

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MÉTADONNÉES ET PROCESSUS POUR L'ARCHIVAGE DE DONNÉES  
MÉDIATIQUES

MÉMOIRE  
PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN INFORMATIQUE

PAR  
MAMA AMAR

MARS 2012

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

## REMERCIEMENTS

Je souhaite exprimer ma profonde gratitude à ma directrice de recherche Brigitte Kerhervé et mon co-directeur de recherche Olivier Gerbé pour m'avoir soutenu et encadré tout au long cette maîtrise. Merci à vous pour la patience, la disponibilité, le soutien et les conseils pour la réussite de ce projet de recherche.

Je tiens aussi à remercier M. Claude Coulombe ainsi que toute l'équipe de la Maison des Technologies de formation et d'apprentissage Roland-Giguère (MATI) pour m'avoir accueilli dans leur équipe.

Je remercie très sincèrement mes parents pour tout le sacrifice consenti. Votre soutien restera inestimable. Je remercie aussi toute ma famille pour le soutien et les encouragements.

Également, j'aimerais remercier tous les professeurs et l'ensemble du personnel de l'Université du Québec à Montréal (UQAM) pour leur soutien tout au long de la maîtrise.

En fin, je tiens également à remercier Nancy Campagna, Guillaume Guay et tous ceux qui de près ou de loin m'ont apporté leur soutien.

## TABLE DES MATIÈRES

|  |      |
|--|------|
| LISTE DES FIGURES .....                    | ix   |
| LISTE DES TABLEAUX.....                    | xi   |
| LISTE DES ABRÉVIATIONS.....                | xiii |
| RÉSUMÉ .....                               | xv   |
| CHAPITRE I                                 |      |
| INTRODUCTION .....                         | 1    |
| 1.1 PRÉSENTATION DU PROJET.....            | 1    |
| 1.2 CONTEXTE ET MOTIVATION.....            | 2    |
| 1.3 PROBLÉMATIQUE.....                     | 3    |
| 1.3.1 L'augmentation des OM.....           | 3    |
| 1.3.2 Caractéristiques des OM.....         | 3    |
| 1.3.3 Problématiques des OM .....          | 4    |
| 1.4 OBJECTIFS DU PROJET.....               | 4    |
| 1.4.1 Présentation des objectifs.....      | 4    |
| 1.4.2 Proposition et contribution .....    | 5    |
| 1.4.3 Pertinence du sujet.....             | 5    |
| 1.5 SYNOPSIS DU MÉMOIRE .....              | 6    |
| CHAPITRE II                                |      |
| ÉTAT DE L'ART .....                        | 7    |
| 2.1 INTRODUCTION .....                     | 7    |
| 2.2 THÈMES ABORDÉS.....                    | 8    |
| 2.2.1 Les métadonnées multimédia .....     | 8    |
| 2.2.2 Gestion et archivage d'OM.....       | 12   |
| 2.2.3 Outils et technologies existant..... | 22   |
| 2.2.4 Discussion.....                      | 24   |
| 2.3 CONCLUSION .....                       | 24   |
| CHAPITRE III                               |      |

|  |    |
|--|----|
| SYSTÈME D'ARCHIVAGE D'OBJETS MÉDIATIQUES (SARDOM).....                       | 25 |
| 3.1 PRÉSENTATION DE NOTRE APPROCHE .....                                     | 25 |
| 3.1.1 Gestion des métadonnées.....   | 26 |
| 3.1.2 Définition d'un processus d'archivage d'un objet médiatique .....      | 27 |
| 3.1.3 Gestion des utilisateurs .....   | 29 |
| 3.1.4 Fonctionnalités du prototype .....                                     | 30 |
| 3.2 PRÉSENTATION DES CAS D'UTILISATION .....                                 | 30 |
| 3.2.1 Diagramme de cas d'utilisation « Initialiser le système » .....        | 33 |
| 3.2.2 Diagramme de cas d'utilisation « Ajouter type de métadonnées ».....    | 34 |
| 3.2.3 Diagramme de cas d'utilisation « Stocker un OM ».....                  | 36 |
| 3.2.4 Diagramme de cas d'utilisation « Annoter un objet médiatique ».....    | 38 |
| 3.2.5 Diagramme de cas d'utilisation « Rechercher un objet médiatique »..... | 40 |
| 3.2.6 Diagramme de cas d'utilisation « Archiver un objet médiatique ».....   | 42 |
| 3.3 PROPOSITION D'UN MODÈLE DE DONNÉES .....                                 | 44 |
| 3.3.1 Diagramme de classe de notre modèle.....                               | 44 |
| 3.3.2 Représentation arborescente de notre modèle.....                       | 45 |
| 3.4 CONCEPTION DU PROTOTYPE .....  | 46 |
| 3.4.1 Diagramme de séquence détaillée « Stocker un OM ».....                 | 47 |
| 3.4.2 Diagramme de séquence détaillé « Annoter un OM ».....                  | 48 |
| 3.4.3 Diagramme de séquence détaillée « Rechercher un OM » .....             | 49 |
| 3.4.4 Diagramme de séquence détaillé « Archiver un objet médiatique ».....   | 50 |
| 3.4.5 Diagramme de classe de conception .....                                | 51 |
| 3.5 ARCHITECTURE CIBLE DU PROTOTYPE.....                                     | 52 |
| 3.5.1 Couche présentation.....   | 53 |
| 3.5.2 Module de gestion des métadonnées .....                                | 53 |
| 3.5.3 Module de gestion des métadonnées .....                                | 54 |
| 3.5.4 Module de gestion des utilisateurs.....                                | 54 |
| 3.5.5 Le référentiel de données.....   | 54 |
| 3.6 CONCLUSION.....  | 55 |
| CHAPITRE IV  |    |
| IMPLÉMENTATION ET EXPÉRIMENTATION.....                                       | 57 |
| 4.1 INTRODUCTION .....   | 57 |

|       |   |    |
|-------|---|----|
| 4.2   | OUTILS ET TECHNOLOGIES UTILISÉS.....                          | 57 |
| 4.2.1 | Java Content Repository.....                                  | 57 |
| 4.2.2 | Apache-Jackrabbit.....  | 59 |
| 4.2.3 | Apache-Sling.....   | 60 |
| 4.2.4 | Apache-Tika.....  | 61 |
| 4.2.5 | Choix des outils.....   | 63 |
| 4.3   | IMPLÉMENTATION.....   | 64 |
| 4.3.1 | Environnement de développement.....                           | 64 |
| 4.3.2 | Architecture applicative.....                                 | 64 |
| 4.4   | EXPÉRIMENTATION.....  | 66 |
| 4.4.1 | Présentation du cadre d'étude.....                            | 66 |
| 4.4.2 | Scénario de l'expérimentation.....                            | 67 |
| 4.4.3 | Exécution détaillée du scénario.....                          | 68 |
| 4.5   | CONCLUSION.....   | 72 |
|       | CHAPITRE V  |    |
|       | CONCLUSION.....   | 73 |
| 5.1   | BILAN.....  | 74 |
| 5.2   | PERSPECTIVES.....   | 75 |
|       | BIBLIOGRAPHIE.....  | 77 |
|       | APPENDICE A   |    |
|       | REPRÉSENTATION DU MODÈLE EN XML.....                          | 81 |
|       | APPENDICE B   |    |
|       | REPRÉSENTATION DU STANDARD DUBLIN CORE DANS NOTRE MODÈLE..... | 83 |
|       | APPENDICE C   |    |
|       | PROBLÈMES DES MÉTADONNÉES : MULTIMÉDIA VS TEXTE.....          | 85 |



## LISTE DES FIGURES

| Figure | page   |
|--------|--|
| 3.1    | Diagramme de cas d'utilisation « Administrative » du système ..... 31  |
| 3.2    | Diagramme de cas d'utilisation « Utilisateur » du système ..... 32     |
| 3.3    | Diagramme de séquence « Ajouter un type de métadonnées » ..... 35      |
| 3.4    | Écran du cas d'utilisation « Ajouter un type de métadonnées » ..... 35 |
| 3.5    | Diagramme de séquence « Stocker un OM » ..... 37                       |
| 3.6    | Écran du cas d'utilisation « Stocker un OM » ..... 37                  |
| 3.7    | Diagramme de séquence « Annoter un OM » ..... 39                       |
| 3.8    | Écran du cas d'utilisation « Annoter un OM » ..... 39                  |
| 3.9    | Diagramme de séquence « Rechercher un OM » ..... 41                    |
| 3.10   | Écran du cas d'utilisation « Rechercher un OM » ..... 41               |
| 3.11   | Diagramme de séquence « Archiver un objet médiatique » ..... 43        |
| 3.12   | Écran du cas d'utilisation « Archiver d'un OM » ..... 43               |
| 3.13   | Diagramme de classe de notre modèle ..... 44                           |
| 3.14   | Représentation arborescente des types modèle de données ..... 45       |
| 3.15   | Représentation du modèle de données ..... 46                           |
| 3.16   | Diagramme de séquence détaillée « Stocker un OM » ..... 47             |
| 3.17   | Diagramme de séquence détaillée « Annoter un OM » ..... 48             |
| 3.18   | Diagramme de séquence détaillée « Rechercher un OM » ..... 49          |
| 3.19   | Diagramme de séquence « Archivage d'un OM » ..... 50                   |
| 3.20   | Diagramme de classe de conception ..... 51                             |
| 3.21   | Architecture cible du prototype ..... 52                               |
| 4.1    | Architecture applicative du prototype ..... 65                         |
| 4.2    | Modèle de données du cadre d'étude ..... 66                            |
| 4.3    | Répertoire des albums photos à archiver ..... 67                       |
| 4.4    | Stockage d'une image dans le référentiel ..... 68                      |
| 4.5    | Métadonnées après stockage de l'objet ..... 68                         |

|      |   |    |
|------|---|----|
| 4.6  | Annotation de la photo .....                    | 69 |
| 4.7  | Métadonnées après ajout .....                   | 69 |
| 4.8  | Formulaire de recherche d'une image .....       | 70 |
| 4.9  | Affichage des résultats de recherche .....      | 70 |
| 4.10 | Archivage d'une image dans le référentiel ..... | 71 |

## LISTE DES TABLEAUX

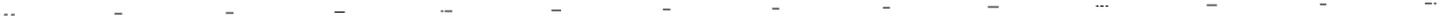
| Tableau  | page |
|--|------|
| 2.1 Eléments de métadonnées du standard Dublin Core..... | 10   |
| 3.1 Catégories d'utilisateurs.....                       | 29   |
| 3.2 Cas d'utilisation de notre système.....              | 32   |



## LISTE DES ABRÉVIATIONS

|        |  |
|--------|--|
| API    | Application Programming Interface                                      |
| CD     | Compact Disc   |
| CREAM  | CREATING Metadata for the Semantic Web                                 |
| DVD    | Digital Versatile Disc   |
| GPL    | General Public Licence   |
| ISO    | Organisation internationale de normalisation                           |
| JCR    | Java Content Repository  |
| JSR    | Java Specification Requests  |
| LINDO  | Large scale distributed INDEXation of Multimedia Objects               |
| MATI   | Maison des Technologies de formation et d'apprentissage Roland-Giguère |
| MPEG   | Moving Picture Experts Group   |
| OM     | Objet(s) médiatique(s)   |
| OSGI   | Open Services Gateway Initiative                                       |
| OWL    | Web Ontology Language  |
| RDF    | Resource Description Framework   |
| REST   | Representational State Transfer  |
| SARDOM | Système d'ARCHivage d'Objets Médiatiques                               |
| SGBDR  | Système de Gestion de Base de Données Relationnel                      |
| SQL    | Structured Query Language  |
| UML    | Unified Modeling Language  |

|       |                                 |
|-------|---------------------------------|
| UQAM  | Université du Québec à Montréal |
| URI   | Uniform Resource Identifiers    |
| URL   | Uniform Resource Locator        |
| W3C   | World Wide Web Consortium       |
| XML   | eXtensible Markup Language      |
| XPath | XML Path Language               |



## RÉSUMÉ

De nos jours, les données multimédia sont de plus en plus utilisées dans de nombreux secteurs de l'informatique. Cela a provoqué l'expansion de grandes quantités de données multimédia. Ainsi, vu le volume de données et le manque de structure d'OM, il devient de plus en plus difficile de les organiser et de les stocker. En effet, l'exploitation de ressources multimédia nécessite des méthodes et des outils informatiques efficaces pour faciliter leur accès, leur utilisation, leur organisation et leur partage. Les métadonnées multimédia peuvent jouer un rôle essentiel à l'élaboration de ces outils. En effet, les métadonnées peuvent rendre les objets multimédia plus explicites et plus faciles à manipuler par les applications informatiques.

Ainsi, un bon nombre d'universitaires et de professionnels s'investissent dans l'élaboration de standards et d'outils pour la gestion et l'archivage d'objets multimédia. Cependant, vu la diversité des standards, il existe souvent une problématique d'interopérabilité entre les standards. De plus, les standards sont définis en général pour des domaines spécifiques.

Dans ce mémoire, nous définissons une approche basée sur les métadonnées et les processus pour une gestion optimale et l'archivage des objets multimédia. Aussi, nous définissons un prototype basé sur cette approche. Les résultats de notre expérimentation ont montré l'importance d'un processus pour la gestion d'OM et de métadonnées pour faciliter la recherche, l'exploitation et l'archivage d'OM. Nos résultats ont aussi montré la flexibilité du modèle de données proposé. Ce dernier est extensible et est adaptable au contexte d'utilisation. À chaque étape du processus de gestion d'un OM, les résultats font ressortir l'importance des métadonnées.

*Mots clés : MultiMedia, Métadonnées, Annotations, Processus, Archivage, Modèle de données*



## CHAPITRE I

### INTRODUCTION

#### 1.1 PRÉSENTATION DU PROJET

De nos jours, la disponibilité d'outils orientés multimédia et l'augmentation de la capacité des systèmes de stockage ont favorisé la multiplication de la production de documents médiatiques : images, vidéos, objets sonores et textes. En effet, les outils pour créer, stocker et partager des OM sont de plus en plus accessibles. Par conséquent les utilisateurs peuvent facilement produire du contenu médiatique. C'est ainsi qu'avec l'apparition de sites web de partage de vidéos tel que YouTube<sup>1</sup>, Footytube<sup>2</sup>, de sites de réseautage social comme Facebook<sup>3</sup> et des sites web de partage de photos comme Flickr<sup>4</sup>, les documents médiatiques connaissent un développement remarquable. De surcroît, les OM sont maintenant utilisés dans plusieurs domaines d'application comme par exemple les compagnies de cinéma, les programmes des télévisions, les vidéos de sport, les laboratoires de médecine, etc.

Cette augmentation exponentielle du volume de données médiatiques produites engendre des problèmes pour les exploiter efficacement. En effet, avec l'expansion de ces contenus médiatiques, l'archivage et l'accès aux OM sont de plus en plus difficiles. Ainsi, faire une recherche pertinente dans un référentiel d'OM ou à travers Internet nécessite la mise en œuvre d'approches spécifiques aux particularités de ces objets. Il existe plusieurs outils pour la gestion et la recherche d'OM, cependant ces outils sont souvent basés sur des standards différents. Se pose ainsi le problème de l'interopérabilité entre les applications et

---

<sup>1</sup> <http://www.youtube.com>

<sup>2</sup> <http://www.footytube.com>

<sup>3</sup> <http://www.facebook.com>

<sup>4</sup> <http://www.flickr.com>

systèmes gérant des OM. De plus avec les caractéristiques de ces objets : taille, volume, peu de structure, il devient aussi difficile de les stocker et de les archiver.

Ainsi, pour une exploitation optimale de ces OM, il est essentiel d'avoir une vue d'ensemble sur les différentes étapes de leur cycle de vie, c'est-à-dire de leur création à leur exploitation en passant par le stockage. Il devient alors nécessaire d'avoir des outils et techniques permettant de suivre leur processus de création, de les archiver, de les stocker, et de pouvoir y accéder et les manipuler.

Dans notre projet de maîtrise, nous nous intéressons à la gestion d'OM et nous proposons une approche qui permet l'archivage des OM et leur exploitation de façon optimale. Cette approche est basée sur les métadonnées associées à ces objets pour faciliter l'organisation, le stockage, l'archivage et la recherche de ces objets. Nous présentons aussi l'application qui nous a servi de cadre d'application pour notre approche. Cette approche est basée sur des outils existant et elle apporte des solutions pour résoudre les problèmes ou manquements de ces outils.

## 1.2 CONTEXTE ET MOTIVATION

De nos jours, avec la forte présence des OM dans les activités de l'homme, le volume de données médiatiques connaît une croissance exponentielle. Ainsi, il devient de plus en plus difficile de les archiver et de les récupérer en cas de besoin. De surcroît, les OM présentent de nombreux avantages par rapport aux autres objets comme les documents texte. Ils ont des contenus plus riches et les outils de production de multimédia ont désormais des coûts qui les rendent accessibles.

Par conséquent, il s'avère nécessaire de disposer d'outils performants pour le stockage, la création, la mise à jour, la suppression, le partage et l'archivage de ces objets. Cependant la plupart des outils qui existent n'offrent que les opérations de base pour la gestion des OM. En outre, pour assurer une bonne conservation et un archivage des OM, nous avons besoin de connaître certaines informations sur l'origine ou encore sur les caractéristiques physiques de ces objets. Les métadonnées peuvent aussi être utilisées pour décrire et organiser la structure du contenu médiatique [Smith, J.R. et al., 2006].

### 1.3 PROBLÉMATIQUE

Ces dernières années, le domaine des technologies de l'information est marqué par la prolifération d'outils de production de documents médiatiques. Cette prolifération a pour conséquence la production d'un grand nombre d'OM par les professionnels ou par de simples utilisateurs. Le problème qui se pose alors est de pouvoir archiver ces OM et de pouvoir les retrouver facilement. La plupart des problèmes liés aux documents médiatiques sont : l'augmentation exponentielle du volume de données, les caractéristiques de ces objets ainsi que l'interopérabilité des standards qui existent dans ce domaine.

#### 1.3.1 L'augmentation des OM

Le volume de données médiatiques produit dans le monde croit de façon exponentielle : Par exemple le volume d'images ou de vidéos stockées sur YouTube ou Picassa. Par exemple : D'après [YOUTUBE], en Mai 2009, 20 heures de vidéo sont mises en ligne chaque minute sur YouTube alors qu'en 2010, 24 heures de vidéo sont mises en ligne chaque minute. La plupart des informations diffusées sur Internet le sont sous format multimédia (Images, vidéo ou fichier audio). De ce fait les exigences deviennent de plus en plus rigoureuses en ce qui concerne la recherche, la gestion et l'archivage des OM.

#### 1.3.2 Caractéristiques des OM

Les OM sont caractérisés par un manque de structure ainsi que par leur très grande taille rendant leur exploitation plus complexe. Par exemple, une vidéo de 30 seconde peut avoir une taille de *26,214,400 octets*. Par conséquent, les OM nécessitent des applications performantes pour leur stockage, leur manipulation, leur archivage et pour faciliter leur recherche. Ces applications doivent ainsi posséder des capacités de stockages élevées pour répondre aux exigences d'utilisation de ces objets.

### 1.3.3 Problématiques des OM

Dans le domaine multimédia, la plupart des standards de représentation des OM ne sont pas interopérables. En effet, la plupart de ces standards sont basés sur une approche fermée, où les modèles de représentation sont limités et ne peuvent facilement être réutilisés dans d'autres applications. De plus, la multiplicité des domaines qui utilisent les OM fait que la plupart des standards sont destinés à un domaine spécifique et ne peuvent être utilisés dans d'autres domaines. Dans [Nack, F. et al., 2005], les auteurs ont présenté des problèmes d'interopérabilité entre des standards de métadonnées pour deux communautés dans le domaine multimédia. Parmi ces communautés on peut citer W3C<sup>1</sup>, ISO<sup>2</sup> qui sont présentées dans [Ossenbruggen, J. v. et al., 2004].

## 1.4 OBJECTIFS DU PROJET

### 1.4.1 Présentation des objectifs

L'objectif principal de ce projet de maîtrise est de proposer une approche facilitant le processus de création et d'archivage d'OM et de développer une plateforme supportant cette approche. Cette approche doit tenir compte des standards existants pour permettre l'interopérabilité afin d'assurer le partage et l'échange d'OM entre plusieurs applications. Elle sera basée sur une étude de différents processus du cycle de vie des OM afin d'en extraire et d'en produire des métadonnées. Ces dernières sont ensuite utilisées lors de l'exploitation de l'OM. La plateforme développée pour valider l'approche proposée offre les fonctionnalités d'annotation afin de produire des métadonnées pour l'indexation, l'organisation et la recherche d'OM. En effet, les métadonnées constituent un moyen pratique pour la description des OM. Cette plateforme doit être facile d'utilisation car celle-ci est destinée à des utilisateurs non avérés et cela dénote une étude approfondie de l'interface homme-machine.

La solution que nous allons proposer va s'inspirer des technologies existantes et prendra en compte les limites de ces dernières.

---

<sup>1</sup> <http://www.w3.org/>

<sup>2</sup> <http://www.iso.org/iso/fr/home.htm>

#### 1.4.2 Proposition et contribution

Nous précisons d'abord que notre approche sera basée sur les recherches en cours dans le domaine multimédia. Ainsi, nous proposons une approche flexible basée sur un modèle de représentation des métadonnées indépendante du type d'OM utilisé. Ce qui a pour avantage d'avoir une approche interopérable avec d'autres systèmes. De plus notre modèle de métadonnées est hiérarchique et est extensible par l'utilisateur selon les types de métadonnées que l'on définit en fonction de notre domaine de recherche.

Nous proposons aussi une approche qui prend en compte l'ensemble du processus de création de l'OM pour récupérer les métadonnées qui existent éventuellement à chaque étape du processus. De surcroît, nous avons défini un modèle de métadonnées qui se base sur les standards existants et qui peut être adapté aux exigences spécifiques d'une application. Cela va permettre d'avoir un modèle flexible et qui s'adapte aux exigences du domaine. En outre, notre approche utilise un langage de requête standard (XPath) qui plus est, est interopérable. Enfin notre approche met l'accent aussi bien sur les descripteurs de bas niveau que sur les descripteurs sémantiques de haut niveau et intègre des algorithmes simples de recherche, d'annotation et d'archivage de contenu médiatique.

#### 1.4.3 Pertinence du sujet

L'information numérique est de plus en plus multimédia. Les documents médiatiques nécessitent beaucoup de gestion et de manipulation (stockage, création, mise à jour, destruction, partage, annotation, recherche, etc.). Pour assurer une utilisation effective de ces données médiatiques par une variété d'utilisateurs, ces objets doivent être facilement accessibles, exploitables et interopérables. Des lors, il s'avère nécessaire d'avoir des outils et techniques permettant l'archivage et l'exploitation de ces OM.

Cependant, la gestion de ces OM pose un certains nombre de problèmes de stockage, d'indexation et de recherche. Par exemple, il est souvent difficile de faire une recherche dans un référentiel d'OM en raison des limitations de description textuelle. De plus les OM sont aussi caractérisés par leurs sources de données différentes.

En outre avec le problème d'interopérabilité entre les standards qui existent dans ce domaine, il devient nécessaire d'avoir une approche ouverte, flexible et indépendante du domaine d'application et de l'environnement.

## 1.5 SYNOPSIS DU MÉMOIRE

Le mémoire est structuré comme suit : nous présentons d'abord l'état de l'art dans le cadre de l'archivage d'OM (§ Chapitre II). Ensuite dans le Chapitre III, nous passons à une présentation détaillée de notre approche avec une description du système implémenté basé sur cette approche que nous appelons SARDOM (Système d'ARchivage d'Objets Médiatiques). Puis nous décrivons l'implémentation de notre approche et l'expérimentation dans le cadre d'un projet de gestion d'archives de photos personnelles ainsi que dans le cadre du projet ePortfolio de la MATI. (§ Chapitre IV). Enfin nous concluons par la présentation de notre contribution et des perspectives de notre travail (§ Chapitre V).

---

-----

-----

## CHAPITRE II

### ÉTAT DE L'ART

#### 2.1 INTRODUCTION

Face à la croissance exponentielle de la production d'objets médiatiques, les différents acteurs de ce secteur sont confrontés à de nombreux problèmes en ce qui concerne le stockage, l'archivage, l'annotation et la recherche d'OM. Par rapport à la complexité et l'hétérogénéité des OM. Ainsi, beaucoup de chercheurs investissent leurs activités dans ce domaine. En effet, il existe maintenant plusieurs projets de recherche qui traitent de la gestion des OM et des métadonnées associées. Certains projets traitent les besoins et les problèmes liés à la gestion et l'exploitation des OM. D'autres sont orientés vers l'indexation des données médiatiques, les standards de données médiatiques et métadonnées multimédia et les problèmes d'interopérabilité entre ces standards. Tandis que d'autres se focalisent sur l'architecture et les processus des systèmes de gestion d'OM et de métadonnées.

Bien qu'il existe plusieurs approches pour la gestion, la recherche de données médiatiques et l'extraction de métadonnées de ces données, la plupart des approches ne permettent que des opérations de recherches et d'extraction de bas niveau du fait de la complexité des OM.

Dans ce chapitre, nous faisons d'abord une présentation des concepts de base du domaine multimédia en définissant les métadonnées ainsi que les standards de métadonnées multimédia. Ensuite nous donnons l'état de l'art au sujet de la gestion et de l'archivage d'OM.

## 2.2 THÈMES ABORDÉS

### 2.2.1 Les métadonnées multimédia

Les métadonnées sont des données qui servent à décrire d'autres données et peuvent être utilisées dans la gestion des données qu'elles décrivent. Elles peuvent être créées automatiquement ou manuellement au besoin par le créateur de la ressource. Les métadonnées permettent ainsi d'améliorer la description de l'OM. Elles peuvent être stockées durant le processus de création de l'OM et être accessibles par les outils de traitement de l'OM. Elles peuvent être modifiables ou non durant le cycle de vie de l'objet qu'elles décrivent.

Une annotation est le mécanisme par lequel le producteur ou l'utilisateur ajoute des informations à un document multimédia. Dans une base de données multimédia, une annotation permet de regrouper, d'organiser ou d'extraire des images, ou d'autres objets.

#### 2.2.1.1 Les types de métadonnées multimédia

Suivant leur utilisation, les métadonnées peuvent être classifiées en plusieurs types. Ainsi il existe des métadonnées descriptives, de gestion, de conservation et des métadonnées techniques. On peut trouver d'autres types de métadonnées selon les objectifs, par exemple, des métadonnées qui peuvent changer durant le cycle de vie de l'OM et d'autres qui ne peuvent changer. On peut aussi trouver des métadonnées qui sont externes à l'OM et d'autres qui sont internes à l'OM. Dans [Pereira, F. et al., 2008] Pereira, F. et al. présentent une liste de types de métadonnées : *les métadonnées sur le contenu, les métadonnées d'identification et de localisation de contenu, les métadonnées d'interaction avec l'utilisateur, les métadonnées de gestion de contenu, les métadonnées sur le contexte de l'utilisateur.*

### 2.2.1.2 Les standards de métadonnées multimédia

- MPEG-7

MPEG-7 est une norme ISO/IEC développée par le MPEG (Moving Picture Experts Group), après le développement des normes MPEG-1 (en 1992), MPEG-2 (en 1994) et MPEG-4 (la Version 1 en 1998 et la version 2 en 1999). Selon J. v. Ossenbruggen 2004, le but de MPEG-7 est de définir un standard pour la description de contenu audiovisuel dans le domaine multimédia.

Selon [Shih-Fu C. et al 2001], MPEG-7 est composé de plusieurs parties dont :

**Des systèmes** qui spécifient des fonctionnalités de niveau système, telles que la préparation des descriptions MPEG-7 pour le transport efficace / stockage, la synchronisation de contenu et de descriptions, et le développement de décodeurs conformes.

**Des descripteurs (Ds)** qui définissent la syntaxe et la sémantique des caractéristiques du contenu audiovisuel.

**Des schémas de description (DSs)** permettant la construction de descriptions complexes en spécifiant la structure et la sémantique des relations entre les constituants descripteurs (Ds) ou schémas de description (DSs).

**Un langage de définition de description (DDL)** qui est un langage normalisé permettant de définir de nouveaux descripteurs (Ds) et schémas de description (DS), ainsi que l'extension ou la modification de descripteurs (Ds) et de schémas de description (DS) existants.

- Dublin Core

Dublin Core comprend un ensemble de quinze éléments d'information qui peuvent être utilisés pour la description d'une large variété de ressources [Reveiu, A., et al., 2008]. Ces éléments sont :

**Tableau 2.1** Eléments de métadonnées du standard Dublin Core

| Élément               | Définition   |
|-----------------------|--|
| <b>Collaborateur</b>  | Une entité responsable d'une contribution à la ressource   |
| <b>Couverture</b>     | Le thème spatial ou temporel de la ressource, l'applicabilité de la ressource dans l'espace ou la juridiction dont dépend la ressource.                      |
| <b>Créateur</b>       | Une entité responsable au premier chef de l'élaboration de la ressource.   |
| <b>Date</b>           | Un point ou une période dans le temps associés à un événement dans le cycle de vie de la ressource.  |
| <b>Description</b>    | Une description peut comprendre, sans limitation : un résumé, une table des matières, une représentation graphique ou une explication libre de la ressource. |
| <b>Format</b>         | Le format de fichier, le support physique ou les dimensions de la ressource.   |
| <b>Identificateur</b> | Une référence univoque vers la ressource dans un contexte donné.   |
| <b>Langage</b>        | La langue de la ressource.   |
| <b>Éditeur</b>        | Une entité responsable de la mise à disposition de la ressource.   |
| <b>Relation</b>       | Une ressource liée.  |
| <b>Droits</b>         | Une information à propos des droits détenus dans et sur la ressource.  |
| <b>Source</b>         | Une ressource liée de laquelle dérive la ressource décrite.  |
| <b>Sujet</b>          | Le sujet de la ressource.  |
| <b>Titre</b>          | Un nom donné à la ressource.   |
| <b>Type</b>           | La nature ou le genre de la ressource.   |

Source : [Dublin Core, DCES] <http://dublincore.org/documents/dces/>

- MPEG-21

MPEG-21 est le nouveau standard développé par le MPEG (Moving Picture Experts Group). Selon [Adriana R. et al., 2008], le MPEG-21 a pour objectif de fournir un framework ouvert pour la distribution et l'utilisation de documents multimédia interopérables. Cette condition pour l'interopérabilité aboutit à une grande diversité de métadonnées basées sur XML, qui décrivent les données médiatiques sur des niveaux sémantiques ou syntaxiques, pour les rendre plus accessibles à l'utilisateur.

Selon [J. v. Ossenbruggen et al., 2004], les différents composants de MPEG-21 sont :

**Déclaration d'objets numériques** : Un schéma uniforme et interopérable permettant de déclarer de documents numériques;

**Identification et Description d'objets numériques**: un framework pour identifier et décrire toute entité quelle que soit sa nature, son type ou sa granularité;

**Manipulation et utilisation de contenu** qui fournit des interfaces et des protocoles permettant la création, la manipulation, la recherche, l'accès, le stockage, la livraison et la réutilisation de contenu à travers la chaîne de distribution de contenu et de consommation;

**Gestion et protection de la propriété intellectuelle** qui gère et protège constamment et de façon fiable les contenus à travers un grand nombre de réseaux et de dispositifs;

**Terminaux et réseaux** qui fournissent un accès interopérable et transparent au contenu à travers les réseaux et les terminaux;

**Représentation de contenu**: Elle détermine comment les ressources multimédia seront représentées;

**Rapports d'événement** : Représente les paramètres et les interfaces qui permettent aux utilisateurs de comprendre précisément les performances de tous les événements à signaler dans le framework.

### 2.2.2 Gestion et archivage d'OM

[Ossenbruggen, J. v. et al., 2004] présentent un ensemble de problématiques liées à la gestion et l'exploitation des OM et plus particulièrement les problèmes et les exigences liées à la description du contenu sémantique des OM. Pour ces chercheurs, la gestion des OM reste difficile malgré l'augmentation de la capacité des réseaux. C'est ainsi qu'il est difficile de chercher et de retrouver un OM sur Internet.

J. v. Ossenbruggen et al. ont d'abord fait une analyse des métadonnées dans le processus de production multimédia. Selon eux la production de contenu médiatique est un processus complexe, ainsi les métadonnées pourraient réduire la complexité de ce processus en rendant explicite l'information implicite dans le contenu de l'OM. Les auteurs ont identifié trois étapes dans le processus de création des OM: L'étape de pré-production où les idées principales de la production sont déterminées. La phase de production correspondant à la création de l'objet à travers un matériel. La phase de post-production correspondant à l'édition, à la présentation et à l'archivage de l'objet. Selon J. v. Ossenbruggen et al., ces étapes de production sont fortement liées et les outils de manipulation des données à chaque étape doivent interagir. Ainsi, l'intégration de la production et du stockage des métadonnées dans ce processus augmente considérablement sa complexité. C'est ainsi qu'ils ont introduit une étape supplémentaire dans le processus appelée *metaproduction* qui implique la restructuration, la représentation, le reclassement, la réorientation et la redistribution des OM.

J. v. Ossenbruggen et al. ont ensuite présenté deux approches de gestion des métadonnées dans le processus de production multimédia défini par les deux communautés dans ce domaine W3C et ISO.

Web Sémantique avec les standards Uniform Resource Identifiers (URI) and Unicode, XML, RDF, RDF Schema, OWL

MPEG framework avec les standards MPEG-4 XMT, MPEG-7, MPEG-21

Ces deux communautés ont défini des standards de gestion de métadonnées multimédia pour faciliter la gestion des objets multimédia. Cependant les approches de ces deux communautés présentent des problèmes d'interopérabilité entre eux [Nack, F. et al., 2005]. Les deux approches bien qu'utilisant le même langage de sérialisation XML, diffèrent largement dans leur façon d'utiliser XML pour décrire des contenus multimédias [J. v. Ossenbruggen 2005]. Ainsi J. v. Ossenbruggen et al., 2005 ont fait une comparaison des standards de chaque communauté et une analyse des différences et des similitudes entre ces standards.

Enfin J. v. Ossenbruggen et al. ont fait une analyse sur les problèmes liés aux métadonnées par une comparaison entre texte et multimédia. Selon eux, les métadonnées sont primordiales pour certaines applications multimédia. Mais la gestion de ces dernières s'accompagne de plusieurs problèmes dont les problèmes de coût, de subjectivité, de restriction, de longévité, de confidentialité et de standardisation. Les détails sur ces problèmes sont résumés à l'Annexe C [Ossenbruggen, J. v. et al., 2004].

F. Pereira et al. [Pereira, F. et al., 2008] ont fait une étude sur les standards de métadonnées multimédia ainsi que les besoins futurs et les défis de ces standards. Selon eux, l'interopérabilité entre les données et leurs métadonnées associées est nécessaire pour une gestion efficace et l'échange d'information médiatique. Pour résoudre ce problème, il faut une standardisation.

Ils ont également passé en revue les standards multimédias existants liés à la recherche d'information et la prestation adaptative de contenu multimédia, en insistant sur la nécessité de telles normes. Ils ont ensuite démontré comment ces standards peuvent faciliter le développement d'outils et d'applications pour la gestion et l'exploitation de contenus médiatiques. Selon eux, l'interopérabilité peut prendre différentes significations à chaque étape du processus de production multimédia; elle représente une des exigences indispensables car elle permet aux outils et aux applications de pouvoir partager des fonctionnalités particulières telles que l'identification, l'extraction, la livraison, la gestion et la consommation. Ils mentionnent également que le plus grand défi pour les standards est

d'adapter les besoins du marché avec les capacités technologiques issues de la recherche; cela doit se faire dans les meilleurs délais.

L'objectif de F. Pereira et al. est de faire une présentation des standards de métadonnées existantes les plus pertinentes, de comprendre l'état de développement du domaine et de fournir une analyse stratégique des besoins futurs de standardisation. Ils proposent une analyse du défi du déploiement des standards de métadonnées et les améliorations dans le processus de normalisation afin d'aider au déploiement des standards de métadonnées. Pour faire la présentation des standards de métadonnées existants, ces auteurs les ont classés de la manière suivante :

- **Les standards pour la description de contenu** (MPEG-7, TV-Anytime, MPEG-21 Digital Item Declaration and Identification)
- **Les standards pour la description des droits et protections** (MPEG-21 Rights Expression Language, MPEG-21 Intellectual Property Management and Protection Components)
- **Les standards pour la description de contexte** (MPEG-21 Digital Item Adaptation, DLNA Media Format Profiles)
- **Les standards pour une nouvelle dimension.** (MPEG Multimedia Application Formats, Music Photo Player MAF Cases)

Ensuite ils font une analyse sur les besoins futurs et les défis pour les standards des métadonnées qui selon eux sont:

- **La facilité d'utilisation :** Les futures normes de métadonnées devraient facilement être utilisées dans différents environnements.
- **La transparence :** Les futures normes de métadonnées devraient être interopérables entre elles de façon transparente pour l'utilisateur.

- **Applications orientées vers les métadonnées :** Des applications génériques orientées vers les métadonnées pourraient faciliter la gestion des métadonnées multimédia et régler le problème d'interopérabilité.
- **Annotation Multimédia :** Selon F. Pereira et al., une application complète d'annotation devrait permettre l'intégration de contenu multimédia interopérable et l'encapsulation des médias et des métadonnées dans un format de fichier standardisé.
- **Standards pour les requêtes :** Ils mentionnent également qu'un format de requête interopérable est nécessaire pour faire en sorte que les serveurs et les référentiels multimédia traitent mieux et répondent de façon efficace aux requêtes des utilisateurs.

Selon F. Pereira et al., les métadonnées sont de plus en plus importantes pour une recherche efficace et efficiente, ainsi que pour le filtrage et la gestion de contenus médiatiques dans un référentiel. Les auteurs mettent en avant le fait que pour résoudre le problème d'interopérabilité entre les standards de métadonnées il faut une harmonisation, sur le plan sémantique et syntaxique des vocabulaires, entre les standards. Cette harmonisation doit se fonder sur une approche de développement modulaire facilitant l'extension d'applications spécifiques.

Dans [Hardman L. et al., 2008] Hardman L. et al. présentent la problématique de l'exploitation des OM annotés sémantiquement dans les systèmes multimédia actuels. Pour résoudre les problèmes de capture de métadonnées, de préservation et d'échange, Hardman L. et al. ont proposé une approche visant à améliorer l'interopérabilité de systèmes multimédia basée sur un modèle de processus canonique de la production médiatique. Leur approche est le résultat de nombreuses discussions avec des chercheurs de la communauté multimédia et. Cette approche est basée sur neuf processus canoniques de production d'OM qui sont :

- **Préméditation** qui précède la création de l'OM.
- **Création des ressources multimédia** ou l'objet média est créé. Cette phase nécessite l'utilisation d'un outil pour la production du média.
- **Annotation** permettant l'association d'informations à l'OM créé.
- **Paquetage** correspondant au regroupement de médias suivant leurs caractéristiques.
- **Requête** qui consiste à faire une sélection de certains OM parmi un ensemble d'objets dans un référentiel.
- **Construction du message** permettant à l'auteur de la création de l'OM de mettre en évidence le message qu'il veut véhiculer.
- **Organisation** consistant à créer une structure regroupant un ensemble d'objets suivant un ordre pouvant être présenté à un utilisateur.
- • **Publication**-qui permet de sélectionner les parties appropriées du document qui seront présentées à l'utilisateur. -
- **Distribution** qui consiste à la transmission de l'OM à un utilisateur final. Ce processus nécessite un dispositif permettant d'interagir avec celui-ci.

Enfin, Hardman L. et al. ont présenté des questions qui ont été soulevées lors des discussions de perspectives d'utilisation de l'approche de ces processus canoniques dans d'autres domaines. Selon Hardman L. et al., cette approche ainsi que les autres recherches sur l'annotation sémantique [Stamou, G et al., 2006] sont une étape vers des structures de données ouvertes et des composants logiciels sur le web pour décrire et partager des OM sémantiquement annotés entre différentes plateformes.

Pour S. Laborie et al [Laborie, S. et al., 2009], l'avantage d'utiliser un « résumé de métadonnées » au lieu d'un référentiel centralisé de métadonnées est d'éviter que les requêtes des utilisateurs soient envoyées à chaque serveur alors que seuls certains serveurs contiennent les informations désirées.

Le système qu'ils ont proposé est composé de plusieurs unités :

- **Un contenu multimédia** correspondant à un OM.
- **Une collection multimédia** regroupe plusieurs contenus multimédias. Chaque collection multimédia est stockée sur un serveur dédié pour obtenir des informations sur le site distant.
- **Un ensemble d'extracteurs** qui permet d'extraire un contenu de métadonnées à partir du contenu multimédia choisi.
- **Un contenu métadonnées** contient des informations sur les caractéristiques d'un contenu multimédia.
- **Une collection de métadonnées** regroupe toutes les métadonnées décrivant le contenu des objets de la collection multimédia.
- **Un « résumé de métadonnées »** qui est une version concise de collections de métadonnées. Ce résumé est utile pour identifier un ensemble de serveurs qui stockent des descriptions spécifiques.

Leur approche utilise un processus basé sur les trois étapes ci-après. D'abord le contenu multimédia est acquis et est stocké dans une **collection multimédia** puis indexé par un **ensemble d'extracteurs**. Ces derniers produisent ensuite **un contenu de métadonnées** associé. Tout contenu de métadonnées est ensuite stocké dans **une collection de métadonnées**. Enfin, un sous-ensemble de collections de métadonnées appelé « **résumé de métadonnées** » est alors transféré au serveur central. Pour exécuter une requête dans le système, toutes les collections de métadonnées sont réparties sur plusieurs serveurs et un

résumé de métadonnées sera produit sur le serveur central pour répondre aux requêtes générales ou localiser les serveurs qui peuvent stocker l'information désirée.

Proposée par S. Laborie et al. [Laborie, S. et al., 2009], l'architecture du système a été testée et les résultats de tests ont démontré que l'utilisation des résumés de métadonnées sur le serveur central améliore les performances de recherche. Ils ont ainsi démontré que des métadonnées distribuées dans plusieurs serveurs est plus efficace que leur centralisation. En outre, ils ont montré que faire une requête d'un résumé de métadonnées permet de localiser rapidement les serveurs qui peuvent stocker l'information désirée ce qui permet d'améliorer la performance d'un système de recherche.

Dans [Brut, M. et al., 2009], M. Brut et al. proposent un framework générique de métadonnées pour l'indexation et la gestion de contenus multimédia distribués qui a pour objectif de résoudre le problème d'interopérabilité qui découle de l'utilisation de différents standards. Pour cela, leur framework définit une représentation abstraite d'un document médiatique encapsulant les normes de métadonnées les plus courantes. M. Brut et al. ont d'abord fait une présentation des standards de documents médiatiques. En effet, les auteurs ont fait une étude sur les standards de documents multimédia et ont fait une présentation des standards pour chaque type de media.

Cependant, M. Brut et al. font remarquer que l'utilisation de standards est confrontée à de problèmes d'interopérabilité [Brut, M. et al., 2009]. Selon eux, ces problèmes sont causés par les aspects suivants :

- **Différentes structures de métadonnées**, car les métadonnées étant créées à l'aide d'outils différents qui ne sont souvent pas interopérables.
- **Imbrication des métadonnées**, car deux vocabulaires différents peuvent partager les mêmes éléments de métadonnées et chacun d'entre eux pourrait intégrer ses propres éléments spécifiques.
- **Sémantiques inhérentes de vocabulaire**, dans un vocabulaire d'un standard de métadonnées basé sur XML, les sémantiques inhérentes de l'information

codée ne sont spécifiées que dans un framework standard basé sur sa structure et sa terminologie.

- **Problème de synonymie des métadonnées**, les normes peuvent inclure des métadonnées qui sont synonymes. C'est à dire des métadonnées qui peuvent avoir des formes syntaxiques différentes faisant ainsi référence à la même information sémantique.

Pour prendre en compte ces aspects, M. Brut et al. ont développé un framework qui tient compte des vocabulaires de métadonnées multimédia les plus courants, qui les encapsule dans une structure globale. Le format de métadonnées proposé est basé sur une structure à deux niveaux car la description d'un document multimédia peut être effectuée de deux façons différentes : soit la représentation des **éléments généraux** sur l'OM par exemple l'auteur, la taille, la date de création etc. soit la description séparée de chaque élément de l'OM.

Le framework proposé est intégré dans le projet LINDO (Large scale distributed INDEXation of Multimedia Objects) dont l'objectif est « *de développer une architecture générique, dans laquelle le stockage de documents médiatiques est non seulement répartie, mais encore l'indexation est répartie sur différentes unités de stockage qui peuvent être hétérogènes et géographiquement éloignés* » [Brut, M. et al., 2009].

Le framework utilise le langage XML comme format de représentation du modèle de données proposé et XQuery comme langage de requête pour l'interrogation. Pour le stockage du modèle, ils ont choisit Oracle Berkeley DB XML. Ainsi, pour surmonter les problèmes d'interopérabilité dans les standards multimédia, M. Brut et al. ont proposé un framework générique d'indexation et de gestion de contenus multimédia distribués basé sur un modèle global qui encapsule les standards de métadonnées multimédia les plus utilisées. L'avantage de leur approche est que le modèle intègre plusieurs standards de métadonnées multimédia dans une structure globale qui facilite le processus de recherche et d'extraction [Brut, M. et al., 2009].

Dans [Ma, Y., Fang et al., 2008], Ma, Y et al. proposent une approche basée sur le schéma de métadonnées MPEG-7 pour la recherche d'OM. Ce framework fournit une méthode de recherche de contenu qui utilise le Schéma de Métadonnées MPEG 7. L'objectif de cette approche est de proposer un framework permettant la récupération de sémantiques de haut niveau. Cette approche propose trois modules pour le système multimédia : Un module d'analyse multimédia, un module de base de données multimédia et un module d'extraction.

Le module d'analyse multimédia permet la description de contenu médiatique. Cette description peut être automatique ou manuelle. Le module de base de données multimédia permet de stocker trois types de documents : les données médiatiques, les métadonnées basées sur MPEG-7 et les métadonnées multidimensionnelles de bas niveau. Le module d'extraction offre une interface pour les applications de recherches. Dans le module d'extraction, les requêtes sont structurées sous formats XML et le processus d'extraction est divisé en deux parties une à base de texte et l'autre à base de contenu.

Dans leur approche, bien que proposant une architecture modulaire, ils ne prennent pas en compte les différentes étapes du processus de création des OM et le standard utilisé c'est-à-dire le schéma MPEG-7 est très complexe pour certaines applications de gestion d'OM.

Dans [Zhou, T. H. et al., 2008], Zhou, T.H. et al. présentent un prototype de système de gestion de métadonnées multimédia permettant de partager l'intégration de métadonnées multimédia provenant de sources hétérogènes. Le système proposé permet la collecte, l'analyse et l'intégration de descriptions sémantiques de métadonnées multimédia pour certains types de services. Ce système est composé de quatre parties : un module d'acquisition, un module d'analyse, un module d'exploitation « mapping » et un module de stockage de métadonnées.

**Le module d'acquisition de métadonnées** permet de récupérer les métadonnées à partir du document multimédia.

**Le module d'analyse de métadonnées** permet de récupérer les informations de recherche des utilisateurs sous la forme d'un document XML.

**Le module d'exploitation « mapping » de métadonnées** permet de transformer, dans une description de sémantique de métadonnées uniforme, les métadonnées qui proviennent du module d'analyse.

**Le module de stockage de métadonnées** permet le stockage des segments de métadonnées dans le référentiel de métadonnées pour des opérations de recherche.

Zhou, T.H. et al. ont considéré leur approche dans deux contextes différents : Le premier est la gestion de collections de données médiatiques d'un utilisateur qui comprend l'archivage et l'extraction des éléments spécifiques dans des conditions sémantiques particulières; le second est la gestion de données médiatiques professionnelles au sein d'un réseau de partage de ressources médiatiques et d'informations sémantiques associées où les droits de propriété et d'autorisation sont importants. Par conséquent, T.H. Zhou et al. ont pris en compte la façon de stocker, d'organiser et de récupérer des ressources médiatiques distribuées; la façon de gérer les algorithmes de traitement d'information ; la façon d'ajouter des annotations sémantiques et la façon d'accéder, de protéger et de partager des informations.

Les résultats d'implémentation montrent que le système supporte des services de ressources distribuées pour un partage et une recherche à travers Internet bien que ces services soient basés sur des normes partagées. Le système permet aussi de réduire la complexité des différentes normes de description de données médiatiques.

### 2.2.3 Outils et technologies existant

Dans le domaine de la recherche ainsi que dans le domaine commercial, Il existe de nombreux outils de gestion de contenu médiatique qui offrent des méthodes d'extraction automatique de métadonnées. Nous allons dans cette partie, présenter quelques outils de gestion (modification, indexation, annotation recherche, gestion des accès) et d'archivage d'OM existants sur le marché.

#### 2.2.3.1 Présentation des outils

**M-OntoMat-Annotizer** est un outil développé à l'intérieur du projet d'aceMedia<sup>1</sup> qui permet l'annotation sémantique d'images et de vidéos pour l'analyse et l'extraction d'OM. C'est une extension du framework CREAM (CREating Metadata for the Semantic Web) utilisant son implémentation de référence OntoMat-Annotizer. M-OntoMat-Annotizer est distribué sous licence GNU LESSER GENERAL PUBLIC LICENSE<sup>2</sup>.

M-OntoMat-Annotizer est une application développée en Java et utilise l'API Java Media Framework (JMF) 2.1.1e pour la lecture des vidéos. Selon [Saathoff, C., et al., 2006], l'outil a été créé pour permettre aux fournisseurs de contenu d'annoter des descripteurs visuels sans expertise préalable dans des technologies du web sémantiques ou d'analyse multimédia. Toujours selon [Saathoff, C., et al., 2006], la première version de l'outil M-OntoMat-Annotizer est disponible comme logiciel sur le site web depuis mai 2005<sup>3</sup>.

**ELAN** est un outil logiciel pour la création manuelle d'annotations textuelles pour les fichiers audio et vidéo mis en œuvre par l'Institut de psycholinguistique Max Planck (MPI)<sup>4</sup> basée en Nimègue au Pays-Bas. ELAN est un outil développé avec le langage de programmation Java et est compatible avec plusieurs systèmes d'exploitation dont Windows, Mac OS X, Linux.

---

<sup>1</sup> <http://www.acemedia.org/>

<sup>2</sup> <http://www.gnu.org/licenses/lgpl.html>

<sup>3</sup> <http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html>

<sup>4</sup> <http://www.mpi.nl>

Cet outil utilise le langage XML (eXtensible Markup Language) pour le stockage d'OM. La première version d'ELAN<sup>1</sup> est sortie depuis Février 2002 [Brugman, H., et al., 2004], [Sloetjes, H. et al., 2008].

**M4NOTE** est un framework pour l'annotation de documents multimédia. Il est composé d'une interface multimodale développée en Java qui permet la capture de vidéo et les annotations [G. Rudinei, et al., 2004]. A la fin du processus de capture, un document XML est généré comme une composition des références de tous les medias capturés : vidéo, audio, images, etc. Selon G. Rudinei, et al., le modèle de données médiatiques est basé sur le standard MPEG-7 et fournit un support pour la représentation des informations contextuelles associées aux contenus médiatiques.

#### 2.2.3.2 Limites de ces outils

Nous avons précédemment présenté un ensemble d'outils qui utilisent les métadonnées pour le stockage et l'extraction des OM. Cependant, la plupart de ces outils ne sont pas interopérables avec d'autres applications car ils utilisent des standards spécifiques qui ne sont pas interopérables avec d'autres existants, par exemple MPEG-7. De plus, la plupart des outils de création de métadonnées multimédia ne sont pas basés sur des normes. Certaines approches utilisées par ces outils sont limitées à un type d'OM spécifique (comme la vidéo) et ne prennent pas en compte les autres types d'objets par exemple l'image. D'autres approches mettent l'accent sur les descripteurs de bas niveau. Ces systèmes aussi ne prennent pas en compte l'ensemble du processus de création d'OM et par conséquent ils omettent certaines informations qui sont utiles dans la vie de l'OM. Un autre problème lié à ces applications est que leur modèle de données ne sont pas extensibles et ne permettent de prendre en compte d'autres modèles existants.

---

<sup>1</sup> <http://www.lat-mpi.eu/tools/elan/elan-description>

#### 2.2.4 Discussion

Du fait de la complexité, du manque de structure et du volume croissant des OM, notre analyse révèle que la gestion et l'archivage de ces objets posent un certain nombre de contraintes. Cette analyse a montré bien qu'il existe des standards de métadonnées, ces standards sont souvent confrontés à des problèmes d'interopérabilité entre eux. Ainsi, nous en concluons que les métadonnées sont essentielles pour des applications de gestion d'OM et qu'il est nécessaire de considérer ceux-ci dans les différentes étapes de production de multimédia. Aussi, nous retenons qu'il est nécessaire d'avoir un standard permettant de garantir une interopérabilité entre les applications dans le domaine multimédia. De surcroît, nous pensons aussi qu'une bonne architecture et une bonne organisation des données et des métadonnées est nécessaire pour une bonne performance des applications lors des opérations de recherche et d'indexation des OM. Nous pensons aussi qu'il peut être utile de classifier les métadonnées par type suivant leurs fonctions pour garantir une bonne organisation.

Nous nous référons sur ces études afin de proposer une nouvelle approche permettant l'archivage d'OM et de concevoir un prototype qui sera basé sur cette nouvelle approche.

### 2.3 CONCLUSION

Dans ce chapitre, nous avons présenté l'état de l'art de la gestion des OM et des métadonnées associées. D'abord, nous avons fait une étude sur les métadonnées en présentant quelques types et standards de métadonnées existants. Ensuite nous avons présenté des travaux relatifs à la gestion et à l'exploitation des OM et des métadonnées. Nous avons étudié quelques outils et technologies existants en présentant leurs fonctionnalités ainsi que leurs limites. Dans le chapitre qui suit nous allons décrire un nouveau système d'archivage d'OM que nous proposons.

## CHAPITRE III

### SYSTÈME D'ARCHIVAGE D'OBJETS MÉDIATIQUES (SARDOM)

Dans ce chapitre, nous proposons le Système d'Archivage D'Objets Médiatiques appelé SARDOM. Ce système permet la gestion, l'archivage et la récupération efficace d'OM grâce aux métadonnées. D'abord, nous présentons le système proposé d'une manière générale à travers une description de notre approche ainsi que du processus d'archivage d'OM. Ensuite, nous donnerons les détails du système SARDOM à travers son modèle de données et son architecture système.

#### 3.1 PRÉSENTATION DE NOTRE APPROCHE

Notre approche se base sur un processus qui couvre l'ensemble des étapes du cycle de vie d'un OM, de la création à l'archivage, puis sur les métadonnées associées à ces objets. Tout au long du cycle de vie de l'OM, d'autres métadonnées peuvent être créées et reliées à l'objet. Ces métadonnées permettent d'avoir plus d'informations sur l'OM, en facilitant ainsi la recherche, la gestion et l'archivage. Cette approche bien que présentant des spécificités par rapport aux approches existantes reprend certaines de leurs fonctionnalités de base.

### 3.1.1 Gestion des métadonnées

#### 3.1.1.1 Types de métadonnées

Comme notre approche est basée sur les métadonnées, nous avons jugé nécessaire de les organiser et de les classer par types. Nous avons défini trois types de base qui nous serviront pour développer le prototype s'appuyant notre approche. Nous avons choisi ces différents types car nous pensons qu'ils s'adaptent à notre contexte d'utilisation. Cependant, notre modèle va permettre selon le contexte d'utilisation d'ajouter d'autres types de métadonnées. Nous avons séparé les métadonnées et les OM pour pouvoir les manipuler plus facilement. Ces types sont :

- **Les métadonnées descriptives** qui seront utilisées dans la recherche et l'identification d'un OM. Exemple : titre, auteur, lieu.
- **Les métadonnées techniques** qui seront utilisées pour afficher et consulter un objet particulier. Exemple : format de fichier, taille, dimension.
- **Les métadonnées administratives** qui seront utilisées pour la gestion et l'archivage d'un OM. Exemple : date archive, information sur les droits, profil archive, version et archive.

#### 3.1.1.2 Formats de métadonnées

Dans notre approche, nous avons choisi d'utiliser le format hiérarchique pour représenter les métadonnées, ce qui va nous permettre d'avoir un schéma dynamique plus facile à modifier et à étendre. Cela va permettre également d'en parcourir le contenu plus facilement. Ce format permet aussi d'assurer une interopérabilité avec les autres standards car les métadonnées pourront être stockées dans des fichiers XML.

### 3.1.2 Définition d'un processus d'archivage d'un objet médiatique

Dans cette partie, nous présentons le processus d'archivage d'un OM proposé dans ce document. Ce processus est composé de plusieurs étapes que nous décrivons en détails plus loin dans ce document. Nous présentons aussi les différents objets qui interviennent dans ces étapes et les interactions qui existent entre eux. Pour décrire les différentes étapes du processus, nous utilisons le langage de modélisation unifié UML<sup>1</sup>.

Les étapes suivantes illustrent notre approche:

D'abord nous avons les informations qui précèdent la création de l'OM. Ensuite ces informations sont enregistrées dans le référentiel de métadonnées avec une référence à l'OM correspondant. Création de l'OM a travers un appareil. Puis l'OM est enregistré dans le référentiel d'OM. Des annotations peuvent ensuite être ajoutées à l'OM par le créateur ou les utilisateurs. Les utilisateurs peuvent ensuite effectuer des recherches sur le référentiel en spécifiant des critères de recherche portant sur les informations associées aux objets. Enfin dans la phase d'archivage, les OM sont compressés avec les métadonnées associées et préservés à long terme pour être exploitables dans le futur.

A partir de cette illustration, nous listons l'ensemble des étapes possibles pour la gestion d'un OM.

#### 3.1.2.1 Phase de pré-production

Cette étape vient avant même la création de l'OM. C'est à cette étape que l'utilisateur va définir le contexte de création de l'OM.

#### 3.1.2.2 Phase de création

Cette étape correspond à la création de l'OM. Cette étape nécessite l'utilisation d'un outil de capture par exemple une caméra numérique. A cette étape, des métadonnées sur l'objet vont être créées. L'objet sera ensuite sauvegardé sur un support de stockage comme un disque dur, un CD, ou DVD ou autres supports pour une exploitation future.

---

<sup>1</sup> UML Unified Modeling Language (<http://www.uml.org/>)

**Remarque :** Ces étapes ne sont pas exécutées par notre système. Cependant, il y a des informations essentielles dans ces étapes qui seront utilisées par notre système. C'est pourquoi nous avons jugé nécessaire de les présenter dans ce document.

#### 3.1.2.3 Phase de stockage

L'OM et les métadonnées associées seront stockées dans une base de données médiatique ou dans un référentiel d'OM. Les métadonnées seront définies dans le modèle du système.

#### 3.1.2.4 Phase d'annotation

Dans cette phase l'utilisateur peut ajouter des annotations pour la description de la structure, du contexte ou du contenu de l'OM. À cette étape, l'utilisateur sélectionne, l'objet à annoter, ensuite l'utilisateur sélectionne le type de métadonnée, le nom et la description.

#### 3.1.2.5 Phase de recherche

Cette étape correspond à une recherche effectuée par un utilisateur dans le référentiel. L'utilisateur fait une recherche par le contenu à travers les métadonnées associées aux OM. À cette étape, le système traduit la requête de l'utilisateur par une requête compréhensible par le système. La requête sera ensuite exploitée par le système et le résultat c'est-à-dire les OM et les métadonnées associées seront présentés à l'utilisateur.

#### 3.1.2.6 Phase d'archivage

À cette étape, les OM sont compressés avec les métadonnées associées et préservés à long terme pour être exploitables dans le futur. Le fichier résultant est stocké dans le système d'archivage.

### 3.1.3 Gestion des utilisateurs

Nous avons défini deux types d'utilisateurs pour notre approche : L'administrateur et les usagers. L'administrateur a la responsabilité d'installer le système et de l'initialiser pour une application particulière. L'installation et l'initialisation du système a pour objectif de spécifier le modèle de métadonnées qui sera utilisé dans cette application. Le modèle de métadonnées pour une application est spécifique et est basé sur les types de métadonnées prédéfinis pour notre approche: Les métadonnées descriptives, les métadonnées techniques et les métadonnées administratives. L'administrateur peut également définir d'autres types de métadonnées en plus des trois types prédéfinis dans le modèle de métadonnées. Cela rend notre modèle de données plus flexible.

Les simples utilisateurs ont pour rôle d'utiliser le système pour pouvoir stocker, annoter, organiser, rechercher et archiver des OM en utilisant le modèle de métadonnées défini lors de l'initialisation du système.

**Tableau 3.1** Catégories d'utilisateurs

| Utilisateur    | Profil         | Caractéristiques  |
|----------------|----------------|---|
| Administrateur | Administrateur | L'administrateur est responsable de l'initialisation du système de la gestion des utilisateurs (ajout, modification, suppression) création de types de métadonnées qui seront utilisés dans le système.<br>Il doit être familier avec l'initialisation du système et la création de type de métadonnées |
| Utilisateur    | Usager         | Il a un accès limité au système et peut utiliser les fonctionnalités de base du système   |

#### 3.1.4 Fonctionnalités du prototype

Le prototype que nous avons développé offre les fonctionnalités suivantes :

**L'ajout de type de métadonnées** est fait par l'administrateur du système et ces types de métadonnées sont utilisés par les utilisateurs lors de l'annotation d'un OM.

**Le stockage d'un OM** consiste à stocker un OM ainsi que ses métadonnées dans le référentiel de notre application. Un OM stocké peut être accédé et des métadonnées peuvent être ajoutées à l'objet.

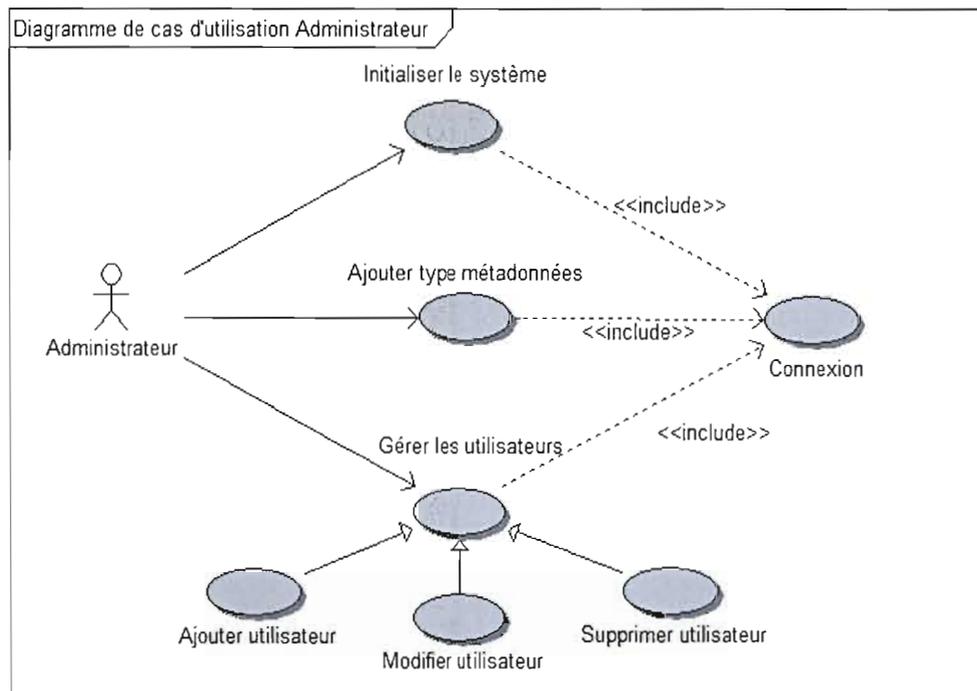
**L'annotation d'un OM** qui permet d'ajouter des informations sur l'OM (les métadonnées associées). Chaque métadonnée est reliée à un type.

**La recherche d'un OM** permettant de rechercher un OM dans le référentiel à travers les métadonnées qui leur sont associées.

**L'archivage d'un OM** consiste à archiver un OM avec ses métadonnées associées dans le référentiel. Quand un OM est archivé, il reste accessible par la recherche pour être désarchivé. Cependant, tant que l'OM n'est pas désarchivé, il n'est pas possible de l'annoter ou de modifier les métadonnées.

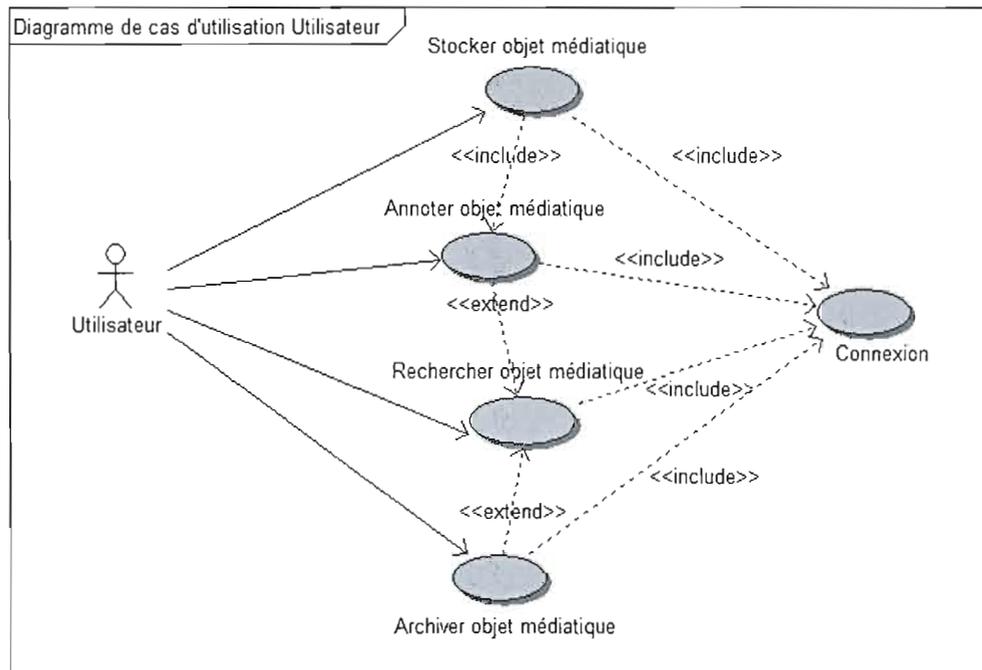
### 3.2 PRÉSENTATION DES CAS D'UTILISATION

Dans cette partie, nous décrivons les cas d'utilisation de notre système SARDOM. Ainsi, l'analyse préalable fait ressortir que les besoins fonctionnels se décomposent de la manière suivante :



**Figure 3.1** Diagramme de cas d'utilisation « Administrative » du système

Ainsi nous avons défini ces différents cas d'utilisation pour le développement du prototype basé sur notre approche.



**Figure 3.2** Diagramme de cas d'utilisation « Utilisateur » du système

Nous allons dans la partie qui suit faire la description détaillée de ces cas d'utilisation.

**Tableau 3.2** Cas d'utilisation de notre système

| Acteur         | Cas d'utilisation             |                          |
|----------------|-------------------------------|--------------------------|
| Administrateur | Initialiser le système        |                          |
| Administrateur | Ajouter un type de métadonnée |                          |
| Administrateur | Gestion des utilisateurs      | Ajouter un utilisateur   |
|                |                               | Modifier un utilisateur  |
|                |                               | Supprimer un utilisateur |
| Utilisateur    | Stocker un OM                 |                          |
| Utilisateur    | Annoter un OM                 |                          |
| Utilisateur    | Rechercher un OM              |                          |
| Utilisateur    | Archiver d'un OM              |                          |

## 3.2.1 Diagramme de cas d'utilisation « Initialiser le système »

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | SARDOM_CU01   |
| <b>Nom du cas d'utilisation</b> | Initialiser le logiciel   |
| <b>Objectif</b>                 | Initialiser le système ainsi que le modèle de données à utiliser                                |
| <b>Acteur principal</b>         | Administrateur  |
| <b>Pré-condition</b>            | Le modèle de données de base est déjà prêt  |
| <b>Déclencheurs</b>             | Aucun   |
| <b>Scénario principal</b>       | L'administrateur installe le système<br>L'administrateur charge le modèle de données à utiliser |
| <b>Scénario alternatif</b>      | Le serveur n'est pas prêt<br>Le modèle de données n'existe pas                                  |
| <b>Post-condition</b>           | Le système est installé et prêt à être utilisé  |

## 3.2.2 Diagramme de cas d'utilisation « Ajouter type de métadonnées »

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | SARDOM_CU02  |
| <b>Nom du cas d'utilisation</b> | Ajouter un type de métadonnées   |
| <b>Objectif</b>                 | Ajouter un type de métadonnées   |
| <b>Acteur principal</b>         | Administrateur   |
| <b>Pré-condition</b>            | Initialiser le système   |
| <b>Déclencheurs</b>             | L'administrateur sélectionne le menu « Ajouter un type de métadonnées »  |
| <b>Scénario principal</b>       | <p>L'administrateur lance l'application principale</p> <p>L'administrateur se connecte à l'application</p> <p>L'administrateur sélectionne l'option « Ajouter un nouveau type »</p> <p>Le système affiche la page d'ajout</p> <p>L'administrateur ajoute les informations du nouveau type</p> <p>L'administrateur appuie sur le bouton sauvegarder</p> |
| <b>Scénario alternatif</b>      | <p>Le serveur n'est pas prêt</p> <p>Le modèle de données n'existe pas</p>  |
| <b>Post-condition</b>           | Le nouveau type est ajouté et prêt à être utilisé  |

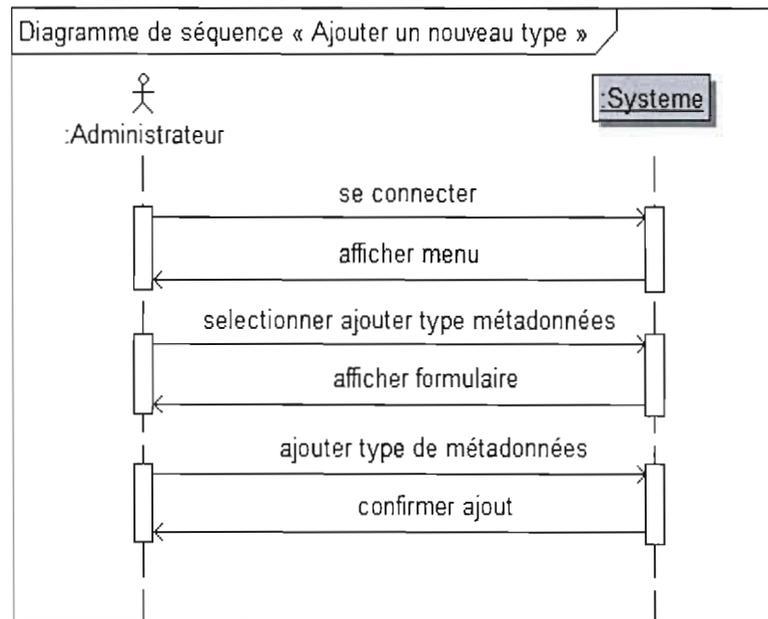


Figure 3.3 Diagramme de séquence « Ajouter un type de métadonnées »

Écran Ajout d'un nouveau type

|            |   |
|------------|---|
| Code Type  | <input type="text" value="STR"/>  |
| Nom type   | <input type="text" value="Structurel"/>   |
| Définition | <input type="text" value="Métadonnées sur la structure de l'objet médiatique"/> |

Figure 3.4 Écran du cas d'utilisation « Ajouter un type de métadonnées »

## 3.2.3 Diagramme de cas d'utilisation « Stocker un OM »

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | SARDOM_CU05   |
| <b>Nom du cas d'utilisation</b> | Stocker un OM   |
| <b>Objectif</b>                 | Stocker un nouvel OM avec ses métadonnées dans le système   |
| <b>Acteur principal</b>         | L'utilisateur   |
| <b>Pré-condition</b>            | Le système SARDOM est déjà installé   |
| <b>Déclencheurs</b>             | Ce cas d'utilisation est déclenché lorsque l'utilisateur demande au système de stocker un nouvel OM dans le système.  |
| <b>Scenario principal</b>       | <p><b>Description générale</b></p> <p>L'utilisateur ajoute un nouvel OM dans le système avec l'ensemble des métadonnées associées à l'OM</p> <p><b>Description détaillée</b></p> <p>L'utilisateur lance l'application principale</p> <p>L'utilisateur se connecte à l'application</p> <p>L'utilisateur sélectionne l'option « Stocker un nouvel objet »</p> <p>L'utilisateur sélectionne l'OM à stocker</p> <p>L'utilisateur saisit les métadonnées descriptives liées à l'OM</p> <p>L'utilisateur appuie sur le bouton valider</p> <p>Le système extrait et crée les métadonnées techniques et administratives liées à l'objet dans le système</p> |
| <b>Scenario alternatif</b>      | <p>Le système n'est pas installé</p> <p>La connexion au système échoue</p>  |
| <b>Post-condition</b>           | L'OM et les métadonnées associées sont stockés dans le système SARDOM   |

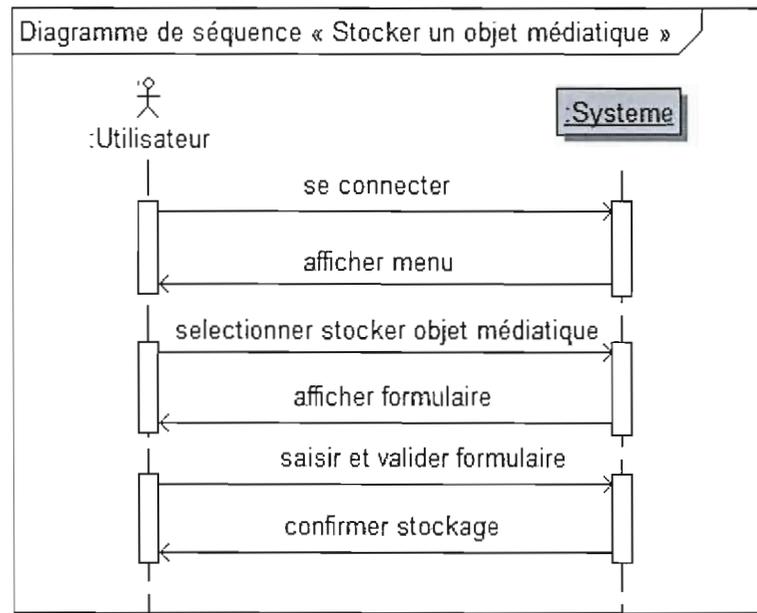


Figure 3.5 Diagramme de séquence « Stocker un OM »

Écran « Stocker d'un objet médiatique »

|                  |   |
|------------------|---|
| Title            | <input type="text"/>  |
| Auteur           | <input type="text"/>  |
| Lieu             | <input type="text"/>  |
| Objet médiatique | <input type="text"/> <input type="button" value="Parcourir ..."/> |
| Commentaire      | <input type="text"/>  |

Figure 3.6 Écran du cas d'utilisation « Stocker un OM »

## 3.2.4 Diagramme de cas d'utilisation « Annoter un objet médiatique »

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | SARDOM_CU06  |
| <b>Nom du cas d'utilisation</b> | Annoter un OM  |
| <b>Objectif</b>                 | L'objectif est de permettre à un utilisateur d'ajouter des métadonnées pour un OM dans le système  |
| <b>Acteur principal</b>         | L'utilisateur  |
| <b>Pré-condition</b>            | Le système SARDOM est déjà installé<br>L'OM existe dans le système SARDOM  |
| <b>Déclencheurs</b>             | Ce cas d'utilisation est déclenché lorsque l'utilisateur demande au système d'ajouter des métadonnées pour un OM.  |
| <b>Scenario principal</b>       | <p><b>Description générale</b></p> <p>L'utilisateur sélectionne et ajoute des métadonnées et/ou des types de métadonnées pour un OM dans le système</p> <p><b>Description détaillée</b></p> <p>L'utilisateur lance l'application principale</p> <p>L'utilisateur se connecte à l'application</p> <p>L'utilisateur sélectionne l'option « Annoter un objet »</p> <p>Le système affiche la liste des objets dans le système</p> <p>L'utilisateur recherche l'objet à annoter</p> <p>Le système affiche les résultats de recherche</p> <p>L'utilisateur sélectionne l'objet</p> <p>L'utilisateur sélectionne le type de métadonnées</p> <p>L'utilisateur ajoute les métadonnées à l'objet</p> |
| <b>Scenario alternatif</b>      | Le système n'est pas installé<br>La connexion au système échoue<br>L'objet n'existe pas dans le système  |
| <b>Post-condition</b>           | Les métadonnées sont ajoutées dans le système  |

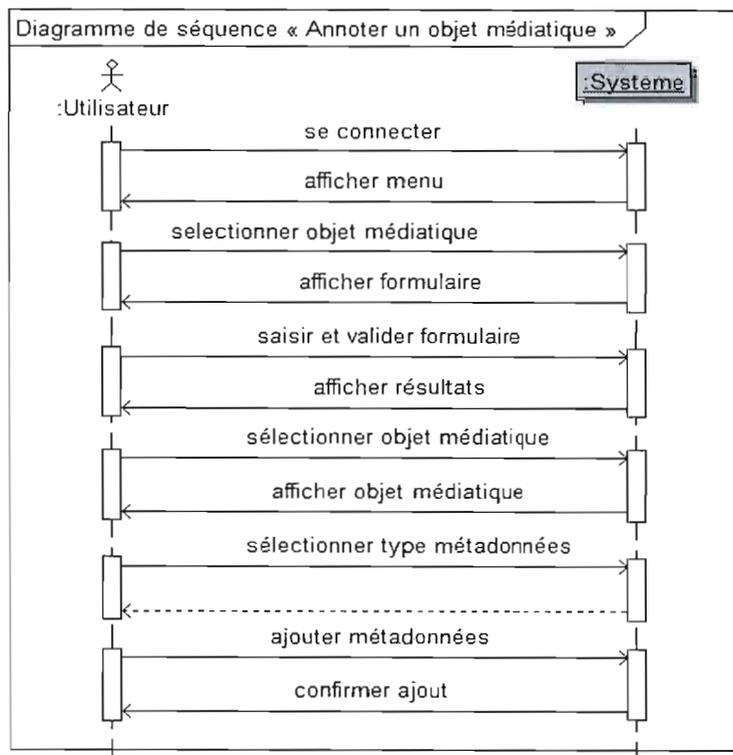


Figure 3.7 Diagramme de séquence « Annoter un OM »

Écran «Annoter d'un objet médiatique»

|                  |   |
|------------------|---|
| Nom              | <input type="text" value="Maison"/>     |
| Titre            | <input type="text" value="Maison"/>     |
| auteur           | <input type="text" value="Mama AMAR"/>  |
| Date de création | <input type="text" value="2011-06-11"/> |

| Type Metadonnée                             | Propriété                                     | Valeur                                  |  |
|---|---|---|--|
| <input type="text" value="Administrative"/> | <input type="text" value="Date de création"/> | <input type="text" value="2011-06-11"/> | <input type="button" value="Valider"/> |

Figure 3.8 Écran du cas d'utilisation « Annoter un OM »

## 3.2.5 Diagramme de cas d'utilisation « Rechercher un objet médiatique »

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | SARDOM_CU07  |
| <b>Nom du cas d'utilisation</b> | Rechercher un OM   |
| <b>Objectif</b>                 | L'objectif est de permettre à un utilisateur de recherche par le contenu un OM déjà stocké dans le système et ce en utilisant les métadonnées liées à l'objet  |
| <b>Acteur principal</b>         | L'utilisateur  |
| <b>Pré-condition</b>            | Le système SARDOM est déjà installé  |
| <b>Déclencheurs</b>             | Ce cas d'utilisation est déclenché lorsque l'utilisateur demande au système de rechercher un OM par le contenu.  |
| <b>Scenario principal</b>       | <p><b>Description générale</b></p> <p>L'utilisateur cherche un OM par son contenu, sa description ou ses caractéristiques</p> <p><b>Description détaillée</b></p> <p>L'utilisateur lance l'application</p> <p>L'utilisateur se connecte à l'application</p> <p>L'utilisateur sélectionne l'option de recherche</p> <p>Le système affiche le formulaire de recherche</p> <p>L'utilisateur saisit les critères de recherche (métadonnées)</p> <p>L'utilisateur appuie sur le bouton de recherche</p> |
| <b>Scenario alternatif</b>      | <p>La connexion au système échoue</p> <p>Il n'y a pas d'OM stockés dans le système</p> <p>Aucun objet ne satisfait aux critères de recherche</p>   |
| <b>Post-condition</b>           | Le système affiche les résultats de recherche  |

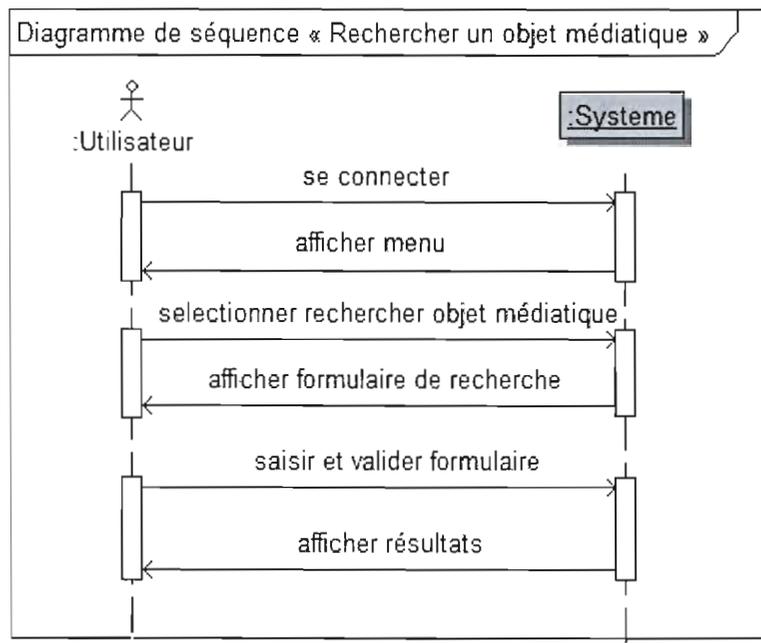


Figure 3.9 Diagramme de séquence « Rechercher un OM »

Écran «Rechercher un objet multimédia»

|                          |            |                             |           |
|--------------------------|------------|-----------------------------|-----------|
| Type d'objets médiatique |            | Image                       |           |
| Lieu                     | Montréal   | Métadonnées descriptives    | Supprimer |
| Taille                   | 300px      | Métadonnées techniques      | Supprimer |
| Date                     | 2003-05-06 | Métadonnées administratives | Supprimer |

Ajouter une ligne

Rechercher    Annuler

Figure 3.10 Écran du cas d'utilisation « Rechercher un OM »

## 3.2.6 Diagramme de cas d'utilisation « Archiver un objet médiatique »

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | SARDOM_CU08  |
| <b>Nom du cas d'utilisation</b> | Archiver un OM   |
| <b>Objectif</b>                 | L'objectif est de permettre de compresser un OM avec les métadonnées associées et d'archiver le fichier compressé dans le système pour une exploitation future   |
| <b>Acteur principal</b>         | L'utilisateur  |
| <b>Pré-condition</b>            | Le système SARDOM est déjà installé  |
| <b>Déclencheurs</b>             | Ce cas d'utilisation est déclenché lorsque l'utilisateur demande au système d'archiver un OM dans le système.  |
| <b>Scenario principal</b>       | <p><b>Description générale</b></p> <p>L'utilisateur sélectionne un OM et l'archive avec ses métadonnées dans le système SARDOM.</p> <p><b>Description détaillée</b></p> <p>L'utilisateur lance l'application</p> <p>L'utilisateur se connecte à l'application</p> <p>L'utilisateur sélectionne l'option d'archivage d'un OM</p> <p>Le système affiche le formulaire de recherche</p> <p>L'utilisateur saisit les critères de recherche (métadonnées)</p> <p>L'utilisateur appuie sur le bouton de recherche</p> <p>Le système affiche les résultats de recherche</p> <p>L'utilisateur sélectionne l'OM à archiver et clique sur le bouton archiver</p> <p>Le système crée l'archive médiatique (OM avec ses métadonnées)</p> |
| <b>Scenario alternatif</b>      | <p>L'application n'est pas lancée</p> <p>La connexion au système échoue</p> <p>L'OM n'existe pas dans le système</p>   |
| <b>Post-condition</b>           | L'OM est compressé avec ses métadonnées et le fichier résultant est archivé dans le système SARDOM   |

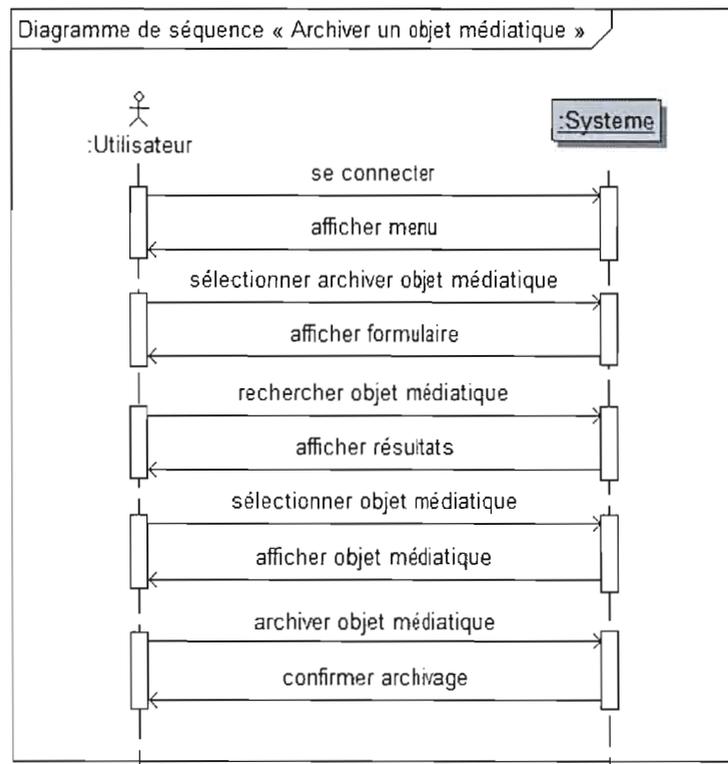


Figure 3.11 Diagramme de séquence « Archiver un objet médiatique »

Écran « Archiver un objet médiatique »

|               |   |
|---------------|---|
| Nom           | Statut de la liberté  |
| auteur        | Mama AMAR   |
| Lieu          | New York  |
| Date création | 2001-12-28  |
| Description   | Statut de la liberté lors de mon voyage à New York en decembre 2010 |

Archiver Annuler

Figure 3.12 Écran du cas d'utilisation « Archiver d'un OM »

### 3.3 PROPOSITION D'UN MODÈLE DE DONNÉES

Dans cette partie nous allons faire la présentation du modèle de données de notre approche c'est-à-dire le moyen d'organiser de façon logique, les objets ainsi que les liens pour l'accès aux objets. Nous allons d'abord faire la représentation du modèle de données avec le formalisme UML puis nous allons faire la représentation hiérarchique ou les données sont organisées en structures arborescentes. Chaque enregistrement a un et un seul parent et peut avoir zéro ou plusieurs enfants.

#### 3.3.1 Diagramme de classe de notre modèle

Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation. Nous représentons ainsi le diagramme de classe de notre système.

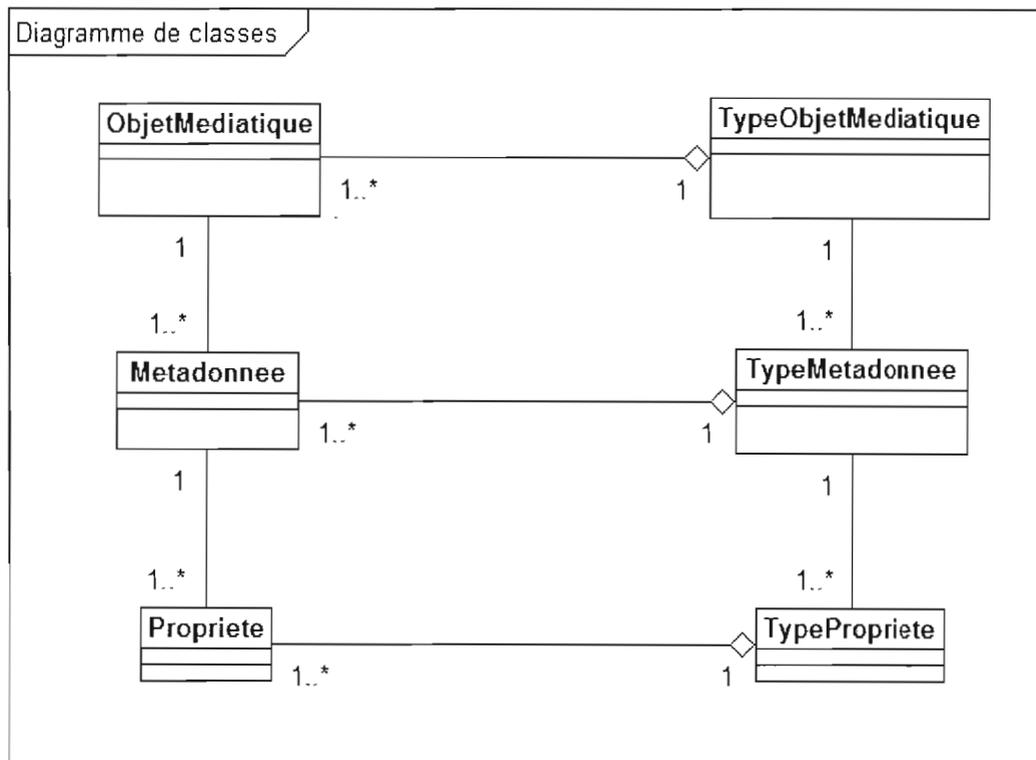


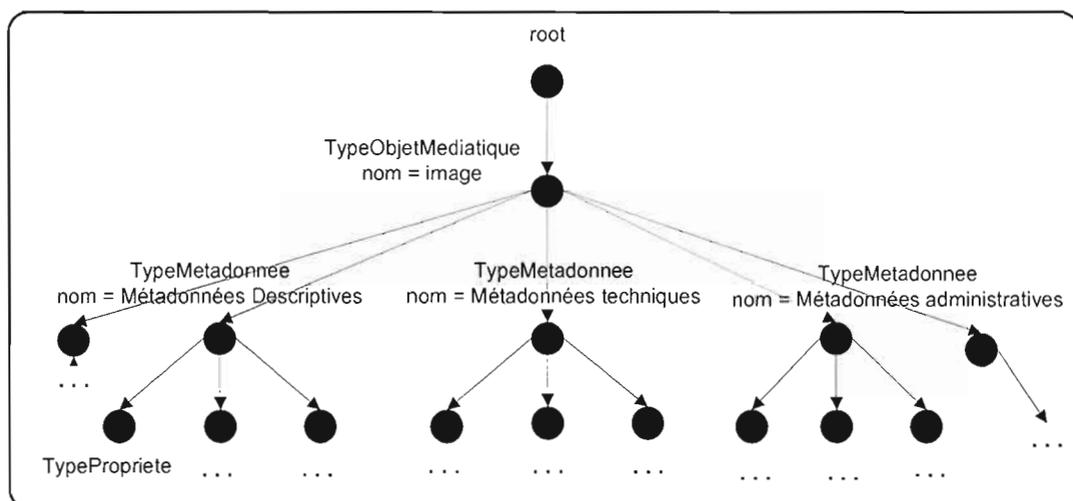
Figure 3.13 Diagramme de classe de notre modèle

La **Figure 3.13** montre qu'un type d'OM (image, vidéo, ou audio) est composé d'OM et peut avoir un ou plusieurs types de métadonnées. Un type de métadonnées est composé de plusieurs métadonnées et peut avoir plusieurs types de propriétés. À chaque OM correspond un ou plusieurs métadonnées et les métadonnées ont des propriétés.

L'avantage du modèle est qu'il permet d'ajouter des types de métadonnées pour chaque type d'OM. Et ainsi d'autres modèles de métadonnées peuvent être ajoutés dans notre système en ajoutant ces modèles comme des types de métadonnées. Il est possible aussi d'avoir plusieurs standards dans ce modèle en les intégrant comme des types de métadonnées.

### 3.3.2 Représentation arborescente de notre modèle

Le **Figure 3.14** est une représentation arborescente des types de notre modèle de données. Ainsi avec ce modèle, d'autres types de métadonnées peuvent être ajoutés avec d'autres attributs. En effet, d'autres nœuds peuvent être ajoutés au nœud *TypeMetadonnee* et celui-ci peut contenir d'autres nœuds avec comme valeur les types de propriétés. Cela rend notre modèle plus flexible et permet de l'adapter selon le contexte ou le domaine d'utilisation.



**Figure 3.14** Représentation arborescente des types modèle de données

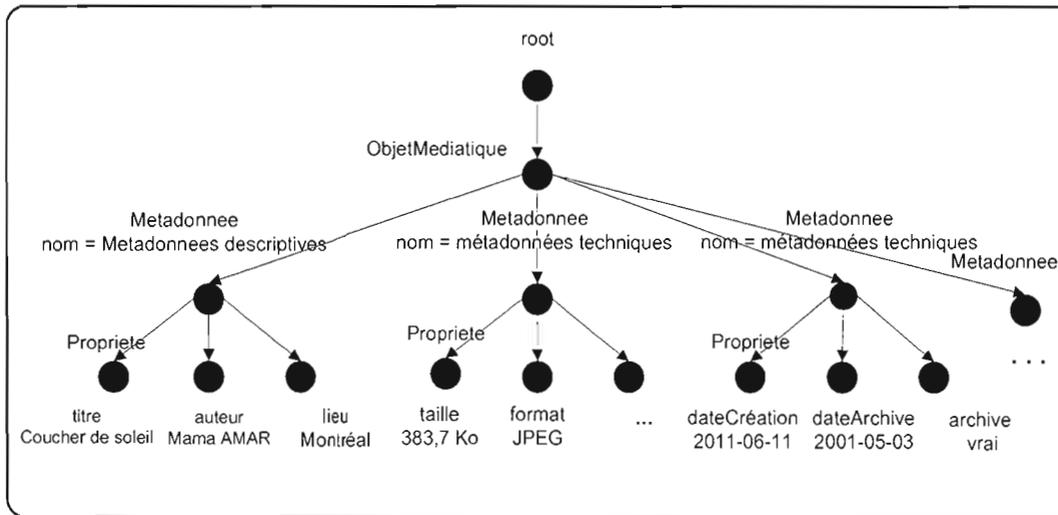


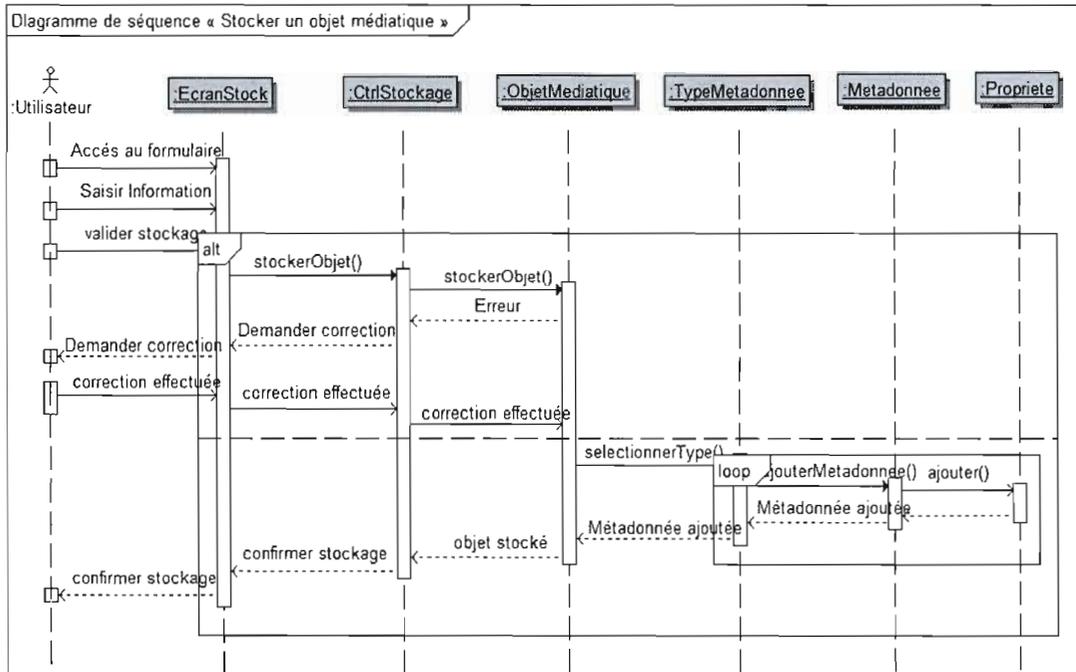
Figure 3.15 Représentation du modèle de données

Dans la **Figure 3.14** nous avons une instance des types de données de base de notre modèle. Ces types de données sont utilisés dans la création des objets médiatiques et des métadonnées. Dans la **Figure 3.15** représente une instance de notre modèle sous forme arborescente. Ainsi à titre d'exemple nous avons pris une image d'un coucher de soleil à Montréal. Et nous avons utilisé trois types de métadonnées qui sont les types de base de notre modèle. Pour chaque type nous avons défini quelques attributs avec leurs valeurs respectives. D'autres types de métadonnées peuvent ainsi être ajoutés dans cette instance et chaque type contenir d'autres propriétés.

### 3.4 CONCEPTION DU PROTOTYPE

Dans cette partie nous présentons la conception du système à développer. Dans un premier temps nous utilisons les diagrammes d'interaction c'est-à-dire les diagrammes de séquence détaillés pour montrer comment les instances des objets de notre système communiquent entre elles. Cela va nous permettre de réaliser le diagramme de classes final (Diagramme de classes de conception) qui servira de base au codage de l'application. Ensuite nous présenterons l'architecture des modules du système.

## 3.4.1 Diagramme de séquence détaillée « Stocker un OM »



**Figure 3.16** Diagramme de séquence détaillée « Stocker un OM »

La **Figure 3.16** ci-dessus décrit le scénario de l'ajout d'un OM

- L'utilisateur sélectionne l'OM à stocker
- L'utilisateur saisit les informations descriptives de l'OM
- L'utilisateur soumet le formulaire
- L'application extrait les métadonnées, crée les types de métadonnées et ajoute les références des métadonnées et des types à l'OM
- L'application stocke l'OM et ses métadonnées dans le système
- L'application confirme le stockage de l'OM ainsi que des métadonnées

## 3.4.2 Diagramme de séquence détaillé « Annoter un OM »

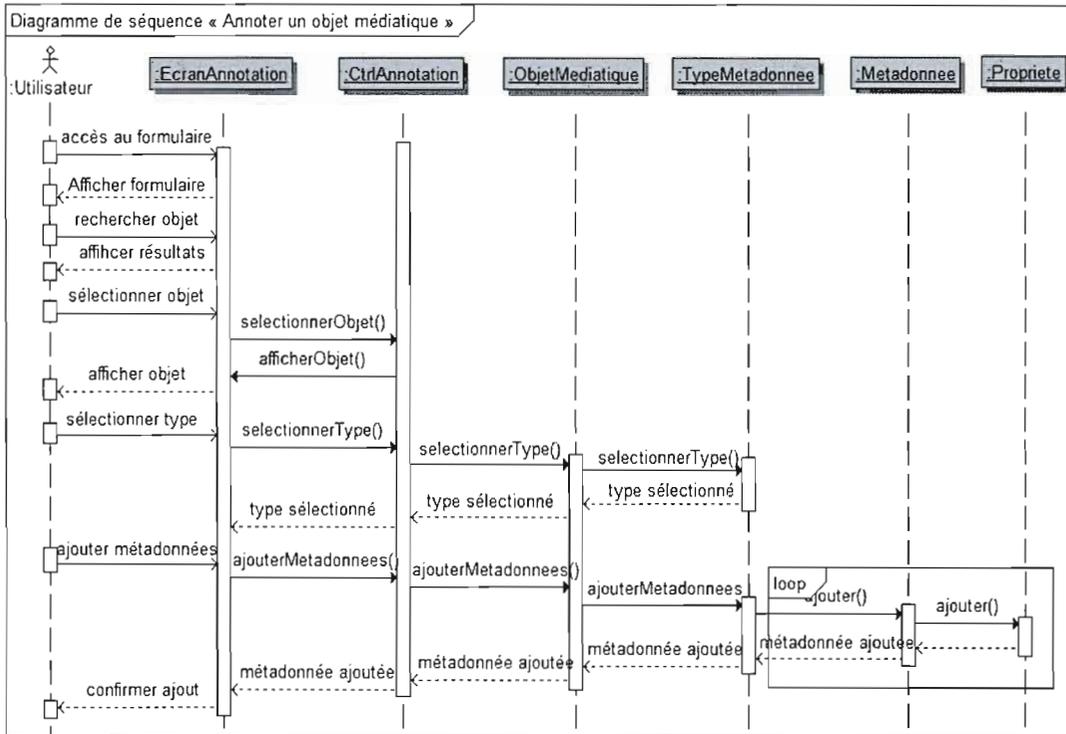


Figure 3.17 Diagramme de séquence détaillée « Annoter un OM »

Le **Figure 3.17** ci-dessus décrit le scénario de l'annotation d'un OM

- L'utilisateur recherche l'OM
- Le système affiche les résultats de recherche
- L'utilisateur sélectionne l'OM
- L'utilisateur sélectionne le type de métadonnées parmi les types qui existent
- L'utilisateur ajoute des métadonnées (une propriété avec un attribut et une valeur) pour ce type
- Le système ajoute les références à l'OM

### 3.4.3 Diagramme de séquence détaillée « Rechercher un OM »

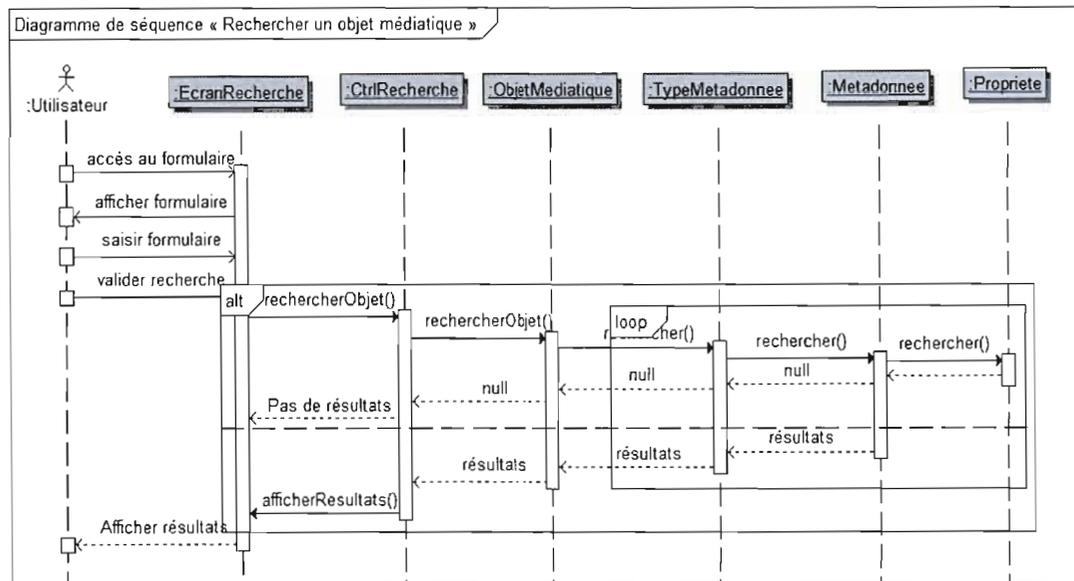


Figure 3.18 Diagramme de séquence détaillée « Rechercher un OM »

Le **Figure 3.18** décrit le scénario de Recherche d'un OM

- Le système affiche le formulaire de recherche
- L'utilisateur saisit les critères de recherche (Type d'OM, type de métadonnées, métadonnées)
- L'utilisateur valide le formulaire
- Le système parcourt les OM dans le système et pour chaque objet
- Le système parcourt les différents nœuds des types de métadonnées et crée une liste de résultats des objets multimédia qui satisfont les critères de recherches.
- Le système affiche les résultats de recherche

## 3.4.4 Diagramme de séquence détaillé « Archiver un objet médiatique »

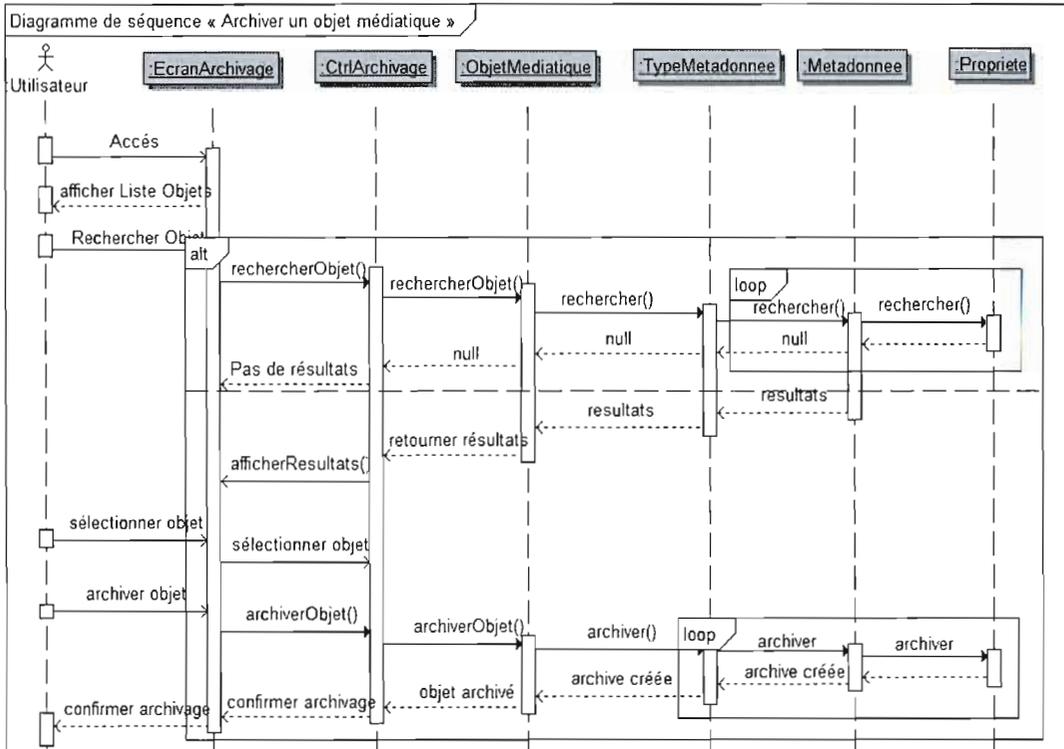


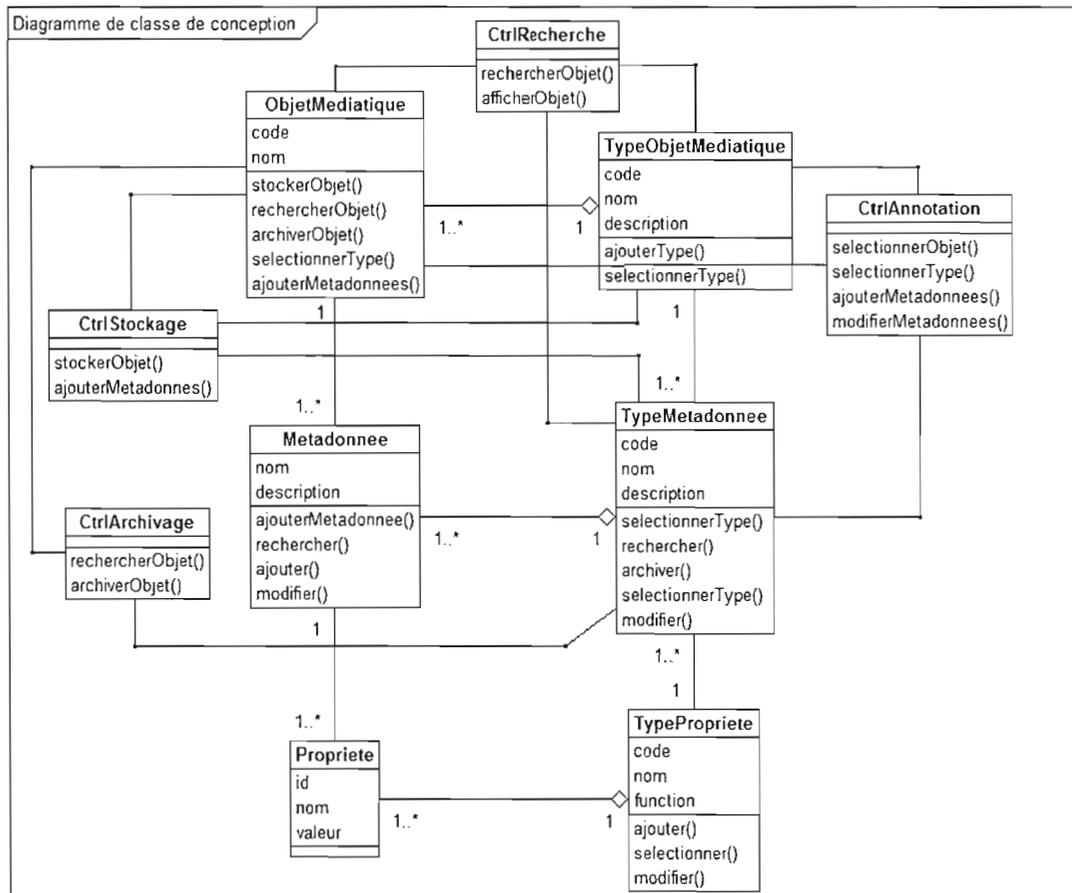
Figure 3.19 Diagramme de séquence « Archivage d'un OM »

Le **Figure 3.19** décrit le scénario de l'archivage d'un OM

- L'utilisateur se connecte
- L'utilisateur recherche l'OM à archiver
- Le système affiche les résultats de recherche
- L'utilisateur sélectionne l'OM
- Le système affiche l'OM
- L'utilisateur valide l'archivage de l'OM
- Le système crée un fichier compressé de l'OM avec ses métadonnées et archive le fichier dans le répertoire archive.

### 3.4.5 Diagramme de classe de conception

Dans cette partie nous allons présenter le diagramme de classes de conception de notre modèle qui représente les différentes entités, les relations entre ces entités ainsi que les classes permettant de gérer ces entités.

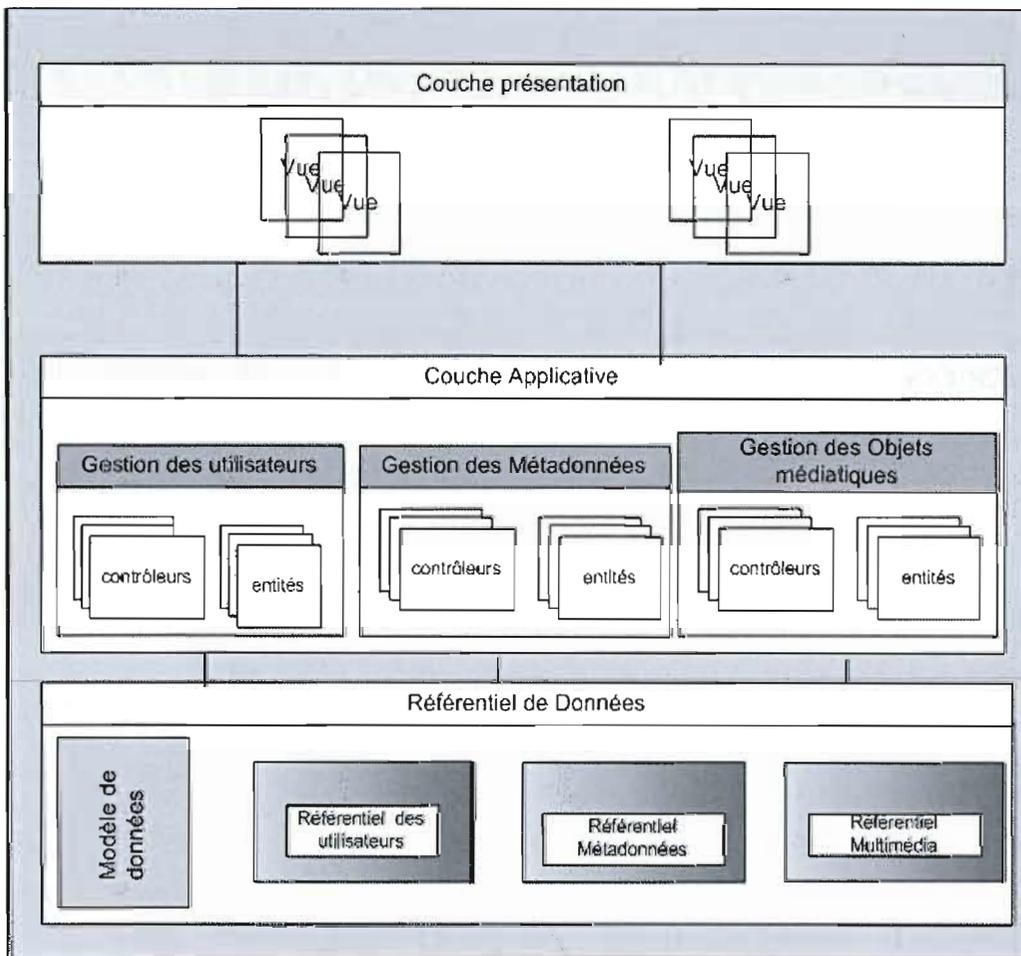


**Figure 3.20** Diagramme de classe de conception

Ainsi la phase de conception nous a permis d'introduire de nouvelles classes qui seront implémentées pour prendre en charge les responsabilités techniques, de redistribuer les messages et les événements du modèle dynamique sur les différentes couches de conception.

### 3.5 ARCHITECTURE CIBLE DU PROTOTYPE

L'architecture de notre système est basée sur une architecture n-tiers. Ainsi nous avons plusieurs couches : La couche présentation, la couche applicative et le référentiel de données.



**Figure 3.21** Architecture cible du prototype

La **Figure 3.21** décrit l'architecture de notre système. Le cœur du système est représenté par le Système d'Archivage D'Objets médiatiques (*SARDOM* *Système d'Archivage d'Objets médiatiques*) qui permet le stockage des OM et des métadonnées associées. Il offre des possibilités de création, d'archivage, de recherche et de récupération des OM et des métadonnées. Il permet aussi la gestion de différents types d'OM et des métadonnées.

Cette architecture est composée de plusieurs modules. Un module pour la gestion des OM et un module de gestion des métadonnées liées aux OM. Le module de gestion des métadonnées permet l'extraction des métadonnées dans les OM ainsi que la gestion des annotations faites par l'utilisateur.

Le module de gestion des OM utilise les services du module de gestion des métadonnées. Ce dernier permet de faciliter la recherche des OM dans le référentiel de données.

### 3.5.1 Couche présentation

Cette couche représente toute l'interface visible de notre système. Cette couche joue le rôle d'intermédiaire entre les utilisateurs et l'application. Elle permet de transformer les actions des utilisateurs en événements de l'application.

Ce module a pour rôle de formuler les requêtes utilisateurs et de les analyser. Après analyse, les requêtes sont transmises à la couche d'application qui est composée du module de gestion des métadonnées et du module de gestion des OM.

### 3.5.2 Module de gestion des métadonnées

Le module gère la création et l'accès aux métadonnées. Ce module gère aussi toutes les requêtes (ajout, mis à jour, recherche) des métadonnées dans le référentiel. Ces requêtes sont sous format XPath. Ce module permet ainsi l'acquisition des métadonnées, l'analyse des métadonnées, la correspondance des métadonnées vers le modèle des métadonnées défini dans le système, et le stockage des métadonnées dans le référentiel de données. Ce module permet aussi l'ajout d'un modèle de métadonnées ou la transformation d'un modèle vers un modèle défini dans le système. En effet, notre approche peut supporter plusieurs modèles et permet de faire une correspondance entre un modèle externe et le modèle défini dans notre système. Par exemple : un modèle qui respecte le schéma de métadonnées Dublin Core peut être importé dans notre système en ajoutant comme type de métadonnées Dublin Core et en ajoutant les différents éléments de ce modèle comme des propriétés avec leurs attributs et valeurs respectifs.

### 3.5.3 Module de gestion des métadonnées

Ce module s'appuie sur le module précédent pour la recherche et l'affichage des OM. Ce module gère le stockage et la recherche des OM. Le module de gestion des OM devra fournir un moteur de recherche, un système d'annotation des OM, un moteur de recherche des OM et un moteur d'archivage des OM. Ce module devra donc s'appuyer sur les services du module de gestion des métadonnées pour les opérations sur les métadonnées. Le stockage des OM et des métadonnées sont assurés par le référentiel de données.

### 3.5.4 Module de gestion des utilisateurs

Ce module permet de la gestion des utilisateurs et leur rôle. Comme expliqué dans la partie présentation de notre approche (§ 3.1.3 *Gestion des utilisateurs*) nous avons défini deux catégories d'utilisateurs dans notre système : l'administrateur et les simples utilisateurs. Ce module permet ainsi d'ajouter de supprimer ou de modifier un utilisateur.

### 3.5.5 Le référentiel de données

Cette couche représente le référentiel utilisé pour le stockage des données (objets médiatique et métadonnées). Elle peut être soit un système de gestion de base de données ou un système de fichier. Ce système devra donc permettre l'accès aux OM à travers un langage de requête. L'archivage et le filtrage des OM sont facilités par les métadonnées associées à ces objets. Ainsi, pour chaque OM on définit un identifiant qui permet de le lier aux métadonnées associées se trouvant dans le référentiel des métadonnées.

Enfin, toutes les métadonnées seront représentées dans le modèle de données défini dans la section précédente avant qu'elles ne soient stockées dans le référentiel de données.

### 3.6 CONCLUSION

Dans cette partie nous avons fait la présentation de notre approche en détails. Nous avons utilisé le langage UML pour la description du modèle de données. Ainsi, nous avons utilisé une architecture modulaire formée de trois parties. L'architecture modulaire de notre approche a pour avantages de faciliter la maintenabilité et la réutilisabilité de notre système. De plus, le modèle de données que nous utilisons permet d'ajouter d'autres types de métadonnées, ce qui rend notre système plus flexible. Enfin, le découpage en couches de notre approche permet une bonne gestion des tests et des évolutions éventuelles du système.

Dans la partie qui suit, nous allons présenter l'implémentation du système ainsi que son expérimentation dans le cadre d'un projet de gestion d'archives de photos personnelles. Nous comptons aussi l'expérimenter dans le cadre du projet ePortfolio de la MATI.



## CHAPITRE IV

### IMPLÉMENTATION ET EXPÉRIMENTATION

#### 4.1 INTRODUCTION

Dans ce chapitre nous présentons d'abord les outils et technologies utilisés et le pourquoi du choix de ces outils. Ensuite nous présentons l'implémentation du prototype que nous avons développé et qui se base sur notre approche. Enfin nous présentons l'expérimentation dans le cadre d'un projet de gestion de photos personnelles.

#### 4.2 OUTILS ET TECHNOLOGIES UTILISÉS

Pour l'implémentation de notre approche, nous nous sommes basés sur le framework **Apache-Sling** qui utilise comme référentiel de contenu **Apache-Jackrabbit** qui est une implémentation de **Java Content Repository** (JSR-170). Nous allons dans cette partie présenter ces différentes technologies.

##### 4.2.1 Java Content Repository

Java Content Repository est une API standard pour accéder à des référentiels de contenu en Java indépendamment de l'application. Java Content Repository est spécifiée dans les documents JSR 170 et 283 [JSR 170]. En plus de fournir des services pour stocker et récupérer des données (Texte, Images, Vidéo, Audio), l'API Java Content Repository est conçu pour gérer, rechercher des documents, des objets binaires, des données hiérarchiques avec des métadonnées. Il offre également la gestion de versions. Elle est une standardisation d'une API qui peut être utilisée pour accéder à un référentiel de contenu.

Avec cette API, les données peuvent être stockées soit dans un SGBDR, un système de fichiers ou d'un document XML. L'un des avantages de l'API c'est le fait d'avoir un standard permettant d'accéder à un référentiel de contenu indépendamment du fournisseur.

#### 4.2.1.1 Modèle de données du référentiel

Selon la Spécification de l'API JSR-170 [JSR 170], avec l'API Java Content Repository un référentiel de contenu est composé d'un ou de plusieurs espaces de travail (Workspaces). Chaque espace de travail contient une arborescence d'éléments, un élément pouvant être soit un nœud ou une propriété. Un nœud peut avoir zéro ou plusieurs enfants, et zéro ou plusieurs propriétés associées. Les données sont stockées dans ces nœuds.

#### 4.2.1.2 Les types de nœuds prédéfinis

Selon [IBM, JCR], un référentiel Java Content Repository contient un certains nombre de types de nœuds qui sont prédéfinis. Ces types sont :

Le type *nt:base* est le type de nœud principal qui doit être supporté par chaque référentiel.

Le type *nt:unstructured* permet à un nœud d'avoir un nombre illimité de nœuds fils ou de propriétés qui peuvent avoir aussi n'importe quel nom.

Le type *nt:file* représente un fichier dans le référentiel

Le type *nt:folder* représente un dossier

Le type *nt:resource* représente généralement le contenu d'un fichier

Le type *nt:version* est le type requis pour les référentiels qui supportent des versions.

## 4.2.2 Apache-Jackrabbit

Apache-Jackrabbit est une application entièrement basée sur l'API Java Content Repository (JCR). Jackrabbit stocke et gère du contenu structuré ou non-structuré sous la forme de référentiel ou dépôt organisé (Repository) et met l'accent sur les services de contenu (Content Services) [JACKRABBIT]. Jackrabbit offre aussi un comme fonctionnalités la gestion des versions (Versioning), la recherche plein texte (Full-Text) basé sur Lucene<sup>1</sup>, le contrôle de l'accès au contenu, l'annotation sémantique du contenu, le suivi des événements associés aux modifications du contenu [JACKRABBIT].

Apache-Jackrabbit intègre à la fois le langage de requête XPath et le langage SQL pour la recherche et l'extraction de contenu comme dans la Spécification de l'API JSR-170. Ainsi le moteur de recherche dans Apache-Jackrabbit est indépendant de la syntaxe de requête utilisée (XPath, SQL) [JACKRABBIT].

### 4.2.2.1 Architecture d'apache-jackrabbit

L'architecture générale du Jackrabbit est composée de trois couches: une couche d'application de contenu, une couche API et une couche d'implémentation du référentiel de contenu [JACKRABBIT, ARC].

### 4.2.2.2 Les types de nœuds

Selon le site web du framework [JACKRABBIT, NT], il existe un type pour chaque nœud dans Jackrabbit. Un type pour un nœud permet de définir les nœuds fils et les propriétés que ce nœud peut avoir. Il existe deux catégories types de nœuds dans Jackrabbit : Les types « **primary** » principaux et les types « **mixin** » [JACKRABBIT, NT]. Un type de nœud « **primary** » définit généralement la structure de ce nœud c'est-à-dire les nœuds fils et les propriétés autorisés et requis pour ce nœud. Les types de nœuds « **mixin** » servent à spécifier d'autres propriétés supplémentaires.

---

<sup>1</sup> Lucene est une librairie Java qui offre des fonctionnalités de moteur de recherche plein texte

### 4.2.3 Apache-Sling

Apache-Sling est un framework basé sur l'API Java Content Repository, il utilise pour cela Apache Jackrabbit pour stocker le contenu. Apache-Sling est basée sur l'architecture REST<sup>1</sup> (Representational State Transfer). Selon le Site web du framework « *Sling est le premier framework web dédié au Java Content Repository JSR-170 pour mettre en œuvre des applications simples, tout en fournissant un framework niveau entreprise pour des applications plus complexes* » [SLING].

#### 4.2.3.1 Ressource dans apache-Sling

Une ressource représente l'un des principaux composants d'Apache-Sling. Chaque ressource contient un chemin par lequel elle est traitée dans l'arborescence des ressources, un type de ressource et certaines métadonnées de ressource. Apache Sling offre ainsi des fonctionnalités pour créer des ressources, accéder aux ressources et faire des traitements sur ces ressources [SLING, RES].

#### 4.2.3.2 Les composants d'apache-Sling

– Selon le site web du framework [SLING, ARC], l'architecture d'Apache-Sling est composé de :

**OSGi :** Sling est basé sur l'architecture de composants OSGI (Open Services Gateway Initiative <http://www.osgi.org>)

**L'API Sling :** « *L'API Sling est une extension de l'API Servlet et fournit plus de fonctionnalités pour interagir avec le framework Sling. Elle permet également d'étendre Sling lui-même et de mettre en œuvre des applications Sling.* [SLING, ARC]»

**Traitement des requêtes :** Comme tout est ressource dans Sling, pour traiter une requête, l'URL de la requête est d'abord converti en ressource ensuite Sling sélectionne la Servlet ou le Script pour traiter la requête.

**Ressources :** Une ressource représente l'un des principaux composants d'Apache-Sling.

---

<sup>1</sup> Representational State Transfer

**Servlets et Scripts :** Une application basée sur Sling peut utiliser soit des Scripts ou des Servlets Java pour exécuter des services web selon l'architecture REST. Les Servlets et Scripts sont considérés comme des ressources dans Sling et sont accessible par un chemin.

**Lanceur « Launchpad » :** « *Sling utilise un lanceur pour s'intégrer à un conteneur de Servlets existantes, permettant de démarrer Sling comme une application Web ou fournissant une classe principale pour représenter une application Java autonome* » [SLING, ARC].

#### 4.2.4 Apache-Tika

Selon le site web du framework [TIKA], Apache-Tika est un framework pour la détection et l'extraction de métadonnées et de contenu texte structuré à partir des différents documents en utilisant des bibliothèques de parseurs existantes. Apache-Tika fournit une interface de programmation d'application générale qui peut être utilisée pour détecter le type de contenu d'un document ainsi que pour analyser le contenu textuel et les métadonnées de plusieurs formats de documents. Ces formats sont entre autres : publication électronique, Texte, Audio, Image, Vidéo etc. Apache-Tika offre ainsi deux fonctionnalités que sont la détection du type et l'analyse de contenu.

##### 4.2.4.1 Formats de documents supportés

Dans cette partie, nous allons présenter quelques formats de documents multimédia supportés par Apache-Tika. Ainsi selon le site Web d'Apache-Tika [TIKA, FORMATS], le framework supporte plusieurs formats de documents dont les formats multimédia suivant :

**Audio:** Apache-Tika, permet d'analyser un document audio et peut détecter plusieurs formats audio courants et extraire des métadonnées sur ces formats. Il contient des pour cela des classes permettant de traiter les formats audio simples comme le format MP3 [TIKA, FORMATS]. **Exemple:** *xmpDM:album, Author, xmpDM:artist, samplerate, xmpDM:genre, Content-Type, resourceName*

**Image:** Apache-Tika permet d'analyser et d'extraire de simples métadonnées à partir de formats d'image pris en charge par la plateforme Java. Il contient aussi d'autres classes permettant d'extraire des métadonnées Exif à partir d'images de formats JPEG et TIFF

[TIKA, FORMATS]. **Exemple:** *tiff:ImageLength, tiff:ImageWidth, tiff:XResolution, Resolution Unit, Image Width, resourceName*

**Vidéo:** La version actuelle d'Apache-Tika supporte seulement le format Flash vidéo et permet d'analyser ce format et d'en extraire des métadonnées [TIKA, FORMATS].

**Exemple:** *Mime, Title, Author, totaldataRate, width, duration, height, has Audio*

#### 4.2.4.2 L'API analyseur

Pour traiter ces documents et extraire des métadonnées, Apache-Tika utilise un Analyseur (Parser). Cet analyseur est une Interface Java (l'interface *org.apache.tika.parser.Parser*) qui permet de façon transparente à l'utilisateur « *d'offrir un mécanisme simple et efficace pour les applications client d'extraire des métadonnées pour un grand nombre de documents* » [TIKA, PARSER].

#### 4.2.5 Choix des outils

Dans cette partie nous avons présenté les outils que nous avons utilisés pour l'implémentation de notre système. Bien qu'il existait d'autres outils et technologies, nous avons choisi ces outils pour les raisons que nous allons citer. D'abord nous avons choisi le langage Java car il est libre, supporté par de nombreux fournisseurs et qu'il offre plusieurs framework libres. De surcroit Java est un langage orienté objet ce qui facilite la maintenance d'une application développée en Java.

Nous avons choisi Apache-Sling car c'est un framework qui permet de stocker et de publier de grandes quantités d'informations de différents types, notamment des OM. En effet, Apache Sling est framework web basé sur les principes REST qui permet de développer facilement des applications orientées contenus. De plus Apache-Sling offre la possibilité de prototypage et de développement d'applications web de façon rapide qui sont dynamique, extensible et facilement maintenable. Apache Sling est un framework orienté vers le contenu permettant de manipuler facilement le contenu. En plus des fonctionnalités offertes par les systèmes de gestion de base de données relationnels, Apache Sling offre la possibilité de verrouillage, de transactions, de gestion des versions, de l'observation et de la recherche de contenu. De plus, en termes de performances, Apache Sling semble avoir aussi un atout majeur pour le stockage de données avec des temps d'accès à la base plus rapides qu'un Système de gestion de base de données relationnelle.

En outre, Apache Sling permet, pour les gestions des OM, le stockage hiérarchique de ces derniers. Ceci présente l'avantage d'avoir un modèle dynamique pour stocker les données qui permet d'enrichir facilement les données au besoin et permet de parcourir le contenu sans connaître sa structure. Apache Sling permet aussi l'annotation de ces OM et facilite la recherche de ses derniers. Ce qui permet de faciliter la gestion et l'archivage des OM.

Ainsi nous avons choisi ces outils car ils répondent à nos besoins pour tous les cas d'utilisation de notre prototype. Nos choix sont ainsi motivés par les besoins de notre approche et les possibilités offertes par ces outils.

### 4.3 IMPLÉMENTATION

Dans cette partie nous présentons l'environnement de développement du prototype. Ensuite nous présentons l'architecture applicative du prototype. Nous rappelons que le prototype est basé sur l'approche que nous avons défini précédemment.

#### 4.3.1 Environnement de développement

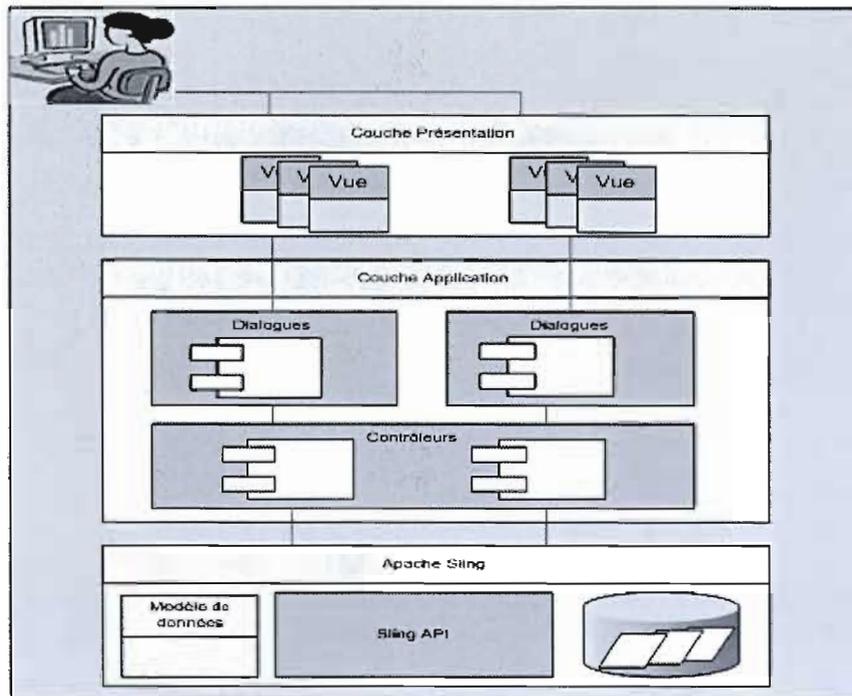
Le projet regroupe plusieurs modules et certains modules étaient dépendants. Tous les modules du projet ont été développés avec l'environnement de développement Eclipse Helios (la version 3.6).

Eclipse est une plateforme de développement libre développée par « The Eclipse Foundation » permettant de développer, de modéliser, de concevoir des applications et des logiciels informatiques. Eclipse est un framework extensible et offre un ensemble de composants permettant de faciliter le développement d'applications de grande envergure.

Pour la modélisation, nous avons utilisé BOUML qui regroupe une série de logiciels comprenant un modèleur UML2 et plusieurs programmes externes dont des générateurs de code et reverse. BOUML est distribué sous licence GPL (General Public Licence).

#### 4.3.2 Architecture applicative

Comme expliqué dans la partie précédente, le prototype que nous développons est basé sur le framework Apache Sling. La figure ci-dessous illustre l'architecture logicielle, c'est-à-dire, les différents composants logiciels du prototype ainsi que les relations qui existent entre eux.



**Figure 4.1** Architecture applicative du prototype

Dans cette figure nous avons les différentes couches de l'application qui sont :

**Dans la couche présentation**, nous avons l'interface homme-machine pour les utilisateurs qui permet de restituer les données à l'utilisateur, et transforme ses actions en événement de l'application.

**Dans la couche applicative** nous avons les classes de contrôle et les classes de dialogue qui se chargent de traiter les requêtes de la vue pour le stockage, l'annotation, la recherche et l'archivage des OM à travers le framework Apache Sling.

**Le référentiel de données** correspondant au framework Apache Sling assure le stockage des OM suivant le modèle de données défini dans le référentiel et des métadonnées associées et l'accès à ces derniers, offre les fonctionnalités de recherche avec le standard XPath et permet la gestion des versions des OM et l'archivage de ces objets dans le référentiel.

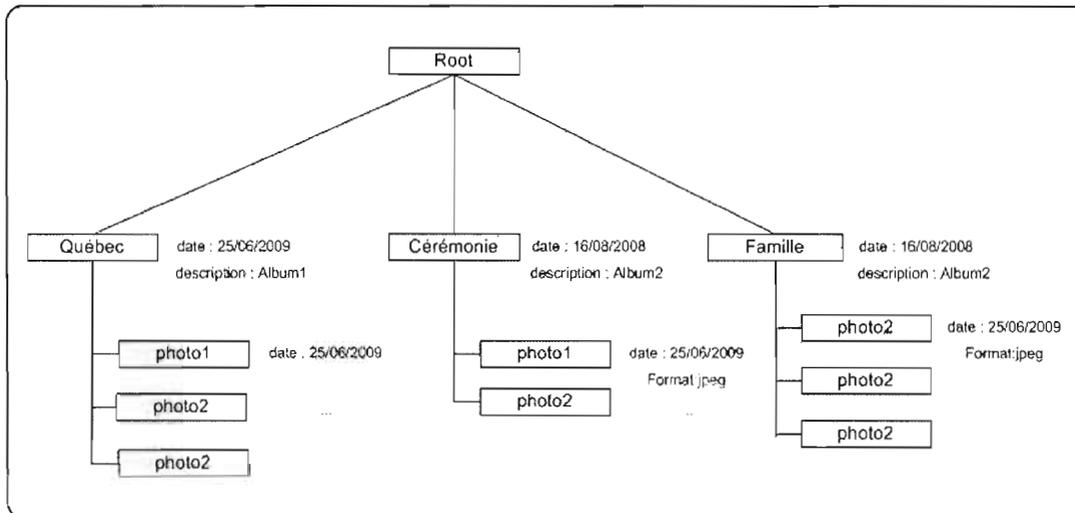
#### 4.4 EXPÉRIMENTATION

Dans cette partie, nous allons présenter l'expérimentation de notre outil dans le cadre d'un projet de la gestion d'une archive d'albums de photos personnel. Cette expérimentation va permettre de démontrer la gestion et l'archivage des OM dans le cadre d'un projet réel. Elle va permettre aussi de comprendre l'interopérabilité de notre approche par rapport à d'autres applications.

##### 4.4.1 Présentation du cadre d'étude

Dans cette partie, nous présentons le cadre qui nous a servi d'exemple pour tester notre application. Dans cette exemple nous nous intéressons à la gestion d'une archive d'albums de photos personnel. Dans cet exemple nous avons un ensemble d'album photos à archiver. Chaque album est dans un répertoire et contient des informations sur l'album par exemple la date de l'album, une description. Dans chaque album nous avons un ensemble de photos et chaque photo contient un ensemble d'information sur elle (ses métadonnées) par exemple la date, le lieu, une description, la longueur, la hauteur, format etc.

Le schéma suivant est une illustration du modèle de données de notre cadre applicatif.



**Figure 4.2** Modèle de données du cadre d'étude

Dans ce schéma nous avons une ébauche de représentation des objets et des métadonnées que nous avons dans notre cadre applicatif.



**Figure 4.3** Répertoire des albums photos à archiver

#### 4.4.2 Scénario de l'expérimentation

##### ➤ **Stockage d'un OM**

Nous allons sélectionner un album existant et stocker l'ensemble des photos de l'album avec notre application.

##### ➤ **Annotation d'un OM**

Pour chaque photo dans l'album nous ajoutons les informations sur la photo (les métadonnées associées). Ces métadonnées seront liées à la photo et seront stockées dans le référentiel.

##### ➤ **Recherche d'un OM**

Nous allons tester la fonctionnalité de recherche en recherchant les photos qui sont stockées dans le référentiel à travers les métadonnées qui leur sont associées.

##### ➤ **Archivage d'un OM**

Les photos que nous stockons vont être archivées en les compressant avec les métadonnées associées dans le référentiel.

#### 4.4.3 Exécution détaillée du scénario

##### ➤ Stockage d'un OM

On choisit un album photo pour exécuter notre scénario de test (Cérémonie). Dans cet album, nous sélectionnons chaque à archiver et la stocker dans le référentiel avec notre application.



Figure 4.4 Stockage d'une image dans le référentiel

Après stockage, les métadonnées descriptives et techniques sont ajoutés pour l'OM.



Figure 4.5 Métadonnées après stockage de l'objet

### ➤ Annotation d'un OM

L'utilisateur sélectionne l'OM à annoter et sélectionne le type de métadonnées, ajoute une propriété avec une valeur. L'utilisateur peut ajouter d'autres propriétés avec pour chaque propriété une valeur.

|        |        |                 |                          |
|--------|--------|-----------------|--------------------------|
| Delete | Update | jcr:mimeType    | application/octet-stream |
| Delete | Update | jcr:mixinTypes  | mix:lockable             |
| Delete | Update | jcr:primaryType | slings:Folder            |

| Type métadonnées | Propriété | Valeur                   | Type propriété |         |
|------------------|-----------|--------------------------|----------------|---------|
| Descriptive      | Lieu      | Méridien President, Daka | Name           | Ajouter |

Close

**Figure 4.6** Annotation de la photo

Les métadonnées sont ainsi ajoutées à la photo et sont stockées dans le référentiel et sont classées sous le type (descriptive, technique ou administrative) sélectionné par l'utilisateur.

| content/resources/images/photo_de_la_promo |        |               |   |  |
|--|--------|---------------|---|--|
| Delete                                     | Update | Lieu          | Méridien President, Dakar                     |  |
| Delete                                     | Update | comment       | Photo à archiver dans SARDOM.                 |  |
| Delete                                     | Update | createDate    | 8/17/11                                       |  |
| Delete                                     | Update | description   | Photo de la promotion DIC Informatique de l'E |  |
| Delete                                     | Update | jcr:created   | Sun Sep 04 2011 18:37:53 GMT-0400             |  |
| Delete                                     | Update | jcr:createdBy | admin   |  |

**Figure 4.7** Métadonnées après ajout

### ➤ Recherche d'un OM

La recherche se fait à travers les métadonnées précédemment ajoutées à l'OM. L'utilisateur sélectionne les types de métadonnées et saisit pour chaque type la valeur d'une propriété. L'utilisateur peut utiliser plusieurs propriétés pour un type ou combiner plusieurs types avec pour chaque type une propriété.

**Figure 4.8** Formulaire de recherche d'une image

L'utilisateur valide la recherche avec le bouton rechercher. Le système parcourt alors pour chaque type sélection les sous nœuds et compare les propriétés de chaque nœud avec les critères de sélection.

| name              | lieu                      | jcr:created                       | jcr:createdBy |
|-------------------|---------------------------|-----------------------------------|---------------|
| Photo de la promo | Méridien President, Dakar | Sun Sep 04 2011 18:37:53 GMT-0400 | admin         |

**Figure 4.9** Affichage des résultats de recherche

L'exécution du scénario donne une liste de résultats qui satisfont les critères de recherche est présentée à l'utilisateur. L'utilisateur peut sélectionner l'OM recherché pour consultation.

### ➤ Archivage d'un OM

L'utilisateur cherche dans le référentiel l'OM à archiver et le système affiche un ou plusieurs résultats qui satisfont les critères de recherche.

L'utilisateur sélectionne alors l'OM qu'il souhaite archiver et valide l'opération d'archivage.

Le système ajoute des métadonnées sur l'archivage de l'objet (archive\_id, archive\_date, archive\_profile, archive\_message) et créer une archive composée de la photo et des métadonnées associées.

SARDOM - ARCHIVER UN OBJET MÉDIATIQUE

content/resources/images/photo\_de\_la\_promo

|        |        |                |   |
|--------|--------|----------------|---|
| Delete | Update | Lieu           | Méridien President, Dakar                     |
| Delete | Update | comment        | Photo à archiver dans SARDOM.                 |
| Delete | Update | createDate     | 8/17/11                                       |
| Delete | Update | description    | Photo de la promotion DIC Informatique de l'E |
| Delete | Update | jr:created     | Sun Sep 04 2011 18:37:53 GMT-0400             |
| Delete | Update | jr:createdBy   | admin   |
| Delete | Update | jr:primaryType | sling:Folder                                  |
| Delete | Update | name           | Photo de la promo                             |

Archiver Annuler

**Figure 4.10** Archivage d'une image dans le référentiel

L'OM archivé peut, dans le futur être désarchivé pour une consultation ou pour d'autres traitements (ajout de métadonnées ou suppression).

#### 4.5 CONCLUSION

Dans cette partie nous avons fait d'abord la présentation des outils et technologies utilisés pour l'implémentation du prototype basée sur notre approche. Nous avons aussi fait la présentation de la motivation du choix de ces outils. Ensuite nous avons présenté l'implémentation du prototype que nous avons développé. Ainsi nous avons implémenté différentes fonctionnalités permettant de stocker, d'annoter, de rechercher et d'archiver un objet multimédia suivant le modèle de données que nous avons défini ou d'autres modèles. Les objets seront ainsi stockés dans un référentiel de contenu Apache-Sling basée sur Java Content Repository avec son implémentation Apache-Jackrabbit. Enfin nous avons fait la présentation de l'expérimentation de l'outil dans le cadre d'un projet de gestion d'archives de photos personnelles.

## CHAPITRE V

### CONCLUSION

Dans ce document, nous avons présenté la problématique de la gestion et de l'archivage d'OM. Celle-ci est liée à l'expansion rapide de ces derniers, aux caractéristiques des OM et aux problèmes d'interopérabilité des standards de représentation qui existent dans ce domaine. L'état de l'art nous a permis d'appréhender les problématiques majeures dans le domaine. Nous avons ainsi proposé une approche de gestion et d'archivage d'OM et développé un prototype se basant sur celle-ci. Dans ce travail, l'approche proposée est centrée sur les processus liés aux OM et sur les métadonnées associées, et permet de faciliter l'archivage d'OM. En effet, cette approche prend en compte l'ensemble des étapes de la gestion d'un OM c'est-à-dire de la création à l'archivage de l'objet; cela permet d'enrichir l'OM de métadonnées à chaque étape du processus.

En outre, notre approche est indépendante de la technologie choisie. Elle est basée sur un modèle de métadonnées qui supporte plusieurs types d'OM et permet de définir plusieurs types de métadonnées pour chaque type d'objet médiatique. Cela offre une flexibilité à notre approche et favorise en effet l'interopérabilité de celle-ci par rapport aux autres systèmes. Par conséquent, nous avons conçu et développé un prototype qui se base sur cette approche et qui offre les fonctionnalités de stockage, d'annotation, de recherche et d'archivage d'OM.

## 5.1 BILAN

Nous avons ainsi réalisé, à quel point, l'importance d'un processus bien élaboré ainsi que une bonne gestion des métadonnées afin de faciliter l'archivage et la gestion des OM. En effet, cela permet de rendre les OM beaucoup plus explicites et facilite à l'utilisateur les possibilités d'organisation (en classifiant les métadonnées suivant leurs types) et de recherche avec des requêtes basées sur le contexte, le contenu et des résultats plus précis.

L'avantage de notre approche est qu'elle offre la possibilité de rajouter d'autres types de métadonnées à un modèle existant et d'importer d'autres modèles de métadonnées. Aussi, pour archiver un objet, notre approche permet de créer un archive multimédia composé à la fois de l'OM et des métadonnées associées. Cela permet de préserver l'objet à long terme et de pouvoir le récupérer de façon simple.

Pour une évaluation, le prototype développé a été expérimenté avec succès dans le cadre d'un projet de gestion d'archive d'album photos personnelles. Une autre expérimentation est prévue dans le cadre du projet ePortfolio de la Maison des technologies de formation et d'apprentissage Roland-Giguère (MATI). Les résultats d'expérimentation ont montré la flexibilité de notre modèle de données et l'importance des métadonnées dans chaque étape du processus de gestion d'un OM. L'outil développé permet une gestion avancée des documents et des ressources médiatiques. Elle utilise pour cela un API standard basé sur les principes d'architecture « Representational State Transfer » REST. Elle offre une interopérabilité avec différentes applications de gestion d'OM (Digital Asset Management). Cet outil est simple d'utilisation et permet de partager des ressources entre utilisateurs éloignés du point de vue géographique.

## 5.2 PERSPECTIVES

Bien que notre approche résolve un ensemble de problèmes liés à la gestion et à l'archivage d'objets multimédia, il serait intéressant d'ajouter un module permettant d'exporter les métadonnées vers d'autres modèles ou d'autres formats de données (XML par exemple). Comme perspective aussi, nous comptons expérimenter le prototype dans le cadre du projet ePortfolio de la Maison des technologies de formation et d'apprentissage Roland-Giguère (MATI).

Enfin dans le module de recherche nous utilisons XPath comme langage de requête, nous souhaitons ainsi intégrer différents langages de requêtes (SQL, XPath et XQuery) pour pouvoir comparer les performances entre ces différents langages car l'outil que nous utilisons Apache Sling supporte ces différents langages.



## BIBLIOGRAPHIE

[Amous, I. et al., 2005] Amous I., Jedidi A., et Sèdes F.. 2005. A Contribution to Multimedia Document Modeling and Querying. In *Multimedia Tools and Applications*, 25(3): *SpringerLink* Vol 25 p.(391-404). 2005.

[Bloehdorn S. et al., 2005] Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y.S., Handschuh, S., Kompatsiaris, I., Staab, S., and Strintzis, M.G. Semantic Annotation of Images and Videos for Multimedia Analysis. In *Proceedings of ESWC. 2005*, 592-607.

[Brugman, H. et al., 2004] Brugman, H., Russel, A. (2004). Annotating Multimedia/ Multi-modal resources with ELAN. In: *Proceedings of LREC 2004, Fourth International Conference on Language Resources and Evaluation*.

[Brut, M. et al., 2009] Brut, M.; Laborie, S.; Manzat, A.-M.; Sedes, F. (2009) ; ,A Generic Metadata Framework for the Indexation and the Management of Distributed Multimedia Contents, *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on* , vol., no., pp.1-5, 20-23 Dec. 2009 doi: 10.1109/NTMS.2009.5384702 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5384702&isnumber=5384652>

[Elliott, B. et al., 2008] Elliott, B. and Özsoyoglu, Z. M. (2008). Annotation suggestion and search for personal multimedia objects on the web. In *Proceedings of the 2008 international Conference on Content-Based Image and Video Retrieval (Niagara Falls, Canada, July 07 - 09, 2008)*. CIVR '08. ACM, New York, NY, 75-84. DOI=<http://doi.acm.org/10.1145/1386352.1386366>

[Goularte, R. et al., 2004] Goularte, R., Camacho-Guerrero, J.A., Inácio, V.R. Jr., Cattelan, R.G., Pimentel, M.G.C. (2004): M4Note: a multimodal tool for multimedia annotation. In: *Proceedings of the Latin-American Web Congress*, pp. 142–149.

[Hardman L. et al., 2008] Hardman L., Obrenović Ž., Nack F., Kerhervé B., Piersol K., (2008). Canonical processes of semantically annotated media production. *Multimedia Systems* 14, 6 (July 2008), 327-340 Springer-Verlag 2008 DOI=<http://dx.doi.org/10.1007/s00530-008-0134-0>

[Laborie, S. et al., 2009] S. Laborie, A.-M. Manzat and F. Sèdes (2009): Managing and querying efficiently distributed semantic multimedia metadata collections, In IEEE MultiMedia special issue on multimedia-metadata and semantic management, IEEE, 2009 (to appear).

[Ma, Y., Fang et al., 2008] Ma, Y., Fang, Z., Liu, J., Wang, T., Xu, P., and Zhao, L. (2008). A Content-Based Multimedia Retrieval System Base on MPEG-7 Metadata Schema. In Proceedings of the Fifth international Conference on information Technology: New Generations (April 07 - 09, 2008). ITNG. IEEE Computer Society, Washington, DC, 1200-1201. DOI= <http://dx.doi.org/10.1109/ITNG.2008.211>

[Nack, F. et al., 2005] Nack, F., Ossenbruggen, J. v., and Hardman, L. (2005). That Obscure Object of Desire: Multimedia Metadata on the Web, Part 2. IEEE MultiMedia 12, 1 (Jan. 2005), 54-63. DOI= <http://dx.doi.org/10.1109/MMUL.2005.12>

[Ossenbruggen, J. v. et al., 2004] Ossenbruggen, J. v., Nack, F., and Hardman, L. (2004). That Obscure Object of Desire: Multimedia Metadata on the Web, Part 1. IEEE MultiMedia 11, 4 (Oct. 2004), 38-48. DOI= <http://dx.doi.org/10.1109/MMUL.2004.36>

[Pereira, F. et al., 2008] Pereira, F.; Vetro, A.; Sikora, T. (2008); Multimedia Retrieval and Delivery: Essential Metadata Challenges and Standards. *Proceedings of the IEEE*, vol.96, no.4, pp.721-744, April 2008. doi:10.1109/JPROC.2008.916384 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4472087&isnumber=4472076>

[Reveiu, A., et al., 2008] Reveiu, A., Dardala, M., Smeureanu, I.: A MPEG-21 Based Architecture for Data Visualization in Multimedia Web Applications. In: International Conference Visualisation, pp. 84–89 (2008). DOI=10.1109/VIS.2008.13 <http://dx.doi.org/10.1109/VIS.2008.13>

[Saathoff, C., et al. 2006] Saathoff, C., Petridis, K., Anastasopoulos, D., Timmermann, N., Kompatsiaris, I. and Staab, S. (2006). M-OntoMat-Annotizer: Linking Ontologies with Multimedia Low-Level Features for Automatic Image Annotation, *3rd European Semantic Web Conference*, ESWC 2006, Budva, Montenegro, June 2006.

[Sloetjes, H. et al., 2008] Sloetjes, H., & Wittenburg, P. (2008). Annotation by category - ELAN and ISO DCR. In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

[Smith, J.R. et al., 2006] Smith, J.R.; Schirling, P. (2006); Metadata standards roundup, *Multimedia, IEEE*, vol.13, no.2, pp. 84- 88, April-June 2006 doi: 10.1109/MMUL.2006.34.URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1621040&isnumber=33952>

[Stamou, G et al., 2006] Stamou, G., Ossenbruggen, J. v., Pan, J. Z., and Schreiber, G. (2006). Multimedia Annotations on the Semantic Web. *IEEE MultiMedia* 13, 1 (Jan. 2006), 86-90. DOI= <http://dx.doi.org/10.1109/MMUL.2006.15>

[Zhou, T. H. et al., 2008] Zhou, T. H., Heo, B. M., Wang, L., Lee, Y. K., Chai, D. J., and Ryu, K. H. (2008). A Prototype of Multimedia Metadata Management System for Supporting the Integration of Heterogeneous Sources. In Proceedings of the 4th international Conference on intelligent Computing: *Advanced intelligent Computing theories and Applications - with Aspects of theoretical and Methodological Issues* (Shanghai, China, September 15 - 18, 2008). D. Huang, D. C. Wunsch, D. S. Levine, and K. Jo, Eds. Lecture Notes In Computer Science, vol. 5226. Springer-Verlag, Berlin, Heidelberg, 1095-1102. DOI= [http://dx.doi.org/10.1007/978-3-540-87442-3\\_135](http://dx.doi.org/10.1007/978-3-540-87442-3_135)

[Dublin Core, DCES] <http://dublincore.org/documents/dces/>

[Dublin Core, DCMI] <http://dublincore.org/documents/dcmi-terms/>

[LINDO] <http://www.irit.fr/LINDO/>

[JSR 170] <http://jcp.org/en/jsr/detail?id=170>

[JACKRABBIT] <http://jackrabbit.apache.org/>

[JACKRABBIT, ARC] <http://jackrabbit.apache.org/jackrabbit-architecture.html>

[JACKRABBIT, NT] <http://jackrabbit.apache.org/node-types.html>

[SLING] <http://sling.apache.org/site/index.html>

[SLING, RES] <http://sling.apache.org/site/resources.html>

[SLING, ARC] <http://sling.apache.org/site/architecture.html>

[TIKA] <http://tika.apache.org/>

[TIKA, FORMATS] <http://tika.apache.org/0.9/formats.html>

[TIKA, PARSER] <http://tika.apache.org/0.9/parser.html>

[YouTube] [http://www.youtube.com/t/press\\_timeline](http://www.youtube.com/t/press_timeline)

--

-

-

---

-

.

## APPENDICE A

### REPRÉSENTATION DU MODÈLE EN XML

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <mediaType name="image/jpeg">
    <mediaObjectName>"coucherSoleil.jpg" description="Coucher de soleil "
      metadonneesTypes>
      <metadonneeType name="Métadonnées Descriptives">
        <propriete name="titre">
          <valeur>"Coucher de soleil"</valeur>
        </propriete>
        <type>"String" type="string"/>
        <propriete name="auteur">
          <valeur>"Mama AMAR"</valeur>
        </propriete>
        <type>"String" type="string"/>
        <propriete name="description">
          <valeur>"Coucher de soleil à Montréal"</valeur>
        </propriete>
        <type>"String" type="string"/>
      </metadonneeType>
      <metadonneeType name="Métadonnées Techniques">
        <propriete name="taille">
          <valeur>"383,7KO"</valeur>
        </propriete>
        <type>"String" type="string"/>
        <propriete name="format">
          <valeur>"JPEG"</valeur>
        </propriete>
        <type>"String" type="string"/>
        <propriete name="dimension">
          <valeur>"630px x 488px"</valeur>
        </propriete>
        <type>"String" type="string"/>
      </metadonneeType>
      <metadonneeType name="Métadonnées Administratives">
        <propriete name="dateModification">
          <valeur>"2011-06-11"</valeur>
        </propriete>
        <type>"Date" type="date"/>
        <propriete name="archive">
          <valeur>"faux"</valeur>
        </propriete>
        <type>"Boolean" type="boolean"/>
      </metadonneeType>
    </metadonneesTypes>
  </mediaObjet>
</mediaType>
</root>
```



## APPENDICE B

### REPRÉSENTATION DU STANDARD DUBLIN CORE DANS NOTRE MODÈLE

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <metadonneeTypes nom= >
    <metadonneeType nom="DublinCore" descriptio="Éléments du standard Dublin Core" >
      <propriete name="dc:collaborateur">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:couverture">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:createur">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:date">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:description">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:format">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:identificateur">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:langage">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:editeur">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:relation">
        <valeur></valeur>
        <type></type>
      </propriete>
      <propriete name="dc:droits">
        <valeur></valeur>
        <type></type>
      </propriete>
    </metadonneeType>
  </metadonneeTypes>
</root>
```

```
<propriete name="dc:source">
  <valeur></valeur>
  <type></type>
</propriete>
<propriete name="dc:sujet">
  <valeur></valeur>
  <type></type>
</propriete>
<propriete name="dc:titre">
  <valeur></valeur>
  <type></type>
</propriete>
<propriete name="dc:type">
  <valeur></valeur>
  <type></type>
</propriete>
</metadonneeType>
</metadonneeTypes>
</root>
```

## APPENDICE C

### PROBLÈMES DES MÉTADONNÉES : MULTIMÉDIA VS TEXTE

Dans [Ossenbruggen, J. v. et al., 2004], les auteurs ont fait une présentation des problèmes liés à la gestion des métadonnées multimédia. Pour se faire ils ont fait une comparaison entre le format texte et le format multimédia. Parmi ces différents problèmes on peut noter :

#### ➤ **Le coût**

La génération de métadonnées est souvent coûteuse en temps car la plupart des applications actuelles nécessitent des métadonnées de haut-niveau et de ce fait nécessitent un travail supplémentaire des hommes.

#### ➤ **La subjectivité**

Les métadonnées peuvent être subjectives si elles proviennent de l'annotation des humains. En effet, un document numérique peut être interprété de façon différente par les humains suivant le contexte.

#### ➤ **Le caractère restrictif**

Certains schémas de métadonnées peuvent fournir des informations qui peuvent être partagées par plusieurs applications. Cependant, les annotations faites par les humains sont souvent restrictives et ne peuvent être réutilisées.

➤ **La longévité**

Il est parfois difficile de définir des schémas de métadonnées applicables à court et à long terme. Cela nécessite des outils flexibles pour l'extension, la modification et le suivi des versions etc.

➤ **La confidentialité**

Les métadonnées peuvent fournir des informations privées ou sensibles à la sécurité nécessitant des soins particuliers.

➤ **La standardisation**

Les outils de gestion des métadonnées ne sont pas interopérables entre eux. Afin de résoudre ce problème, il est nécessaire de définir des standards de métadonnées aussi bien sur le plan syntaxique, afin d'assurer qu'un outil peut analyser les autres formats, que sur le plan sémantique, pour s'assurer que les outils peuvent comprendre les concepts partagés.

D'autres problèmes relatifs à la gestion des métadonnées multimédia existent. D'abord l'association des annotations avec des OM. La gestion du workflow dans le processus de production multimédia peut être très complexe car chaque étape du processus peut engendrer des métadonnées de haut-niveau. Enfin, la complexité du processus de gestion des objets multimédia rend la gestion des droits numériques plus complexe car plusieurs intervenants peuvent avoir leurs droits sur un seul objet multimédia.