il est bloqué -Vérifier le nombre des exemplaires empruntés par cet adhérent -Vérifier la disponibilité du support (non emprunté ou non réservé) -Enregistrer le prêt
	2. Enregistrer les restitutions - Entrer et vérifier ID - Débloquer le compte dans le cas d'un compte bloqué - Confirmer la restitution
	3-Envoyer un avertissement -Demander la liste des adhérents ayant dépassé le délai -Envoyer l'avertissement à l'adhérent -Bloquer le compte de l'adhérent
	Gestion des Inscriptions
	1. Inscription d'un adhérent -Recevoir les informations concernant l'adhérent -Enregistrer les informations -Envoyer carte adhérent.
	2. Suppression d'un adhérent -Recevoir une demande de des inscription -Expiration du délai de l'inutilisation du système
	3-filtrer les adhérents ayant dépassé le délai. -Retirer adhérent

	4. Modifier les informations d'un adhérent -Recevoir les informations -Mise à jour du compte
	Gestion des documents
	-Ajouter Document -Retirer Document
Adhérent	**1. Consulter** Demander liste des supports non encore empruntés
	2. Réserver -Entrer ID -Remplir le formulaire de réservation -Envoyer le formulaire
	3. Restitution ouvrage -Entrer ID -Remplir le formulaire de restitution -Envoyer le formulaire
Comptable	**1**-Gérer les comptes de la bibliothèque

Tableau 4 : Les acteurs du système médiathèque et leurs responsabilités

Les cas d'utilisation décrivent les fonctionnalités fournies par le système à un acteur du système. Ils sont utilisés par les clients, les concepteurs, les développeurs, et les testeurs.

Un cas d'utilisation est une description générique d'une utilisation du système et joue le rôle des scénarios de merise **[8]** . Les fonctionnalités d'un système sont décrites donc dans le modèle des cas d'utilisations par un ensemble de cas d'utilisation. Les cas d'utilisation supportent la notion d'Extension qui permet d'ajouter des cas d'utilisation pour gérer des cas spéciaux d'un cas d'utilisation. Le diagramme des cas d'utilisations est présenté dans la figure suivante comme suit (Figure 6):

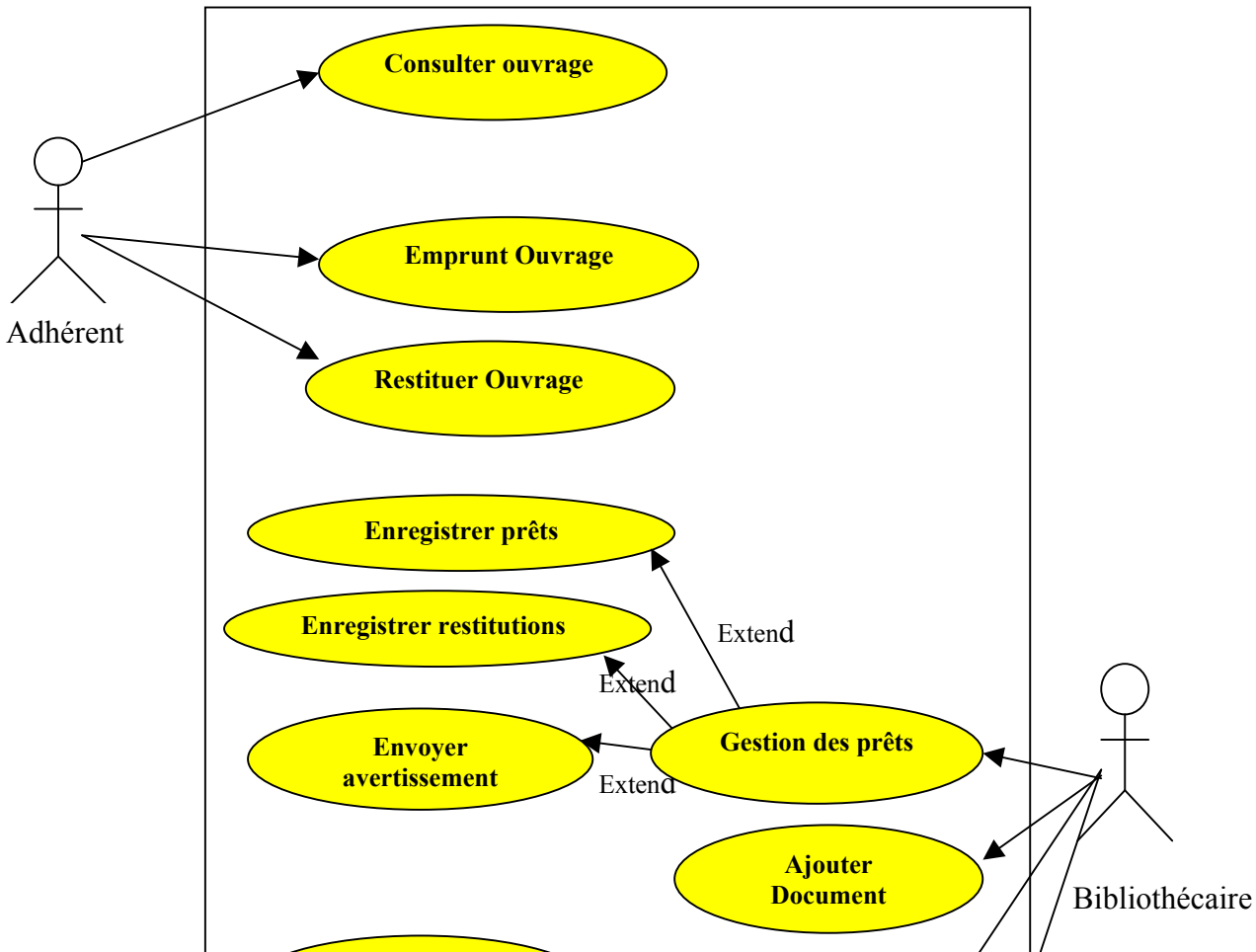

Figure 6 : Le diagramme des cas d'utilisations du système médiathèque

5. Les diagrammes de séquences

L'objectif du diagramme de séquences **[15]** est de montrer les interactions entre les objets du système sur une échelle de temps. Il est surtout utilisé pour étudier les interactions qui dépendent du temps (problèmes de synchronisation, d'ordonnancement...).
On distingue deux opérations à exécuter :

- La première est l'opération emprunter.
- La seconde est l'opération restituer.

L'opération emprunter

Lors d'un emprunt une création d'une instance de Fiche Emprunt est faite par le système

- L'adhérent fournit son identification et le titre de l'œuvre a emprunté.
- Le bibliothécaire vérifie si l'emprunt est possible
- Le bibliothécaire crée d'une fiche d'emprunt
- Le bibliothécaire détermine de la durée du prêt.
- Le bibliothécaire incrémente le nombre d'Emprunts effectués par l'adhérent
- Le bibliothécaire décrémente le nombre d'exemplaires empruntés de l'oeuvre.
- Le bibliothécaire envoie l'exemplaire à l'adhérent

La figure suivante (Figure 7) montre le diagramme de séquences de l'opération emprunt :

.

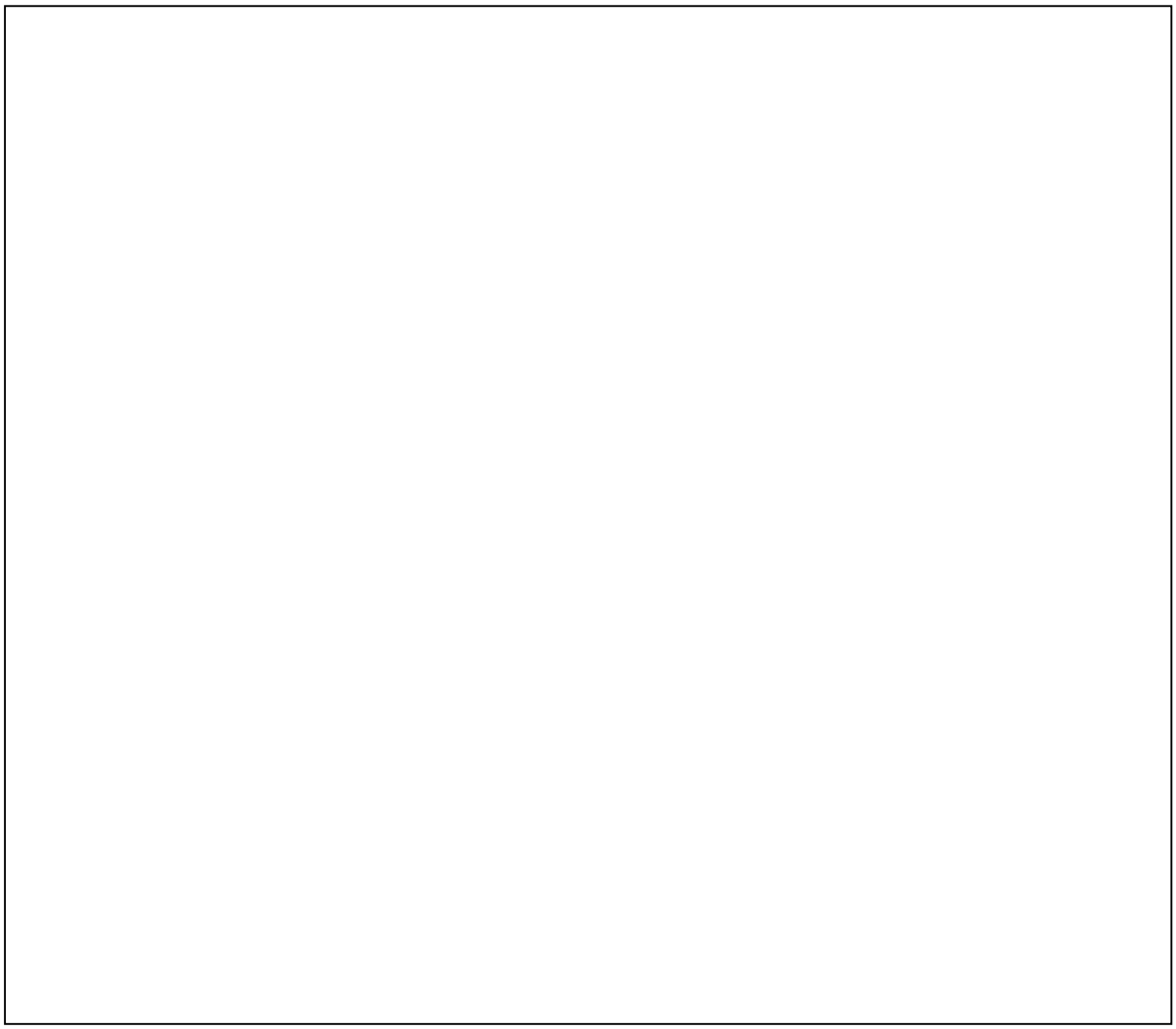

Figure 7 : Le diagramme de séquences de l'opération emprunt

La deuxième et dernière opération que notre système médiathèque utilise est celle de la restitution des emprunts.

L'opération restituer

- L'adhérent restitue l'œuvre qu'il a empruntée.
- Le bibliothécaire cherche la fiche d'emprunt relative à l'œuvre restituée.
- Le bibliothécaire incrémente le nombre d'exemplaires.
- Le bibliothécaire décrémente le nombre d'emprunts de l'adhérent.
- Le bibliothécaire détruit la fiche d'emprunt.

La figure suivante (Figure 8) montre le diagramme de séquences de l'opération emprunt :

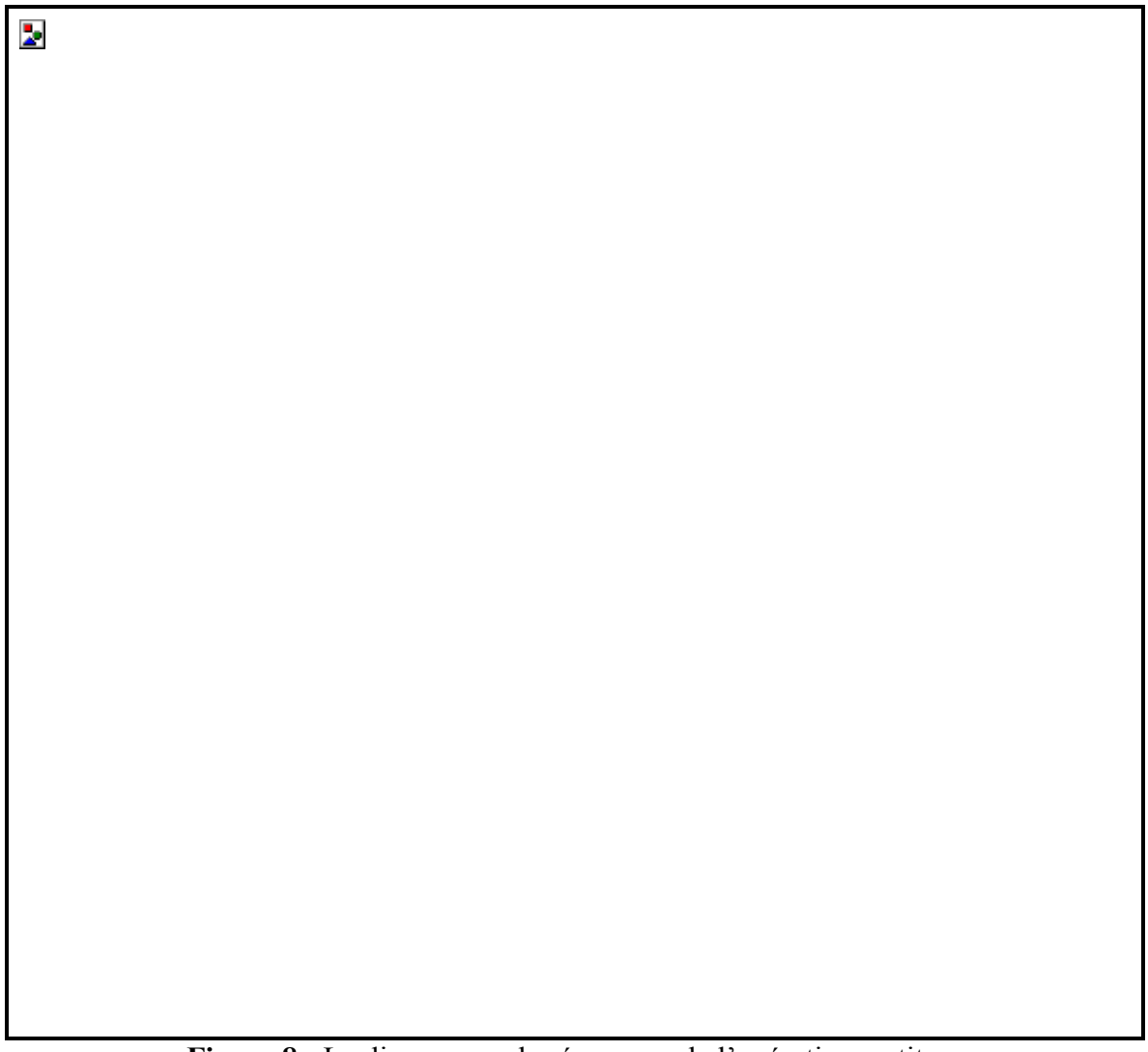

Figure 8 : Le diagramme de séquences de l'opération restituer

6. Le diagramme de Classes

Le diagramme de classes donne une vue statique du système logiciel puisqu'il décrit les types et leurs objets de ce dernier. Typiquement, il met en relation des classes mais aussi des interfaces, des types de données, des types énumérés. C'est donc est un réseau statique de classes et d'associations. En partant des classes et des associations trouvées précédemment, il faut construire un schéma sous forme de représentation graphique dans lequel les classes seront représentées par des rectangles et les associations par des traits pleins. Il faut ajouter à ce schéma des informations concernant les classes et leurs associations. Le diagramme de classes simplifié est donné dans la figure suivante (Figure 9)

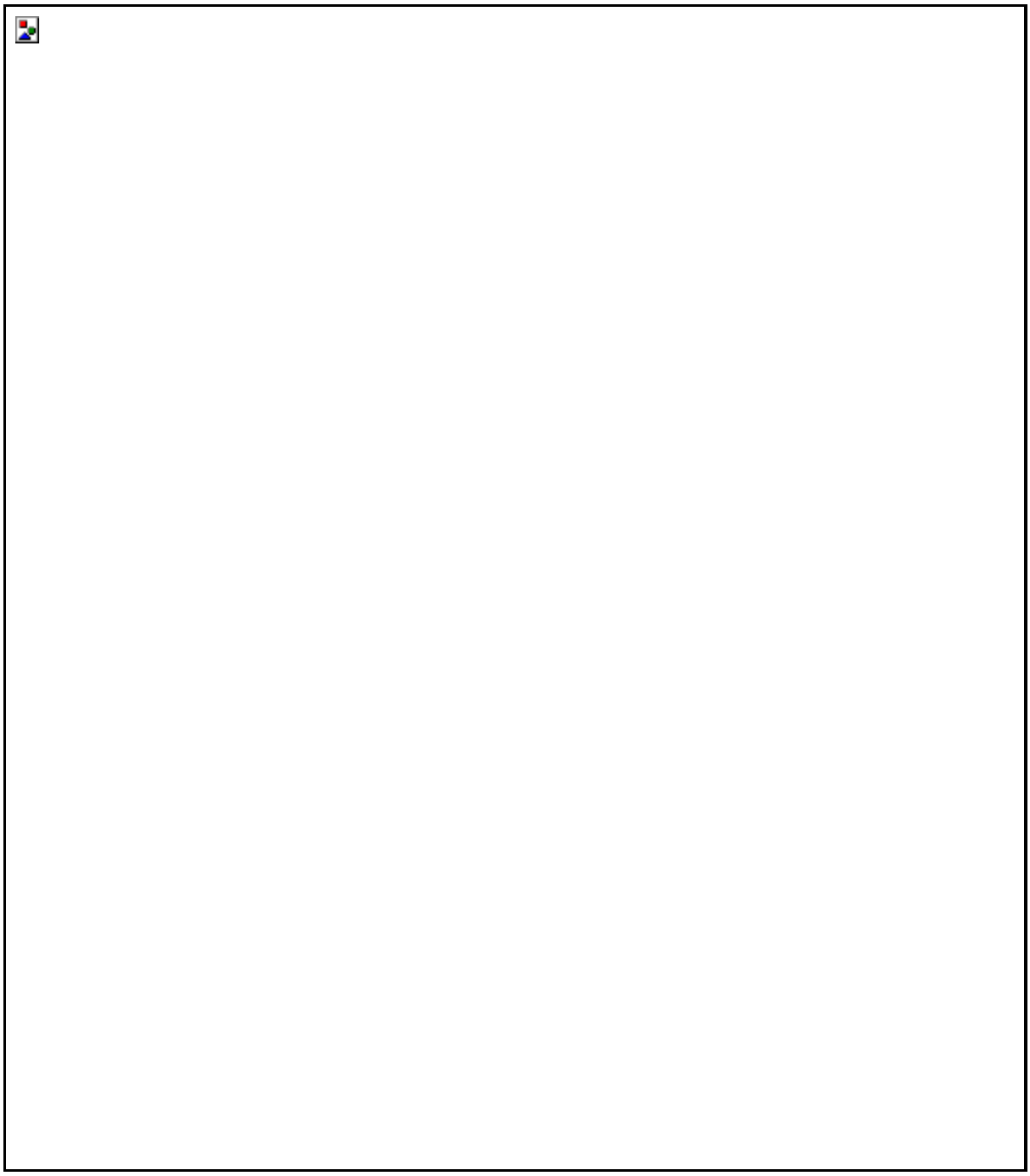

Figure 9 : Le diagramme de classes du système médiathèque

Notre diagramme de classes comprend également deux classes interface. La première, Empruntable, donne le comportement (ensemble d'opérations) des documents vis-à-vis de l'emprunt (le tarif et la durée de prêt en fonction du type de document) ; la seconde, Emprunteur, donne le comportement des clients vis à vis de l'emprunt d'un type de document (le nombre maximal de documents en fonction du type de client et de modificateurs du comportement dit « nominal » du document en ce qui concerne la durée et le tarif).

---

7. Les Diagrammes d'états de transitions

À partir du diagramme de classes établi dans un premier temps, il s'avère nécessaire de construire un diagramme de transitions d'états (DTE) pour certaines classes afin de montrer leurs utilisations. Ce diagramme permet de compléter la liste des opérations d'une classe (événements extérieurs et traitements internes). Les états initiaux et terminaux sont distingués. Il ne faut conserver que les états ayant une certaine stabilité dans le temps ou bien les états pendant lesquels de nombreuses et/ou importantes actions sont effectuées. Les états de type en train de, en cours de, etc. sont très intéressants ; par exemple pour un document, l'état en retard est intéressant puisque le système évolue suite à la survenue d'un événement. Les transitions d'états font suite à des évènements internes ou externes ; dans le premier cas, c'est une action interne à l'objet qui déclenche l'évènement. Les actions associées aux évènements sont supposées atomiques. Les diagrammes d'états transitions de la fiche d'emprunt, documents et clients sont présentés respectivement dans les figures suivantes (Figure 10, 11,12)

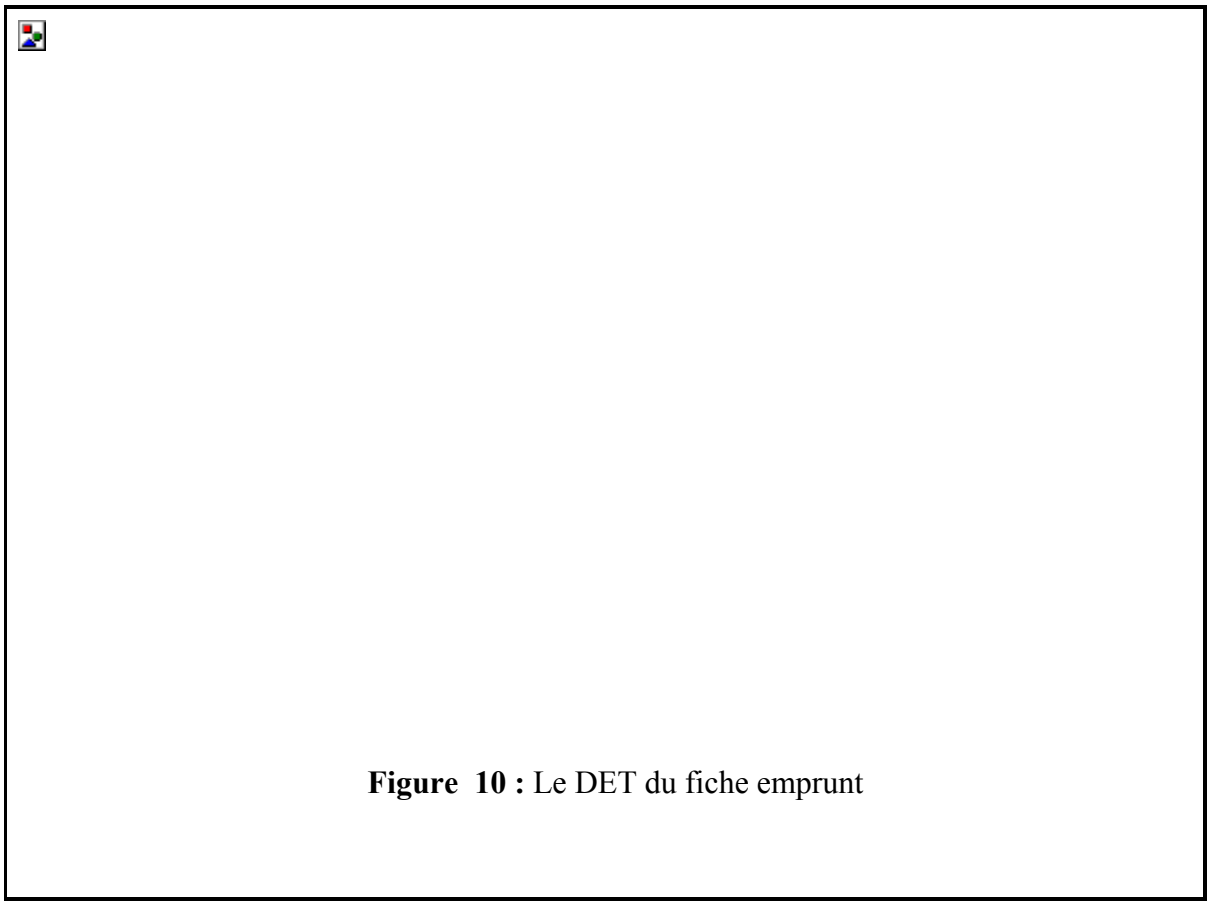

Figure 10 : Le DET du fiche emprunt

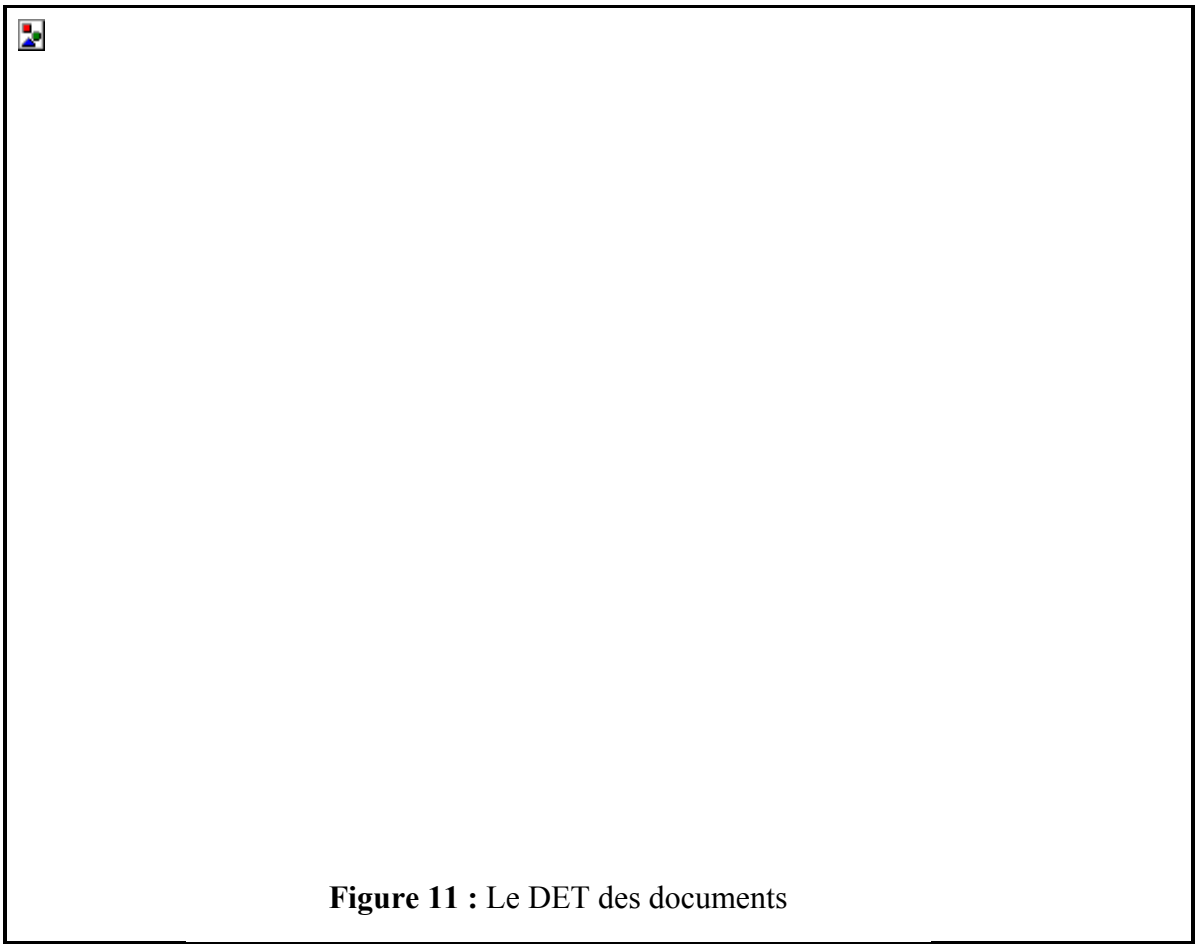

Figure 11 : Le DET des documents

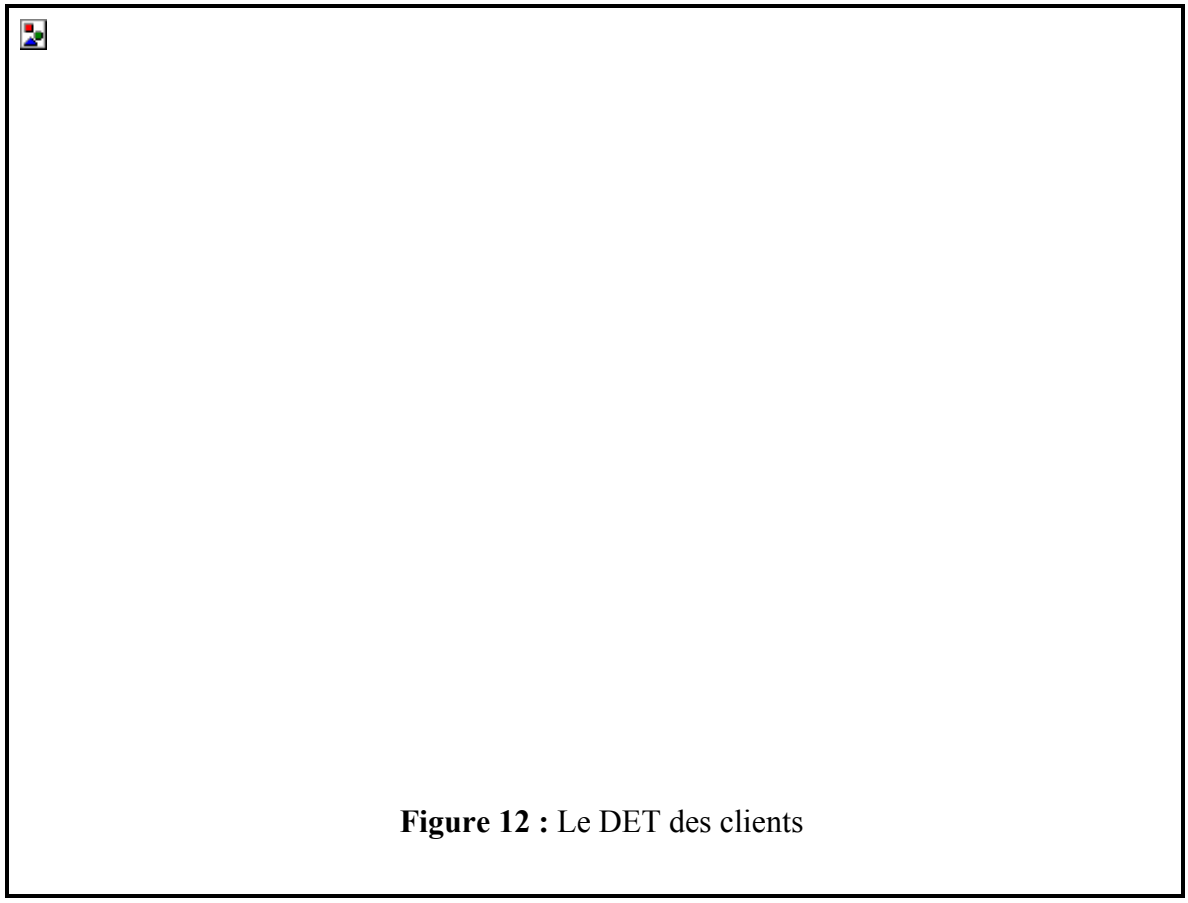

Figure 12 : Le DET des clients

Conclusion

L'approche présentée dans cet article a pour intérêt de modéliser selon la norme UML un système médiathèque. Cette approche nous a permis à la fois d'utiliser la norme universelle et de profiter de la multitude des Ateliers de génie logiciel d'UML afin de créer une application purement orientée objets. Cette manière de faire rend possible l'utilisation de la majorité des langages orientés objets que l'on désire. Dans notre cas on a utilisé l'outil rational rose et on a généré du code avec le langage Java qui représente un certain nombre d'avantages tels que la simplicité, l'orientation objet, la portabilité et la sécurité.
Le travail présenté dans cet article fait partie d'un projet dont l'objectif est de définir une méthodologie de développement basé sur l'UML et intégrant les composants définis dans la norme UML 2.0. Parmi les travaux qui restent à entamer, nous pouvons citer :
- la définition de la notion de composant pour une médiathèque ;
- l'élaboration d'une base de patrons de conception pour réaliser et implémenter ce système
- l'utilisation d'autres AGL UML pour générer d'autres langages
- l'utilisation d'un middleware pour créer des applications client/serveur avec des langages différents

Bibliographie

[1] : UML 2.0 superstructure final adopted specification. Object Management Group, document ptc/03-08-02, août 2003.
[2] : UML 2.0, OCL 2nd revised submission. Object Management Group, document ad/03-01-07, janvier 2003.
[3] : Meta-Object Facility MOF, version 1.4. Object Management Group, document formal/2002-04-03, avril 2002.
[4] : Unified Modeling Language 2.0, http://www.OMG.org/uml
[5] : CORBA Component Model, version 3.0. Object Management Group, document formal/2002-06-65, juin 2002.
[6] : U2 Partners, UML, version 2.0, http://www.U2-Partners.org
[7] : MDA Guide, version 1.0, version 2.0, document OMG : http://www.OMG.org/mda
[8] A.Hair, U_VBOOM: Unified Analysis and Design Process Based on the Viewpoint Concept. ICEIS (3) 2004: 217-224
[9] :www.mediatheque.cg68.fr/3reso/3page_informatique/La%20proc%E9dure%20d'informatisation.pdf
[10] : N. Lopez, J. Migueis, E. Pichon , Intégrer UML dans vos projets par ISBN 2-212-08952-X Editions Eyrolles, 2002.
[11] : G. Booch, J. Rumbaugh, I. Jacobson, The UML reference manual, ISBN 0-201-30998-X Addison-Wesley, 2003.
[12] : G. Booch, J. Rumbaugh, I. Jacobson, The Unified Software Development Process ISBN 0-201-57169-2 Addison-Wesley,2002.
[13] : N. Tawbi et H. Yahyaoui, Génie Logiciel IFT-16859 Partie 2, Avril 1999.
[14] : P.Leblanc, Implementation of the UML Testing Profile and Production of Executable Test Cases , White Paper Telelogic, www.telelogic.com , janv. 2004.
[15] B.sbihi, The integration of the points of view notion in UML. *Act The IADIS Applied Computing 2004 conference*, Lisbon, Portugal 2004.
[16] B.sbihi, La modélisation par les réseaux de pétri du mécanisme de filtrage UML; WONTIC'05, Workshop sur les Technologies de l'Information et de la communication, Kénitra, Maroc, 24- 25 juin 2005