

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PROJET DE 15 CRÉDITS PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN TECHNOLOGIE DE L'INFORMATION
M.Sc.

PAR
HORIA ILIE

Identification et annotation des images représentatives des
séquences d'une vidéo

MONTREAL, 2005

CE PROJET A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

1. M. Alain April, Professeur en Génie Logiciel, directeur de projet
- Département de génie logiciel et des TI, École de technologie supérieure.

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT JURY ET
PUBLIC
EN 2005
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

SOMMAIRE

Le projet vise à réaliser une application permettant le repérage basé sur contenu pour des images tirées d'un flux vidéo, afin de détecter les changements de scènes. On utilise la technologie Intermedia d'Oracle qui permet l'extraction et l'utilisation des descripteurs de bas niveau pour des objets multimédia. Les descriptions des images identifiant un changement de scène seront sauvegardées en format XML compatible au standard MPEG-7 pour les images. Les modules développés seront intégrés dans l'application déjà développé par l'équipe du professeur Harald Kosch (université Klagenfurt - Autriche), en ajoutant des nouvelles fonctionnalités au prototype existant. Les grandes étapes seront a chargement de la vidéo dans la BDMM, l'extraction des images à partir de la vidéo et leurs stockage dans la BDMM, l'analyse des images pour identifier celles indiquent un changement de scène. La méthode de travail utilisé sera l'analyse des distances entre deux images successives, basée sur des attributs de bas niveau (couleur, texture, forme). Les éléments identifiant les images résultats à la suite de cette analyse feront le sujet d'une sauvegarde en format XML, conforme au standard MPEG-7.

TABLE DES MATIÈRES

SOMMAIRE	3
AVANT PROPOS ET REMERCIEMENTS	Error! Bookmark not defined.
LISTE DES TABLEAUX.....	6
LISTE DES FIGURES	7
LISTE DES FIGURES	7
LISTE DES ABREVIATIONS ET DES SIGLES	9
LISTE DES ABREVIATIONS ET DES SIGLES	9
1. INTRODUCTION	12
1.1 Contexte.....	12
1.2 Motivation de recherche.....	13
1.3 Méthodologie de recherche	17
1.4 Sommaire.....	20
2. CHAPITRE 1 – recherche bibliographique	20
2.1 Objectifs et stratégie de la recherche bibliographique	20
Objectif.....	20
Choix bases de recherche et stratégie	21
2.2. État de l’art sur le repérage de l’information multimédia	22
2.2.1 BDMM	23
2.2.2 Vidéo	27
2.2.3 MPEG-7	32
2.2.4 Annotation	41
2.2.5 CBIR	51
2.3 Analyse du prototype proposé par l’université d’Autriche	58
2.3.1 références provenant d’étudiant de M. Harald	etc. 58
2.3.2 Commentaires sur l’analyse (actuellement en Annexe 1).....	58
2.4 Sommaire.....	58
3. CHAPITRE 2 - Exécution	60
3.1 Introduction	60
3.2 Préparation	60
3.3 Conception.....	70
3.4 Programmation et tests.....	74
3.5 Chargement des données et essais	77
3.6 Sommaire.....	87
4. CHAPITRE 3 – Interprétation	88
4.1 Introduction	88
4.2 Contexte d’interprétation.....	88
4.3 Résultats obtenus	89
4.4 Limites identifiés, améliorations possibles	96
4.4 Travaux futurs	98
4.5 Sommaire.....	99
5. CONCLUSION.....	100
Annexes	102
ANNEXE 1 - Analyse application Autriche – approche technique	103
ANNEXE 2 – Module MEI – diagrammes UML.....	115
ANNEXE 3 –Module MTI – diagrammes UML.....	119
ANNEXE 4 –Module MV – diagrammes UML.....	124
ANNEXE 5 – Exigences environnement logiciel.....	130

ANNEXE 6 – Diagrammes séquences de travail programmation.....	132
ANNEXE 7 – Vues des scénarios et détails code Java	135
ANNEXE 8 - Images tirées de la vidéo	141
ANNEXE 9 – Résultats interrogations BD (Oracle SQL Plus)	143
BIBLIOGRAPHIE	154

LISTE DES TABLEAUX

Tableau1 – cadre Basili modifié.....	20
Tableau2 – choix et justification bases de recherche documentaire	22
Tableau3 – groupement références documentaires	23
Tableau4 – exigences projet.....	64
Tableau5 – plan de travail (cédule).....	69
Tableau 6 – résultats expérimentaux.....	94

LISTE DES FIGURES

Figure 1 : Vue d'ensemble de la recherche	14
Figure 2 : Structure d'un document vidéo	29
Figure 3 : Les trois couches du sous-réseau DN (<i>document network</i>)	30
Figure 4 : Structure du sous-réseau QN (<i>query network</i>)	30
Figure 5: Exemple utilisation réseau inférence	32
Figure 6 : Architecture de MDC(<i>multimedia data cartridge</i>).....	32
Figure 7 : Intégration contextuelle de MDC (<i>multimedia data cartridge</i>).....	34
Figure 8 : Architecture de ERIC7 – <i>Computer Reserch Institut Montreal</i>	36
Figure 9 : Organisation MPEG-7	38
Figure 22: Hiérarchie MPEG-7	40
Figure 10 : Architecture de IFinder.....	43
Figure 11 : Système de recherche multimédia (annotation mixte).....	45
Figure 12 : Exemple fichier XML (application M4Note).....	46
Figure 13 : Exemple annotation encre électronique (application M4Note)	48
Figure 14 : Schéma de description multimédia de MPEG-7	49
Figure 15 : Exemple utilisation TextAnnotation (MPEG-7).....	50
Figure 16 : Exemple utilisation métadonnées (interrogations sur la BDMM)	50
Figure 17 : approche système SIRSALE [4]	51
Figure 18 : Systèmes de repérage basés sur le contenu (WEB)	53
Figure 19 : Exemples des attributs utilisés pour la recherche des images	54
Figure 20 : Exemple système dual de repérage MM	56
Figure 21 : Liaison entre les objets ORDSYS.ORDImage d'Oracle 10g et Ordimage de Java.[20].....	63
Figure 22 : Modèle en V du cycle de développement [22]	69
Figure 23: Conception générale système (Groupement fonctionnalités).....	70
Figure 24 : Conception détaillée MEI (module extraction images).....	71
Figure 25 : Conception détaillée MTI (module traitement images)	72
Figure 26 : Conception détaillée MV (module visualisation)	73
Figure 27 : Séquence de travail pour la réalisation des modules logiciels.....	76
Figure 28 : Ecran initial (valeurs par défaut)	77
Figure 29 : Message erreur paramétrage MEI	78
Figure 30 : dialogue chargement fichier vidéo	78
Figure 31 : erreur démarrage lecteur vidéo.....	78
Figure 32 : déroulement extraction images.....	79
Figure 33 : Images extraites de la vidéo	80
Figure 34 : Écran démarrage MTI	81
Figure 35 : Déroulement module MTI	81
Figure 36 : Exemple fichier MPEG-7 (<i>DominantColor</i>)	83
Figure 37 : Messages analyse similarité et nombre finale insertions dans la BD	84
Figure 38 : Interface graphique MV – connexion réussie à la BD.....	85
Figure 39 : Interface graphique MV – visualisation comparative images.....	86
Figure 40 : Interface graphique MV – visualisation fichier MPEG-7.....	87
Figure 41: Bruit et silence	91

Figure 42: Les images début des scènes	93
Figure 43: Résultats expérimentaux	94
Figure 44: transaction lente.....	97
Figure 45: changement très localisé de luminosité	97

LISTE DES ABREVIATIONS ET DES SIGLES

ACM	Association for Computer Machinery – group de recherche dans le domaine de l'informatique et la construction des ordinateurs
API	Application Programming Interface
ASR	Automatic speech recognition
BD	Base de données
BDMM	Base de données multimédia
BDMM	Base de données multimédia
BIM	Binary Format for MPEG-7
C++	Langage de programmation orienté objet (extension du langage C)
CBIR	Content-based image retrieval
CBR	Content based retrieval
CC/PP	Composite Capabilities / Preferences profile
CCD	Common Color Dataset
CLD	Color layout Descriptor
CORBA	Common Object Request Broker Architecture – architecture et spécifications pour la création, la distribution et l'administration des applications dans un réseau informatique
CRIM	Computer Research Institute of Montreal
CSD	Color Structure Descriptor
dbXML	XML database
DCD	Dominant Color Descriptor
DDDC	Dozen Dimensional Digital Content
DDL	Description Definition language
DN	Document network
DOM	Document Object Model – API proposée par W3C (World WideWeb Consortium), et basée sur un modèle objets (structure arborescente)
Eclipse	Plateforme logiciel pour la création des applications Java
EXIF	Exchangeable Image File Format
FSA	Finite State automata
GPS	Global Positioning System – technologie permettant de calculer la position sur la terre en utilisant les satellites de communication
GUI	Graphic user interface
IDL	Interface Definition Standard
IEEE	Institute of Electrical & Electronic Engineers – organisme reconnu mondialement pour le développement sur les standards de transmission de l'information.
IETF	Internet Engineering Task Force – responsabilité sur des aspects techniques de WEB (protocole TCPIP, RDF etc.)
INSPEC	La plus grande ressource documentaire en format électronique pour les domaines de la physique, électronique, informatique, la technologie de l'information.

ISIWEB	Base documentaire très utile pour le repérage des articles qui citent des articles déjà trouvés
ISO/IEC	The International Standardization Organization and International Electrotechnical Commission Committee
Java	Langage de programmation orienté objet (Sun Microsystems)
JAXB	Java Architecture for XML Binding
JAXP	Java API for XML Parsing, librairie proposée par SUN Micro System pour traiter les documents XML
JMF	Java Media Framework – paquet optionnel de Java permettant l'addition du multimédia aux applications java.
JSApi	Java Speech Api
JVM	Java Virtual Machine
LDD	Langage de description des données
MADIS	MPEG-7 Audio-visual Documentation Indexing System
MARC	Machine Readable Cataloging – standard pour la représentation et la communication des ressources bibliographiques
MCF	Meta content framework
MDC	Multimedia data cartridge
MDM	Multimedia data model
MDS	Multimedia description schemes
MIF	Multimedia indexing framework
MMDBS	Multimedia database system
MMDS	Multimedia distributed systems
MPEG	Moving Pictures Experts Group - Format standard de compression audio et vidéo
MySQL	Système de gestion de base de données qui utilise SQL (open source)
OO	Object Oriented
PICS	Platform for Internet Content Selection – Spécification de W3C pour l'ajout des métadonnées
PL/SQL	Procedural Language/Structured Query Language – langage de requêtes Oracle
QBD	Query by draw
QBE	Query by exemple
QN	Query network
RDF	Resource Description Framework, (perspectives pour le WEB sémantique)
RGB	Matrice Red-green-Blue
RMI	Remote Method Invocation (RMI) - technologie permettant aux applications Java de communiquer sur un réseau.
SAX	Simple API for XML – API proposée par le groupe XML.org, et basée sur un modèle de traitement séquentielle du fichier XML
SDM	Schémas de description multimédia
SGBD	Système de Gestion de Bases de Données
SGBDR	Acronyme pour Système de gestion de bases de données relationnelles
SMMD	Systèmes multimédia distribués
SMPTE	Society of Motion Picture and Television Engineers – organisation qui a établi le standard pour la vidéo, dans un format de type : hh:mm:ss:ff: heures, minutes,

	secondes, frames
SQL	Structured Query Language – langage standardisé pour l’interrogation et la mise à jour des bases de données relationnelles.
SQL/MM	Extension de SQL qui supporte la multimédia
SVG	Scalable Vector Graphics
TBIR	Texte based image retrieval
UML	Unified Modeling Language - méthodologie de modélisation orientée objet
URI	Uniform Resource Identifier – nom générique quant à l’identification des ressources sur Internet
URL	Uniform Resource Locator – L’adresse identifie la location d’une page WEB
URN	Uniform Resource Name
VIRS	Video Image retrieval system
W3C	World Wide Web Consortium – organism international qui travaille sur les protocoles du WEB
WEB	World Wide Web
XM	eXperimental Model
XML	eXtensible Markup Language
XML	Extensible Markup Language
XSL	Extensible stylesheet language
XSU	XML SQL Utility

1. INTRODUCTION

1.1 Contexte

La croissance sans précédent du contenu multimédia dans les documents, alimentée par le développement rapide de l'Internet occupe sans doute un des premières positions parmi les plus marquants développements technologiques des dernières années. La croissance des aspects multimédia dans l'information actuelle est certainement un avantage mais soulève toutefois de nombreux défis de nature pluri ou interdisciplinaires : structure (compression, normalisation de la représentation, etc.), stockage, transfert, facilité d'utilisation (accès, recherche et repérage, indexation etc.).

Un des problèmes actuels que les chercheurs tentent de résoudre est celui du repérage rapide et précis de l'information multimédia.

Les systèmes actuels de recherche et localisation de l'information sont généralement basés sur l'analyse et l'interprétation du texte. Cependant, de plus en plus c'est le contenu visuel ou sonore celui qui caractérise l'information. Les efforts des chercheurs se sont concrétisés, et dans plusieurs domaines des succès ont été présentés [8] :

- Systèmes de repérage basés sur le contenu pour dépister des images sous l'incidence de la loi de la propriété intellectuelle
- Dépistage automatiquement des images ayant un contenu inadéquat
- Repérage des images basées sur le contenu appliqué dans des domaines comme la loi et la prévention du crime. (reconnaissance des empreintes ou du visage, analyse similarité ADN, impression des marques dans le sol, etc.)
- Systèmes d'aide touristique, réseaux de transport terrestre, sur l'eau ou aérien.
- Éducation et entraînement : recherche selon des sujet particuliers, très utilisé dans le monde pédagogique
- Mode, architecture, design, médecine, recherche historique, art, Météorologie, astrologie, sciences en général : (systèmes experts, systèmes à base de connaissances)

Pour aider la recherche basée sur le contenu multimédia, des normes ont été introduites afin de décrire le contenu plus précisément. Au cours des dernières années plusieurs normes ont été adoptées, par exemple [8] :

- RDF (*resource description framework*) – proposée par *World Wide Web Consortium W3C*, (les schémas RDF utilisent comme langage XML)
- Dublin Core – métadonnées (auteur, titre, mots clés sur le sujet, le type du document, le format du document)
- MPEG-7 – considéré présentement comme la référence de base pour la description du contenu multimédia.

La norme ISO MPEG-7, réalisée par Moving Picture Experts Group, présente un ensemble d'outils de description permettant de définir et d'identifier le contenu de données multimédia pour en faciliter l'indexation et la recherche subséquente. Pour les images, par exemple, les outils de description fournis par MPEG-7 sont capables de préciser un grand nombre d'informations regroupées en cinq catégories :

- informations de bas niveau (couleur, texture, forme)
- informations sur la source (titre, auteur, date, format, etc.)
- informations sur les conditions de l'utilisation (droit d'auteur, etc.)
- caractéristiques techniques (format, codage, etc.)
- structure spatiale et temporelle (positions, déplacements, etc.)

La norme MPEG-7 offre un cadre de développement pour des applications basées sur l'analyse du contenu de l'information multimédia, permettant une localisation précise, objective et rapide de l'information voulue.

1.2 Motivation de recherche

Le présent projet a comme principale motivation l'étude Identification et annotation des images représentatives des séquences d'une vidéo. Plus précisément, il s'agira d'une analyse des images tirées de vidéo afin de faire l'annotation des descripteurs inter-séquences.

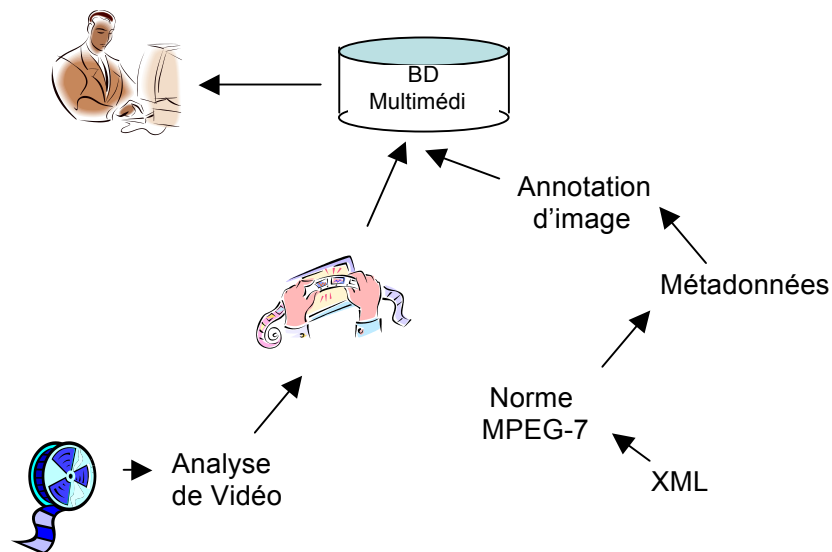


Figure 1 : Vue d'ensemble de la recherche

La figure 1 décrit sommairement l'objectif de cette recherche. La vidéo (stocké dans une BDMM Oracle) sera utilisée comme entrée pour l'application d'extraction des images. Les images seront déposées en la BDMM et, un module logiciel s'occupera de les analyser et d'identifier les changements de scène. Pour ces images, représentatives pour la séquence vidéo, on va créer des fichiers XML, conforme au standard MPEG-7, qui seront également déposées dans la BDMM.

Harald Kosch ([15]) groupe en deux grandes catégories les principales taches concernant l'avenir du multimédia : 1) comment accéder à l'information désirée et 2) comment échanger effectivement cette information. N'importe quel modèle de données pourra être utilisé localement et peu importe le mécanisme de stockage local il sera possible de retracer et transférer, en un minimum de temps, l'information multimédia.

La clé du processus de repérage, très bien identifié par Kosch ([15]) et Dunckley ([16]), est la similarité entre deux objets (le modèle et la cible). Mais, le problème de la similarité est complexe avec des objets multimédia, nécessitant parfois plusieurs approches simultanées. Dunckley ([16]) décrit deux approches populaires pour interroger les bases de données multimédia. La première est liée au "quoi" et la deuxième traite "le comment". En effet, Dunckley propose pour le « quoi » trois niveaux de complexité :

- 1) repérage des attributs de base (couleur, forme, texture, localisation spatiale, mouvement).

- 2) Repérage des attributs logiques directement liés à l'identité d'objet dans le média.
- 3) Repérage des attributs abstraits associés à la nature ou destination de l'objet

Les deux derniers niveaux de complexités sont considérés comme le noyau du repérage sémantique du média, présentement en état embryonnaire. Quant "au comment", l'auteur présente une classification selon les méthodes d'indexation et de recherche de l'information :

- systèmes basés sur les attributs (*attribute-based systems*) – utilisent des attributs structurés dans la façon des SGBD traditionnelles, mais on travaille avec des données ayant des dimensions considérables (Mb et Gb). C'est une méthode donnant des bons résultats avec des données textuelles.
- Systèmes basés sur le texte (*text-based systems*) – permet une annotation textuelle de l'information (souvent mélange de descriptions succinctes et données structurées.) Le principal inconvénient est qu'une telle méthode est difficile de mettre en place pour des grosses bases de données multimédia ou pour certains domaines d'activité, est qu'elle laisse beaucoup de place à la subjectivité humaine (une même image est décrite autrement par des observateurs différents). Pour ce type de recherche, la chance joue un rôle important, car impossible de savoir les termes utilisés pour l'annotation.
- Systèmes basés sur le contenu (*content-based systems*) - vise à contrecarrer ces faiblesses, étant basés sur l'extraction automatique des détails (attributs) identifiant l'information désirée. C'est la méthode permettant une importante augmentation quant à l'efficacité du processus de repérage.

En renforçant ces idées, Kosch ([15]) souligne l'importance de l'analyse de contenu, des métadonnées associées au contenu et des modèles utilisés pour représenter les données multimédia. La synthèse fournie par Kosch concernant la modélisation est représentative des préoccupations futures en ce qui concerne le repérage de l'information multimédia :

- la BD multimédia (BDMM) doit permettre de stocker la modélisation des données. Le mécanisme de stockage joue un rôle important quant à la performance du système multimédia

- L'existence d'une représentation de la structure logique est essentielle. (requêtes et représentation)
- Le niveau sémantique doit être modelé et lié directement aux caractéristiques de bas niveau.
- Les métadonnées nécessaires doivent être stockées dans la BDMM
- Il est nécessaire d'assurer l'interopérabilité de partage et l'échange de l'information. Cela signifie l'alignement aux standards internationaux, comme c'est MPEG-7. En effet, l'apparition du standard MPEG-7 a influencé de façon majeure le CBR. Il offre une standardisation de contenu (descripteurs de bas niveau et haut niveau) permettant la réalisation des mécanismes d'interrogation interopérables.

Ce que soulignent ces deux auteurs (Kosch ([15]) et Dunckley ([16])) est l'importance des métadonnées pour le processus de repérage des objets multimédia. Littéralement "données sur des données", les métadonnées ont été déjà utilisées pour la gestion de l'information dans les BD classiques. (Schémas, contraintes, limites acceptés pour les données, etc.). Dans le cas du multimédia le rôle des métadonnées est primordial, permettant un processus d'interrogation plus efficace. Les métadonnées peuvent augmenter la performance du processus et donnent un aide en temps réel pour l'utilisateur, car, un changement du média apporte automatiquement un changement des métadonnées associées (consistance de l'information dans la BDMM). L'information visuelle joue aussi un rôle primordial en multimédia. Dunckley ([16]) fait l'analyse des méthodes populaires pour le repérage des images ou de la vidéo et identifie les approches suivantes :

- Repérage d'image :
 - Basée sur la couleur (histogrammes, invariance de la couleur)
 - Basée sur la texture (variation de l'intensité et topographie de la surface)
 - Basée sur la forme (allongement, circularité, contour, etc.)
 - Basée sur la position (indexation spatial)
 - Basée sur la transformation de l'image (transformations complexes de l'intensité des pixels composant l'image)
 - Basée sur l'apparence (combinaison de couleur, texture, intensité, etc.).
- Repérage des vidéos (ou séquences vidéo) :
 - basée sur analyse des images

- Basée sur l'analyse des mouvements
- Basée sur la détection des objets et des trajectoires
- Basée sur la reconnaissance de la parole
- Basée sur l'identification d'orateur (visage et parole)

Ce qui est essentiel, souligne Kosch ([15]), c'est l'utilisation d'une approche ou méthode permettant de quantifier la similarité entre deux objets multimédia. C'est la distance entre la requête de l'utilisateur et l'objet situé dans la BDMM. En ce qui concerne l'outil idéal pour arriver au but désiré, serait un langage de requêtes multimédia qui aurait l'habilité de travailler avec des données interreliées de façon spatiale et temporelle. Il serait capable d'utiliser à la fois des mots clé, des index de mots clés, du contenu sémantique et des objets multimédia pour effectuer sa requête.

1.3 Méthodologie de recherche

La démarche du projet est décrite en utilisant le cadre expérimental proposé par Basili et al. (1986) et modifié par A. Abran et P. Bourque (1997) afin de l'adapter pour le travail de recherche exploratoire.

Le projet est divisé en quatre étapes.

Étape 1 : Définition du projet

La motivation principale de ce travail de recherche est d'identifier les possibilités offertes par ORACLE 10g en ce qui concerne le repérage des informations multimédia basées sur les descripteurs de bas niveau. Plus précisément, on vise de faire l'analyse des images tirées du flux vidéo pour annoter les descripteurs inter séquences. L'identification des changements de scènes sera faite à la suite des analyses de la similarité des images successives.

L'objet de l'étude est principalement le repérage des images basée sur le contenu et l'objectif est la création d'un module logiciel traitant les images, module qui s'occupera à la fois du stockage de la vidéo et des images dans la BDMM et de la détection des séquences.

L'objectif - Créer un module logiciel capable de traiter des images tirées de vidéo. (stockage de la vidéo en Oracle, extraction des images à partir de la vidéo, détection des séquences, sauvegarde des caractéristiques de l'image représentative pour la séquence vidéo en format XML (MPEG-7)).

Les utilisateurs - à la fois les étudiants de l'ETS et ceux de l'université d'Autriche. Le présent projet offre des informations récentes sur les préoccupations et les réalisations dans le domaine de multimédia. Le prototype développé peut être le point de départ pour des futures recherches

Étape 2 : Planification du projet

On commence par faire un état de l'art sur le repérage de l'information multimédia. Ensuite, un prototype réalisé en Autriche fera l'objet d'une analyse approfondie. On se propose d'ajouter des nouvelles fonctionnalités pour celui-ci. Il s'agit d'un module traitant des images représentatives pour la vidéo (changements de scènes).

Étape 3 : Exécution du projet

On commence par analyser et grouper les informations selon leur utilité pour la présente recherche (annotation automatique ou manuelle, ontologie, métadonnées, modèles de données, xml, vidéo, image, texte, etc). Les étapes suivantes seront de développement logiciel proprement dit : création des modules traitant le stockage des images en Oracle et l'analyse de la similarité, développement de module d'extraction des images à partir du flux vidéo, et identification des séquences, Création des documents XML conformes aux spécifications MPEG-7 pour les images

Étape 4 : Interprétation des résultats

On analyse les résultats obtenus et on les compare aux attentes. On analyse aussi le comportement dynamique de l'application (impact des changements des intrants sur les extrants). On va modifier les valeurs associées aux attributs de bas niveaux (couleur, texture, forme) et le seuil de similarité et on va interpréter les résultats obtenus. Les conclusions tirées permettront d'identifier les limites existantes et les améliorations possibles.

Le tableau suivant donne les détails concernant l'approche Basili pour le projet :

1- Définition				
Motivation	Objet	Objectif	Utilisateurs	
▪ repérage d'images basé sur le contenu (descripteurs de bas niveau) ▪ analyse d'images (tirées de vidéo) pour annoter les descripteurs inter séquences)	▪ repérage d'images basé sur le contenu.	▪ Ajouter une fonction traitant les images. (stockage de la vidéo, extraction des images à partir de la vidéo, détection des séquences, sauvegarde des caractéristiques de l'image représentative pour la séquence vidéo en format XML (MPEG-7)	▪ étudiants baccalauréat - informations récentes sur le repérage de l'information multimédia; exemple pratique lecteur vidéo et module traitement images basé sur MPEG-7 ▪ des étudiants à la maîtrise – point de départ pour continuer la recherche sur le repérage multimédia basé sur le contenu. (développement du prototype existant, ajout des nouvelles fonctionnalités, etc.) ▪ chercheurs de l'université d'Autriche (collectif Dr.Kosch).	
2- Planification				
Étapes du projet	Intrants		Livrables	
▪ état de l'art sur la recherche et le repérage de l'information multimédia basé sur le contenu. ▪ analyse du prototype de player multimédia développé par l'université d'Autriche ▪ analyse et conclusion	▪ ressources bibliographiques. ▪ Prototype Autriche (logiciel), spécifications, documentation ▪ Documentation (support papier ou enligne) pour l'environnement du travail (MPEG-7, Oracle10g, Java, JAXP, XML, Eclipse, etc.)		▪ état de l'art sur le repérage multimédia (image et vidéo) ▪ rapport d'analyse application (prototype) Autriche ▪ description modules application locale (stockage vidéo et images dans la BDMM, extraction images du flux vidéo, analyse similarité images, création documents XML conformes MPEG-7) ▪ rapport final	
3- Exécution				
Préparation	Conception	Programmation unitaires	et tests	Chargement des données et essais
▪ grouper les informations collectées ▪ Choix et Installation environnement ▪ Choisir la méthodologie appropriée pour le cycle de vie du projet logiciel ▪ Réaliser le plan de	▪ Conception générale ▪ Conception détaillée	▪ développement des modules traitant le stockage des images en Oracle et l'analyse de la similarité ▪ développement du module d'extraction des images à partir du flux vidéo, et identification des séquences. ▪ Création des documents XML conformes aux spécifications MPEG-7 pour		▪ Vérification fonctionnalités modules ▪ Tests fonctionnement système

- | | | | |
|---|---|--|--|
| <ul style="list-style-type: none"> ▪ grouper les informations collectées ▪ Choix et Installation environnement ▪ Choisir la méthodologie appropriée pour le cycle de vie du projet logiciel ▪ Réaliser le plan de travail ▪ modélisation UML | <ul style="list-style-type: none"> ▪ Conception générale ▪ Conception détaillée | <ul style="list-style-type: none"> ▪ développement des modules traitant le stockage des images en Oracle et l'analyse de la similarité ▪ développement du module d'extraction des images à partir du flux vidéo, et identification des séquences. ▪ Création des documents XML conformes aux spécifications MPEG-7 pour les images. | <ul style="list-style-type: none"> ▪ Vérification fonctionnalités modules ▪ Tests fonctionnement système |
|---|---|--|--|

Tableau1 – cadre Basili modifié

1.4 Sommaire

Ce chapitre a présenté une vue d'ensemble de la recherche. On a débuté par spécifier le contexte et d'identifier la motivation de la présente recherche. Ensuite, on a présenté la méthodologie utilisée pour arriver au but désiré. Le projet a été divisé en quatre grandes étapes : définition, planification, exécution et interprétation des résultats. La démarche du projet est décrite en utilisant le cadre expérimental proposé par Basili et al. (1986) et modifié par A. Abran et P. Bourque (1997) afin de l'adapter pour le travail de recherche exploratoire.

2. CHAPITRE 1 – recherche bibliographique

2.1 Objectifs et stratégie de la recherche bibliographique

Objectif

- Retracer des informations (les coordonnées, les résumés et le texte intégral) concernant les préoccupations et les réalisations les plus récentes (depuis janvier 2002) concernant le standard MPEG-7 (Moving Picture Experts Group)¹, et l'utilisation

¹ « MPEG-7 Overview ». URL: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

contextuelle de XML² (balisage, métadonnées), dans le domaine de la vidéo et des images.

- Si c'est possible, retracer des documents concernant le développement des applications réalisés en utilisant un environnement Oracle³

Choix bases de recherche et stratégie

Voici les sources consultées pour la revue bibliographique :

Le choix fait est les justifications sont décrites dans le tableau suivant :

Nom	Justification
ACM Digital Library (http://www.acm.org/dl)	Renommé (ACM – Association for Computer Machinery est l'une des plus renommées compagnies en technologie de l'information et informatique. Plus de 100 pays bénéficient de ces services en divers domaines (industrie, niveau académique, gouvernement etc.) ⁴ Couverture du domaine (1947-à présent) Information (pertinente, disponible en texte intégral, mise à jour de façon régulière)
IEEE Xplore (http://ieeexplore.ieee.org/)	Renommé (IEEE - Institute of Electrical and Electronic Engineers est un grand organisme américain de normalisation en informatique et technologie de l'information) Couverture du domaine (des transactions, périodiques, magazines et conférences publiés par l'IEEE ainsi qu'à leurs normes courantes, depuis 1988, (des références depuis 1950)). ⁵ Information (pertinente, disponible en texte intégral, mise à jour de façon régulière)
INSPEC (http://www.engineeringvillage2.org)	Renommé : Inspec est considéré comme une des plus complètes bases bibliographiques du monde, couvrant les domaines de la physique, l'électronique, la construction et l'utilisation des ordinateurs, la technologie de l'information. ⁶ Couverture : Inspec contient plus de 8 millions d'enregistrements pris de 3400 revues techniques et scientifiques et 2000 présentations aux diverses conférences scientifiques. La mise à jour est faite chaque semaine. Information (pertinence, disponible texte intégrale, mise à jour chaque semaine)
IsiWEB (http://isiknowledge.com/wos)	Base très utile pour le repérage des articles qui citent des articles déjà trouvés.
Gale	Gale est développée par Thomson, ce qui donne confiance quant à

² « Extensible Markup Language (XML) ». URL : <http://www.w3.org/XML/>

³ « Oracle » URL : <http://www.oracle.com/index.html>

⁴ « The ACM Digital Library ». URL : http://www.acm.org/about_acm/

⁵ « IEEE Xplore ». URL : <http://ieeexplore.ieee.org/Xplore/DynWel.jsp>

⁶ « Engineering Village 2 ». URL : <http://www.engineeringvillage2.org/>

http://infotrac.galegroup.com/itw/eb/alise_udm	la qualité de l'information disponible (pertinence sources, qualité liens, mise à jour, etc.). Gale vise le monde informationnel et éducationnel en offrant l'information désirée sous multiples formats: livres, solutions Web, microfilms. (www.gale.com). Les bases de données Thomson Gale sont extrêmement vastes, contenant plus que 5550 titres actives dont 3100 avec texte intégral, et le partenariat avec Ingenta apportera encore 5400 d'articles texte intégral. (source : le site de Gale : www.gale.com). La mise à jour est quotidienne.
---	---

Tableau2 – choix et justification bases de recherche documentaire

La stratégie de recherche utilisée se résume comme suit:

- l'identification et l'analyse des concepts (mots clés, synonymes, équivalences en autres langues, etc.)
- Recherche approfondie dans ACM Digital Library, IEEE Xplore, Inspec.
- Analyse de la pertinence des documents trouvés. Identification des documents vraiment pertinents
- Recherche des citations des documents pertinents trouvés (utilisation IsiWeb).
- Recherche dans Gale (des documents inexistant dans les trois bases déjà utilisées)
- Utilisation de Copernic (utilisation de la puissance et des fonctionnalités d'un métamoteur.)

En suivant un processus par itération cyclique, les documents non pertinents ont été éliminés.

2.2. État de l'art sur le repérage de l'information multimédia

La croissance accrue des sources multimédia et le besoin d'assurer constamment leur disponibilité exigent de plus en plus des systèmes capables d'administrer rapidement, avec maximum d'efficacité des grands volumes des données hétérogènes. On a besoin des systèmes intelligents, flexibles, adaptables. Le standard MPEG-7 est depuis quelque temps la référence quant à l'analyse et la l'utilisation des données

multimédia. Il assure l'interopérabilité des données, en se basant sur des descripteurs associés au contenu de l'information audio-visuelle.

Le standard MPEG-7 (ISO/IEC, standard international 15938) a été développé par MPEG (Moving Picture Experts Group) et définit une normalisation du contenu multimédia à plusieurs couches (de bas niveau jusqu'à la description sémantique). Il permet d'enregistrer de l'information au niveau administration du contenu (navigation, accès, interaction) et contenu proprement dit (structure et sémantique). Il permet d'aborder plusieurs formats multimédia supportant CBR et assure l'interopérabilité entre les systèmes véhiculant du contenu MM. Un grand avantage est la compatibilité de MPEG-7 avec d'autres standards comme W3C(XML, XMLSchema), IETF (URI,URN,URL), Dublin Core, SMPTE, Metadata Dictionnary, etc.

En ce qui concerne la présente recherche, l'analyse de la vidéo, l'identification des cadres représentatives et l'annotation des images nécessitent la coexistence et l'interaction de plusieurs domaines, outils et techniques : BDMM, Vidéo, MPEG-7, métadonnées et annotation, CBIR (repérage de information basé sur le contenu. A cette lumière, le tableau suivant synthétise le groupement des ressources documentaires que j'ai identifié selon ces catégories :

Catégorie	Référence documentaire
BDMM	[2]
Vidéo	[6], [9]
MPEG-7	[3], [5], [14]
Annotation	[1], [7], [10], [12], [13]
CBIR	[4], [8], [11]

Tableau3 – groupement références documentaires

2.2.1 BDMM

Ce qui apporte de la valeur est, comme souligne Harald Kosh en [2], le fait que MPEG-7 n'est pas un concurrent ou un remplaçant pour les systèmes de gestion de bases de données multimédia. Il est pensé et doit travailler en complémentarité avec ceux-ci. Il faut donc comprendre très bien les interliaisons entre ce standard et les bases de données multimédia, MMDB. Il faut mettre en valeur qu'il s'agit d'une complémentarité et pas d'une concurrence entre les deux notions. Pour argumenter, ([2]) base son raisonnement sur les aspects suivants :

- **Les MMDBS** (systèmes de base de données multimédia) organisent les structurent l'information multimédia afin de faciliter le repérage basé sur le

contenu. D'où l'importance accrue de la modélisation, c'est-à-dire quelle information doit être gardée et quelles sont les liaisons entre les composants de la base de données. Généralement, les modèles proposés au cours des années ont été basés sur les besoins des développeurs des bases de données et celles des usagers finaux. Récemment, l'utilisation de la technologie orientée objet enrichisse la palette des applications possibles, apportant flexibilité, économie et performance (encapsulation, réutilisation du code, héritage, etc.). Dans notre domaine d'intérêt, des nouveaux types de données et des méthodes associées ont été introduits, pour utiliser pleinement les possibilités offertes par le texte, les images ou la vidéo. L'utilisation de la technologie relationnel objet (RO) et implanté et supporté par SQL/MM qui a été standardisée en mai 2002 (le sous-comité SC32 WORKING GROUP 4.). SQL/MM introduit des types objet structurés pour les images, des méthodes de garder, manipuler et représenter l'information basée sur le contenu (les histogrammes par exemple). C'est le contexte dans lequel, en mai 2002, un autre sous-comité de ISO (SC29 WG11 MPEG – Moving Picture Experts Group) a publié un standard, nommé MPEG-7 basé sur un set de descripteurs (pour quantifier de façon quantitative les propriétés audio-visuelles des composants) et des schémas de description (pour mettre en valeur les relations entre les composants, donc la dimension qualitative du phénomène).

- **Les modèles MMDBS et SQL/MM** : le modèle MMDBS doit résoudre les problèmes liés aux divers opérations avec des données multimédia : sélection, insertion, édition, indexation, furetage, repérage, interrogation. Les dernières années ont proposées plusieurs modèles pour les MMDBS (ou SBDMM- les systèmes de bases de données multimédia) :
 - Modèle algébrique – fait la différence entre le « *stream* » vidéo physique et celui logique et permet à l'utilisateur d'utiliser une algèbre vidéo (mécanisme pour combiner et mettre en évidence les relations temporelles entre les représentations vidéo)
 - Modèle logique hypervidéo – LHVDM (*logical hypervideo data model*) – permet de faire des abstractions multicouches pour les objets choisis et de spécifier aussi les relations avec autres objets. Le modèle supporte des associations sémantiques « vidéo hyperlinks ».

- Modèle DISMA – spécifique pour images et applications spatiales. Permet de changer la sémantique d'un objet partie composante d'une image sans produire des effets dans l'application le contenant. C'est un modèle basé sur un langage « query »m MOQL qui enrichi OQL.
- Modèle VIDEX – basé sur la coexistence des classes sémantiques et relations sémantiques. Le noyau de ce modèle déclare les classes de base et les classes des applications spécifiques sont déclarées comme sous-classes des classes de base. (classes de contenu)

Pour assurer une grande portabilité pour un système supportant des données multimédia, c'est le standard SQL/MM qui a été développé. Il n'existe pas actuellement un standard pour les requêtes sur les données vidéo, malgré le fait que certaines propriétés de SQL/MM spécifiques pour les images sont applicables pour la vidéo.

À la suite d'une analyse comparative des MPEG-7 / SBDMM (systèmes de base de données multimédia) on peut identifier plusieurs différences ([2]) :

- **Complexité** : **MPEG-7** essaye de minimiser l'hiérarchie liée à la description du contenu. Au contraire, les modèles comme VIDEX sont très complexes.
- **Modalité de définition** des schémas de descripteurs :
 - MPEG-7 segment DS permettant des liaisons pour navigation rapide dans l'arbre hiérarchique, utile pendant l'échange informationnel.
 - MMDBS – plusieurs niveaux d'abstractions (niveau « soft » – cadres successives temps et espace pour les actions et niveau « scène » - plusieurs cadres liées de façon sémantique)
- **Schéma** de codage pour les documents MPEG-7 basé sur XML – MPEG-7 vient avec BIM (*Binary Format*) qui permet d'effacer ou modifier les « unités d'accès » des documents.
- **Traitement** des éléments descriptifs :
 - MPEG-7 : beaucoup d'outils spécialisés, comme :
 - SemanticDS – description des entités sémantiques et leurs relations. (dépendance arborescente objets, événements, espace et temps)
 - ContolledTermType – control d'un champ de type texte en utilisant un SC (schéma de classification).

- MMDBS : plus simple (par exemple une simple référence comme BFILE d'Oracle)

Considérant tous ces aspects, ([2]) considère comme parfaitement possible, même désirable une association des MMDBS et MPEG-7. La conclusion est que **MPEG-7** et MMDBS peuvent se compléter réciproquement pour réaliser des systèmes multimédia distribués (SMMD, anglais MMDS). Pour assurer effectivement la participation de MPEG-7 en SMD il faut utiliser des mécanismes de stockage et des requêtes des documents XML basés sur le standard MPEG-7. Les modalités concrètes sont :

- **Soit** utiliser une extension de la SGBD qui permet d'implémenter le modèle MM et générer documents MPEG-7 pour le dialogue avec l'extérieur (par exemple XSU⁷ d'Oracle)
- **Soit** stockage et interrogation directe des MPEG-7 (utilisation de XML-DBMS, capables de réaliser des interfaces et langages respectant la nature des documents XML- par exemple IPSIS⁸, dbXML⁹)

Le « codec » MPEG-7 est un autre atout pour SMD, car le codage et le décodage est fait en respectant les versions et les extensions. L'interopérabilité est assurée, l'utilisateur pouvant paramétrer selon son désir l'encapsulation de l'information dans « unités d'accès », fragments qui seront décodés d'un coup à la destination. C'est un grand avantage car cela signifie qu'on peut contrôler la décodification à la source.

Plusieurs applications ont déjà démontrée la force des affirmations faites en ([2]), mettant en pratique des systèmes activant concret sur la collaboration entre les SGBDMM et le MPEG-7.

Un SGBDMM (Système de base de données multimédia) stocke et organise les données multimédia afin de pouvoir exécuter des opérations de repérage base sur le contenu. On peut affirmer que à présent existent plusieurs modèles basés sur requêtes, indexation etc., mais ils sont tous limités, soit de point de vue données multimédia acceptées, soit en ce qui concerne la capacité sémantique de modélisation. Malgré l'impact apporté par le standard MPEG-7, il n'existe jusqu'à présent un SGBDMM capable d'intégrer toutes les avantages concernant l'indexation et les requêtes possibles.

⁷ XML SQL Utility

⁸ <http://xml.darmstadt.gmd.de/xql/>

⁹ <http://www.dbxml.org>

2.2.2 Vidéo

La disponibilité accrue de la vidéo digital crée un fort besoin de pouvoir analyser de façon automatique le contenu MM (trouver, accéder, filtrer et administrer les données vidéo). Le MPEG-7 s'occupe du contenu des vidéos, en appuyant sur les dimensions spatiales et temporelles. Le cadre du modèle expérimental de MPEG-7, XM, est aussi à la base de l'application développée en ([6]). Cette fois-ci on exploite le fait que MPEG-7 standardise quelques descripteurs de mouvement, facilitant l'accès à la dimension temporelle. L'application décrite en ([6]) propose donc un descripteur visuel qui analyse la trajectoire vidéo, lié aux mouvements dans une séquence vidéo. On étudie les positions occupées successivement par les objets présents dans la vidéo, considérant chacun d'entre eux comme étant une région spatio-temporelle. Une méthode d'extraction des trajectoires et de génération des documents XML associées, conformes aux spécifications MPEG-7 est présentée. On a utilisé le XM (eXperimentation model) de MPEG-7, qui permet d'utiliser comme entrée une liste de points clés de vérification des objets segmentés et de fournir la sortie sous forme de document XML.

En grandes lignes, l'application est basée sur :

- Les objets peuvent être observés à travers plusieurs images, en analysant la mise à jour de leur contour dans des cadres successifs. Plusieurs algorithmes ont été inventés, basé sur l'analyse de la région occupé par l'objet ou son contour.
- La méthode exige les étapes suivantes :
 - o Développer une estimation globale de mouvement – algorithme qui fait la compensation et enchaîne le cadres vidéos l'un à l'autre.
 - o Détecter les particules vidéo en mouvement
 - o Analyser les évolutions des trajectoires
 - o Analyser et faire des transformations sur les données obtenues – raffiner les trajectoires
 - o Sortie – des documents XML conformes à MPEG-7

Plus précisément on a :

- L'estimation du mouvement : La technique est basée sur quelques paramètres dont la surface, l'inclinaison et l'échelle. Le déroulement cadre par cadre peut être approximé par les vecteurs de mouvement (MV) dans le flux MPEG. On appliquant des formules mathématiques

qui font la liaison entre les frames I, P, B on peut calculer le mouvement d'un macro block (MB) dont on connaît le centre. La différence entre le déplacement réel et celui estimé nous donne le vecteur de mouvement résiduel. Une moyenne des ces vecteurs est calculée et la valeur est comparée avec un seuil fixé en préalable. Si on a un dépassement on considère qu'une mouvement a été effectué sinon on initialise à zéro tous les paramètres calculés. (on conclut qu'on n'a pas eu un mouvement).

- L'extraction de la trajectoire :
 - Détection des gouttes (unités) de mouvement : analyse existence mouvement, construction des images binaires entre les cadres suivantes et modifications des images binaires (fusion ou effacement) selon le seuil fixé.
 - L'évolution des trajectoires – addition successive des gouttes de mouvement, en calculant la distance entre le centre de l'unité de mouvement et le sommet de la chaîne déjà construit. (on considère la matrice RGB et, si c'est le cas, on crée d'autres trajectoires)
 - L'amélioration des trajectoires - éliminer les fausses trajectoires, filtrage contre le "bruit", correction allure du mouvement.
 - Format de sortie pour les trajectoires : l'information de la trajectoire du mouvement (la liste des points) est stocké en format XML, car c'est un format accessible, simple, interchangeable. (même MPEG-7 a choisi XML pour son DDL, pour la description du contenu MM)

Une des forces du standard MPEG-7 est qu'il fait le pas vers la sémantique du contenu. MPEG-7, véritable interface de description pour le contenu MM, lui attache des métadonnées et spécifie plusieurs outils capables de faire le décrire. Le système présenté en ([9]) apporte en premier plan l'importance des deux notions de base pour l'analyse du contenu : syntaxe (dans ce cas les cadre du flux vidéo) et sémantique (dans ce cas les scènes identifiées)

Le système est basé sur l'idée que la recherche et la localisation de la vidéo peuvent utiliser la décomposition du flux vidéo en cadre et scènes, selon le désir de l'utilisateur. Jusqu'à présent, l'idée générale de bâtir les systèmes de repérage ont adoptés le concept QBE (query by exemple), qui quantifie les données vidéo selon les propriétés

déTECTABLES (couleur, texture, forme, mouvement). C'est le début d'une contradiction, car, dans la plupart des cas l'utilisateur exprime ses doléances sous le format des questions de haut niveau. Alors, soit il fait lui-même la translation de la requête vers les attributs décrivant le contenu sémantique, soit donne un exemple complet (une image), considérée comme modèle de recherche. Dans l'application présentée ([9]), un document vidéo est considéré étant caractérisé par des scènes qui à leurs tour sont formées de plusieurs cadres, comme on spécifie dans la figure suivante :

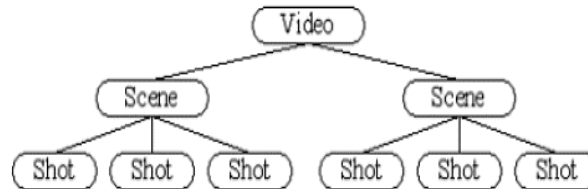


Figure 2 : Structure d'un document vidéo

Source image : A. Graves, M. Lamas. *Video retrieval using an MPEG-7 based inference network*, [9]

Alors, une description des sémantique de haute niveau peut prendre l'allure d'une "évidence héritée", dans le sens que, une information associée à une flux vidéo automatiquement est associée à ses composantes (scènes et cadres), et inversement, chaque information contenu dans un cadre contribue à définir le profil de l'information du vidéo entier. Le modèle adopté pour réaliser le système est nommé INM (inference network model), modèle de réseau d'inférence. Il est utilisé pour modéliser les documents, leur contenu et les requêtes. Ce réseau d'inférence utilise deux sous-réseaux, le DN (document network) et le QN (query network). Le DN est constitué des nœuds et concepts du document analysé, qui sont en fait des éléments retrouvables pour les futures requêtes. Chaque nœud est lié aux plusieurs concepts, les liaisons de causalités étant spécifiées par différents poids, ou indicateurs de force de la liaison. Le QN représente de son côté des clichés, des vues des plusieurs nœuds affectés à une requête donnée, et leur liaison, sous forme d'un arbre inversé.

Pour réaliser le repérage voulu, il faut attacher le QN au DN correspondant, formant ainsi le IN (inference network). Ensuite, on passe à une autre étape, l'évaluation qui exige l'analyse de chacun des nœuds de l'IN pour trouver la probabilité de similarité avec la requête. Le DN a trois couches, comme dans la figure suivante :

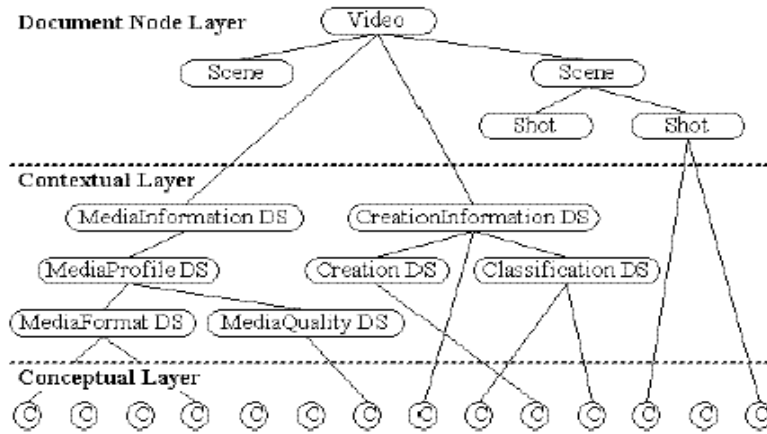


Figure 3 : Les trois couches du sous-réseau DN (*document network*)

Source image : A. Graves, M. Lamas. **Video retrieval using an MPEG-7 based inference network**, [9]

Couche 1 : Document Node Layer : c'est la structure qui représente la structure correspondante à la description MPEG-7. C'est une structure provenant de l'extraction des composants structurels MPEG-7. De plus, on ajoute des probabilités donnant la puissance de la liaison entre les nœuds et les concepts associées.

Couche 2 : Contextual layer : C'est une couche qui exploite pleinement la richesse des informations offerte par le MPEG-7 et la location des concepts dans la hiérarchie DS (descripteur schème) du standard. Cette information structurelle est utilisée pour :

- estimer plus précisément combien « pèse » un concept pour le contenu
- identifier exactement les concepts qui interviennent dans un contexte donné

Couche 3 : contient les concepts identifiés pendant l'analyse de la requête. (Analyse de document, identification des concepts, identification des inter liaisons et de leur force).

Le QN contient des nœuds de requête et des opérateurs de liaison comme dans la figure suivante :

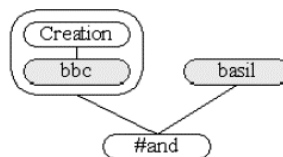


Figure 4 : Structure du sous-réseau QN (*query network*)

Source image : A. Graves, M. Lamas. **Video retrieval using an MPEG-7 based inference network**, [9]

L'attachement entre le SDN et QN est fait à la suite d'un calcul rigoureux des "distances" entre les noyaux du DN et du QN. Le candidat ayant de meilleur score est

retenu. Une fois l'attachement réalisé on procède à l'évaluation des résultats basée sur une estimation de la probabilité conditionnelle entre les nœuds et les concepts, estimation faite seulement sur le DN (le QN utilise la dépendance donnée par les opérateurs pour s'accrocher au DN approprié). On considère deux types de probabilités conditionnelles :

- Contexte → Contexte (ou parent → enfant, de genre vidéo→scène, scène→cadre)
- Contexte→ Concept (ici intervient MPEG-7 qui offre un élément important, la durée (*frame duration*). On peut appliquer les formules suivantes pour calculer les probabilités :

$$\begin{aligned} \text{Weight} &= \text{tf}(t,d) * \text{idf}(t) \\ \text{Pweight} &= 0.5 + (0.5 * \text{weight}) \end{aligned}$$

t: terme (élément)

D : document

Tf : fréquence de l'élément (*term frequency*)

Idf : poids du terme (*inverse document frequency*)

Il faut considérer la description MPEG-7 comme la source des poids associés aux termes recherchés. Par exemple, une structure comme la suivante :

```
<StructuredAnnotation>
  <Who>Basil,Sybil,Andre</Who>
</StructuredAnnotation>
```

Contient trois concepts (Basil, Sybil et Andre) dans un seul descripteur MPEG-7, séparées par des virgules. Cela impose automatiquement que aucun mot ne peut être considéré comme représentative pour une recherche future. C'est normal alors de considérer une probabilité de genre ($1/\text{nombre_evenements_possibles}$). Comme titre d'exemple, une situation de la recherche du concept "banana" pourrait être la suivante :

- On peut considérer que le MDC est défini par 2 grandes parties :
 - Le MDM = Le modèle de données multimédia (multimedia data model) – qui contient des métadonnées reliées aux spécifications MPEG-7 (descripteurs). Le schéma MPEG-7 est mappée à l'aide d'Oracle à un schéma BD (objets et tables)¹⁰. Le principe de cette relation est de mettre en valeur la différence entre les types MPEG- 7 importants, (utilisés comme ancres pour les descripteurs et appelées souvent dans les requêtes) et les autres, constituant des éléments optionnels dans les requêtes.
 - Le MIF = le cadre d'indexation multimédia. C'est un environnement extensible pour le repérage multimédia. Il est construit par 3 modules :
 - GistService - réalisé dans l'espace d'adresses externes et actionnant comme un processus indépendant dans le SE (système d'exploitation). À son tour, il a deux composants :
 - GistCommunicator = réalise la communication entre la BD et les méthodes d'accès.
 - GistHolder= s'occupe de l'administration des arbres d'indexation et de l'accès à ces arbres.
 - GistWrapper- bibliothèque partagée utilisée pour que la BD se connecte à GistService.
 - Multimedia Index Type = extension du mécanisme d'index d'Oracle. Consiste en quelques types d'index représentant des méthodes d'accès et les opérateurs associés. Chacun index de ce type est associé à un objet.

La figure suivante montre la modalité dont le MDC est incorporé dans le contexte du système distribué.

¹⁰ <http://www-itec.uni-klu.ac.at/~harald/codac/schema.pdf>

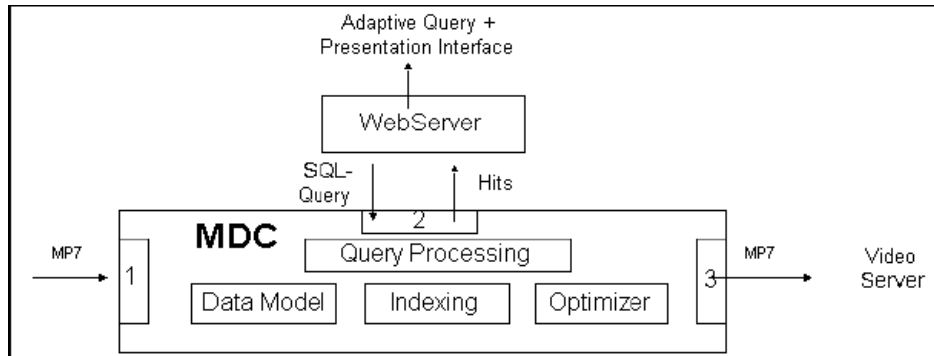


Figure 7 : Intégration contextuelle de MDC (*multimedia data cartridge*)

Source image : (M. Doller, H. Kosch, B. Dorflinger, et al. *Demonstration of an MPEG-7 multimedia data cartridge*,) [3].

On y trouve 3 interfaces :

- a) la première interface fournit les connexions pour les différents outils véhiculant ou produisant des données MPEG-7. Les documents MPEG-7 à inclure dans la BD sont analysés et transformés en objets et table spécifiques MDC.
- b) La deuxième interface permettra l'interrogation et la présentation du contenu de la BDMM. C'est une interface construite de façon dynamique dans un serveur Web, basé sur des descripteurs de l'environnement utilisé. Cette interface (en anglais Adaptive Query and Presentation interface) considère les exigences directement liées aux capacités hardware et software du terminal utilisé, et aux préférences de l'utilisateur.
- c) La troisième interface offre la connexion vers le serveur vidéo. Elle est responsable d'assurer la synchronisation entre les informations MPEG-7 (les données contenues dans les requêtes) et le flux de données pour assurer un transfert optimum.

Schématiquement, le processus d'adaptation offert par MDC est le suivant :

- a) L'utilisateur s'identifie et donne quelques éléments pour personnaliser sa démarche (performances du terminal, exigences ou préférences concernant les résultats)
- b) Toutes ces données sont interprétées à l'aide de CC/PP (Composite Capabilities / Preferences Profile) qui est un cadre standardisé basé sur le format RDF développé par W3C. Ces descriptions sont envoyées vers un serveur de traitement requêtes, qui, après les analyser, retourne la classe

dont l'utilisateur appartiendra pendant toute la session. L'appartenance à l'une ou d'autre classe va personnaliser la démarche de l'utilisateur.

- c) L'affichage et utilisation des éléments caractéristiques pour chaque utilisateur sont spécifiés à l'aide d'un langage basé sur un moteur adaptatif, capable de travailler avec des données multimédia.
- d) L'utilisateur spécifie les critères de et envoie la requête au serveur Web adaptatif qui le présente à son tour au serveur de la BD. Les résultats obtenus font le chemin inverse, étant retournés et présentés à l'utilisateur en utilisant le chemin adaptatif.

Un autre exemple de la collaboration fructueuse de MPEG-7 et des SGBDMM est illustré par le système développé en ([5]). Le système (dont le nom est ERIC7, développé par CRIM - *Computer Research Institute of Montreal*) utilise XM (eXperimentation model) comme environnement de travail. ERIC7 c'est à la base un logiciel qui implémente CBIR (content based image retrieval) compatible avec le standard MPEG-7. Il permet le codage MPEG-7/XML jusqu'à 15 descripteurs (couleur, texture et forme) utilisant 2 façons d'interrogation repérage à partir d'un modèle (image) et recherche à bas niveau (clustering). Le système permet aussi la navigation parmi les descripteurs XML et peut être facilement développé pour traiter l'indexation et le repérage des séquences vidéo. Plusieurs idées principales sont à retenir :

- ERIC7 – un environnement générique de recherche images par contenu compatible MPEG-7. Il a été développé par une équipe de CRIM (Computer Research Institute of Montreal).
- Utilise des outils d'analyse de bas niveau (descripteurs de couleur, texture et forme) développées maison ou intégrés.
- CBIR a incité le développement de plusieurs prototypes, mais jusqu'à présent aucun n'est totalement compatible avec MPEG-7.
- Une particularité d'ERIC7 est qu'il vise le codage automatique et la navigation dans le schéma MPEG-7 pour des raisons de recherche et analyse.

La description MPEG-7 est en fait une description XML capable d'instancier deux catégories d'outils :

- Les descripteurs D (définissant la syntaxe XML et la sémantique de chaque propriété du contenu MM)

- Les schémas des descripteurs DS (spécifiant la structure et la sémantique des relations entre les D)

Les D est les DS sont des structures définies par DDL (Description Definition Language), basé sur le schéma W3C XML. On peut avoir accès à 25 outils spécifiques du contenu visuel (images fixes, vidéo, modèles 3D). Une plateforme qui simule les outils MPEG-7, destinée pour assurer l'interopérabilité entre plusieurs plateformes a été réalisée. Il s'agit de XM (eXperimentation Model), qui fournit un cadre commun permettant de traiter les composants standard et non standard des propriétés spécifiques aux opérations de codification, extraction et recherche. Pour chaque D ou DS on a deux applications liées, une pour le serveur (extraction) et l'autre pour le client (recherche, filtrage, codage).

L'architecture du système :

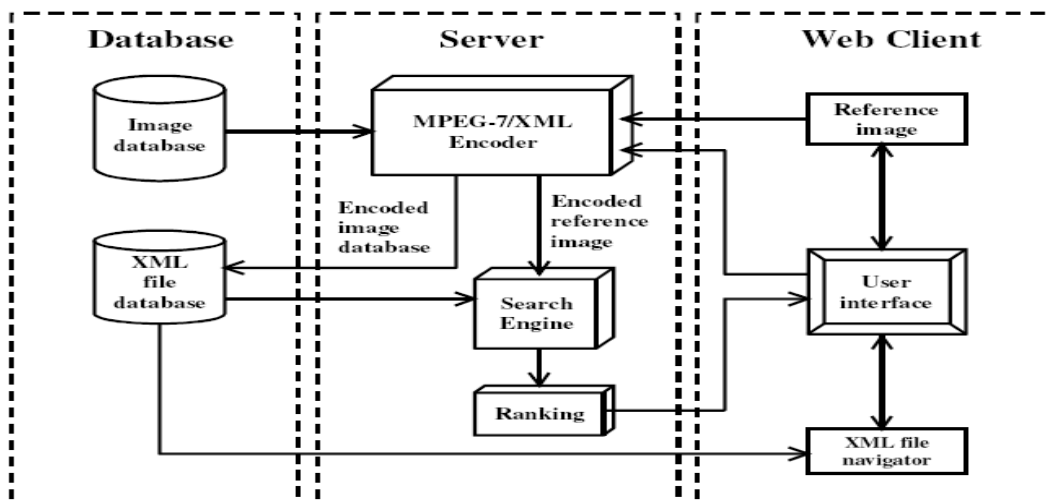


Figure 8 : Architecture de ERIC7 – Computer Reserch Institut Montreal

Source image : L.Gagnon, S. Foucher, V. Couaillier. *ERIC7: an experimental tool for Content-Based Image encoding and Retrieval under the MPEG-7 standard*, ([5])

On peut identifier facilement trois parties :

- La BD : ou les mages et les descriptions MPEG-7/XML sont stockées.
- Le serveur : la partie principale concernant le codage et la recherche. Les fichiers MPEG-7/XML sont générés selon les spécifications du l'utilisateur et en utilisant une version à jour de XM et des bibliothèques spécialisées pour le codage de bas niveau. Dans cette version, 15 propriétés peuvent être paramétrés (couleur, texture, contour, localisation spatiale, etc.). Le moteur calcule et normalise en utilisant une échelle de 0 à 1 la distance entre l'image de la requête et celle en cours d'être analysée.

- La partie utilisateur : une interface graphique (HTML GUI) permettant le codage, la recherche et la visualisation. L'utilisateur a deux possibilités : recherche basée sur l'exemple (QBE) ou de bas niveau (détaille avancé - clustering mode). De plus, l'utilisateur a accès directement à la structure XML étant muni d'un outil capable de générer des diagrammes UML.

Plusieurs possibilités sont offertes aux usagers, possibilités qu'on peut grouper en deux grandes catégories : des requêtes basées sur la technique QBE (query by exemple), et la recherche basée sur des attributs (clustering database). Une interface conviviale permet la navigation pour choisir les attributs et paramétrer la recherche. Les résultats obtenus peuvent être affichés selon leur pertinence et le logiciel leur associe en fichier XML et une structure UML correspondantes. Une application complexe est en déploiement, et elle est fortement basée sur l'utilisation de ERIC7. Il s'agit de MADIS, plus exactement MPEG-7 Audio-visual Documentation Indexing System.

Basé entièrement sur la décomposition des séquences vidéo et l'extraction des attributs vidéo et audio constituant la base des futures recherches basées sur le contenu, MADIS vise d'atteindre en quelques années un niveau de compatibilité totale avec le standard MPEG-7. ([5]).

La pertinence des résultats obtenus à la suite d'une requête est essentielle. La précision et le rappel sont-ils directement influencés par le nombre des descripteurs utilisés ? C'est la question qui se pose ([14]). Le système proposé par ([14]), se charge de localiser des vidéos et des images, en se basant sur des descripteurs visuels du standard MPEG-7. L'article présente également les résultats des expériences faites en considérant deux approches : celle qui utilise simultanément plusieurs descripteurs et une deuxième considérant seulement un seul descripteur.

Le standard MPEG-7, considéré véritable interface pour la description du contenu multimédia tente d'offrir l'élément clé dans la recherche efficace de l'information. Mais, son contenu est volumineux, assez compliqué et pas facile d'aborder pour les applications pratiques. Des efforts sont faites partout dans le monde pour développer des systèmes capables d'utiliser de façon pratique, efficace, les spécifications théoriques du standard. On peut considérer que l'organisation du MPEG-7 couvre 7 grandes zones principales, selon l'image suivante :

MPEG-7 standard	Contents
Part-1 : System	Specifies the tools that are needed to prepare MPEG-7 descriptions for efficient transport and storage to allow synchronization between content and descriptions and the tools related to managing and protecting intellectual property
Part-2 : DDL (Description Definition Language)	Specifies the language for defining new description schemes
Part-3 : Visual	Specifies the descriptors and description schemes dealing exclusively with visual information
Part-4 : Audio	Specifies the descriptors and description schemes dealing exclusively with audio information
Part-5 : MDS (Generic Entities and Multimedia Description Schemes)	Specifies the descriptors and description schemes dealing exclusively with generic and multimedia features
Part-6 : Reference Software	Includes software corresponding to the tools included in the standard
Part-7 : Conformance Testing	Define guidelines and procedures for testing conformance of MPEG-7 descriptions and terminals

Figure 9 : Organisation MPEG-7

Source:Jae-Ho Lee, et al. , *Video/image retrieval system based on MPEG-7 (VIRS)*, [14]

Le système VIRS (*video image retrieval system*) ([14]) est basé sur l'utilisation des descripteurs spécifiques à la troisième partie, celle qui traite l'information visuelle du multimédia et vise à réaliser le repérage automatique sans aucune intervention ou annotation de l'utilisateur.

Plusieurs descripteurs couvrant la couleur, la texture et la forme ont été considérés. Étant donné la complexité des expériences et la difficulté liée au travail avec trop de variables, l'analyse des descripteurs de la couleur a été identifiée potentiellement ayant le meilleur poids pour le but visé. La base de données (*CCD – Common Color Dataset*) est formée de 5466 images, divisées en 5 groupes. Le système permet deux types de requêtes, QBE (*Query by Example*) et QBD (*Query by draw*). Pour QBE on a utilisé trois descripteurs visant la couleur (*DCD – Dominant Color Descriptor*, *CLD – Color Layout Descriptor*, *CSD- Color Structure Descriptor*), et pour QBD seulement un. (*CLD – Color Layout Descriptor*). Il a fallu normaliser la valeur de similarité de chaque descripteur (de 0 à 1) selon l'équation :

$$\text{normalized } D_{i,j} = \frac{D_{i,j} - D_{\min}}{D_{\max} - D_{\min}}$$

where the distance value of image i and image j is $D_{i,j}$,
the minimum distance value in database is $D_{\min}$ and
the maximum distance value in database is $D_{\max}$

Source: Jae-Ho Lee, et al. , *Video/image retrieval system based on MPEG-7 (VIRS)*, [14]

Si on considère simultanément plusieurs descripteurs pour l'analyse de la similarité, on a :

$$\text{integrated } D_{i,j} = \sum_k^{\text{utilized descriptor}} w_k \bullet \text{normalized } D_{i,j}$$

where the weight of kth descriptor is w_k

$$\text{and } \sum_k^{\text{utilized descriptor}} w_k = 1$$

Source: Jae-Ho Lee, et al. , *Video/image retrieval system based on MPEG-7 (VIRS)*, [14]

La conclusion des expériences faites sur la BDMM, en utilisant des QBE et QBD avec 1, 2 ou 3 descripteurs ont montré que la précision du repérage augmente avec le nombre des descripteurs utilisés. Mais, la recherche basée sur un seul descripteur peut offrir des perspectives intéressantes, si une analyse préalable est faite de façon rigoureuse. Par exemple, si la BDMM est de grand volume, on peut utiliser d'abord DCD (*Dominant Color Descriptor*) pour faire un filtrage préliminaire. Une fois le filtrage exécuté on peut travailler sur la similarité en utilisant un autre descripteur, le CSD (*Color Structure Descriptor*).

MPEG-7 [21], spécifie une description standard de différents types d'informations multimédia, associée au contenu lui-même pour permettre la recherche rapide et efficace des informations qui intéressent l'utilisateur (images, graphiques, audio, vidéo). La description peut être attachée à n'importe quel matériau multimédia, quel que soit le format de la représentation. Le format MPEG-7 est complètement indépendant de la technique de codage ou de stockage du contenu du document multimédia.

Au-delà des informations sur le contenu, MPEG-7 permet d'inclure des informations concernant la création, le format (codage, compression, etc.), les conditions

d'accès et d'utilisation du matériel (copyright, prix, liens utiles, etc.). Le tableau suivant synthétise les 5 types des éléments d'un document MPEG-7 :

Éléments	Description
Création et production	Des méta-informations décrivant :le titre, le créateur, le but de la création.
Utilisation	Des méta-informations reliées à : droits d'accès, des informations financières, des droits de publication.
Média	Caractéristiques de stockage: Format, éléments d'identification.
Structure	Informations sur les composantes spatiales, temporelles ou spatio-temporelles du contenu audio-visuel. (la couleur, la texture, la forme, la motion, caractéristiques audio, etc.)
Concepts	Descriptions du contenu d'un point de vue conceptuel (en cours de standardisation).

La description MPEG-7 est basée sur XML, étant donc constituée d'objets imbriqués. La structure arborescente écoule d'un schéma descripteur établi à partir d'un LDD (langage de définition des descriptions) manipulant deux éléments : les descripteurs (D) et les schémas descripteurs (SD). L'image suivante donne une représentation de l'hiérarchie MPEG-7 :

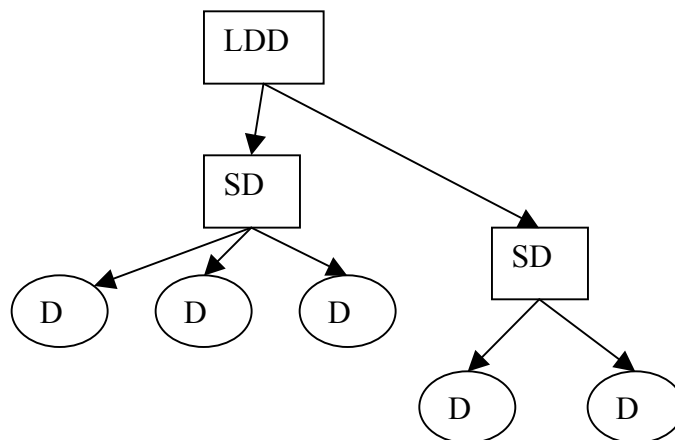


Figure 22: Hiérarchie MPEG-7

Les notions véhiculées par MPEG-7 sont :

Les descripteurs (D) - des présentations de caractéristiques des éléments (définissent la syntaxe et les sémantiques de chaque représentation)

Les schémas descripteurs (SD) - spécifient la structure et les sémantiques des relations entre les composantes (soit des descripteurs soit des autres SD)

Le langage de définition des descriptions (LDD) – fournit les outils nécessaires pour l'administration et la manipulation des D et SD.

Pour le présent projet, on s'intéresse aux descripteurs visuels offerts par MPEG-7, spécialement aux descripteurs traitant la couleur.

Les descripteurs de couleur MPEG-7 sont les suivants [2], [4] :

1. descripteur espace de couleur (**ColorSpaceDescriptor**) – outil permettant de préciser l'espace de couleur dans lequel on exprime les autres descripteurs (RGB, YCbCr, HSV, HMMD)
 - a. RGB – red, green, blue
 - b. HSV – hue, saturation, value (composition spectrale, pureté de la couleur, niveau de luminance)
 - c. YCbCr – luminance, différences de couleurs
 - d. HMMD – Hue, Min, Max, Diff – ici S est remplacée par Diff (différence entre deux valeurs, maximale et minimale des trois composantes primaires R, G, B)
2. Descripteur quantification de Couleur (**ColorQuantizationDescriptor**) – outil permettant de quantifier chaque composante de l'espace couleur (R, G, B, H, S, V, Y, Cb, Cr, etc.)
3. Descripteur histogramme de couleur (**ColorHistogramDescriptor**) – traite les fréquences relatives d'apparition des couleurs :
 - a. $H(i) = \text{Nombre pixels couleur } i / \text{Nombre total pixels}$
4. Descripteur histogramme par transformé de Harr (**ScalableColorDescriptor**) – permet de déduire des variantes de l'histogramme à divers résolutions.
5. Descripteur histogramme de couleur d'un ensemble de frames (**GoF/GoP Histogram Descriptor**) – s'occupe de la nature dynamique des frames. (ensemble d'images).
6. Descripteur histogramme couleur-structure (**CS HistogramDescriptor**) – introduit la cohérence spatiale locale, par l'utilisation d'un masque binaire pour l'image analysée.
7. Descripteur couleurs dominantes (**DominantColorDescriptor**) – considère un nombre réduit (maximum 8) de couleurs représentatives pour l'image et quantifie leur apparition (nombre, valeurs et fréquences relatives)
8. Descripteur distribution spatiale de couleur (**ColorLayoutDescriptor**) – capture la manière dont les couleurs sont réparties spatialement dans l'image. On divise l'image en 64 blocs, chacun représenté par sa dominante. On obtient 3 matrices 8 x 8 pour les trois couleurs de base.

2.2.4 Annotation

Étant donné l'importance croissante des métadonnées pour le média numérique, beaucoup d'initiatives pour leur standardisation ont démarré les dernières années. Selon [1], plusieurs effets immédiats sont facilement identifiables :

- la standardisation permettra aux détenteurs de multimédia d'engager des affaires électroniques (changement des descriptions standardisées)
- développement des autres services connexes (moteurs de recherche)
- appui pour l'annotation média et l'extraction automatique du contenu.

On peut considérer deux grandes classes de standards pour les métadonnées ([1]) :

- les schémas de description : MARC, DublinCore, CDWA, etc.
- les cadres (environnements) de travail : PICS, MCF, RDF, etc.

L'exemple détaillé en ([7]), propose IFINDER, un système d'indexation et de repérage basé sur la technologie XML et sur celle de la base de données open source. Le système est capable d'extraire automatiquement des métadonnées à partir du contenu audio-visuel. On peut accéder aux données en utilisant une application basée sur la technologie client serveur. L'application permet la recherche et le repérage dans une BDMM étant à présent utilisée pour l'indexation des dissertations tenues dans le parlement d'Allemagne.

IFINDER se distingue parmi les autres systèmes de recherche et repérage basé sur le contenu par une adaptation très ciblée modules d'extraction (reconnaissance de la parole et du visage) et par l'utilisation des composants "open source", (comme Xindice XML database, Micro Corba -implementation, ISIP – outil de reconnaissance de la parole) permettant une réduction importante des coûts d'implémentation. En ce qui concerne les algorithmes utilisés, on a ([7]):

- Analyser la partie audio : basée sur la réalisation d'une correspondance entre texte (sténogrammes) et séquences audio. . L'idée principale est la génération des FSA (finite state automata) pour toutes les parties audio à partir des sténogrammes, et les considérés comme modèles de langages pour la recherche de similarité avec les données audio existantes.
- Analyser la vidéo : basée sur l'extraction des informations sémantiques à partir de la composante temporelle de la séquence vidéo. La segmentation est basée sur le changement dans l'histogramme de couleur pour des cadres successifs. dans ces segments on applique un système capable d'identifier le visage (reconnaissance axée sur la composante spatiale). Seulement les

cadres qui contiennent des vues “de visage direct” sont considérés comme candidates pour le système de reconnaissances des visages étant annotées comme cadre clés pour le repérage.

Architecture : système distribué basé sur l'architecture client serveur.

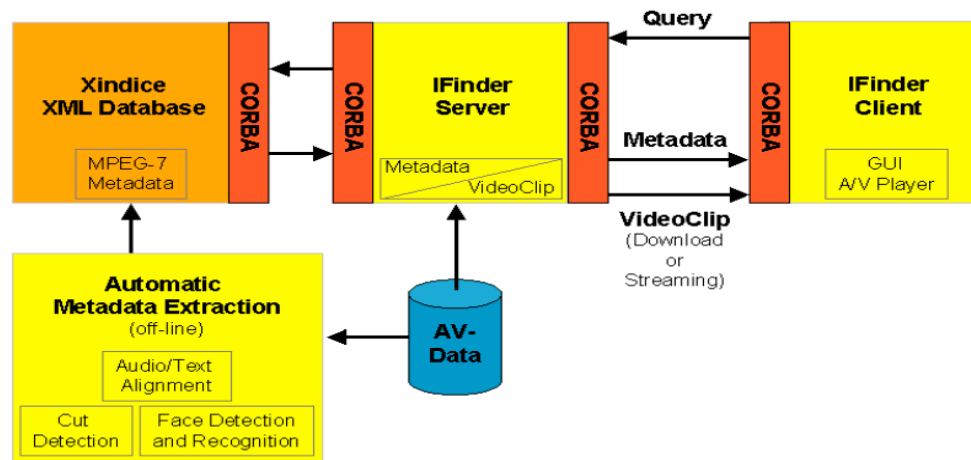


Fig. 2: The iFinder system architecture.

Figure 10 : Architecture de iFinder

Source image : J. Löffler, K.Biatov, et al. *IFINDER: an MPEG-7-based retrieval system for distributed multimedia content*, ([7])

Les composantes principales sont :

- application serveur
- application client
- la base de données XML (stockage des métadonnées)
- module d'extraction des métadonnées

L'application client permet les fonctionnalités suivantes : connexion au serveur, choix et saisie des paramètres pour la requête, interface graphique pour observer et naviguer parmi les résultats renvoyés par le serveur. De même, l'interface permet de rouler le film voulu.

La recherche dans la BDXML est fait à partir des données choisis ou saisis par l'utilisateur et envoyés vers le serveur. Celui-ci utilise ces données pour bâtir une chaîne de caractères pour la requête (XPath query string). Le SGBD retourne au serveur le résultat de la requête sous forme de document XML, chacune des métadonnées considérées comme pertinente pour la recherche étant des éléments dans ce document

XML. Une tri selon la pertinence suit, et le résultat est envoyé vers l'application client à travers l'interface graphique.

Implémentation : La solution utilisée pour la réalisation du système est basée sur le langage IDL (interface définition langage) du CORBA. Celui-ci a été utilisé pour réaliser les interfaces entre le serveur, le client et la BDXML. Le serveur a été réalisé en utilisant C++ et le client a bénéficié de l'apport de Java. (GUI in Java Swing pour la saisie des données utilisées pour les requêtes, JMF (Java Multimedia Framework) pour l'intégration des lecteurs audio et vidéo. Pour la BDMM on a choisi Xindice, une BDXML de type open source réalisée par Apache, permettant la connexion au serveur et le téléchargement du contenu similaire aux requêtes envoyées par les usagers.

Un système de recherche et repérage de l'information MM, basé sur l'annotation mixte (à la fois manuelle et automatique) est présenté en ([10]). L'article propose un système capable de repérer des données multimédia concernant le jeu de bowling. Le système permet la recherche basée sur le contenu multimédia, ayant comme critères l'information provenant de trois sources principales : information sur le contenu vidéo du match, l'information reliée au match proprement dit et celle concernant le joueur. Pour représenter ces types d'informations et leur inter liaisons les schémas de descripteurs MPEG-7 ont été utilisés. Pour réaliser l'annotation il a fallu utiliser à la fois une manière manuelle (pour le joueur et les détails sur le jeu, considérées comme des informations conceptuelles) et automatique (extraction automatique des attributs concernant la vidéo, dans ce cas la trajectoire des objets en mouvement). Un schéma du système est la suivante :

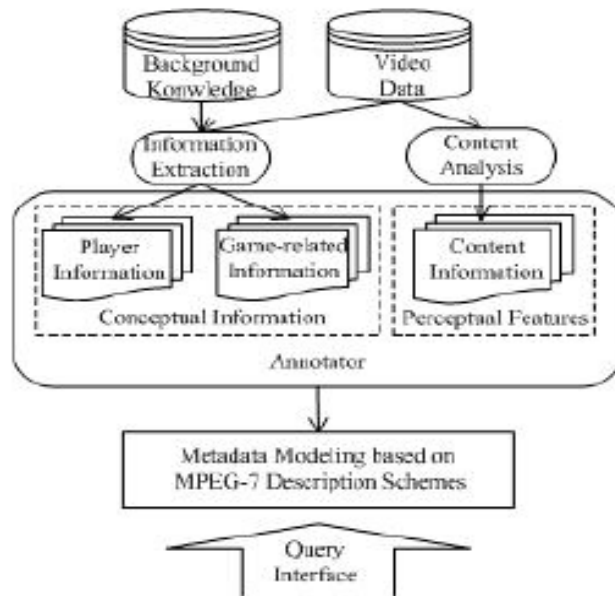


Figure 11 : Système de recherche multimédia (annotation mixte)

Source image : W. W. Hsieh, et al *Constructing a bowling information system with video content analysis*, [10]

Un des plus importants événements est considéré le moment de la frappe de la cible. C'est la raison pour laquelle le système sauve des vidéos et des cadres associés aux événements pour utilisation ultérieure.

Une interface graphique permet à l'utilisateur, professionnel ou débutant en bowling d'introduire les éléments permettant de personnaliser sa requête et d'observer le résultat de la recherche. Étant donné l'intérêt des usagers pour certaines événements et cadres, le système permet une méthode d'analyse des propriétés audio et visuelles pour l'extraction automatique de l'information.

Plus précisément, ce type d'information complète celle disponible à partir des données existantes sur les joueurs et les lieux. Toutes les données sont sauvegardées dans des fichiers XML, pour assurer l'interopérabilité caractéristique au standard MPEG-7 (XML est par ailleurs le format choisi pour représenter le LDD, le langage de définition des données de MPEG-7).

L'annotation est faite donc de façon mixte, semi-automatique, en cumulant les données saisies manuellement (informations sur les joueurs, les lieux) et les informations extraites directement du flux vidéo (séquences vidéo, moment du frappe, segmentation de la trajectoire du ballon, analyse du dernier cadre, celui qui donne le résultat de la frappe).

La façon dont les annotations sont faites a des implications directes sur la performance du repérage des informations multimédia. C'est la raison pour laquelle ([12]) propose un système d'annotation spécialement conçu pour enrichir le contenu du multimédia. Le système M4Note c'est un outil multimodal d'annotation multimédia, basé sur le standard MPEG-7. Les annotations sont considérés comme des objets multimédia, et le système génère automatiquement les documents associées, contenant à la fois les annotations, les attributs du contenu et les éléments de liaisons entre les deux. .

L'interface multimodale de M4Note est conçue en Java et permet simultanément la capture vidéo et l'annotation. Cette dernière peut être faite soit à l'aide d'un stylo électronique, soit à travers un système de reconnaissance de la parole. Dans les deux cas, les annotations sont converties en texte. Comme sortie, le système offre des documents XML, générés de façon automatique et contenant les éléments MM capturées : vidéo, audio, images, cadres, annotations par encre, texte.

L'interface graphique permet de sélectionner le cadre dont on veut ajouter l'annotation, et offre aussi des possibilités de faire l'ajout de façon manuelle ou en utilisant la parole. Dans les deux cas les annotations sont synchronisées dans le flux MM et des fichiers XML associées sont générés automatiquement.

Pour chaque session de capture vidéo et audio, le système M4Note utilise le JMF (Java Media Framework API) et analyse les flux multimédia pour générer des documents XML associés aux attributs extraites de façon automatique, comme la figure suivante :

```
<Mpeg7 ... xmlns: ...
  <MediaFormat>
    <Content>Local frame object</Content>
    <FileFormat>MPEG-4</FileFormat>
    <Size>640x480</Size>
    <FrameRate variable="false">30</FrameRate >
    <Frame>3553</Frame>
    <MediaTimePoint>00:00:12</MediaTimePoint>
  </MediaFormat>
</Mpeg7>
```

Figure 12 : Exemple fichier XML (application M4Note)

Source image :R.Goularte, R. G. Cattelan, et al. *Interactive multimedia annotations: enriching and extending content*, [12] .

En ce qui concerne les annotations faites oralement, le système est muni d'un module de reconnaissance et synthèse de la parole, qui utilise JSAPI (Java Speech API, qui fournit des spécifications pour définir une interface d'accès) interfaçant avec le module ViaVoiceTM d'IBM. (moteur de reconnaissance et synthèse vocale).

Les annotations permettent également d'enrichir le contenu multimédia. Les objets média capturés peuvent être liés utilisant une instanciation d'un élément de connexion : XLinkObject, définit à l'intérieur de chaque objet de type MediaObjet. Par exemple, une annotation faite avec d'encre électronique et le texte associés appartenant au document *document.xml* peuvent être reliés comme objets multimédia de façon suivante, en utilisant les pointeurs XPE1, XPE2, les éléments de localisation (*Locator*), et la relation (*Means*) :



```

<MediaObject> ...
  <XLinkObject xmlns:xlink="http://www.w3.org/2000/10/xlink-ns"
    xlink:type="extended" xlink:title="Objects' rels">
    <Locator xlink:type="locator"
      xlink:href="file://path/document.xml#XPE1"
      xlink:role="file://path/object"
      xlink:label="ID004"> <!-- Stroke -->
    </Locator>
    <Locator xlink:type="locator"
      xlink:href="file://path/document.xml#XPE2"
      xlink:role="file://path/object"
      xlink:label="ID005"> <!-- Text -->
    </Locator>
    <Arc xlink:type="arc"
      xlink:from="ID004" <!-- Stroke -->
      xlink:to="ID005" <!-- Text -->
      xlink:arcrole="file://path/relations/Means">
    </Arc> ...
  </XLinkObject>
</MediaObject>

```

Figure 13 : Exemple annotation encre électronique (application M4Note)

Source image : R. Goularte, R. G. Cattelan, et al. *Interactive multimedia annotations: enriching and extending content*, [12] .

Les schémas de description du standard MPEG-7, les SDM, sont à la base du système analysé en ([13]).

On nous propose un outil de description sémantique du contenu multimédia conçu comme une extension de l'outil StructuredAnnotation de MDS (Multimedia Description Schemes) du standard MPEG-7. L'outil permet l'annotation du contenu multimédia en utilisant 12 attributs à la fois aux caractéristiques "statiques" et "dynamiques" du document analysé. Les douze attributs sont choisis pour s'aligner à la syntaxe 5W1H+3D concernant le document analysé, ou :

5W1H = *who, what, when, where, why, how*

3D = *direction, distance, duration*

La définition du contenu MM combinant la vidéo, les images, la musique et les douze attributs proposés est faite dans le système DDDC (Dozen Dimensional Digital Content). Le système est basé sur le MDS (*Multimedia Description Scheme*) de MPEG-7.

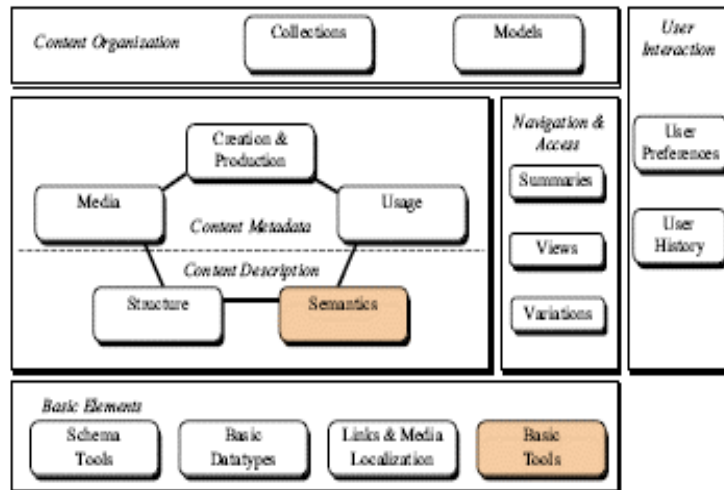


Figure 14 : Schéma de description multimédia de MPEG-7

Source image: Kuo, P.-J., et al, Tokyo Univ., Japan, **MPEG-7 based dozen dimensional digital content architecture for semantic image retrieval services**, [13]

La nouvelle sémantique proposée par le système est basée sur l'utilisation d'outil *TextAnnotation*, type de donnée de base de MDS de MPEG-7. Une des formes possibles de *TextAnnotation* est *StructuresAnnotation* qui sera utilisée pour l'application. C'est une structure qui permettra la description de l'annotation du contenu MM en répondant aux questions proposées aux début (quel acteur -who, quel objet/action-what, ou – where, quand-when, pourquoi –why, comment-how, etc.)

Les images suivantes montrent la source analysée (une image prise dans un café-bar) et le *TextAnnotation* associé, qui spécifie à la fois la description sous la forme de *FreeTextannotation* (sans structurer l'information) et sous forme de *StructuredAnnotation* (identification structurée des sous composantes) :

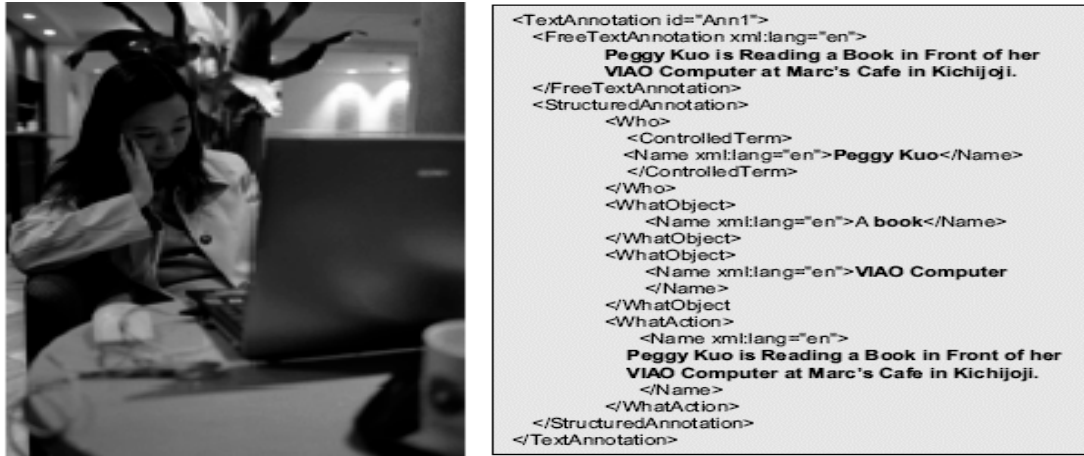


Figure 15 : Exemple utilisation TextAnnotation (MPEG-7)

Source image: Kuo, P.-J., et al, Tokyo Univ., Japan, *MPEG-7 based dozen dimensional digital content architecture for semantic image retrieval services*, [13]

La génération semi-automatique des métadonnées DDDC (*Dozen Dimensional Digital Content*) a comme entrées plusieurs sources : la technologie GPS, une BD de localisation géographique, des logiciels spécialisés et des instruments d'enregistrement digital. Ainsi, le GPS, la direction, la distance, le temps et l'auteur peuvent être associés avec le fichier digital à l'extension Exif (*Exchangeable Image File Format*). Les informations spatiales et temporelles (étiquettes associées aux événements) peuvent servir comme données d'entrée pour des requêtes vers la BD. Ayant ces informations, la BD associe à l'image les autres étiquettes sémantiques.

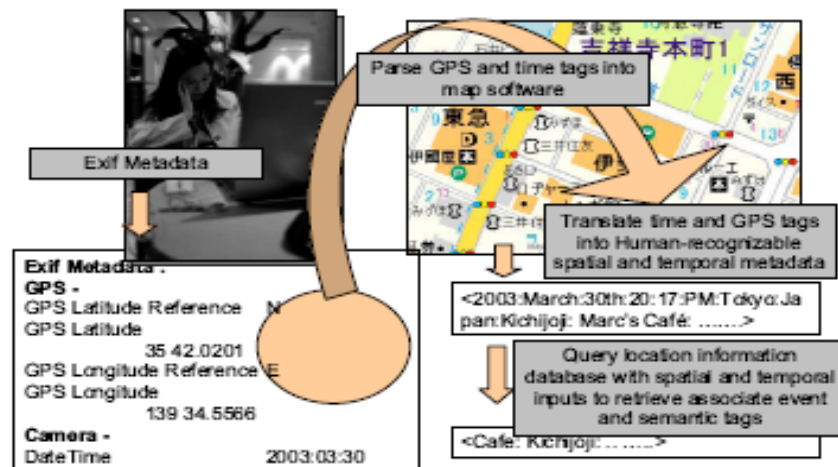


Figure 16 : Exemple utilisation métadonnées (interrogations sur la BDMM)

Source image: Kuo, P.-J., et al, Tokyo Univ., Japan, *MPEG-7 based dozen dimensional digital content architecture for semantic image retrieval services*, [13]

2.2.5 CBIR

Sirsale, le système CBIR développé en [4], et qui réalise une recherche et repérage des séquences audio-visuelles, utilise une approche modulable, multi niveaux :

- indexation structurelle (exploitation de la structure de la vidéo) pour la navigation
- indexation sémantique, pour la recherche par le contenu.

La figure suivante donne un aperçu de cette stratégie utilisée :

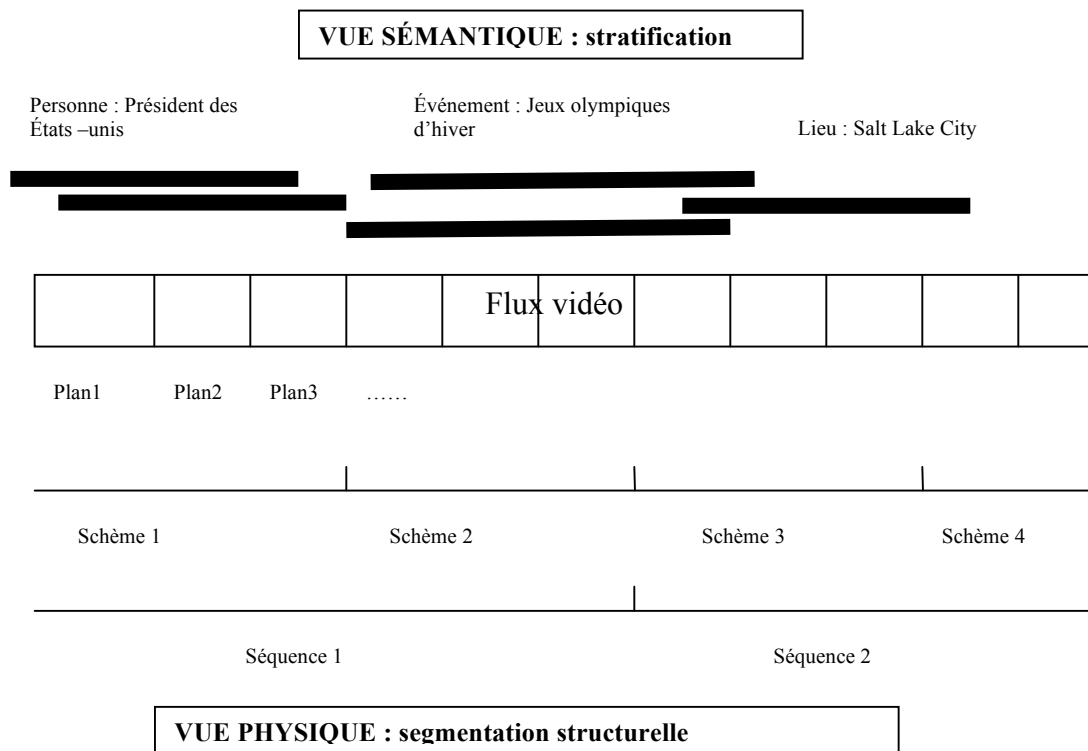


Figure 17 : approche système SIRSALE [4]

Dans la figure antérieure, la vue physique réfère la segmentation structurelle ou :

- plan (*shot*) : unité composante de la vidéo, composée à son tour d’un ensemble continu d’une ou plusieurs images fixes. (*frames*). Même si le plan peut être considéré comme un premier niveau de découpage de la vidéo, son contenu sémantique est insuffisant pour l’indexation.
- Schème : c’est l’organisation pertinente des plans (plusieurs plans reliés dans le temps ou l’espace par leur contenu sémantique). On peut la considéré comme le premier niveau sémantique.

- Séquence : Si la longueur de la vidéo ou la richesse en contenu est élevée, les scènes sont limitatives pour l'indexation. En regroupant les scènes en séquences, selon le lien temporel ou spatial on obtient une amélioration du repérage par le contenu.

Quant à la vue sémantique, elle utilise une approche qui divise le document en "strates", selon le contenu audio-visuel. Ces « strates » sont indépendantes de la structure de la vidéo, pouvant donc se superposer pour un instant donné. Alors, on a une réunion de "strates" qui définit l'état. Cela permet une indexation plus fine et donc plus efficace pour la recherche par le contenu. De point de vue technique l'application a été réalisée à l'aide de Java, Java RMI, JMF, MySQL. Quant à l'utilisation, plusieurs étapes sont prévues :

- Client → requête (thème choisie) → ST (serveur de thèmes)
- ST → GUI (personnalisation, furetage, interrogation)
- GUI → SMD (serveur métadonnées)
- SMD → GUI (réponse)

Le client choisit donc un thème, le serveur des thèmes analyse la requête et lui retourne une interface graphique personnalisé, lui permettant le fureter la base de données spécifique au thème ou construire des requêtes en spécifiant des attributs. Cette fois, c'est le serveur des métadonnées qui se charge d'analyser la requête (c'est une recherche et un repérage CBIR) et de retourner la réponse au client.

Un des domaines les plus dynamiques de la technologie de l'information, l'utilisation de WEB, a incité le développement des plusieurs systèmes de repérage basé sur le contenu de l'information.

Les plus importantes sont illustrés dans le tableau suivant ([8]):

System	Main characteristic	Descriptors	Query style	References
ImageRover	Experimented on 100,000 images	—Pictorial —HTML text	—textual —By example images	Sciaroff [1995] Sciaroff et al. [1997]; Taycher et al. [1997]; La Cascia et al. [1998]
Diogenes	Specific for celebrities' faces	—Human faces —HTML text	—Textual: celebrity's name	[Aslandogan and Yu 2000a, 2000b]
Atlas WISE	Combines positive and negative examples	—Pictorial —HTML text	—Textual —By example images	Kherfi et al. [2003a, 2003b]
WebSeer	Distinguishes photographs from graphics	—Pictorial —HTML text	—Textual enriched with image attributes	Athitsos and Swain [1997]; Frankel et al. [1996]; Swain et al. [1997]
WebSeek	Retrieves images and videos	—Pictorial and image attributes —Page URL	—Textual —By example images	Smith and Chang [1996a, 1997]
ImageScope	Retrieves images from intranets and the WWW	—Image content —Image tag text	—Textual —By icons and sketches	Lew [2000]; Lew et al. [1997]
PicToSeek	Distinguishes photographs from graphics	—Pictorial	—By example images	Gevers et al. [2000]; Gevers and Smeulders [1997, 1999]
WebMars	Retrieves whole HTML documents	—Pictorial —HTML text	—Textual —By example images	Ortega-Binderberger et al. [2000]
Web-WISE	Allows queries by estimated feature values	—Pictorial	—By example images —By estimating features' values	Wei et al. [1998]
RIME	Replicated image detector	—Pictorial	—By requested image	Chang et al. [1998]
DrawSearch	Allows queries by sketch	—Pictorial	—By sketch or by texture content	Di Sciascio et al. [1999]
Monash	Combines images retrieved by text and those by content	—Pictorial —HTML text	—Textual —By example images	Lu and Williams [1999]

Figure 18 : Systèmes de repérage basés sur le contenu (WEB)

(Source image : M. L. Kherfi, et al. *Image Retrieval from the World Wide Web: Issues, Techniques, and Systems*, [8])

Comme on peut observer, deux grandes approches ont été utilisées dans l'implantation des systèmes de repérage des images :

- annotation des images avec du texte, et l'utilisation de celui –ci pour des techniques de repérage basé sur l'analyse textuelle : c'est TBIR (texte based image retrieval) ou recherche d'image basée sur le texte (utilisée par exemple par Google ou Altavista)
- utilisation des attributs de l'image, comme la couleur ou la forme. Il s'agit cette fois-ci de CBIR (*content based image retrieval*).

Dans chacune des deux cas, la recherche est basée sur des requêtes bâties à partir des données saisies par les usagers. Dans le cas de TBIR, les attributs textuels prennent la forme des mots clés, recherchés par le moteur parmi les mots pertinents associés aux images. Plusieurs techniques ont été développés (étiquette image, titre image, hyperlien

image, mot clés dans le texte associé). Mais, c'est une méthode qui laisse porte ouverte à une grande marge d'erreur car, la plupart les annotations sont faites manuellement, donc de façon subjective.

Il s'impose donc de trouver des outils pour l'annotation automatique des images. Mais, même avec ces outils, l'utilisateur est souvent intéressé dans des propriétés difficilement à décrire en mots. C'est le cas de CBIR, et cette fois les attributs vont faire référence directement aux propriétés de bas niveau : couleur, orientation, texture, forme, contour. C'est un véritable défi, et, selon le but visé il faut reconsidérer les attributs utilisés comme éléments clés. (Pour l'analyse des empreintes la texture ne donne pas les meilleurs résultats, la reconnaissance du visage exige plus que l'attribut forme, etc.) Souvent, c'est la combinaison des plusieurs attributs celle qui nous approche de la réussite. À titre d'exemple voila quelques systèmes de recherche images et les attributs utilisés :

System		Image Rover	Diogenes	Atlas WISE	WebSeer	WebSeek	ImageScape	PicToSeek	WebMars
Keywords		*	*	*	*	*			*
Face detection			*		*		*		
Layout		*					*		*
Shape	Fourier descriptors						*		*
	Elementary description						*		
	Angles between edges, and cross ratios of them							*	
Texture	Wavelet, Fourier transform								*
	Edge-orientation histogram	*		*					
	Local binary patterns						*		
	Automatic texture features	*					*		*
Color	Eigen image								
	Dominant colors						*		
	Region histogram								
	Fixed subimage histogram	*							*
	Farthest neighbor histogram				*				
	Global histogram	*		*	*	*		*	*
Laplacian							*		

Figure 19 : Exemples des attributs utilisés pour la recherche des images

Source image : M. L. Kherfi, et al. *Image Retrieval from the World Wide Web: Issues, Techniques, and Systems*, [8]

Chaque fois l'utilisateur envoie sa requête, le système cherche dans la base de données multimédia les informations ayant des caractéristiques comparables à celles spécifiés. Souvent, l'utilisation d'un index s'impose pour réduire le temps de la recherche. La réponse prend la forme d'une liste des éléments semblables, qui s'impose être trié selon la similarité avec la cible (la requête). Si dans le cas des TBIR l'utilisateur introduit des mots clés, pour la recherche CBIR les choses se compliquent, car :

- c'est difficile d'introduire directement la couleur, la texture, la forme

- plusieurs usagers ne sont pas familiarisés avec ces notions
- même pour les spécialistes c'est difficile d'estimer avec exactitude les poids des attributs identifiant la cible.
- Une technique plus conviviale est d'utiliser une image comme modèle, comme matrice pour la recherche. L'utilisation de cette interface QBE (*query by example*) apporte cependant des inconvénients, car, tout dépend de l'image choisie pour le démarrage de la recherche ("page zero problem").

C'est évident donc qu'une standardisation s'impose. C'est un besoin vital de pouvoir uniformiser la description des images et de standardiser les descripteurs utilisés pour personnaliser nos recherches. C'est le point de départ pour données aux moteurs de recherche des jalons relevant, permettant une localisation précise, objective, et rapide. Plusieurs standards pour la définition du multimédia ont été adoptés les dernières années :

- RDF (*resource description framework*) – proposée par World Wide Web Consortium W3C, (les schémas RDF utilisent comme langage XML)
- Le set de métadonnées Dublin Core – métadonnées (auteur, titre, mots clés sur le sujet, le type du document, le format du document)
- MPEG-7 – (standardisation du contenu de l'information MM)

Le cas de repérage de l'information MM décrit en ([11]) fait une comparaison intéressante entre deux systèmes de recherche, l'un muni par des outils de recherche basé sur le contenu.

Les deux systèmes sont basés sur le même principe, les différents existant seulement au niveau des outils fournis au usager. Il s'agit plutôt des deux variantes du même système, l'une qui incorpore des attributs de bas niveau et le système de reconnaissance automatique de la parole (ASR – *automatic speech recognition*) et l'autre bénéficiant seulement d'ASR.

L'architecture du système, basé sur XML et le standard MPEG-7 est la suivante ([11]) :

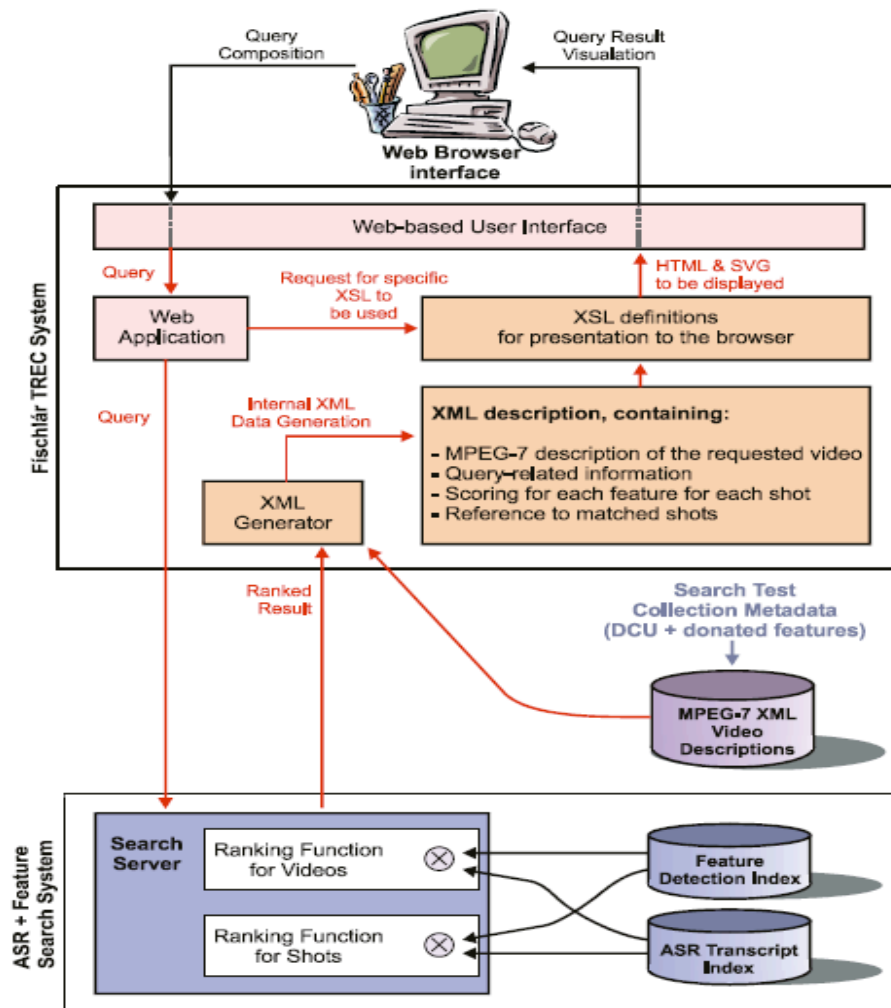


Figure 1. Architecture of Físchlár-TREC2002

Figure 20 : Exemple système dual de repérage MM

Source image : G. Gaughan, et al. *Design, implementation and testing of an interactive video retrieval system*, [11]

L'utilisateur saisit et envoie une requête vers le système à travers l'interface graphique. Cette requête sera traitée par l'application Web qui va envoyer une demande au moteur de recherche. Celui-ci traite la demande et retourne le résultat obtenu sous forme trié selon la compatibilité avec la demande vers un générateur XML basé sur la structure de standard MPEG-7. Les descriptions XML sont générées automatiquement et utilisées par XSL (extensible Stylesheet Language) pour fournir le code HTML et le SVG (Scalable Vector Graphics) nécessaires pour afficher l'information à l'interface utilisateur. L'utilisateur compose la requête en utilisant soit des attributs soit du texte, soit une combinaison des deux.

En utilisant la variante basée seulement sur l'ASR, on a en réalité un système de repérage basé sur l'analyse du texte. Il s'agit de chercher parmi les documents textes associés aux cadres des vidéos dans la BDMM. Quant au deuxième cas, celui qui combine l'ASR et les attributs, le système permet une recherche basée sur le contenu, en analysant les signatures des diverses vidéos de la BDMM. Chaque vidéo a sa propre signature, donné par la moyenne pondérée des poids des attributs existants dans les cadres le composant. L'identification de la vidéo s'approchant le plus de la demande est faite en utilisant la technique de fréquence inversée (*idf – invers document frequency*), permettant de mettre en valeur la pertinence des documents analysés. A la base on considère la relation qui donne le poids d'un attribut dans un document :

$$FW_{t_{vf}} = 1.0 + \log\left(\frac{N}{vf_f}\right)$$

Ou : $FW_{t_{vf}}$ = le poids de l'attribut f dans la vidéo v

N = le nombre total des vidéos dans la BDMM

vf_f = la fréquence de l'attribut f dans la vidéo

Ainsi, une liste des vidéos est offerte à l'utilisateur comme réponse à la requête posée. La position de chaque vidéo dans cette liste est déterminée par la somme des influences normalisées des attributs spécifiées au moment de description de la requête. Le deuxième pas est produit quand l'utilisateur sélectionne une vidéo parmi ceux existantes dans la liste. Alors, un algorithme de tri semblable (basé sur les poids des attributs) fonctionne pour discriminer les cadres appartenant à la vidéo sélectionnée.

2.3 Analyse du prototype proposé par l'université d'Autriche

2.3.1 références provenant d'étudiant de M. Haraldetc.

2.3.2 Commentaires sur l'analyse (actuellement en Annexe 1)

2.4 Sommaire

Le présent chapitre propose un survol sur les actualités concernant le repérage multimédia basé sur le contenu. La stratégie de recherche utilisée m'a permis de retracer des articles à la fois récentes et pertinents.

Plusieurs conclusions peuvent être tirées après ce survol sur les actuelles préoccupations pour le repérage de l'information multimédia :

- développement croissant des outils et techniques multimédia, apparition des nouvelles méthodes basées sur l'analyse des ontologies, de la sémantique.
- nécessité accrue concernant l'organisation, l'indexation, le repérage de l'information multimédia
- tendance de migrer vers l'annotation automatique du contenu (objectivité)
- nécessité d'assurer l'interopérabilité (standardisation, normalisation)
- volume croissant, complexité, interliaison et caractère hétérogène des données – nécessité des outils performantes, puissants. (bases de données relationnelles, programmation orienté objets)

Les articles choisis sont représentatifs pour les tendances actuelles concernant le multimédia : annotation mixte, (croissance de l'automatisation des extraction des attributs), détection similarité (analyse distance entre deux objets caractérisés par des attributs extraits en préalable) standardisation et normalisation (utilisation des fichiers XML compatibles MPEG-7). Pour le domaine à l'étude, plusieurs technologies sont souhaitables, étant déjà utilisé avec succès :

- SGBDMM – bases de données multimédia capables de réaliser le stockage et l'interrogation directe des fichiers MPEG-7 ou prévues d'une extension permettant d'implémenter le modèle MM et de générer des documents MPEG-7 (IPSIS, dbXML, XSU (XML SQL Utility d'Oracle), Xindice (Apache), MySQL, Oracle, etc.)
- Langages orientés objets permettant de construire des interfaces graphique (la saisie des données et la visualisation des documents multimédia) et des

applications web ou client serveur (C++, Java avec Java RMI, JMF, Swing, Awt, etc.)

- Annotation faite de façon semi-automatique (mélange annotation manuelle et automatique) ou totalement automatique, (extraction des informations directement à partir du flux vidéo) : JSAPI (Java Speech API, ViaVoiceTM d'IBM, etc.)

Deux approches principales ont été identifiées : l'une ayant comme point de départ des informations de bas niveau fournies par l'utilisateur, l'autre considérant comme entrant pour sa recherche un modèle (exemple), à partir duquel on extrait les informations nécessaires. (Dans la plupart des cas l'utilisateur exprime ses doléances sous le format des questions de haut niveau). Dans les deux approches, le moteur de recherche se comporte pareil, en calculant et normalisant selon une échelle quelconque la distance entre l'image de la requête et celle couramment analysée.

A cette lumière, mon choix technique a été :

- Base de données : Oracle10g, (facilités offertes par Intermedia - extension permettant la manipulation des données multimédia)
- Langage de programmation : Java (JDK 5.0, JMF, JAXP).

L'approche que j'utiliserai sera basée sur l'extraction automatique des attributs de bas niveau. Comme idée générale, l'analyse des images tirées de la vidéo portera sur l'analyse des changements dans l'histogramme de couleur pour des cadres successifs.

3. CHAPITRE 2 - Exécution

3.1 Introduction

L'environnement de travail choisi pour le développement de l'application est basé sur l'utilisation d'Oracle10g comme base de données. C'est l'extension Intermedia d'Oracle celle qui fournira des facilité à exploiter quant au stockage et la manipulation des images et de la vidéo.

Pour le développement du logiciel de l'application, on a choisit Java comme langage de programmation. Parmi les principales taches à accomplir : des interfaces graphiques pour la saisie des données et la visualisation de l'information, des algorithmes de calcul, des modules pour la connexion à la base de données, la création des tables, l'insertion et le repérage des données.

Trois modules seront réalisées : le premier permettant l'extraction des images à partir d'un fichier vidéo, un deuxième traitant le stockage des images en Oracle et l'identification des changements de scène et le dernier s'occupant de la visualisation des images et des documents MPEG-7 associés déposés dans la base de donnés.

Le dépistage des images identifiant des séquences sera faite en analysant les signatures des images successifs extraites de la vidéo et déposées dans la BD. Le résultat de la comparaison de la distance calculée entre deux images successives et un seuil donné encadrera l'image analysée dans l'une des deux catégories : image similaire avec ou différente de la précédente.

3.2 Préparation

L'étape d'exécution suit après une préparation complexe, couvrant à la fois la recherche documentaire, la lecture individuelle et les contacts professionnels. Ainsi, les lectures particuliers et le articles choisis à la suite de la recherche bibliographique m'ont permis d'identifier les principales préoccupations dans le traitement des images : standardisation et normalisation (MPEG-7), annotation automatique (extraction automatique des attributs de bas niveau) et CBIR (analyse basée sur ses attributs saisis ou extraits à priori). La pertinence de ces informations et de l'environnement de travail choisi a été confirmée à la suite du contact eu avec l'Université Klagenfurt (Autriche).

L'analyse du prototype de player multimédia basé sur la norme MPEG-7 réalisé dans le laboratoire dirigé par Ph.D.Harald Kosch, présentée dans l'annexe 1, m'a permis de mieux encadrer mon travail (environnement et objectifs).

L'environnement utilisé et le suivant :

- SGDBR Oracle10g – base de données
- Oracle Intermedia – extension Oracle pour la manipulation des données multimédia
- JDEV Oracle – environnement de développement Java fourni par Oracle. C'est véritablement facile à utiliser et assez performant
- Eclipse - environnement de développement intégré (*Integrated Development Environment*) fourni par IBM. (performant, rapide)
- Java – JDK 5.0 (*java developer kit*)
- JMF – Java Media Framework
- JAXP – (Java API for XML Parsing) – bibliothèque nécessaire pour le traitement des documents XML. Elle propose des implémentations des deux modèles : le modèle SAX et le modèle DOM.

Poste de travail utilisé :

- Processeur : Pentium 4, cadence : 2,4 Ghz
- Mémoire : 256 Mb RAM.

Quelques raisons concernant le choix de l'environnement technique et l'approche utilisée sont brièvement développées en ce qui suit:

Java, BD et XML

Plusieurs raisons ont justifié mon choix quant à l'utilisation de Java :

- langage multiplateforme (Linux, Unix, Windows, Solaris, etc.) - Les programmes Java sont portables. Leur exécution est indépendante de la plateforme d'installation (type de machine). Il suffit de disposer d'un environnement d'exécution Java pour l'exécuter.
- Il existe des environnements de développement Java gratuits
- Langage robuste et sécuritaire (plusieurs niveaux de sécurité)
- Langage fortement typé - beaucoup d'erreurs détectées automatiquement
- Langage orienté objet - programmation de haut niveau: orienté objet, exceptions, polymorphisme, gestion de la mémoire, etc.

- Accès aux BD - gestionnaire JDBC interne permet d'accéder aux données depuis Java.
- Préservation de SQL – transparence quant à l'utilisation des instructions SQL (et d'autres aspects des bases de données relationnelles)
- Sun a mis au point une interface API (*application program interface*) pour analyser les documents XML (le standard adopté par MPEG-7). Un document XML, (le métalangage permettant de représenter les données), peut être interpréter principalement en utilisant deux types de parseurs (analyseurs) : SAX (Simple API for XML) et DOM (Document Object Model) [17] [18].
 - SAX – ou Simple API for XML c'est la façon traditionnelle d'analyse XML, basée sur l'interprétation des événements rencontrés pendant la lecture séquentielle du fichier. Des fonctions de rappel (callback fonctions) sont générées au début et à la fin des événements rencontrés. C'est la responsabilité du programmeur d'écrire ces « callback fonctions. Pour Java, les classes Java correspondantes forment le paquetage : « *org.xml.sax* ».à
 - DOM - ou Document Object Model c'est une technique qui décrit le document XML comme une structure arborescente, ayant chaque élément XML comme un node. Cette fois, le document est lu entièrement et on forme en mémoire l'arbre correspondant. Pour Java, les classes correspondantes forment le paquetage *org.w3c.dom*. (implémentation de l'interface « *org.w3c.dom.Node* », qui permet des fonctions nécessaires pour explorer ou modifier la structure arborescente : (*getChildNodes()*, *appendChild()*, *removeChild()*...) et des méthodes pour interroger chaque node afin d'obtenir ses attributs.)

Oracle Intermedia

L'extension Intermedia d'Oracle 10g, [18], permet de manipuler des documents audio-visuels. Le paquetage ORDSYS Oracle fourni des classes (relationnel objet) pour stocker les documents multimédia. A titre d'exemple :

- **La classe ORDImage** : Supporte le stockage, la gestion et la manipulation des données image.

- **La classe ORVideo** : Le type ORVideo supporte le stockage et la gestion de fichiers vidéo.
- **La classe ORDAudio** :Le type ORDAudio supporte le stockage et la gestion de fichiers audio
- **La classe OrdImageSignature** : c'est une classe utilisée pour représenter une instance d'un champ Oracle de type *ORDSYS.ORDImageSignature* dans une application Java. Cette classe inclut une méthode permettant de générer la signature d'une image donnée et des méthodes statiques pour comparer deux images. (deux signatures):
 - **generateSignature()** – permet la création d'une signature pour une image donnée.
 - **isSimilar()** - La méthode réalise la comparaison des deux signatures en utilisant des poids spécifiés pour un ou plusieurs attributs visuels (descripteurs de bas niveau). Si le résultat de la comparaison est plus petit ou égal au seuil fixé, les images sont considérées identiques et la méthode retourne la valeur 1, au cas contraire la valeur retournée étant 0.

Pour l'application présente, on a utilisé ORImage et OrdImageSignature. L'image suivante donne une représentation de la liaison directe entre les objets ORImage de Oracle et Java à travers la connexion JDBC :

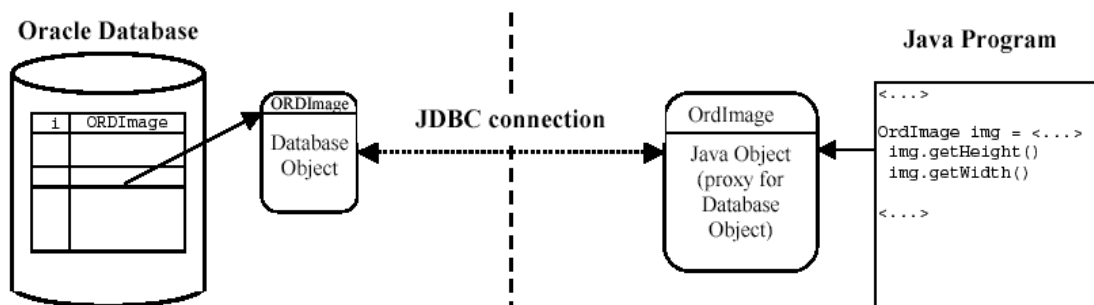


Figure 21 : Liaison entre les objets ORDSYS.ORDImage d'Oracle 10g et Ordimage de Java.[20].

Une analyse des exigences envisagées permet de réaliser le découpage du projet et de générer les fonctions et sous-fonctions principales suivantes :

Exigences	Sous fonctions principales
Extraction des images	▪ Paramétrage traitement ▪ Chargement de la vidéo ▪ Gestion erreurs ▪ Affichage déroulement
Traitement des images	▪ Connexion à la BD ▪ Création de la table ▪ Insertion des images ▪ Création des fichiers MPEG-7 ▪ Insertion des fichiers dans la BD ▪ Détection scènes ▪ Gestion erreurs ▪ Affichage déroulement
Visualisation	▪ Connexion à la BD ▪ Choix images à visualiser ▪ Affichage images ▪ Affichage fichiers MPEG-7 ▪ Gestion erreurs

Tableau4 – exigences projet

Quant aux scénarios d'utilisation, j'en ai identifié 3 principaux :

Scénario 1 : Obtention des images à partir d'un flux vidéo

Les opérations possibles sont :

- choisir les paramètres de roulage (durée du morceau vidéo analysée et cadence d'échantillonnage)
- chargement de la vidéo

Choisir les paramètres de roulage

Description sommaire	L'utilisateur utilise l'interface graphique pour saisir les valeurs désirées quant aux paramètres d'application extraction images. Cette étape pourra manquer si l'utilisateur utilise les valeurs par défaut.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur utilise les zones prévues pour la saisie des paramètres.
Description du processus	Après le lancement de l'application, l'utilisateur doit choisir la durée de la vidéo qu'il veut utiliser pour extraire les images (secondes) et la fréquence des extractions (images/seconde). Il peut accepter les valeurs par défaut offertes par l'application ou peut les changer, en respectant les indications affichées sur l'écran.
Règles de	L'utilisateur fait son choix et il actionne les boutons du menu pour lancer le

terminaison	chargement du fichier. L'utilisateur sort complètement du programme.
Exception	Aucune
Extension	Aucune
Complémentaires	Aucun

Chargement de la vidéo

Description sommaire	L'utilisateur sélectionne l'option "Fichier" du menu pour lancer le chargement de la vidéo. Il utilise ensuite un menu dialogue pour arriver à charger le fichier désiré.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur utilise l'option "Fichier" du menu.
Description du processus	En actionnant l'option "Fichier", l'utilisateur déclenche une procédure de vérification paramètres de roulage. Si le résultat est positif, un menu dialogue de chargement fichier apparaîtra. L'utilisateur fouille les répertoires jusqu'il trouvera le fichier voulu et il va le charger en appuyant sur "Ouvrir". Il a la liberté d'annuler l'opération en appuyant sur "Annuler". Une fois le chargement déclenche, le module logiciel passe à l'extraction des images tout en affichant dans une zone prévue à l'écran le déroulement du processus.
Règles de terminaison	L'utilisateur fait son choix et actionne le bouton "Ouvrir". L'utilisateur actionne le bouton "Annuler" L'utilisateur sort complètement du programme.
Exception	Le fichier d'entrée n'existe pas.
Extension	Aucune
Complémentaires	Aucun

Scénario 2 : Insertion des images dans la BD et détection des changements de scène.

Les opérations possibles sont :

- connexion à la BD

Connexion à la BD

Description sommaire	L'utilisateur introduit ses coordonnées (nom usager et mot de passe) et actionne le bouton « Connexion ». Une réussite de connexion à la BD
-----------------------------	---

	déclanche la procédure de traitement lot images.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur actionne le bouton "Connexion"
Description du processus	En actionnant le bouton « Connexion », l'utilisateur déclenche le lancement de la procédure de connexion sur Oracle. Si la connexion est réussie, on passe à la création de la table, à l'insertion des images dans la table et aux procédures d'analyse des images et de création des fichiers MPEG-7 associés. Après leur insertion dans la BD, on déclenche un module logiciel permettant d'identifier et noter les changements de scène.
Règles de terminaison	Connexion non réussie sur la BD (message d'erreur affiché) Connexion réussie (message de confirmation affiché) L'utilisateur sort complètement du programme.
Exception	La BD n'existe pas sur la machine distante.
Extension	Aucune
Complémentaires	Aucun

Scénario 3 : Visualisation (images et fichiers MPEG-7)

Les opérations possibles sont :

- Connexion à la BD
- Affichage images
- Affichage fichiers MPEG-7

Connexion à la BD

Description sommaire	L'utilisateur introduit ses coordonnées (nom usager et mot de passe) et actionne le bouton « Connexion ». Une réussite de connexion à la BD déclenche la procédure de création listes des images disponibles.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur actionne le bouton "Connexion"
Description du processus	En actionnant le bouton « Connexion », l'utilisateur déclenche le lancement de la procédure de connexion sur la BD Oracle. Si la connexion est réussie, on passe à la création de deux listes : l'une qui contienne toutes les images de la BD, l'autre spécifiant les images identifiées comme début de scène.
Règles de terminaison	Connexion non réussie sur la BD (message d'erreur affiché) Connexion réussie (message de confirmation affiché)

	L'utilisateur sort complètement du programme.
Exception	La BD n'existe pas sur la machine distante.
Extension	Aucune
Complémentaires	Aucun

Affichage images

Description sommaire	L'utilisateur fait son choix quant aux images à visualiser et actionne le bouton « Visualiser ». On cherche les images choisies dans la BD et on les affiche à l'écran.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur actionne le bouton "Visualiser"
Description du processus	Le bouton "Visualiser" va considérer les valeurs choisies pour les deux listes comme paramètres pour bâtir ses requêtes vers la BD. Les images déposées dans la BD seront repêchées et affichées dans deux zones établies sur l'interface graphique. On affiche également un bouton nommé « MPEG-7 » qui permet d'accéder au fichier MPEG-7.
Règles de terminaison	Affichage des images choisies L'utilisateur sort complètement du programme.
Exception	Aucune
Extension	Aucune
Complémentaires	Aucun

Affichage fichier MPEG-7

Description sommaire	L'utilisateur actionne le bouton «MPEG-7 ». On affiche dans une zone prévue le contenu du fichier XML reçu comme réponse à la suite de la requête vers la BD.
Acteurs primaires	L'utilisateur
Acteurs secondaires	aucun
Règles d'initiation	L'utilisateur actionne le bouton "MPEG-7"
Description du processus	En cliquant le bouton MPEG-7, on lance une requête paramétrée (avec la valeur courante de la liste de droite) vers la BD. Le résultat retourné prend la forme d'un fichier XML affiché dans une zone spécialement prévue sur l'interface graphique.

Règles de terminaison	Affichage du fichier XML L'utilisateur sort complètement du programme.
Exception	Aucune
Extension	Aucune
Complémentaires	Aucun

J'ai considéré plusieurs aspects pour choisir la méthodologie de développement. J'ai cherché d'utiliser une méthodologie pouvant me répondre aux exigences établies dès le début :

- optimisation de la gestion du temps
- facilité et simplicité de réalisation et de suivi
- concordance avec l'ampleur du projet
- assurer la qualité requise

J'ai opté pour le modèle en V, dérivé du modèle classique en cascade (*waterfall*). C'est un modèle approprié à l'envergure de mon projet et offre certains avantages quant aux interliaisons logiques entre les phases du cycle de développement. Il apporte des importantes corrections au modèle en cascade, car, en pratique, le développement d'un projet logiciel n'est pas linéaire, mais itératif, avec boucles de rétroaction. Il arrive souvent qu'au cours d'une phase on découvre des erreurs commises dans des phases antérieures (ou l'absence des éléments indispensables au fonctionnement).

L'image suivante donne l'aperçu des principales étapes à suivre dans l'acceptation du modèle en V du cycle de développement :

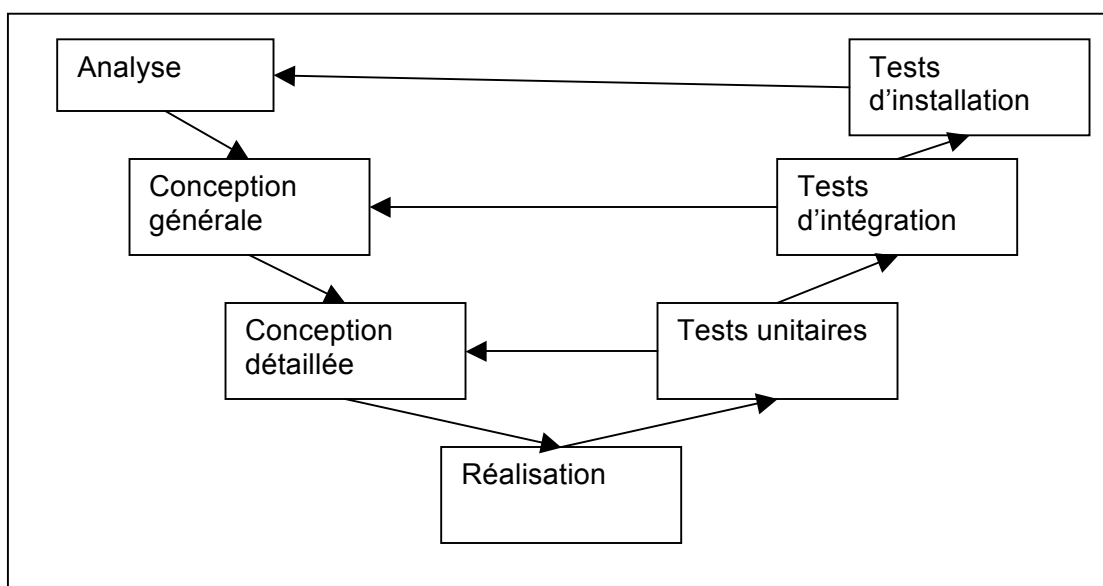


Figure 22 : Modèle en V du cycle de développement [22]

Le tableau suivant synthétise les étapes à suivre, les tâches à accomplir et les livrables pour le processus de développement du logiciel :

Étape	Tâches	Livrables
Analyse	- Analyse des besoins d'utilisateur - Définition des fonctionnalités du système	- Spécifications du logiciel (cahier des charges du logiciel)
Conception générale	- Etablir des variantes de solutions - Description et détails sur la solution retenue - préparer la phase intégration	- Document de conception générale (schéma d'architecture, organisation en entités, interfaces, interactions) - plan de test d'intégration - Diagrammes des cas d'utilisation - Scénarios possibles
Conception détaillée	- Identification des composants logiciels élémentaires - préparer la phase des tests unitaires	- Document de conception détaillée (description détaillée des composants élémentaires) - plan de tests unitaires - Diagrammes des cas d'utilisation - Diagrammes des séquences
Réalisation	- création des modules logiciels à l'aide du langage de programmation	- modules logiciels - description environnement de production - vues et interfaces graphiques pour les principaux scénarios
Tests	- tests unitaires : évaluation des composantes - tests d'intégration : évaluation des composantes assemblés	- document tests unitaires, et d'intégration
Installation	- Installation ensemble modules - test d'exploitation et d'usage	- document test d'exploitation - document configuration de l'application - manuel d'utilisateur

Tableau5 – plan de travail (cédule)

3.3 Conception

La conception générale du système a été pensée de façon modulaire, en respectant les exigences exprimées au début. L'approche modulaire permettra d'Associer à chaque exigence un module ou sous-module logiciel. Cela apporte flexibilité, identification bien ciblée des problèmes, augmentation de la qualité (identification rapide et facile des « bugs »). Un autre avantage offert consiste dans la possibilité de faire une agrégation des modules réalisés selon les besoins courants. (Trajet par étapes, répétition certains phases, etc.). Chacune des exigences identifiées durant l'étape d'analyse sera prise en charge par un des modules réalisé. Un aperçu de la conception générale met en évidence l'existence de trois modules principaux :

- MEI (module extraction images)
- MTI (module traitement images)
- MV (module visualisation)

La figure suivante donne une vue de principe sur le couvrement proposé par chacun des trois modules.

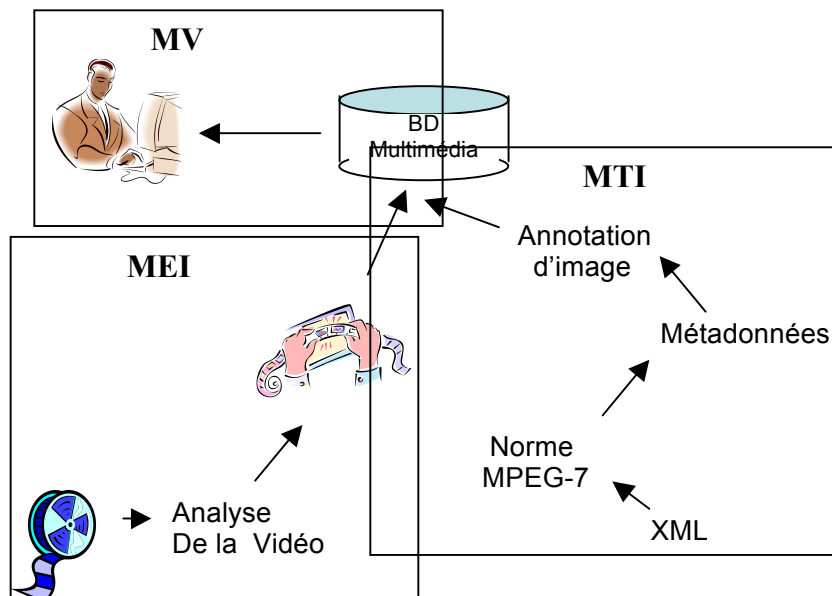


Figure 23: Conception générale système (Groupement fonctionnalités)

Plus précisément, les activités développées sont les suivantes :

- MEI – charge la vidéo, et fait l'extraction des cadres (images) du flux vidéo sous le format JPG. Les images seront déposés dans un répertoire sur le disque dur, afin d'être utilisés par le module MTI.

- MTI – réalise la connexion à la BD, repêche les images déposées par MEI dans le répertoire, et les dépose dans une table créée auparavant. Les images seront déposées dans l'ordre de leur création, dans l'ordre de succession naturelle dans le flux vidéo. Les images déposées dans la BD seront analysées deux à deux (une image et la suivante) pour détecter un changement de scène. L'analyse sera basée sur la mesure de la similarité des signatures des images. Les enregistrements correspondants aux images identifiant un changement de scène seront marqués (champ indicateur) pour faciliter la tâche du module MV. Pour ces images, un document XML selon la norme MPEG-7 sera généré.
- MV – permet la connexion paramétrée à la BD et la visualisation des images déposées. (à la fois toutes les images de la BD et celles identifiées comme représentatives pour les scènes). On a aussi l'accès aux fichiers XML/MPEG-7 associés.

Chacun des modules identifiés prend en charge plusieurs tâches, assurant plusieurs fonctionnalités à travers certains sous-modules. Les sous-modules chargés d'assurer les fonctionnalités nécessaires à MEI sont identifiés dans l'image suivante qui donne la conception détaillée du module :

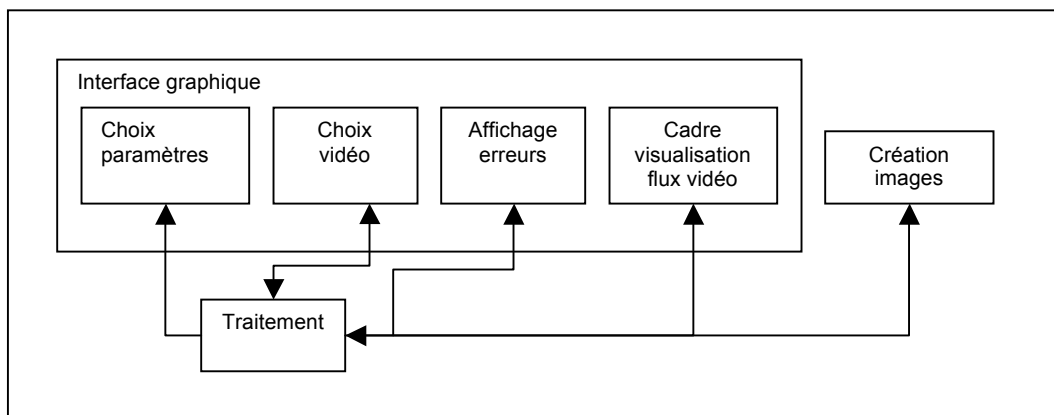


Figure 24 : Conception détaillée MEI (module extraction images)

L'allocation des fonctionnalités est clairement identifiée et allouée :

- choix paramètres roulage algorithme extraction images
- choix et chargement fichier vidéo
- extraction cadres à partir du flux vidéo

- création images
- traitement information et gestion des entrées, sorties et erreurs.

Une approche UML du fonctionnement du module MEI est présenté dans l'annexe 2. On y trouve :

- diagramme des cas d'utilisation
 - niveau modèle
 - niveau instances
- diagramme séquences (scénario)
- diagramme des classes Java
 - classes, attributs, méthodes

Le deuxième module, celui de traitement images est à son tour conçu de façon modulaire, allouant de manière arborescente ses fonctionnalités. La figure suivante identifie la conception détaillée du MTI :

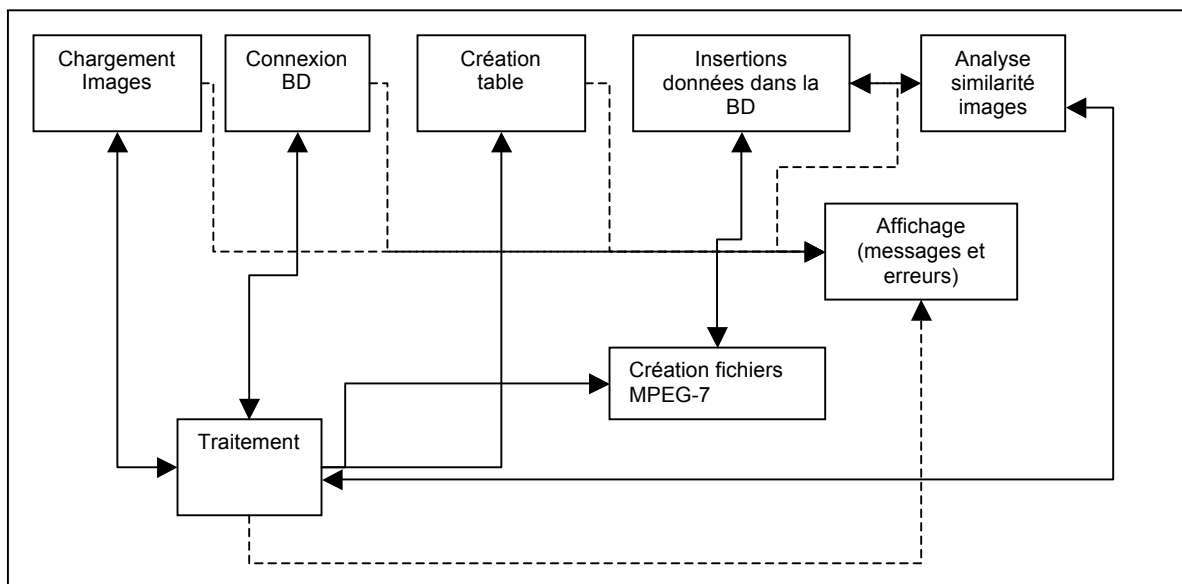


Figure 25 : Conception détaillée MTI (module traitement images)

Les fonctionnalités du module MTI sont réalisées à travers ses composants :

- Création des listes à partir des images
- Connexion à la base de données
- Création de la BD
- Insertion des données dans la BD

- Création des fichiers MPEG-7 pour les images
- Détection des changements de scène
- traitement information et gestion des entrées, sorties et erreurs.

Une approche UML du fonctionnement du module MTI est présenté dans l'annexe 3. On y trouve :

- diagramme des cas d'utilisation
 - niveau modèle
 - niveau instances
- diagramme séquences (scénario)
- diagramme des classes Java
 - classes, attributs, méthodes

Le dernier module, celui de visualisation propose le schéma fonctionnel suivant :

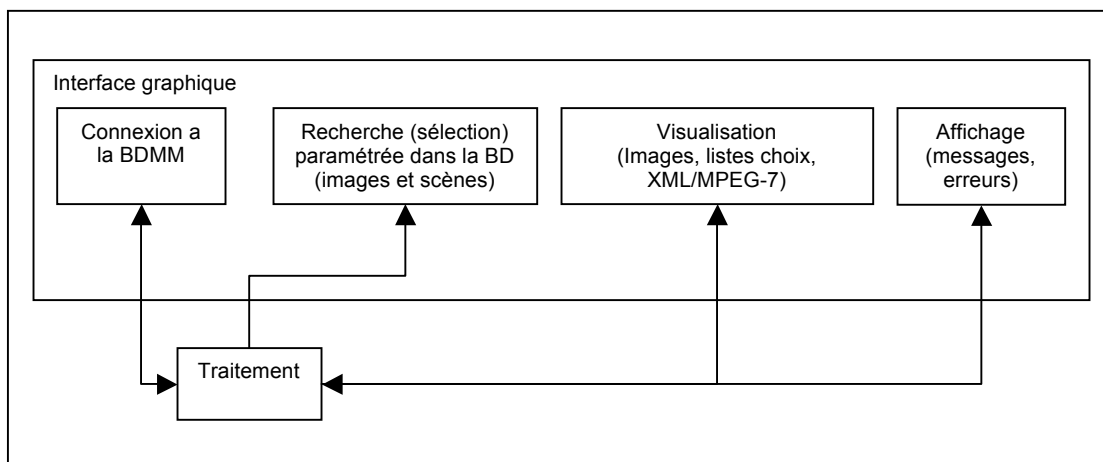


Figure 26 : Conception détaillée MV (module visualisation)

L'allocation des fonctionnalités du module MV est la suivante :

- connexion à la BD
- repérage des images dans la BD
- Affichage des images et des fichiers MPEG-7
- Traitement de l'information et gestion des entrées, des sorties et des erreurs.

Une approche UML du fonctionnement du module MV est présenté dans l'annexe 4. On y trouve :

- diagramme des cas d'utilisation
 - niveau modèle

- niveau instances
- diagramme séquences (scénario)
- diagramme des classes Java
 - classes, attributs, méthodes

3.4 Programmation et tests

Dans cette étape, j'ai passé à la réalisation pratique des modules logiciels capables de réaliser les fonctionnalités requises. À partir des diagrammes des séquences, des cas d'utilisation et des scénarios identifiés à l'étape de conception j'ai commencé réaliser les modules tout en abordant une approche procédurale, modulaire, de découpage pour le développement.

C'est une approche qui permettra de corriger l'architecture suite à la découverte des problèmes. Des points de vérification et de validation ont été prévus pour chaque module développé, permettant de découvrir le plus tôt les problèmes et de les résoudre rapidement.

Les tests unitaires faites, on eu comme principal objectif de vérifier individuellement que chaque unité fonctionne correctement, qu'elle répond de façon quantitative et qualitative aux attentes. J'ai appliqué la technique de tests manuelles, effectué à l'aide des données préétablies.

L'approche modulaire (création indépendante des modules fonctionnels MEI, MTI, MV) adoptée a été structurée en plusieurs phases, dont les plus importantes sont :

- réalisation des sous-modules permettant de communiquer avec « l'extérieur » du système : répertoires des fichiers, base de données
- réalisation des sous-modules assurant les fonctionnalités requises
- réalisation de l'interface graphique :
 - menus (navigation, choix, etc.),
 - zones de saisie données
 - boutons des actions
 - boîte de dialogue

- zone affichage erreurs
 - zone affichage graphique
- essayer de garder une concordance entre les scénarios identifiés et les « vues » graphiques, d'assurer la convivialité de l'interface.
- Ajout au fur et à mesure des fonctionnalités nouvelles (itérations création - tests)
- Gestion dynamique de l'écran : zones actives/inactives, boutons visibles/invisibles, événements déclenchés automatiquement, etc.
- Prévoir les interfaces de liaison avec les autres modules et sous-modules de l'application, envisager le comportement de l'ensemble final.
- Un diagramme qui décrit de façon générale la séquence de travail pour la réalisation des modules logiciels est montré dans la figure suivante.

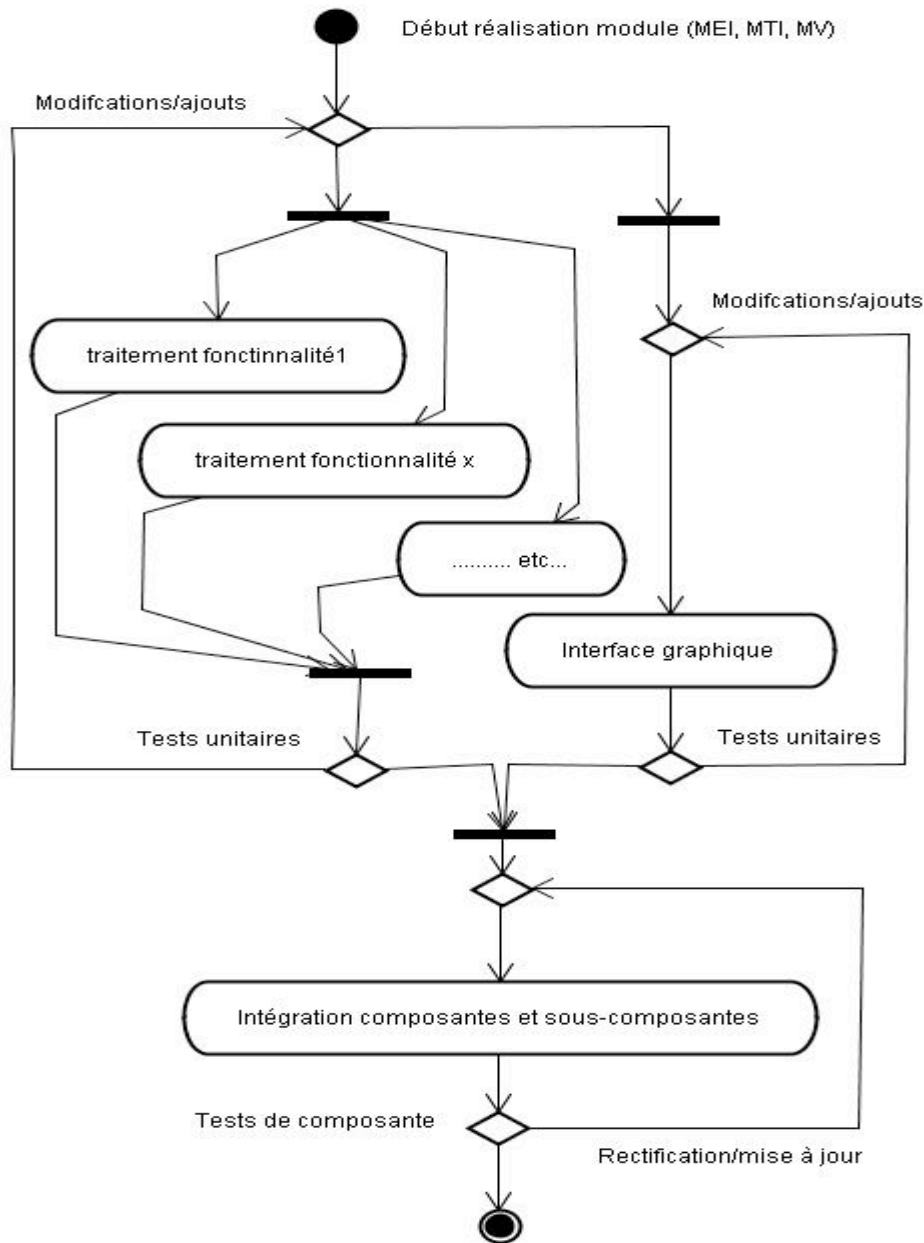


Figure 27 : Séquence de travail pour la réalisation des modules logiciels

L'annexe 5 contient plusieurs documents liés à l'activité déposée pendant l'étape de programmation :

- Exigences de l'environnement logiciel pour chacun des modules réalisés.

Dans l'annexe 6 on trouve les diagrammes d'activités liées à l'étape de programmation, plus précisément :

- Séquence de travail pour la réalisation du module MEI

- Séquence de travail pour la réalisation du module MTI
- Séquence de travail pour la réalisation du module MV

L'annexe 7 donne un aperçu des vues pour les scénarios des cas d'utilisation identifiés et offre des informations sur les principales interfaces, classes et méthodes utilisées pour réaliser de code

3.5 Chargement des données et essais

Fonctionnement MEI (module extraction images)

Tout d'abord, à travers l'interface graphique, l'utilisateur doit paramétrer sa démarche, en choisissant des valeurs désirées pour la durée du morceau vidéo analysé (valeur en seconde) et pour la cadence d'échantillonnage (nombre images/seconde). Des valeurs par défaut sont offertes, et une vérification est faite avant de pouvoir passer à l'étape suivante, le chargement de la vidéo. Seulement si les deux valeurs sont acceptables de point de vue logique, le chargement est permis, sinon, un message d'erreur avise l'utilisateur sur le problème à résoudre. Les images suivantes donnent un aperçu des étapes à suivre :

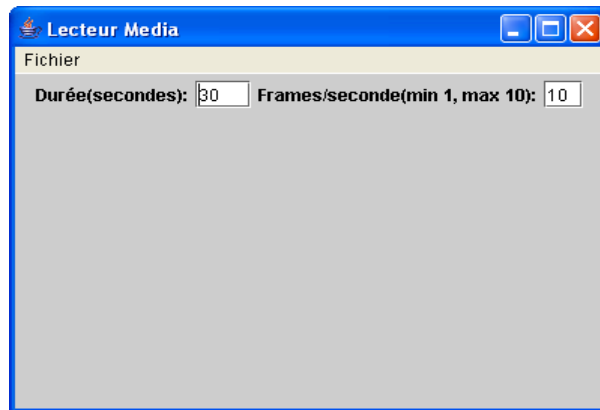


Figure 28 : Ecran initial (valeurs par défaut)

Une tentative de chargement vidéo, vérifie si il y a des erreurs de saisie (lettres au lieu de chiffres, valeur négative pour la durée, valeurs hors gamme pour la cadence d'échantillonnage, etc.). Si c'est le cas, le chargement n'est pas permis, et on affiche un message d'erreur :

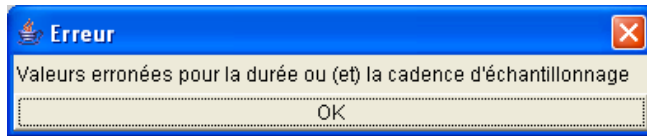


Figure 29 : Message erreur paramétrage MEI

Après la résolution du problème, le chargement est permis, à travers un menu déroulant :



Figure 30 : dialogue chargement fichier vidéo

On cliquant le bouton choix fichier on peut naviguer jusqu'on trouve le fichier voulu. On confirme le chargement (bouton Ouvrir), ou on annule (bouton annuler). Si le bouton ouvrir a été actionné, l'application tente de charger le fichier et de démarrer le lecteur multimédia. Si le format de fichier choisi n'est pas accepté, un message d'erreur avertisse du problème rencontré :

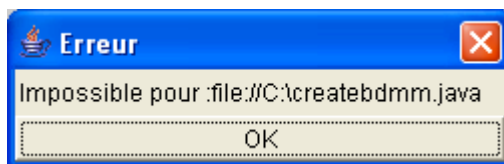


Figure 31 : erreur démarrage lecteur vidéo

Si il s'agit d'un fichier vidéo, le lecteur démarre et l'utilisateur a la possibilité de voir simultanément le déroulement de la vidéo et l'avancement d'extraction images, dans une zone d'affichage déroulante.

La composante vidéo du fichier chargé est démarrée dans le lecteur JMF (utilisation de la class Manager pour réaliser un Player).

On analyse les événements associés pour le contrôleur et si l'état du *player* le permet (*Prefetch* réalisé) on procède à :

- démarrer le lecteur
- capture (paramétrée) des images du flux vidéo
- chaque image portera un texte indiquant la date de la création
- affichage déroulement extraction images
- sauvegarde des images dans un répertoire(c :/testimag) créé antérieurement sur le disque dur

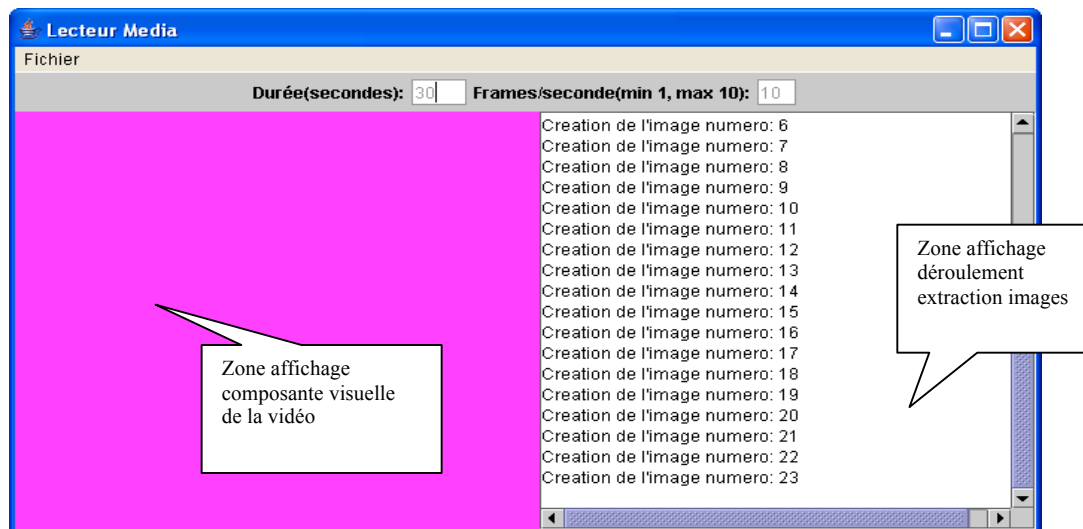


Figure 32 : déroulement extraction images

L'image suivante donne un aperçu du répertoire *c:\testimag* après l'exécution du module logiciel MEI (fichier *LecteurMedia.java*, dans le projet *coupevideo.jpr*) :

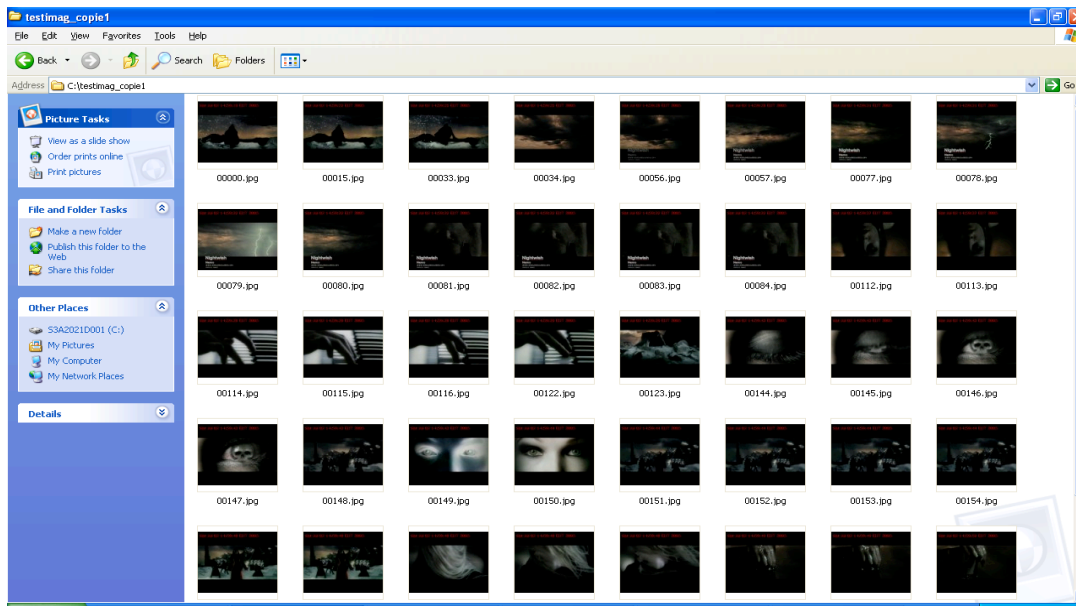


Figure 33 : Images extraites de la vidéo

Fonctionnement MTI (module traitement images)

Le fonctionnement suppose l'existence des fichiers images (jpg) dans le répertoire c:\testimag. C'est le module MEI qui se charge d'assurer cette condition. Tout d'abord, l'utilisateur doit s'identifier pour pouvoir se connecter à la base de données Oracle. La chaîne des caractères composant l'identification de l'utilisateur est réalisée à partir des données saisies en utilisant des champs prévus dans l'interface graphique. Celle-ci est prévue avec une zone d'affichage pour les erreurs de connexion à la base de données et une autre pour visualiser le déroulement des événements. (Chargement, création table, insertion des données, détection images début scènes.)

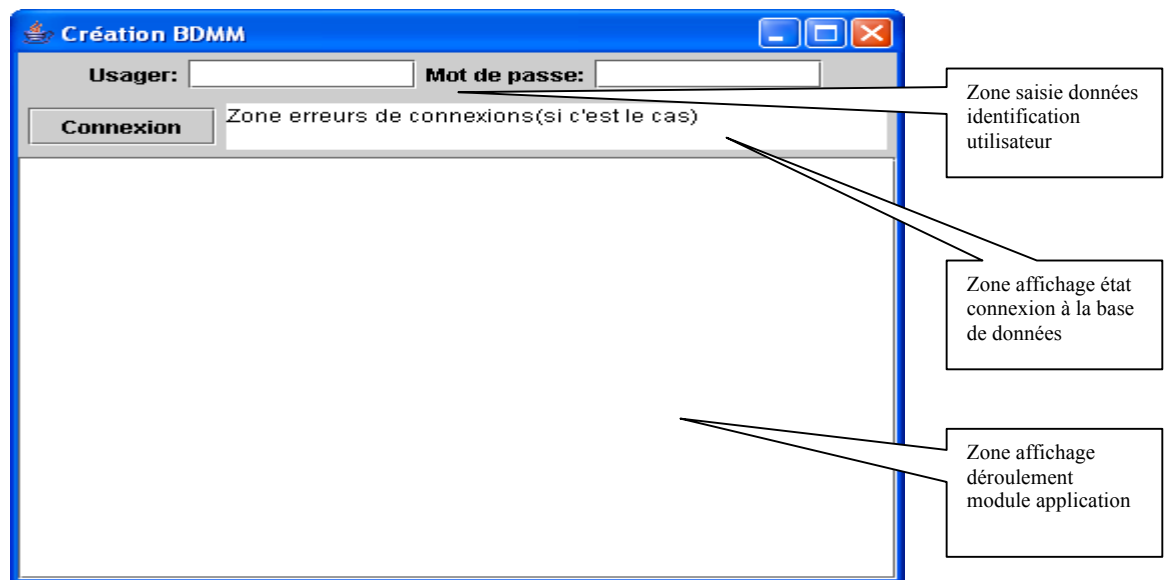


Figure 34 : Écran démarrage MTI

Si la connexion échoue, un message d'erreur est affiché dans la zone dédiée, sinon, le message Connexion réussie s'affiche, le bouton Connexion passe en état inactif et la succession d'événements pourra être observé dans la zone d'affichage déroulante :



Figure 35 : Déroulement module MTI

Le déroulement des opérations, suppose l'exécution des étapes suivantes :

- chargement des images sous forme de liste java (filtre pour l'extension jpg, tri selon le nom. Le module MEI donne des noms aux fichiers d'une façon respectant la succession temporelle d'extraction).
- Connexion à la BD (enregistrement pilote Oracle, connexion paramétrée respectant la forme : ***pilote :drive@machine :port :bd, usager, mot_de_passe.***
- Création de la BD (utilisation des champs spécifiques à l'extension Intermedia d'Oracle :ORDImage et ORDImageSignature)
- Insertion des images dans la BD : création des objets Ordimage ayant comme flux d'entrée (*getBinaryOutputStream()*) le flux de sortie provenant des images (*FileInputStream*). L'accès aux images est faite à travers les listes Java construites en analysant le contenu du répertoire c ://testimag. Les listes sont déjà triées selon le nom des fichiers, qui ont été choisis d'une façon qui respecte le déroulement temporel de la création des images.
- Création et insertion des fichiers MPEG-7 pour chaque image déposée dans la BD. Pour arriver à obtenir les informations nécessaires on a utilisé une classe Java capable de capter l'information contenue dans l'image (la classe PixelGrabber qui fait partie du paquetage java.awt.image). L'essentiel apport de l'utilisation de cette classe est qu'elle permet de faire la correspondance entre une image et un tableau de pixels. Ce tableau est un tableau d'entiers contenant les valeurs RGB des pixels de l'image donnée. Cette classe offre la méthode grabPixels() qui permet effectivement le chargement des pixels de l'image vers le tableau d'entiers. Une fois obtenu le tableau des pixels, on peut extraire la couleur associée à chaque pixel si on connaît le modèle de couleur associé à image analysée, celui qui détermine façon dont les composantes de couleur sont stockées. C'est une autre méthode de la classe PixelGrabber , getColorModel(), celle qui donne un objet de la classe ColorModel, associé en Java pour un modèle de couleur. Ayant ces informations, on peut obtenir les couleurs de chaque pixel (on applique des méthodes getRed, getBlue, getGreen de la classe ColorModel), et on peut approximer une couleur dominante pour l'image, en normalisant les valeurs trouvés. (valeurs moyennes calculés en tenant compte de la totalité des pixels de l'image). On obtient ainsi la valeur de la couleur

dominante (*DominantColor*) de l'image analysée, dans l'acception MPEG-7.
L'image suivante exemplifie le cas du fichier associé à image ph00000.jpg :

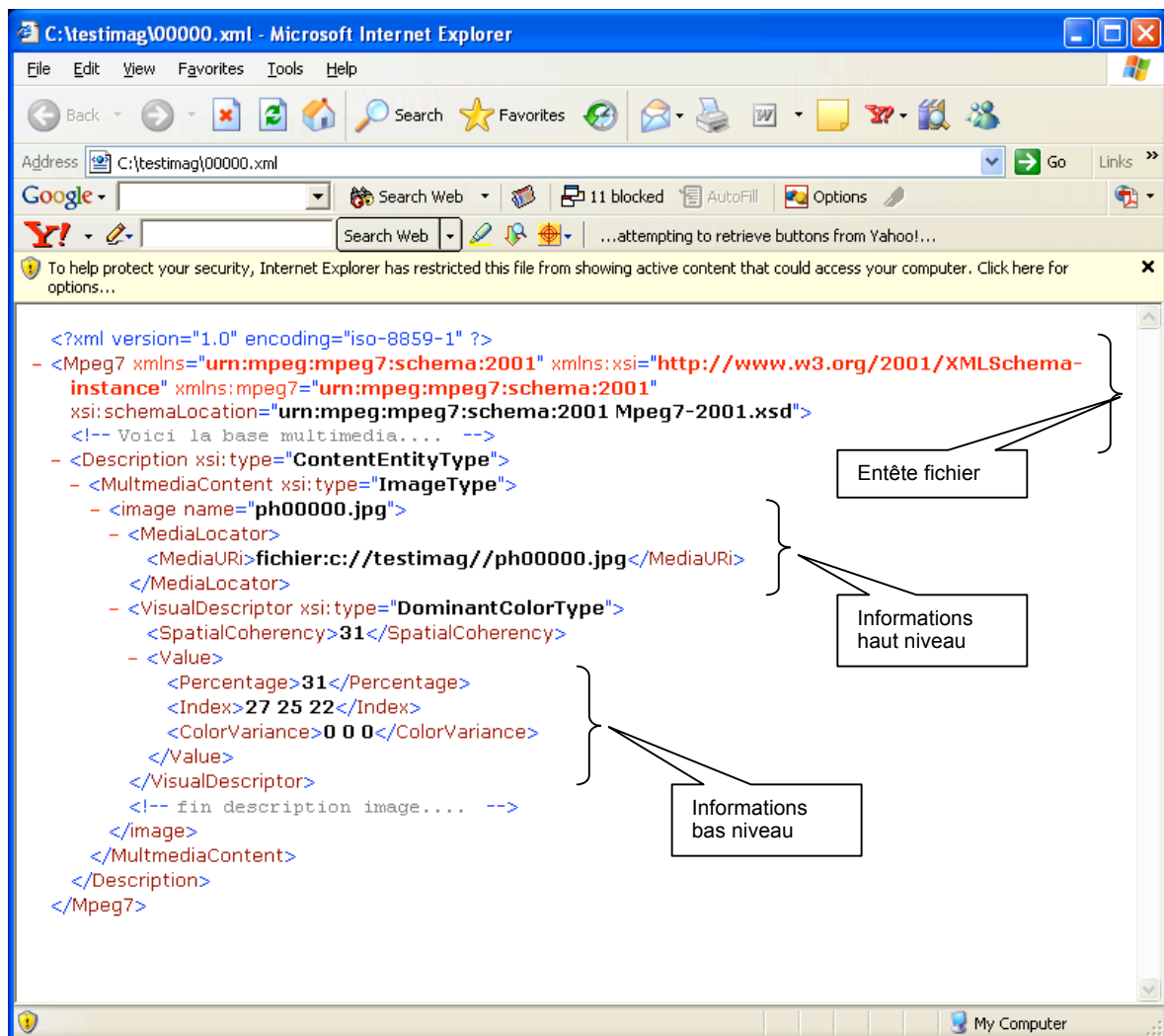


Figure 36 : Exemple fichier MPEG-7 (*DominantColor*)

Entête

- Une fois les images transférés dans la BD on procède à la recherche des images identifiant les changements de scène. Pour y arriver on analyse la similarité des signatures des images successives. Si deux images successives sont considérés différentes (selon certains critères concernant les attributs de bas niveaux) on considère qu'on a un début de scène et on donne la valeur "s" à un champ indicateur (ind) de l'enregistrement de la BD. La clé de l'analyse est l'utilisation de la méthode `IsSimilar` de la classe `ORDImageSignature` qui

retourne un entier (1 pour images similaires, 0 pour le contraire). La syntaxe générale est la suivante :

- IsSimilar(sign1, sign2, chaîne attributs, seuil similarité)
 - Sign1, 2 les deux signatures à comparer
 - Chaîne attributs est une chaîne alphanumérique spécifiant la combinaison désirée pour les attributs de bas niveaux. A titre d'exemple, on peut avoir : *"color=0.3 texture=0.5 shape=0.1 location=0.1"*
 - Seuil similarité (*similarity_threshold*) – valeur (maximale 100) à partir de laquelle on considère que les deux signatures ne sont plus identiques.

Le résultat des comparaisons est affiché à l'écran, pour chaque paire des images analysée. Au final, on affiche le nombre des insertions faites avec succès dans la BD.

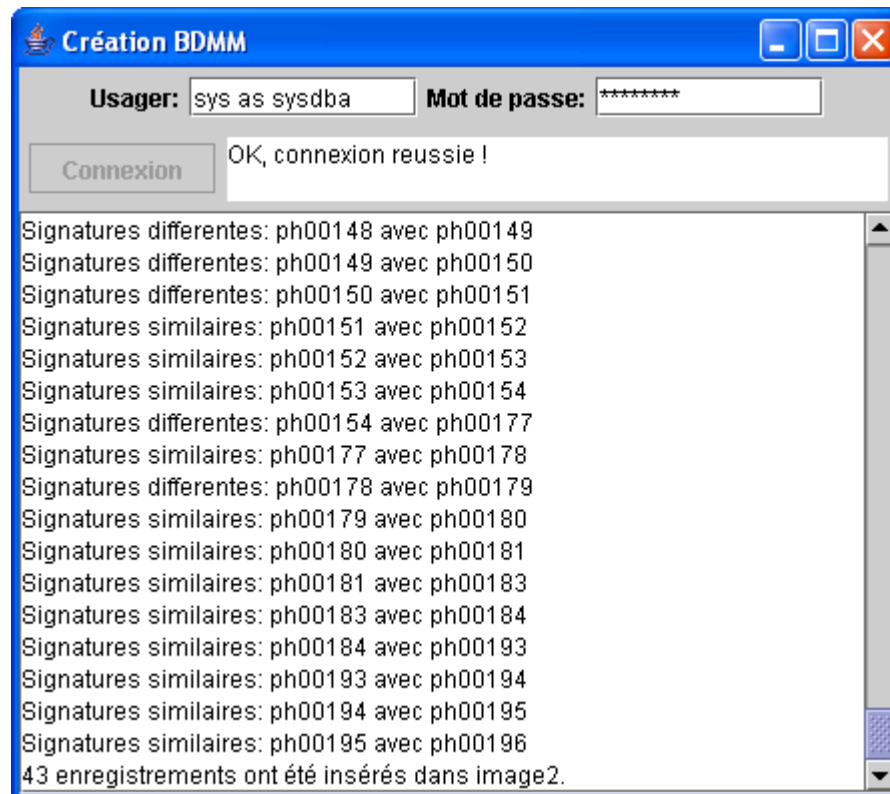


Figure 37 : Messages analyse similarité et nombre finale insertions dans la BD

Fonctionnement MV (module visualisation)

Ce module permet l'affichage des images déposées dans la base de données à travers une interface graphique. Deux listes sont disponibles pour le choix de la visualisation. L'une permet de choisir parmi toutes les images de la base de données, et l'autre donne accès seulement pour les images identifiées comme début de scène. L'affichage simultanément des deux images pourra être accompagné par l'affichage du fichier MPEG-7 associé à l'image de début de scène.

Tout d'abord l'utilisateur doit s'identifier pour accéder à la BD. Les données saisies à l'aide des zones "Usager" et "Mot de passe" seront utilisées pour bâtir la chaîne alphanumérique utilisée pour la connexion à la BD Oracle. En actionnant le bouton "connexion" on déclenche plusieurs événements :

- tentative de connexion sur la BD (basée sur les données saisies)
- affichage de l'état de la connexion (message d'erreur si c'est le cas) à la suite de cette action.
- Si la connexion a été faite, on met en état inactif le bouton connexion et on affiche les listes des options pour les images disponibles dans la BD
 - o liste1 – contenant toutes les images de la BD
 - o liste 2 – contenant seulement les images des tuples ayant le champ ind="s" (images identifiant les scènes)

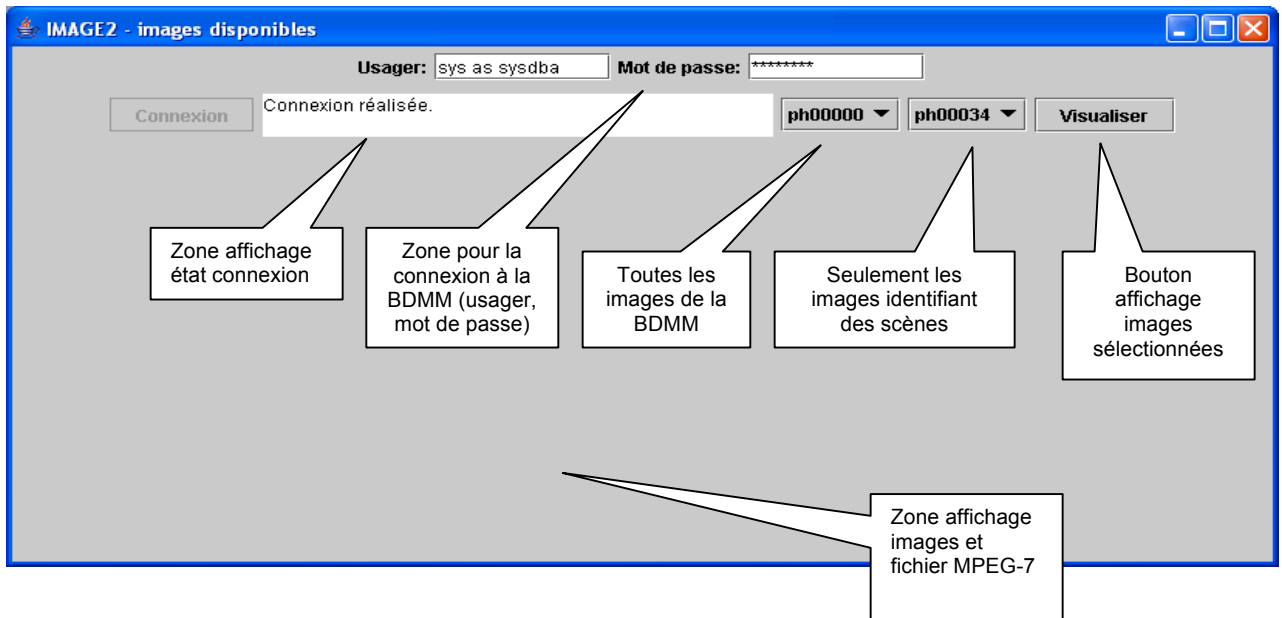


Figure 38 : Interface graphique MV – connexion réussie à la BD

Une fois les deux listes disponibles, l'utilisateur peut les parcourir en toute liberté et choisir lesquelles des images qu'il veut visualiser.

Cela permet de faire un suivi visuel du moment détecté comme changement de scène, en choisissant par exemple de visualiser une image début de scène et l'une des images voisines. C'est une fonctionnalité permettant de renforcer (et parfois de contester) les résultats obtenus à la suite d'analyse similarité.

Le bouton "Visualiser" va considérer les valeurs choisies pour les deux listes comme paramètres pour bâtir ses requêtes vers la BD. Les images déposées dans la BD seront repêchées et affichées dans deux zones établies sur l'interface graphique. Simultanément, un bouton permettant d'accéder au fichier MPEG-7 de l'image identifiant la scène devient disponible. Un exemple possible est montré dans l'image suivante :

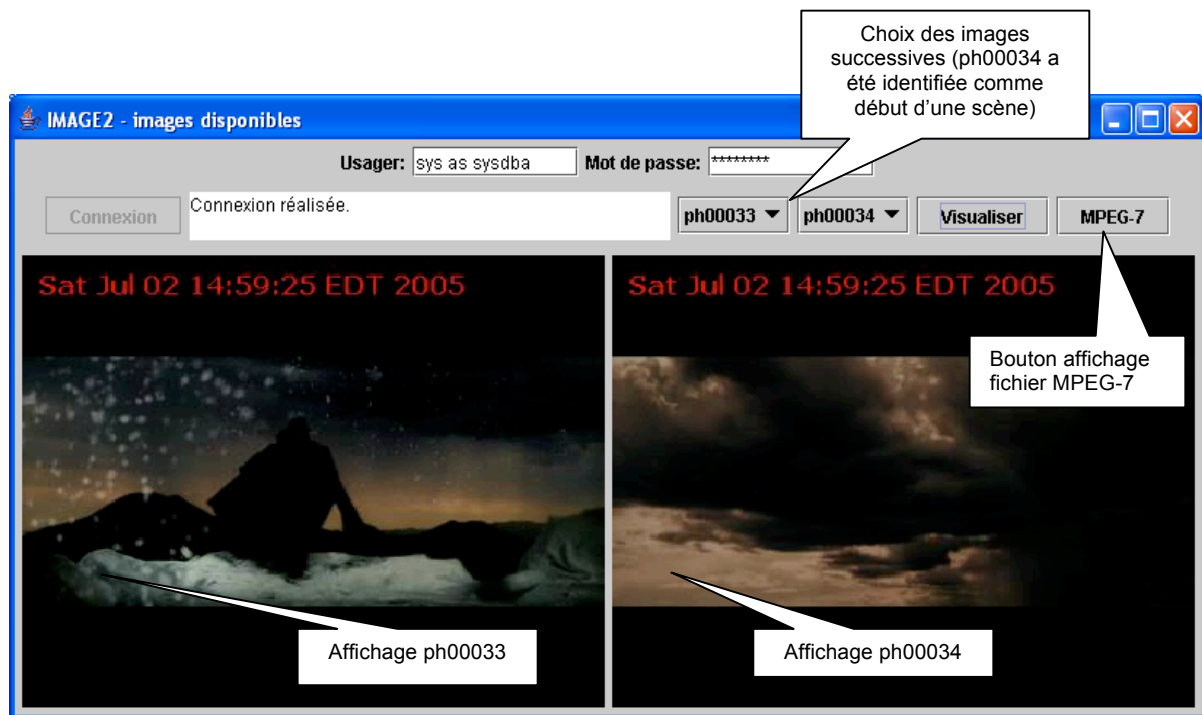


Figure 39 : Interface graphique MV – visualisation comparative images

En cliquant le bouton MPEG-7, on affiche dans une zone spécialement prévue le fichier MPEG-7 associé à l'image choisie pour la liste trouvée à droite sur l'interface graphique, celle des images qui identifient les scènes.

L'événement click souris est associé au lancement d'une requête paramétrée (avec la valeur courante de la liste de droite) vers la base de données. A la suite de

cette requête, le fichier MPEG-7 déposé dans la BD est téléchargé et affiché dans une zone de visualisation de l'interface graphique. L'image suivante exemplifie le cas de l'image ph00034 :

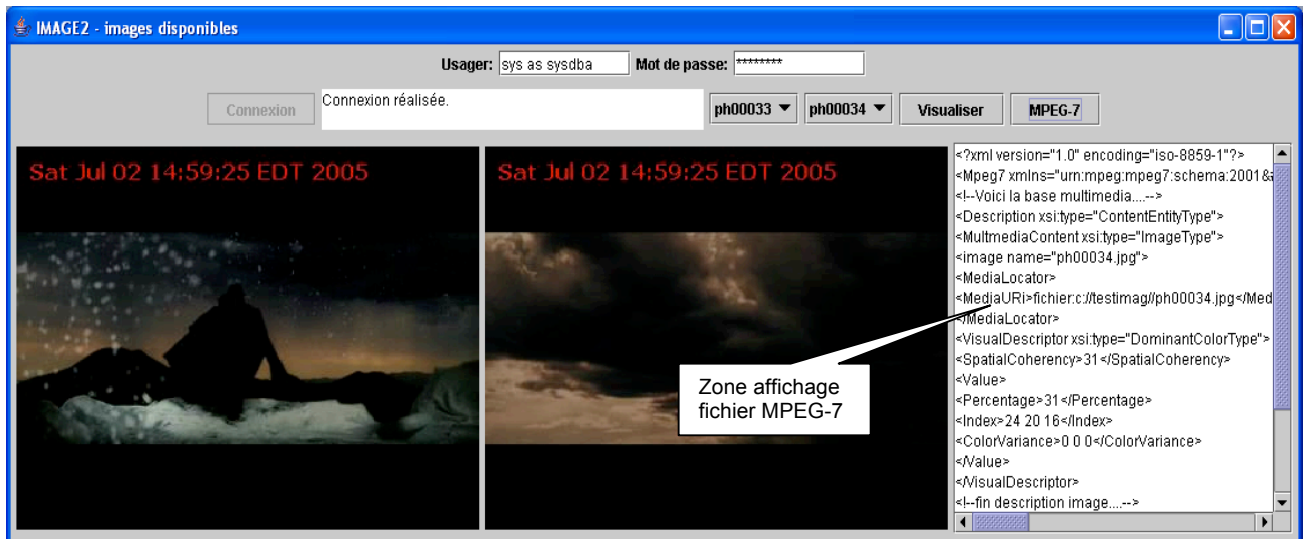


Figure 40 : Interface graphique MV – visualisation fichier MPEG-7

3.6 Sommaire

Dans ce chapitre j'ai présenté les activités reliées à la conception et la réalisation des modules logiciels de l'application. On a présenté au début la préparation nécessaire pour bien encadrer le projet (documentation, environnement, choix méthodologie, établissement cas d'utilisation et scénarios, rédaction ébauche plan de travail) Ensuite, on a présenté les démarches faites pour choisir l'Architecture (générale et détaillée) des modules fonctionnels. On a continué en spécifiant et expliquant l'allocation des chacune des fonctionnalités à un module ou sous-module et on a présenté la séquence de travail et le fonctionnement des modules réalisés, pendant l'étape de chargement données et essais.

Une fois cette étape atteinte, on est prêt d'aborder l'étude de l'impact des changements contextuels sur le déroulement et les performances du système. On vise à modifier les conditions d'entrée pour l'algorithme d'analyse similarité images, en modifiant les poids des attributs de bas niveau et le seuil de similarité.

L'analyse des résultats obtenus donneront des indices sur les limites du système réalisé, et se constituera dans un bon jalon de départ pour l'identification et la proposition des développements possibles.

4. CHAPITRE 3 – Interprétation

4.1 Introduction

Ce chapitre présente tout d'abord le contexte d'interprétation des résultats obtenus et l'approche utilisée.

Ensuite, on passe à l'analyse proprement dite, axée sur l'interprétation des résultats et la comparaison avec les attentes.

Les conclusions tirées permettront d'expliquer les différences face aux attentes, et de mettre en évidence les limites du système développé.

Finalement, on propose quelques améliorations et on trace des lignes directrices pour le développement futur.

4.2 Contexte d'interprétation

La gestion et la manipulation des données audiovisuels posent plusieurs défis, à la fois techniques et d'interprétation. Représenter ou analyser une vidéo exige une approche multi niveau [26], ou multidimensionnelle, nécessitant de considérer :

- la dimension spatiale – volumes, surfaces, lignes, positionnement relatif, etc.
- la dimension temporelle – ordre des images, durée, synchronisation, etc.
- la dimension hiérarchique – structure arborescente : vidéo → clips → scènes → prises de vue.
- la dimension du contenu – objets, relations entre objets.
- la dimension technique – métadonnées acquises de façon automatique.

De plus, comme est souligné en [4] et [24], le niveau sémantique est d'importance capitale, car c'est celui qui donne la description de haut niveau du contenu. (Représentation des objets, concepts événements et actions impliqués dans la vidéo).

Étant tributaire au contexte d'utilisation, le multimédia peut donner lieu à plusieurs interprétations [26], parfois divergentes, de son contenu. C'est à cette lumière que la présence humaine est fortement nécessaire. En effet, une analyse et une indexation totalement automatique est impossible, car, étant purement descriptive (identification des discontinuités temporelles et métadonnées basées sur des descripteurs de bas niveaux) laisse non traité un l'aspect essentiel, celui lié à la sémantique du contenu.

C'est le facteur humain celui qui se charge finalement avec l'analyse sémantique du contenu, celui qui peut comparer les résultats obtenus avec les attentes.

L'approche que j'ai choisie pour traiter les résultats mettra donc en prime plan l'importance de la boucle de rétroaction assurée par l'intervention humaine. L'amélioration successive des résultats sera possible à la suite de l'évaluation cyclique, interactive de l'utilisateur, qui bâtira ses nouvelles requêtes en se basant sur l'histoire des actions passées.

Les descripteurs utilisés seront considérés pertinents si le résultat obtenu correspondra à une réalité visuelle, à l'analyse effectuée de façon manuelle par l'observateur humain. L'utilisateur interagira avec le système en assurant la rétroaction nécessaire à la modification des critères de recherche. Cela permettra des résultats plus personnalisés et plus proches des attentes.

Cette approche assurant le "feedback" nécessaire est traduit dans notre cas, par la modification des poids associés aux descripteurs de bas niveaux, et de la valeur numérique associé au seuil de similarité. (Valeur au-delà de laquelle le modèle analysé et la cible peuvent être considérés comme différents).

Enfin, cette démarche nous permettra d'associer les dimensions techniques et sémantiques, car, en modifiant des valeurs numériques (valeurs pour des poids des attributs), on tire des conclusions sur l'influence des uns ou des autres des attributs (couleur, texture, motion, forme) sur l'efficacité de la recherche.

4.3 Résultats obtenus

Le processus d'obtention des images identifiant des débuts des scènes s'est déroulé de façon itérative, tout en respectant les étapes suivantes :

- analyse de la vidéo par l'observateur humain (comprendre la sémantique de la vidéo)
- analyse des images obtenues à la suite de la coupure de la vidéo (analyse pertinence images retenues)
- identification « manuelle » des changements de scène (donc subjective, mais décrivant la solution attendue)
- analyse résultats obtenus par l'identification automatique (réponse application), étape couvrant deux aspects :
 - aspect quantitative : calcul précision et rappel

- aspect qualitative : observation directe (analyse sémantique) des réponses
- modification des poids associés aux descripteurs de bas niveaux et du seuil de similarité.
- Itérations lancement application → analyse résultats → modifications paramètres, afin de repêcher les meilleurs résultats possibles.

La démarche utilisée pour arriver aux résultats est basée sur celle présentés en [26], et respecte ces grandes idées :

- On compare les images en comparant leurs “étiquettes”, associées en préalable. (dans notre cas, les étiquettes prenant la forme des signatures des images, dans l’acception Oracle 10g)
- On utilise des opérateurs numériques pour analyser la similarité des images (on construit une chaîne des attributs et poids associées que l’on envoie comme paramètre pour les requêtes de similarité)
- On calcule la distance entre les deux images (modèle et cible) à partir de leurs traits (caractéristiques).

La fonction utilisée pour tester la similarité des images est isSimilar d’Intermedia (Oracle 10g) dont la formule générale est la suivante :

- isSimilar (sign1, sign2, chaîne attributs, seuil similarité)
 - Sign1, 2 les deux signatures à comparer
 - Chaîne attributs est une chaîne alphanumérique spécifiant la combinaison désirée pour les attributs de bas niveaux. A titre d’exemple, on peut avoir : *"color=0.3 texture=0.5 shape=0.1 location=0.1"*
 - Seuil similarité (*similarity_threshold*) – valeur (maximale 100) à partir de laquelle on considère que les deux signatures ne sont plus identiques.

Durant les expériences pratiques, on a modifié les valeurs numériques représentant les poids associées aux attributs de bas niveaux (la couleur, la texture, la forme et la motion). On a enregistré les résultats obtenus et, en les confrontant aux attentes, on a pris les corrections nécessaires. L’évaluation des résultats a été faite en considérant deux aspects :

- Quantitative : calcul rappel et précision (aussi bruit et silence)
- Qualitative : inspection et validation visuelle des résultats.

Basé sur les définitions données en [26] et [27], les formules appliquées ont été :

- Précision - nombre des réponses pertinentes ramenées divisé par le nombre total des réponses
 - $P = RP / R$
- Rappel – nombre des réponses pertinentes ramenées divisé par le nombre des données pertinentes
 - $R = RP / DP$
- Bruit – information non pertinente ramenée
 - $B = (R-RP) / R$
- Silence – information pertinente non ramenée
 - $S = (DP-RP) / DP$

Où :

- RP = nombre réponses pertinentes
- R = nombre total réponses
- DP = nombre données considérées comme pertinentes

Une image suggestive pour l'interprétation des résultats est la suivante :

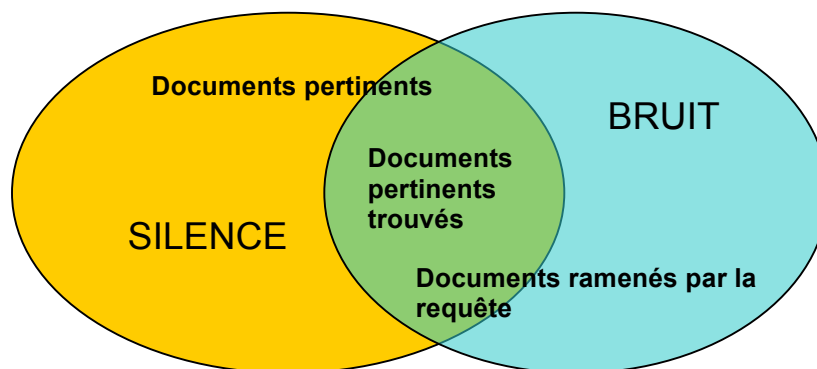


Figure 41: Bruit et silence

Source image : <http://www.webmaster-hub.com/publication/IMG/gif/bruit-silence.gif>

La boucle de réaction assurée par l'observateur humain, intervenant sur les entrées du processus (modifications poids et seuil similarité), a permis d'améliorer progressivement la pertinence des résultats obtenus.

On présente quelques exemples des modifications faites sur la chaîne de caractère utilisée comme paramètre pour la fonction isSimilar :

- 1 : (img_sig1, img_sig2, "color=0.3 texture=0.7 shape=0.0 location=0,0",10)
- 2 : (img_sig1, img_sig2, "color=0.1 texture=0.9 shape=0.0 location=0,0",10)
- 3 : (img_sig1, img_sig2, "color=0.1 texture=0.0 shape=0.9 location=0,0",10)
- 4 : (img_sig1, img_sig2, "color=0.1 texture=0.0 shape=0.0 location=0,9",10)
- 5 : (img_sig1, img_sig2, "color=0.5 texture=0.5 shape=0.0 location=0,0",10)
- 6 : (img_sig1, img_sig2, "color=0.7 texture=0.3 shape=0.0 location=0,0",10)
- 7 : (img_sig1, img_sig2, "color=0.4 texture=0.6 shape=0.0 location=0,0",10)
- 8 : (img_sig1, img_sig2, "color=0.35 texture=0.65 shape=0.0 location=0,0",10)
- 9 : (img_sig1, img_sig2, "color=0.35 texture=0.25 shape=0.4 location=0,0",10)
- 10 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.0 location=0,3",10)
- 11 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.3 location=0,0",10)
- 12 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.3 location=0,0",15)
- 13 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.3 location=0,0",20)
- 14 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.0 location=0,3",15)
- 15 : (img_sig1, img_sig2, "color=0.7 texture=0.0 shape=0.0 location=0,3",20)
- 16 : (img_sig1, img_sig2, "color=0.7 texture=0.3 shape=0.0 location=0,0",15)
- 17 : (img_sig1, img_sig2, "color=0.7 texture=0.3 shape=0.0 location=0,0",20)
- 18 : (img_sig1, img_sig2, "color=0.7 texture=0.3 shape=0.0 location=0,0",19)
- 19 : (img_sig1, img_sig2, "color=0.5 texture=0.5 shape=0.0 location=0,0",15)
- 20 : (img_sig1, img_sig2, "color=0.5 texture=0.4 shape=0.1 location=0,0",15)

La liste complète des résultats obtenus à la suite des expérimentes est présentée dans l'annexe 8.

Les étapes ont été :

- identification visuelle des images identifiant les changements des scènes.
- Action en boucle (dizaines de fois) contenant :
 - Modification des poids associés aux attributs de bas niveaux
 - Lancement module logiciel création BDMM.
 - Lancement module logiciel lecture BDMM (Validation visuelle – Évaluation qualitative)
 - Évaluation quantitative des résultats (calcul rappel, précision, bruit et silence)

En considérant une approche présenté et développé par 4, 25, et 26, identification d'une scène est basée sur l'identification d'un ensemble des plans reliés dans le temps et dans l'espace par leur contenu sémantique.

1	0.3	0.7	0	0	10	7	6	0.8571	0.5	0.143	0.5
2	0.1	0.9	0	0	10	3	3	1	0.25	0	0.75
3	0.1	0	0.9	0	10	32	8	0.25	0.6667	0.75	0.3333
4	0.1	0	0	0.9	10	63	11	0.1746	0.9167	0.825	0.0833
5	0.5	0.5	0	0	10	21	9	0.4286	0.75	0.571	0.25
6	0.7	0.3	0	0	10	26	9	0.3462	0.75	0.654	0.25
7	0.4	0.6	0	0	10	18	9	0.5	0.75	0.5	0.25
8	0.35	0.65	0	0	10	10	7	0.7	0.5833	0.3	0.4167
9	0.35	0.25	0.4	0	10	32	10	0.3125	0.8333	0.688	0.1667
10	0.7	0	0	0.3	10	43	12	0.2791	1	0.721	0
11	0.7	0	0.3	0	10	38	12	0.3158	1	0.684	0
12	0.7	0	0.3	0	15	28	10	0.3571	0.8333	0.643	0.1667
13	0.7	0	0.3	0	20	22	8	0.3636	0.6667	0.636	0.3333
14	0.7	0	0	0.3	15	25	9	0.36	0.75	0.64	0.25
15	0.7	0	0	0.3	20	18	9	0.5	0.75	0.5	0.25
16	0.7	0.3	0	0	15	21	9	0.4286	0.75	0.571	0.25
17	0.7	0.3	0	0	20	8	7	0.875	0.5833	0.125	0.4167
18	0.7	0.3	0	0	19	13	7	0.5385	0.5833	0.462	0.4167
19	0.5	0.5	0	0	15	8	7	0.875	0.5833	0.125	0.4167
20	0.5	0.4	0.1	0	15	13	8	0.6154	0.6667	0.385	0.3333

Tableau 6 – résultats expérimentaux

Le graphique ci-dessous, associé aux valeurs du tableau 6 donne l'aperçu visuel des résultats obtenus, tout en facilitant l'interprétation :

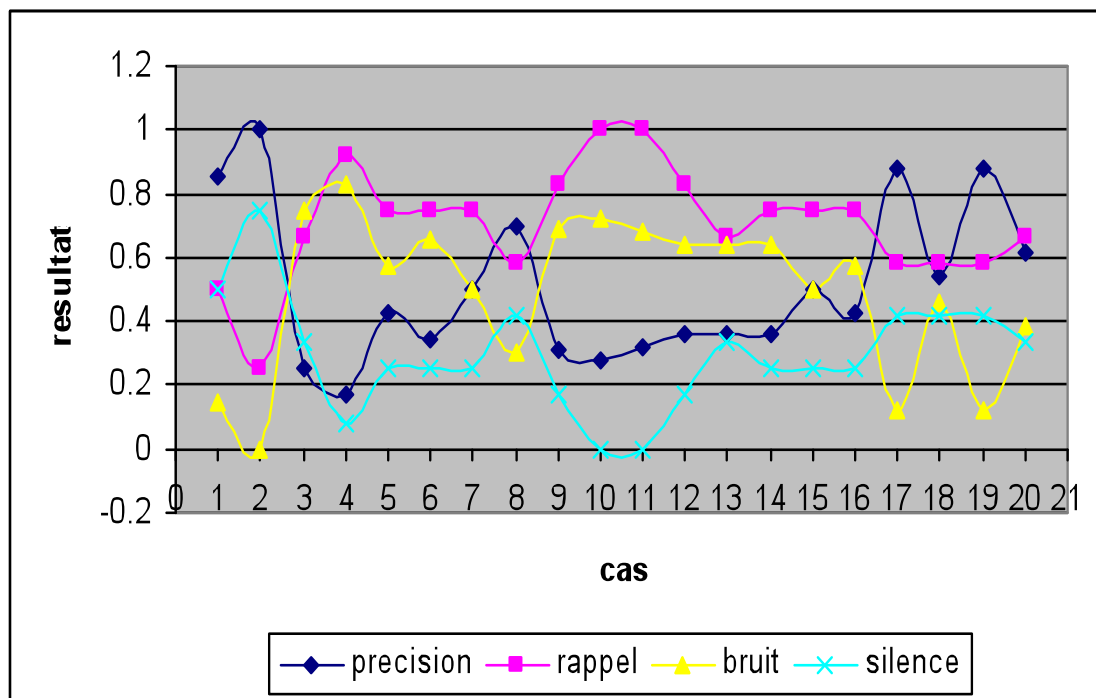


Figure 43: Résultats expérimentaux

Le but de la recherche et des expérimentes est d'obtenir des résultats aussi proches que possible de la réalité (dans ce cas, une réalité subjective, car l'identification des scènes est faite à l'aide de l'oeil humaine, tributaire à une subjectivité dérivée de l'interprétation sémantique du contenu). En observant l'allure des graphiques on peut noter quelques brefs observations :

- la précision et le rappel évoluent en antiphasé (si l'un augmente l'autre baisse) – confirmation de la théorie et en concordance avec les attentes
- Le bruit et le silence évoluent en antiphasé ce qui confirme nos attentes et la théorie.
- Les courbes de rappel et de bruit ont des allures similaires, donc, on approche de la situation idéale (repêcher tous les images identifiées comme début de scène) en payant le prix de ramener simultanément assez d'information non pertinente.
- Les courbes de précision et de silence ont des allures similaires, donc, les plus précises des réponses font rabat à la quantité d'information pertinente ramenée.
- Comme ce qui nous intéresse le plus et le rappel (doublé d'une bonne précision) on peut considérée comme intéressantes les situations offertes par le cas où les couleurs bleu foncé et violet sont au-dessus des couleurs bleu pâle et jaune. On cherche donc de la pertinence sans « trop d'oubli ou trop de bruit ». À cette lumière, les zones susceptibles comme pertinentes sont pour l'abscisse 8, 15, et de 17 à 20.
 - Cas 8 : le rappel et la précision de valeurs comparables, la précision est plus élevée.
 - Cas 17 à 20 : pour cette intervalle, c'est le cas 20 celui qui offre la plus élevée valeur pour le rappel. Le rappel retourné par ce cas est meilleur que celui du cas 8. Considérant le compromis rappel-précision / bruit-silence on peut considérer ce cas comme le meilleur parmi les expérimentes.
 - Cas 15 : offre un rappel très bon, mais est moins précis et plus bruité que les autres deux cas identifiés.
- On peut considérer que, l'augmentation de la valeur associée au poids de l'attribut couleur mène à une augmentation du rappel et de la précision. Cela est aussi valable pour le cas de la texture. Plus précisément, la dualité couleur-texture semble avoir le plus d'impact sur la pertinence des résultats obtenus. Les

résultats obtenus expérimental vient de confirmer les affirmations et les conclusions des références de spécialité. Comme s'est spécifié en [26], la plupart des bases de données de type CBR utilise la notion de couleur, et jusqu'à 90% de CBIR est fondé sur la couleur. Cette approche est utilisée aussi par [4]. Cette fois-ci, l'utilisation de la dominance d'une couleur dans une image (le nombre de pixels d'une certaine couleur divisé par le nombre de pixels total de l'image) permet de détecter les changements des plans, car, souvent, un changement de plan signifie une variation brusque en ce qui concerne la couleur dominante. Le même principe est appliqué en [25], où, pour la segmentation en plans de la vidéo (approche descendante), on utilise l'information globale sur les images, basée sur la couleur et la texture.

4.4 Limites identifiées, améliorations possibles

Les expériences effectuées ont permis d'identifier plusieurs limites du système développé. Sans avoir la prétention d'être exhaustive, la liste qui suit fait une énumération des plus importantes :

- **traitement globale** – une analyse basé sur la couleur donne des résultats satisfaisantes, mais, un telle approche pose des problèmes déjà connues.([25]). Une des plus fréquentes est lié à l'incapacité de traiter de façon pertinente les changements très localisés de courte durée ou celles de longue durée, graduelles, apportant des changements progressives. C'est comme souligne En effet, dans notre cas on observe les deux phénomènes :
- **incapacité de détecter des changements graduels** (par exemple, la différence entre l'image 00026.jpg et 00041.jpg est significative, mais indétectable par le système. Cette transaction lente est "perdue". Heureusement, dans ce cas, on peut considéré les deux extrêmes comme faisant partie de la même séquence de point de vue sémantique) :



Figure 44: transaction lente

- détection des fausses séquences, si un changement local de luminosité c'est produit. C'est le cas de image 00010, souvent considérée comme début de scène car, à cause de l'éclair apparut dans la photo.



Figure 45: changement très localisé de luminosité

Une amélioration possible est celle proposée par [25], qui découpe l'image en plusieurs blocs, analysés de façon indépendante à travers plusieurs images successives. Alors, un changement local affectera seulement un bloc, qui aura seulement un petit pourcentage d'influence sur l'image totale.

- **contrôle humaine indispensable** – le système dépend totalement de la présence humaine, car la validation visuelle est indispensable. Pour le moment, le système, exclusivement basé sur la détection des discontinuités temporelles utilise des descripteurs globaux comme la couleur ou la texture et laisse à coté les aspects liés à la sémantique. Cette carence doit être corrigé par un observateur humain.
- **subjectivité** – La valeur du seuil de similarité est présentement établie à la suite des itérations successives, tout en analysant les résultats et en appliquant la rétroaction corrective. C'est une méthode qui donne des résultats approximatifs, car, l'appartenance d'une image à l'une ou à l'autre séquence doit être dicté par une analyse faite sur une succession des images appartenant à la même séquence. Pour améliorer ce processus, [23] et [25] proposent des seuils de similarité moyens par séquence, en introduisant le seuillage adaptatif, qui tient compte de bruit et des mouvements plus ou moins rapides de la caméra.
- **Approximative** – en effet, l'application est seulement une ébauche, un commencement. Pour les images considérées on stocke l'information dans des fichiers XML représentant un mince sous-ensemble des possibilités offertes par MPEG-7. C'est un aspect à développer en ajoutant plusieurs attributs de bas et de haut niveaux.

4.4 Travaux futurs

Le système développé pourra se constituer comme noyau pour une futur application complexe, traitant l'indexation de la vidéo, la recherche de l'information multimédia et le repérage basé sur le contenu. On peut envisager plusieurs scénarios pour des futurs travaux :

- **Possibilité 1** : développement d'une partie permettant la recherche des images par similarité. On peut avoir :
 - **Recherche à partir d'un modèle** - on analyse l'image modèle et on calcule les valeurs des descripteurs associés. On compare ces valeurs avec celles des descripteurs correspondants existants dans la BDMM (on « *parse* » les documents XML déjà existantes dans la BDMM pour les images tirées de la vidéo, ou on re-analyse les images eux-mêmes) de l'image extrait les traits et on cherche dans la BDMM
 - **Recherche à partir des mots clés** – c'est une approche permettant de faire la liaison avec la sémantique du contenu. Il faut prévoir des modules logiciels permettant une annotation manuelle capable d'attacher des métadonnées sur la signification du contenu. Cette une approche mixte, utilisant à la fois l'indexation manuelle et automatique, susceptible donc d'apporter une considérable amélioration.
- **Possibilité 2** : Considérer une autre dimension de la vidéo, le son. On peut ajouter des modules logiciels permettant le traitement de son. L'analyse croisée image-son pourra donner des bonnes résultats car, dans la vidéo, souvent un changement de plan est traduit aussi par des changement sonores importantes.
- **Possibilité 3** : Avancer progressivement vers un système capable de réaliser le résumé d'une vidéo, comme propose [24]. Ce serait un grand pas, car, on ne doit pas visualiser (ou analyser chaque fois) un vidéo en son intégralité pour extraire l'information voulue. Au-delà d'une apparence facile, cette tâche est bien plus q'une simple association des images fixes. Il faut trouver comment représenter la substance des séquences, d'identifier la sémantique du contenu, d'assurer la synchronisation et le sous-titrage. On peut imaginer des bibliothèques stockant des résumées des vidéos facilitant une recherche rapide, pertinente, interactive

de l'information, avec des requêtes aussi proches que possible du langage naturel.

4.5 Sommaire

Ce chapitre a été dédié à l'étude des résultats expérimentaux. Tout d'abord, on a présenté le contexte d'interprétation, et l'approche utilisée. On a souligné l'importance de la présence du facteur humaine pour la réussite du processus. L'analyse des résultats obtenus, (comparaison avec les attentes, l'interprétation des écarts, identification des causes) a permis d'assurer le feed-back nécessaire à l'amélioration du processus, d'augmenter le rappel et la précision jusqu'au valeurs acceptables.

À l'occasion de l'analyse faite, plusieurs limites ont été identifiées. On a essayé de comprendre le pourquoi de leur existence et plusieurs améliorations possibles ont été proposées.

Finalement, on a brièvement présenté des éventuels scénarios pour le développement du système réalisé.

5. CONCLUSION

Le projet présente mes premiers travaux dans l'analyse de la vidéo, et j'espère, qu'il encouragera des futures actions vers la construction d'un système d'indexation et de repérage des séquences audiovisuelles. En ce sens j'espère que ma contribution sera utile pour les futurs développeurs du système.

Ayant comme principale motivation l'identification et annotation des images représentatives des séquences d'une vidéo, ce projet propose un système d'analyse des images tirées d'une vidéo afin de faire l'annotation des descripteurs inter-séquences.

Parlant de l'information en général, et plus précisément de domaine de multimédia, deux aspects ont devenu préoccupantes pour les spécialistes : comment accéder à l'information (rapidité, pertinence) et comment le rendre disponible (interopérabilité). Ce projet se propose de toucher un peu les deux :

- On accorde grande importance aux métadonnées, qui ont une importance primordiale pour le multimédia. Les métadonnées peuvent augmenter la performance du processus et donnent un aide en temps réel pour l'utilisateur, car, un changement du média apporte automatiquement un changement des métadonnées associées. C'est pour cela qu'on propose une modalité d'extraction automatique des descripteurs de bas niveaux.
- Il est nécessaire d'assurer l'interopérabilité de partage et l'échange de l'information. Cela signifie l'alignement aux standards internationaux, comme c'est MPEG-7. En effet, l'apparition du standard MPEG-7 a influencé de façon majeure le CBR. Il offre une standardisation de contenu (descripteurs de bas niveau et haut niveau) permettant la réalisation des mécanismes d'interrogation interopérables. C'est à cette lumière que, dans l'approche proposée par le système développé, on stocke les caractéristiques identifiant les images directement dans la BDMM, dans des fichiers MPEG-7.

Plusieurs aspects susceptibles à contribuer à la réussite ont été considérés :

- stockage : le mécanisme de stockage joue un rôle important quant à la performance du système multimédia. On a opté donc pour un environnement de travail performant, basé sur SGBDR Oracle 10g.
- outils de travail :

- méthode permettant de quantifier la similarité entre deux objets multimédia – méthodes et fonctions offertes par Intermedia d'Oracle 10g
- génération automatique des fichiers MPEG-7 (Java, JAXP)
- programmation orienté objet, GUI, requêtes, connexion à la BDMM (Java)
- validation de la sémantique du contenu – boucle de rétroaction assurée par un observateur humain.

Inévitablement, le système a des limites, à la fois techniques, liés à la réalisation ou à la conception. C'est en système nécessitant évidemment des améliorations, quelques-unes déjà identifiées, autres à découvrir.

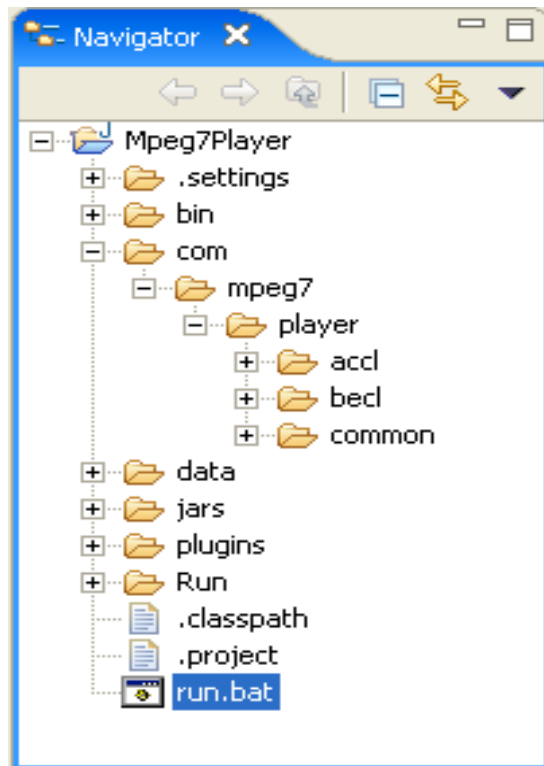
Les résultats obtenus ont été encourageants et les perspectives s'avèrent plutôt intéressantes. On peut considérer ce projet comme une première contribution à la réalisation d'un système d'indexation et de repérage multimédia, permettant la mise en place d'un environnement capable de travailler de façon mixte, inter reliée, à la fois spatiale et temporelle. Le futur permettra peut-être d'ajouter des éléments permettant d'identifier et de retracer la sémantique du contenu. Ce sera un grand pas, car, l'utilisateur sera dorénavant capable d'utiliser à la fois des mots clé, des index de mots clé, du contenu sémantique et des objets multimédia pour effectuer sa requête.

Annexes

ANNEXE 1 - Analyse application Autriche – approche technique

Les suivantes aspects ont été identifiées à la suite de l'analyse faite sur l'application :

1. Le lancement de l'application est faite à l'aide d'un fichier de commandes (run.bat) trouvé dans le répertoire principal. Ce fichier établit le chemin (PATH) vers les bibliothèques utilisées. La dernière ligne de commande lance en exécution une classe java qui reçoit comme paramètre le nom du fichier XML (/data/nightwish.xml).
 - a. Les constructeurs de la classe appellent une méthode qui construit le GUI pour l'application
 - b. Dans le fichier XML on retrouve des informations concernant la source
 - i. Type : Videotype
 - ii. Localisation :
 - iii. Titre :
 - iv. Descripteurs sémantiques
 - v. Descripteurs de décomposition temporelle (start frame+durée, tranches et texte associé)
 - c. À titre d'exemple, on a 6 parties, 3 répétitions et un solo
2. Structure arborescente des répertoires :



3. **Le répertoire ...\\Player\\Data** contient les composants définissant l'utilisation de parser XML, JAXB (Java Architecture for XML Binding). On y trouve des fichiers concernant la source analysée (le fichier nightwish.xml) et des schémas associés (audio_2001.xsd, definitions2.xsd, visual_2001.xsd, mds_2001.xsd, xml_1998.xsd, mpeg7_2001.xsd, ddl_2001.xsd). JAXB est une technologie permettant de générer automatiquement des classes et sous classes Java à partir de (ou correspondant à) éléments d'un fichier XML. En grandes lignes, les principales caractéristiques du JAXB sont :
 - a. S'occupe de façon transparente pour l'utilisateur de l'opération de conversion et migration
 - b. C'est une modalité OO de travail avec XML
 - c. Les classes sont générées en respectant les règles de validation (utilisation d'un compilateur)
 - d. 2 opérations principales : unmarshalling et marshalling.

La ligne de commande est : **xjc -b binding.xjb -extension definitions2.xsd**,
selon le modèle générale : **xjc [options] <schema>**

Ou :

- **-b<fichier>** : spécifie un ou plusieurs fichiers « binding ». On peut avoir un seul fichier avec les particularités des plusieurs schémas (notre cas, le fichier binding.xjb), ou plusieurs fichiers, pour chaque schéma.
- **-extension** : option qui permet flexibilité quant à l'utilisation des extensions non compatibles avec JAXB v1.0;

En ce qui concerne l'utilisation de JAXB, on a deux possibilités de faire cette adaptation des fichiers¹¹ :

- adaptation en ligne (*inline customization*)
- utilisation de fichier paramétrage externe (*external binding customization*)

Dans notre cas, il s'agit de la deuxième situation, avec fichier externe de paramétrage qui spécifie le schéma distant (definition2.xsd), le package utilisé (com.mpeg7.player.becl.parser.structure), et le suffixe utilisé (MyTyp). :

```
<jxb:bindings version="1.0" xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xjc
="http://java.sun.com/xml/ns/jaxb/xjc" jxb:extensionBindingPrefixes="xjc">
  <jxb:bindings schemaLocation="definitions2.xsd" node="/xs:schema">
    <jxb:globalBindings>
      <xjc:typeSubstitution type="complex"/>
    </jxb:globalBindings>
    <jxb:schemaBindings>
      <jxb:package
name="com.mpeg7.player.becl.parser.structure"/>
      <jxb:nameXmlTransform>
        <jxb:typeName suffix="MyTyp"/>
      </jxb:nameXmlTransform>
    </jxb:schemaBindings>
  </jxb:bindings>
</jxb:bindings>
```

¹¹ <http://java.sun.com/webservices/docs/1.5/tutorial/doc/JAXBUUsing4.html>

</jxb:bindings>
</jxb:bindings>

- Répertoire **ACCL** :
 - Sous répertoire **GUI** :
 - Un répertoire avec des images (nommé .../GUI/IMAGES) pour le dessin de GUI; le chargement est fait avec pictload.java existante en\common\tools .
 - Des classes java pour la construction de l'interface :
 - ControlPanel.java : utilisation d'un certain style, définit et importé ici de ... \common\styleSheet.java
 - « chargeur des images » - ... \common\tools\pictLoad.java
 - « dessinateur des panels » - ... \common\tools\Pictureborder.java
 - Filtre des fichiers acceptés :(seulement des fichiers avec une certaine extension, .jpg; .gif) – ExampleFileFilter.java
 - MainPanel.java : utilise SCHEMA comme paramètre pour accéder à un fichier externe des description schémas : MPEG-2001.xsd
 - Marker.java : construction de la ligne de temps (position en bas et à droit sur l'écran)
 - Menubar.java : construction menu (en haut à gauche – data, open, close, view, showMenu, showPlug, reloadP)
 - Menupanel.java : construction du panel à l'aide des images de sous-repertoire ... \Images.
 - PlayerPanel.java : Construction du player . Le constructeur reçoit un objet de type MainPanel et applique une méthode (initMe()) .
 - Pluginpanel.java : Similaire à Playerpanel.java, mais pour les *plugins*.
 - Rootpanel.java : c'est une extension de MainPanel. Son constructeur lance initMe() et une autre méthode qui retourne une chaîne (de forme parseXML) en recevant

comme entrée le fichier. La variable utilisée, “schéma”, est initialisée en superclasse comme étant la description du PEG7-2001.xsd (qui s’occupe simultanément de la description de données, de la composante vidéo, celle audio et les outils MDS en utilisant ddl-2001.xsd, visual-2001.xsd, audio-2001.xsd et mds-2001.xsd). À la suite de l’analyse faite sur la chaîne VALIDATE on applique un des trois méthodes pour obtenir le “parseFile” (validatorDOM, validaorXDK ou validatorSchema). Le résultat constitue paramètre pour la classe JAXBparser.java, qui reçoit un fichier XML et réalise l’opération “unmarshalling” ayant comme sortie un document MPEG7. Les possibles erreurs sont captées et affichées à l’écran.

- statusPanel.java : étiquette significative (défaut- OK)
- TimelinePanel.java : construction dérouleur fichier

- Sous répertoire **JMF** :

- ClockObserver.java : constructeur qui reçoit un “Player” et un entier (combien observateurs pour le player)
- MediaClocklabel.java : le constructeur de la classe reçoit un player p et lance une méthode qui affiche le temps en format mm:ss (à partir de MediaTime divisé par 1000000)
- MoviePlayer.java : le constructeur reçoit un panel de format MainPanel.java et initialise un player. Il le positionne sur la panel en lui associant un MediaClockLabel (format mm:ss) et des boutons de play, stop, pause. (aussi le traitement des événements)

- Sous répertoire **PLUGIN** :

- MultiPluginPlugin.java : utilise un constructeur qui appelle la superclasse Plugin (trouvée dans le répertoire ...becl\plugin\ comme plugin.java)

- Répertoire **BECL** :

- Sous répertoire Clock :
 - class clock qui utilise la classe MainPanel et autres méthodes pour administrer le temps écoulé (getTime()=obtention du temps, reset()= mise à zéro, inc()=incrémenter avec une valeur passée comme paramètre, dec()= décrémenter avec une valeur, setTime()=fixé une valeur de début pour le temps.)
- Sous répertoire datamanager :
 - class Manager.java : le constructeur de la classe reçoit comme paramètre un « mapper » (La source est la classe mapper.java de répertoirebecl\mapping). La classe utilise aussi des méthodes pour administrer l'ajout des *plugins*. L'Ajout est faite de façon paramétrée (utilisation des méthodes jump(who,int,int) et step(who,int,int)).
- Sous répertoire MAPPING
 - Class Mapper.java : le constructeur de la class reçoit un MainPanel; il fait l'analyse (parsing) du fichier passé aussi comme paramètre. Des *plugins* sont utilisés en appelant plusieurs méthodes les concernant : showRootPlugin(), getPlugin(), reloadPlugin(), showPlugin(), getActualPlugin().
- Sous répertoire PARSER
 - Class JAXBParser.java : établir le point d'entrée pour l'API JAXB par l'utilisation de « newInstance(contextPath) ». Dans ce cas, contextpath est « com.mpeg7.player.becl.parsr.structure». L'utilisation de « unmarshaller » permet la conversion des données XML dans une structure arborescente d'objets Java. Les possibles erreurs sont déposés dans un tableau des événements de validation et affichés à l'écran
 - Class ValidatorAll.java : utilise SAX d'Oracle¹² et des méthodes déclarées pour interpréter les erreurs de validation. (variables

¹² <http://www.onjava.com/pub/a/onjava/2002/06/26/xml.html?page=2>

associées aux `error(SaxParseException)`,
`fatalerror(SaxParseException)` et `warning(SaxParseException)`

Parenthèse : SAX, ou Simple API for XML, est une façon traditionnelle d'analyse XML. Elle est basée sur les événements rencontrés durant l'analyse du fichier. Celui-ci est lu séquentiellement et des fonctions de rappel (callback fonctions) sont générées au début et à la fin des événements rencontrés. C'est la responsabilité du programmeur d'écrire ces « callback fonctions » en respectant les spécifications de l'interface « `org.xml.sax.ContentHandler` ». Mais, si l'application ne s'intéresse qu'à quelques événements rapportés par l'analyseur XML, la flexibilité est assurée par l'utilisation de la classe « `org.xml.sax.helpers.DefaultHandler`, qui fournit des implémentations vides pour tous les événements. C'est un avantage, car l'utilisateur peut utiliser l'extension (clause « `extend` » en programme) de `DefaultHandler` et réécrire seulement les fonctions de rappel qui l'intéressent. Ceci est en fait le cas de notre application qui utilise `DefaultHandler`.

- Class `ValidatorDOM.java` : reçoit comme paramètre un schéma XML et le document à l'étude. Les possibles erreurs sont sauveées et affichées à l'écran.

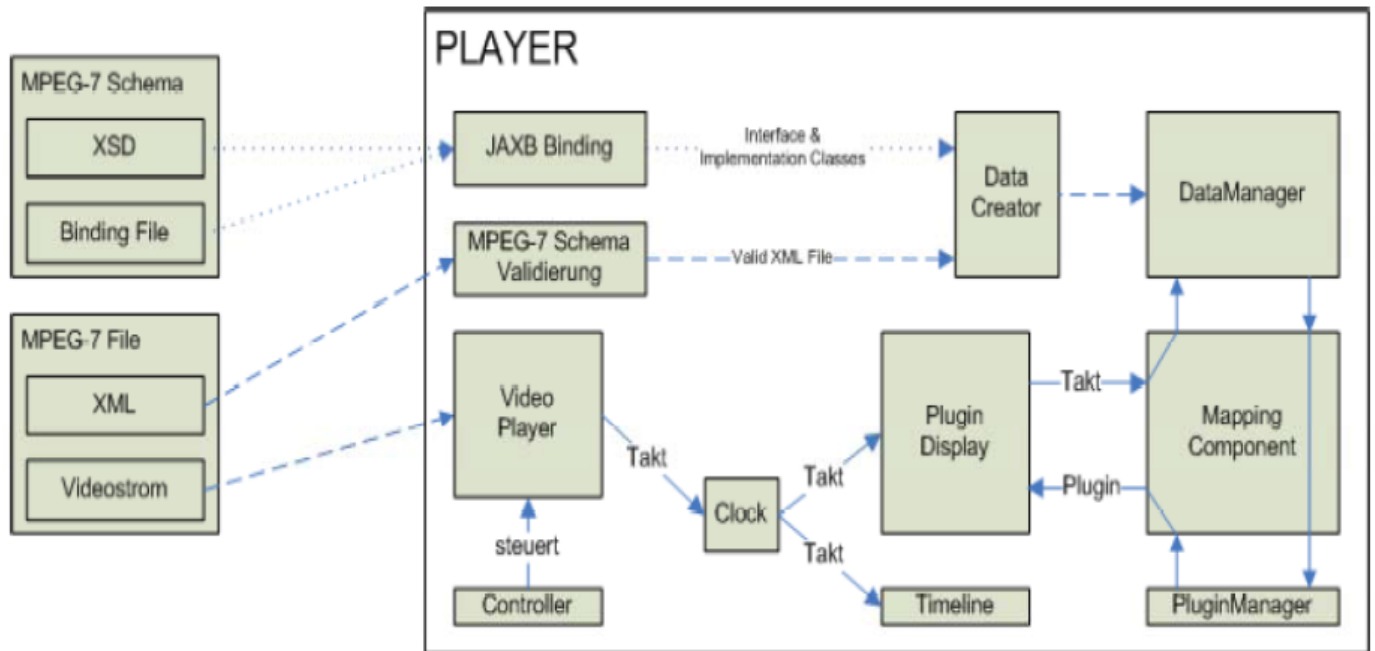
Parenthèse : DOM¹³ ou Document Object Model c'est une technique qui décrit le document XML comme une structure arborescente, ayant chaque élément XML comme un node. Un analyseur XML basé DOM lit le document entier et forme en mémoire l'arbre correspondant. L'arbre DOM est formé de classes qui doivent implémenter toutes l'interface « `org.w3c.dom.Node` », qui permet des fonctions nécessaires pour explorer ou modifier la structure arborescente : (`getChildNodes()`, `appendChild()`, `removeChild()`...) et des méthodes pour interroger chaque node afin d'obtenir ses attributs.

¹³ <http://www.onjava.com/pub/a/onjava/2002/06/26/xml.html?page=3>

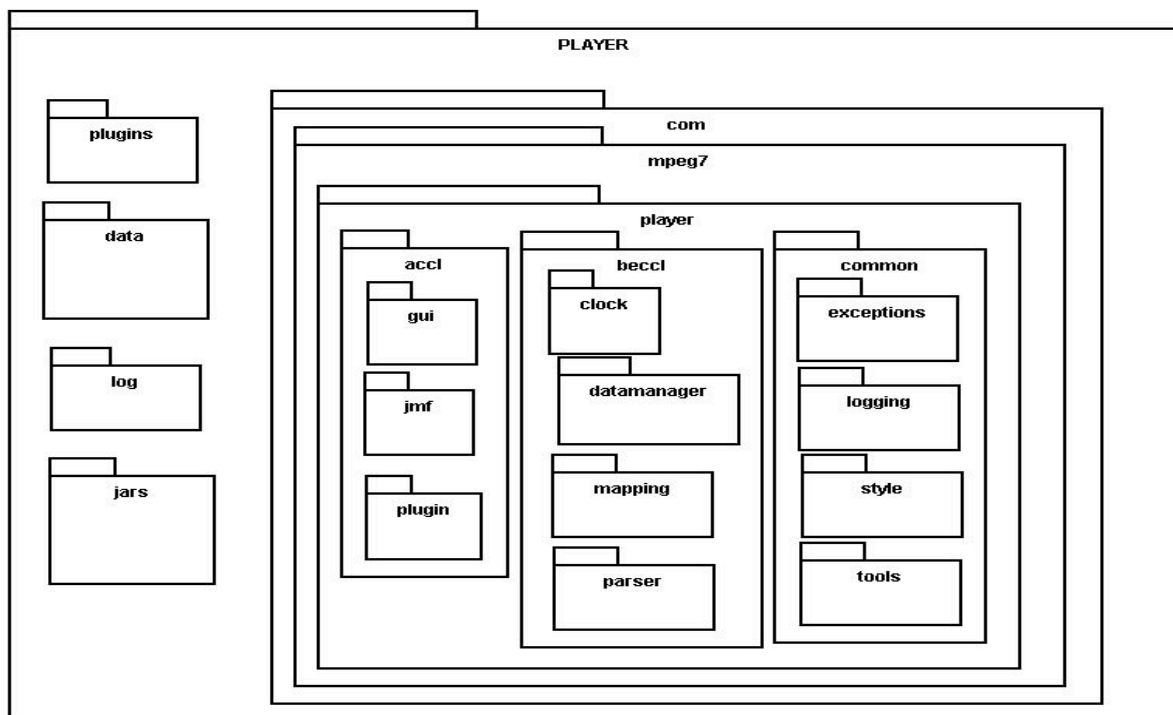
- Sous répertoire PLUGIN
 - Class plugin.java : le constructeur crée le *plugin* (style importé pour Background, foreground, texte). Des méthodes prévues pour ajouter le « mapper » et des champs pour un *plugin* déjà créé.
 - Class DefaultPlugin.java : ayant comme superclasse plugin.java, cette classe a le constructeur qui reçoit le nom de « *plugin* » et ajout ses composants (appel à une méthode de chargement basée sur l'identification de la source : nom et localisation physique)
 - Class pluginField.java : cette classe a trois constructeurs paramétrés pour mettre en action les différents « *plugins* ». (appel simple, appel avec link sans paramètre texte, appel avec link avec paramètre texte)
 - Class pluginManager.java : Le constructeur de la classe utilise une « Hashtable <String, String> » pour déposer les « *plugins* ». Ceux derniers sont chargés ayant comme source le répertoireplugins. Les possibles erreurs de chargement sont sauvées et affichées à l'écran (méthode getPlugin paramétrée).
- Répertoire COMMON :
 - Répertoire exception : utilise trois méthodes, toutes les trois étant dérivées de Exception. (ParsingException.java, "PlayerException.java ValidationException.java). Chaque fois, un message associé est envoyé à l'écran.
 - Répertoire Loggin : logger.java – création et utilisation d'un fichier pour noter les événements → log.html;.
 - Répertoire Style : appel des méthodes pour retourner les attributs des composants : couleur « background », couleur « foreground », dimension player, dimension plugin, control, menu, timeline.
- Répertoire TOOLS :
 - Calc.java : reçoit un chaîne de caractères et reconstitue la valeur en seconde (temps écoulé) - analysant progressive, caractère par caractère.

- mySplitPane.java : classe étendue de JSplitPane. Le constructeur reçoit un entier et deux composantes. Il appelle la méthode de la superclasse : super (entier, composante1, composante2) pour délimiter l'espace sur le canvas.
- picLoad.java:méthode pour charger des images à partir d'une location spécifiée. (ici c'est « com/mpeg7/player/accl/gui/images »)
- PictureBorder.java : class étendue de JPanel; le constructeur reçoit comme paramètres la source (l'image), les dimensions du future panel et un entier (1, 2, 3 ou 4) pour spécifier l'emplacement future : 1 = Nord, 2=Sud, 3=Est, 4=Ouest).
- Répertoire **PLUGINS** :
 - Répertoire contenant plusieurs plugins chargé automatiquement au moment de démarrage du player. Quand le fichier source est analysé (parsing) en utilisant JAXB, les plugins sont initialisés.
 - Les plugins sont stockés en structure arborescente, chacun contenant ou faisant la liaison avec le suivant plugin.
 - Toutes ces plugins sont en fait des JPanels (sont des classes étendues de PLUGIN (situé en ...player./becl/)qui à son tours a comme super class JPanel).

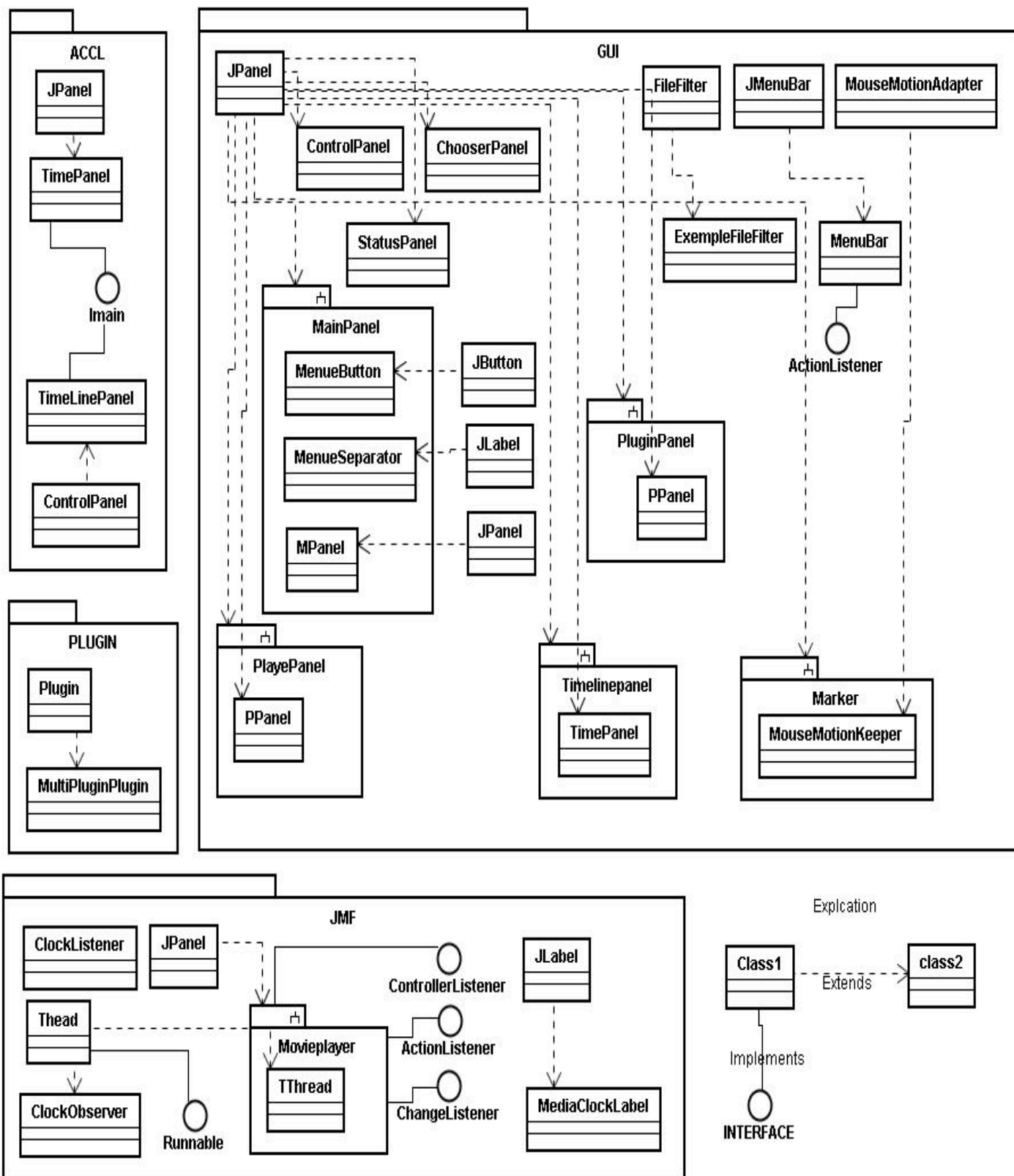
Schéma de fonctionnement :



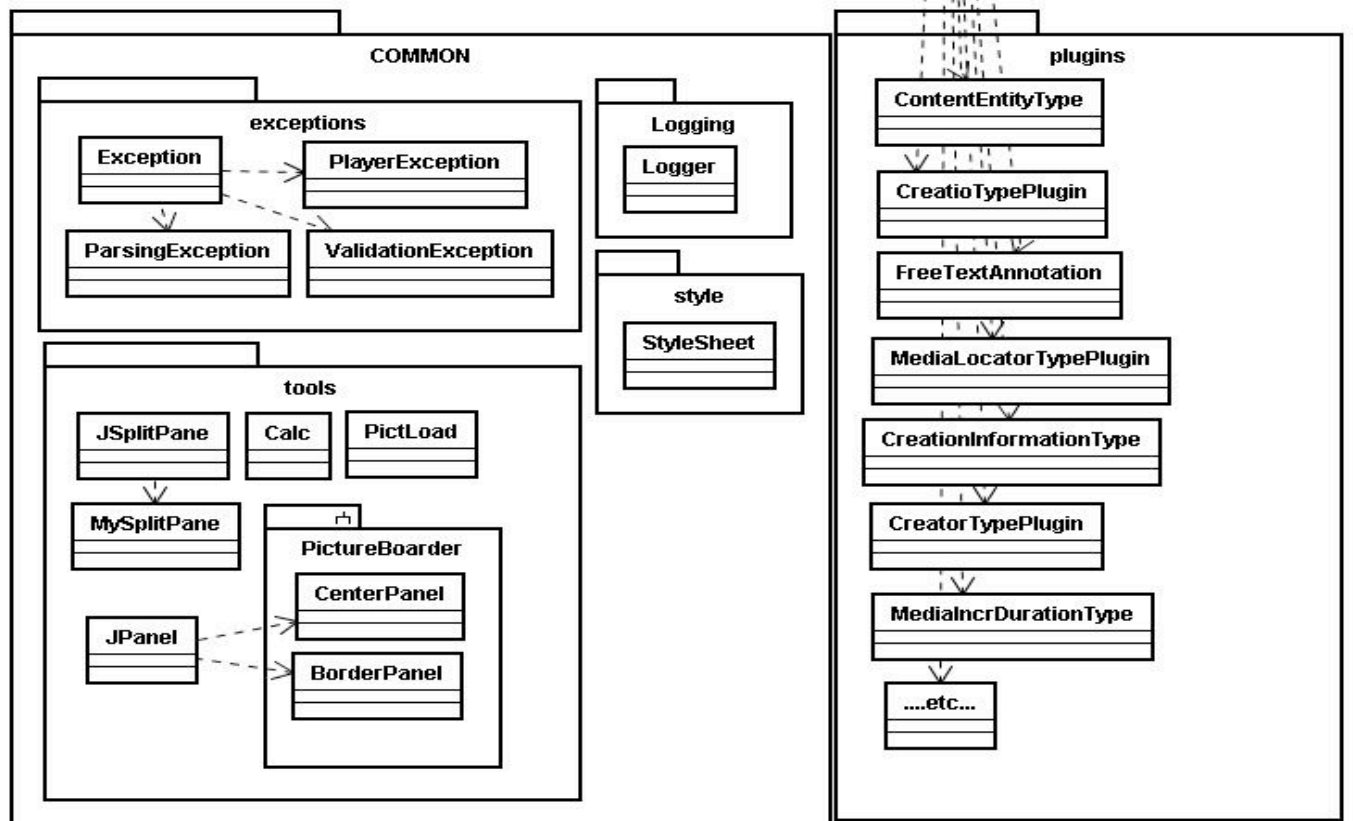
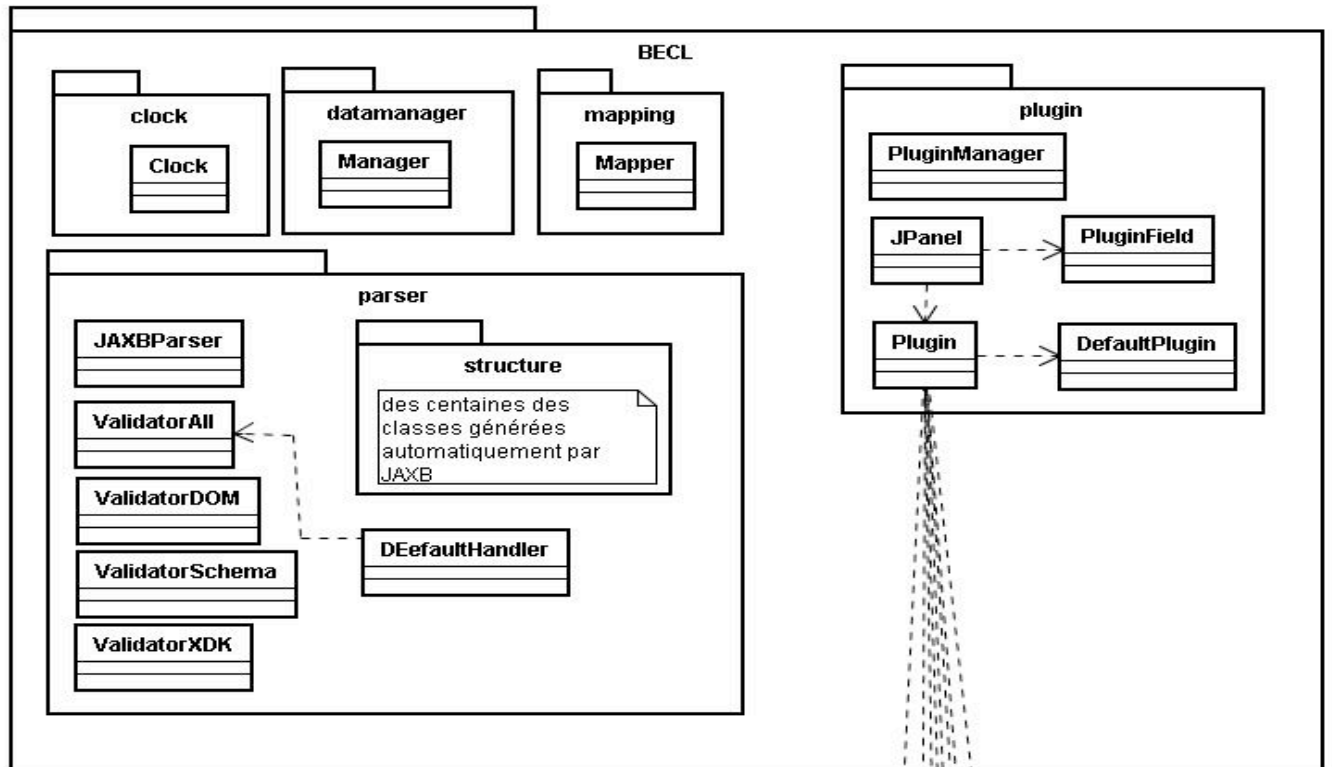
Structure arborescente :



Répertoire ACCL (classes, sous-classes, interfaces Java):



Répertoires BECL, COMMON, PLUGINS (classes, sous-classes, interfaces Java) :



ANNEXE 2 – Module MEI – diagrammes UML

Diagrammes UML : diagrammes des cas d'utilisation, scénarios, diagramme des classes, diagramme de séquences.

Diagramme des cas d'utilisation :

Pour le module d'extraction image, on a identifié :

- acteur : l'utilisateur
- cas d'utilisation : extraction images à partir d'une fichier vidéo

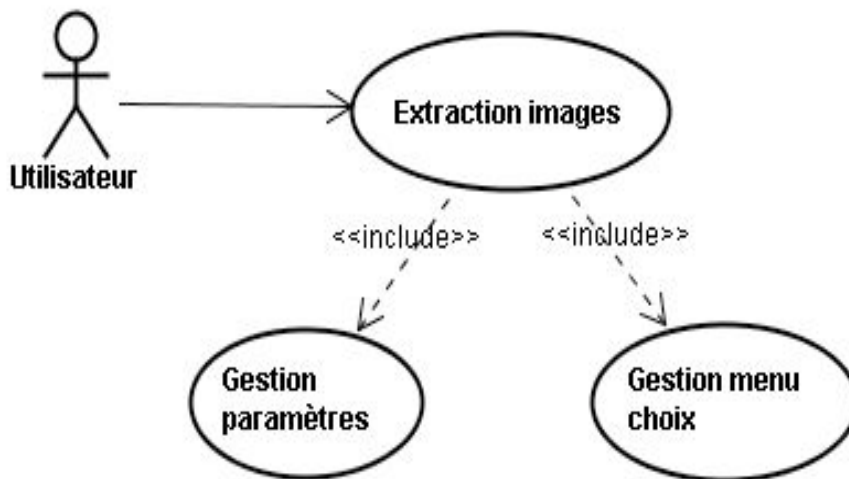


Figure : représentation use case MEI - niveau modèle

Le cas d'utilisation "Extraction images" est activé par l'acteur Utilisateur. Plusieurs scénarios peuvent le composer :

- **Scénario 1** : Choix paramètres
 - L'utilisateur fait son choix quant aux paramètres utilisés par l'algorithme de traitement durant le processus d'extraction images
 - L'utilisateur tente de faire son choix quant le fichier à charger
 - Le système vérifie si les valeurs des paramètres conviennent (type, valeur)
 - Le système autorise la poursuite du processus

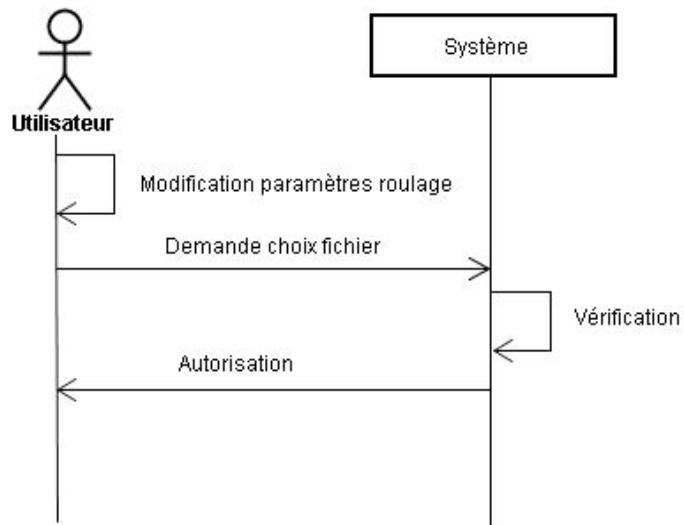


Figure : représentation scénario1 use case MEI - niveau instances

- **Scénario 2** : Extraction images
 - L'utilisateur fait son choix quant au fichier à charger
 - Le système tente de démarrer le lecteur JMF
 - Le système affiche le lancement du lecteur
 - Le système effectue le traitement et la sauvegarde de l'information obtenue
 - Le système affiche le déroulement du processus
 - L'utilisateur est informé de la fin des opérations.

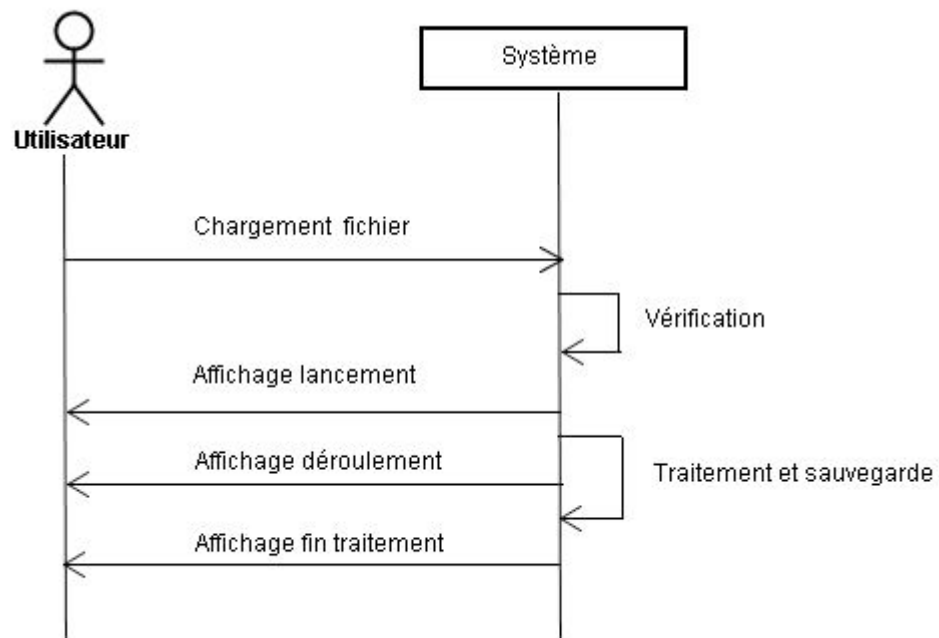
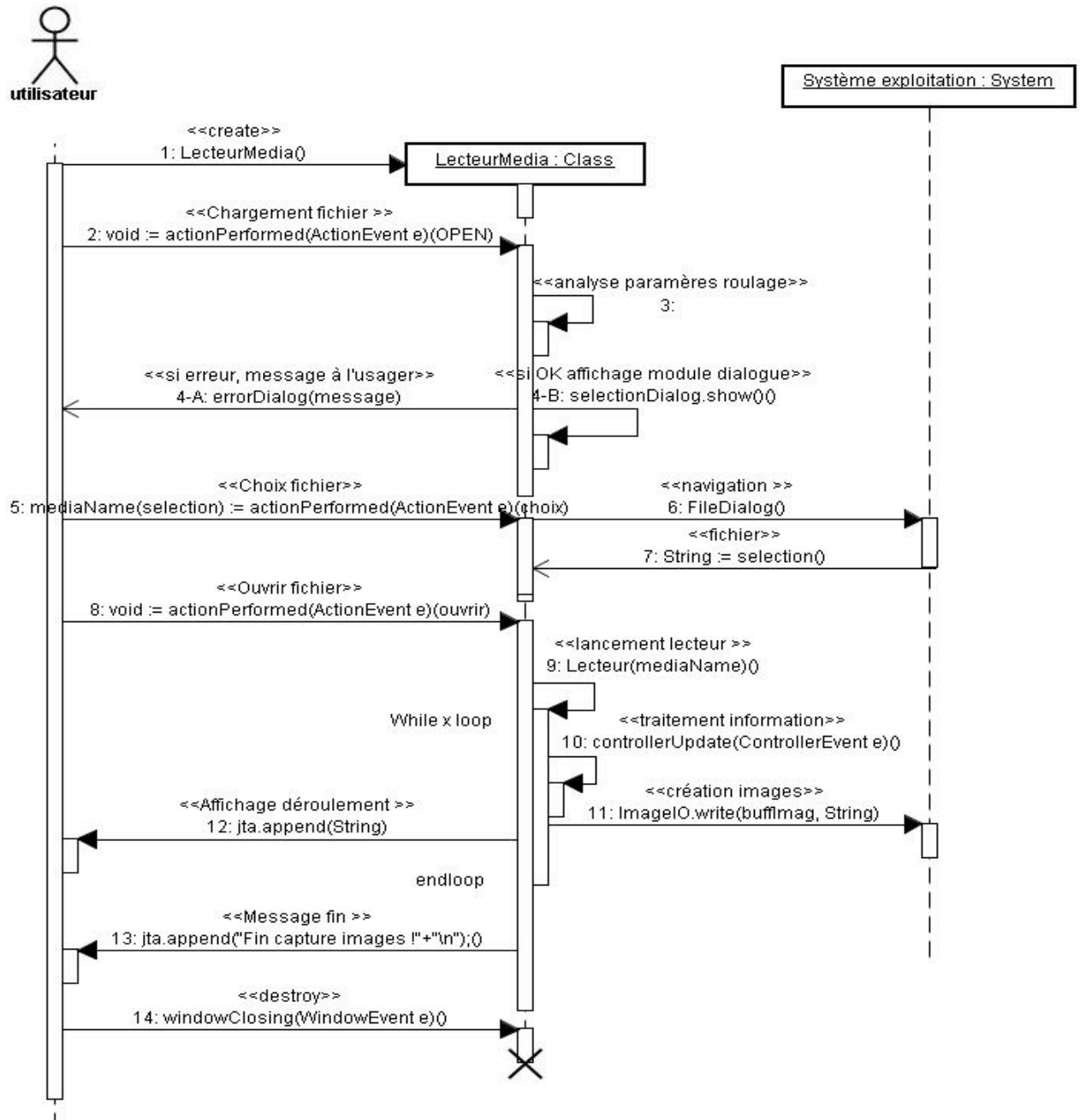


Figure : représentation scénario 2 use case MEI - niveau instances

Diagramme de classes UML : classes, attributs, méthodes
(MEI – module extraction images)

LecteurMedia
JTextField jdur = new JTextField("30",3); JLabel jldur = new JLabel("Durée(secondes):"); JTextField jcad = new JTextField ("10",2); JLabel jload = new JLabel("Frames/seconde(min 1, max 10:)"); JPanel jp = new JPanel(); JPanel pan1 = new JPanel(); Container cp; MediaLocator locator; Player player; Dialog selectionDialog; Button annuler; Button ouvrir; Button choix; Button ok; Button continuer; TextField mediaName; MenuBar bar; Menu fileMenu; Dialog errorDialog; Label errorLabel; Component visualComponent; Component progressBar; Dimension controlSize; Dimension visualSize; JTextArea jta = new JTextArea(20,30); JScrollPane jsp = new JScrollPane(jta); int menuHeight = 50;
LecteurMedia() LecteurMedia(String name) public void actionPerformed(ActionEvent e) protected void Lecteur(String nameOfMedia) protected void errorDialog(String errorMessage) public void controllerUpdate(ControllerEvent e) static public void main(String[] args)

Diagramme séquences UML (MEI – module extraction images)



ANNEXE 3 –Module MTI – diagrammes UML

Diagrammes UML : diagrammes des cas d'utilisation, scénarios, diagramme des classes, diagramme de séquences.

Diagramme des cas d'utilisation :

Pour le module de traitement images, on a identifié :

- acteur : l'utilisateur
- cas d'utilisation : traitement images déposées dans un répertoire donnée.

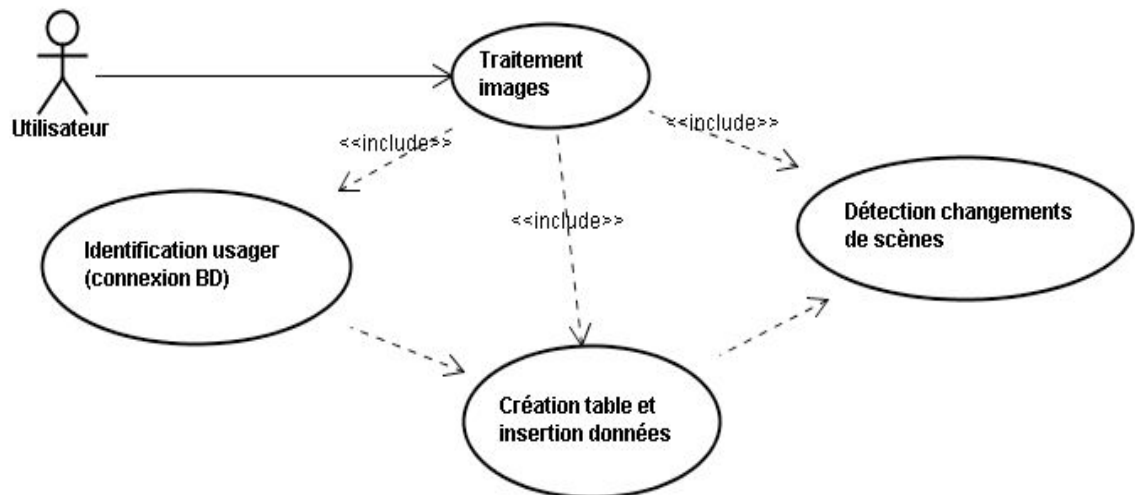


Figure : représentation use case MTI - niveau modèle

Le cas d'utilisation “Traitement images” est activé par l’acteur Utilisateur. Plusieurs scénarios peuvent le composer :

- **Scénario 1** : Identification utilisateur
 - L'utilisateur introduit les données exigées pour se connecter à la base de données (nom usager et mot de passe), et actionne le bouton de connexion.
 - Le système vérifie les données nécessaires pour la connexion à la BD.

- Le système renvoie à l'utilisateur un message d'erreur.

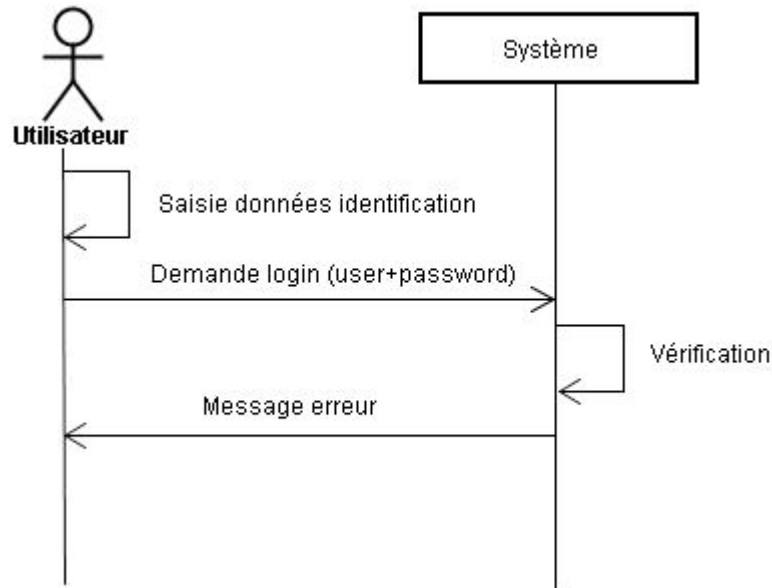


Figure : représentation scénario1 use case MTI - niveau instances

- **Scénario 2** : Traitement images

- L'utilisateur introduit les données exigées pour se connecter à la base de données (nom usager et mot de passe), et actionne le bouton de connexion.
- Le système vérifie les données nécessaires pour la connexion à la BD.
- Le système donne l'autorisation de connexion sur la BD (un message de réussite est affiché dans une zone prévue sur l'interface graphique)
- Le système crée la table Oracle selon les spécifications de la requête
- L'utilisateur est informé de la création de la table
- Le système démarre les opérations de traitement images (lecture, analyse, création fichiers MPEG-7, insertion des données dans la BD) et de sauvegarde informations obtenues.
- L'utilisateur reçoit des messages sur le déroulement des opérations (zone d'affichage déroulante sur l'interface graphique)

- Le système démarre l'analyse de la similarité images déposées dans la BD, selon les spécifications du code.
- L'utilisateur est informé à travers une zone d'affichage du déroulement du processus et de la fin des opérations.

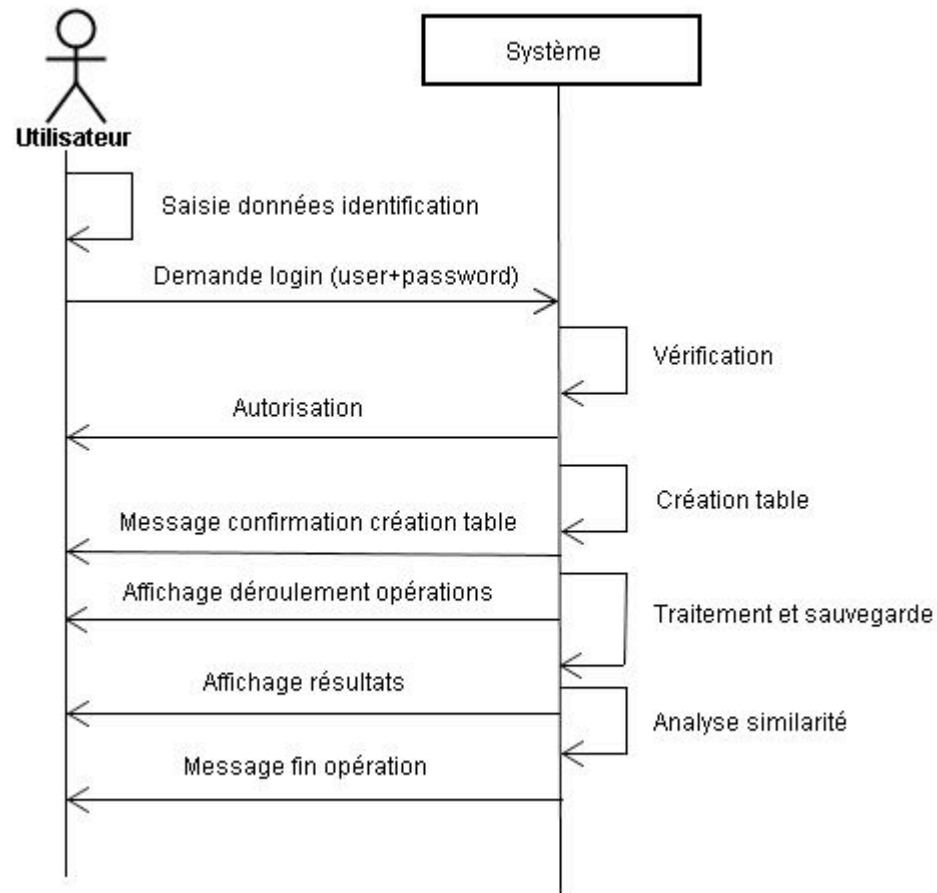


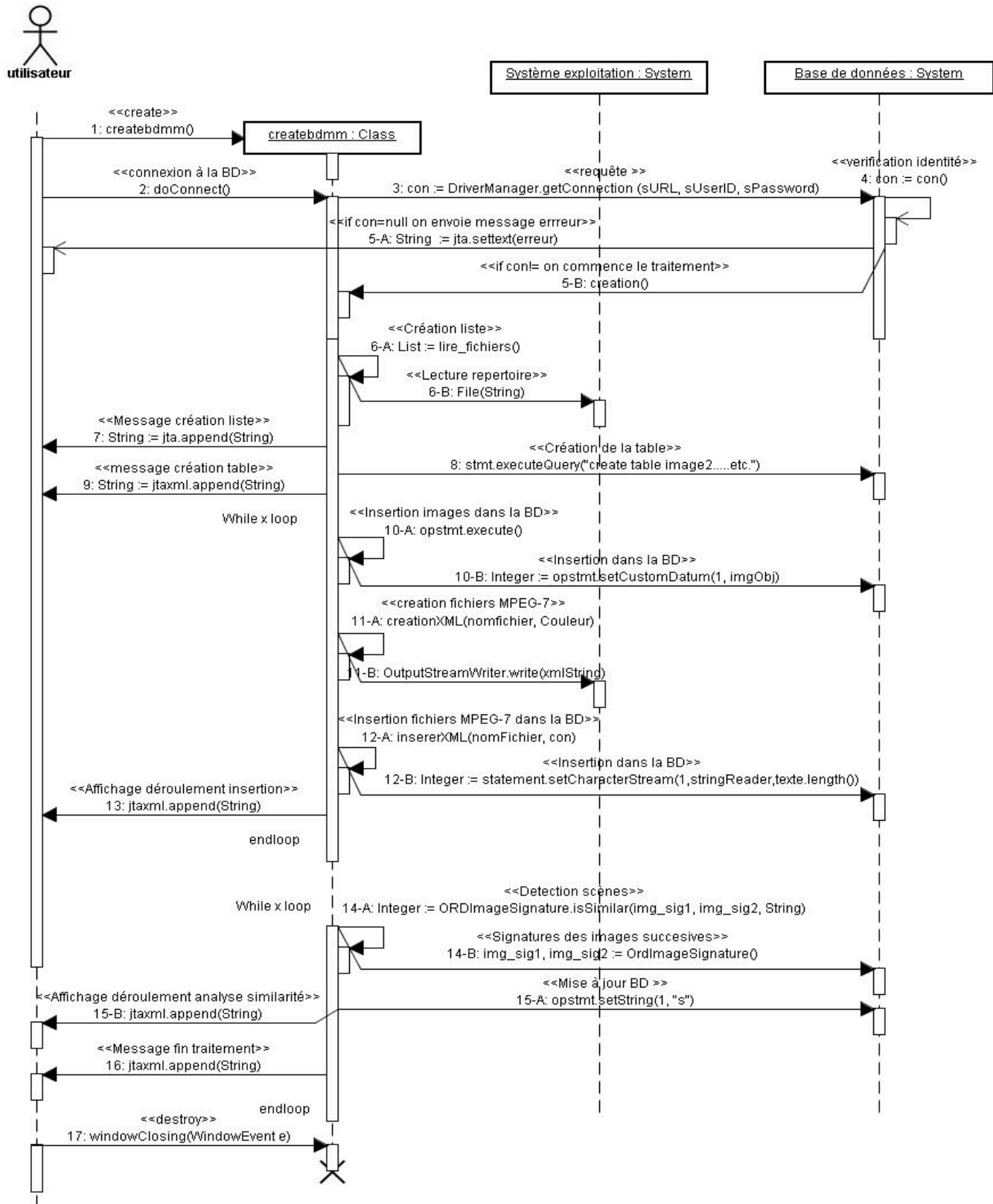
Figure : représentation scénario2 use case MTI - niveau instances

Diagramme de classes : classes, attributs, méthodes

(MTI – module traitement images)

createbdmm
<pre> Blob blob; Connection con = null; int iLength; ImageIcon ii; JPanel jpCenter = new JPanel(); JPanel jpNorth = new JPanel(); JPanel jpSouth = new JPanel(new BorderLayout()); JButton jbConnect = new JButton("Connexion"); JLabel jlUser = new JLabel("Usager:"); JLabel jlPassword = new JLabel("Mot de passe:"); JPasswordField mdp = new JPasswordField(10); JTextArea jta = new JTextArea("Zone erreurs de connexions(si c'est le cas)", 2, 30); static JTextArea itaxml = new JTextArea("",18,25); static JScrollPane scrollpane = new JScrollPane(itaxml); JTextField jtUser = new JTextField(10); PreparedStatement pstmt = null; ResultSet rs; ResultSetMetaData rsmd; Statement stmt = null; String sDriver; String sPassword; String sURL; String sUserID; public createbdmm() public void doConnect() public void actionPerformed(ActionEvent e) public void windowOpened(WindowEvent e) public void windowClosing(WindowEvent e) public void windowClosed(WindowEvent e) public void windowIconified(WindowEvent e) public void windowDeiconified(WindowEvent e) public void windowActivated(WindowEvent e) public void windowDeactivated(WindowEvent e) static public List lire_fichiers() public int compareTo(Object o) static public int[][] conversion(int[] pixel, int imgcols, int imgrows) static public String analyseimage(Image image, int wi, int hi, int[] pixels) static public void creationXML(String nomfichier, String Couleur) static public void insererXML(String nomFichier, Connection con) static public void creation() static public void main(String[] args) </pre>

Diagramme séquences UML :



ANNEXE 4 –Module MV – diagrammes UML

Diagrammes UML : diagrammes des cas d'utilisation, scénarios, diagramme des classes, diagramme de séquences.

Diagramme des cas d'utilisation :

Pour le module de traitement images, on a identifié :

- acteur : l'utilisateur
- cas d'utilisation : Visualiser données BDMM.

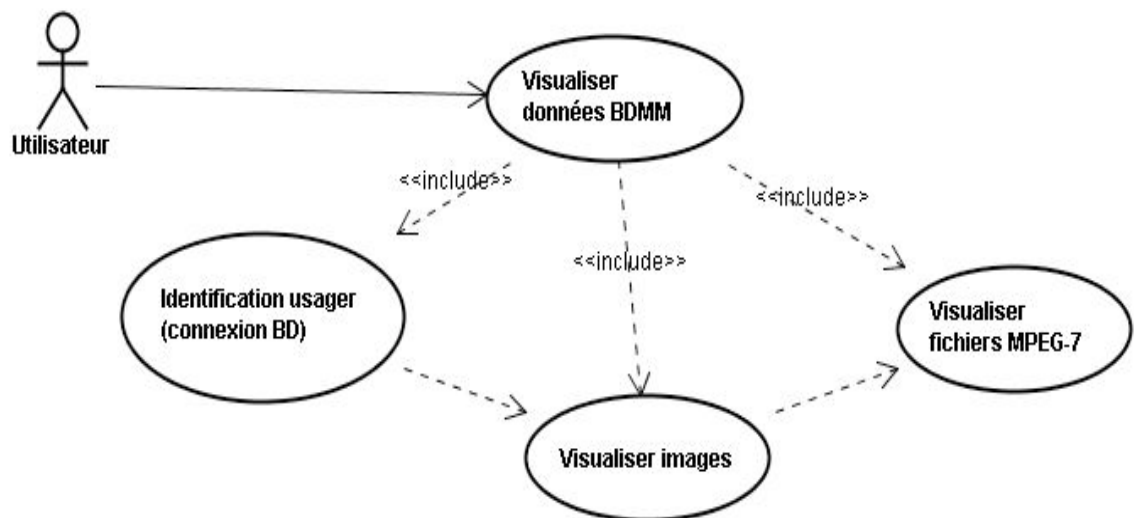


Figure : représentation use case MV - niveau modèle

Le cas d'utilisation “ Visualiser données BDMM ” est activé par l'acteur Utilisateur. Plusieurs scénarios peuvent le composer :

- Scénario 1 : Identification utilisateur
 - L'utilisateur introduit les données exigées pour se connecter à la base de données (nom usager et mot de passe), et actionne le bouton de connexion.
 - Le système vérifie les données nécessaires pour la connexion à la BD.
 - L'usager reçoit un message d'erreur (connexion échouée)

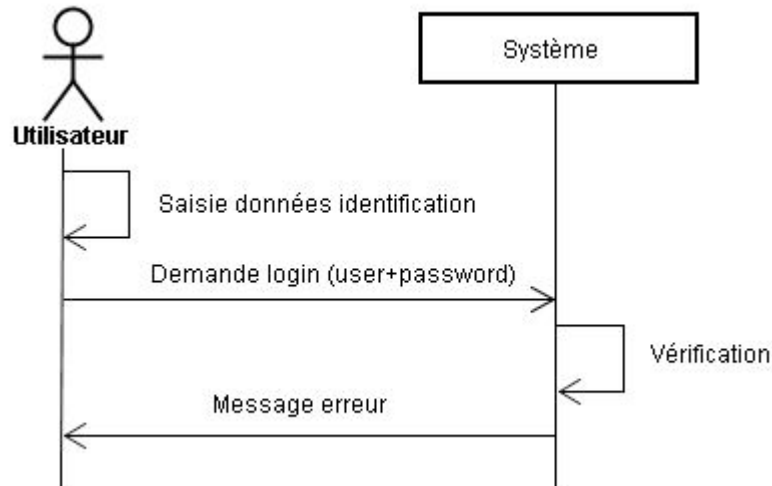


Figure : représentation scénario1 use case MV - niveau instances

- Scénario 2 : Visualisation images

- L'utilisateur introduit les données exigées pour se connecter à la base de données (nom usager et mot de passe), et actionne le bouton de connexion.
- Le système vérifie les données nécessaires pour la connexion à la BD.
- Le système donne l'autorisation de connexion sur la BD (un message de réussite est affiché dans une zone prévue sur l'interface graphique)
- Le système crée deux listes : une contenant toutes les images disponibles dans la BDMM, l'autre spécifiant les images identifiées comme début des scènes
- Le système envoie à l'utilisateur les deux listes (activation des deux listes de choix sur l'interface graphique)
- L'utilisateur actionne le bouton "Visualiser"
- Le système analyse la demande (interprétation des requêtes paramétrées, selon les valeurs courantes des deux listes)
- Le système affiche les images retrouvées dans la BD (deux zones d'affichage prévues sur l'interface graphique, celle de droite étant pour les scènes)

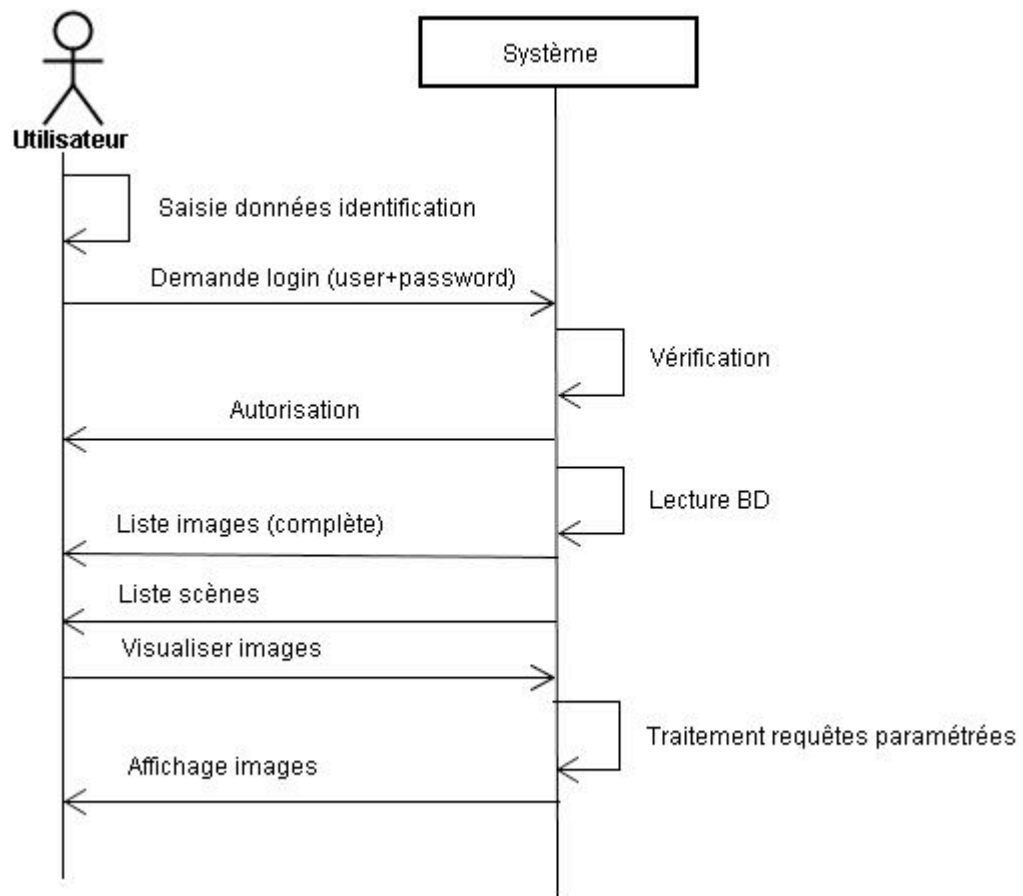


Figure : représentation scénario2 use case MV - niveau instances

- Scénario 3 : Changement choix et visualisation fichier MPEG-7
 - L'utilisateur fait son choix quant à l'image et (ou) la scène désirées, en utilisant les listes déroulantes.
 - L'utilisateur actionne le bouton "visualiser"
 - Le système analyse la demande (interprétation des requêtes paramétrées, selon les valeurs courantes des deux listes)
 - Le système affiche les images retrouvées dans la BD (deux zones d'affichage prévues sur l'interface graphique, celle de droite étant pour les scènes)
 - L'utilisateur actionne le bouton "MPEG-7"
 - Le système analyse la demande (interprétation de la requête paramétrée, selon la valeur courante de la liste des scènes)

- Le système affiche le contenu du fichier MPEG-7 associé à l'image choisie.

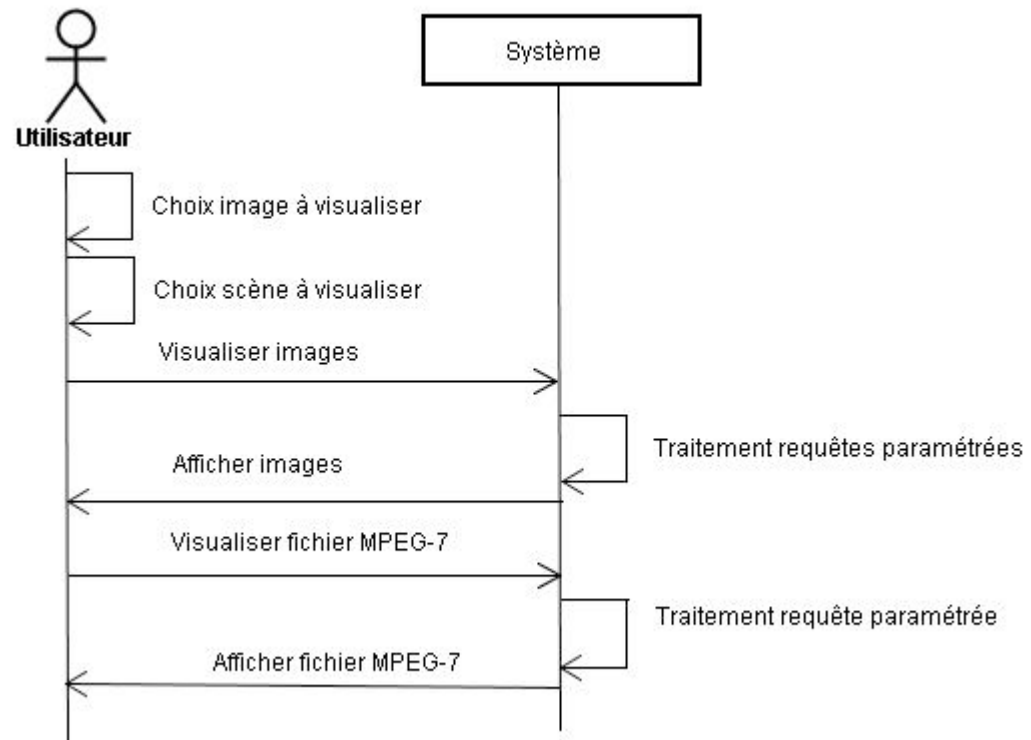


Figure : représentation scénario3 use case MV - niveau instances

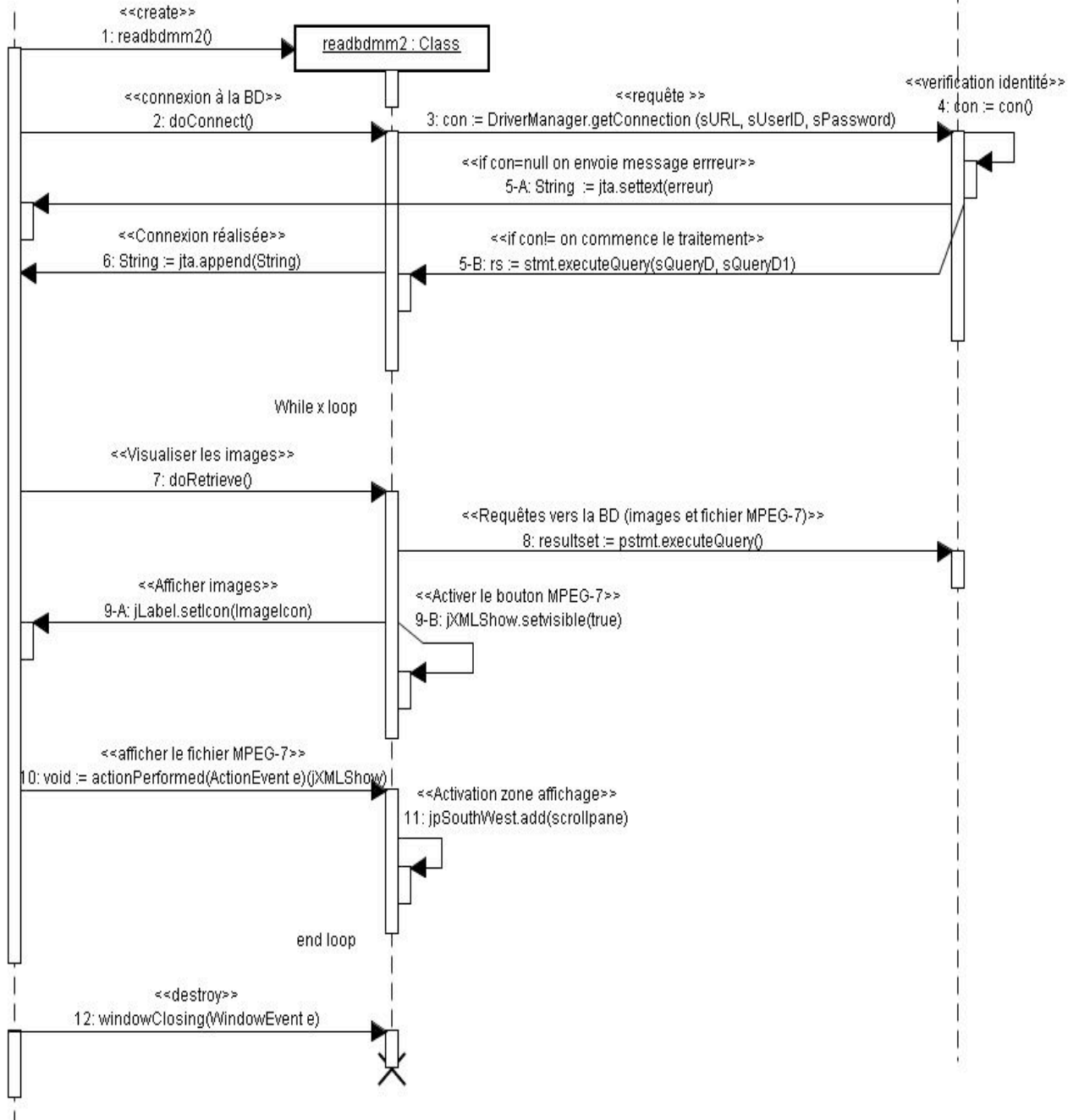
Diagramme de classes UML : classes, attributs, méthodes
(MV – module visualisation)

readbdmm2
<pre> Blob blob; Connection con = null; int iLength; ImageIcon ii; JPanel jpCenter = new JPanel(); JPanel jpNorth = new JPanel(); JPanel jpSouth = new JPanel(new BorderLayout()); JPanel jpSouthEast = new JPanel(); JPanel jpSouthWest = new JPanel(); JPanel jpXML = new JPanel(); JButton jbConnect = new JButton("Connexion"); JButton jbShow = new JButton("Visualiser"); JButton jpXMLShow = new JButton("MPEG-7"); JComboBox jcb = new JComboBox(); JComboBox jcb1 = new JComboBox(); JLabel jlImage = new JLabel(); JLabel jlImage1 = new JLabel(); JLabel jlUser = new JLabel("Usager:"); JLabel jlPassword = new JLabel("Mot de passe:"); JPasswordField mdp = new JPasswordField(10); JTextArea jta = new JTextArea(JTextArea jtaxml = new JTextArea("",18,25);'est le cas)", 2, 30); JScrollPane scrollpane = new JScrollPane(jtaxml); JTextField jtUser = new JTextField(10); PreparedStatement pstmt = null; ResultSet rs; ResultSetMetaData rsmd; Statement stmt = null; String sDriver; String sPassword; String sQueryD = "SELECT DISTINCT imNom " + "FROM image2"; String sQueryP = "SELECT imBlob, xmlmpeg7 " + "FROM image2 " + "WHERE imNom = ?"; String sQueryD1 = "SELECT DISTINCT imNom " + "FROM image2 " + "WHERE Ind LIKE '%s%'"; String srbName = "ConnectJ"; String sURL; String sURLKey = "CSURL"; String sUserID; public readbdmm2() public void doConnect() public void doRetrieve() static public void lectureXML(String nomArticle) public void endApp() public void actionPerformed(ActionEvent e) public void reportSQLException(SQLException SQLe, String s) public void windowOpened(WindowEvent e) public void windowClosing(WindowEvent e) public void windowClosed(WindowEvent e) public void windowIconified(WindowEvent e) public void windowDeiconified(WindowEvent e) public void windowActivated(WindowEvent e) public void windowDeactivated(WindowEvent e) static public void main(String[] args) </pre>

Diagramme séquences UML :



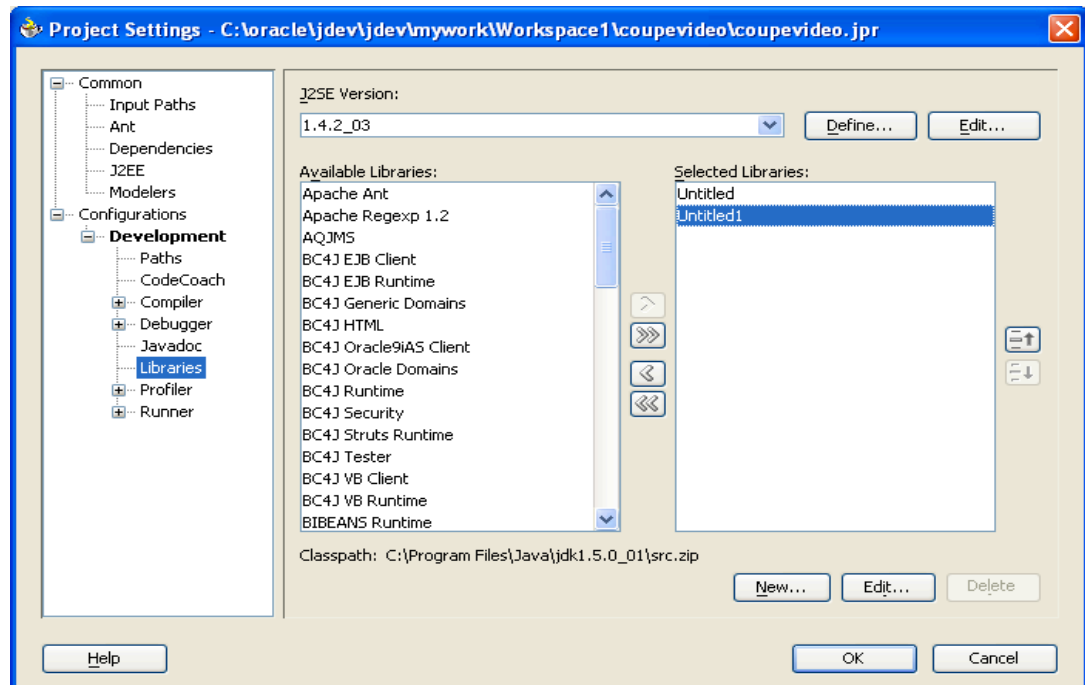
Base de données : System



ANNEXE 5 – Exigences environnement logiciel

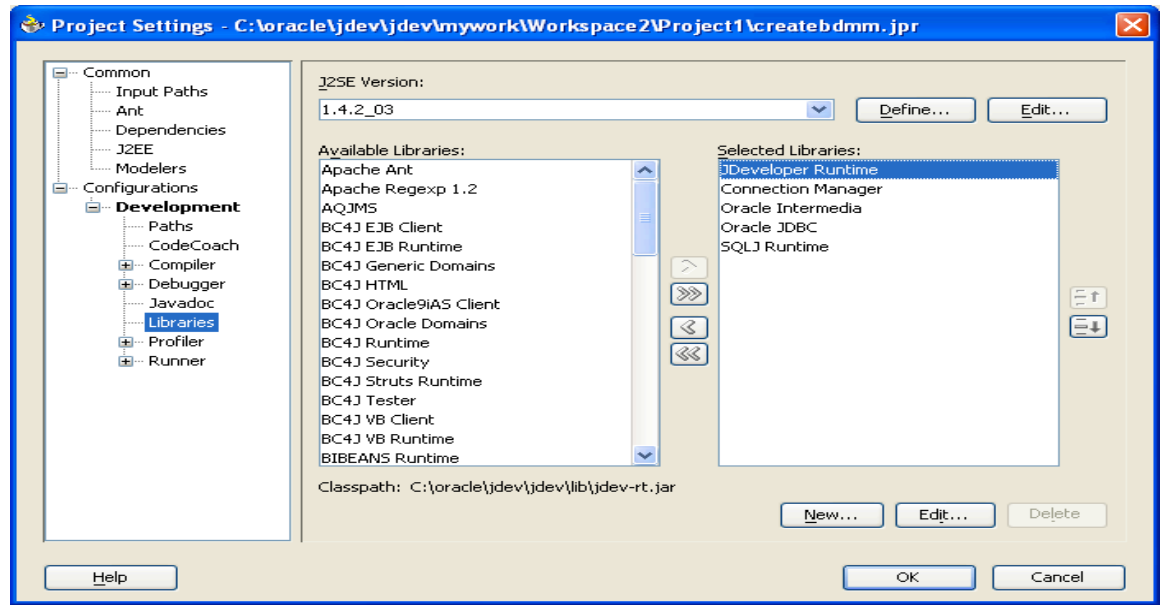
Le module MEI, développé à l'aide de JDEV d'Oracle exige l'accès à deux bibliothèques externes :

- celle de la variante 1.5 de Java : src.zip de JDK1.5.0_01
- celle de JMF 2.1.1 - jmf.jar de JMF 2.1.1 e



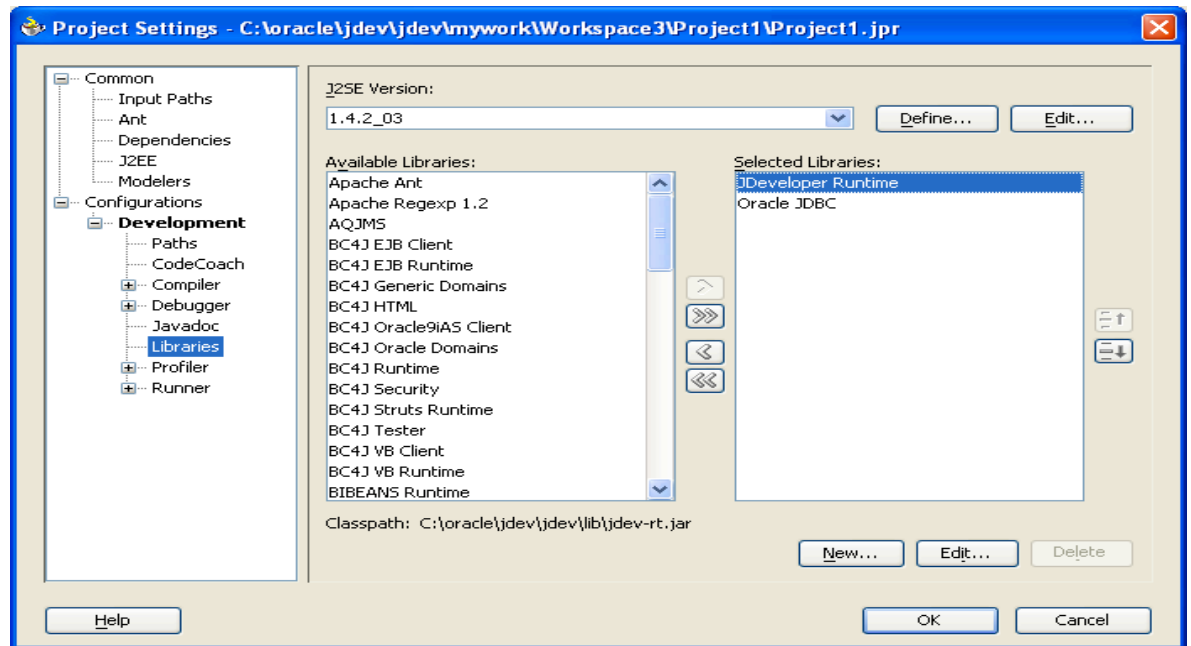
Le module MTI, développé à l'aide de JDEV d'Oracle exige l'accès aux bibliothèques suivantes:

- JDeveloper Runtime,
- Connection Manager
- Oracle Intermedia,
- Oracle JDBC,
- SQLJ Runtime.



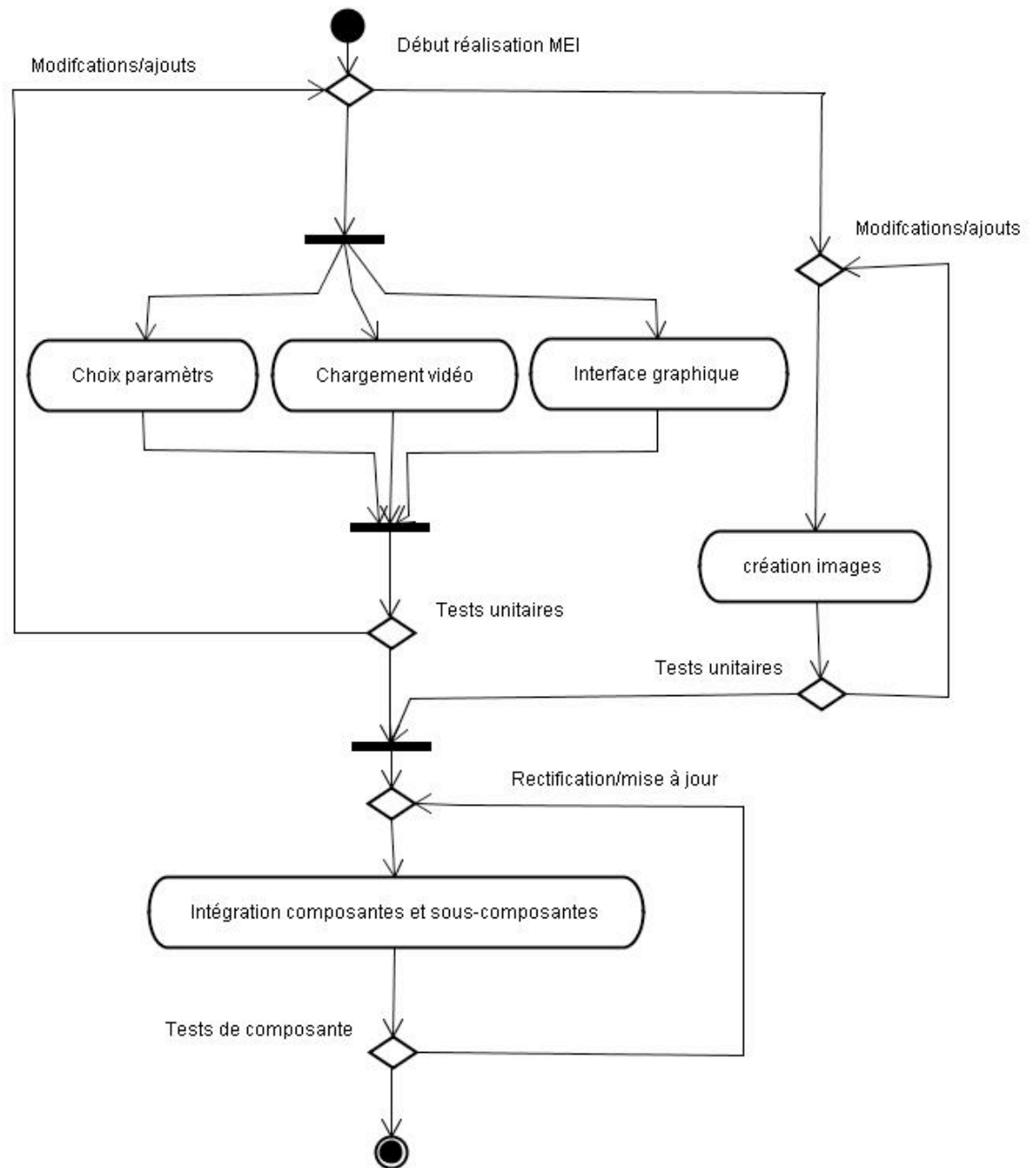
Le module MV, développé à l'aide de JDEV d'Oracle exige l'accès aux bibliothèques suivantes:

- JDeveloper Runtime,
- Oracle JDBC

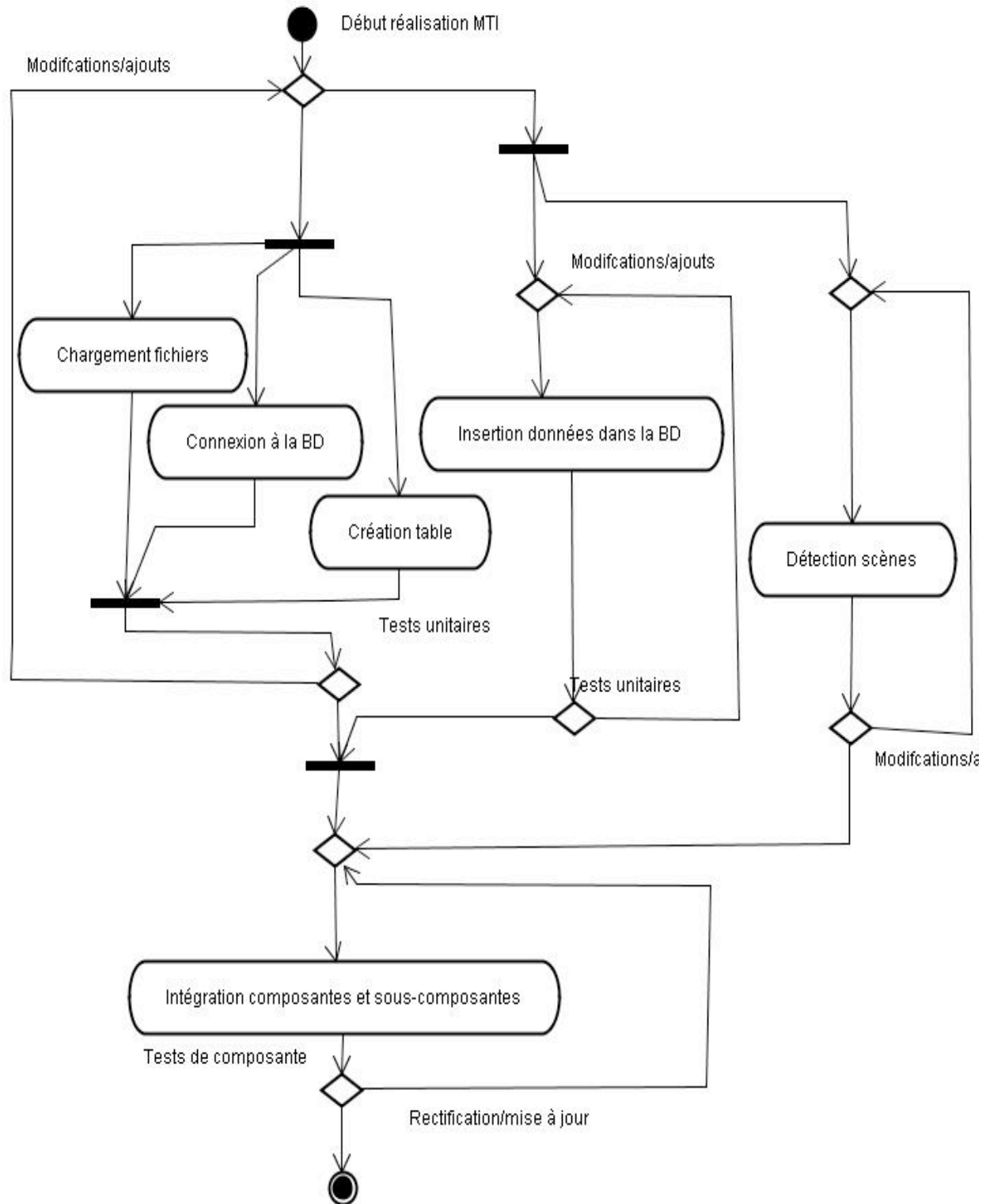


ANNEXE 6 – Diagrammes séquences de travail programmation

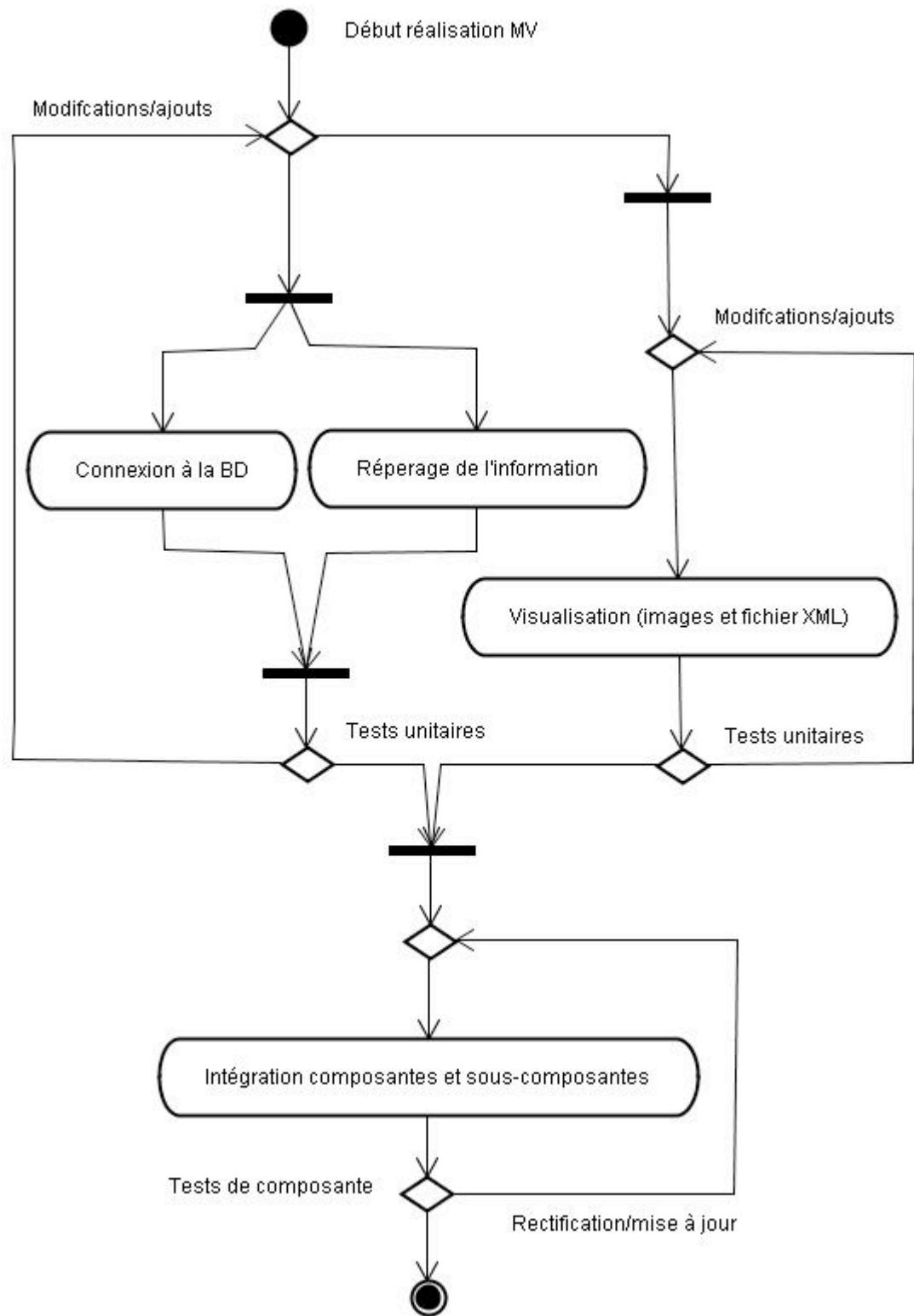
Réalisation module extraction images (MEI)



Réalisation module traitement images (MTI)



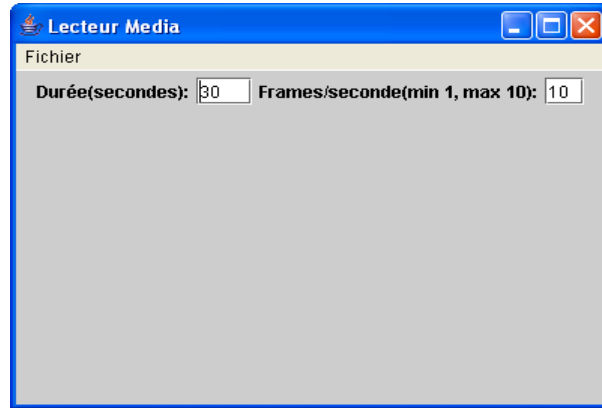
Réalisation module visualisation (MV)



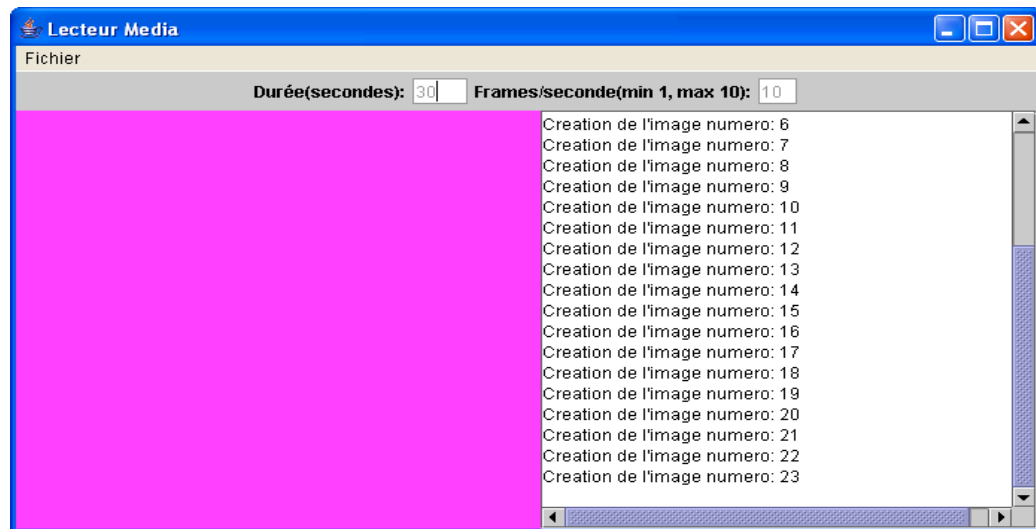
ANNEXE 7 – Vues des scénarios et détails code Java

MEI :

- scénario 1 :



- Scénario 2 :



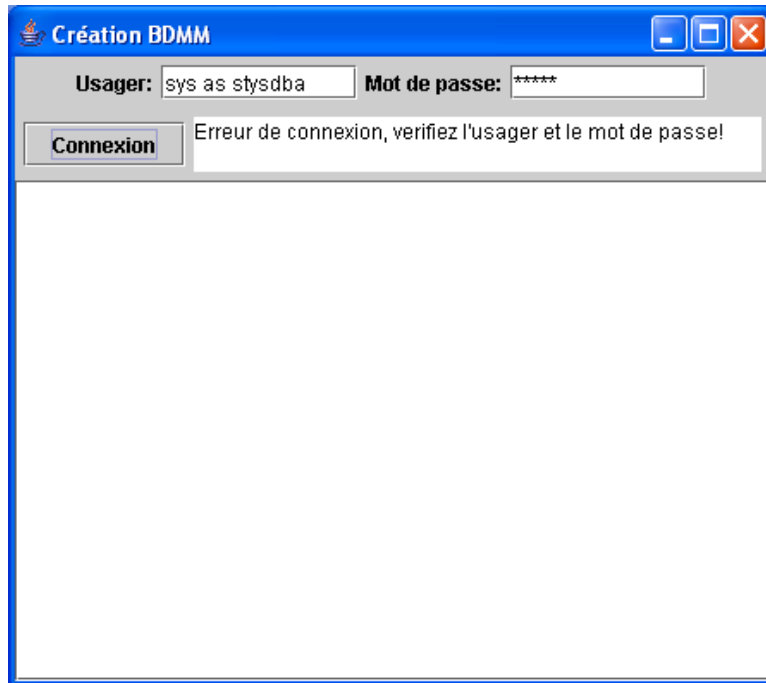
MEI - principales interfaces, classes et méthodes utilisées

- **FrameGrabbingControl** – interface permettant d'obtenir (*grab*) un cadre (*still video frame*) à partir d'un flux vidéo. Le contrôle peut être exporté vers un lecteur en utilisant la méthode *getControl* de la classe *Player*.
- class **BufferToImage** – est une classe utilitaire permettant la conversion d'un objet *buffer* vidéo dans un objet Image AWT. Celui ci peut être utiliser à travers les méthodes disponibles dans la classe AWT (dans notre cas, il s'agit de la méthode *createImage*)

- **BufferedImage** – c'est une sous-classe de la classe **Image** décrivant un objet *Image* par un *buffer* de données spécifiques. Un *BufferedImage* est composé d'un *ColorModel* et un *Raster* des données spécifiant une image donnée. L'avantage de *BufferedImage* consiste dans la création d'une copie de l'image en mémoire, d'où on peut l'utiliser plusieurs fois.
- **Type_INT_BGR** - C'est une représentation d'une image selon le modèle de couleurs RGB (chaque pixel est spécifié par un nombre entier, qui est la codification sur 8 bits des trois couleurs de base *r-red, g-green, b-blue*)
- **drawImage**(image, AffineTransform xform, ImageObserver obs) – permet de dessiner une Image créée par *BufferedImage*. Permet aussi d'appliquer une éventuelle transformation pour l'image donnée.
- **ImageIO** – classe contenant des méthodes permettant la lecture et l'écriture des images (aussi des méthodes pour simple codage et décodage)
- **ImageIO.write**(buffImg, "jpg", new File("c:\\testimag\\"+ajouter+".jpg")) – méthode permettant de sauvegarder une image sous la forme d'un fichier. La commande écrase l'ancien fichier existant dans le répertoire.

MTI – vues des scénarios :

- scénario1



- scénario2



MTI - principales interfaces, classes et méthodes utilisées

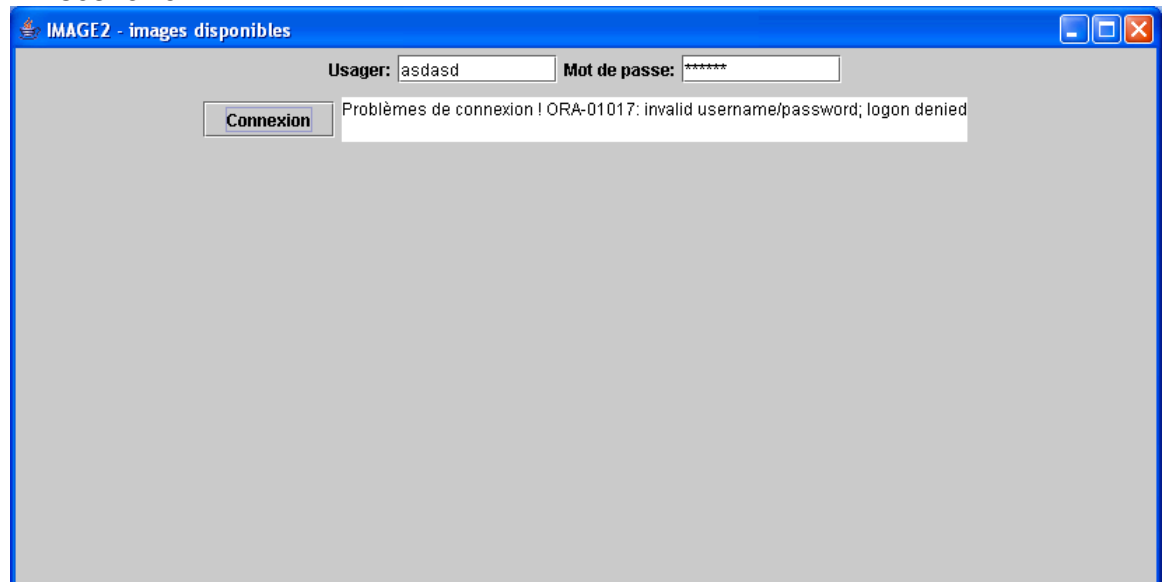
- **DriverManager.registerDriver(new oracle.jdbc.OracleDriver());**-
enregistrement du driver JDBC (méthode de la classe gestionnaire des pilotes)

- **con = DriverManager.getConnection** – établir une connexion à la BD sous la forme url, usager, mot de passe (url = jdbc :sous-protocole :bd)
- **stmt = con.createStatement()** – un objet Statement, qui a les méthodes nécessaires pour les requêtes (*statement* = requêtes simples, *preparestatement* = requêtes paramétrées, *callablestatement* = procédures stockées) . On a alors : *executeQuery* pour les requêtes qui retournent un *ResultSet*, *executeUpdate* pour celles qui retournent un int (nombre des touples traités), *execute* (procédures stockées)
- **pstmt = con.prepareStatement** = création d'un objet de type *PreparedStatement*, qui permet l'envoi des requêtes paramétrées vers la base de données.
- **pstmt.setString(1, nomImage[i])...** = permet de faire passer la valeur exprimée par le deuxième paramètre vers le premier. (ici, le paramètre spécifié par 1 aura la valeur de nomImage[i]).
- **getcustomdatum** – permet l'accès à l'objet image (c'est un objet de type proxy) (conformément au exemple donné sur le site d'Oracle : http://www.oracle.com/technology/products/intermedia/htdocs/intermedia_quickstart/intermedia_java_qs.html)
- **ordimage.setproperties** – permet la détection automatique des propriétés de l'image (dimensions, format, etc.)
- **OrdImageSignature** - c'est une classe utilisée pour représenter une instance d'un champ Oracle de type *ORDSYS.ORDImageSignature* dans une application Java. Cette classe inclut une méthode permettant de générer la signature d'une image donnée et des méthodes statiques pour comparer deux images. (deux signatures).
- **generateSignature()** – permet la création d'une signature pour une image donnée.

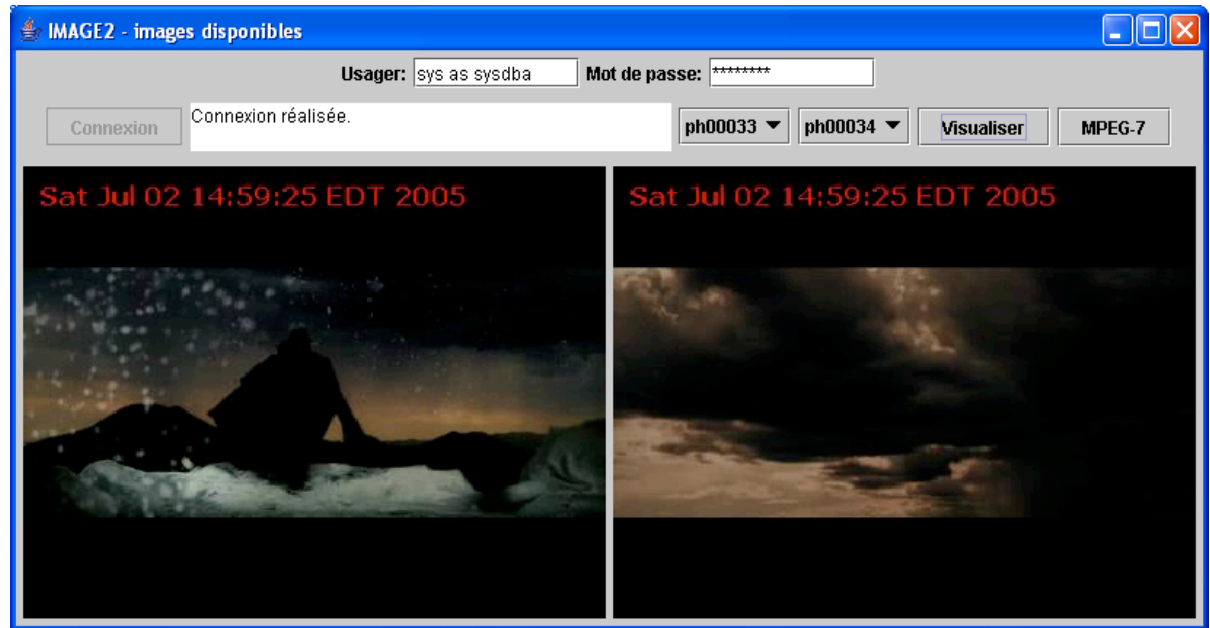
- **isSimilar()** - méthode qui compare les signatures des deux images (données comme paramètres). Elle retourne un entier spécifiant si le degré de différence entre les deux signatures se trouve ou non dans un écart donné. La méthode réalise la comparaison des deux signatures en utilisant des poids spécifiés pour un ou plusieurs attributs visuels (descripteurs de bas niveau). Le résultat de la comparaison est quantifié entre 0,0 et 100,0 ou une valeur plus petite indique une plus forte ressemblance. Si le résultat est plus petit ou égal au seuil fixé, les images sont considérées identiques et la méthode retourne la valeur 1, au cas contraire la valeur retournée étant 0.

MV : vues des scénarios :

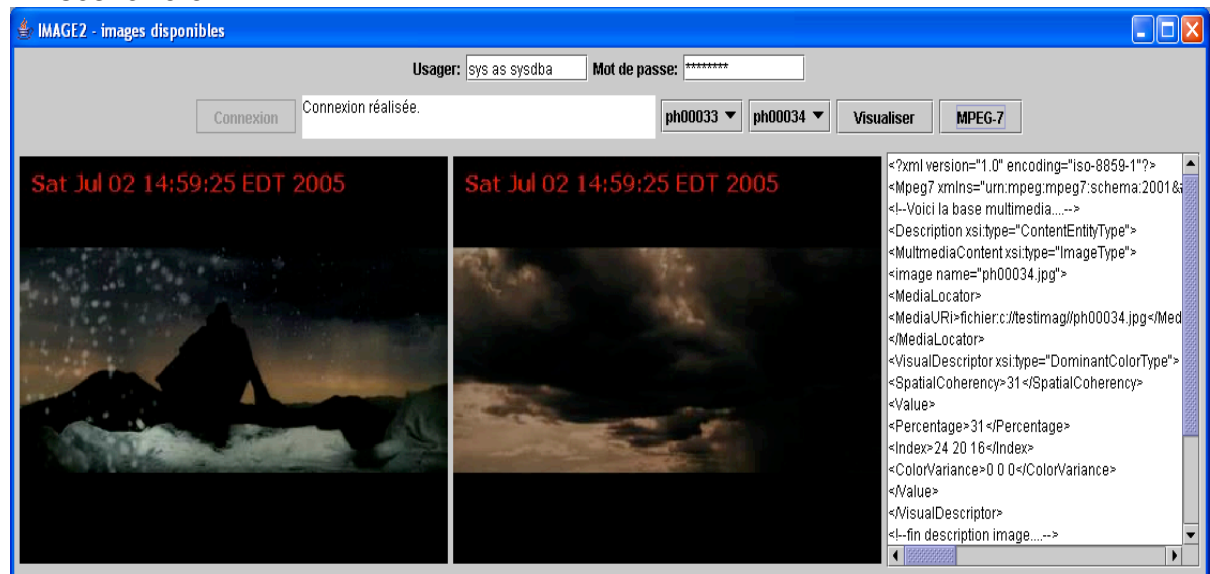
- scénario 1



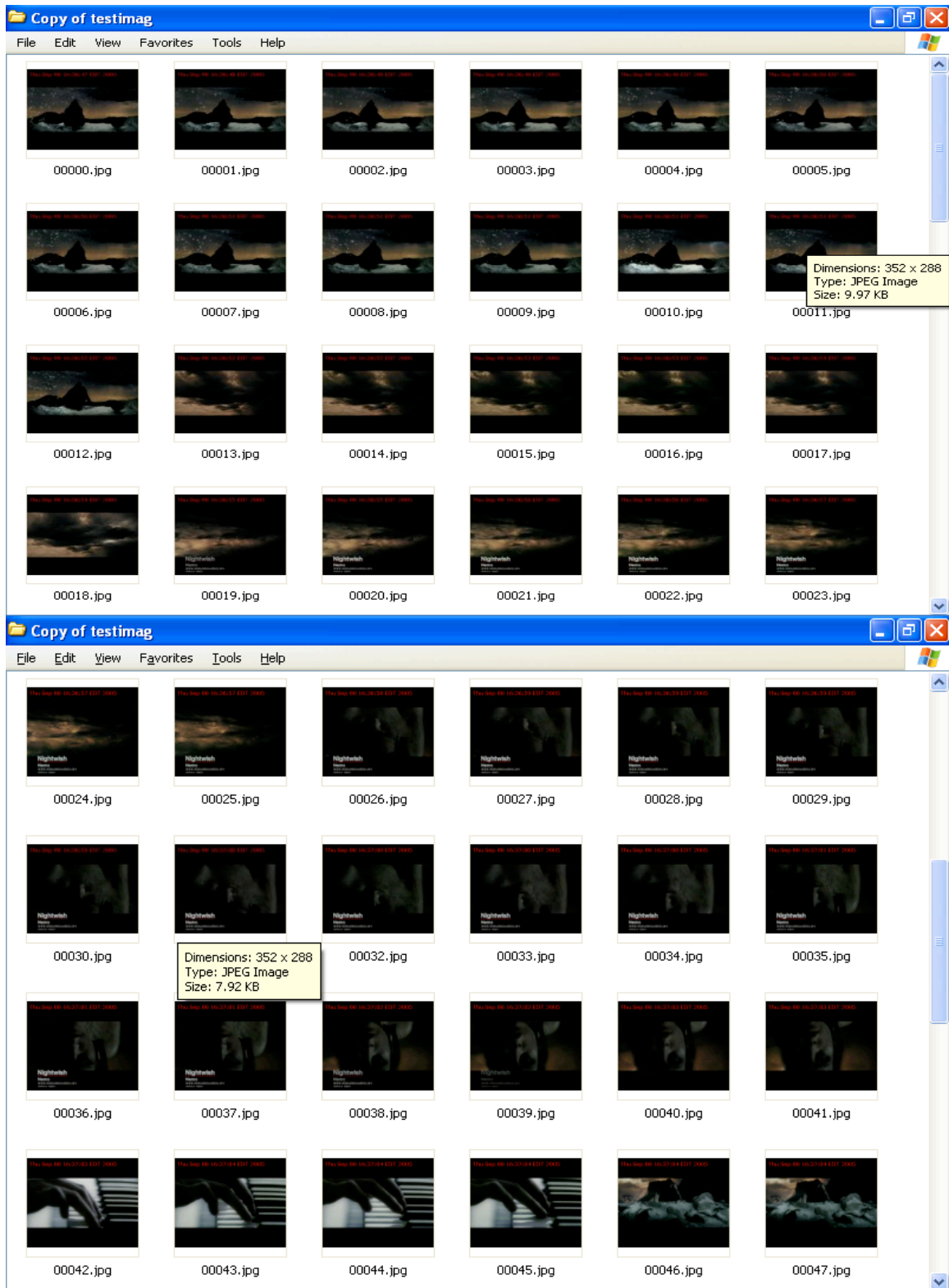
- scénario 2

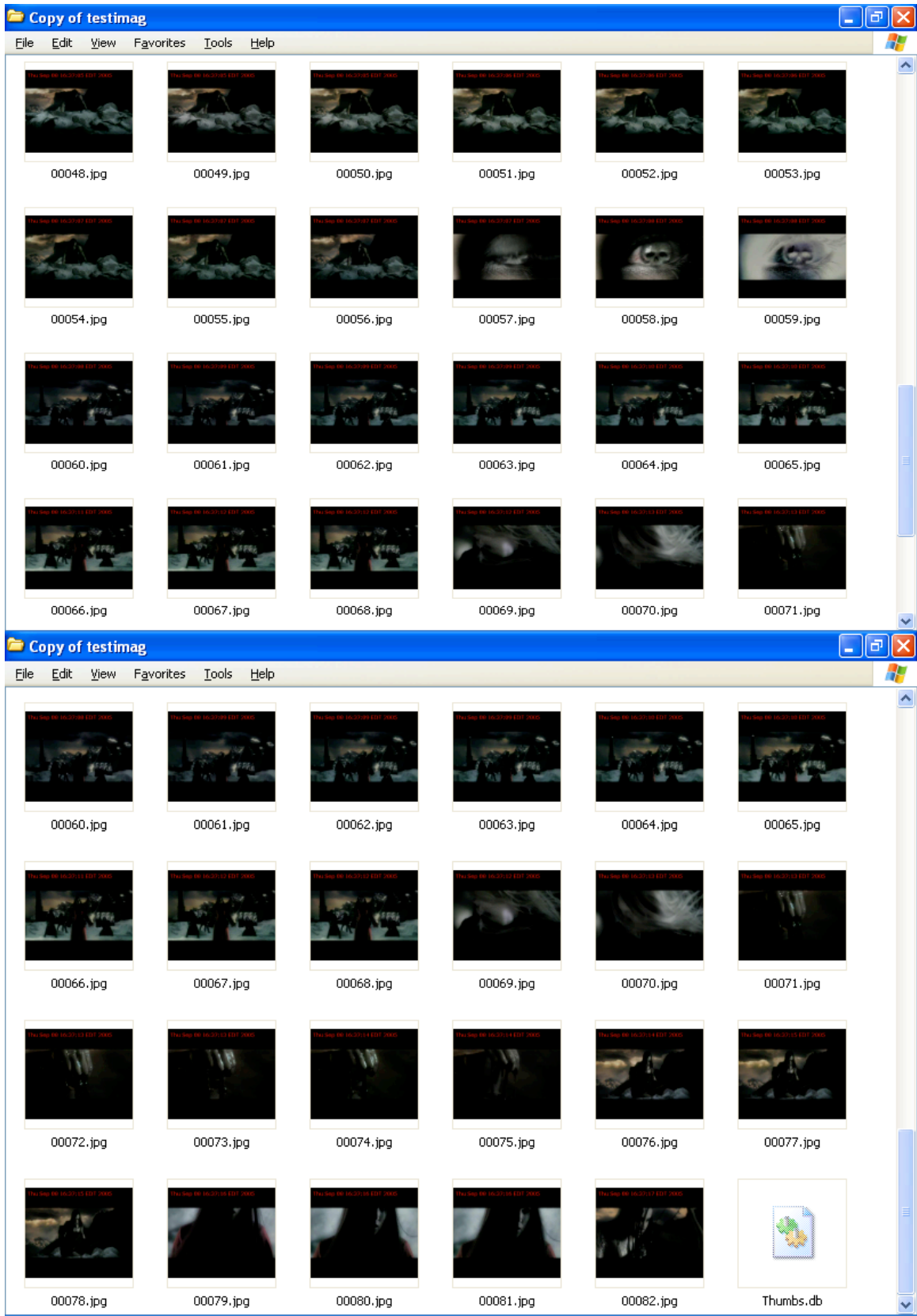


■ scénario 3



ANNEXE 8 - Images tirées de la vidéo





ANNEXE 9 – Résultats interrogations BD (Oracle SQL Plus)

Cas 1: 0,3-0,7-0,0-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00013
ph00042
ph00046
ph00059
ph00060
ph00069
ph00082

7 rows selected.

Cas 2: 0,1-0,9-0,0-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00042
ph00059
ph00060

3 rows selected.

Cas 3: 0,1-0,0-0,9-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00001
ph00007
ph00009
ph00010
ph00011
ph00023
ph00024
ph00026
ph00042
ph00043
ph00044
ph00045
ph00046
ph00049
ph00050
ph00052
ph00053
ph00054
ph00056
ph00057
ph00058
ph00059
ph00060
ph00062
ph00066

ph00069
ph00070
ph00071
ph00076
ph00078
ph00080
ph00082

32 rows selected.

Cas 4 : 0,1 – 0,0-0,0-0,9-10(pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00001
ph00002
ph00003
ph00005
ph00006
ph00007
ph00008
ph00009
ph00010
ph00011
ph00012
ph00013
ph00014
ph00015
ph00017
ph00018
ph00019
ph00023
ph00024
ph00025
ph00026
ph00027
ph00029
ph00031
ph00032
ph00033
ph00036
ph00037
ph00039
ph00040
ph00042
ph00043
ph00047
ph00048
ph00049
ph00050
ph00051
ph00052
ph00053
ph00054
ph00055
ph00056
ph00057
ph00059
ph00060
ph00061

ph00062
ph00063
ph00064
ph00065
ph00066
ph00067
ph00069
ph00070
ph00071
ph00074
ph00075
ph00076
ph00077
ph00079
ph00080
ph00081
ph00082

63 rows selected.

Cas 5 : 0,5-0,5-0,0-0,0-10(pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00046
ph00057
ph00059
ph00060
ph00062
ph00069
ph00071
ph00077
ph00078
ph00082

21 rows selected.

Cas 6 : 0,7-0,3-0,0-0,0-10; (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00031

ph00032
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00046
ph00049
ph00053
ph00057
ph00059
ph00060
ph00062
ph00066
ph00069
ph00071
ph00077
ph00078
ph00082

26 rows selected.

Cas 7 : 0,4-0,6-0,0-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00010
ph00013
ph00017
ph00019
ph00033
ph00034
ph00036
ph00037
ph00042
ph00046
ph00057
ph00059
ph00060
ph00062
ph00069
ph00071
ph00077
ph00082

18 rows selected.

Cas 8: 0,35-0,65-0,0-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00010
ph00013
ph00042
ph00046
ph00057
ph00059

ph00060
ph00062
ph00069
ph00082

10 rows selected.

Cas 9 : 0,35-0,25-0,4-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00009
ph00010
ph00011
ph00013
ph00017
ph00019
ph00023
ph00024
ph00033
ph00034
ph00037
ph00042
ph00043
ph00044
ph00045
ph00046
ph00049
ph00050
ph00052
ph00053
ph00057
ph00059
ph00060
ph00062
ph00066
ph00069
ph00070
ph00071
ph00076
ph00077
ph00078
ph00082

32 rows selected.

Cas 10 : 0,7-0,0-0,0-0,3-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00007
ph00009
ph00010
ph00011
ph00013
ph00015
ph00017
ph00019

ph00026
ph00031
ph00032
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00044
ph00046
ph00049
ph00050
ph00051
ph00052
ph00053
ph00054
ph00055
ph00056
ph00057
ph00059
ph00060
ph00062
ph00065
ph00066
ph00069
ph00070
ph00071
ph00074
ph00075
ph00076
ph00077
ph00078
ph00079
ph00082

43 rows selected.

Cas 11: 0,7- 0,0-0,3-0,0-10 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00009
ph00010
ph00011
ph00013
ph00017
ph00019
ph00023
ph00024
ph00026
ph00031
ph00032
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00044

ph00045
ph00046
ph00049
ph00050
ph00052
ph00053
ph00054
ph00057
ph00059
ph00060
ph00062
ph00066
ph00069
ph00070
ph00071
ph00076
ph00077
ph00078
ph00079
ph00082

38 rows selected.

Cas 12 : 0,7-0,0-0,3-0,0-15 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00044
ph00045
ph00046
ph00049
ph00053
ph00057
ph00059
ph00060
ph00062
ph00066
ph00069
ph00070
ph00071
ph00076
ph00077
ph00078
ph00082

28 rows selected.

Cas 13 : 0,7-0,0-0,3-0,0-20 (pour couleur, texture, forme, localisation)

```
SQL> select imnom from image2 where ind='s' order by imnom;
```

IMNOM

ph00009
ph00010
ph00013
ph00033
ph00034
ph00037
ph00042
ph00043
ph00044
ph00045
ph00046
ph00053
ph00057
ph00059
ph00060
ph00062
ph00069
ph00070
ph00071
ph00077
ph00078
ph00082

22 rows selected.

Cas 14 : 0,7-0,0-0,0-0,3-15 (pour couleur, texture, forme, localisation)

```
SQL> select imnom from image2 where ind='s' order by imnom;
```

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00031
ph00032
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00046
ph00053
ph00057
ph00059
ph00060
ph00062
ph00066
ph00069
ph00071
ph00077
ph00078
ph00082

25 rows selected.

Cas 15 : 0,7-0,0-0,0-0,3-20 (pour couleur, texture, forme, localisation)

```
SQL> select imnom from image2 where ind='s' order by imnom;
```

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00033
ph00036
ph00037
ph00042
ph00046
ph00057
ph00059
ph00060
ph00062
ph00069
ph00071
ph00077
ph00082

18 rows selected.

Cas 16 : 0,7-0,3-0,0-0,0-15 (pour couleur, texture, forme, localisation)

```
SQL> select imnom from image2 where ind='s' order by imnom;
```

IMNOM

ph00009
ph00010
ph00013
ph00017
ph00019
ph00033
ph00034
ph00036
ph00037
ph00042
ph00043
ph00046
ph00057
ph00059
ph00060
ph00062
ph00069
ph00071
ph00077
ph00078
ph00082

21 rows selected.

Cas 17 : 0,7-0,3-0,0-0,0-20 (pour couleur, texture, forme, localisation)

```
SQL> select imnom from image2 where ind='s' order by imnom;
```

IMNOM

```
-----  
ph00013  
ph00042  
ph00046  
ph00057  
ph00059  
ph00060  
ph00069  
ph00082
```

8 rows selected.

Cas 18 : 0,7-03-,0-0,0-19 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

```
IMNOM  
-----  
ph00013  
ph00033  
ph00034  
ph00036  
ph00037  
ph00042  
ph00046  
ph00057  
ph00059  
ph00060  
ph00062  
ph00069  
ph00082
```

13 rows selected.

Cas 19: 0,5-0,5-0,0-0,0-15 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

```
IMNOM  
-----  
ph00013  
ph00042  
ph00046  
ph00057  
ph00059  
ph00060  
ph00069  
ph00082
```

8 rows selected.

Cas 20 : 0,5-0,4-0,1-0,0-15 (pour couleur, texture, forme, localisation)

SQL> select imnom from image2 where ind='s' order by imnom;

```
IMNOM  
-----  
ph00009  
ph00010  
ph00013  
ph00042  
ph00046
```


ph00053
ph00057
ph00059
ph00060
ph00062
ph00069
ph00071
ph00082

13 rows selected.

BIBLIOGRAPHIE

- [1] Utz Westermann, Wolfgang Klaus. ***An analysis of XML database solutions for the management of MPEG-7 media descriptions***, ACM Computing Surveys Volume 35 , Pages: 331 – 373, 2003, ISSN:0360-0300, <http://doi.acm.org/10.1145/954339.954340>
- [2] Harald Kosch. ***MPEG-7 and multimedia database systems***, ACM SIGMOD Record , Volume 31 , Issue2 (June 2002) , pages: 34 – 39, 2002 , ISSN:0163-5808, <http://doi.acm.org/10.1145/565117.565123> .
- [3] Mario Doller, H. Kosch, B. Dorflinger, A. Bachlechner, G. Blaschke. ***Demonstration of an MPEG-7 multimedia data cartridge***, International Multimedia Conference, Proceedings of the tenth ACM international conference on Multimedia , Pages: 85 – 86, 2002 , ISBN:1-58113-620-X , <http://doi.acm.org/10.1145/641007.641024> .
- [4] A. Mostefaoi, F. Prêteux, V. Lecuire, J.M.Moureaux. ***Sirsale :un système d'indexation et de recherche de séquences audiovisuelles à large échelle***, Gestion des données multimédias, pages 283-306, Hermes, Paris, 2004, ISBN 2-7462-0824-5.
- [5] L.Gagnon, S. Foucher, V. Couaillier. ***ERIC7: an experimental tool for Content-Based Image encoding and Retrieval under the MPEG-7 standard***, ACM International Conference Proceeding Series , Proceedings of the winter international symposium on Information and communication technologies, Cancun, Mexico, Pages: 1 – 6, 2004, <http://tinyurl.com/9dr6y>.
- [6] Yi Haoran, Deepu Rajan, Chia Liang-Tien. ***Automatic generation of MPEG-7 compliant XML document for motion trajectory descriptor in sports video***, ACM International Workshop On Multimedia Databases, Proceedings of the 1st ACM international workshop on Multimedia databases, New Orleans, LA, USA, SESSION: Video analysis, retrieval, and summarizing, Pages: 10 – 17, 2003, ISBN:1-58113-726-5, <http://doi.acm.org/10.1145/951676.951680>.
- [7] Jobst Löffler, Konstantin Biatov, Christian Eckes, Joachim Kohler. ***IFINDER: an MPEG-7-based retrieval system for distributed multimedia content***, International Multimedia Conference, roceedings of the tenth ACM international conference on Multimedia, Juan-les-Pins, France, Pages: 431 – 435, 2002, ISBN:1-58113-620-X, <http://doi.acm.org/10.1145/641007.641102>.
- [8] M. L. Kherfi, D. Ziou, A. Bernardi. ***Image Retrieval from the World Wide Web: Issues, Techniques, and Systems***, ACM Computing Surveys (CSUR),

- Volume 36 , Issue 1 (March 2004), Pages: 35 – 67, 2004, ISSN:0360-0300,
<http://doi.acm.org/10.1145/1013208.1013210>.
- [9] Andrew Graves, Mounia Lamas. **Video retrieval using an MPEG-7 based inference network**, Annual ACM Conference on Research and Development in Information Retrieval, Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, Tampere, Finland , Pages: 339 – 346, 2002, ISBN:1-58113-561-0,
<http://doi.acm.org/10.1145/564376.564436>.
 - [10] Wen Wen Hsieh, Arbee L.P. Chen. **Constructing a bowling information system with video content analysis**, ACM International Workshop On Multimedia Databases, Proceedings of the 1st ACM international workshop on Multimedia databases, New Orleans, LA, USA, Pages: 18 – 24, 2003, ISBN:1-58113-726-5, <http://doi.acm.org/10.1145/951676.951681>.
 - [11] G. Gaughan, A. F. Smeaton, C. Gurrin, H. Lee, K. McDonald. **Design, implementation and testing of an interactive video retrieval system**, International Multimedia Conference, Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, Berkeley, California, Pages: 23 – 30, 2003, ISBN:1-58113-778-8,
<http://doi.acm.org/10.1145/973264.973269>.
 - [12] R. Goularte, R. G. Cattelan, J. A. Camacho-Guerrero, V. R. Inacio Jr., M. de Graca C. Pimentel. **Interactive multimedia annotations: enriching and extending content**, Document Engineering, Proceedings of the 2004 ACM symposium on Document engineering, Milwaukee, Wisconsin, USA, Pages: 84 – 86, 2004, ISBN:1-58113-938-1, <http://doi.acm.org/10.1145/1030397.1030414>.
 - [13] Kuo, P.-J., Aoki, T., Yasuda, H., Yasuda-Aoki Lab., Tokyo Univ., Japan, **MPEG-7 based dozen dimensional digital content architecture for semantic image retrieval services**, *This paper appears in: e-Technology, e-Commerce and e-Service*, 2004. IEEE '04. 2004 IEEE International Conference 28-31 March 2004, On page(s): 517 – 524
Inspec Accession Number: 8107375, <http://tinyurl.com/3jzmo> .
 - [14] Jae-Ho Lee, Hyoung-Joon Kim, Whoi-Yul Kim, Image Eng. Lab., Hanyang Univ., Seoul, Korea , **Video/image retrieval system based on MPEG-7 (VIRS)**, *his paper appears in: Information Technology: Research and Education*, 2003. Proceedings. ITRE2003. International Conference 11-13 Aug. 2003, On page(s): 79 – 83, Inspec Accession Number: 8030313, <http://tinyurl.com/6fdyi> .
 - [15] Kosch, Harald, **Distributed multimedia database technologies supported by MPEG-7 and MPEG-21**, 2004, CRC press LLC, USA, ISBN 0-8493-1854-8.
 - [16] Dunckley, Lynne, **Multimedia Databases. An object-relational approach**, Pearson Education Ltd 2003, Great Britain, ISBN 0-201-78899-3.
 - [17] page Web sur le site de Sun : <http://java.sun.com/xml/>
 - [18] Gardarin Georges, **XML**, 2002, Dunod, Paris, 2002, ISBN 2100069330

- [19] Oracle interMedia:
<http://www.oracle.com/technology/products/intermedia/index.html>
- [20] Oracle *interMedia* Image Quick Start :
http://www.oracle.com/technology/products/intermedia/htdocs/intermedia_quickstart/intermedia_java_qs.html
- [21] INTERNATIONAL STANDARD ISO/IEC FDIS 15938-6, Information technology — Multimedia content description interface — Part 6:Reference software,
http://www.ncits.org/ref-docs/FDIS_15938-6.PDF
- [22] Lévesque Ghislain, **Analyse de système orientée – objet et génie logiciel. Concept, méthodes et applications**, Chenelière/McGraw-Hill, 1998, ISBN 2-89461-161-7.
- [23] B.Ionescu, D.Coquin, P.Lambert, V.Buzuloiu, **Analyse et caractérisation de séquences de films d'animation**, présentation à ORASIS2005 - 9ème Congrès Jeunes chercheurs en Vision par Ordinateurs, Mai 2005,-
<http://orasis2005.univ-bpclermont.fr/user/www/orasis/papiers/042.pdf> -
- [24] Philippe Mulhem, Jérôme Gensel, Hervé Martin, **Modèles pour résumés adaptatifs de vidéos** - Bases de données et multimédia (Ingénierie des systèmes d'information RSTI série ISI-NIS Vol.7 N° 5-6/2002), pages 91-118, ISBN : 2-7462-0684-6
- [25] L.Chen, Y. Chahir, **Indexation de la vidéo numérique**, Gestion des données multimédias, pages 306-334, Hermes, Paris, 2004, ISBN 2-7462-0824-5.
- [26] Alain April, Cours de **Bases de données multimédia(GTI440)**, Département de génie logiciel et des TI, ÉTS, Université de Québec, session été 2005.
- [27] Clément Arsenault, Cours de **Recherche d'information avancée(BLT6322)**, École de bibliothéconomie et des sciences de l'information, Université de Montréal, session hiver 2005.