

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA

FACULTE DE SCIENCES EXACTES

DEPARTEMENT D'INFORMATIQUE

Mémoire de projet de fin d'études  
En vu d'obtention du diplôme  
d'ingénieur d'état en informatique

Option : Système d'Information

Thème

CONCEPTION ET REALISATION D'UNE  
INTERFACE D'ANNOTATION POUR UN  
SYSTEME D'INDEXATION ET DE RECHERCHE  
DE SEQUENCES AUDIOVISUELLES A LARGE  
ECHELLE.

Proposé par :

M.MOSTFAOUI Ahmed

Encadré par :

M.HADJYAHIA Ouahid

M.NEHALE Djilali

Jugé par :

Présidente des jurés:

M<sup>lle</sup> .Mokhtari

Les jurés :

M .Ould Aissa

Mme .Benstiti

présenté par.:

M HADJ HACENE Mohamed

M BELHDJER Othmen



Année Universitaire 2003-2004

16007-26-1

---

---

## Remerciement

---

---

Nous tenant à remercier notre dieu «ALLAH» tout puissant qui nous a donné la puissance et la volonté pour réaliser ce modeste travail, nos parents qui ont véliés sur nous pour que nous puissions arriver à ce jour là, Pour leurs sacrifices.

Nous remercions notre promoteur HadjYahia Ouahid et le co-promoteur Nehal Djilali pour leurs efforts, leurs patiences, leurs encadrements, et leurs soutiens.

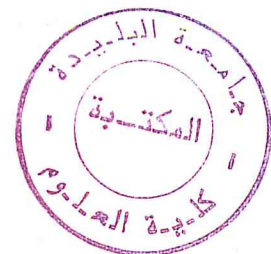
Nous remercions les membres du jury pour nous avoir fait l'honneur de juger notre travail.

Sans oublier M<sup>me</sup> Boumahdi et M<sup>lle</sup> Farah pour leurs précieuses aide et leurs conseils.

Nous adressons nos sincères remerciements à M.KARABERNOU HASAN, M.ABD\_ELLAH ELHADJ HICHAM, M.TOUMI REDHOUANE, M.FOUZI, M.ABD ELHAMIDE, M.HAMZA, M.MOHRÉ MADANI, M DJELLAL, M.ABASSE TURKIE, M.BALA, M.DJAMEL pour leurs précieuses aides.

Nous adressons nos sincères remerciements à tous ce qui nous a aidé de prêt ou de loin pour réaliser ce travail.

Nous remercions tous les enseignants de la faculté des sciences de BLIDA et surtout nos enseignants du département informatique à leur tête M<sup>me</sup> BENSTITI.





---

---

## Dédicace

---

---

*Ce mémoire est dédié :*

*À ma mère qui a sacrifié pour moi avec tout précieux et vital, qui ma toujours pousser vers l'avant, qui a été présente ou les autres m'on oublie.*

*À mon très cher père qui ma appris le sens de la responsabilité dans les situations pénibles, qui m'a enseigné l'honnête, la patience, la volonté,.... Que dieu «ALLAH» tous puissant vous protège !*

*Sans eux je ne peux arriver jusqu'a-là.*

*À mes fidèles frères, et ma précieuse sœur et ses filles.*

*À ma grande-mère et ma tante sans oublier mes tontons abd elkrime et benyoucef. Pour tous les BELHDJER et tous les FASSI.*

*À la population de ma ville natale : MÉDÉA.*

*À mon binôme Mohamed pour sa grande patience avec moi, et pour toute sa famille HADJHCENE.*

*À mes compagnons Tassis, Stambouli, redhouane, . redha, m'hamed, hicham, merouane, hamza, Billale, mechtie, melzi, ABD ELAH.*

*Pour tous mes amis de l'université de Blida et de Médéa qui ont passé avec moi les bonnes et les mauvais temps. À tous mes biens aimés.*

Othmen.



---

---

## Dédicace

---

---

*Ce mémoire est dédié :*

*A mes très chers parents pour leurs soutiens durant toute ma carrière, Pour leurs bienveillances, leurs efforts constants dans mes études, et Pour leurs encouragements,*

*A mon frère HAMZA, et mes sœurs surtout la petite Samira.*

*A tout mes nièces, et en particulier la vedette Nessrine.*

*A toutes la familles HADJHACENE et BENGHERBIA,*

*A mon binôme Othmen et à toute sa famille*

*A Lies, Ahmed, Tassis, Redhouane, Redha, Fateh, M'Hamed, Sofiane,*

*Hamza, Fouzi, Hichem, Abd\_ellah elhadj,*

*A tous mes amis.*

MOHAMED.





# Sommaire

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre 1: le projet SIRSALE</b>	
<b>1.1 Introduction :.....</b>	<b>4</b>
<b>1.2 Architecture de SIRSALE :.....</b>	<b>4</b>
1.2.1 Le composant serveur :.....	6
1.2.2 L'interface d'indexation :.....	5
1.2.3 L'interface de définition de thème :.....	5
1.2.4 L'interface d'interrogation et de navigation :.....	6
<b>1.3 Le fonctionnement de SIRSALE :.....</b>	<b>6</b>
<b>1.4 Le modèle de données existant :.....</b>	<b>6</b>
<b>1.5 Serveurs :.....</b>	<b>8</b>
1.5.1 Serveur vidéo :.....	8
1.5.2 Serveur de Méta-Données :.....	8
1.5.3 Serveur de thème ou serveur de modules:.....	9
<b>1.6 Les interfaces :.....</b>	<b>9</b>
1.6.1 Interface de définition de thèmes:.....	9
1.6.2 Interface d'annotation :.....	9
1.6.3 Interface d'interrogation :.....	11
<b>1.7 Conclusion :.....</b>	<b>12</b>
<b>Chapitre:II Gestion multimédia</b>	
<b>2.1 Introduction :.....</b>	<b>14</b>
<b>2.2 Caractéristiques d'un document multimédia :.....</b>	<b>14</b>
<b>2.3 Édition d'un document multimédia :.....</b>	<b>15</b>
<b>2.4 Évolution des données :.....</b>	<b>16</b>
<b>2.5 Modèle de document multimédia :.....</b>	<b>16</b>
2.5.1 Le modèle des objets de base :.....	16
2.5.2 Modèle de composition :.....	17
<b>2.6 Environnement auteur :.....</b>	<b>18</b>
2.6.1 Exemple de construction d'un environnement : la boîte à outils Kaomi :.....	18
<b>2.7 La réutilisation des unités documentaires :.....</b>	<b>19</b>
<b>2.8 Présentation d'un document multimédia:.....</b>	<b>20</b>
2.8.1 Notion de synchronisation multimédia :.....	20
2.8.1.1 Synchronisation intra-objet :.....	20
2.8.1.2 Synchronisation inter-objet :.....	20
2.8.1.3 Synchronisation des lèvres :.....	21
2.8.2 Support de présentation :.....	21
2.8.2.1 Caractéristiques et besoins de présentation multimédia :.....	21
2.8.3 Architecture générale :.....	22
2.8.3.1 Ordonnancement d'une présentation multimédia:.....	23
2.8.3.2 Contrôleur de l'exécution d'une présentation multimédia:.....	26
<b>2.9 Conclusion :.....</b>	<b>27</b>
<b>Chapitre:III Indexation audiovisuelle</b>	
<b>3.1 Introduction :.....</b>	<b>30</b>
<b>3.2 Les problèmes spécifiques de l'audiovisuel :.....</b>	<b>31</b>

<b>3.3 Le concept d'objectif :</b> .....	32
3.3.1 Les propriétés du concept objectif :.....	32
3.4 le concept interprétatif:.....	32
3.4.1 Les propriétés du concept interprétatif:.....	33
<b>3.5 SIRSALE en vue du concept objectif et le concept interprétatif:</b> .....	33
3.5.1 L'approche structurale dans le cadre de SIRSALE :.....	33
3.5.2 L'approche par stratification dans le cadre de SIRSALE :.....	35
3.5.3 Approche proposée par SIRSALE (approche multi-niveaux):.....	36
<b>3.6 Conclusion:</b> .....	36

## Chapitre :IV Le concept d'UML

<b>4.1 Introduction:</b> .....	38
4.1.1 Historique D'UML:.....	38
4.1.2 Approche Objet:.....	39
4.1.3 Pourquoi une méthodologie Objet :.....	39
4.1.4 Les concepts fondamentaux de l'approche orienté objet :.....	39
4.1.5 Quelque notion de modélisation :.....	41
4.1.6 Avantages de l'UML :.....	42
4.1.7 Les concepts :.....	43
<b>4.2 Les diagrammes d'UML :</b> .....	46
4.2.1 Les diagrammes de vues statiques :.....	46
4.2.1.1 Diagramme de cas d'utilisation:.....	46
4.2.1.2 Les diagrammes d'objets :.....	49
4.2.1.3 Les diagrammes de classe :.....	49
4.2.1.3 Diagramme de composants :.....	50
4.2.1.4 Diagramme de déploiement :.....	50
4.2.2 Les diagrammes de vues dynamiques :.....	50
4.2.2.1 Diagramme de séquence :.....	51
4.2.2.2 Diagramme de collaboration :.....	52
4.2.2.3 Les diagrammes d'états transitions :.....	53
4.2.2.4 Diagramme des activités :.....	54
<b>4.3 Les points forts d'UML :</b> .....	54
<b>4.4 Conclusion :</b> .....	55

## Chapitre :V Démarche de développement

<b>5.1 Introduction :</b> .....	57
<b>5.2 Choix de la démarche :</b> .....	57
<b>5.3 Spécification des besoins :</b> .....	58
5.3.1 Cas d'utilisation :.....	58
5.3.2 Les cas d'utilisation du system :.....	59
5.3.2.1 Les cas d'utilisation principaux :.....	59
5.3.2.1.1 Cas d'utilisation« identification du Journal Télévisé » :.....	60
5.3.2.1.2 Cas d'utilisation « Annotation » :.....	61
5.3.2.1.3 Cas d'utilisation « Interrogation » :.....	64
5.3.3 Les diagrammes de séquence :.....	70
5.3.3.1 Diagramme de séquence : Identification du Journal Télévisé :.....	70
5.3.3.2 Diagramme de séquence de l'Annotation :.....	71
5.3.3.1 Diagramme de séquence de l'interrogation :.....	71
5.3.4 Les diagrammes de collaboration :.....	77
5.3.5 Diagramme de collaboration :.....	81



5.3.6 Diagrammes d'activités :	82
5.3.7 Diagrammes d'états-transitions :	83
<b>5.4 Conception :</b>	<b>84</b>
5.4.1 Conception globale :	85
5.4.2 Conception détaillée :	87
5.4.2.1 Le Module « Identification » :	87
5.4.2.2 Le Module « Annotation » :	87
5.4.2.3 Le Module « Interrogation »:	88

## **Chapitre:VI Implémentation**

<b>6.1 introduction :</b>	<b>93</b>
<b>6.2 Choix du langage de programmation:</b>	<b>93</b>
<b>6.3 Implémentation du serveur Méta_Données :</b>	<b>93</b>
<b>6.4 Implémentation du Composant Application :</b>	<b>95</b>
6.4.1 L'interface de visualisation :	96
6.4.2 Scénario de l'implémentation :	97
<b>6.5 Modélisation :</b>	<b>99</b>
<b>6.6 Réalisation des cas d'utilisation :</b>	<b>99</b>
6.6.1 Validation du Cas d'utilisation « identification du Journal Télévisé »:	100
6.6.2 Validation du cas d'utilisation « Annotation » :	100
6.6.3 Validation du cas d'utilisation « Interrogation » :	101
<b>6.7 Test :</b>	<b>102</b>
6.7.1 Introduction :	102
6.7.2 Démarche du test :	102
6.7.3 Conclusion :	104
<b>Conclusion générale :</b>	<b>105</b>
<b>Bibliographie :</b>	<b>107</b>
Annexe A :	109
Annexe B :	113
Annexe C :	117
Annexe D :	123
Annexe E :	128

## Liste des figures :

Fig1.1 Architecture générale du SIRSALE .....	5
Fig1.2 Modèle de données de SIRSALE.....	7
Fig1.3. Interface d'annotation du SIRSALE.....	10
Fig1.4 Interface d'interrogation du SIRSALE.....	11
Fig2.1 Modèle des objets de base de document multimédia.....	17
Fig2.2 Modèle de composition de document multimédia.....	17
Fig2.3 Boîte à outils Kaomi.....	19
Fig2.4 Prise en compte de la réutilisation des unités documentaires .....	19
Fig2.5 Synchronisation des lèvres.....	21
Fig2.6 Architecture générale d'une machine de présentation multimédia.....	22
Fig2.7 Structure de l'Ordonnanceur.....	24
Fig2.7.1 Graphe de transitions d'objet média.....	25
Fig2.8 Contrôleur de l'exécution .....	26
Fig3.1 Approche structurelle.....	34
Fig3.2 Approche par stratification.....	35
Fig3.3 Vue multi-niveaux d'un document vidéo.....	36
Fig. 4.1 Représentation d'une note.....	43
Fig.4.2 Représentation graphique d'un paquetage contenant deux paquetages.....	44
Fig.4.3 Représentation graphique d'un sous système.....	44
Fig. 4.4 Une ligne entre deux classes représente une association.....	44
Fig. 4.5 Représentation d'une agrégation.....	45
Fig. 4.6 Représentation graphique de la composition.....	45
Fig. 4.7 Représentation de la relation généralisation simple entre classes.....	45
Fig. 4.8 Exemple de généralisation multiple.....	46
Fig. 4.9. Représentation graphique d'un acteur.....	46
Fig. 4.10 présentation d'un scénario.....	47
Fig.4.11 Représentation des cas d'utilisation.....	47
Fig.4.12 Représentation d'une relation de généralisation.....	48
Fig.4.13 représentation de la relation d'inclusion.....	48
Fig 4.14 Représentation de la relation d'extension.....	48
Fig.4.15 Exemple de diagramme d'objets.....	49
Fig.4.16 Exemple d'un diagramme de classes.....	49
Fig 4.17 Exemple d'un diagramme de composants.....	50
Fig.4.18 Agencement de messages.....	51
Fig.4.19 Les différents types de messages.....	52
Fig.4.20 Activation d'un objet de manière simple .....	52
Fig 4.21 Formalisme de base du diagramme de collaboration.....	52
Fig.4.22 Exemple de diagramme d'état-transition.....	53
Fig.4.23 Les points d'exécution pour un état.....	54
Fig.4.24 Diagramme d'activités, ouvrir porte –ascenseur.....	54
Fig. 5.1. Le modèle de développement en cascade .....	58
fig.5.2 Les cas d'utilisation principaux du système.....	59
Fig.5.3. Cas d'utilisation « identification du Journal Télévisé ».....	60
Fig.5.4.Cas d'utilisation « identification de l'animateur ».....	60
Fig.5.5 cas d'utilisation « Identification du invités ».....	60
Fig.5.6 Cas d'utilisation« Identification de type du Journal Télévisé» .....	61
Fig.5.7 Cas d'utilisation« Identification d'Identification de la Date ».....	61
Fig.5.8 Cas d'utilisation « Annotation ».....	61



Fig.5.9 Cas d'utilisation « Visualisation ».....	62
Fig.5.10 Cas d'utilisation « limitation des séquences ».....	62
Fig. 5.11 Cas d'utilisation « Description » .....	63
Fig.5.12 Cas d'utilisation « Validation du choix ».....	64
Fig.5.13 Cas d'utilisation « Interrogation » .....	64
Fig.5.14 Cas d'utilisation « Interrogation par Evènement » .....	65
Fig. 5.15 Cas d'utilisation « Interrogation par lieu ».....	65
Fig.5.16 Cas d'utilisation « Interrogation par Acteur ».....	66
Fig.5.17 Cas d'utilisation « Interrogation par mot clé ».....	66
Fig.5.18 Cas d'utilisation « Interrogation par date ».....	66
Fig. 5.20 Cas d'utilisation« Interrogation par Journal Télévisé ».....	67
Fig. 5.21 Cas d'utilisation« Interrogation par personnalité ».....	67
Fig. 5.22 Cas d'utilisation « Interrogation par animateur ».....	67
Fig.5.23 Cas d'utilisation « Interrogation par Date ».....	68
Fig.5.24 Cas d'utilisation « Interrogation par Nom-JT.».....	68
Fig5.25 Diagramme de cas d'utilisation de l'Indexation Audiovisuelle (Journal Télévisé)....	69
Fig.5.26 diagramme de séquence de l'identification du journal télévisé .....	70
Fig.5.27 Digramme de séquence de l'annotation .....	71
Fig.5.28 Diagramme de séquence interrogation par Lieu .....	72
Fig.5.29 diagramme de séquence interrogation par Acteur.....	73
Fig.5.30 diagramme de séquence Mot Clé.....	73
Fig.5.31 diagramme de séquence d'interrogation par date d'évènement.....	74
Fig.5.32 diagramme de séquence interrogation par type du journal télévisé .....	75
Fig.5.33 diagramme de séquence interrogation par date du journal télévisé.....	76
Fig.5.34 diagramme de séquence interrogation par Invités.....	76
Fig.5.35 diagramme de séquence interrogation par animateur.....	77
Fig. 5.36 diagramme de collaboration d'une demande de visualisation.....	78
Fig.5.37 Ébauche du diagramme de classe.....	78
Fig.5.38 diagramme de collaboration d'identification du journal télévisé.....	78
Fig.5.39 Ébauche du diagramme de classe.....	79
Fig.5.40 diagramme de collaboration de l'annotation.....	80
Fig. 5.41 Ébauche du diagramme de classe.....	80
Fig.5.42 diagramme de collaboration de l'interrogation.....	81
Fig. 5.43 diagramme de collaboration « Annotation ».....	81
Fig.5.44 diagramme de collaboration « Interrogation ».....	82
Fig.5.45 diagramme d'activité de l'annotation .....	82
Fig.5.46 diagramme d'activité de l'interrogation .....	83
Fig.5.47 diagramme d'états transition.....	84
Fig.5.48. Schéma d'IJT(Indexation des Journaux Télévisé ).....	85
Fig.5.49 Diagramme de Composant des différents modules de système.....	86
Fig.5.50 les éléments d'identification.....	87
Fig.5.51 les éléments d'annotation.....	87
Fig.5.52 les éléments d'interrogation.....	88
Fig.5.53 Diagramme de classe de l'IJT.....	90
Fig.6.1 un survol de la structure de l'application.....	93
Fig.6.2 Code présente une connexion vers une BD_MySQL.....	94
Fig.6.3 Modèle de données de IJT.....	95
Fig.6.4 le lecteur vidéo.....	96
Fig.6.5 Spécification de la ressource multimédia.....	96
Fig.6.6 Spécification du player.....	96

Fig.6.7 Processus d'identification.....	97
Fig.6.8 Processus d'annotation.....	98
Fig.6.9 interrogation par journal télévisé.....	98
Fig.6.10 interrogation par contenu sémantique d'événement.....	99
Fig.6.16 validation du cas d'utilisation identification du journal télévisé.....	100
Fig.6.17 validation du cas d'utilisation « Annotation ».....	100
Fig.6.18 validation du cas d'utilisation interrogation par journal télévisé.....	101
Fig6.19 Identification du journal du canal MADJD.....	102
Fig6. 20annotation du premier événement.....	103
Fig6.21 interrogation par journal télévisé.....	104
Fig6.22 interrogation par événement.....	104
FigC.1Pile matériel et logicielle d'un programme JMF.....	118
Fig C.2Modèle du Player JMF.....	118
Fig C.3 Représentation UML du player.....	119
FigC.4 Etats d'un Player.....	120
FigC.5 Architecture à niveau élevé de JMF RTP.....	121
Fig C.6Transmission du flux RTP en JMF.....	121
Fig C.7 Réception du flux RTP en JMF.....	122
FigE.1 Les couches RMI.....	128
FigE.2 architecture de RMI.....	129



# *Introduction générale :*

La manipulation et la gestion de données multimédias en général et audiovisuelle en particulier a suscité ces dernières années un engouement sans précédent de la part du grand public. Les applications multimédias ont investi un spectre applicatif particulièrement large allant de l'imagerie médicale aux systèmes d'information géographiques et aux applications grand public comme les jeux vidéo et les boutiques virtuelles. Cet intérêt pour le multimédia résulte d'une part de l'avancée remarquable qu'ont enregistrée les nouveaux supports de communication et d'autre part, de l'augmentation spectaculaire de la puissance des ordinateurs (puissance de calcul, capacité de stockage, etc.) permettant aujourd'hui un traitement efficace des documents multimédias.

Néanmoins la majorité des outils de traitement du vidéo ne profitent pas de l'énorme potentiel de la vidéo numérique et la considèrent comme une simple digitalisation des anciens supports analogiques (cassettes VHS, etc.), malgré l'apparition de nouveaux matériels comme le CD et le DVD, qui permettent d'accéder directement à une scène (ce qui offre une panoplie de fonctionnalités). Une première possibilité est de pouvoir accéder à une séquence particulière sans avoir à visualiser toute la vidéo, si la recherche se fait dans plusieurs vidéo, le problème devient insupportable. Diverses solutions de recherche d'information audiovisuelle ont été proposées, par mis lesquelles, un projet est en cours de développement qui s'appelle «SIRSALE» : un système d'indexation et de recherche de séquence audiovisuelle à large échelle, avec la collaboration de l'université de Franche-Comté à Montbéliard –LIFC et les universités LIRISA,INSA de Lyon. Notre projet s'intègre dans le cadre du projet SIRSALE.

Les créateurs du projet « SIRSALE » attendent de nous une conception d'un outil qui permet l'annotation des documents audiovisuels avec certaines exigences, comme le domaine d'utilisation, le langage de programmation, etc. afin que cet outil puisse être intégré dans le projet « SIRSALE ».

La recherche d'information audiovisuelle devient aujourd'hui une nécessité absolue pour un nombre croissant d'utilisateurs qui gèrent des informations audio/vidéo diverses dans leurs activités. Beaucoup de travaux de recherche, émanant d'institutions académiques et industriels, ont focalisé leurs efforts plutôt sur les aspects technologiques comme l'analyse et l'extraction automatique de l'information audiovisuelle (vue structurelle) et il existe très peu de travaux qui tiennent compte des vrais besoins des utilisateurs (vue sémantiques).

Cependant les besoins des utilisateurs varient selon le contexte d'utilisation. Ainsi, certains professionnels souhaiteront retrouver une séquence précise parmi un grand nombre de vidéo, en revanche un amateur préférera parcourir une vidéo pour trouver une scène qui l'intéresse.

**Objectif :**

L'objectif de notre projet est de concevoir et réaliser une interface qui permet l'annotation par contenu sémantique des séquences vidéo d'un centre d'intérêt (les journaux télévisés) pour un système d'indexation et de recherche des séquences audiovisuelles à large échelle.

Ce travail est partagé sur deux parties qui sont :

1. Conception et réalisation d'un lecteur vidéo avancé.
2. Conception et réalisation d'une interface d'annotation et d'interrogation via une connexion à une base de données.

Pour arriver aux objectifs de notre projet le mémoire est organisé comme suit :

- ❖ Chapitre 1 : définit une étude du projet SIRSALE (System d'Indexation et de Recherche des Séquences Audiovisuelles à Large Échelle), concernant son architecture, ses composants, son fonctionnement, ...etc.
- ❖ Chapitre 2 : définit le domaine du multimédia, en détaillant les principes du multimédia avec des exemples extraits du monde réelle.
- ❖ Chapitre 3 : définit l'indexation audiovisuelle, explique la notion d'indexation avec la présentation de ses divers types notamment la notion d'indexation audiovisuelle au sein du projet SIRSALE.
- ❖ Chapitre 4 : définit le concept UML, puisque nous avons utilisé la notation UML (*Unified Methode Language*) ce chapitre est consacré à clarifier les schémas ou notations déployées dans cette étude.
- ❖ Chapitre 5 : décrit une étude détaillée de la démarche du développement de notre outil à l'aide du langage UML, en suivant le modèle en cascade comme un cycle de vie.
- ❖ Chapitre 6 : décrit l'implémentation : comme n'importe quelle étude, elle s'achève avec une partie d'implémentation qui donne un rapport sur la réalisation des solutions conçues pendant le chapitre précédant, et la vérification de la réalisation avec le test de l'outil sur la fiabilité et l'intégrité du déroulement.

# chapitre:1

## projet SIRSALE



## **1.1 Introduction :**

SIRSALE est un projet a été lancé en 2002, ce projet consiste à la conception d'un système d'Indexation et de recherche des séquences audiovisuelles à large Échelle.

SIRSALE se charge de la gestion et la manipulation de grand volume de données audiovisuelles dans un environnement distribué, il est fondé sur le concept de réseau, fournissant les outils suivants :

a) Un outil d'indexation selon la sémantique audiovisuelle :

Dans cet outil, on propose des techniques d'indexation semi\_ automatique, basant sur la nature des images qui vont être proposées à l'annotateur, afin que n'importe quel utilisateur quelles que soient ses connaissances en informatique (journaliste, informaticien,...) peut annoter et représenter la sémantique de ses documents audiovisuels d'une façon easy, relatif à un thème spécifié;

b) Un outil de manipulation des documents audiovisuels :

L'objectif de cet outil est de permettre à un utilisateur d'exploiter les documents audiovisuels, en exécutant en premier temps des requêtes sémantiques par document, ou de naviguer structurellement dans les différents documents vidéo [SICH02].

c) Un outil de création de thème :

Cet outil permet aux utilisateurs de créer, de manière simple et intuitive via des interfaces graphiques dédiées, des paquetages logiciels(interfaces et base d'annotation), qui seront utilisés par la suite aussi bien pour l'indexation de documents que pour la recherche et l'accès par le contenu.

Remarquant que les outils précédents représentent que la partie apparente du SIRSALE, c'est-à-dire l'interface utilisateurs, évidemment l'ensemble de ces outils s'appuiera sur un système multimédia avancé intégrant un ou plusieurs serveurs vidéo, des serveurs de méta \_données et des modules de personnalisation.

## **1.2 Architecture de SIRSALE :**

Le projet SIRSALE manipule des données audiovisuelles, qui ont des contraintes différentes que d'autres types tel que les données textuelles. etc. Alors que ces données audiovisuelles nécessitent des méthodes de stockage, de manipulation et d'accès performant, afin de remédier les contraintes de volume et de continuité inhérente de ces données. L'architecture globale du projet SIRSALE est représentée dans la figure 1.1 :

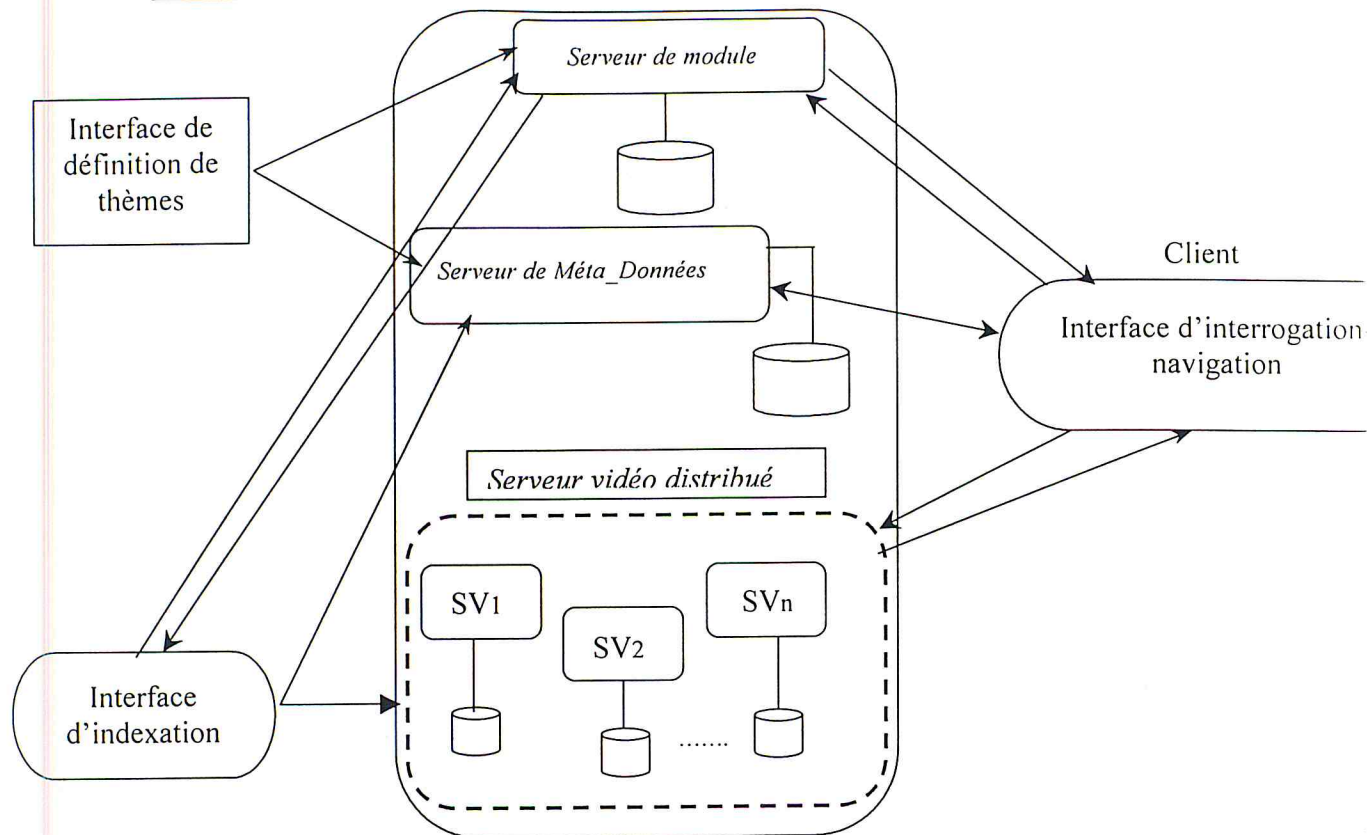


Fig1.1- Architecture générale du SIRSALE

Le système SIRSALE est représenté sous forme de quatre composants principaux :

### 1.2.1 Le composant serveur :

Ce composant est subdivisé en trois éléments principaux définis brièvement comme suit :

#### **Serveur vidéo :**

A comme but de stocker tous les documents vidéo présents dans le système.

#### **Serveur de module (thème) :**

Sert à charger le modèle de données adapté dans les applications d'annotation et d'interrogation.

#### **Serveur de méta\_ données :**

Contient les informations qui caractérisent les documents vidéo comme la taille du document, le taux de flux, la référence du document vidéo, notamment les indexations structurelles et sémantiques des documents vidéo.

### 1.2.2 L'interface d'indexation :

Le rôle principal de cette interface est d'indexer les documents vidéo.

### 1.2.3 L'interface de définition de thème :

Cette interface donne la possibilité de définir de nouveau thème et de l'intégrer, en plus qu'on peut ajouter de nouveaux critères d'indexation et de recherche.

### **1.2.4 L'interface d'interrogation et de navigation :**

C'est la partie visible de SIRSALE, elle est déployée chez le client, elle permet aux clients d'interroger la base de données et de naviguer structurellement dans les documents audiovisuels.

### **1.3 Le fonctionnement de SIRSALE :**

Tout d'abord le client fixe son choix ou le domaine d'intérêt (Thème) à travers une interface spécialisée au client, à la suite le programme principal envoie le choix de l'utilisateur vers le serveur de thèmes, puis ce dernier va transmettre les éléments nécessaires afin de faire une mise à jours à l'interface de l'utilisateur.

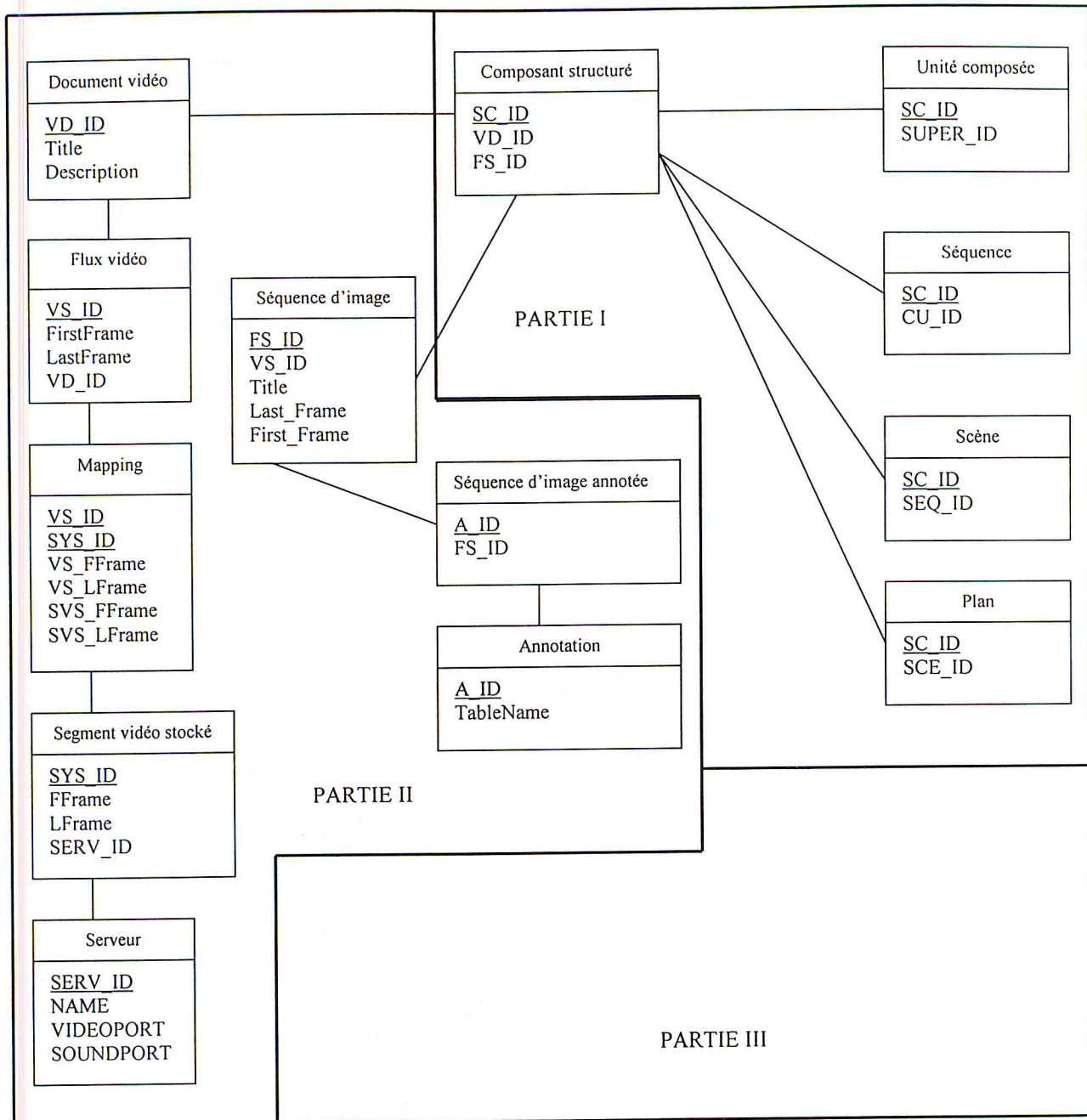
Ensuite le client peut exploiter les documents vidéo via l'interface d'interrogation par la navigation en utilisant le serveur vidéo, ou bien, interroger la base des documents pour rechercher un contenu vidéo bien précis en lançant des requêtes vers le serveur de méta \_ données.

Pour satisfaire les besoins d'exploitation des documents visuels, l'interface graphique dispose d'un lecteur multimédia, ce dernier nous permet de visualiser les documents vidéo sélectionnés pendant la navigation ou l'interrogation à travers des requêtes aux serveurs vidéo.

### **1.4 Le modèle de données existant :**

Actuellement dans le système SIRSALE le modèle de données offre aux utilisateurs, une solution qui gère d'une part l'exploitation des documents vidéo et d'autre part l'interrogation et la recherche des documents audiovisuels. Le modèle de données est schématisé comme suit:





**Description des parties de modèle de données :**

Partie I :

L'idée essentielle de cette partie est de comprendre qu'un document est classifié d'une façon structurale sous forme *Plan, Scène, Séquence* et en fin *l'Unité composée*, cette dernière peut rassembler plusieurs séquences et même par des autres *Unités composées*. Cette partie fournit ainsi un niveau d'abstraction structurel suffisant pour répondre à tous les besoins de découpage structurel qui permet de décrire à la fois un document entier ou bien un sous document seulement, en d'autres termes, elle permet de disposer d'une arborescence structurelle des documents et sous documents vidéo.

## Partie II :

Cette partie joue le rôle d'un gestionnaire de la distribution des documents vidéo numériques. La détection de l'emplacement physique d'un document vidéo se fait à partir de cette partie du modèle de données proposé.

Un document vidéo peut être décomposé en sous documents, ces derniers sont caractérisés par des coordonnées de début et de fin dans le document parent.

## Partie III :

Cette partie varie selon le thème (choix) sélectionné, d'autant plus elle détermine les critères de recherche par rapport à la sémantique d'un document vidéo.

### **1.5 Serveurs :**

#### **1.5.1 Serveur vidéo :**

Le rôle de serveurs vidéo est de stocker les documents vidéo qui ont des caractéristiques importantes celle de l'espace mémoire, rappelons que ce sont les plus consommateurs en espace de stockage et « *Le temps prise en calcul[uc]* » d'une part, d'autre part, la cadence de transaction des documents vidéo qui peuvent s'effectuer entre les serveurs et le/les utilisateurs, pour cela le serveur vidéo doit satisfaire les deux contraintes suivantes :

- i. Le serveur vidéo doit vérifier la continuité de transaction des documents vidéo vers le client sans aucune contrainte, exp. Sa structure physique ou bien sa localisation, on veut dire que l'utilisateur possède uniquement les coordonnées de lecteur du document vidéo via l'interface du SIRSALE.
- ii. La notion du temps réel doit être respecté concernant l'affichage des documents vidéo devant l'utilisateur, puisqu'on peut pas récupérer une portion de séquence vidéo qui arrive tard, cela implique une lecture distante en continu du flux audiovisuel.

#### **1.5.2 Serveur de Méta-Données :**

Le serveur de méta-données contient les informations concernant les documents vidéo c'est à dire ses caractéristiques.

Un Méta-Donnée se définit en deux formes :

La première forme se base sur la notion structurelle, elle regroupe la gestion de la distribution des documents vidéo concernant la taille, la localisation, serveurs on trouve ces informations dans la partie I du modèle de données.

La deuxième notion se base sur la notion sémantique, on trouve les annotations par contenu des documents vidéo.

Il existe deux solutions pour la mise en place du serveur de Méta-Données.



- ⊘ Créer une seule base de données stockant tous les thèmes avec toutes ses annotations. L'avantage de cette idée est la gestion centralisée des données, par contre elle présente une faible fiabilité et d'efficacité.
- ⊘ La deuxième est l'inverse de la première solution, elle consiste d'en créer des bases de données pour chaque thème pour assurer un certain niveau d'efficacité et de tolérance aux pannes, par contre cette solution nécessite des structures de maintien des bases de données.

### 1.5.3 Serveur de thème ou serveur de modules:

Le projet SIRSALE est fondé sur un ensemble de parties génériques comme la visualisation des données vidéo ou la navigation structurelle dans les documents, et de parties spécifiques qui peuvent être développées et intégrées au système afin de s'adapter aux contraintes particulières de chaque centre d'intérêt des documents vidéo stockés. C'est à ce niveau l'architecture SIRSALE se démarque d'autres outils d'indexation et de recherche.

Le concept est le suivant : toutes les fonctionnalités, qui concernent l'interrogation par le contenu sémantique, sont regroupées en packages stockés sur le serveur de thèmes. Ces paquetages sont chargés dynamiquement et ajoutés à l'interface client après que ce dernier ait fait son choix de thème (centre d'intérêt).

Cette approche présente l'avantage d'une application de faible poids qui peut être facilement adaptée aux besoins des utilisateurs. Néanmoins, elle pose des problèmes de mise en œuvre.

## 1.6 Les interfaces :

### 1.6.1 Interface de définition de thèmes:

L'interface de définition de thèmes permet de définir et d'intégrer de nouveaux thèmes (nouveaux critères de recherche et d'indexation) au système SIRSALE. Elle réalise trois tâches principales :

- Création d'une base de données par thème.
- Génération d'un paquetage permettant la mise à jour dynamique de l'interface de client afin d'effectuer des requêtes sémantiques spécifiques à ce thème.
- Génération d'un paquetage d'adaptation dynamique de l'interface d'annotation.

### 1.6.2 Interface d'annotation :

Cette interface regroupe toutes les fonctionnalités nécessaires pour décrire un document vidéo aussi bien structurellement que sémantiquement. Il s'agit donc de construire, à partir d'un document vidéo <<brut>>, à la fois sa vue structurelle et ses éventuelles vues sémantiques. Ces tâches nécessitent l'intervention d'un annotateur et peuvent être réalisées de manière semi-automatique. Compte tenu de leur complexité et de leur nature répétitive, un impératif majeur s'est imposé : rendre cette interface la plus facile et la plus efficace possible, Ceci se traduit par la proposition à la fois des techniques de découpage automatique et paramétrable et par la proposition d'un lecteur

vidéo avancé incluant toutes les fonctionnalités magnéscope.

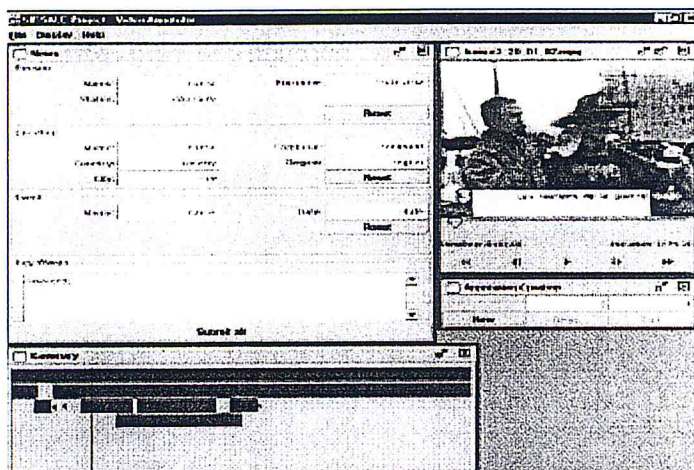


fig1.3. Interface d'annotation du SIRSALE

L'interface d'annotation est donc constituée de trois parties importantes

- L'annotation structurelle.
- L'annotation sémantique.
- Le lecteur vidéo.

*a) Annotation structurelle :*

Cette partie est dédiée à la construction de la vue structurelle du document vidéo conformément au modèle de données. Cette tâche peut être réalisée de manière semi-automatique en se basant sur les caractéristiques physiques des images composant la vidéo. En d'autres termes, l'annotateur a la possibilité de lancer une segmentation automatique de la vidéo en plans.

Le résultat est un ensemble de plans contigus. Bien entendu, cette segmentation n'est pas totalement fiable et pertinente puisque les méthodes automatiques se basent uniquement sur les caractéristiques physiques des images et non pas sur leurs contenus sémantiques. Dans une deuxième étape, l'annotateur peut corriger cette première segmentation en validant les plans jugés pertinents, et en désactivant les plans impertinents. Ainsi, de cette manière, un gain de temps appréciable peut être obtenu. Il reste à l'annotateur de relier les plans validés pour en construire les niveaux d'abstraction supérieurs (scènes, séquences et unités composées).

*b) Annotation sémantique :*

Cette interface permet à l'annotateur de décrire le contenu sémantique de la vidéo à annoter. Donc elle est liée au contexte sémantique de l'annotation.

Cette partie doit par conséquent être d'abord chargée à partir du serveur de module (thème) de la même manière que l'interface de recherche. Elle traduit la partie III du modèle de données de SIRSALE.



### c) Lecteur vidéo:

Dans son activité d'annotation, l'annotateur sera amené à visualiser la vidéo à plusieurs reprises, via une interface spécialisée sous forme un lecteur multimédia.

### 1.6.3 Interface d'interrogation :

L'interface d'interrogation est la partie visible de SIRSALE car elle est déployée chez le client. Elle est composée de quatre parties :

- La fenêtre de navigation
- La fenêtre d'interrogation
- Le lecteur vidéo
- La fenêtre des résultats

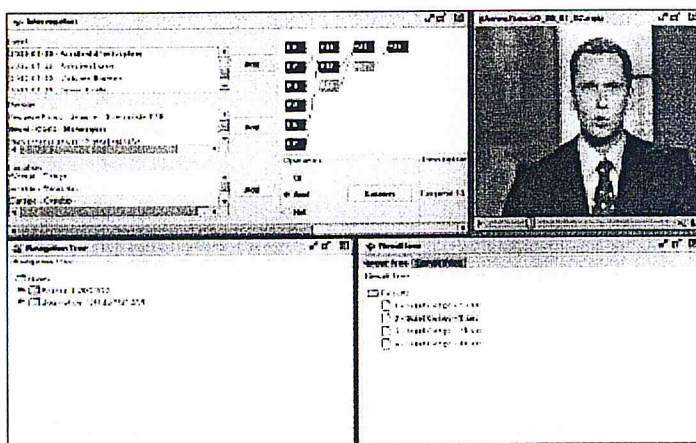


fig.1.4 Interface d'interrogation du SIRSALE

#### a) La fenêtre de navigation :

Elle permet de naviguer au travers de la structure des documents vidéos, grâce à la segmentation structurée. Le navigateur est proposé sous la forme d'un arbre de navigation. Chaque nœud de l'arbre représente un objet structurel de bas niveau<sup>1</sup>.

Les premiers nœuds représentent les unités composées, suivis des séquences, puis des scènes et enfin des plans. La représentation des documents est par des arbres car ils offrent une vue rapide et claire de la structure de chacune des vidéos.

#### b) La fenêtre d'interrogation:

Elle permet en revanche de construire des requêtes sémantiques complexes, reliées par des opérateurs logiques (et, ou ,non), pour l'interrogation de la base d'annotation. Cette interface n'est pas figée puisqu'elle est contenue dans les packages du serveur de thèmes.

<sup>1</sup>Voire chapitre 3

**c) Le lecteur vidéo distant :**

Il permet de jouer la séquence vidéo sélectionnée. Cette opération est réalisée en continue depuis le serveur vidéo. Le transport sur le réseau étant basé sur le protocole RTP<sup>1</sup>.

**d) La fenêtre des résultats :**

Elle permet de visualiser les informations de gestion des séquences vidéo résultantes des requêtes, ou de la navigation structurale. Il y est affiché notamment l'identité du serveur vidéo sur lequel est stockée la séquence ainsi que le nom du fichier vidéo, sa durée... etc.

**1.7 Conclusion :**

Dans ce chapitre nous avons essayé de résumer le projet SIRSALE : un Système d'indexation et de recherche de séquences audiovisuelles à large échelle. Ce projet est fondé sur une approche modulable, utilise à la fois l'indexation structurale de bas niveau pour la navigation et l'indexation sémantique pour l'interrogation, tout en adoptant des modèles de données spécifiquement conçus pour des domaines hétérogènes (news, sport, etc.). Cette approche présente l'avantage de fournir aux utilisateurs des outils pertinents et efficaces afin de rechercher et d'accéder, par le contenu, aux séquences audiovisuelles.

Néanmoins nous avons remarqué les points suivants concernant le thème des journaux télévisés. Premièrement, la structure de l'événement ne couvre pas toute la sémantique du contenu de la séquence, comme le producteur d'événement, deuxièmement l'interface ne permet pas de mettre plusieurs producteurs dans un même événement, troisièmement il n'y a pas un mécanisme de mémorisation des champs qui se répètent fréquemment comme le champ de pays, etc.

Pour cela, nous allons implémenter un lecteur vidéo proposant une panoplie de fonctionnalités magnétoscopes totalement paramétrables : lecture normale, lecture rapide, en ralentie, image par image, avance et retour 5 seconds, recul rapide, etc. Cette approche requiert en revanche une mise en œuvre plus complexe, remédiant aux problèmes cités précédemment.



# chapitre:II

## gestion multimédia

## 2.1 Introduction :

De nombreux secteurs d'activité sont concernés par l'émergence des techniques multimédias : les jeux électroniques, le tourisme, l'enseignement assisté par ordinateur (EAO) sont actuellement les cadres des plus nombreuses applications car l'apport de ces techniques est important. D'autres domaines s'intéressent aux documents structurés multimédias est notamment la médecine. Les données issues des plateaux techniques d'imagerie médicale peuvent être exploitées sous forme numérique et intégrées à des données textuelles, et sonores afin de former de véritables documents multimédias médicaux qui peuvent être consultés par les médecins.

Cet intérêt pour le multimédia résulte d'une part de l'avancement remarquable qu'ont enregistrées les nouveaux supports de communication et d'autre part, de l'augmentation spectaculaire de la puissance des ordinateurs (puissance de calculs, capacité de stockage, etc.) permettant aujourd'hui un traitement efficace des documents multimédias [Most00]. Aussi, l'augmentation de la qualité de l'information présentée, et la variété des types de média utilisés apportent de nouvelles possibilités aux créateurs en leur permettant de jouer sur un plus grand nombre de mode de perception. Enfin, les enjeux économiques importants que génèrent les applications multimédias conjugués à la vulgarisation d'Internet ont encore accentué cet intérêt.

Parmi l'ensemble des structures des données multimédia, la vidéo est très certainement celle porteuse des perspectives applicatives les plus prometteuses, d'ores et déjà, des projets, représentant des investissements considérables se structurent dans tous les pays du monde autour de la télévision numérique ; De même, de plus en plus de sociétés, s'intéressent aux applications de l'information audiovisuelle sur le WEB et de télévision sur Internet.

## 2.2 Caractéristiques d'un document multimédia :

Afin de répondre aux besoins de représentation multimédia, un document multimédia doit :

- Offrir un pouvoir d'expression puissant (structuration, définition d'informations temporelles et spatiales ) qui répond aux besoins de spécification des documents multimédias ;
- Être fondé sur les concepts qui sont proches de la représentation mentale des auteurs et qui facilitent le processus d'édition [LayM99];
- Rendre possible la réutilisation et l'évolution des informations multimédias dans les documents.
- Avoir une syntaxe qui respecte les standards existants (utilisation XML) de façon à tirer parti des outils disponibles (comme les *parsers*) [ChFr00] ;
- Accepter les différents formats de codages qui existent pour chaque catégorie ( ASCII pour le texte, jpeg, gif, ... pour les images, mpeg, avi, ... pour la vidéo, etc.) ;
- Prendre en charge les nouveaux formats.

### 2.3 Édition d'un document multimédia :

Les documents multimédias intègrent une dimension temporelle en plus des dimensions spatiales et spatio-temporelle.

Dans la dimension spatiale, la composition de média est faite par rapport à des positions dans un espace à deux ou trois dimensions. Cependant la dimension temporelle est définie d'une part par les caractéristiques temporelles des objets multimédias (image, son, vidéo, etc.) et, d'autre part, par l'enchaînement de ces objets. Par contre la dimension spatio-temporelle c'est une composition des deux approches spatial et temporelle dans la qu'elle les objet médias sont composés dans le temps et dans l'espace.

L'édition de documents comportant des objets multimédias (image, son, vidéo, etc.) organisés dans la dimension temporelle est une tâche complexe, car l'entité à construire est une entité dynamique dont il faut spécifier le scénario temporel.

De plus, le comportement d'un document multimédia peut varier d'une présentation à une autre à cause des réactions aux interactions du lecteur et/ou du contexte de présentation. De ce fait, la nature dynamique des objets manipulés ainsi que la variété des comportements qui en résulte empêchent la fusion simple d'édition et de présentation multimédia.

L'objectif d'un environnement de composition des documents multimédias est de combiner au mieux un fort pouvoir d'expression tout en conservant un mode d'édition interactif et conviviale [LayM99]. Aujourd'hui, on distingue plusieurs catégories d'approches pour les environnements d'édition multimédia :

- Les approches fondées sur des langages de script. Avec ce type de langage de programmation, il est possible d'exprimer des scénarios aussi complexes que l'on veut. Le scénario est spécifié sous forme opératoire par un ensemble d'instructions <<événement & conditions -> actions >>. Le point faible de ces approches vient de leurs capacités limitées à répondre aux besoins des auteurs. Enfin, comme pour tout langage de programmation, la mise à jour d'un scénario est une tâche souvent délicate du fait des erreurs qu'il facile d'introduire.
- Les approches fondées sur l'utilisation d'une interface conviviale, mais qui ont souvent un pouvoir d'expression relativement pauvre, ce type offre un ensemble de modèles prédéfinis où l'auteur n'a plus qu'à affecter un contenu aux objets prédéfinis. L'inconvénient majeur de cette approche, qui a la mérite de permettre à des auteurs non informaticiens de réaliser des présentations multimédias, est son incapacité à s'adapter à des besoins de création plus avancés, notamment en ne permettant pas à l'auteur de sortir des structures prédéfinies[Sabi99].
- Les approches qui essayent d'avoir à la fois un environnement convivial tout en offrant un fort pouvoir d'expression dans la dimension temporelle. Ces langages permettent d'exprimer l'enchaînement temporel entre les objets par le biais d'opérateurs temporels comme l'opérateur séquence et l'opérateur parallèle. Ces opérateurs sont suffisamment simples et intuitifs pour être utilisés par des auteurs non informaticiens. Cependant, on peut regretter dans ces environnements la faible intégration de la phase d'édition, pendant laquelle l'auteur définit la structure spatiale et temporelle du document, avec la phase de présentation où l'auteur voit le résultat de ses opérations d'édition. [ChFr00]



## 2.4 Évolution des données :

Il est courant, dans les processus actuels, que les unités documentaires subissent des évolutions (correction des données, ajout d'information...). Pour une documentation classique de type texte et graphique, ces évolutions sont gérées à la production des documents finaux : l'incorporation de l'unité documentaire met à jour automatiquement ces derniers.

Dans le cadre de documents multimédias, la présence de la dimension temporelle induit que la prise en compte des évolutions de données soit réalisée avant la production des documents. On pourrait en effet se retrouver devant l'impossibilité d'assurer l'assemblage du document : une vidéo et une bande son doivent avoir la même durée lors de la présentation du document mais la bande son a vu sa durée doublée. Dans ce cas une partie de la bande son risque d'être accompagnée d'une image vidéo.

Il est donc nécessaire de vérifier que la donnée qui évolue pourra être présentée en accord avec les spécifications temporelles de l'unité documentaire dont elle fait partie, et que le système disposera des informations nécessaires à sa présentation. Du fait de la réutilisation des unités documentaires, cette vérification devra être réalisée dans chacun des contextes d'utilisation de la données qui évolue.

## 2.5 Modèle de document multimédia :

Les documents multimédias se caractérisent par un ensemble des objets médias de nature très divers, il existe deux modèles de document multimédia :

- Un modèle pour les objets médias de base.
- Un modèle de composition.

### 2.5.1 Le modèle des objets de base :

L'objectif de ce modèle est de regrouper de façon homogène ces médias, grâce aux mécanismes d'encapsulation et d'abstraction de l'approche objet<sup>1</sup> [ChFr00].

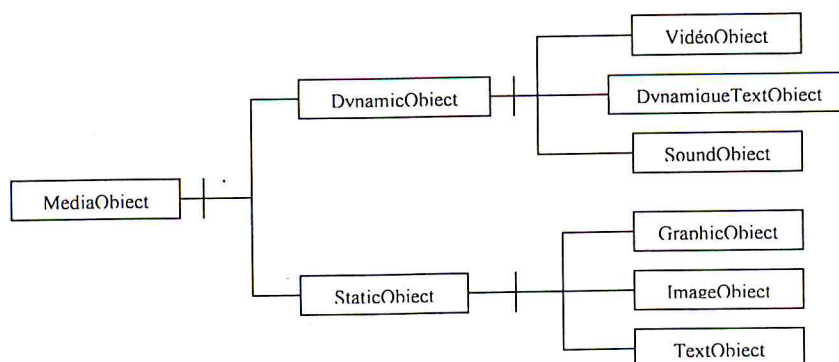


Fig.2.1 Modèle des objets de base

<sup>1</sup>Voire chapitre 4

Cette figure est la hiérarchie de classes du modèle des objets de base. La classe MediaObject caractérise les propriétés communes à tous les médias de base. Ces propriétés peuvent être par exemple la taille d'affichage, la nature critique ou non de l'information contenue, etc. Cette classe peut être spécialisée ( par relation d'héritage ) afin de répondre à l'évolution de nouveaux types de médias. Les médias sont classés selon qu'ils sont statiques (GraphicObject, TextObject, ImageObject) ou dynamiques (VideoObject, SoundObject, DynamiqueTextObject.)

### 2.5.2 Modèle de composition :

Au modèle des objets médias, nous ajoutons un modèle qui décrit les différents types de relation de composition et de synchronisation qui existent entre ces objets à l'intérieur d'un document multimédia [ChFr00].

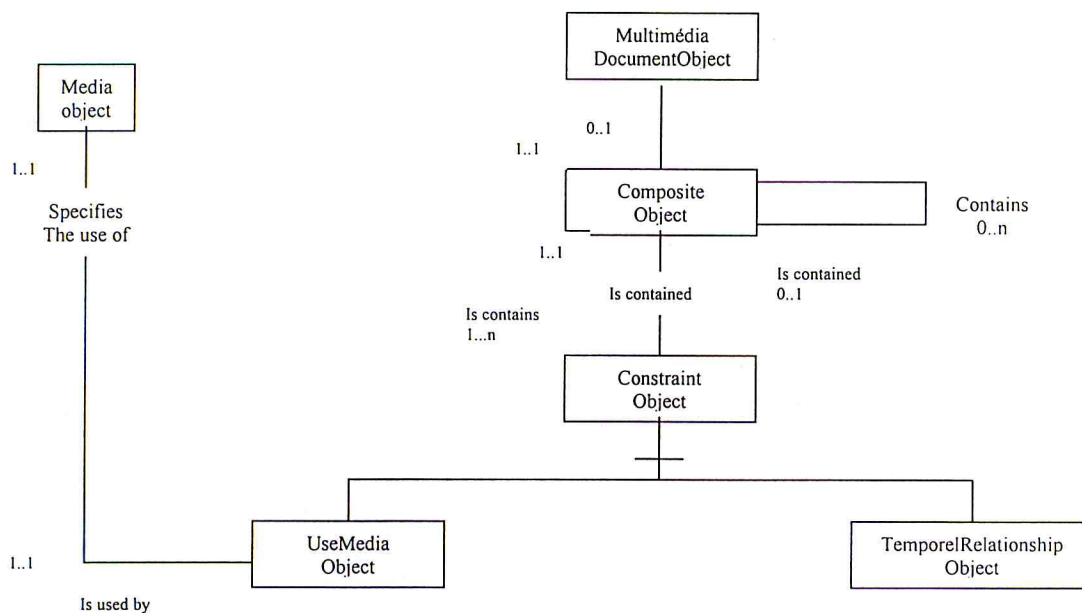


Fig.2.2 Modèle de composition

Un document multimédia (MultimediaDocumentObject) se décompose ainsi en une structure arborescence dont les nœuds portent le nom de CompositeObject. Ceux-ci contiennent en plus de leurs CompositeObject fils, des objets de type ConstraintObject qui correspondent soit à des contraintes d'utilisation des médias (UseMediaObject) soit à des relations temporelles (TemporalRelationshipObject) [ChFr00]. L'objet UseMediaObject est défini pour apporter une solution à la possible réutilisation d'un même média dans un ou plusieurs documents multimédia, comme par exemple une bande sonore réutilisée dans plusieurs tâche de maintenance. L'objet TemporalRelationshipObject est la représentation de relations de synchronisation de type binaire entre des objets de type CompositeObject ou UseMediaObject, comme par exemple, A before (t) B, l'objet A se joue avant l'objet B avec un décalage de t unités de temps.

## 2.6 Environnement auteur :

Certains critères doivent être respectés pour un environnement de création de document multimédia, l'environnement aide l'auteur à maîtriser la complexité de la tâche d'édition :

- L'environnement doit permettre à l'auteur de réduire la complexité de son document et de lui faciliter la réutilisation des parties déjà spécifiées. Ceci peut être permis par exemple l'utilisation d'une méthode de décomposition du document (décomposition hiérarchique et/ou logique [DecF99]) ;

L'environnement auteur doit permettre de modifier facilement les documents en cours de conception de façon à permettre un processus incrémental de création. Ceci peut être permis par exemple l'utilisation de relations entre les objets pour décrire leur placement plutôt que l'utilisation d'un placement absolu ;

- L'environnement auteur doit offrir des supports visuels permettant de percevoir le scénario en cours de spécification ;
- L'environnement auteur doit être utilisable sans compétences en programmation informatique.

Cependant, une boîte de Kaomi [DecF99] est proposée, qui intègre les fonctions de présentation au sein de l'environnement d'édition, et offre aussi la vérification du scénario etc. c'est à dire les possibilités liées à l'usage des contraintes.

### 2.6.1 Exemple de construction d'un environnement : la boîte à outils Kaomi :

Kaomi a été développée en Java en utilisant une architecture orientée objet qui permet l'implantation d'un ensemble de services pour l'édition dans un environnement multidocuments et multivues extensible.

Les principaux critères retenus pour la conception de cette boîte à outils ont été :

- La portabilité de l'application ;
- La réutilisation de composants logiciels existants (JMF, xml4j, Swing)<sup>1</sup> ;
- L'extensibilité pour l'ajout de nouveaux services, de nouvelles vues ;
- L'évolutivité des services existants qui permet de remplacer certains des composants utilisés (résolveurs de contraintes, players de médias, etc.) .

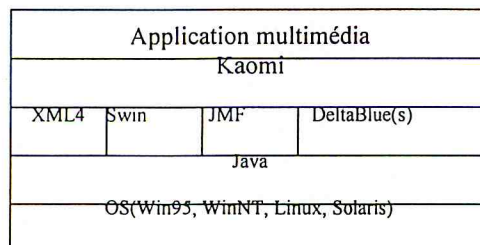


Fig. 2.3 Boîte à outils Kaomi

<sup>1</sup>Voire Annexes B&C



## 2.7 La réutilisation des unités documentaires :

Cependant, il apparaît que dans le cadre du multimédia un niveau de réutilisation n'est pas suffisant. Un système multimédia complet doit permettre la réutilisation non seulement des médias mais aussi des unités documentaires déjà réalisées : elles sont alors utilisées comme les médias décrits dans le modèle des objets de base.

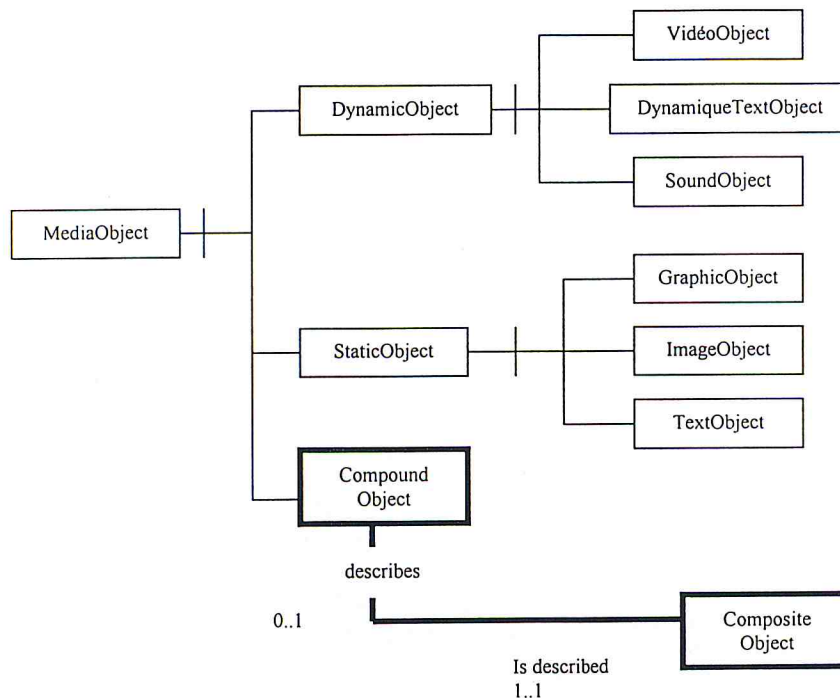


Fig.2.4 Prise en compte de la réutilisation [chfran00]

Dans [ChFr00] les auteurs proposent de spécialiser l'objet MediaObject par l'objet CompoundObject. Cet objet est la représentation dans ce modèle d'un assemblage multimédia déjà réalisé et réutilisable. C'est pourquoi cet objet est en relation avec un objet de type CompositeObject qui assure la description du scénario du CompoundObject.

La réutilisation est donc rendue possible et complète avec cette extension. parfois des contraintes temporelles d'un document multimédia ne peuvent être satisfaites, De même, dans le contexte de la réutilisation des unités documentaires déjà créées l'évaluation de l'une d'elles peut avoir un impact si important sur celles qui l'utilisent, qu'il sera nécessaire de gérer son évolution manuellement pour éviter de propager des modifications à de trop nombreuses unités documentaires. [ChFr00]

## 2.8 Présentation d'un document multimédia:

Avant de pouvoir effectuer la présentation d'un document, il est nécessaire de le rassembler à partir de son modèle et des unités documentaires contenues dans le fonds

documentaire multimédia. Cette phase de production est réalisée par un parcours de l'arbre des composites d'un document multimédia au cours duquel on reconstruit un document XML<sup>1</sup> décrivant les médias utilisés et le scénario temporel. Il n'y a pas au cours de cette étape de vérifications temporelles car celles-ci ont été réalisées lors de la gestion des unités documentaires dans le fonds.

### **2.8.1 Notion de synchronisation multimédia :**

Le facteur temps représente une dimension importante dans les applications multimédia, surtout pour les données dynamiques, à l'opposé des données statiques où cette dimension n'a pas une grande importance. On trouve une forme très primitive dans les outils d'édition interactifs d'exposés comme PowerPoint, dont le temps est simplement considéré comme une succession d'évènements.

En générale, la transmission et la restitution de ce type de documents sur Internet soulève de nouveaux problèmes liés principalement au manque de stratégies de gestion de ressources qui prennent en compte les contraintes de synchronisation liées à cette dimension temporelle.

#### **2.8.1.1 Synchronisation intra-objet :**

Ce type de synchronisation s'applique aux relations temporelles d'unités d'information de base formant un objet d'un même média, un exemple caractéristique de synchronisation intra-objet est la relation entre les images successives d'une séquence vidéo. Pour une vitesse de 25 images/secondes, chaque image de la séquence doit être affichée à l'utilisateur pour une durée 40 milli-secondes.

#### **2.8.1.2 Synchronisation inter-objet :**

La synchronisation inter-objet s'applique aux enchaînements de la présentation de plusieurs éléments multimédias, par exemple, la présentation simultanée d'un élément audio et d'une séquence vidéo suivie par des éléments de type texte et image.

#### **2.8.1.3 Synchronisation des lèvres :**

Ce type de synchronisation est une synchronisation des deux dernières. La synchronisation des lèvres impose un couplage temporel fort entre la progression temporelle de deux ou plusieurs objets multimédia (leur flux). Ce couplage est généralement exprimé en terme de décalage temporel admissible entre les flux.

L'exemple typique de cette forme de synchronisation correspond à la présentation simultanée d'un discours audio et de la séquence vidéo qui lui est associée.

Ces contraintes permettent d'exprimer les conditions de maintien de la voix en accord avec le mouvement des lèvres. Cette forme de synchronisation n'est cependant pas réservée qu'au couplage des médias de type audio et vidéo. On peut en effet la retrouver dans d'autres situations, comme celle illustrée dans la figure suivante. Dans cet exemple, un objet graphique (carré) est utilisé pour annoter un objet en mouvement dans une séquence vidéo.

---

<sup>1</sup> XML : permet de structurer des données à l'aide d'une norme de balisage



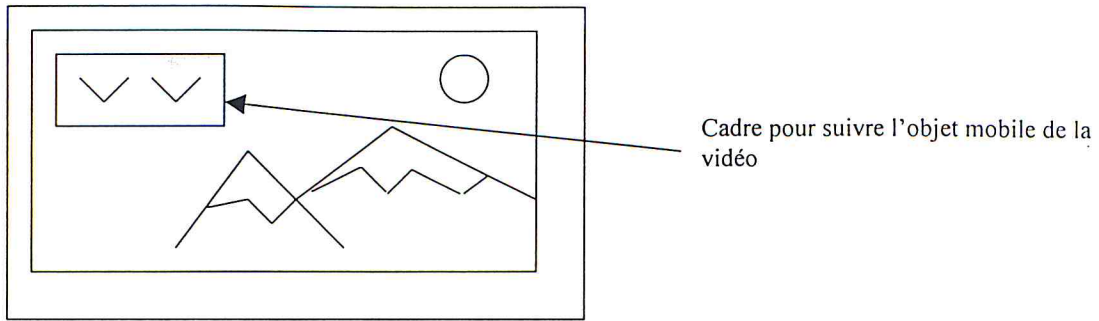


Fig.2.5 Synchronisation des lèvres

## 2.8.2 Support de présentation :

Voici les principes de conception d'une machine d'exécution multimédia au travers de son architecture générale, et ensuite les composants qui réalisent les fonctions d'ordonnancement et de contrôle de l'exécution.

### 2.8.2.1 Caractéristiques et besoins de présentation multimédia :

Voici cinq caractéristiques dont il est nécessaire de tenir compte lors de la réalisation d'une machine de présentation multimédia. L'importance des différents aspects est plus au moins grand selon le type de présentation considéré.

#### a) Hétérogénéité des objets médias :

Les objets médias qui font partie d'une présentation multimédia ont des caractéristiques hétérogènes comme leur comportement temporel, leur format de codage, leur volume et leur mode de perception par l'utilisateur.

#### b) Organisation des objets médias entre eux :

Une présentation multimédia n'est pas une simple présentation d'objets médias indépendant. C'est un processus qui traite un ensemble d'objets organisés logiquement, spatialement et temporellement, et qui doit donc assurer que les spécifications de composition (en particulier temporelle) sont respectées lors de la présentation, quelles que soient les conditions extérieures.

#### c) Portabilité des applications multimédias :

C'est une caractéristique qui touche non seulement à la possibilité de jouer des présentations multimédias sur plusieurs plates-formes (code exécutable, cartes périphériques pour les médias), mais aussi à la qualité de leur exécution : En effet, le comportement temporel d'une présentation est très sensible à des paramètres comme la vitesse du processeur, la charge de la machine ou encore les accès au réseau et aux périphériques.

#### d) Dynamique de la présentation multimédia :

Le processus de présentation d'un document multimédia est un processus dynamique où, à chaque instant de la présentation, les objets médias changent d'état selon le scénario spécifié et les interactions par l'utilisateur. La façon selon laquelle les



changements d'état doivent se produire est définie par le mode de synchronisation de la présentation multimédia : synchronisation intra-objet, synchronisation inter-objet et synchronisation des lèvres.

### e) Répartition des objets multimédias :

Les objets médias faisant partie d'une même application multimédia peuvent être distribués sur différents serveurs : Soit sur le même site, soit sur des sites différents. Du fait de sa grande taille et des aléas du débit des liaisons numériques, l'accès à un objet peut subir un délai plus au moins important et d'une valeur aléatoire. Par conséquent, la qualité de la présentation de ces objets peut être dégradée à cause du non respect des échéances temporelles.

### 2.8.3 Architecture générale :

La machine d'exécution réalise les fonctions de présentations en tenant compte de l'interaction utilisateur : elle gère la synchronisation entre les objets médias, la navigation temporelle intra-document et inter-document ainsi que l'indétermination de la présentation.

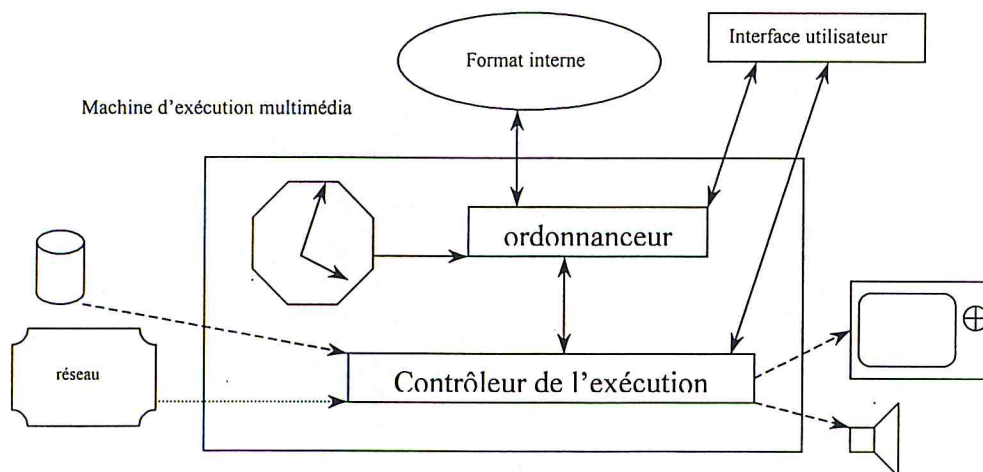


Fig.2.6 Architecture générale d'une machine de présentation multimédia[walda01]

Les fonctions à réaliser sont divisées en deux groupes : la première constitue l'ensemble des fonctions de haut niveau qui réalisent l'Ordonnement des actions de présentations, tandis que la seconde contrôle et gère les accès aux ressources physiques. Plus précisément, on a :

- Les fonctions d'ordonnement : synchronisation à gros grain (interobjet), synchronisation fine (inter et intra-objet), gestion de la navigation temporelle, gestion de la présentation des objets médias ;
- Les fonctions de contrôle de l'exécution (en générale dépendant de la plate-forme d'exécution) : gestion des accès et des périphériques.

### 2.8.3.1 Ordonnement d'une présentation multimédia :

La présentation d'un document multimédia est une séquence d'actions d'exécution appliquée aux dates correspondantes à la spécification temporelle donnée par le scénario et qui s'applique sur l'ensemble des objets impliqués dans le document. Ces actions sont les suivantes :

- afficher/cacher un objet sur l'écran ou commencer/arrêter la sortie d'un objet audible sur un canal sonore ;
- jouer une entité d'un flot composant un objet contenu comme par exemple, une image d'une séquence vidéo ou un échantillon d'audio. Cette action nécessite des opérations supplémentaires pour accéder à cette entité et, en général, pour la décompresser ;
- jouer un effet de style tel que déplacer un objet sur l'écran, changer le niveau de volume d'une séquence audio ou changer périodiquement la couleur d'un texte.

C'est l'ordonnement qui décide les actions de présentation à effectuer, en utilisant pour cela un ensemble d'informations internes et externes fournies par différentes sources :

- la représentation intermédiaire (format interne) qui donne l'ordonnement de présentation d'objets selon le scénario spécifié ;
- l'état d'exécution des objets médias (actif, affiché, suspendu, terminé, etc.) que contrôle le gestionnaire d'objets ;
- l'horloge système qui sert à réaliser effectivement la durée spécifiée pour différents types d'objets comme :
  - les délais temporels associés à certaines relations temporelles ou spécifiés explicitement par l'auteur ;
  - les objets discrets temporisés, par exemple une image qui s'affiche pendant 5 seconds ;
  - les effets de style de présentation, par exemple le déplacement d'un objet sur l'écran avec une vitesse spécifiée.
- l'interface utilisateur qui spécifie la commande de présentation choisie par l'utilisateur .

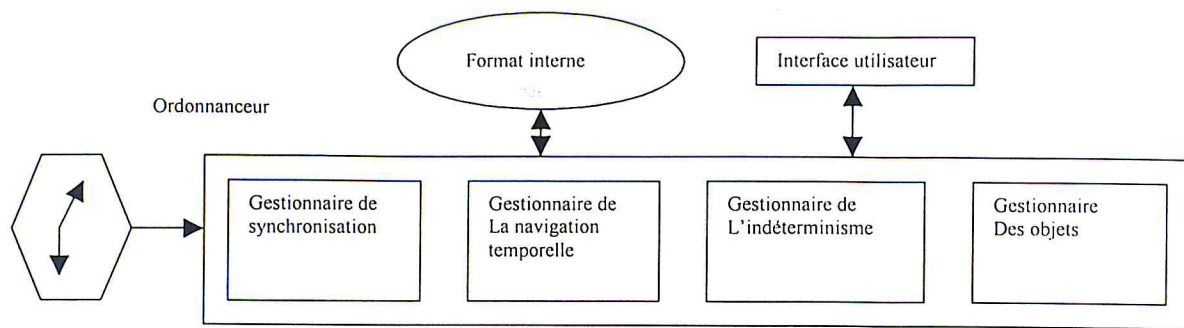


Fig.2.7 Structure de l'Ordonnanceur

L'organisation de l'ordonnancement est décrite dans cette figure. Celui-ci se compose d'un gestionnaire d'évènements et de quatre autres gestionnaires : le gestionnaire de synchronisation, le gestionnaire des objets médias, le gestionnaire de la navigation temporelle et le gestionnaire de l'indéterminisme. Le gestionnaire d'évènements de l'Ordonnanceur filtre les évènements engendrés par la présentation et les notifie au(x) gestionnaire(s) concerné(s).

### Gestionnaire de synchronisation :

Deux types de synchronisation sont à prendre en compte lors d'une présentation multimédia :

- *La synchronisation à gros gain* qui définit la synchronisation entre les instants de début et/ou de fin des objets telle quelle définie dans le scénario.
- *La synchronisation fine* est définie au niveau d'un seul objet ou entre deux objets fortement liés, comme par exemple, un film qui se compose d'une séquence vidéo et d'une séquence audio.

### Gestionnaire de la navigation temporelle :

Le gestionnaire de la navigation temporelle interprète les commandes de navigation lancées par l'utilisateur pour lui permettre de se déplacer temporellement à travers les documents multimédias. Cette forme de navigation est fournie au travers de boutons de contrôle du temps au niveau du système de présentation, comme les boutons de pause, reprise, accélération en avant et en arrière, etc. Un tel gestionnaire est facile à mettre en œuvre pour le contrôle d'un seul média (une séquence vidéo par exemple), mais se révèle plus complexe dans le cas d'informations multimédias structurées et fortement synchronisées. En effet les changements de l'unité temporelle (pour l'accélération) et les sauts doivent être effectués tout en respectant la structure temporelle du document.

### Gestionnaire dynamique de l'indéterminisme :

Dans la section précédente, la synchronisation prend en compte uniquement l'aspect déterministe des objets multimédias, elle ne traite pas les retards et le comportement indéterministe des objets multimédias. En effet, les systèmes d'exploitation actuels, y compris les systèmes temps réel, les protocoles réseau et la nature de certains médias comme les programmes sont autant de sources d'indéterminisme qu'il faut prendre en compte. Ces comportements incontrôlables peuvent conduire à une désynchronisation des objets médias présentés et à un non-respect du scénario spécifié par l'auteur.



Donc il est très nécessaire d'effectuer un traitement pendant la présentation du document multimédia. Ce traitement comporte deux phases : la phase d'observation et la phase de recouvrement. La phase d'observation permet de relever les valeurs effectives des intervalles incontrôlables, par exemple la durée d'un objet incontrôlable (comme un bouton d'interaction) est observée lors de sa terminaison. Ces valeurs sont ensuite utilisées pour ajuster le scénario dynamiquement. Cet ajustement se traduit par la resynchronisation des objets en cours de présentation et des objets futurs qui se trouvent en retard ou en avance les uns par rapport aux autres. Pour cela, la durée de certains délais, celles de certains objets ou éventuellement leurs vitesses nominales sont modifiées dans les limites prévues par le scénario (ces limites sont identifiées sous la forme de valeurs de flexibilité des objets[LayM99]).

### Gestionnaire des objets médias :

Le gestionnaire d'objet a pour rôle de créer et de contrôler les objets médias nécessaires à la présentation du document. Cette gestion concerne tous les traitements indépendants du contexte d'exécution proprement dit (périphériques utilisés, espace de stockage local ou distant). Ce sont les gestionnaires d'accès et des périphériques qui contrôlent ces traitements spécifiques. On peut associer un graphe de transitions qui décrit le comportement des objets médias en fonction des événements qui leur sont adressés et des actions effectuées ou subies, cette figure montre le graphe de transitions d'un objet [Sabi99].

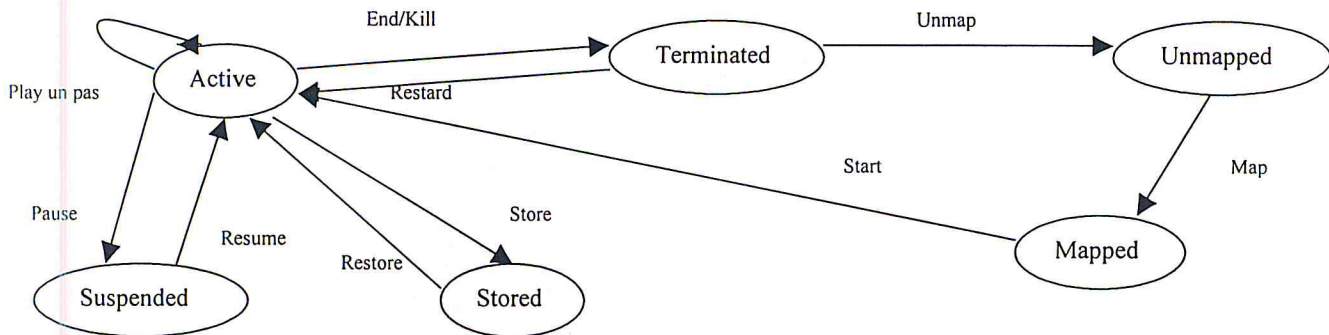


Fig.2.7 Graphe de transitions d'objet média

Les différents états de graphe de transitions sont :

- Unmapped : si l'objet n'est pas exécuté et qu'aucune ressource de présentation (fenêtre d'affichage, canal audio, etc.) ne lui est allouée ou s'il est terminé et que toutes ses ressources de présentation sont libérées ;
- Mapped : l'objet est prêt à être exécuté, le flot de données qu'il doit interpréter est prêt et les ressources de présentation sont allouées ;
- Active : l'objet média est en cours d'exécution ;
- Suspended : l'objet média est suspendu pour des raisons externes (un lecteur a stoppé l'objet) ou internes (pour une synchronisation : l'objet est bloqué jusqu'à l'occurrence d'un événement donné) ;
- Terminated : l'exécution de l'objet média s'est terminée mais ressources de présentation sont encore allouées ;
- Stored : un objet média atteint cet état lorsque, par exemple, un hyperlien vers un autre document est activé. A ce moment, l'exécution de l'objet est interrompue et il

passé dans l'état stored. Les informations sur la présentation et l'exécution de l'objet média sont alors stockées ( par exemple le numéro de l'image courante pour un clip vidéo). Ces informations peuvent être réutilisées pour réactiver l'objet à partir de l'instant exact de sa suspension.

### 2.8.3.2 Contrôleur de l'exécution d'une présentation multimédia :

Ce composant se charge d'effectuer les fonctions de présentation au niveau du système d'exploitation et du système de communication. En effet, il gère : la création des processus qui se charge de la présentation des différents objets médias, la communication entre ces processus et l'Ordonnanceur, le contrôle des périphériques de présentation, et l'accès aux objets médias locaux et distants[Walda01].

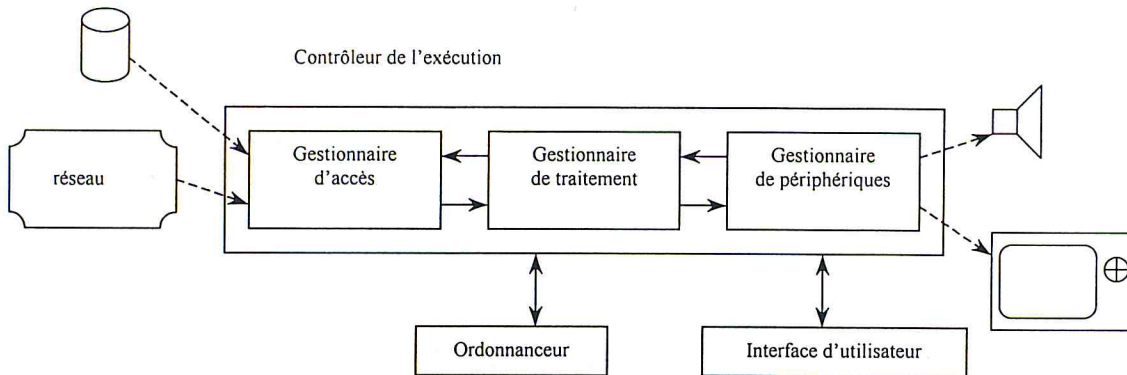


Fig.2.8 Contrôleur de l'exécution

Le contrôleur de l'exécution se compose d'un gestionnaire d'accès, d'un gestionnaire de traitement et d'un gestionnaire de périphériques.

#### Gestionnaire d'accès :

Le gestionnaire d'accès est la partie de l'application qui est responsable du chargement des objets médias et, à ce titre, il joue un rôle important dans la performance du système de présentation. Le temps qui s'écoule entre le lancement d'une requête de chargement et le chargement effectif doit être le plus court possible. Cependant le transfert d'un objet média continu ( comme la vidéo ) est soumis à des contraintes temps réel. Le protocole spécifique a été spécifier pour permettre de transmettre des médias continus, il est décrit dans le paragraphe ci-dessous.

#### Protocole d'accès distant pour médias continus :

RTSP (Real Time Streaming Protocol) est le protocole temps réel conçu pour transférer et contrôler un ou plusieurs flux synchronisés d'objets médias continus comme les produits vidéos et audio. Le protocole est fondé sur le modèle client-serveur. Il permet le contrôle de la présentation d'un objet média en agissant sur le flot de données transmis entre le client et le serveur. Autrement dit, les commandes de contrôle de la présentation



d'un objet (comme démarrer, stopper, faire une pause, etc.) sont traduites sous la forme de requêtes d'accès RTSP envoyées par le client et exécutées par le serveur.

### **Gestionnaire de traitement :**

Le but du gestionnaire de traitement est de fournir les mécanismes permettant l'exécution de différents types et formats d'objets médias. Il assure également le traitement des événements. En effet, au niveau le plus bas du système, tous les changements d'état des objets ainsi que les événements externes se traduisent par l'échange d'un ensemble d'événements. L'occurrence de chaque événement implique le déclenchement d'un traitement particulier au sein de système de présentation. La gestion d'événements est donc un élément central dans la mise en œuvre du gestionnaire de présentation.

### **Gestionnaire de périphériques :**

La restitution des données à l'utilisateur à travers les périphériques de la machine est effectuée par deux gestionnaires principaux : le gestionnaire graphique et le gestionnaire audio. Ces gestionnaires fonctionnent comme des serveurs qui reçoivent des demandes de création ou de suppression de modules destination attachés à chaque objet de base. Ils permettent aussi de prendre en compte les relations graphiques comme le recouvrement d'objet sur l'écran ainsi que le périphérique audio (mixage, musique de fond, etc.).

### **2.9 Conclusion :**

Nous avons étudié la représentation de l'information multimédia, les caractéristiques, l'édition, la production et la représentation des documents multimédias. Il est clair que pour chacune de ces phases, tous les besoins ne sont pas encore complètement couverts. En particulier, il faut prendre en compte les propriétés de placement spatial des objets médias visibles. Dans un contexte de document multimédia, ces informations spatiales peuvent dépendre de contraintes temporelles ou peuvent être définies avec un paramètre temporel (déplacement d'objet par exemple). Comme pour la dimension temporelle, le placement spatial doit être spécifié sous forme relative et non absolue, pour faciliter le processus d'édition.

Enfin, avant d'envisager et intégrer la technologie multimédia dans les circuits de production documentaire, il est nécessaire d'arriver à un niveau d'automatisation des processus de production qui soit équivalent à ceux qui existent pour la documentation technique actuelle. En particulier, il est nécessaire de travailler à la spécification de modèles génériques qui prennent en compte la dimension temporelle[Walda01].





# chapitre:III

## Indexation audiovisuelle

### 3.1 Introduction :

L'indexation audiovisuelle est un sujet d'actualité. Plusieurs raisons peuvent expliquer l'engouement pour cette ancienne problématique, Jusqu'à présent, en effet, l'indexation était surtout le fait de pratiques spécialisées propres à des communautés bien identifiées.

Le problème est de trouver une ou plusieurs séquences vidéo parmi plusieurs documents vidéo. On distingue deux aspects dans cette problématique : la création des index d'une part, et la consultation des documents à l'aide de ces index, d'autre part. Traditionnellement, l'indexation doit être identifiée avec le premier problème, le problème de la consultation étant la base de la recherche documentaire.

Ainsi l'indexation est le processus de créer et constituer des index pour permettre la recherche documentaire. La création des index repose sur l'analyse ou l'interprétation du continu du document de manière à en dégager les caractéristiques les plus représentatives. Cette définition appelle les remarques suivantes :

- l'indexation est une interprétation qui est soumise, d'une part, à la subjectivité de l'indexeur, dont on tente de surmonter la variabilité et, d'autre part, à la finalité de la consultation ( pour qui index-t-on ?, que veulent faire les usagers avec les documents retrouvés ? etc. ), que l'on veut optimiser ;
- l'indexation ne rend pas compte des propriétés intrinsèques et objectives d'un document mais exprime un point de vue en fonction d'une exploitation visée des documents (retrouver les contenus pour quoi faire ? ). Autrement dit, il ne peut exister d'indexation universelle [BacB99], ou d'indexation qui soit indépendante du contexte.

Le fait qu'il n'ait pas d'indexation universelle n'implique pas qu'elle soit arbitraire ni qu'elle ne puisse être motivée ou argumentée[ChaM00]. Cependant il faut rappeler que la description des documents audiovisuels dépend pleinement de la finalité de l'indexation et de l'exploitation de ces documents. On peut, de manière générale distinguer deux principales classes d'exploitation :

- Une exploitation patrimoniale ou globale, dans laquelle le document est conservé à des fins de mémoire. L'enjeu est de préserver, d'une part, l'intégrité du support matériel et, d'autre part, l'intelligibilité du contenu pour les temps futurs. Cette indexation est donc conçue sur un temps long de l'exploitation du contenu, indépendamment d'un usage particulier. D'ailleurs, elle s'effectue quand le document ne possède plus de valeur d'usage ou commercial, c'est-à-dire au terme de son cycle de vie. C'est pourquoi cette indexation s'adresse en particulier aux chercheurs, qui interrogent dans leur travail scientifique, les conditions d'intelligibilité du contenu et maintiennent ainsi sa lisibilité.
- Une exploitation professionnelle ou locale [ChaM00], dans laquelle le document est conservé à des fins de ré-exploitation ou réutilisation. L'enjeu est de retrouver et de réutiliser le contenu dans des applications définies par des objectifs professionnels, techniques, ou commerciaux. Cette indexation

est donc conçue sur un temps court, le temps de la valeur d'usage et d'exploitation des documents. Elle sera exploitée, par exemple, par les journalistes, les producteurs audiovisuels, qui veulent réutiliser des contenus audiovisuels dans leurs programmes, ou les enseignants qui veulent mobiliser de tels contenus dans leurs supports pédagogiques.

Il est très important aussi de noter que l'indexation ne dépend pas seulement du contexte d'utilisation, le processus d'archivage est aussi fortement lié au contenu de la vidéo. En effet, nous ne pouvons pas utiliser les mêmes modèles pour indexer une rencontre de football et un journal télévisé.

### 3.2 Les problèmes spécifiques de l'audiovisuel :

Un document audiovisuel a une particularité de la forme, l'audiovisuel est un document multimodal car il s'adresse à la vue et à l'ouïe. Cela entraîne plusieurs conséquences :

- Regarder 10 minutes de vidéo prend 10 minutes. Si l'information que je recherche est à la 10<sup>e</sup> minute, il faut que je regarde les 9 premiers pour la trouver. Il n'existe pas de moyen d'accéder directement à un contenu sinon que de le repérer explicitement et d'en noter la localisation pour être capable de s'y reporter directement. La recherche d'information nécessite l'indexation explicite du contenu ;
- La forme audiovisuelle ne mobilise pas d'unités répertoriées composant le contenu (comme des lettres de l'alphabet ) [ChFr00]. Par conséquent, il n'existe pas de moyens d'indexer efficacement l'image par image, ou le son par du son, car le problème d'associer une signification à l'image se retrouve avec celui d'attribuer une signification à l'index. Pour que cela soit possible, il faut que les images destinées à être des index soient structurées entre elles comme un code et qu'elles fonctionnent comme un système. C'est pourquoi l'explication des index recourt souvent au code linguistique. Ce n'est pas par archaïsme technologique que la plupart des centres d'archives audiovisuels recourent à la description linguistique manuelle des contenus, mais pour disposer d'un code d'index possibles pour décrire le contenu.

La tâche d'indexation audiovisuelle consiste donc à découper, dans un premier temps, le document audiovisuel à indexer selon sa dimension temporelle en un ensemble de séquences où chacune contient un objet sémantique utile auquel on attache, par la suite, une description textuelle ou multimédia[SICH02]. La recherche par le contenu dans les séquences vidéo se traduit alors par un scénario requêtes-réponses. La requête concerne les descriptions faites sur les objets sémantiques.

En retour, le résultat indique une ou plusieurs séquences correspondant à la spécification de la requête.

Différents concepts ont été élaborés afin d'analyser la structure sémantique des données audiovisuelles on présente tout de suite ses définitions et ses principales caractéristiques et l'implémentation du projet SIRSALE de ces concepts.



### 3.3 Le concept d'objectif :

Ce concept a comme but de proposer l'extraction des descriptions à partir d'une analyse du contenu. Cette analyse n'est pas une interprétation du sens du document, mais un calcul effectué à partir de ses caractéristiques physiques. Ce concept renvoie à ce que l'on appelle communément l'indexation automatique.

#### 3.3.1 Les propriétés du concept objectif :

- Il exploite l'information contenue dans le document lui-même et véhiculée par son codage. Ce codage reflète les informations physiques associées au contenu, en particulier sur la lumière, et l'on essaie de reconnaître le contenu de l'image et du flux audiovisuel à partir de là ;
- Les descriptions physiques obtenues ne reflètent pas un point de vue d'analyse : elles semblent objectives, intrinsèques au document et peuvent ainsi prétendre à une universalité valant pour les différentes utilisations possibles du contenu ;
- Les descriptions physiques restent très proches du contenu perceptif des documents audiovisuels. Elles n'apportent souvent que peu de valeur ajoutée au contenu du document vis-à-vis des utilisations envisagées. Ainsi, savoir qu'il y a telle ou telle couleur dominante, telle ou telle rupture de plan est dans la plupart des cas inintéressants;

Par conséquent, si l'indexation a pour tâche d'extraire et de constituer une information l'exploitation du contenu audiovisuel, il n'est pas possible d'affirmer que l'indexation automatique est une véritable indexation. En toute rigueur scientifique, cette indexation n'est pas de l'indexation [BacB99]. Donc quel est alors l'intérêt de ce concept d'objectif ? .

il n'est pas inutile, certes non, mais il ne peut pas servir directement à l'indexation. L'un des principaux intérêts de ce concept est d'apporter aux documents audiovisuels des outils de filtrage permettant de naviguer efficacement dans le document pour retrouver le contenu recherché[BacB99]. Les outils automatiques apportent au document une manipulabilité permettant la navigation.

### 3.4 le concept interprétatif :

Pour palier le problème de manque d'information sémantique entre les différentes vues, constaté avec le concept objectif, le concept interprétatif repose sur le fait de considérer le contenu dans le contexte de l'application escomptée du document. L'interprétation est alors contextuelle.

#### 3.4.1 Les propriétés du concept interprétatif :

- Elle mobilise les capacités interprétatives humaines et seule l'annotation manuelle permet de l'effectuer;
- elle est élaborée en fonction d'une application visée , elle produit des descripteurs proches de l'usage et donc pertinents ;

- les descripteurs concernés sont des concepts, donc abstraits par rapport à une description du contenu physique de l'image. En particulier, il n'existe pas toujours de lien direct entre le concept associé à une image et le contenu physique de celle-ci. C'est la raison pour laquelle l'interprétation humaine est indispensable[BacB99].

Ce concept permet donc une indexation plus fine et plus efficace dans la mesure où l'utilisateur s'intéresse en premier temps aux contenus sémantiques des documents vidéo.

Dans ce concept, on rencontre deux problèmes essentiels : la variabilité de la description et le rapport au contenu physique du document audiovisuel. La variabilité de la description s'explique par le fait que pour des documents similaires et un même contexte applicatif, différents indexeurs produiront différentes descriptions.

Ce problème n'est pas propre au document audiovisuel et correspond au problème général de l'indexation. Ce problème ne possède pas de solution en générale et doit être traité localement dans le cadre d'un environnement d'indexation donné. Autrement dit, les prototypes sont des méthodes qui indiquent comment reconnaître physiquement, avec l'image, l'évocation d'une pensée [BacB99]. Par conséquent, le travail interprétatif confié à l'intervention humaine reposerait sur l'identification des prototypes socioculturels, et leur traduction en méthodes de reconnaissance. Néanmoins, en dehors de cas simples facilement identifiables, ce concept introduit une complexité encore inacceptable pour le concept objectif.

### 3.5 SIRSALE en vue du concept objectif et le concept interprétatif:

Les deux concepts abordés précédemment sont implémentés dans le projet SIRSALE, afin d'aboutir à une solution pour la problématique de l'indexation audiovisuelle.

#### 3.5.1 L'approche structurelle dans le cadre de SIRSALE :

Le concept objectif est implémenté dans le projet SIRSALE sous forme d'une approche appelée «l'approche structurelle». Suivant cette approche, le document audiovisuel est subdivisé structurellement (selon sa dimension temporelle) en plusieurs *vues* atomiques dont chacune représente un concept de base. On distingue trois niveaux de vues structurelles d'un document audiovisuel :

- **plan ( *Shot* )** : Le document vidéo est divisé en plusieurs unités appelées plans. Le plan est composé d'un ensemble continu (sans coupure caméra) d'une ou plusieurs images fixes ( *frames* en anglais ). Le contenu sémantique de chaque plan peut alors être décrit individuellement[SICH02], offrant ainsi un premier niveau de découpage du document. Cependant, il faut rappeler qu'un plan peut ne correspondre qu'à une seule image, et qu'un vidéo est composé en moyenne de 25 images par seconde. Le plan devient donc une unité de découpage sémantique trop petite pour indexer des documents courants.



- **Scène ( *Scene* )** : Afin d'offrir une réelle capacité d'indexation et de recherche de contenus vidéo, il s'agit d'organiser de manière pertinente les plans. Habituellement, une structure de découpage est ajoutée au-dessus des plans. La scène peut ainsi être définie comme le premier véritable niveau sémantique. En effet, les limites du plan peuvent être clairement définies par les caractéristiques physiques des images (dominance des couleurs, etc.). Par conséquent, il n'est pas dépendant de l'interprétation d'un éventuel utilisateur. Contrairement à la scène qui demande une réelle interprétation du contenu de chaque plan afin de pouvoir les relier entre eux.
- **Séquence ( *Sequence* )** : sur des documents vidéo d'une certaine longueur ou sur des documents riches en contenus sémantiques (journaux télévisés par exemple), le découpage en scène se révèle rapidement très limitatifs. Une structure d'abstraction de plus haut niveau est donc nécessaire. La structure de la séquence permet de regrouper des scènes suivant un lieu temporel ou spatial permettant ainsi d'améliorer la recherche par le contenu dans ces documents. Enfin pour des besoins d'organisation, on peut choisir de regrouper les séquences en unités composées ( *compound units* ).

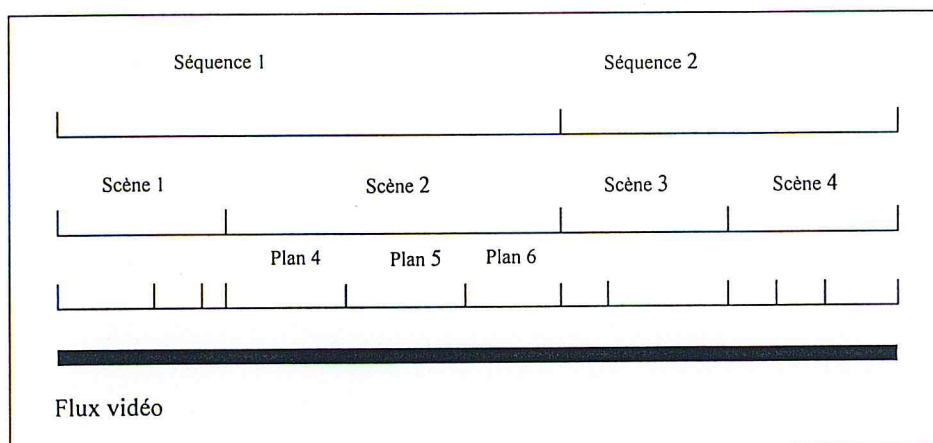


Fig. 3.1 Approche structurale

Des algorithmes permettent de définir les limites de chaque plan, ont déjà été proposés. Ils permettent de détecter les variations de couleurs ou les frames noires, caractéristiques d'une coupure caméra. En revanche, [SICH02]

### 3.5.2 L'approche par stratification dans le cadre de SIRSALE :

De la même manière, le projet SIRSALE a implémenté le concept interprétatif sous forme une approche qui s'appelle «L'approche par stratification ». Dans cette approche on divise sémantiquement le document audiovisuel, selon son contenu, en un ensemble de strates. Il s'agit donc, non d'un découpage structurel comme c'est le cas dans l'approche précédente, mais plutôt d'un découpage sémantique. Ce dernier produit un ensemble de strates définissant dans le temps des contenus sémantiques du document vidéo. Ces strates étant indépendantes de la structure de la vidéo, elles peuvent se superposer et ainsi le contenu vidéo à un instant donné peut être défini comme l'union des différentes strates.



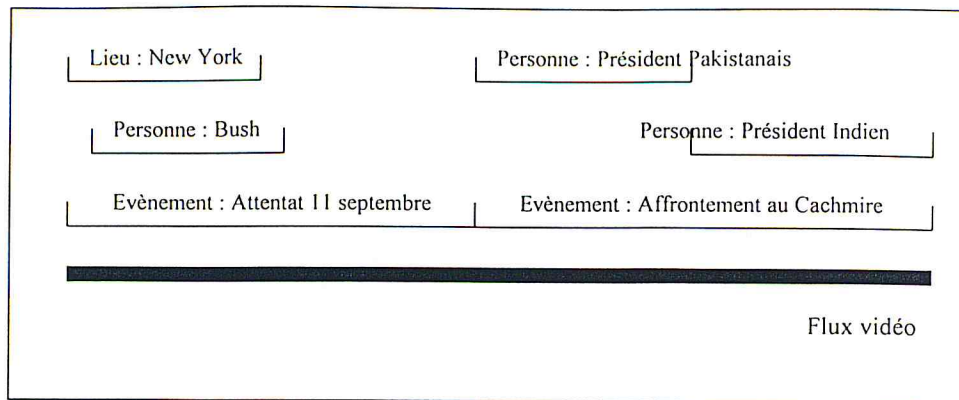


Fig. 3.2 Approche par stratification

Il faut noter aussi que le modèle de données utilisé pour décrire le contenu du document audiovisuel dépend obligatoirement du thème traité dans ce dernier. En effet, les critères de recherche par le contenu dans des documents vidéo contenant des cours à distance ne sont pas les mêmes que pour les documents de journaux télévisés par exemple.

Dans le cadre de SIRSALE deux approches ont été explorées :

- Modèle spécifique à un centre d'intérêt : l'avantage de cette solution est qu'elle offre un outil parfaitement adapté qui donne des résultats précis et facilement utilisables. En revanche son domaine d'utilisation est limité à un seul thème et il ne peut donc pas être employé dans le cas des banques de données audiovisuelles proposant des documents de différents thèmes ;
- Modèle générique : le manque intentionnel de précision dans le choix des critères de description adoptés dans ces modèles permet de les adapter à n'importe quel thème. Mais les résultats des recherches sont souvent moins précis et donc inefficace car l'indexation et la recherche d'une séquence demande une interprétation des critères d'annotation.

### 3.5.3 Approche proposée par SIRSALE (approche multi-niveaux) :

Dans le cadre de SIRSALE, une approche multi-niveaux est proposée qui permet à la fois d'exploiter la structure de la vidéo à des fins de navigation et d'utiliser l'indexation sémantique à des fins de recherche par le contenu dans les documents audiovisuels, cette approche est illustrée dans la figure suivante où les deux niveaux de description de la vidéo sont explicités.

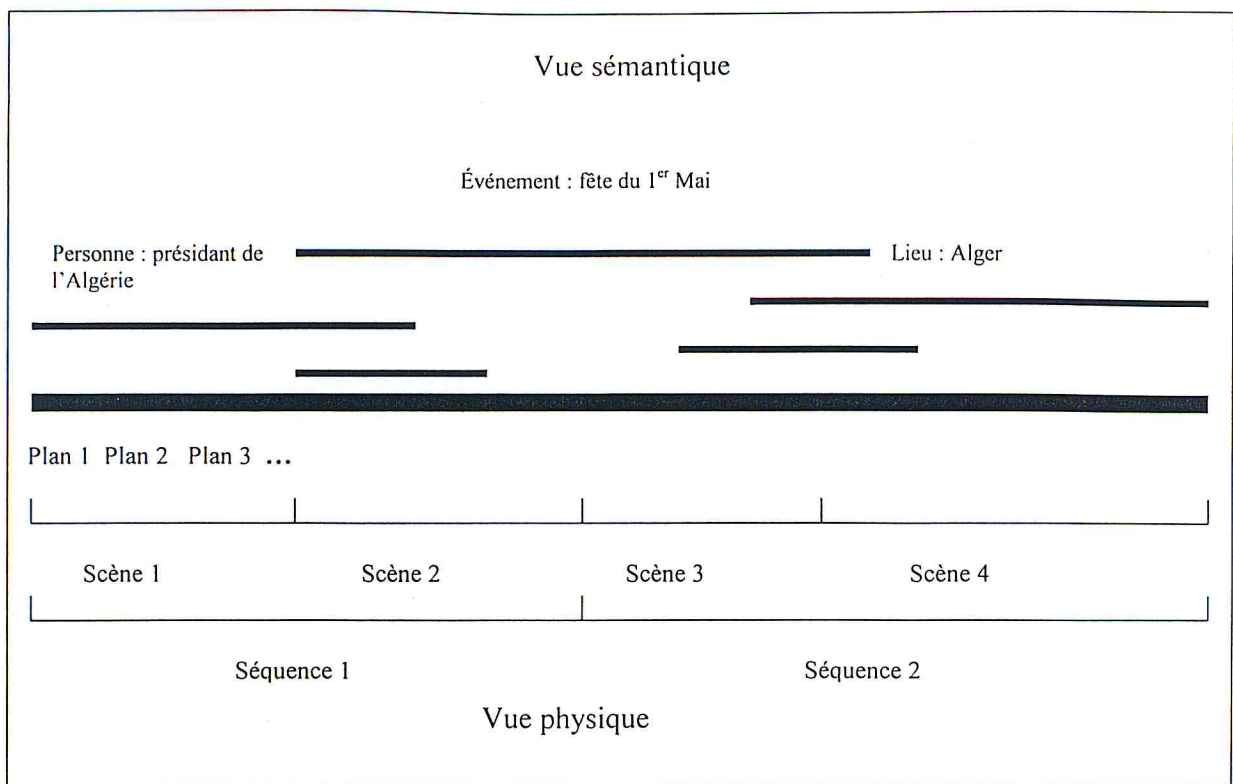


Fig.3.3 Vue multi-niveaux d'un document vidéo

### 3.6 Conclusion :

Le modèle de données du projet SIRSALE permet d'intégrer simultanément les deux niveaux d'indexation. Cela offre à l'utilisateur une solution gérante à la fois la navigation au travers des documents vidéo [SICH02] et intégrant une recherche sémantique, adaptable, optimisée pour chaque thème, sur l'ensemble de la base documentaire. L'avantage de cette proposition est une meilleure prise en compte des spécificités sémantiques de chaque thème et par conséquent la garantie d'une meilleure recherche par le contenu. En revanche, elle présente l'inconvénient d'une mise en œuvre complexe. En effet, il s'agit d'une part de supporter efficacement des contextes d'utilisation hétérogènes (journal télévisé, sport, etc.) et d'autre part de proposer des interfaces utilisateurs adaptables automatiquement en conséquence.

# chapitre:IV

## Le concept UML



## 4.1 Introduction :

Pour décrire la démarche de développement de notre logiciel, nous devons expliquer et faire comprendre les diverses fonctionnalités de notre produit, pour cela nous avons utilisé la notation UML grâce à ses avantages, et qui se base sur l'approche objet.

L'approche objet a envahi le monde du Génie Logiciel et de la conception de systèmes informatiques. Les objets sont utilisés dans les interfaces graphiques, les architectures clients/serveurs, l'Internet, etc. c'est une approche modulaire qui favorise la qualité de développement. Le développement à objets est en cours de normalisation, grâce notamment à l'OMG (Object Management Group)<sup>1</sup>, regroupement de plusieurs acteurs majeurs du marché, créé en avril 1989 à Boston. L'OMG a ainsi proposé des standards pour les SGBDO ( *Système de Gestion de Base de Données Objet*), CORBA ( *Common Object Request Broker Architecture* ) l'architecture de communication entre objets distants, et plus récemment la notation UML ( *Unified Modeling Language* ).

UML est un langage de modélisation objet. En tant que tel, il facilite l'expression et la communication de modèles en fournissant un ensemble de symboles et de règles qui régissent l'assemblage de ces symboles. Il permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation.

### 4.1.1 Historique D'UML :

UML est une notation pour la modélisation des applications construites avec des langages objets. A l'origine de cette nouvelle notation se trouve l'OMG (Object Management Group) qui partait du constat suivant :

- 📖 Les méthodes fonctionnelles ne permettaient pas d'exploiter le développement objet. Le mélange de plusieurs paradigmes n'est ni commode, ni naturel.
- 📖 Le grand nombre de méthodes n'aidaient pas le choix des utilisateurs.

Les méthodes suivantes sont à la base d'UML :

- . OMT (Object Modeling Technique) conçue par James Rumbaugh.
- . OOD (Object Oriented Design) conçue par Grady Booch.
- . OOSE (Object Oriented Software Engineering) conçue par Ivar Jacobson.

Il faut insister sur le fait qu'UML n'est pas une méthode mais une notation. Il est donc possible d'utiliser la notation UML avec une démarche de conception inspirée d'OMT. La première version d'UML est sortie le 17 janvier 1997. Entre temps des partenaires importants sont venus collaborer à la mise en œuvre de cette notation : IBM, DEC, Microsoft, Rational Rose Software, Oracle, Unisys). [ Jguy99]

<sup>1</sup>[[www.omg.org](http://www.omg.org)]



### 4.1.2 Approche Objet :

L'approche orientée objet considère le logiciel comme une collection d'objets dissociés définis par des *propriétés*. Une propriété est soit un attribut : une entité élémentaire (donnée) de la description de l'état de l'objet, ou une opération : entité élémentaire de la description du *comportement* de l'objet. Un objet comprend donc à la fois une structure de données (son état sous forme de collection d'attributs) et une collection d'opérations (son comportement).

### 4.1.3 Pourquoi une méthodologie Objet :

La technologie objet, tout en n'étant pas la panacée, offre une solution aux problèmes des systèmes informatiques d'aujourd'hui, quant à :

- 📖 La complexité du logiciel  
Le logiciel est de plus en plus intégré à l'entreprise. Il offre de plus grandes fonctionnalités. En outre, les logiciels doivent pouvoir s'interfacer et communiquer entre eux ;
- 📖 Les évolutions technologiques  
Il faut s'assurer de la pérennité de la solution mise en place. C'est à dire, développer une solution suffisamment ouverte pour incorporer les technologies de demain;
- 📖 La taille des équipes  
Il faut savoir gérer plusieurs types de compétences techniques, s'assurer que la communication sera réelle entre les différentes équipes et gérer le travail en parallèle sur une même tâche ;
- 📖 Les délais de plus en plus courts  
Les entreprises ne peuvent attendre longtemps un système qui leur permettra de gagner en productivité ;
- 📖 Les spécifications peu précises ;
- 📖 L'évolution rapide des applications.

### 4.1.4 Les concepts fondamentaux de l'approche orienté objet :

Dans cette section on va présenter quelques concepts fondamentaux sur l'approche objet.

#### Objet :

Un objet est un concept, une abstraction ou une chose ayant des limites très claires et un sens bien précis dans le contexte du problème étudié. Un objet se distingue même des autres objets ayant les mêmes caractéristiques : deux pommes de la même couleur, du même poids et de la même taille sont deux objets distincts.

Formellement un objet est l'unité formée d'un état et un comportement. L'état d'un objet est formé des valeurs instantanées de ses attributs, ainsi que le comportement regroupe les compétences de l'objet. Ce comportement est décrit par les méthodes



déclenchées par des stimulations externes appelées messages. Pour identifier les objets, chaque objet est caractérisé par un identificateur unique *oid* (Object Identifier)[DjMs03].

La définition d'objet s'appuie largement sur les principes d'abstraction et d'encapsulation.

### **Abstraction :**

L'abstraction signifie que l'on se concentre sur ce qu'est un objet et ce qu'il fait en mettant l'accent sur ses propriétés essentielles, inhérentes (dans le domaine d'application), et en ignorant les propriétés accidentelles (ce qui relève des détails d'implantations) [01].

### **Encapsulation :** (ou le *masquage de données*)

L'encapsulation est une occultation des informations contenues par un objet [Mull97], c'est-à-dire un mécanisme consistant à rassembler les données et les méthodes au sein d'une structure, en cachant l'implémentation de l'objet, et par conséquent en empêchant l'accès aux données par un autre moyen que les services proposés [Pill02]. L'encapsulation permet donc de garantir l'intégrité des données contenues dans l'objet.

### **Classe :**

La classe est la description d'un groupe d'objet ayant des propriétés similaires, et un comportement commun [Mull97]. Chaque objet membre du groupe est dit instance de la classe.

Une classe d'objets est constituée d'une partie statique *les attributs* et d'une partie Dynamique *les méthodes*.

### **Les attributs :**

Les attributs représentent la description des données propres à chaque classe d'objets.

### **Opérations et méthodes :**

Une **opération** est une action ou une transformation qu'un objet peut effectuer ou subir. L'ensemble des opérations définit le *comportement* de l'objet. Une opération est une abstraction définie par sa **signature** : le nom de l'opération, le type de valeur de retour et le nombre et les types de ses arguments. Concrètement une opération peut être implantée de manières différentes dans différentes classes. Une telle implantation est appelée **méthode**. Une opération a un objet cible comme argument implicite permettant ainsi d'identifier la méthode à appeler

### **Héritage :**

L'héritage est un mécanisme permettant de créer et de définir une sous classe par la transmission automatique des propriétés de la classe de base vers la nouvelle sous-classe. L'héritage permet un *partage hiérarchique* des propriétés (attributs et opérations). Une sous-classe (ou *classe fille*) peut incorporer, ou *hériter*, des propriétés d'une super-classe (ou *classe de base*).



**Agrégation :**

C'est un autre type de relation entre deux classes qui traduit par les relation « **Est composé de..** » ou « **Possède...** » ou encore « **a....** » [Pill02]. Relation de structuration, l'agrégation est une relation qui permet de décrire un objet composite en terme d'objets qui le constituent. Un objet ( le composé ) peut inclure un autre objet ( le composant). L'une des caractéristiques principales de l'agrégation est sa **cardinalité. 1**

**Polymorphisme :**

Le polymorphisme signifie qu'une même opération peut se comporter différemment sur différentes classes. L'opération déplacer agira de manières différentes sur un fichier, une fenêtre graphique ou un véhicule.

Le polymorphisme signifie que les différentes méthodes d'une opération ont la même signature. Lorsqu'une opération est invoquée sur un objet, celui-ci « connaît » sa classe et par conséquent est capable d'invoquer automatiquement la méthode correspondante.

**4.1.5 Quelques notions de modélisation :**

Il est préférable d'éclairer certains notions afin d'enlever toute ambiguïté dans la compréhension de notre démarche de conception.

**Modélisation :**

Le terme modélisation est souvent employé comme synonyme d'analyse, c'est-à-dire de décomposition en éléments simples, plus faciles à comprendre. En informatique, la modélisation consiste tout d'abord à décrire un problème, puis à décrire la solution de ce problème ; ces activités s'appellent respectivement : L'analyse et la conception.

**Modèle :**

Les définitions du concept modèle sont nombreuses : Un modèle tel que défini par N.Kettani [kett01] est une abstraction de la réalité qui, pour un domaine, est prise comme une représentation d'une classe de phénomènes plus ou moins habilement dégagés de leur contexte par un observateur, pour servir de support à l'investigation et/ou à la communication.

**Métamodèle :**

Un métamodèle décrit de manière formelle les éléments de modélisation ainsi que la syntaxe et la sémantique de la notation qui permet de les manipuler. Le gain d'abstraction induit par la construction d'un métamodèle facilite l'identification d'éventuelles incohérences et encourage la généralité. Le métamodèle d'UML sert de description de référence pour la construction d'outils et le partage de modèles entre outils et le partage de modèles entre outils différents[Mull01].

**Langage :**

Le Langage est un ensemble de construction qui permettent de décrire formellement les spécifications du système d'information élaboré aux différents stades du processus de conception, les éléments constitutifs d'un langage de modélisation sont : Des concepts, une syntaxe et une sémantique [Mull97].

**Formalisme :**

Ce terme est utilisé pour désigner un langage de modélisation. Il désigne un ensemble d'éléments de modélisation au qu'elle est associée une représentation souvent graphique, qui permet de manipuler aisément les modèles.

**Diagramme :**

Un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. Une représentation graphique d'une collection d'éléments de modèle, le plus souvent présenté comme un graphe connexe d'arcs [kett01]. Un diagramme n'est pas un modèle mais une représentation graphique de quelques éléments du modèle.

Le langage UML constitue une unification des méthodes objets, tirant donc profit des avantages de chacune de ces méthodes à l'origine de l'unification, il constitue un standard pour la modélisation objet.

**4.1.6 Avantages de l'UML :**

Le choix du langage de modélisation porte sur le langage UML de part le fait qu'il constitue une unification des méthodes objets, tirant donc profit des avantages de chacune de ces méthodes de l'origine de l'unification et qu'il constitue un standard pour la modélisation orientée objet. Ce choix est également justifié par plusieurs autres avantages qu'offre UML par rapport à d'autres méthodes objet et par rapport à nos besoins. Nous les décrivons dans ce qui suit :

- UML est fondé sur un métamodèle. Ce métamodèle donne une vision plus formalisée du langage et définit toutes les possibilités d'extensibilité de celui-ci.
- UML se veut un langage non fermé : les éléments de modélisation qu'il propose sont génériques, extensibles et configurable par l'utilisateur [Mull01].
- La simplicité et la capacité d'expression visuelle qu'offre l'UML, et qui permet de faciliter la communication autour des besoins entre les différents acteurs du système. UML offre une bonne communication avec les utilisateurs. Les concepts manipulés en UML correspondent à des réalités concrètes pour l'utilisateur.
- La diversité des diagrammes qu'offre UML (au nombre de neuf) et qui permet de présenter plusieurs perspectives ou vues de l'architecture du système à développer.



Le modèle conceptuel d'UML comprend les notions de base génériques du langage. Il définit trois sortes de briques de base :

1. Des Concepts (structurels, comportementaux, annotationnels, groupement).
2. Des relations (association, généralisation, les agrégations, les compositions...etc).
3. Des diagrammes (statiques et dynamiques).

Maintenant, on va présenter le langage de modélisation de données : UML de manière qu'on puisse définir les concepts fondamentaux, les différentes relations entre classes, et enfin présenter les diagrammes.

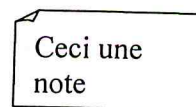
#### 4.1.7 Les concepts :

UML se veut être une notation simple, précise, et homogène, permettant un bon rendu visuel. Il décrit le réalisé plutôt que le processus de réalisation. UML support quatre (04) types de concepts :[kett01]

1. Les concepts structurels : représentés par les classes, les interfaces, les collaboration....etc.
2. Les concepts comportementaux : représentés par les interactions et les états des objets.
3. Les concepts annotationnels : représentés par les notes .

Dans UML On utilise une note pour attaché un commentaire à un ou plusieurs éléments de modélisation [Mull97].

Fig. 4.1 Représentation d'une note



4. Les concepts de groupement : représentés par les sous systèmes, les paquetages.

#### Un paquetage:

Un paquetage est un mécanisme qui permet de grouper des éléments. Il permet de structurer la spécification mais ne correspond pas à une abstraction d'un élément de conception comme le composant. Il contient des classes, des objets, des relations, des composants et les diagrammes associés. Les paquetages sont surtout utilisés pour regrouper des classes. Un paquetage peut contenir d'autres paquetages. Le paquetage est représenté par un dossier. Un paquetage peut avoir des éléments publics et des éléments privés (non visibles de l'extérieur de la classe).

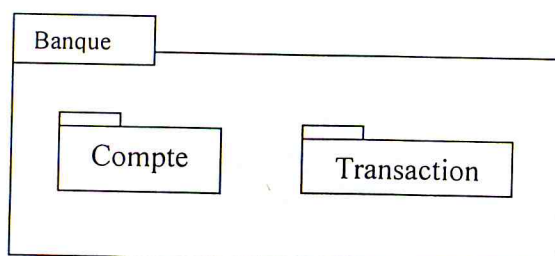


Fig.4.2 Représentation graphique d'un paquetage contenant deux paquetages



### Sous Système :

Un système est l'élément qu'on développe et pour lequel on construit des vues. Un sous système est une partie d'un système. Chaque sous système d'un système est représenté par un paquetage. La relation principale est l'agrégation mais la spécialisation est aussi autorisée. Le paquetage d'un sous système comprend trois éléments : l'interface (opération), les éléments de spécification et les éléments de réalisation[PaAl].

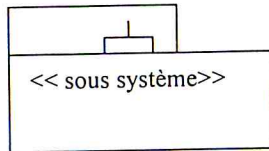


Fig.4.3 Représentation graphique d'un sous système

### Les relations :

Les relations permettent de structurer les éléments d'un modèle à objets. Ayant des objets et des classes, il trouve des relations entre objets, des relations entre classes, et des relations entre objets et classes.

### Les associations :

Les associations représentent les relations structurelles entre classe d'objets. Une association symbolise une information dont la durée de vie n'est pas négligeable par rapport à la dynamique générale des objets instances des classes associées. Une association compte au moins deux extrémités d'association, reliées à des classificateurs[Mull01].



Fig. 4.4 Une ligne entre deux classes représente une association

La plupart des associations sont binaires, c'est-à-dire qu'elles connectent deux classes. Les associations se représentent en traçant une ligne entre les classes associées.

### Les agrégations :

Une agrégation représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité. Une classe fait partie d'une autre classe, les valeurs et attributs d'une classe se propagent dans les valeurs et attributs d'une autre classe, l'agrégation ne concerne qu'un seul rôle de l'association.



Fig. 4.5 Représentation d'une agrégation

### Les compositions :

La composition est un cas particulier d'agrégation avec un couplage plus important. La classe ayant le rôle prédominant dans une composition est appelée classe composite ou classe conteneur.



Fig. 4.6 Représentation graphique de la composition

### La généralisation :

UML emploie le terme généralisation pour désigner la relation de classification entre un élément plus général et un élément plus spécifique. En fait, le terme généralisation désigne un point de vue porté sur un arbre de classification[Mull01]. Par exemple, un animal est un concept plus général qu'un chat, ou un chien. Inversement un chat est concept plus spécialisé qu'un animal.

Il existe deux types de généralisation :

- **Généralisation simple** : la relation de généralisation exprime le fait que les éléments d'une classe sont aussi décrits par une autre classe. La relation de généralisation signifie *est un*. par exemple, un chat est un animal ;

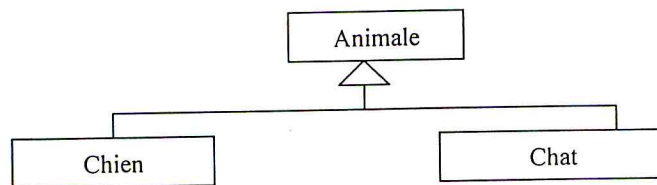


Fig. 4.7 Représentation de la relation généralisation simple entre classes

- **Généralisation multiple** : les classes peuvent avoir plusieurs super classes : dans ce cas, la généralisation est dite multiple et plusieurs flèches partent de la sous-classe vers les différentes super classes. La généralisation multiple consiste à fusionner plusieurs classes en une seule classe.

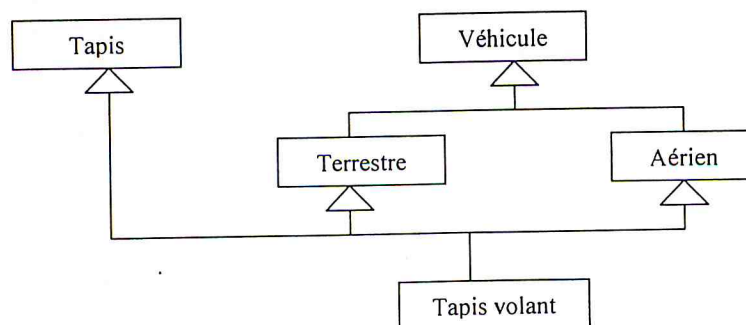


Fig. 4.8 Exemple de généralisation multiple

La classe *Tapis volant* possède deux ancêtres complètement disjoints, les classes *Tapis* et *Véhicule*.

### 4.2 Les diagrammes d'UML :

Parmi les points forts d'UML, la représentation graphique par diagrammes faciles à comprendre et à développer. Un diagramme est une représentation graphique, qui

s'intéresse à un aspect précis du modèle. De plus, il accepte plusieurs niveaux d'abstraction et utilise la sémantique du méta modèle d'UML.

#### 4.2.1 Les diagrammes de vues statiques :

Ces diagrammes sont utiles pour définir tous ce qui ne change pas dans le temps (classe, objet, paquetages...). Ces diagrammes sont à la base de la modélisation car ils représentent les aspects cellulaires du système. Ils sont donc le reflet de la modélisation en terme d'effectifs : classe, objet, et en terme de collaboration : cas d'utilisation ou uses case.

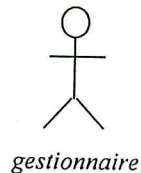
##### 4.2.1.1 Diagramme de cas d'utilisation :

Le formalisme des cas d'utilisation sert à représenter le comportement du système d'un point de vue abstrait en insistant sur les interactions avec son environnement. Il est basé sur deux concepts majeurs : *l'acteur* et *le cas d'utilisation*.

##### L'acteur :

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs des cas d'utilisation, un acteur peut représenter une personne mais aussi un système matériel, une application logicielle.

Fig. 4.9. Représentation graphique d'un acteur



##### Le cas d'utilisation : (UseCase)

Un cas d'utilisation (UC) décrit ce que fait le système en réponse à un événement provenant d'un acteur, la notion recouvre plusieurs aspects. Un UC décrit un ensemble d'actions et représente une exigence fonctionnelle du système. Un UC décrit un ensemble d'interactions entre le système et les acteurs. Nous les présentons sous forme de scénarios. Un scénario représente alors une *instance* du cas d'utilisation.

##### Scénarios :

Un scénario met en évidence les interactions qui existent entre le système et les acteurs de son contexte [PaAI]. Il illustre un cas d'utilisation.

Exemple : Le concepteur se connecte au système et donne son mot de passe, Le système vérifie l'identité du concepteur et autorise la connexion.



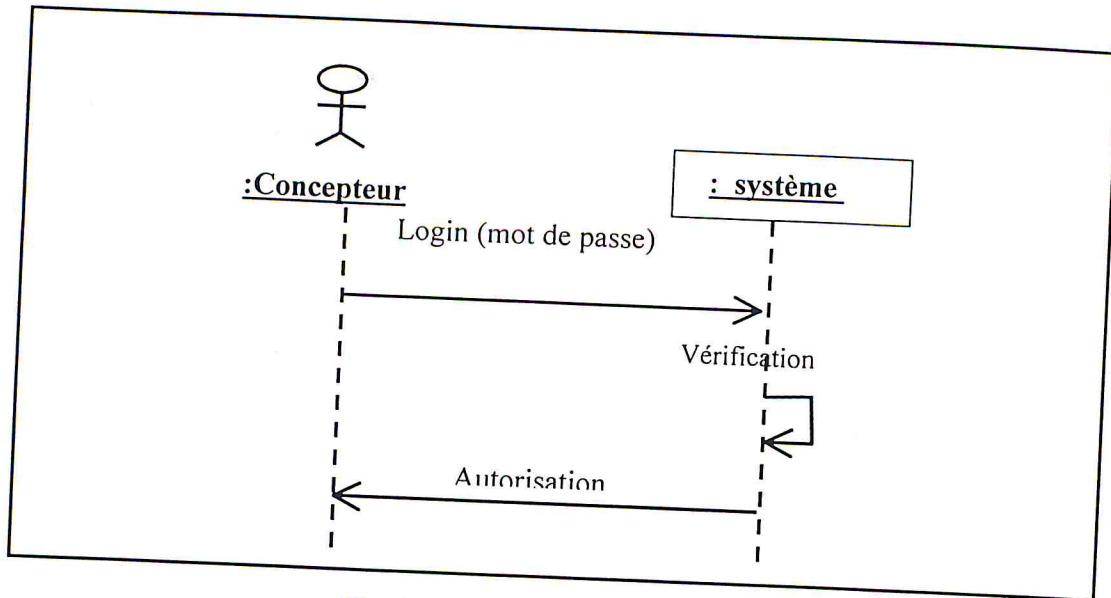


Fig. 4.10 présentation d'un scénario

**Représentation d'un diagramme de cas d'utilisation :**

Les cas d'utilisations sont représentés par des ellipses à l'intérieur desquelles figure le nom du cas d'utilisation. Ils peuvent être contenus dans un rectangle qui représente les limites du système. Les acteurs sont alors forcément à l'extérieur du rectangle puisqu'ils ne font pas partie du système.

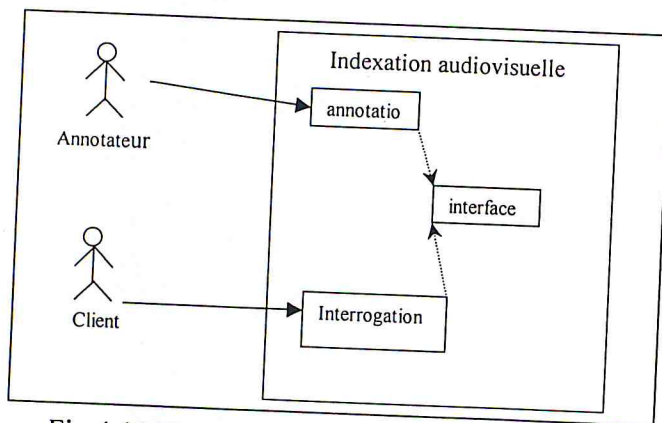


Fig.4.11 Représentation des cas d'utilisation

**Les relations entre cas d'utilisation :**

UML définit trois types de relations entre cas d'utilisation :

**a) La relation de généralisation :**

Dans une relation de généralisation entre deux cas d'utilisation, le cas d'utilisation enfant est une spécification du cas d'utilisation parent. Le cas d'utilisation parent peut être abstrait.

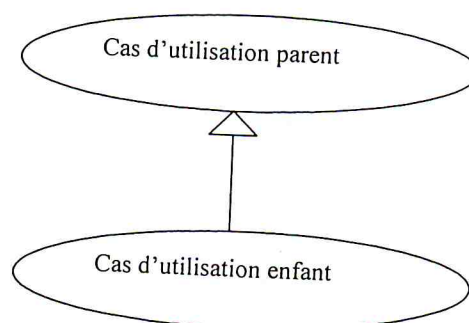


Fig.4.12 Représentation d'une relation de généralisation

### b) La relation d'inclusion :

Dans une relation d'inclusion entre cas d'utilisation, une instance du cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destination. Cette relation permet ainsi de décomposer des comportements partageables entre plusieurs cas d'utilisation.

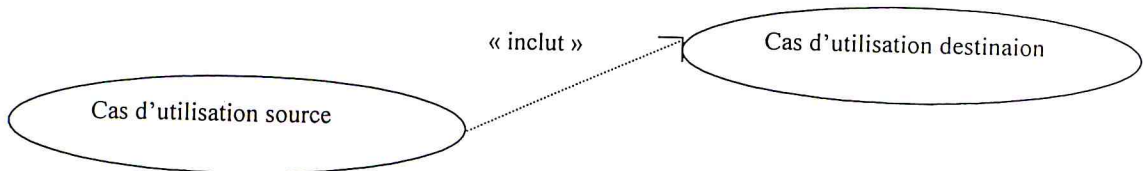


Fig.4.13 représentation de la relation d'inclusion

### c) La relation d'extension :

Dans une relation d'extension entre cas d'utilisation, le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à une condition, le comportement ajouté est inséré au niveau d'un point d'extension défini dans le cas d'utilisation destination [PaAl]. Cette relation permet de modéliser des variantes de comportement d'un cas d'utilisation selon les interactions des acteurs et l'environnement du système.

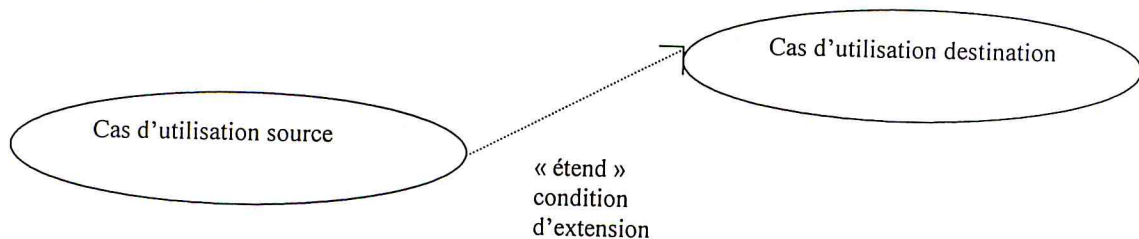


Fig. 4.14 Représentation de la relation d'extension

#### 4.2.1.2 Les diagrammes d'objets :

Les diagrammes d'objets ou diagrammes d'instances, montrent des objets et des liens, ils représentent la structure statique. Dans ces diagrammes les classes et les objets ont la même représentation. Un diagramme d'objet contient des objets, avec ou sans lien. Les objets peuvent être nommés, avec ou sans noms de classes, ou anonymes (avec un nom de classe mais sans nom d'objet). Lorsqu'un nom de classe est donné, il représente implicitement la relation d'instanciation entre l'objet et sa classe.

Exemple : Un diagramme d'objets montre une partie de la structure générale des voitures. Chaque voiture possède un moteur et quatre roues (roue de secours exclue !).

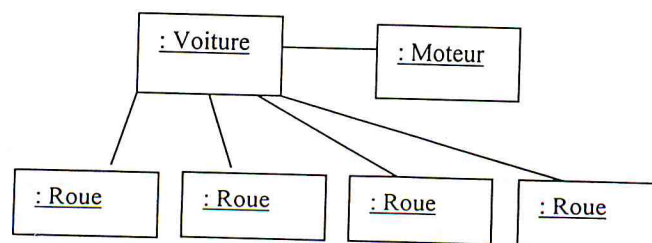


Fig.4.15 Exemple de diagramme d'objets

### 4.2.1.3 Les diagrammes de classe :

Le diagramme de classe est à la fois un élément essentiel de la modélisation à objet et la notation la plus riche parmi les diagrammes. Il exprime de manière générale la structure statique d'un système, en termes de classe et de relations entre ces classes. Ils présentent un ensemble d'interfaces et de paquetages, ainsi que leurs relations.

De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens, les objets sont instances des classes et les liens sont instances des associations. Le diagramme de classes n'exprime rien de particulier sur les liens d'un objet donné, mais décrit de manière abstraite les liens potentiels d'un objet vers d'autres objets[Mull01].

Les opérations peuvent être rangées dans des catégories, en utilisant des stéréotypes [PaAl].

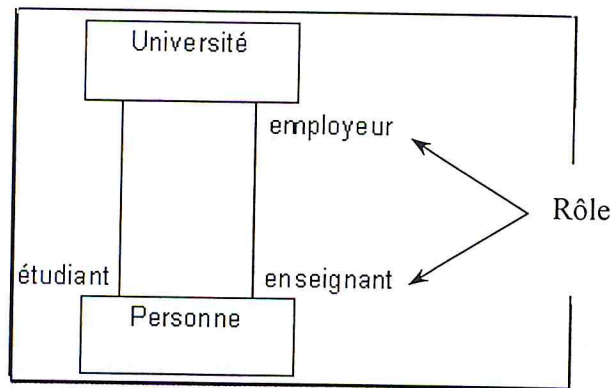


Fig.4.16 Exemple d'un diagramme de classes

#### Stéréotypes :

La notion de stéréotype est un élément clé de l'extensibilité de la notion UML et de son adaptabilité aux AGL (Atelier génie logiciel) et aux méthodes propriétaires. Chaque concept de la notation peut être spécialisé par stéréotype. Le stéréotype est une annotation, qui enrichit la description, mais n'intervient pas a priori dans les vérifications des modèles produits. Par exemple, une classe abstraite est un stéréotype de la classe.

### 4.2.1.3 Diagramme de composants :

Le diagramme de composants met en évidence la structure logicielle du système, c'est-à-dire les dépendances entre types de composants logiciels.

Un composant est une partie de l'implantation d'un système. Les composants représentent les modules de programmes sources ou compilés. Un objet, une base de données, un paquetage de classes, une bibliothèque de programmation sont des exemples de composants. Le diagramme des composants met en évidence les relations de dépendance entre les types de composants. [PaAl].



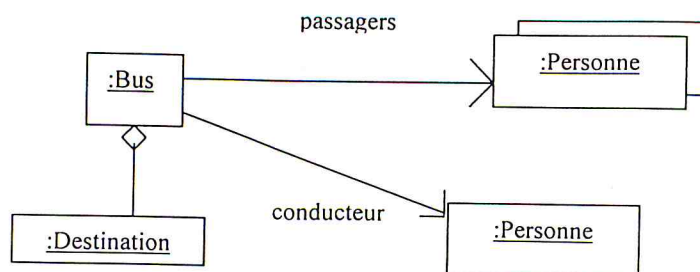


Fig. 4.17 Exemple d'un diagramme

Les relations de dépendance sont utilisées dans les diagrammes de composants pour indiquer qu'un élément d'implémentation d'un composant fait appel aux services offerts par les éléments d'implémentation d'un autre composant.

#### 4.2.1.4 Diagramme de déploiement :

Le diagramme de déploiement met en évidence la répartition des éléments de calcul ( processus, composants, objets ) sur les unités matérielles (les nœuds ). A ce niveau, on a des instances de composants et non plus des types de composants comme le diagramme de composants. Le diagramme de déploiement est un graphe dont les sommets (appelés nœuds ) sont des ressources de calcul ( processeur ou périphérique ) et les arcs (non-orientés ) sont des supports de communication ( réseau par exemple). On peut aussi représenter le contenu des nœuds en y incluant des objets ou des instances de composants.

#### 4.2.2 Les diagrammes de vues dynamiques :

Les diagrammes dynamiques d'UML représentent l'aspect dynamique des systèmes (signal, flux, synchro, messages ).

##### 4.2.2.1 Diagramme de séquence :

Un diagramme de séquence établit une vision temporelle des échanges entre objets. Les objets sont les colonnes du diagramme. Un envoi de message ( un événement ) est représenté par une flèche entre deux objets. Une séquence est une suite d'envois de messages. Un envoi de messages induit une activité chez l'objet receveur.

##### ➤ Interactions :

Les interactions modélisent un comportement dynamique entre objets [Mull97]. Elles se traduisent par l'envoi de messages entre objets. Un diagramme de séquence représente une interaction entre objets, en insistant sur la chronologie des envois de messages [BerF02].

##### ➤ Les messages :

Un message est une requête adressée à un objet demandant l'exécution d'une méthode. Un message comprend un objet destinataire, appelé le receveur, un nom de méthode, appelé sélecteur, les paramètres, et parfois une continuation, un objet auquel est transmis le résultat [PaAl]. Un envoi de message est une invocation d'opération, dans laquelle certains arguments désignent le ou les receveurs. Les messages échangés sont représentés au moyen de flèches horizontales partant de l'émetteur vers le récepteur. L'ordre de l'envoi est donné par la positions sur l'axe [FaHe00].

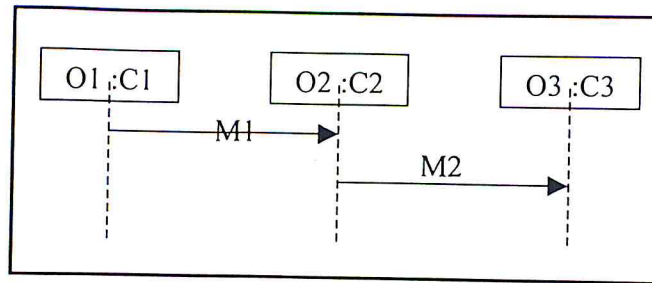


Fig.4.18 Agencement de messages

Le diagramme de séquence distingue 5 types de messages prédéfinis qui sont présentés dans le tableau suivant :

Type de message	La signification
	Message simple, exemple : à une passation de contrôle en mono tâche.
	Message asynchrone, pas de réponse attendue par l'émetteur.
	Message synchrone, réponse nécessaire du destinataire
	Message dérobant, le destinataire doit être à l'écoute
	Message minuté, l'émetteur est bloqué pendant un laps de temps.

Fig.4.19 Les différents types de messages

➤ **Période d'activation**

Correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet

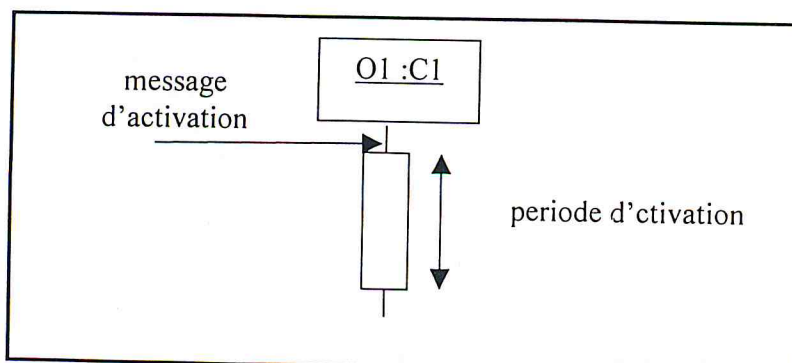
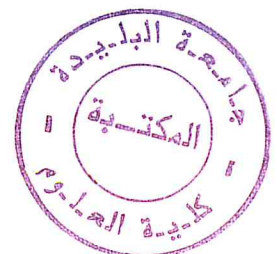


Fig. 4.20 Activation d'un objet de manière simple



#### 4.2.2.2 Diagramme de collaboration :

Un diagramme de collaboration est un diagramme d'objet qui met en évidence une vision spatiale des échanges entre objets. Le support de la collaboration est un réseau d'objets interconnectés (objet et liens)[PaAl]. Les diagrammes de collaboration présentent un ensemble de rôles joués par des objets dans un contexte particulier, ainsi que les liens entre ces objets. Ils montrent également des interactions entre ces objets à travers la représentation d'envoi de messages. Ils insistent plus particulièrement sur la structure spatiale qui permet la mise en collaboration d'un groupe d'objets.

La figure suivante présente le formalisme de base d'un diagramme de collaboration : un échange de message entre deux objets.

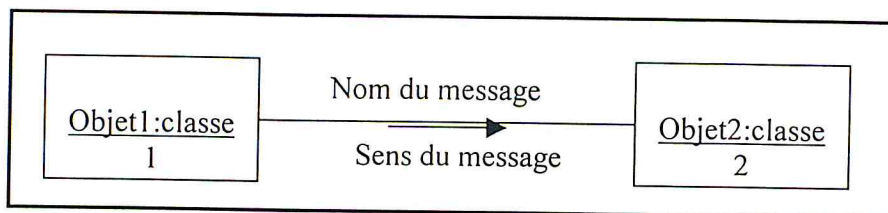


Fig. 4.21 Formalisme de base du diagramme de collaboration

#### 4.2.2.3 Les diagrammes d'états transitions :

Un diagramme d'état-transition est un graphe constitué de nœuds représentant des états ainsi que des flèches représentant des transitions, portant des paramètres et des noms d'événements [FaHe00]. Les diagrammes d'états permettent de définir le comportement d'un objet particulier vis-à-vis des sollicitations internes ou externes auxquelles il peut être soumis [GabJ98]. Ils permettent aussi de décrire l'évolution dans le temps les états des objets d'une certaine classe, les événements auxquels ils réagissent et les transitions qu'ils effectuent.

Les diagrammes d'états visualisent des automates (Figure suivante) d'états finis, du point de vue des états et des transitions [Mull97].

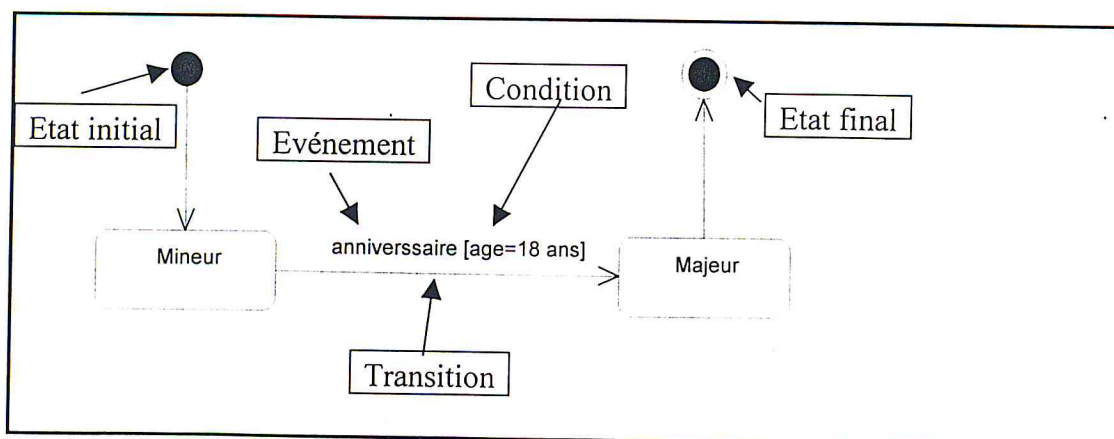


Fig.4.22 Exemple de diagramme d'état-transition



### ➤ Points d'exécution des opérations

Il existe cinq points pour spécifier les opérations qui doivent être exécutées[Muller01]. Ces points sont dans l'ordre d'exécution :

- L'action associé à la transition d'entrée (op1).
- L'action d'entrée d'état (op2).
- L'activité dans l'état (op3).
- L'action de sortie d'état (op4).
- L'action associée à la transition de sortie de l'état (op5).

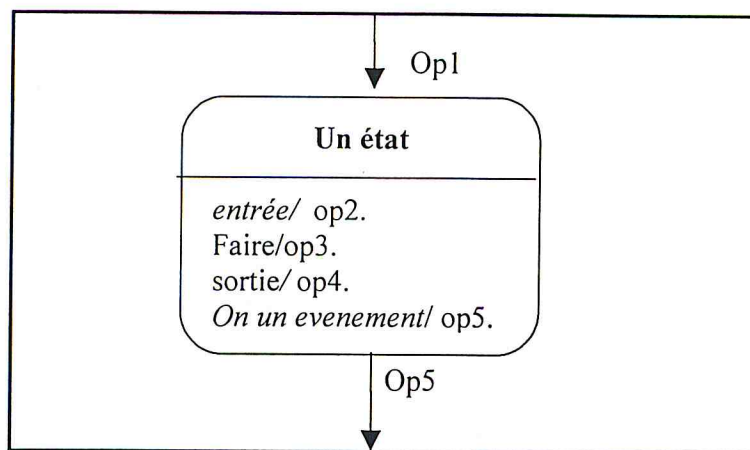


Fig.4.23 Les points d'exécution pour un état [Mul, 97]

#### 4.2.2.4 Diagramme des activités :

Un diagramme d'activités est une variante des diagrammes d'états transition. Dans un diagramme d'états transitions, les états et les transitions sont mis en avant alors que dans un diagramme d'activités, ce sont les activités et les transitions qui sont mises en avant. Les deux types de diagramme permettent ainsi d'avoir deux vues différentes sur des automates donnés. Un diagramme d'activité visualise un graphe d'activité qui modélise le comportement interne d'une méthode ( la réalisation d'une opération ). D'un cas d'utilisation[Mull01].

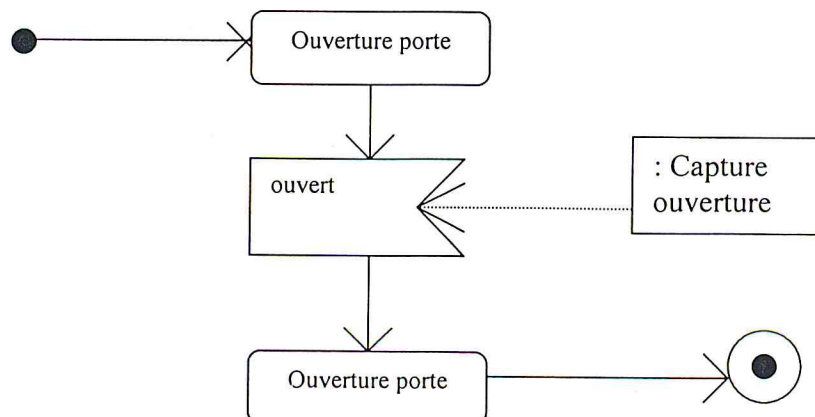


Fig.2.24 Diagramme d'activités, ouvrir porte –ascenseur[PaAl]

### 4.3 Les points forts d'UML :

1. Sa notation graphique permet d'exprimer visuellement une solution objet;
2. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions ;
3. Son aspect visuel facilite la comparaison et l'évaluation de solutions ;
4. Sans indépendance (par rapport aux langages d'implémentation, domaine d'application, ...) en font un langage universel ;
5. Tient compte les besoins réels du système en intégrant le *cas d'utilisation* qui est une étude préalable pour identifier le but exact du système à modéliser ainsi que ses besoins ( humaines ou matériaux ).

### 4.4 Conclusion :

UML dispose d'un nombre non négligeable d'éléments, il collecte tous les éléments de modélisation [PaAI]. Ces éléments sont rangés dans des diagrammes traitant l'aspect complémentaire des systèmes informatique. Ce noyau couvre l'essentiel des besoins de modélisation à objets, donc UML est suffisamment général pour être employé dans tous les domaines informatiques.

# Chapitre 5: démarche de développement



## 5.2 Introduction :

L'indexation audiovisuelle, permet de fournir des bénéfices merveilleux d'exploitation dans les documents audiovisuels. On trouve plusieurs domaines à indexer, nous allons travailler sur le domaine des journaux télévisés. Ce domaine nécessite un travail sérieux pour implémenter un outil qui permet de satisfaire les besoins des utilisateurs (journalistes... etc.).

Dans ce chapitre nous présentons notre conception qui est basé sur la démarche de développement d'un logiciel qui permet d'indexer des journaux télévisés, cette indexation se divise en deux parties, la première partie permet d'annoter les journaux et la deuxième permet l'interrogation des journaux .

Pour développer un logiciel, il est nécessaire de passer par plusieurs étapes, cependant ce chemin rencontre plusieurs difficultés (crise de génie logiciel). Pour cela des méthodes de développement de logiciel ont été définies, avec les quelles on peut organiser nos objectifs et nos travaux d'une façon compréhensible, afin de minimiser la complexité des applications et d'exprimer les concepts logiciels d'une façon plus simple.

Avant de commencer, il faut citer les étapes nécessaires pour faire le développement d'un logiciel qui sont :

- Analyse ;
- Conception ;
- Implémentation ;
- Test ;
- Validation.

### 5.2 Choix de la démarche :

Une méthode de développement de logiciel est définie pour représenter le processus de travail, elle comprend [Mull97] :

1. Des éléments de modélisation qui sont les briques conceptuelles de base.
2. Une notation dont l'objectif est d'assurer le rendu visuel des éléments de modélisation.
3. Un processus qui décrit les étapes à suivre lors du développement du système.

La notation UML couvre le cycle de vie d'un logiciel depuis l'analyse des besoins jusqu'au test. UML est une notation très riche, où toutes les phases du développement sont couvertes, tout d'abord :

- Les besoins du futur système concernant les utilisateurs sont traduits par les cas d'utilisations.
- La spécification complète du système s'exprime à l'aide des diagrammes afin de couvrir le profil statique et dynamique du système.
- Les testes sont utilisés pour vérifier que l'implémentation candidate est effectivement conforme à la spécification.

Rappelons que l'UML est une notation qui représente un langage de modélisation et non pas une méthode objet [Mull97], ça veut dire que l'UML ne décrit pas une

démarche de développement de logiciel. Nous avons suivi comme cycle de vie, le modèle « en cascade » comme processus de développement.

Ce modèle est décrit par Royce en 1970, qui a été largement employé depuis, pour la description générale des activités liées aux logiciels [Mull97].

Le modèle en cascade présente un cycle de vie d'un logiciel par une suite de phases qui s'enchaînent dans un déroulement linéaire (fig. 5.1), depuis l'analyse des besoins jusqu'à la maintenance [Mull97].

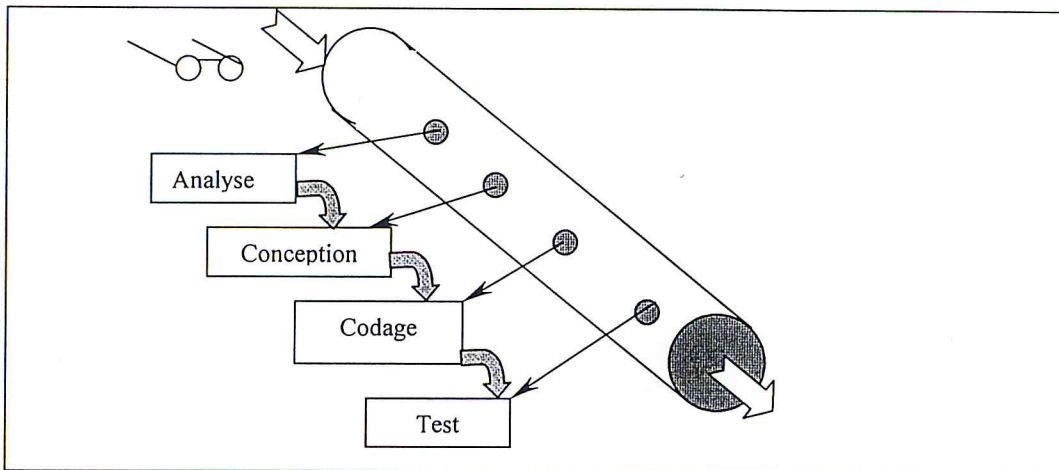


Fig. 5.1. Le modèle de développement en cascade ouvre des point de visibilité sur le processus de développement

### 5.3 Spécification des besoins :

Cette étape a pour objectif de répondre à des questions de type « quel sont les fonctions du système ? », « Les utilisateurs du système qui sont ils ? » « Qu'attendent ils du système ? ». Ces questions doivent trouver la réponse à la fin de cette étape par étude du comportement du système comme des cas d'utilisations, le contexte du système les acteurs et les scénarios.

#### 5.3.1 Cas d'utilisation :

La spécification détermine le quoi faire, il s'agit de définir les besoins de l'utilisateur. L'expérience montre que la technique des cas d'utilisation (Use Case ) se prête à la détermination des besoins d'utilisateurs [Mull97].

Tout d'abord , commençons par la reconnaissance des acteurs.

#### Acteur :

Les acteurs sont par définition des éléments qui interagissent avec le système, La réponse de la question « *Les utilisateurs du système qui sont ils ?* » Se traduit par la délimitation des acteurs ? .

#### a) Annotateur :

C'est l'acteur principal, qui représente un utilisateur du système, il est chargé d'annoter des séquences vidéo à partir des documents audiovisuels d'une façon subjective, puisque chaque annotateur traduit un événement à sa propre vision.

L'annotateur représente les personnes qui annotent les journaux télévisés par exemple : journaliste, archiviste, documentaliste, ...etc.

### b) Interrogateur :

C'est un acteur secondaire, il ne manipule pas notre système mais il a un rôle de consultation. Un interrogateur peut être : journaliste, documentaliste, ..etc.

### 5.3.2 Les cas d'utilisation du système :

Notre système offre plusieurs fonctionnalités qui présentent plusieurs cas d'utilisations. L'utilisation des diagrammes de cas d'utilisations fournies par le langage UML permet d'organiser ces fonctionnalités qui vont être offertes aux utilisateurs par notre futur système, brièvement nous avons trois cas d'utilisation principaux, par la suite nous allons présenter chaque cas d'utilisation avec l'explication détaillée :

#### 5.3.2.1 Les cas d'utilisation principaux :

Nous allons présenter notre système en trois cas d'utilisation principaux qui sont à la suite :

1. Cas d'utilisation « identification du Journal Télévisé » ;
2. Cas d'utilisation « Annotation du Journal Télévisé » ;
3. Cas d'utilisation « Interrogation ».

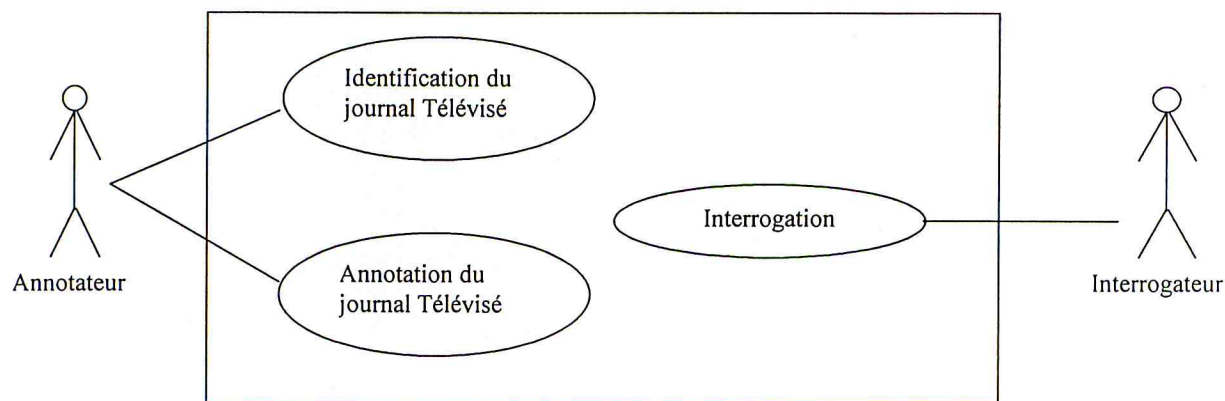


fig.5.2 cas d'utilisation principaux du système

#### Note :

par la suite, nous avons expliqué chaque partie du système avec un diagramme de cas d'utilisation particulier, afin de comprendre le diagramme de cas d'utilisation général.

#### 5.3.2.1.1 Cas d'utilisation « identification du Journal Télévisé » :

L'acteur principal (Annotateur) se charge d'identifier le journal, cette identification consiste à identifier les différents composants d'un journal télévisé, ses composants sont : les animateurs, les invités, le type du journal (exp. 7<sup>h</sup> du matin, principal, ...etc.) et la date du journal.



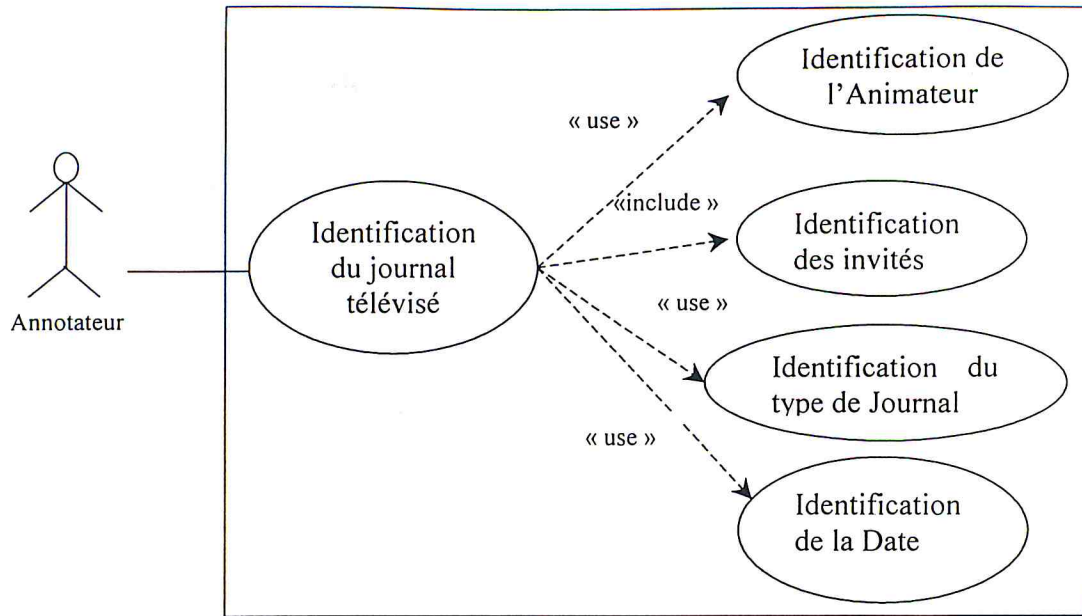


Fig.5.3. Cas d'utilisation « identification du Journal Télévisé »

Parmi les caractéristiques d'un journal télévisé, c'est l'Animateur qui fait la présentation des événements du journal, alors un journal peut être identifié par un tel Animateur.

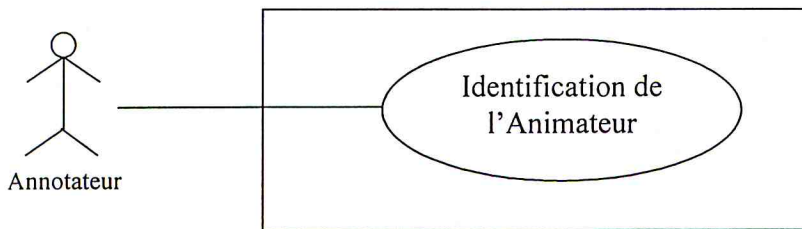


Fig.5.4.Cas d'utilisation « identification de l'animateur »

**Cas d'utilisation« Identification des Invités » :**

Un journal peut avoir des invités, l'identification de ces derniers est représentée par un cas d'utilisation « identification des invités ».

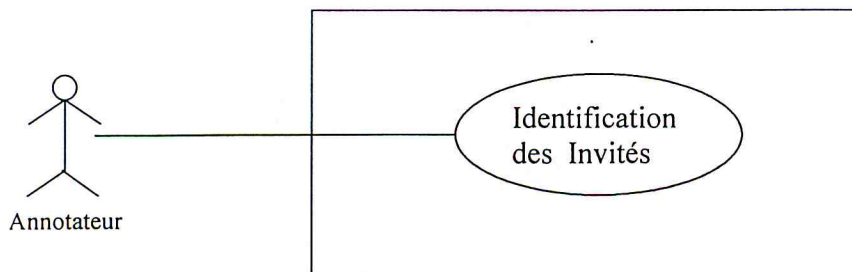


Fig.5.5 cas d'utilisation « Identification du invités »

**Cas d'utilisation« Identification de type du journal télévisé » :**

il existe plusieurs types de journaux télévisés qui sont en générale :

1. journal principal (Journal de 8<sup>h</sup> du soir ) ;
2. journal auxiliaire (Journal de matin, de minuit,..etc.) ;
3. journal spécial (journal d'été , événement urgent,..etc.).

Pour cette raison nous avons utilisé le cas d'utilisation « identification de type du journal télévisé ».

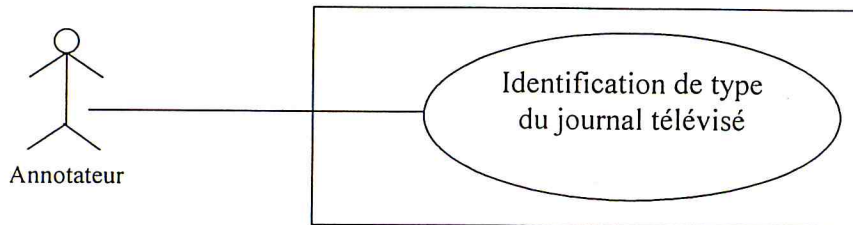


Fig.5.6 Cas d'utilisation« Identification de type du Journal Télévisé»

#### Cas d'utilisation « Identification de la date» :

Chaque journal télévisé peut être localisé avec certaines informations, par mis les quelles « la Date », le cas d'utilisation « identification de la date » est présenté pour ce rôle.

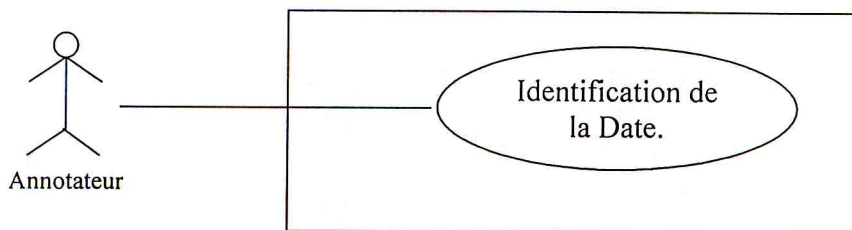


Fig.5.7 Cas d'utilisation« Identification d Identification de la Date ».

#### 5.3.2.1.2 Cas d'utilisation « Annotation » :

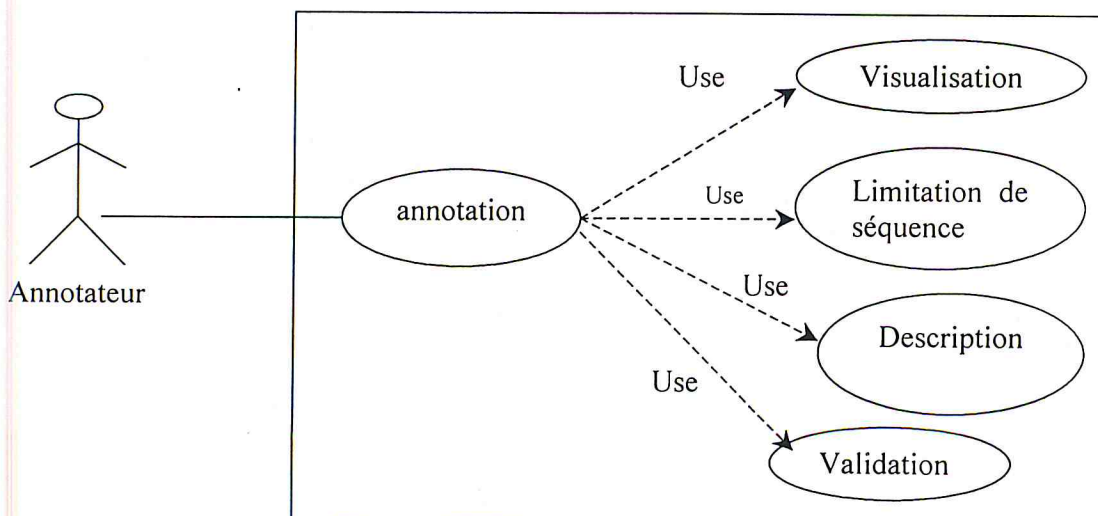


Fig.5.8 Cas d'utilisation « Annotation »

Le cas d'utilisation appelé « annotation » utilise trois autres cas d'utilisation qui sont : « visualisation », « limitation des séquences », « Description » et enfin « validation ».

#### Cas d'utilisation « Visualisation » :

Pour commencer le processus d'annotation, l'annotateur doit tout d'abord demander au système de visualiser le document multimédia (Journal télévisé).

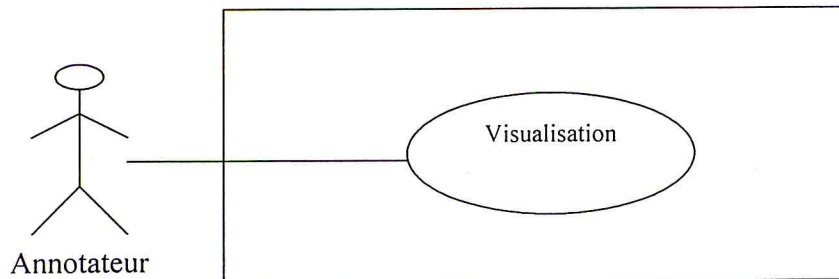


Fig.5.9 Cas d'utilisation « Visualisation »

#### Cas d'utilisation « Limitation de séquence » :

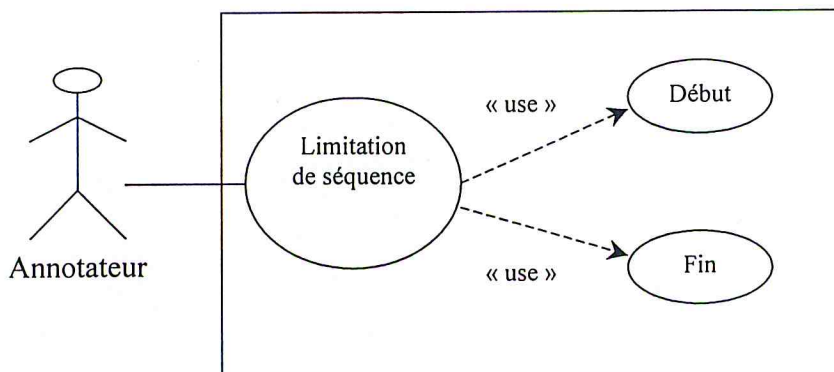


Fig.5.10 Cas d'utilisation « limitation des séquences »

Dans ce cas d'utilisation, l'annotateur choisit une séquence, et pour cela il doit délimiter la séquence vidéo par un délimiteur de début et de fin, chaque séquence doit comporter deux bornes celle de début et de fin.

Alors la relation entre la limitation des séquences et le début et la fin de la séquence est une relation d'utilisation représentée par « use ».



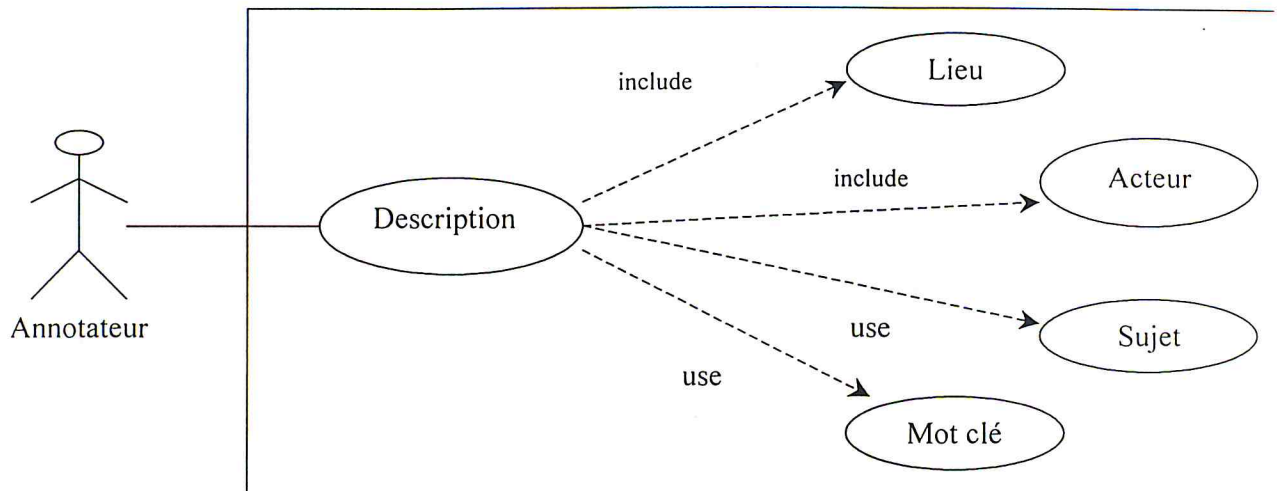
**Cas d'utilisation « Description » :**

Fig. 5.11 Cas d'utilisation « Description »

Pour que l'annotateur puisse faire une description subjective par rapport à une séquence vidéo il doit modéliser ses idées selon quatre cas d'utilisation inclus dans le cas d'utilisation « Description ».

C'est-à-dire un évènement peut avoir un *lieu*, et ce dernier englobe un cas d'utilisation celui de « définition du plateau » (un plateau veut dire la structure où l'évènement se déroule), et utilise deux autres cas d'utilisation celle de « définition du pays » et « définition de la région ».

Le deuxième cas d'utilisation est celui de « définition de l'acteur », l'annotateur peut détecter le ou les acteurs de l'évènement et cet acteur peut être individu comme il peut être une entité logique (association, group, ...etc.), donc la relation entre le cas d'utilisation « description » et « définition d'acteur » est une relation d'inclusion « include ».

Le troisième cas d'utilisation est celui de « définition du sujet » et dans ce cas d'utilisation l'acteur principal « Annotateur » résume l'évènement dans quelques mots décrivant le contenu de la séquence appropriée. Rappelons que la relation entre le cas d'utilisation « description » et le cas d'utilisation « définition de sujet » est une relation d'utilisation représentée par « use », chaque évènement annoté doit avoir un identifiant du sujet.

Le quatrième cas d'utilisation est « l'identification des mots clé » afin d'améliorer la recherche, l'annotateur attribut à l'évènement des mots clé caractérisant l'évènement à travers ces profils principaux, chaque évènement annoté possède ses propres mots clé c'est-à-dire la « description » utilise « l'identification des mots clé ». La relation est représentée par une relation « use ».

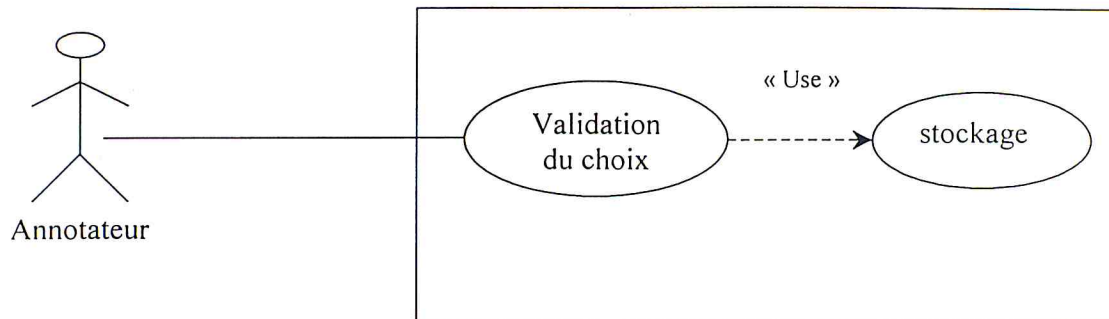
**Cas d'utilisation «Validation du choix»**

Fig.5.12 Cas d'utilisation « Validation du choix »

Après l'ajustement des bornes de la séquence et remplissage des informations concernant le cas d'utilisation «description », l'annotateur valide les informations de l'évènement, le cas d'utilisation Validation du choix se charge de stocker ces informations.

L'indexation audiovisuelle engendre une autre dimension pas moins d'importante que l'annotation, c'est l'interrogation, il existe plusieurs façons d'interrogation.

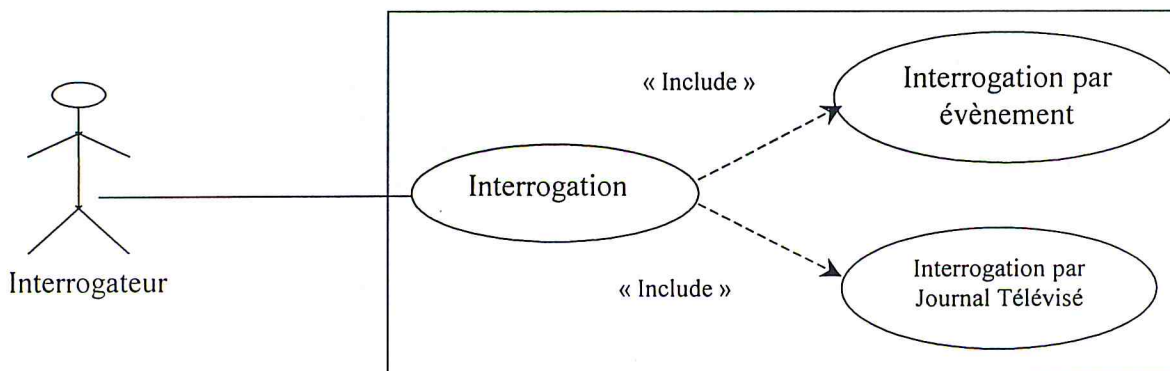
**5.3.2.1.3 Cas d'utilisation « Interrogation » :**

Fig.5.13 Cas d'utilisation « Interrogation »

Le cas d'utilisation interrogation contient deux autres cas d'utilisation qui sont « interrogation par évènement » et « interrogation par journal télévisé ». La relation entre le cas d'utilisation « interrogation » et les deux autres cas d'utilisation est représentée par « include ». Rappelons que l'acteur de ce cas d'utilisation est l'interrogateur et il joue un rôle de consultation.

**a) Cas d'utilisation « Interrogation par Evènement » :**

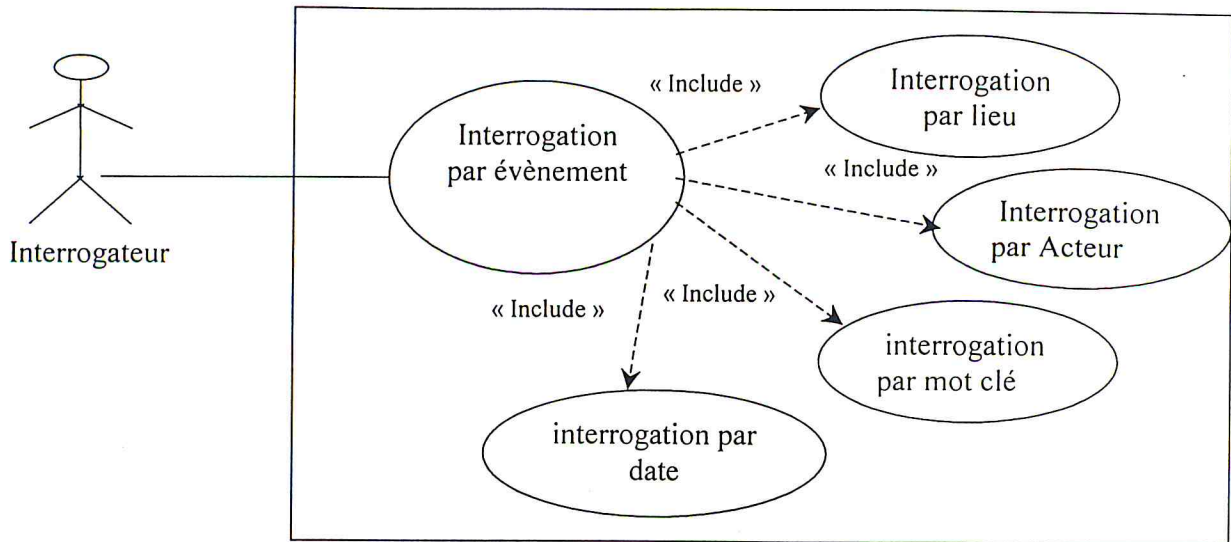


Fig.5.14 Cas d'utilisation « Interrogation par Evènement »

L'interrogateur peut interroger le système à travers quatre cas d'utilisation « interrogation par lieu », « interrogation par acteur », « interrogation par mot clé » et « interrogation par date », la relation est représentée comme « include » entre les quatre derniers cas d'utilisations et le cas d'utilisation « interrogation par évènement ».

Commençons à détailler les choses, prenons le cas 'utilisation « l'interrogation par lieu ».

**Cas d'utilisation « Interrogation par lieu » :**

Ce cas d'utilisation « interrogation par lieu » présente à l'interrogateur la possibilité d'interroger le système par rapport le lieu à travers le pays ou la région ou le plateau.

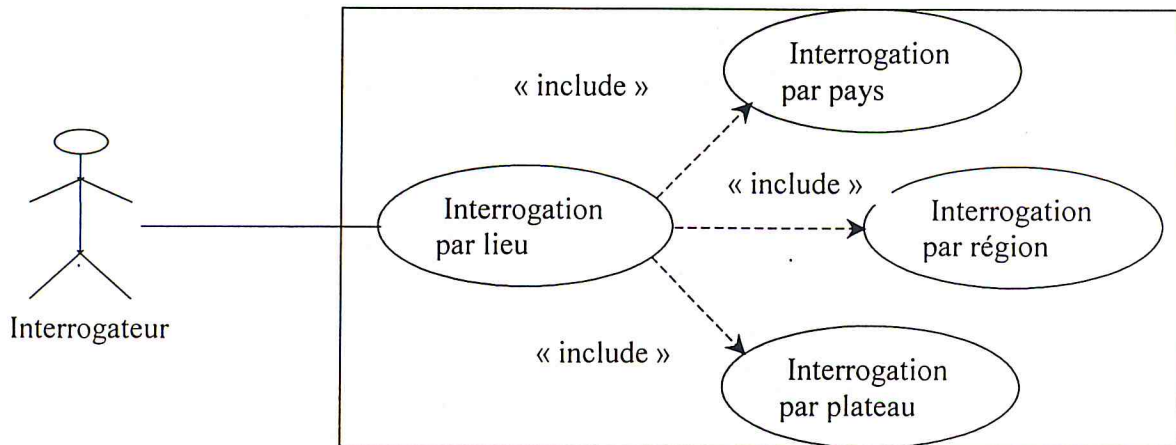


Fig. 5.15 Cas d'utilisation « Interrogation par lieu »

La relation entre le cas d'utilisation « interrogation par lieu » et les trois autres cas d'utilisation est une relation d'inclusion .



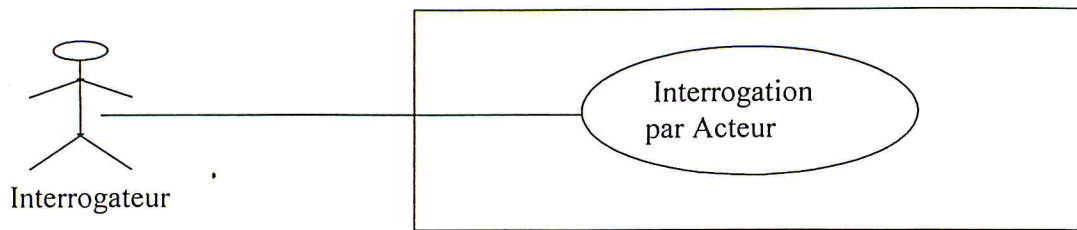
**Cas d'utilisation « Interrogation par Acteur » :**

Fig.5.16 Cas d'utilisation « Interrogation par Acteur »

L'interrogateur peut interroger le système par Acteur (l'acteur de l'évènement).

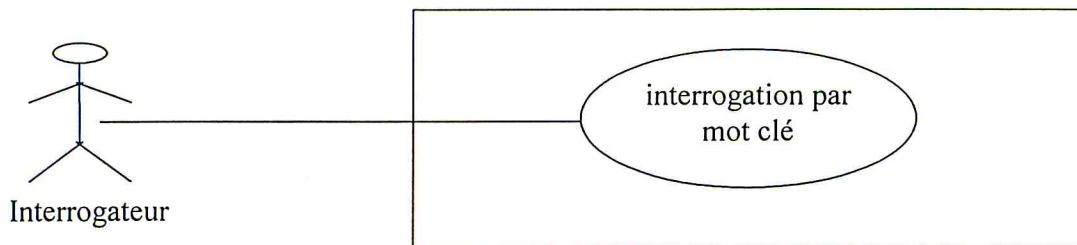
**Cas d'utilisation « Interrogation par mot clé » :**

Fig.5.17 Cas d'utilisation « Interrogation par mot clé »

Les mots clé que l'annotateur a mis au paravent dans le cas d'utilisation « identification par mot clé » peuvent être utilisés comme un outil d'accès à des séquences audiovisuelles, où l'interrogateur peut interroger le système à travers ces mots clé.

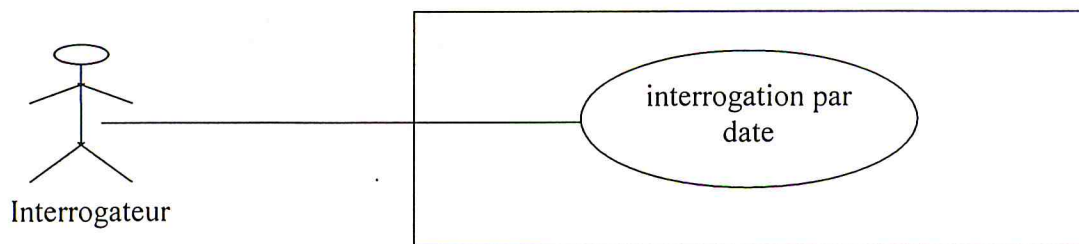
**Cas d'utilisation « Interrogation par date » :**

Fig.5.18 Cas d'utilisation « Interrogation par mot clé »

Ce cas d'utilisation présente la possibilité d'interroger le système par date d'évènement. Le prochain cas d'utilisation représente l'interrogation par journal télévisé c'est le deuxième cas d'utilisation principal dans l'interrogation.

**b) Cas d'utilisation « Interrogation par Journal Télévisé » :**

Ce cas d'utilisation présente une interrogation par journal télévisé, le résultat de cette requête n'est pas une séquence vidéo mais un journal télévisé complet.

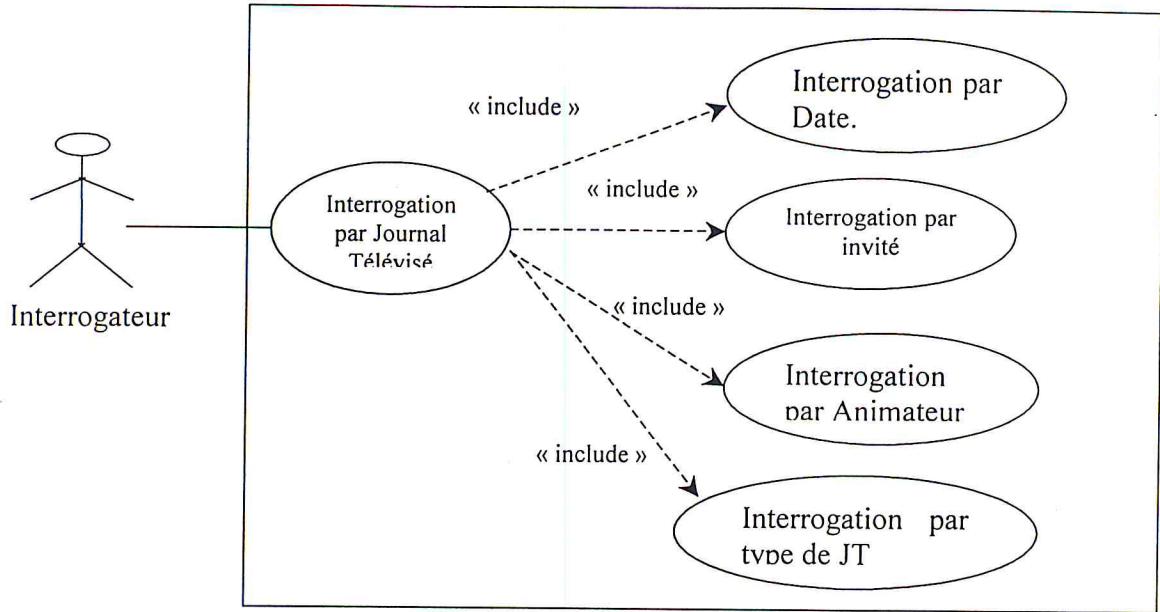


Fig. 5.20 Cas d'utilisation « Interrogation par Journal Télévisé »

Le cas d'utilisation « interrogation par Journal Télévisé » contient quatre sous cas d'utilisation : « interrogation par invités », « interrogation par animateur », « interrogation par date » et « interrogation par type Journal ».

**Cas d'utilisation « Interrogation par invité » :**

Ce cas d'utilisation offre la possibilité d'interroger le système en utilisant les (le) noms des invités .

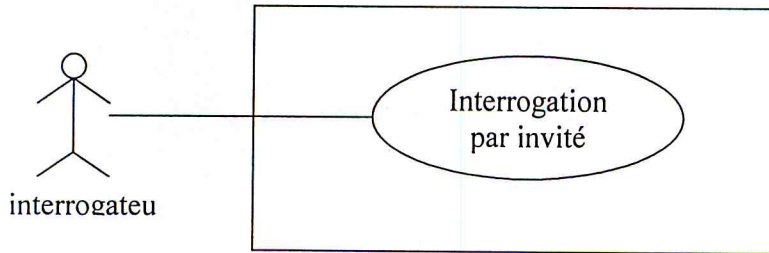


Fig. 5.21 Cas d'utilisation « Interrogation par personnalité »

**Cas d'utilisation « Interrogation par animateur » :**

De la même façon l'interrogation est possible par animateur du journal.

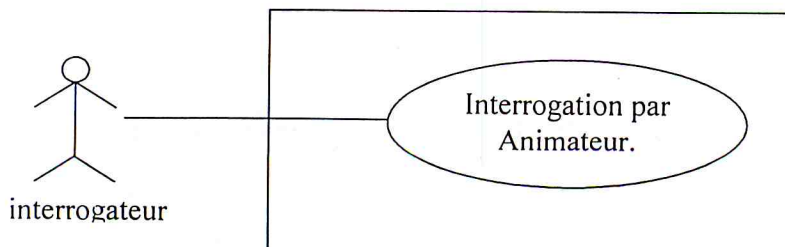


Fig. 5.22 Cas d'utilisation « Interrogation par animateur ».

**Cas d'utilisation « Interrogation par Date » :**

Chaque journal télévisé a une date, ce cas d'utilisation offre la possibilité d'interroger le système en utilisant cette propriété.

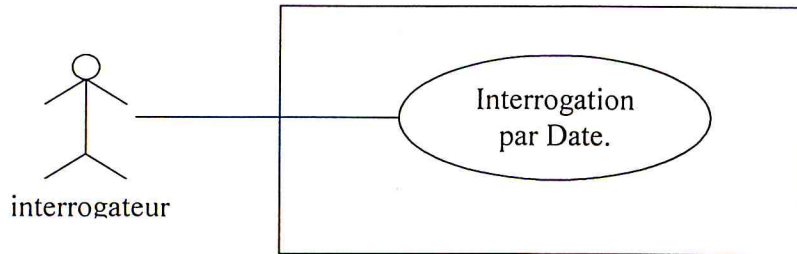


Fig.5.23 Cas d'utilisation « Interrogation par Date ».

**Cas d'utilisation « Interrogation par Nom-JT » :**

Le nom d'un journal télévisé est utile dans l'interrogation de notre système.

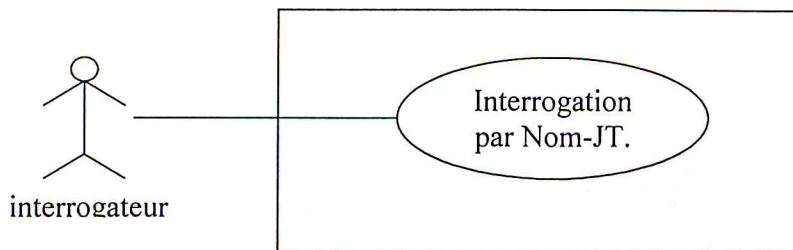


Fig.5.24 Cas d'utilisation « Interrogation par Nom-JT ».

Maintenant que nous avons cerné tous les cas d'utilisation possibles du système, le diagramme de cas d'utilisation suivant montre les diverses fonctionnalités fournies.



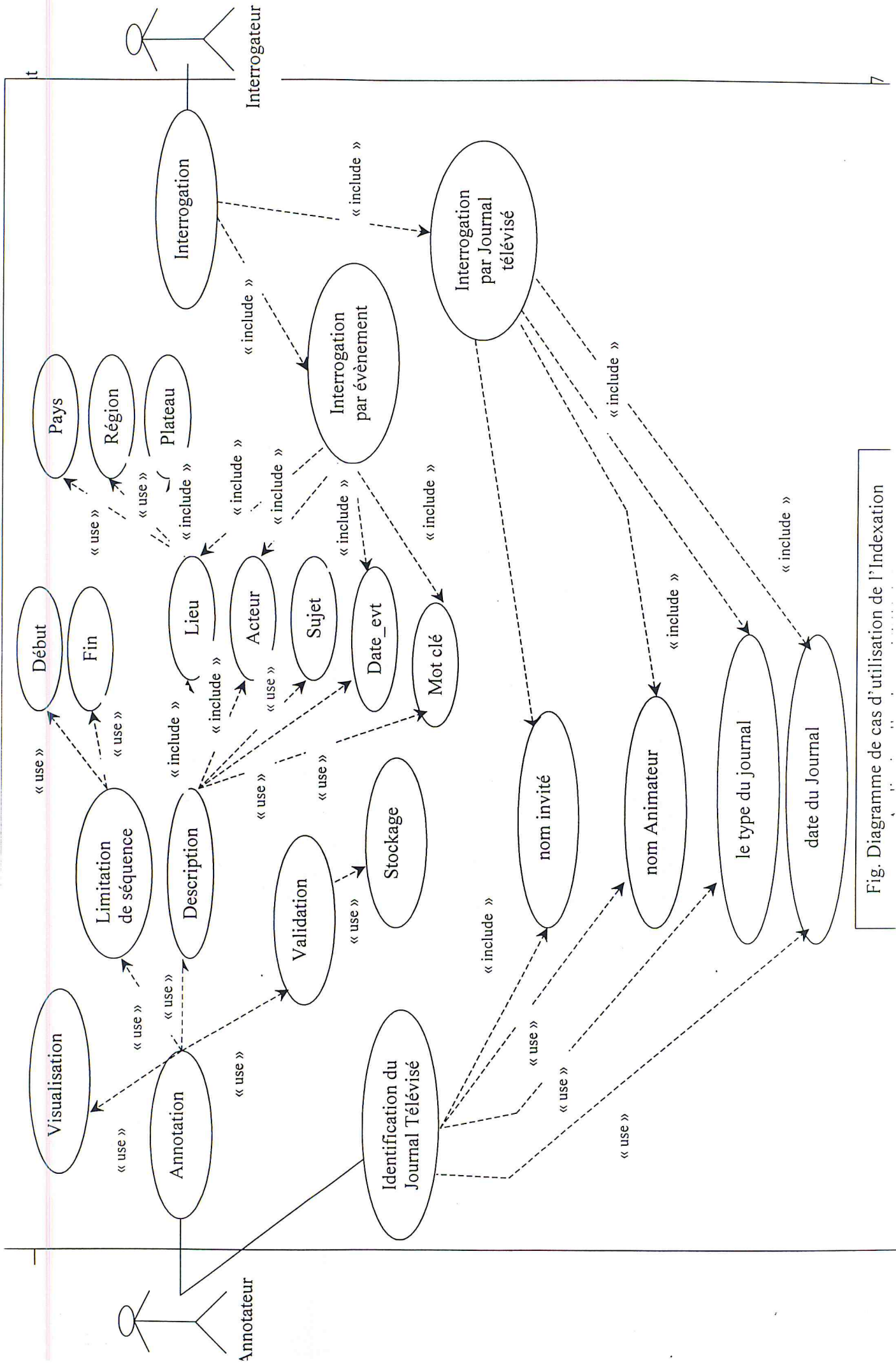


Fig. Diagramme de cas d'utilisation de l'Indexation

### 5.3.3 Les diagrammes de séquence :

La notation UML nous permet de représenter le facteur temps à travers les diagrammes de séquence. Ces diagrammes montrent l'interaction entre les objets selon un point de vue temporel, comme nous l'avons cité dans l'introduction des cas d'utilisation notre système a deux principales fonctions l'annotation et l'interrogation.

#### 5.3.3.1 Diagramme de séquence : Identification du Journal Télévisé :

Il faut rappeler que le processus d'identification d'un journal télévisé correspond au coté structurel du journal, le responsable de la manipulation de cette phase c'est l'acteur principal (l'Annotateur).

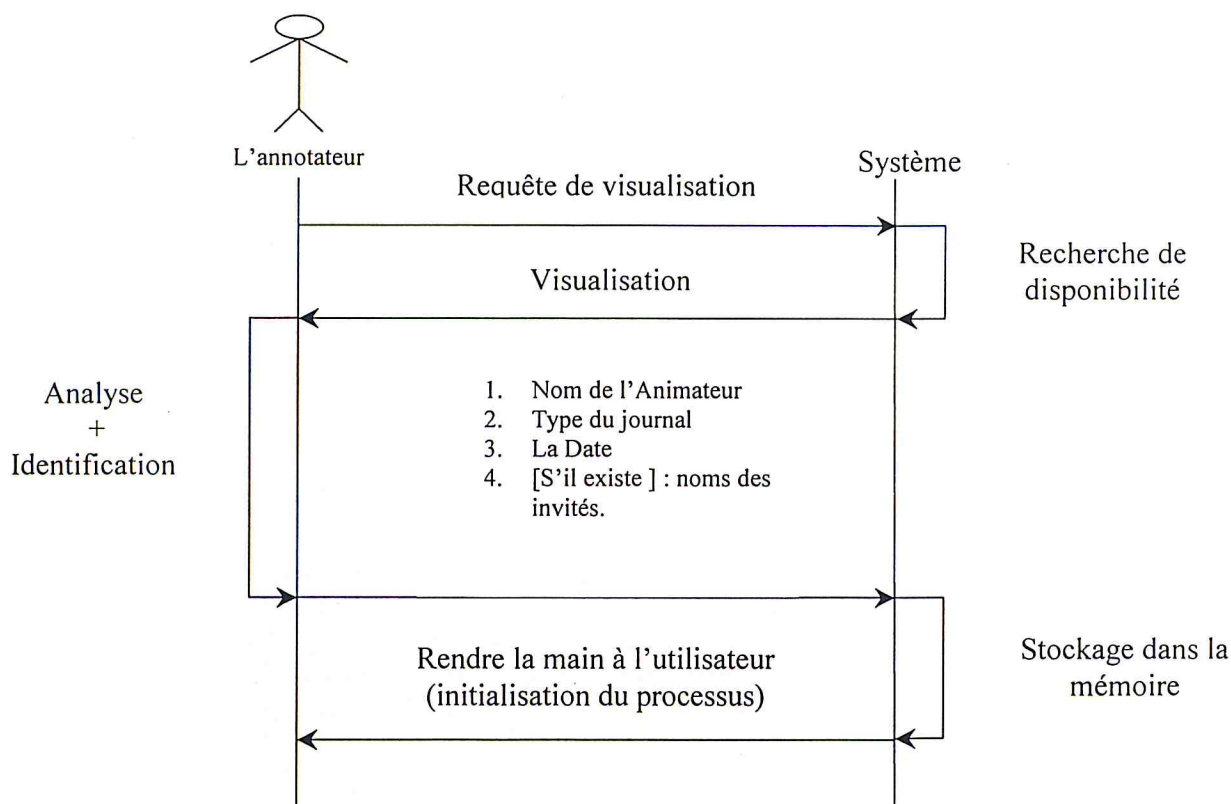


Fig.5.26 diagramme de séquence de l'identification du journal télévisé

#### Scénario :

1. L'annotateur demande au système de visualiser un document multimédia.
2. Le système après le chargement visualise le document multimédia.
3. L'annotateur analyse et identifier le journal, après il fournit au système des informations concernant le journal comme Le type du journal.
4. Le système enregistre les informations précédentes dans la mémoire.
5. Après enregistrement, le système réinitialise le processus.

#### 5.3.3.2 Diagramme de séquence de l'Annotation :

Le premier diagramme sera consacré à l'annotation, il est animé par l'annotateur dont l'objectif est de commenter subjectivement l'événement par rapport le contenu de la séquence.

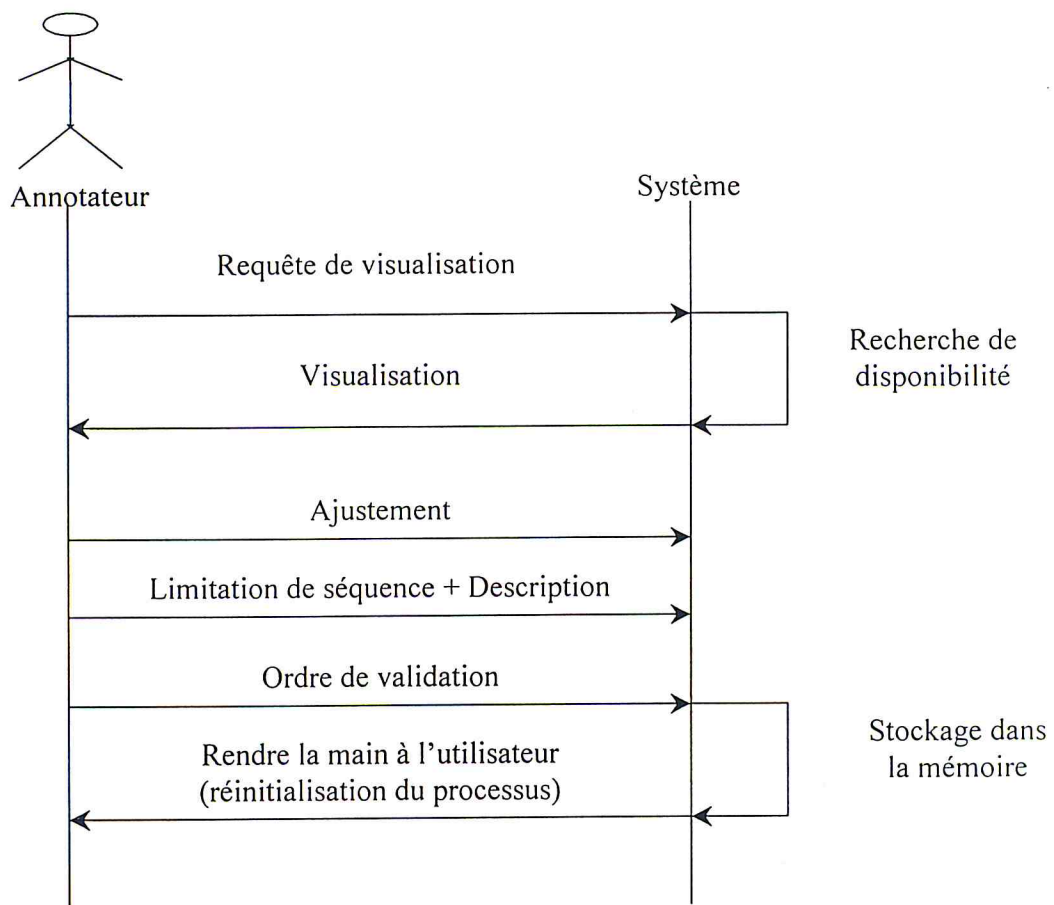


Fig.5.27 Diagramme de séquence de l'annotation

**Scénario :**

1. L'annotateur demande au système de visualiser un journal télévisé déjà identifié.
2. le Système visualise ce journal télévisé..
3. L'annotateur ajuste la séquence ciblée.
4. L'annotateur décrit les limites de la séquence en plus de la description appropriée.
5. L'annotateur valide le choix.
6. Le système enregistre les informations précédentes.
7. Le système réinitialise le processus.

**5.3.3.3 Diagramme de séquence de l'interrogation :**

En analysant le diagramme de cas d'utilisation nous pouvons déduire que le système doit offrir aux utilisateurs deux profils principaux d'interrogation (par contenu sémantique «événement», par journal) inspirés du domaine traité (Journaux télévisés) dans le but de répondre aux besoins (les requêtes) de l'interrogateur.

**a) Diagramme de séquence de l'interrogation par Évènement :**

Le processus actuel est animé par l'acteur secondaire (l'interrogateur) et le système. Il existe dans ce mode d'interrogation quatre sous modes qui sont : interrogation par lieu, interrogation par Acteur, interrogation par mot clé et interrogation par date, comme il peut interroger le système en combinant les modes d'interrogation voulus.



## Diagramme de séquence de l'interrogation par lieu :

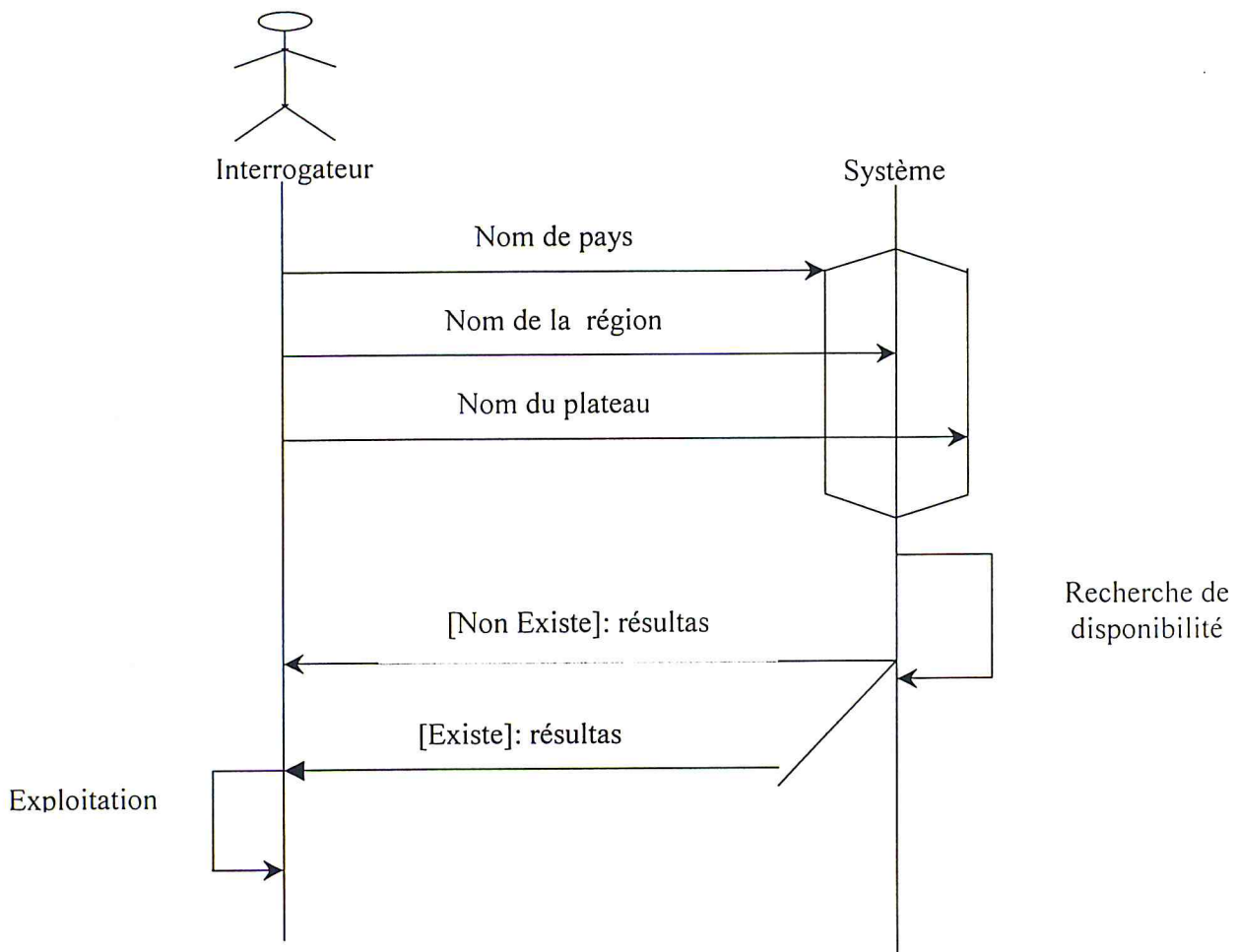


Fig.5.28 Diagramme de séquence interrogation par Lieu

**Scénario :**

1. L'interrogateur interroge le système avec ces requêtes possibles concernant le lieu: soit par Pays, par Région, Par Plateau, ou il les combine.
2. Le système recherche de disponibilité.
3. Après acquisition de la ou les séquences, il les restitue à l'interrogateur.
4. L'interrogateur exploite ces résultats.

**Diagramme de séquence de l'interrogation par Acteur :**

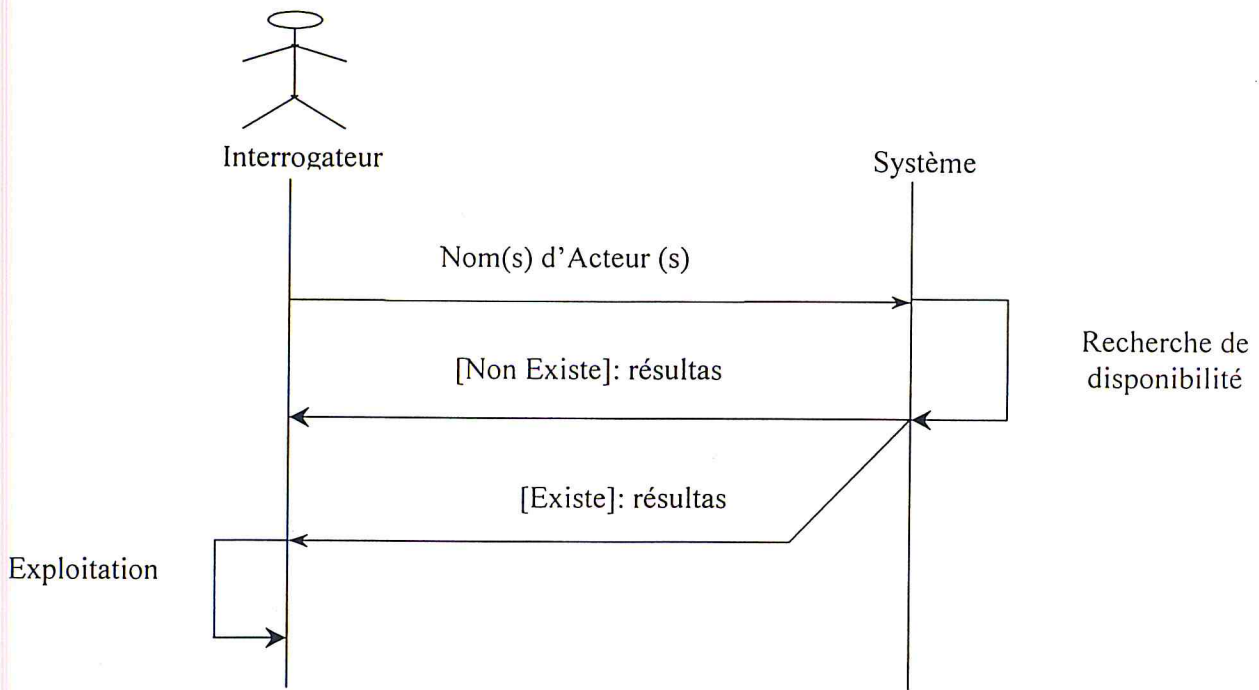


Fig.5.29 diagramme de séquence interrogation par Acteur

**Scénario :**

1. L'interrogateur interroge le système par nom d'acteur(s).
2. Le système recherche de disponibilité.
3. Après acquisition de la ou les séquences il les restitue à l'interrogateur.
4. L'interrogateur exploite ces résultats.

**Diagramme de séquence de l'interrogation par Mot Clé :**

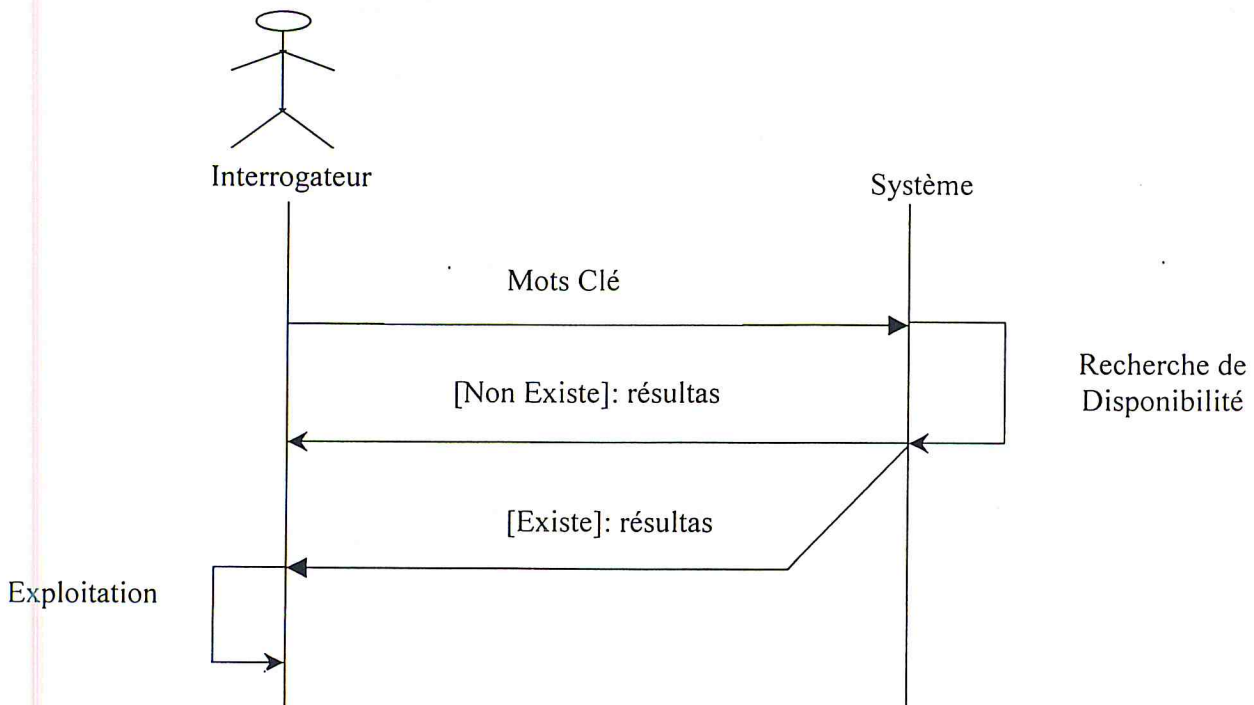


Fig.5.30 diagramme de séquence Mot Clé

**Scénario :**

1. L'interrogateur interroge le système à propos des mots clé.
2. Le système recherche de disponibilité.
3. Après acquisition de la ou les séquences, il les restitue à l'interrogateur.
4. L'interrogateur exploite ces résultats.

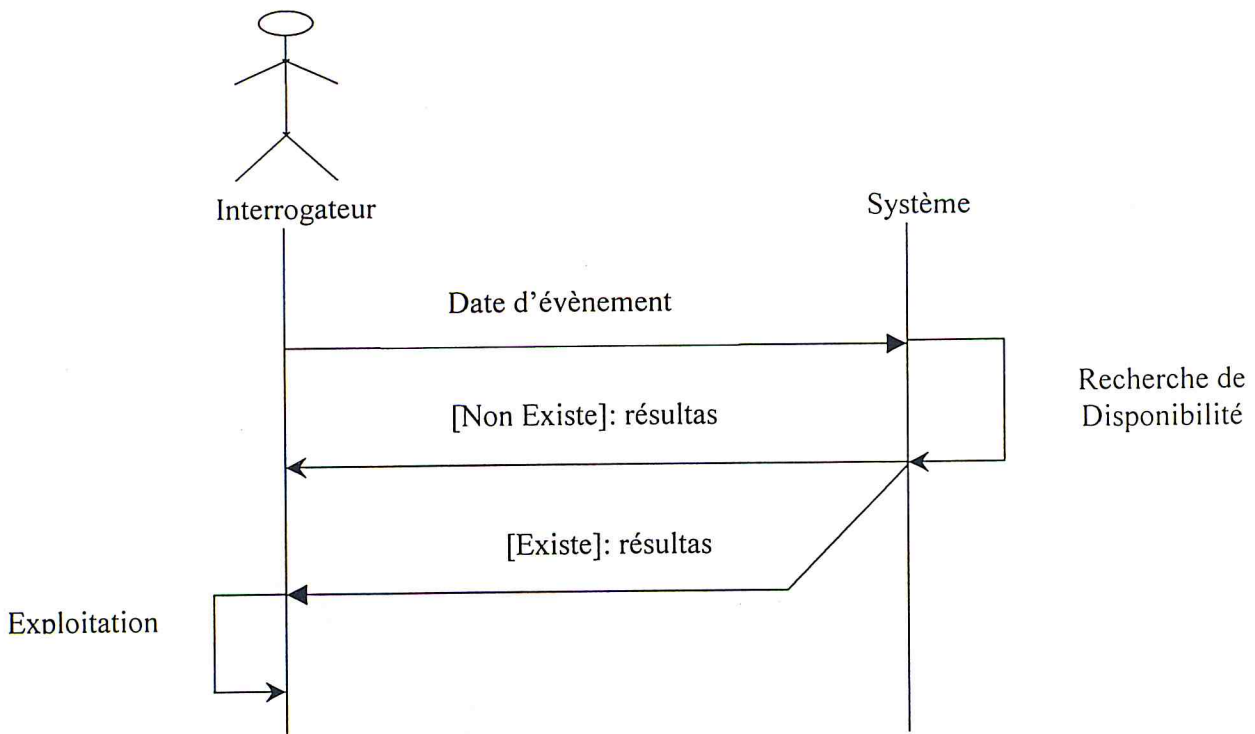
**Diagramme de séquence de l'interrogation par date d'évènement :**

Fig.5.31 diagramme de séquence d'interrogation par date d'évènement

**Scénario :**

1. L'interrogateur interroge le système par date.
2. Le système recherche de disponibilité.
3. Après acquisition de la ou les séquences, il les restitue à l'interrogateur.
4. L'interrogateur exploite ces résultats.

Ce type d'interrogation peut nous donner tout un historique par mois ou par année .etc.

**b) Diagramme de séquence de l'interrogation par Journal télévisé :**

Nous avons vu au paravent que le système permet l'interrogation des séquences vidéo par mot clé, date ...etc. par la suite, nous allons voir que le système permet la recherche des journaux eux même. Dans ce cas trois profils d'interrogation dérive de l'interrogation par journal télévisé, sachant que l'interrogateur peut combiner entre ces trois profils.



**Diagramme de séquence de l'interrogation par type du journal :**

Pendant la journée, on rencontre plusieurs journaux, on peut les classer selon le temps de présentation dans la télévision, l'utilisateur peut interroger le système par type de journal. Exemple : le principal de huit heure de soir, le journal de l'après midi.

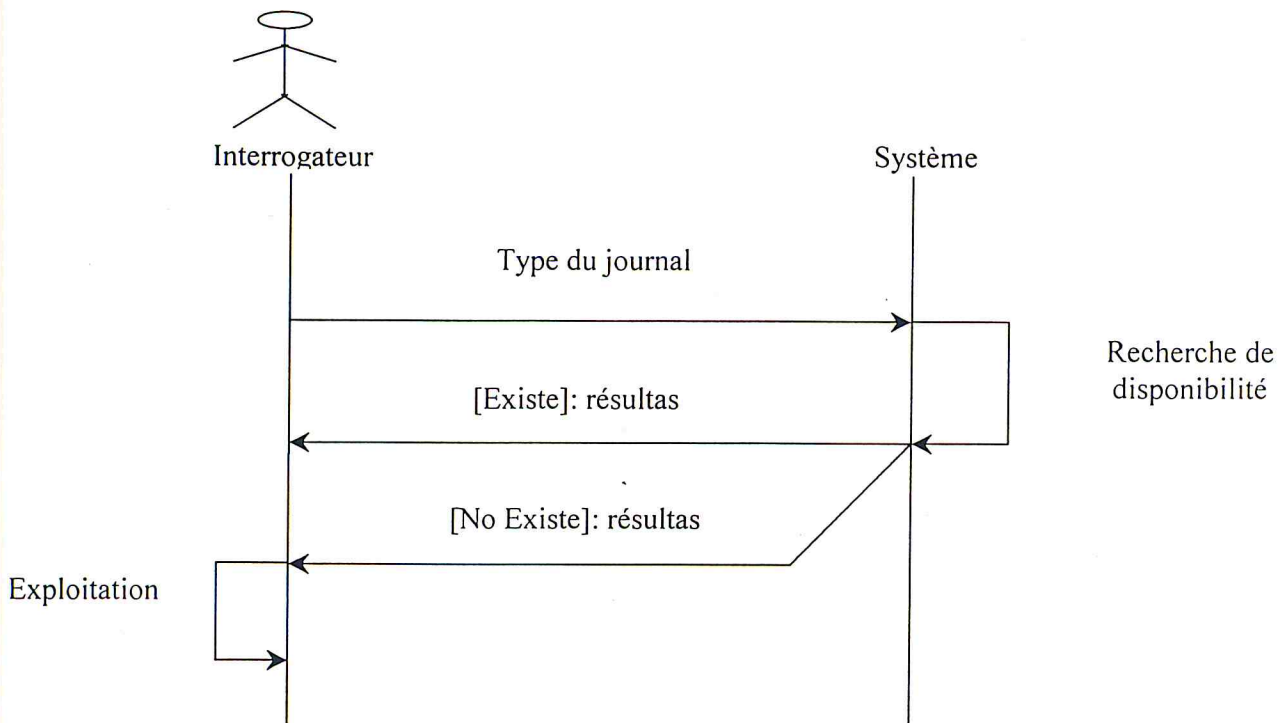


Fig.5.32 diagramme de séquence interrogation par type du journal télévisé .

**Scénario :**

1. L'interrogateur interroge le système par type du journal.
2. Le système fait une recherche de disponibilité.
3. Après acquisition de le ou les journaux, il les restitue à l'interrogateur.
4. l'interrogateur exploite le résultats.

**Diagramme de séquence de l'interrogation par date du journal :**

L'interrogateur peut interroger le système par date journal télévisé, le diagramme suivant présente le séquençage de ce type d'interrogation.

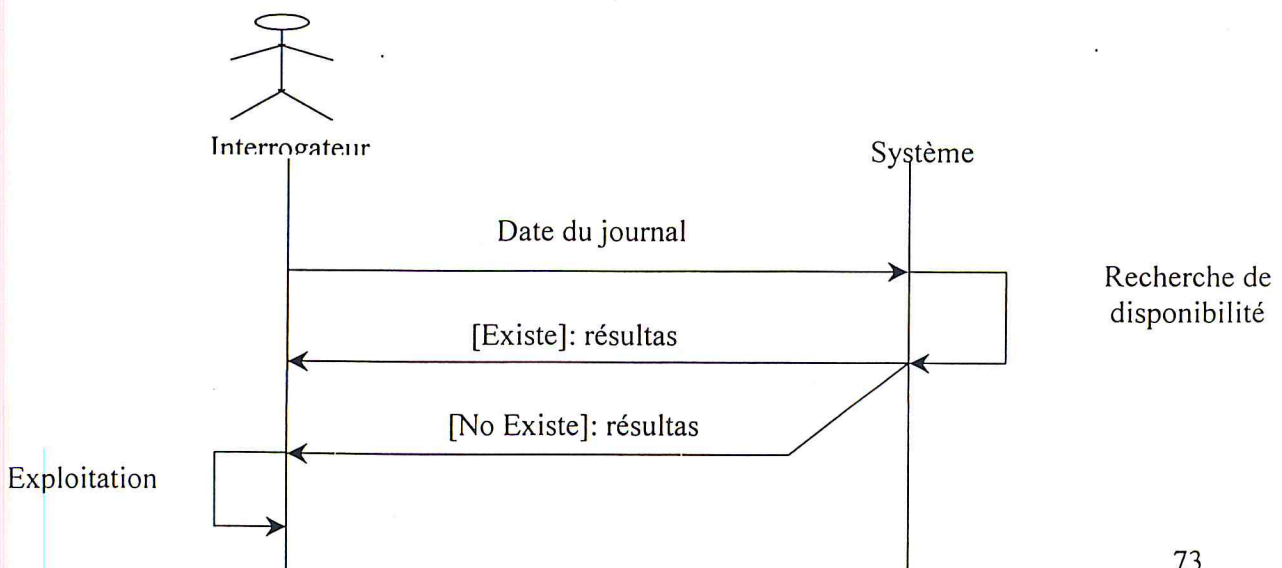


Fig.5.33 diagramme de séquence interrogation par date du journal télévisé

**Scénario :**

1. L'interrogateur interroge le système par date du journal.
2. Le système fait une recherche de disponibilité.
3. Après l'acquisition de le ou les journaux, il les restitue à l'interrogateur.
4. l'interrogateur exploite le résultats.

La recherche par date peut rendre à l'interrogateur l'historique d'un mois ou une année... etc.

**Diagramme de séquence de l'interrogation par Invités :**

Par mis les propriétés d'un journal télévisé les invités, qui peuvent être appelés pour assister à un ou plusieurs journaux télévisés.

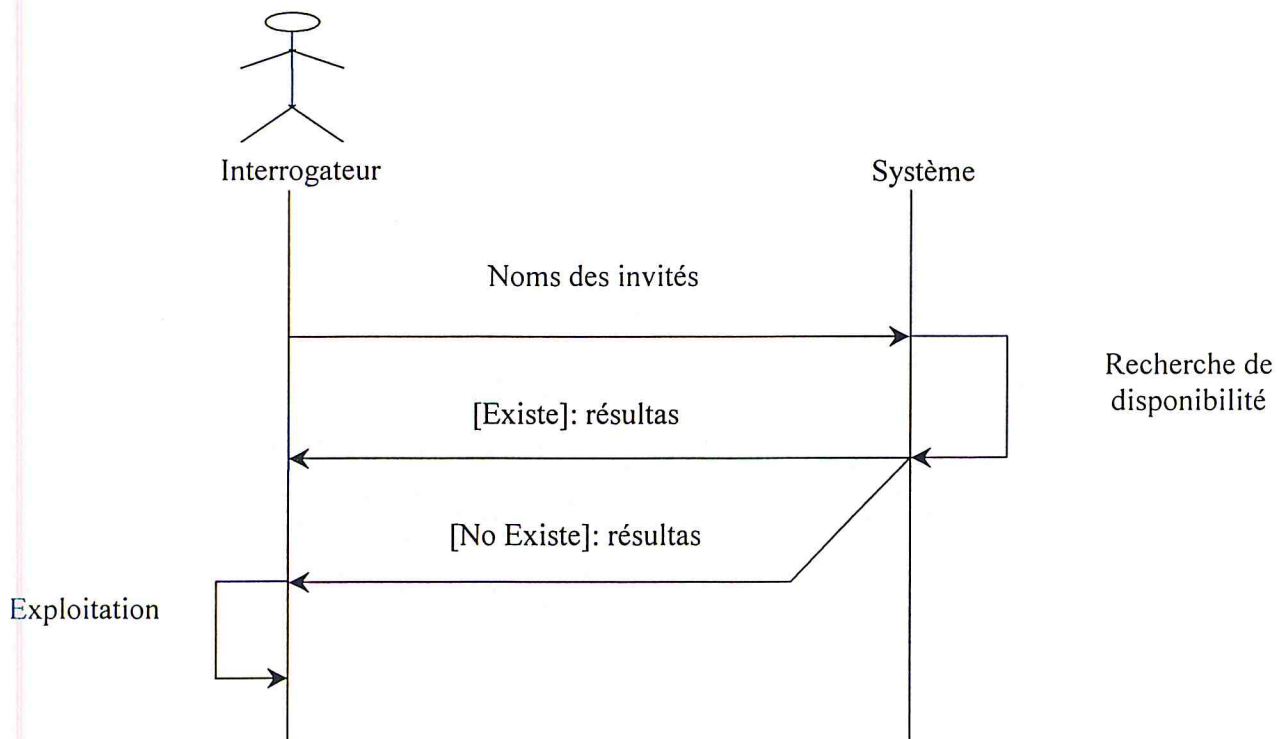


Fig.5.34 diagramme de séquence interrogation par Invités.

**Scénario :**

5. L'interrogateur interroge le système par noms des invités.
6. Le système fait une recherche de disponibilité.
7. Après acquisition de le ou les journaux, il les restitue à l'interrogateur.
8. L'interrogateur exploite ces résultats.

**Diagramme de séquence de l'interrogation par Animateur:**

Une autre propriété du journal télévisé est l'animateur, qui anime le journal télévisé.

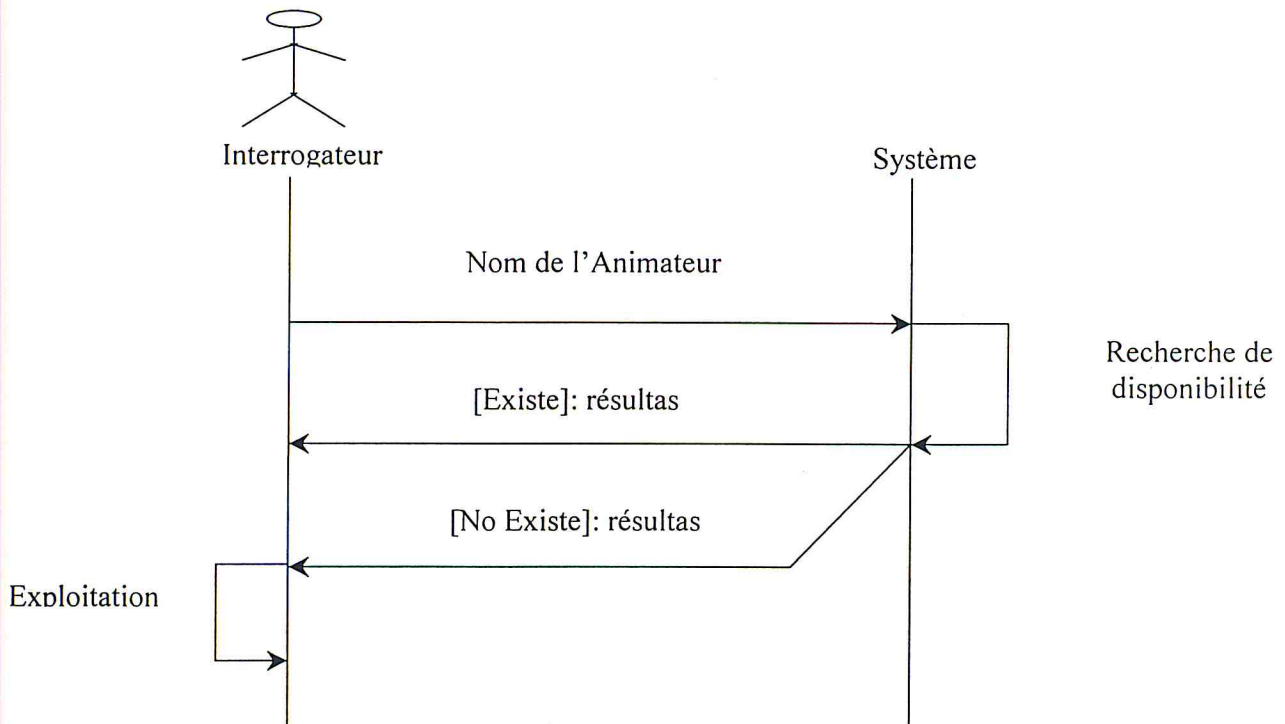


Fig.5.35 diagramme de séquence interrogation par Animateur

**Scénario :**

1. L'interrogateur interroge le système par le ou les nom(s) de l'animateur(s) .
2. Le système fait une recherche de disponibilité.
3. Après acquisition de le ou les journaux, il les restitue à l'interrogateur.
4. L'interrogateur exploite ces résultats.

**5.3.4 Les diagrammes de collaboration :**

La réalisation des collaborations peut également faire intervenir des objets supplémentaires qui n'appartiennent pas au domaine de l'application mais qui sont nécessaires à son fonctionnement. Ces objets effectuent généralement l'interface entre le système et l'utilisateur, ou entre le système et un autre système. La représentation de ces objets permet d'évaluer le coût de l'application. [Mull01].

**a) Demande Visualisation :**

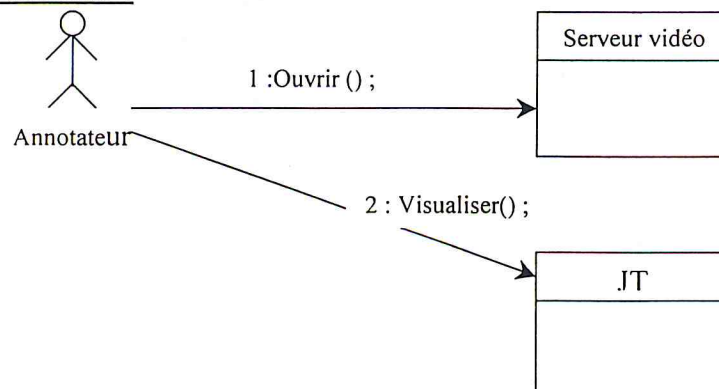
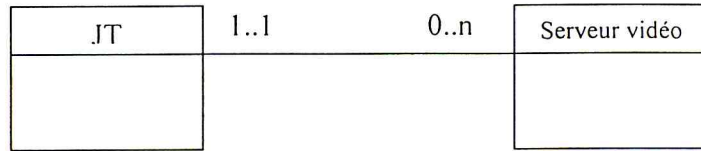


Fig. 5.36 diagramme de collaboration d'une demande de visualisation



Tout d'abord, l'annotateur pour qu'il annote une séquence vidéo, il demande au système de visualiser le journal télévisé, ce dernier est stocké dans le serveur vidéo, après chargement du fichier il le visualise. Le diagramme suivant présente une ébauche de diagramme de classe correspondant au diagramme de collaboration.

Fig.5.37 Ébauche du diagramme de classe



**b) Identification du journal:**

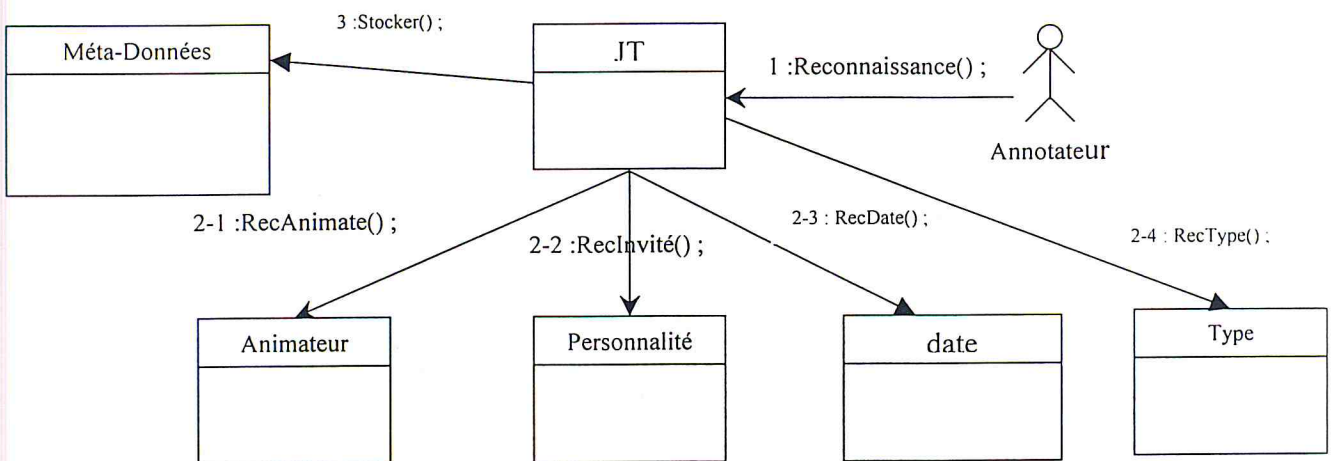


Fig.5.38 diagramme de collaboration d'identification du journal télévisé

L'identification d'un journal télévisé veut dire la détermination de son type, son ou ses animateurs et la date du journal, les invités s'ils y'avaient. Cependant, il faut noter que le serveur Méta-Données regroupe deux sortes de méta-données : Celle relative à la gestion des journaux (localisation, type, date, etc.), et celle relatives aux annotations sémantiques définies sur les contenus des journaux télévisés.

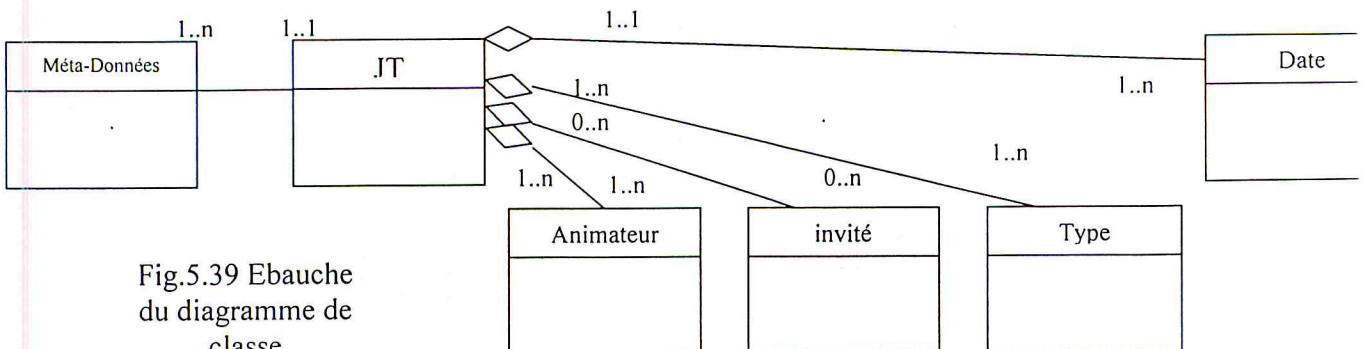


Fig.5.39 Ebauche du diagramme de classe

**c) Annotation :**

Nous avons adopté une approche dans notre système qui consiste à prendre un journal télévisé comme une seule séquence vidéo sans le découper sur des éléments logiques (reportage, information bref, générique, etc.), une autre chose, l'évènement est annoté dans un journal télévisé sur cinq parties qui sont :

*Un évènement = Acteur + Description + Date+ Lieu + Mot clé.*

- **L'Acteur** : signifie l'élément active de l'évènement, pas forcément un individu, il peut être une présentation logique.
- **La Description**: signifie un commentaire décrivent l'évènement.
- **Lieu** : signifie l'emplacement géographique de l'évènement.
- **Date** : signifie la date de l'évènement.
- **Mot clé** : signifie des mots qui caractérisent l'évènement.

Même pour le lieu, il se divise en trois sous parties qui sont :

*Un Lieu = Pays + Région + Plateau.*

- **Pays** : le pays ou l'évènement s'était produit.
- **Région** : la région ou l'évènement s'était produit.
- **Plateau** : c'est la scène contenant l'évènement.

**Application :****Exemple 1 :**

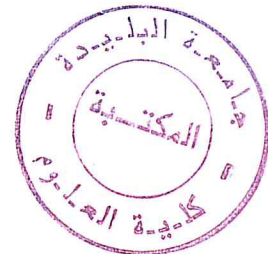
« L'ouverture de l'année universitaire à la présence du ministre de l'enseignement supérieure à l'université de Blida ».

**Description :** *Ouverture de l'année universitaire*  
**Acteur :** *Le ministre de l'enseignement supérieur*  
**Lieu :** *Algérie, Blida, l'université de Blida*  
**Mot clé :** *Université, universitaire, Blida, ...etc.*  
**Date :** *3/10/2004*

**Exemple 2 :**

« l'association 'IKRAA' organise une journée à Tlemcen pour les analphabètes »

**Description :** *Journée pour les alphabètes*  
**Acteur :** *L'association 'IKRAA'*  
**Lieu :** *Algérie Tlemcen,----*  
**Mot clé :** *Analphabètes, Tlemcen, ....etc.*  
**Date :** *6/6/2004*



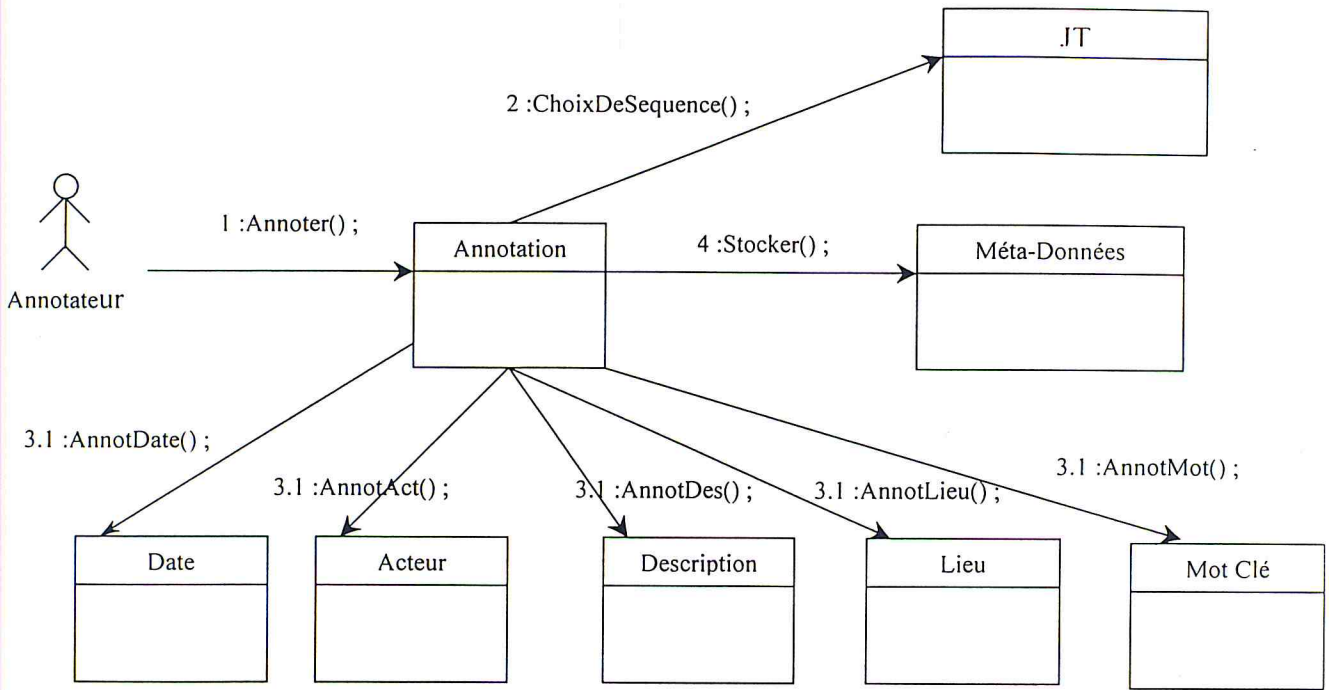


Fig.5.40 diagramme de collaboration de l'annotation

Après choix de la séquence appropriée l'annotateur annote l'évènement comme nous avons expliqué auparavant, en suite, il enregistre cette annotation dans le serveur Méta-Données<sup>1</sup>.

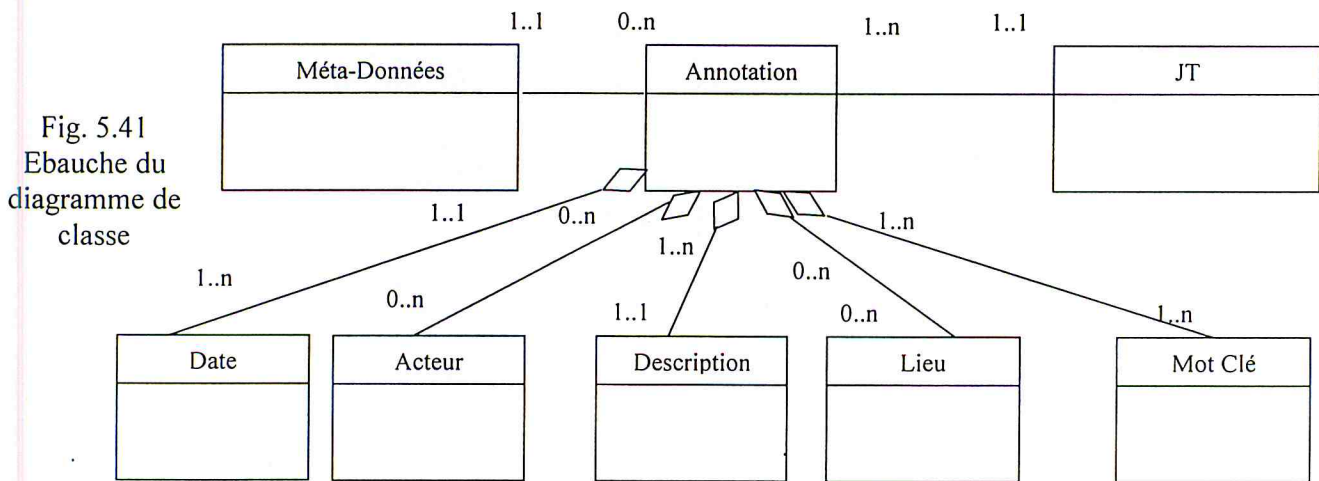


Fig. 5.41 Ebauche du diagramme de classe

<sup>1</sup> Ce terme est inspiré de [SIRSALE]



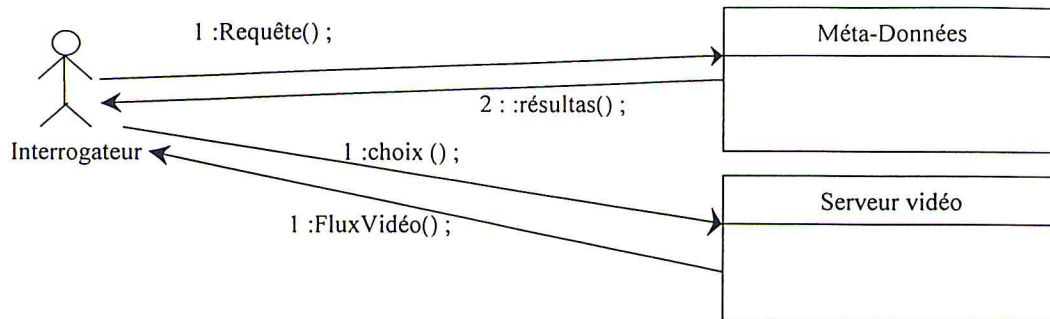
**d) interrogation :**

Fig.5.42 diagramme de collaboration de l'interrogation

'Requête' : c'est la requête établie par l'interrogateur, après l'acquisition du résultats, l'interrogateur fait son choix par 'Choix'.

**5.3.5 Diagramme de collaboration :**

Les fonctionnalités décrites par les cas d'utilisation sont réalisées par des collaborations d'objets du domaine [Mull97], d'où il est envisageable, d'employer les diagrammes de collaborations, ces diagrammes de collaborations se ne sont que des variantes des diagrammes de séquences, par conséquent ils présentent le même sémantique.

Nous présentons deux diagrammes de collaborations, celle de l'annotation et l'autre de l'interrogation.

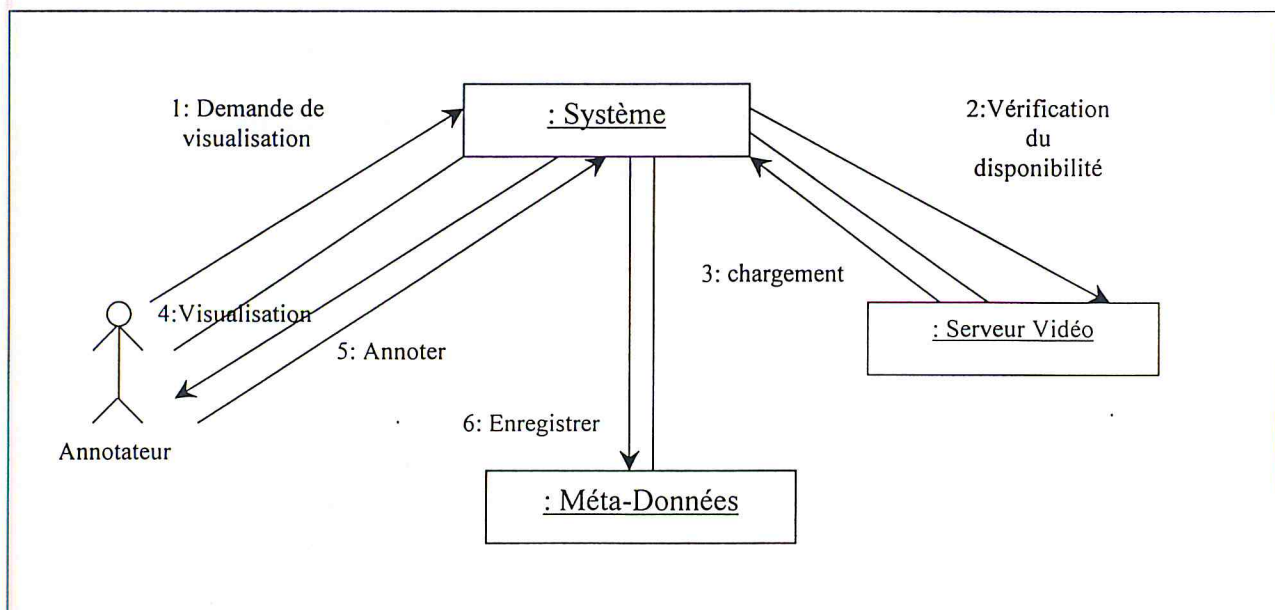


Fig. 5.43 diagramme de collaboration « Annotation »

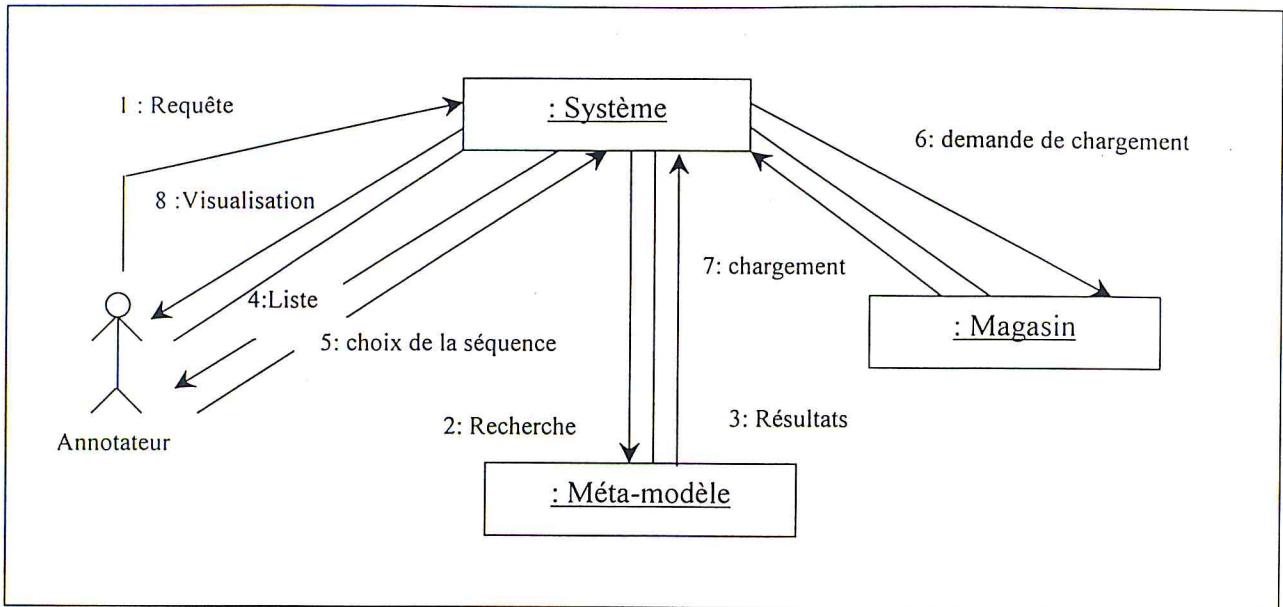


Fig.5.44 diagramme de collaboration « Interrogation »

**5.3.6 Diagrammes d'activités :**

Les diagrammes d'activités sont principalement destinés à représenter le comportement interne d'une méthode (la réalisation d'une opération) ou d'un cas d'utilisation. [Mull02].

On peut distinguer deux types d'activités, les activités d'annotation et les activités de l'interrogation.

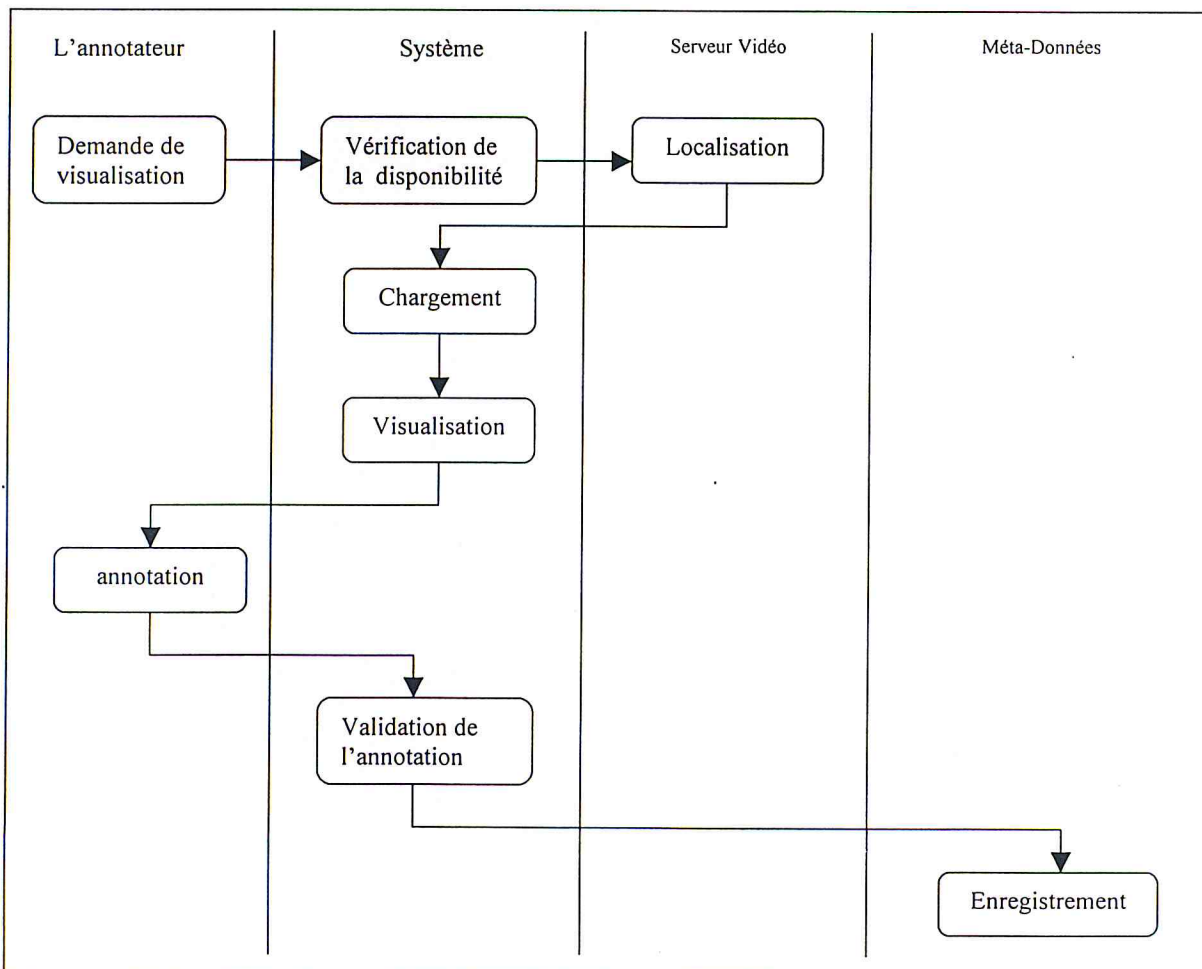


Fig.5.45 diagramme d'activité de l'annotation

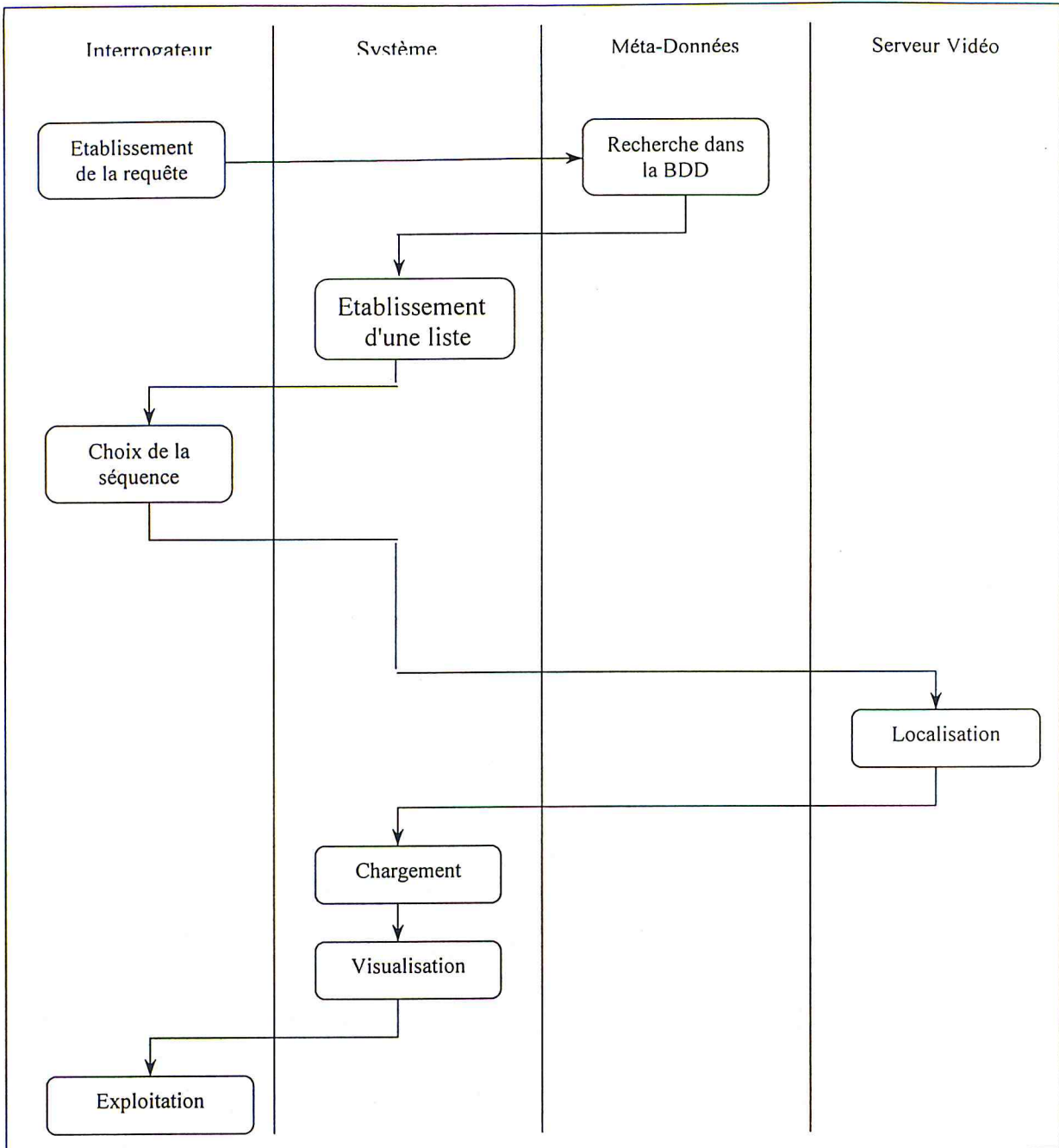


Fig.5.46 diagramme d'activité de l'interrogation

**5.3.7 Diagrammes d'états-transitions :**

Les diagrammes d'états-transitions visualisent les automates d'états finis, du point de vue des états et des transitions.



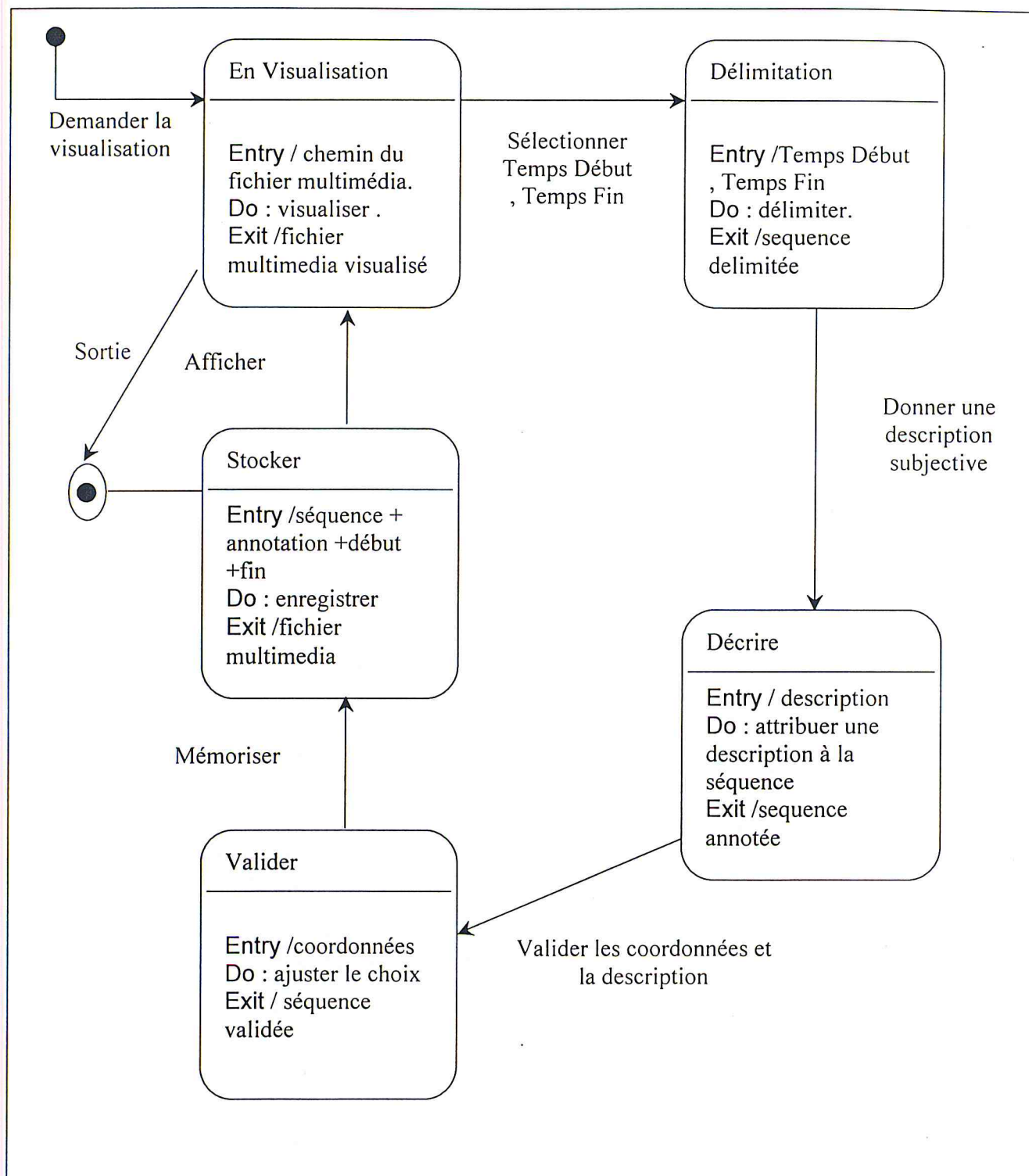
**Diagramme d'états-transition pour la classe Annotation :**

Fig.5.47 diagramme d'états transition.

**5.4 Conception :**

La Conception veut dire la détermination du *Comment* d'une application. [Mull01], afin de répondre à la problématique citée dans le cahier de charge, tout d'abord,

nous allons élaborer une conception de nature « globale », ensuite nous passons vers une conception de nature « détaillée ».

Pour accomplir la conception la notation UML propose trois diagrammes qui sont : diagramme de class, diagramme de composants et de packages.

**5.4.1 Conception globale :**

La conception globale a pour but de décomposer le logiciel en modules et de préciser les interfaces et les fonctions de chaque module. A l'issue de cette étape, on obtient une description de l'architecture du logiciel et un ensemble de spécifications de ses divers composants [Bres93].

Tout d'abord nous présentons le schéma de notre conception qui se base sur l'architecture du projet SIRSALE.

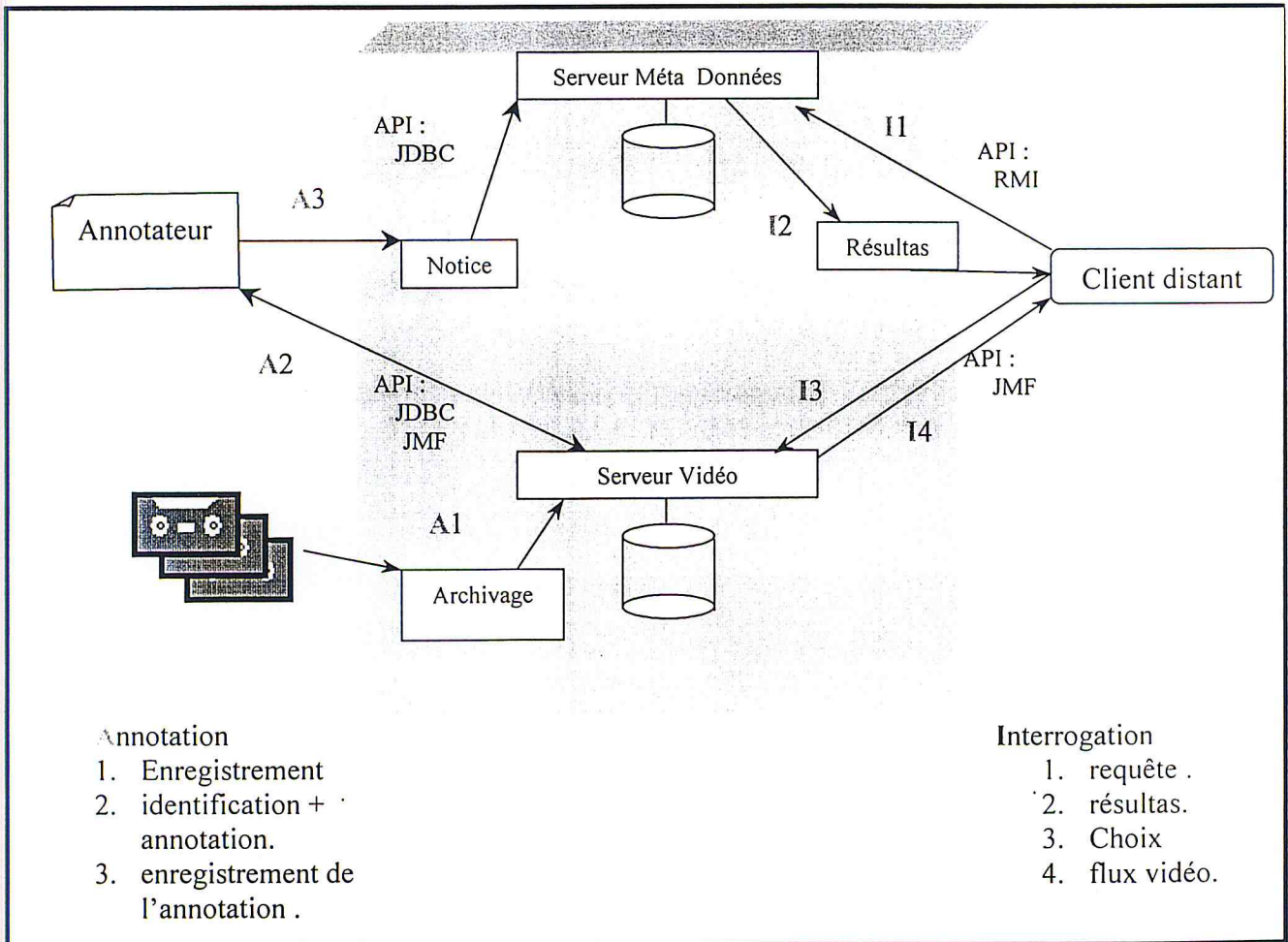


Fig.5.48. Schéma d'IJT( Indexation des Journaux Télévisé )

L'architecture de notre logiciel se présente en trois modules qui sont :

- 1. Module 1 : Identification.
- 2. Module 2 : Annotation.
- 3. Module 3 : Interrogation.

Le Module « Identification » se charge de l'identification des journaux télévisés, le système fait l'enregistrement des fichiers vidéo (Journaux télévisés) dans le serveur vidéo, ensuite l'annotateur fait l'importation des journaux vers l'interface de visualisation afin d'identifier le journal télévisé, cette identification se résume dans la détermination des caractéristiques du journal comme : le type du journal, l'animateur, les invités (s'ils existent) et la date, ensuite ces informations vont être exportées vers le serveur de Méta\_Données pour l'enregistrement.

Le Module « Annotation » a comme but de faire instancier des séquences vidéo par leurs contenus sémantiques, le scénario est comme suit:

L'annotateur fait importer un journal télévisé vers une interface d'annotation à l'aide du serveur vidéo afin qu'il annote le journal, chaque annotation se résume dans la détermination de la séquence vidéo appropriée, la description, le ou les acteurs de l'événement (s'il existe), le lieu (s'il existe), les mots clé, la date. Ces données vont être exportées vers le serveur de Méta\_Données pour l'enregistrement.

Le Module « Interrogation » a comme rôle presque l'inverse du rôle du deuxième module, ce module comme son nom l'indique fait l'extraction des données par l'interrogation du système à propos des séquences vidéo par leurs contenus, le scénario est le suivant :

Le client (L'interrogateur) établit une requête à travers une interface d'interrogation offerte par le système, cette requête va être traduite et le serveur Méta\_Données retourne les résultats vers le client, ce dernier fixe son choix, après, le serveur vidéo crée un flux vidéo qui va être exporté vers l'interface du client, le client dernier exploite le résultats.

Dans la ' Fig.5.48 ' puis que l'application est implémentée sous réseau, elle se trouve dans le serveur et les clients se connectent à distance à l'aide de l'API JAVA RMI <sup>1</sup> (cette solution est imposé dans le cahier de charge).

Nous présentons le diagramme de composants de notre système basant sur notre conception.

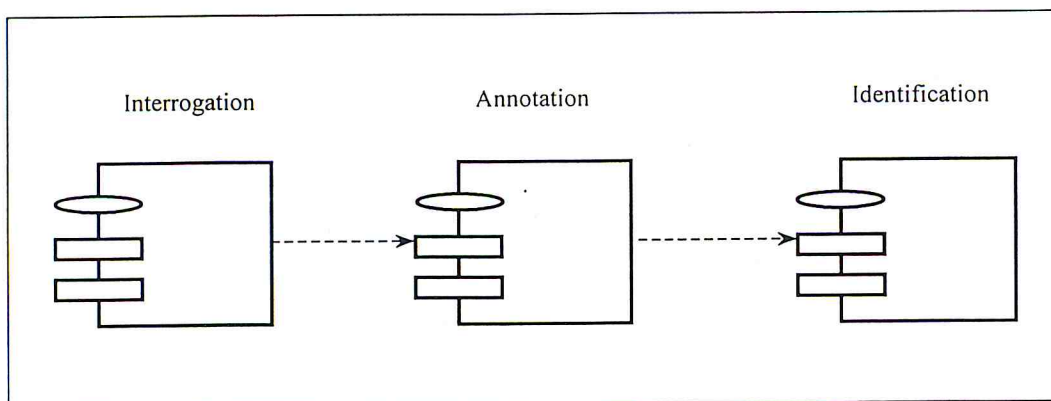


Fig.5.49 Diagramme de Composant des différents modules de système.

<sup>1</sup> voir annexe E



### 5.4.2 Conception détaillée :

La conception détaillée fournit pour chaque module une description détaillée de la manière dont les fonctions du composant sont réalisées: algorithmes, représentation des données [SomL88].

#### 5.4.2.1 Le Module « Identification » :

Nous allons présenter les différents éléments, que nous avons utilisé pour l'établissement du module d'Identification.

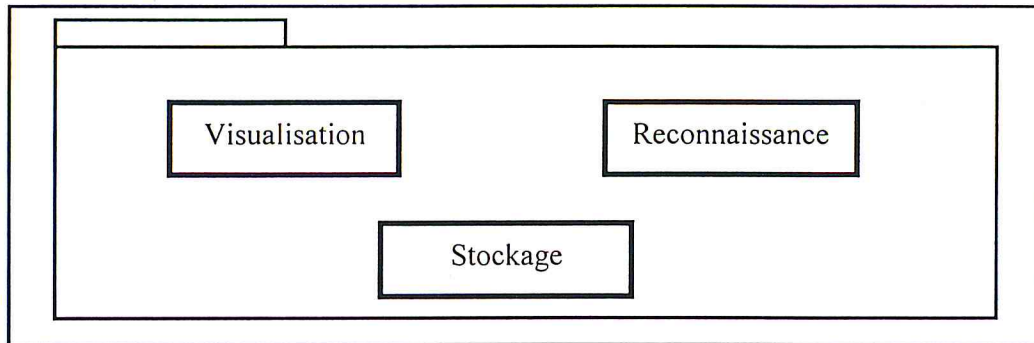


Fig.5.50 les éléments d'identification

La première fonction de notre système afin d'indexer les journaux télévisés c'est l'identification du journal télévisé, ce module englobe trois éléments de modélisation qui sont : Visualisation, Reconnaissance et Stockage.

#### a) Visualisation :

Pour identifier un journal télévisé il faut l'afficher via un outil d'affichage.

#### b) Reconnaissance :

Cet élément est nécessaire dans notre système puisqu'il a comme rôle de reconnaître les différentes parties structurelles du journal.

#### c) Stockage :

Pour garder les informations captées par le module de reconnaissance, il faut enregistrer ces informations, pour qu'elles puissent être utilisables par le module qui suit. Alors dans chaque opération d'identification il faut passer par les trois tâches : Visualisation, Reconnaissance et Stockage.

#### 5.4.2.2 Le Module « Annotation » :

Le deuxième module contient trois éléments de modélisation : Description, Séquence vidéo et Stockage.

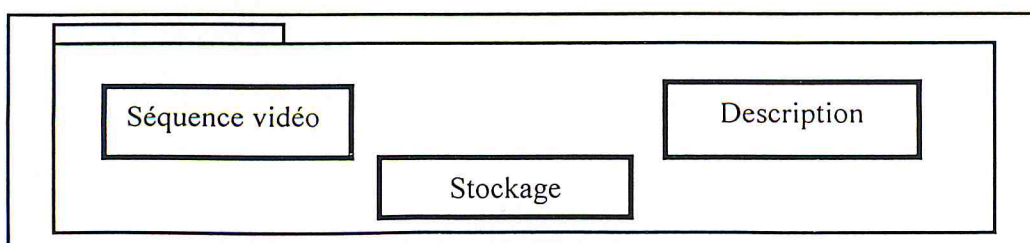


Fig.5.51 les éléments Annotation

**a) Séquence vidéo :**

Tout d'abord il faut choisir la séquence vidéo pour l'annoter.

**b) Description :**

Dans cet élément la description d'un événement se fait pour décrire une séquence vidéo par son contenu sémantique.

**c) Stockage :**

Cet élément est nécessaire pour sauvegarder les informations qui ont été récoltées pour être utiles dans une phase ultérieure. Le module « annotation » veut dire une séquence vidéo, une description et un stockage.

**5.4.2.3 Le Module « Interrogation »:**

Ce module contient seulement deux éléments de modélisation : Etablissement de la requête, Visualisation de résultats.

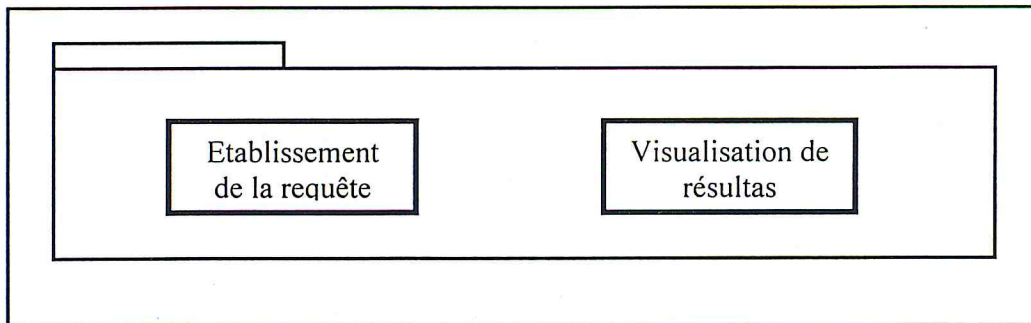


Fig.5.52 les éléments Interrogation

**a) Établissement de la requête :**

Dans cet élément, la requête est établie selon le choix du client sur quelques critères.

**b) Visualisation de résultats :**

Le client exploite le résultat obtenu par l'établissement de la requête

C'est à dire que le module d'interrogation se modélise par une établissement de requête en plus d'une visualisation des résultats.

Après une telle modélisation il faut faire présenter le système d'une façon statique, à l'aide de diagramme de classe. Les diagrammes de classes expriment de manière générale la structure statique d'un système en termes de classes et de relation entre ces classes, un diagramme de classes n'exprime rien de particulier sur les liens d'un objet données, mais décrit de manière abstraite les liens potentiel d'un objet vers d'autres objets [Mull02].

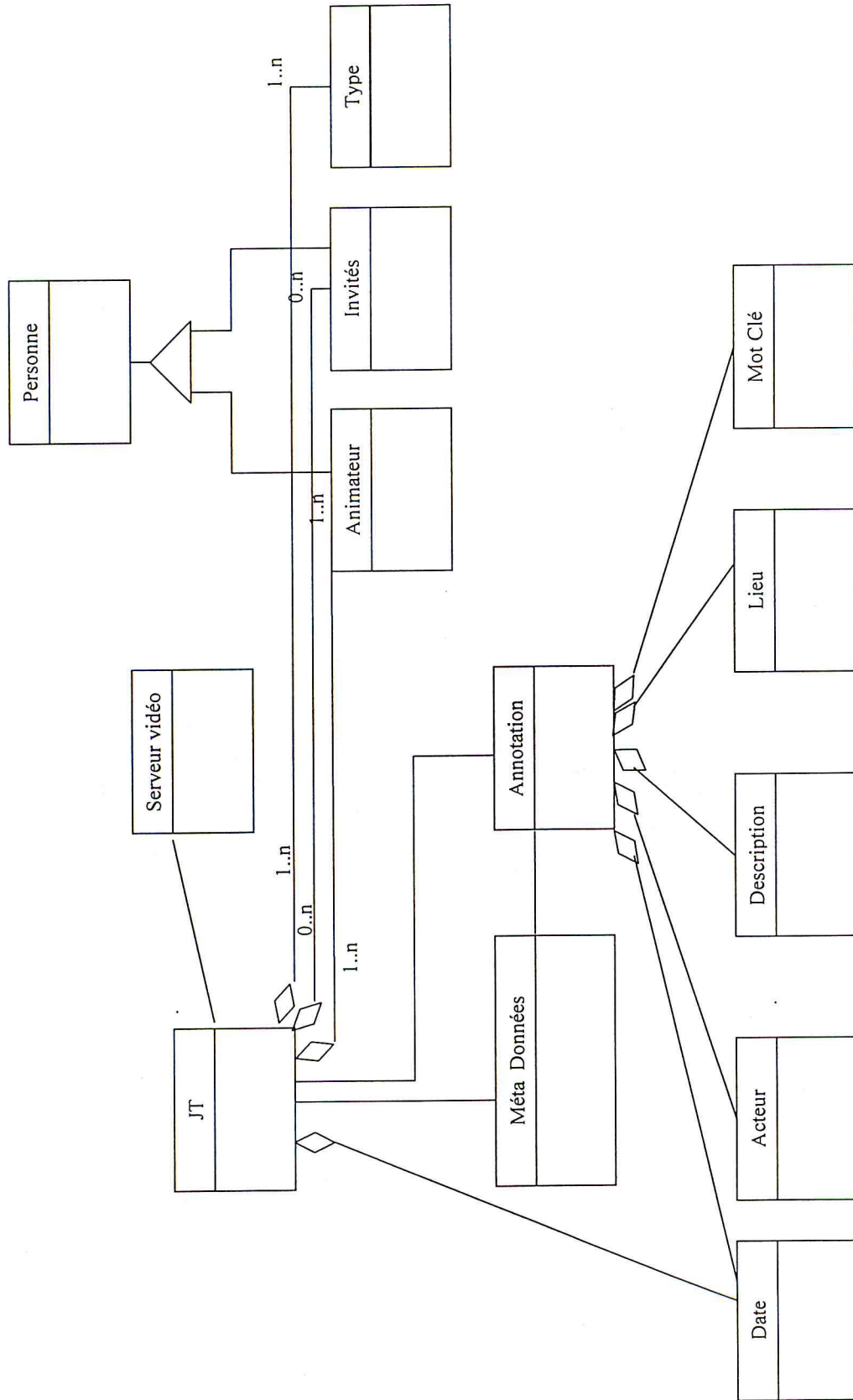


Fig.5.53 Diagramme de class de l'IJT.



# chapitre:VI

# implémentation



### 6.1 introduction :

La suite de ce chapitre exprime la partie implémentation de notre projet, dans le cadre de développement du logiciel, il s'agit d'implémenter la solution évoquée dans la partie conception, l'implémentation c'est une phase aux cours du quelle les algorithmes définis dans la partie conception traduits dans un langage de programmation ou une base de données.

### 6.2 Choix du langage de programmation:

Il faut prendre en compte de manière réaliste les choix de réalisation qui peuvent être imposés à un projet, avant même la conception. Cette situation est malheureusement trop courante en informatique, les informaticiens doivent résoudre un problème à partir d'un fragment de solution.[Mull01].

Dans ce cas le langage de programmation dans lequel doit développer notre logiciel est choisi parmi les langages suivants : C, C++, java, etc. Dans notre cas le choix du langage de programmation a été imposé dans le cahier de charge, c'est java de Sun MicoSystems, c'est un langage orienté objet.

la programmation dans un langage objet est la manière la plus commode de traduire une conception objet en un langage machine [Mull97].

L'implémentation de notre logiciel a été faite sur deux composants principaux qui sont l'application et un serveur de Méta\_Données.

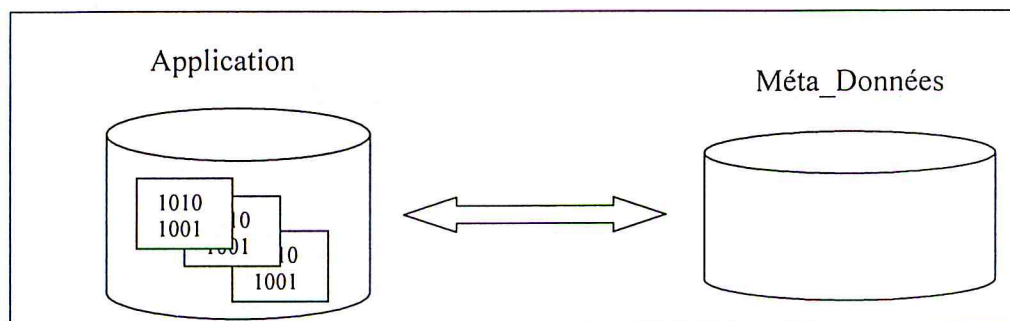


Fig.6.1 un survol de la structure de l'application.

L'application : Engendre les modules de la réalisation de la conception du logiciel.

Méta\_Données : celui qui stocke et envoie les données manipulées par le composant Application.

### 6.3 Implémentation du serveur Méta Données :

L'implémentation de ce serveur consiste à utiliser une base de données unique pour stocker les données relatives à la gestion structurelle des journaux et les annotations sémantiques définies sur les continus des séquences vidéo. Cette implantation présente l'avantage d'une gestion centralisé des données, ainsi nous avons besoin d'un SGBD pour la gestion de la base de données, le choix était le SGBD MySQL pour deux raisons :

-l'exigence du SGBD MySQL dans le cahier de charge :

1. Notre projet est un parti du projet SIRSALE, et ce dernier est implémenté avec le SGBD MySQL, pour cette raison nous avons utilisé ce SGBD.

-les caractéristiques du MySQL :

2. MySQL est sous Licence GPL (aussi bien sous Linux que Windows), ce qui signifie qu'il peut être utilisé gratuitement. *MySQL* est un véritable serveur de base de données SQL multi utilisateur et multi-threaded.

Pour implémenter une connexion à la base de données avec le langage JAVA il faut utiliser l'API JDBC, Le but de cette API (*Application Programming Interface*) est de fournir un niveau commun de connexion à toutes les bases de données, libre ensuite à chaque Editeur (MySQL, ORACLE, ...etc.) de fournir des API de plus haut niveau.

Le code suivant présent l'accès à une base de données MySQL avec l'API JDBC.

```
try {  
Class.forName("org.gjt.mm.mysql.Driver").newInstance();  
    }  
    catch (Exception E) {  
System.err.println("problème de chargement du pilote.");  
        E.printStackTrace();  
    }  
}
```

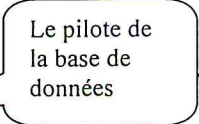


Fig.6.2 Code présente une connexion vers une BD\_MySQL

Les diagrammes de classes servent de base pour l'élaboration des tables relationnelles, le diagramme qui suit présente le modèle de données de l'application IJT.

Nous avons aussi implémenté un programme java, qui est responsable de la création de la base de données d'une façon automatique, en prenant en considération le fait que cette solution ne nécessite pas une connaissance du SGBD MySQL par l'utilisateur.

**Note :** l'utilisateur doit créer la base de données avant l'utilisation de l'application, une contrainte que nous nous somme imposée dans notre projet. Nous verrons par la suite comment cette contrainte a été supportée.



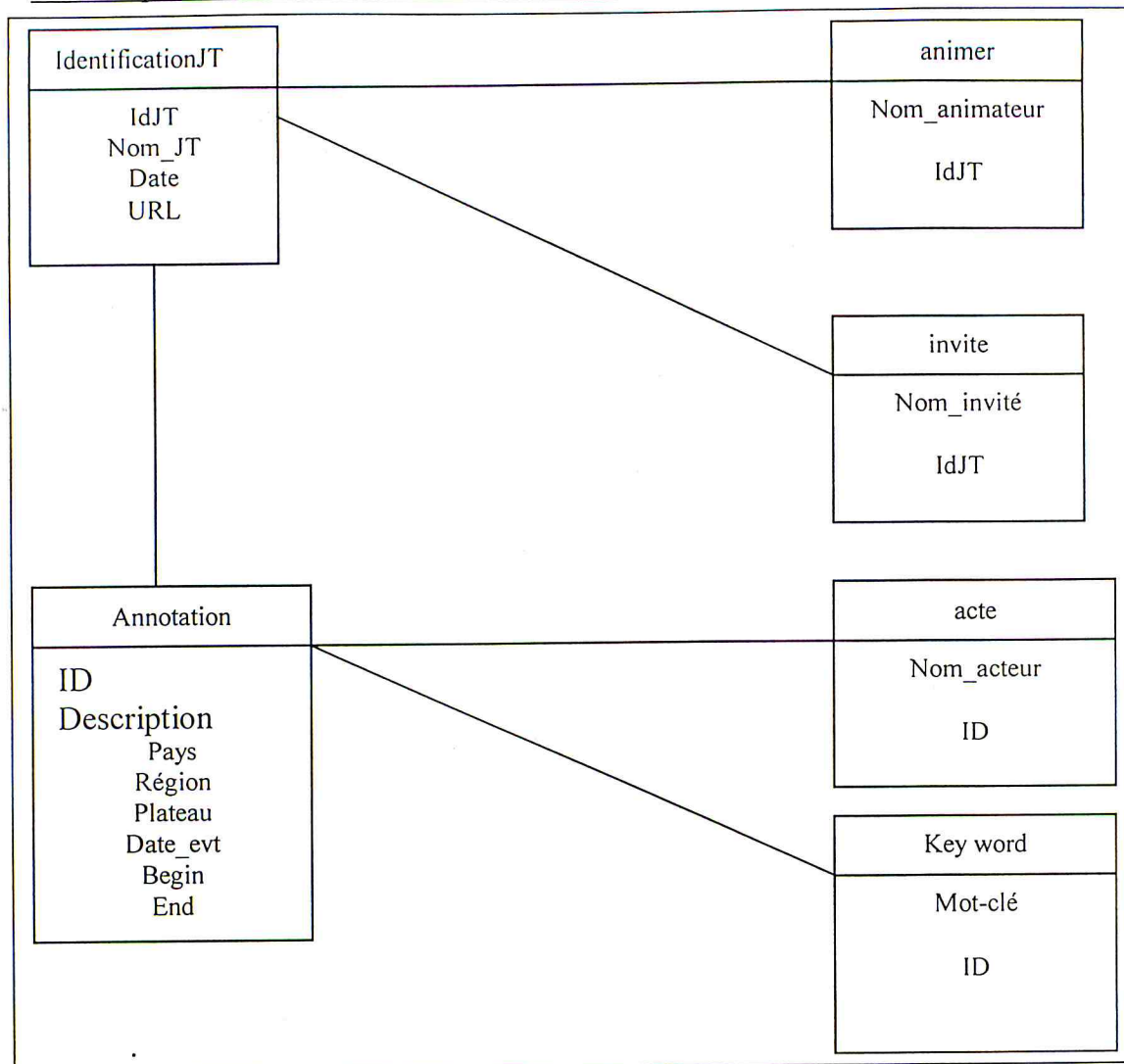


Fig.6.3 Modèle de données

Ce modèle présente la partie III du modèle de données de SIRSALE

#### **6.4 Implémentation du Composant Application :**

Notre application est modélisée d'une façon fonctionnelle sous trois modèles qui sont :

1. Une interface d'identification des journaux télévisés.
2. Une interface d'annotation par contenu sémantique des séquences vidéo.
3. Une interface d'interrogation par contenu sémantique des séquences vidéo.

Dans l'activité d'annotation et d'interrogation, l'annotateur sera amené à visualiser la vidéo à plusieurs reprises, pour cela, nous avons implémenté une interface de visualisation de vidéo. L'interface de visualisation implémentée propose une panoplie de fonctionnalités magnétoscopes totalement paramétrable : lecteur rapide, ou en ralentie, lecteur image par image, le recul rapide, etc.

### 6.4.1 L'interface de visualisation :

Notre interface de visualisation de vidéo est basée sur la technologie JMF (*Java Media Framework*), développée par Sun Microsystems, Inc, c'est-à-dire le créateur de Java, ainsi nous pouvons utiliser librement cette technologie dans le cadre de notre projet.

Le traitement de la vidéo par l'API JMF est organisé en plusieurs paquetages, le paquetage fondamental est `javax.media`. Il contient les interfaces et les classes fondamentales pour le traitement vidéo.

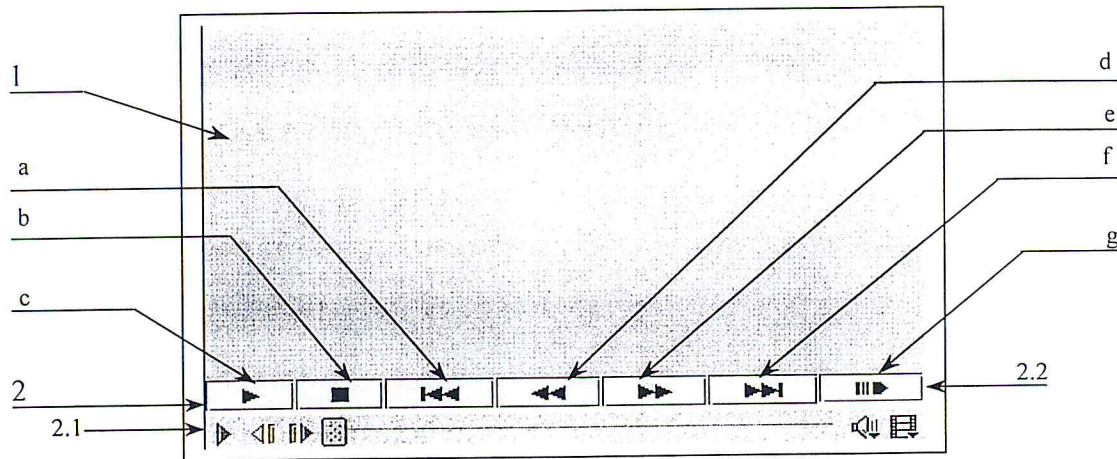


Fig.6.4 le lecteur vidéo

Afin d'afficher un vidéo, tout d'abord on repère la source multimédia par une URL (Uniform Ressource Locator) par le code suivant :

```
URL mediaURL = new URL (urlDocumentMultimedia) ;
```

Fig.6.5 Spécification de la ressource multimédia

Puis on crée une instance de type *Player* qui est associé à la ressource multimédia par le code suivant :

```
Player player = Manager .creatplayer (mediaURL) ;
```

Fig6.6 Spécification du player

L'interface de visualisation est décomposée en deux principaux composants graphiques :

1: *l'écran de visualisation* de la vidéo, il se charge d'afficher d'une manier continue n'importe quel morceau de vidéo indépendamment de sa structure et de sa localisation. Il respecte la contrainte de temps réel, nous avons utilisé une technique *Streaming*<sup>1</sup>. C'est une technique de transfert de flux multimédia en temps réel, la lecture d'un document vidéo peut s'effectuer avant la totalité de leur chargement

<sup>1</sup>Voire Annexe C

2 : le *panneau de commande* de gestion de la vidéo, ce panneau est similaire à celui d'un magnétoscope, son rôle est de fournir des fonctionnalités qui contrôlent la manipulation du vidéo courant, il est même décomposé en deux sous composants :

*Panneau de contrôle* : permet d'arrêter le déroulement de la vidéo, de la reprendre, d'avancer et/ou de reculer pas à pas, de découper la sortie sonore et d'avoir des informations sur la vidéo en cours, de se déplacer avec le glisseur selon le désir.

*Les Boutons de contrôle*: ce composant offre les fonctionnalités suivantes :

- a : Démarre la lecture (Play).
- b : arrêt de la lecture (Stop).
- c : retour 5 secondes.
- d : retour rapide.
  - e : avance rapide.
  - f : avance 5 secondes
  - g : lecture ralentie.

#### 6.4.2 Scénario de l'implémentation :

Notre logiciel (IJT) a des tâches bien précises et l'algorithme de manipulation est défini dans les processus qui sont :

##### ❖ *Processus d'identification :*

Ce processus a comme objectif d'identifier les journaux télévisés avec ses parties structurales définies dans la partie conception, en remplissant un formulaire bien défini, après la validation de ce formulaire, un programme spécifique récupère ses données et les exporter vers le serveur de Méta\_Données via l'API JDBC dans des tables qui sont :

« IdentificationJT » « Animer » et « invite » .comme le présente le schéma suivant :

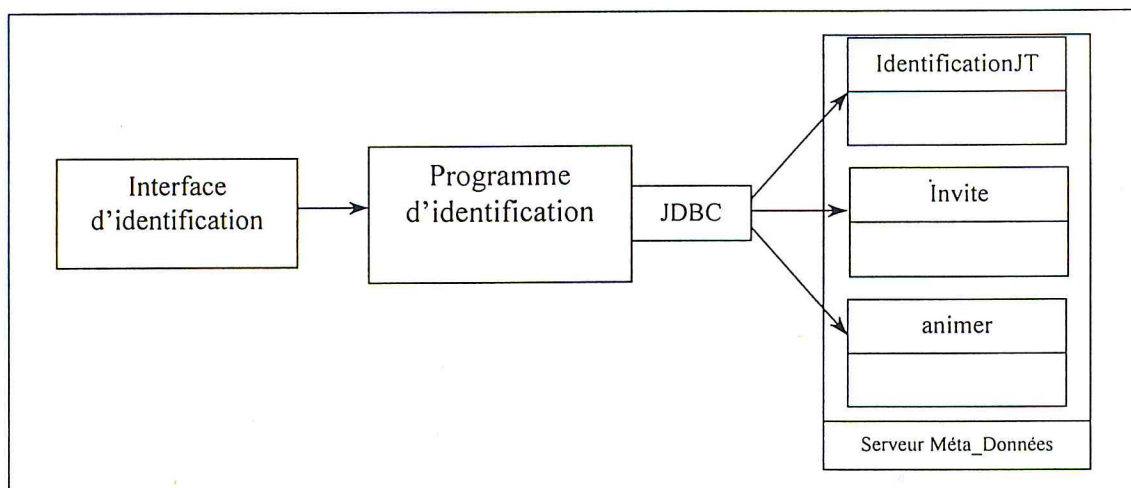


Fig.6.7 Processus d'identification



❖ *Processus d'annotation :*

Toujours via une interface graphique, l'annotation se fait après une visualisation d'une séquence vidéo (pour sélectionner le *début* et la *fin* de la séquence vidéo), en remplissant un formulaire qui contient des informations concernant le contenu sémantique de cette séquence. Un programme récupère ces informations et les envoie vers la base de données dans les tables suivantes : « Annotation », « acte » et « KeyWord » le schéma suivant explique ce processus:

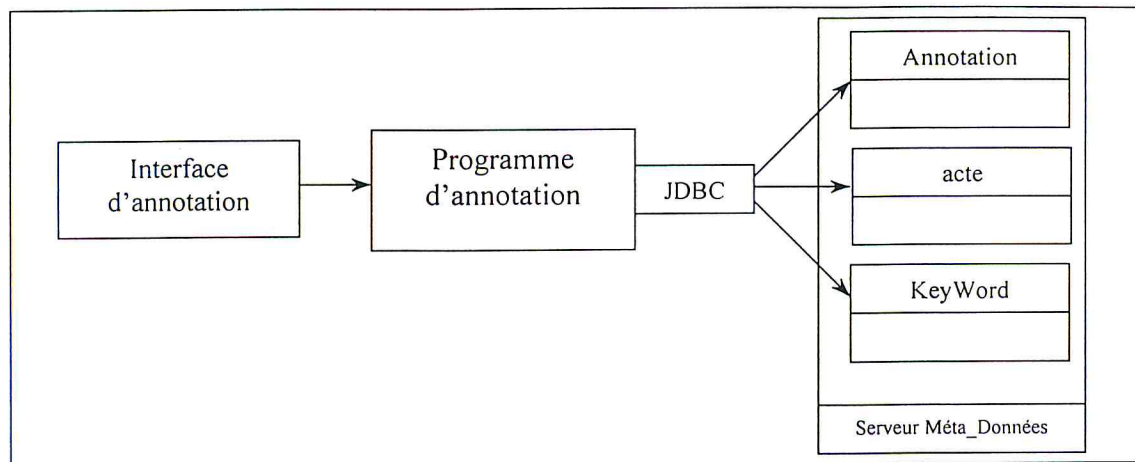


Fig.6.8 Processus d'identification

❖ *Processus d'interrogation :*

Ce processus a comme rôle d'un moteur de recherche, cette recherche se fait sur deux niveaux.

**Interrogation par journal télévisé :**

Dans cette interface l'utilisateur peut construire des requêtes en remplissant un formulaire qui décrit le journal, ensuite un programme collecte ces informations, il établit des requêtes, ces requêtes vont être envoyées vers l'SGBD de la base de données pour qu'il puisse faire une recherche, ensuite le programme fait le chargement des résultats dans une table graphique, après une sélection du résultat, un autre programme charge la vidéo du journal choisie pour qu'elle puisse être exploitable.

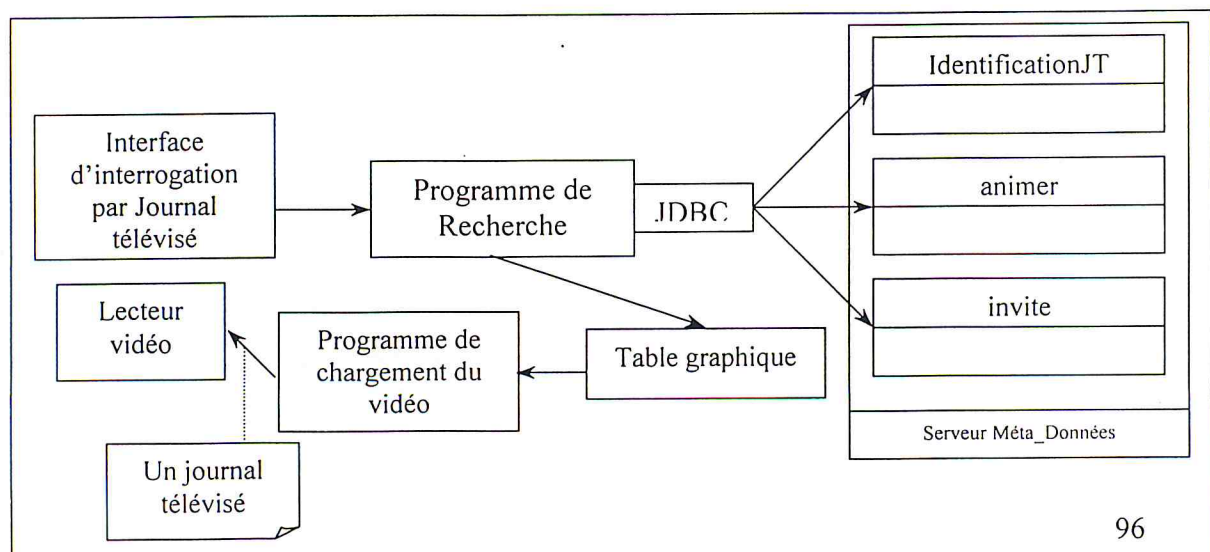


Fig.6.9 interrogation par journal télévisé

### Interrogation par évènement :

cette interface est presque identique à la précédente, la seule différence réside dans le type de résultat. L'interrogation par évènement fournit une table graphique contenant les informations de gestion des séquences vidéo résultant des requêtes, en revanche l'interrogation par journal sa table de résultats contient des informations de gestion des journaux télévisés.

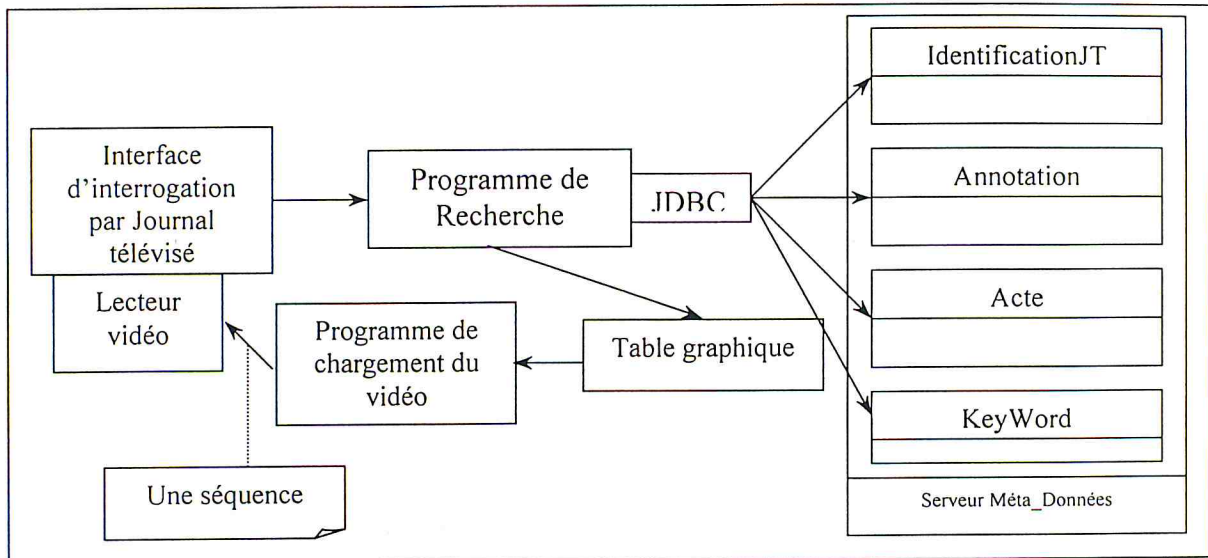


Fig.6.10 interrogation par contenu sémantique d'évènement

### 6.5 Modélisation :

L'IJT comprend une interface graphique afin de permettre à tout utilisateur, quel que soit son niveau et sa pratique en informatique, de pouvoir visualiser, annoter, rechercher des séquences vidéo et de les indexer par contenu sémantique. L'IJT est modélisée comme suit :

1. cinq interfaces graphiques :
  - interface d'identification des journaux télévisés ;
  - interface d'annotation des journaux télévisés ;
  - interface d'interrogation par journal télévisé ;
  - interface d'interrogation par évènement ;
  - interface des annotations contenues dans un journal.

### 6.6 Réalisation des cas d'utilisation :

Cette section va être présentée à l'aide des diagrammes de séquences suivis par des images de notre logiciel, dont le but d'argumenter l'activité de validation, et d'assurer la réalisation des cas d'utilisations décrits dans la spécification des besoins.

Les paragraphes suivants présentent les cas d'utilisation principaux de notre outil, l'annexe A contient une partie qui décrit tous les outils proposés par notre logiciel, ainsi que les cas d'utilisations secondaires.

**6.6.1 Validation du Cas d'utilisation « identification du Journal Télévisé » :**

Ce cas d'utilisation définit quatre sous cas d'utilisation qui sont :

1. identification Animateur ;
2. identification Invités ;
3. identification type du journal ;
4. identification de la date.

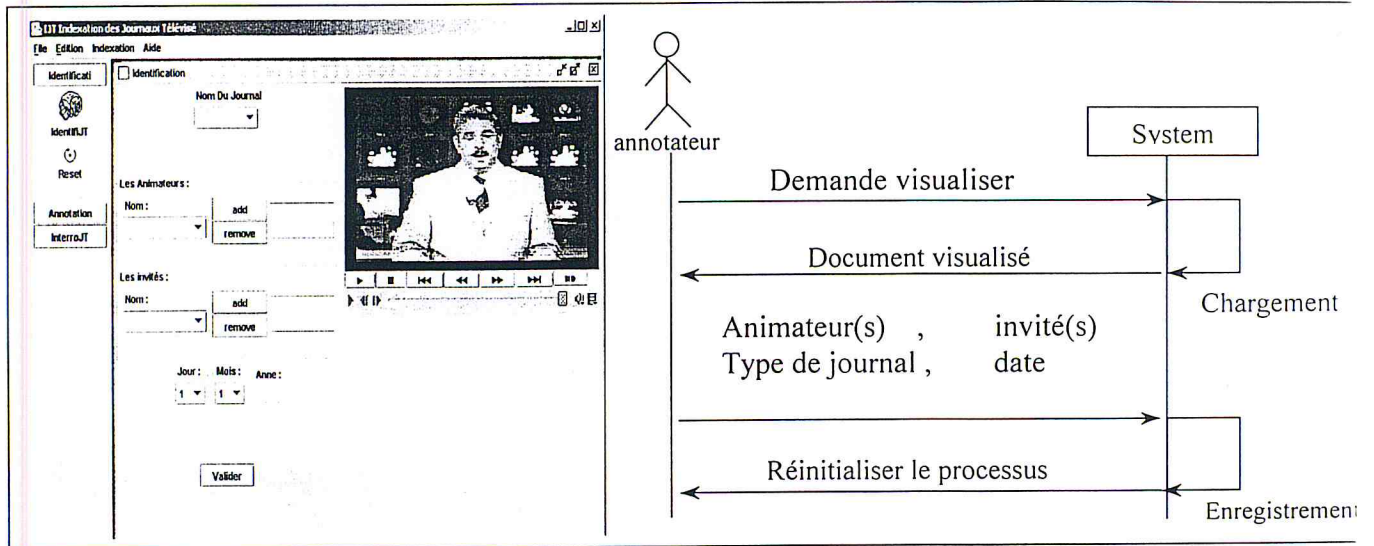


Fig.6.16 validation du cas d'utilisation identification du journal télévisé

**6.6.2 Validation du cas d'utilisation « Annotation » :**

Ce cas d'utilisation engendre quatre sous cas d'utilisation qui sont :

1. Visualisation ;
2. Limitation des séquences ;
3. Description ;
4. Validation .

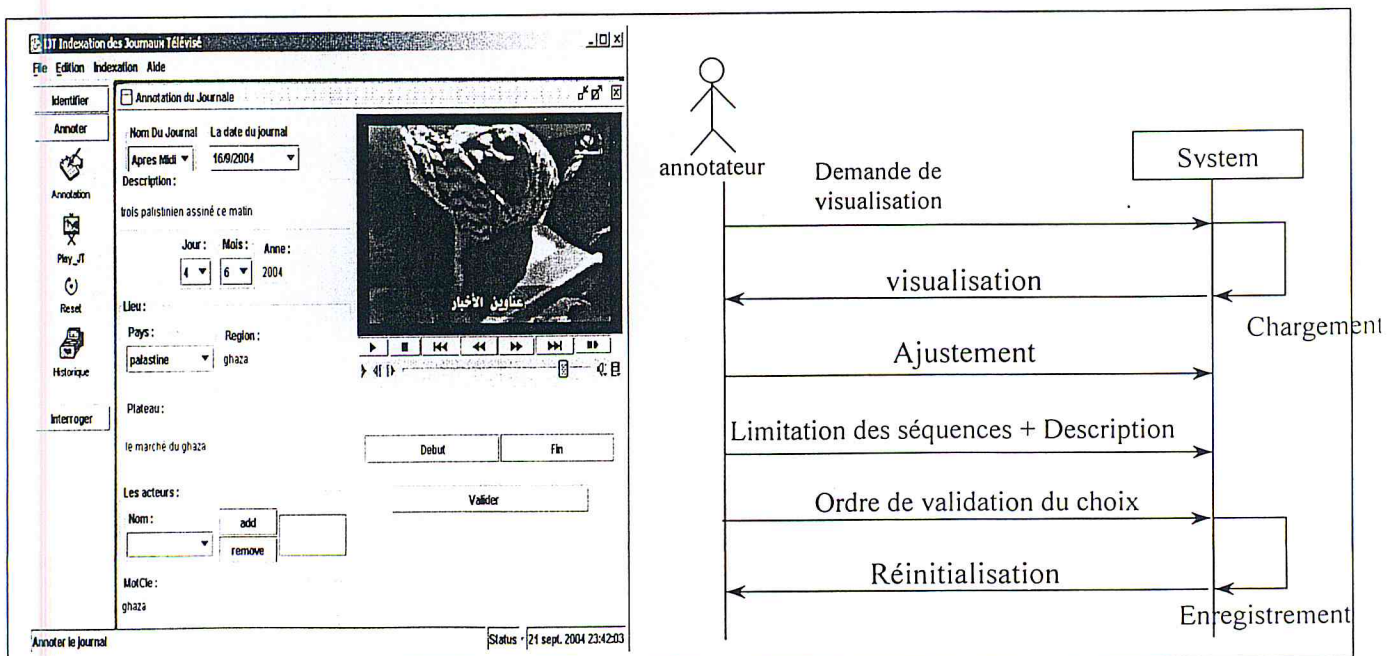


Fig.6.17 validation du cas d'utilisation « Annotation »



**6.6.3 Validation du cas d'utilisation « Interrogation » :**

Le cas d'utilisation « Interrogation » est divisé en deux cas d'utilisations principaux qui sont « interrogation par journal télévisé, et par événement ».

**a) Validation du cas d'utilisation « interrogation par journal télévisé » :**

Ce cas d'utilisation comporte quatre sous cas d'utilisation qui sont :

1. interrogation par date ;
2. interrogation par invité ;
3. interrogation par animateur;
4. interrogation par type du journal télévisé.

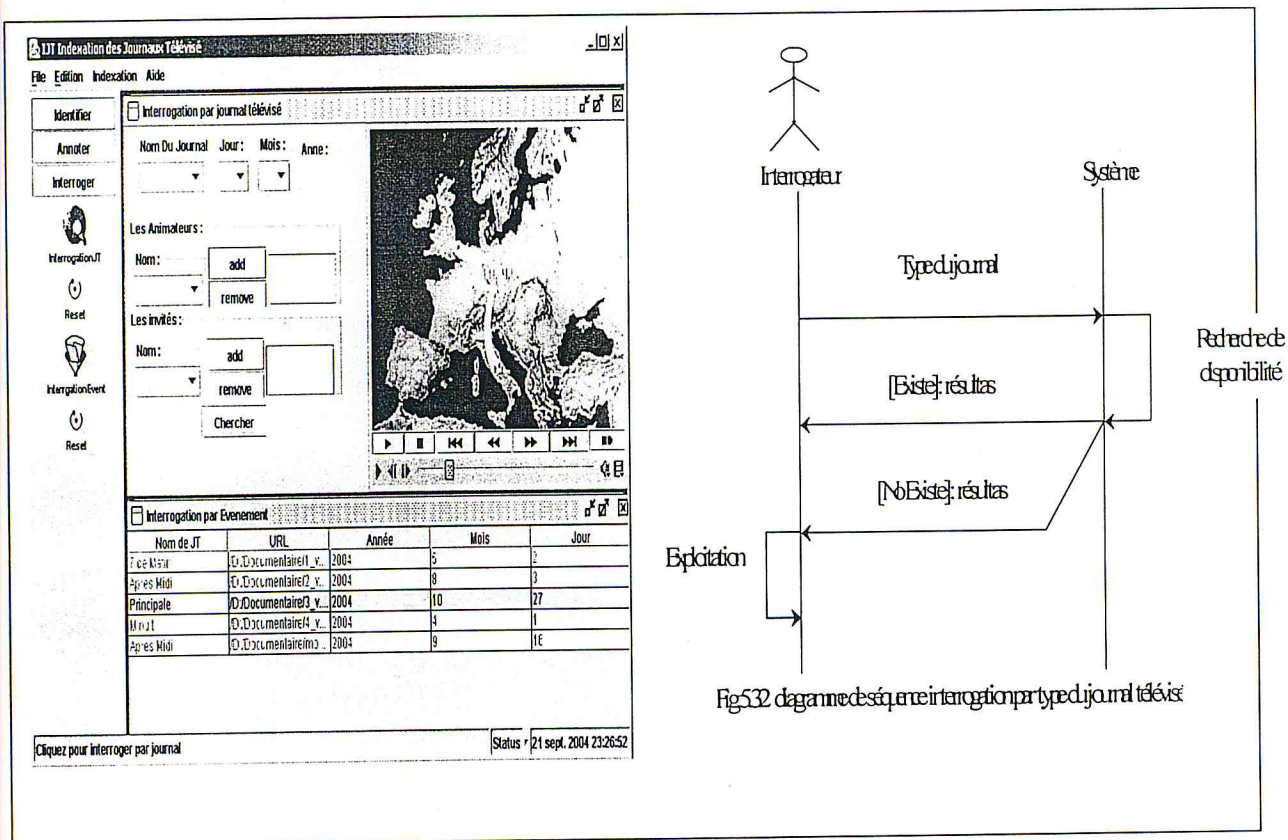


Fig.6.18 validation du cas d'utilisation interrogation par journal télévisé

**b) Validation du cas d'utilisation « interrogation par événement » :**

C'est presque comme l'interrogation par journal télévisé, à l'exception que le rendement n'est pas le même, l'interrogation par événement rend une /des séquences vidéo tirées par journal, par contre l'interrogation par journal télévisé rend un ou des journaux complets.

Le cas d'utilisation interrogation par événement engendre les quatre sous cas d'utilisation suivants :

1. interrogation par lieu ;
2. interrogation par acteur ;
3. interrogation par date

## 4. interrogation par mot clé.

**6.7 Test :****6.7.1 Introduction :**

Ensemble des mesures et des activités qui visent à garantir le fonctionnement satisfaisant du logiciel. Du fait de la complexité liée à l'élaboration des logiciels il y a fort peu de chances qu'un logiciel soit exempt de défauts, au contraire, le risque de déficience gravement élevé. L'informaticien est confronté du même problème, quelles que soient ses compétences, il n'est pas à l'abri d'un beugue les logiciels sont écrits par des humains, les humains sont faillibles, les logiciels aussi. [Mull01]

Tous les défauts n'ont pas les mêmes conséquences. Les défauts d'analyse se traduisent par des logiciels qui ne satisfont pas leurs utilisateurs. Les défauts de conception génèrent des logiciels lourds à manier, trop lents ou peu efficaces. Les défauts de réalisation induisent des comportements non prévus. Tout défaut est indésirable ; ceux introduits dans les activités en amonts restent cependant les plus difficiles à corriger. [Mull01]

**6.7.2 Démarche du test :**

L'utilisation principale du système d'archivage du téléjournal est le processus de la recherche dans les téléjournaux pour des futures productions (reportages, magazines, documentaire, etc.). L'identification, l'annotation et l'interrogation dans les journaux télévisés constituent trois processus fortement interconnectés et complémentaires.

Par la suite, nous allons tester notre outil d'indexation des journaux télévisés.

Nous avons choisi le canal « MADJD » Arabie saoudite, son journal est riche en matière d'informations locales et internationales, nous avons enregistré le journal de l'après midi du 16 septembre 2004, après numérisation du journal et transformation sous un format de document vidéo enregistré sur ordinateur, le traitement est le suivant : tout d'abord l'identification du journal.

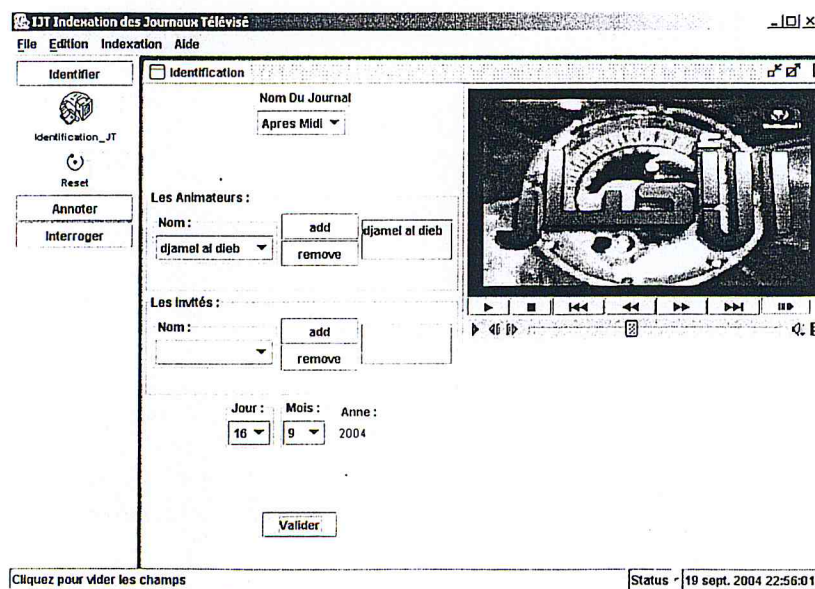


Fig. Identification du journal du canal MADJD



Nous avons saisi les informations pour l'identification du journal, réalisation de l'opération avec succès.

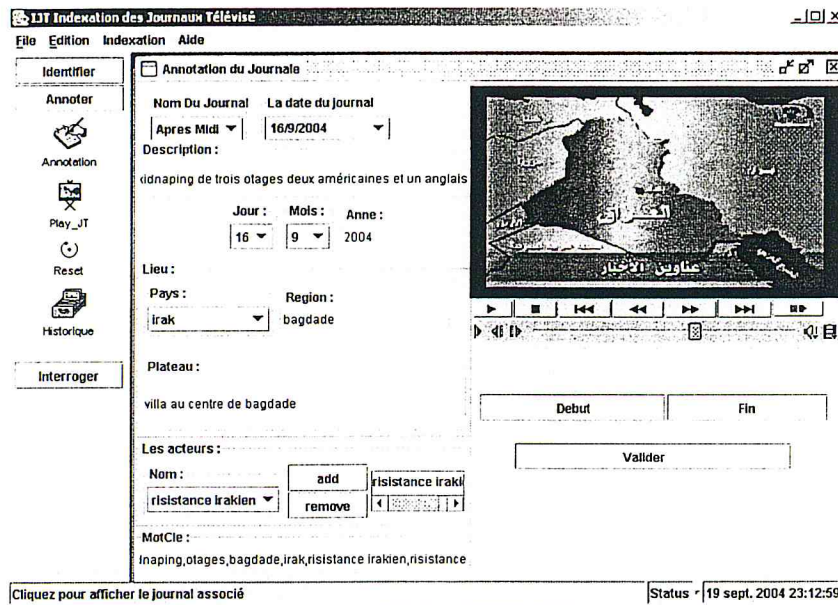


Fig. annotation du premier événement

Après sélection du type du journal qui est "après midi" et la date qui est 16 / 9 / 2004, on lance le journal par le Play\_jt, on sélectionne le début et la séquence et la fin.

Le premier événement est comme suit :

Description : kidnapping de trois otages deux américaines et une british.

Pays : Irak ;

région : Bagdad ;

Plateau : villa au centre de Bagdad ;

Les Acteurs : résistants irakiens.

Mot clés : kidnapping, otages, Bagdad , résistance irakien ...etc.

De la même manière les autres événements sont annotés, on passe maintenant au volet d'interrogation par journal.

On sélectionne la jour du journal «16», on lance la recherche, le programme cherche tous les journaux du jour '16' dans tous les mois et les années, puis le résultats s'affiche dans la table des résultats dans la fenêtre ci-après, en cliquant sur la ligne du résultats le journal s'affiche .



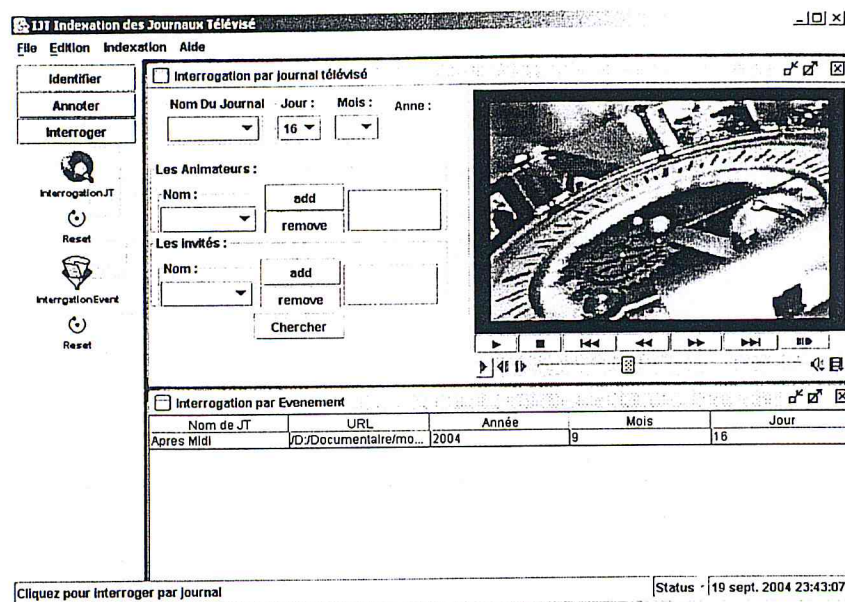


Fig. interrogation par journal télévisé

On passe vers le processus l'interrogation par événement. De la même manière on sélectionne la fenêtre d'interrogation par événement, on saisit dans le champ mot clé le mot «kidnapping», le résultats s'affiche de la même manière dans la table, en cliquant sur la ligne la séquence s'affiche.

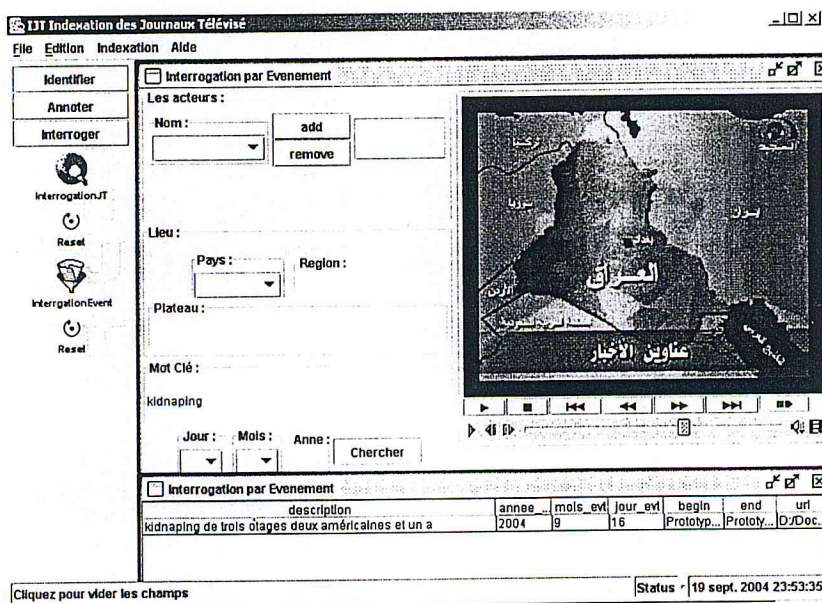


Fig. interrogation par événement

**6.7.3 Conclusion :**

Cette partie montre la réalisation des besoins des utilisateurs, établis dans le cahier des charges, et la vérification de l'implémentation du logiciel avec un exemple réel utilisant ces diverses fonctionnalités,

## Conclusion générale :

Notre projet IJT (Indexation des Journaux Télévisés) concerne la recherche d'informations audiovisuelles. Son rôle est la réalisation d'une plate-forme prototype d'indexation et d'annotations des Journaux télévisés.

Le projet vise à concevoir et à mettre en œuvre un système de librairie audiovisuelle pour différents utilisateurs, tels que journalistes, documentalistes, producteurs, voire des chercheurs. Pour des raisons d'organisation, la démarche de notre étude avait besoin d'une méthodologie de développement.

Un processus de développement d'un logiciel a pour objectif la formalisation des activités à l'élaboration des systèmes informatiques[Mull01]. Pour cela nous avons choisi le modèle en cascade comme un modèle de développement avec le langage de modélisation UML.

Ce mémoire a comme objectif le développement et l'analyse d'un outil d'indexation des journaux télévisés, afin d'accéder à des contenus sémantiques dans les documents vidéo.

Nous avons développé une interface d'annotation pour un système d'indexation et de recherche des séquences audiovisuelles à large échelle. Cette interface est fondée sur une approche qui utilise l'indexation sémantique pour l'interrogation et l'annotation ainsi qu'une indexation structurelle pour la navigation dans les journaux télévisés, en adoptant un modèle de donnée spécifique à un centre d'intérêt «News».

Notre logiciel est destiné à des utilisateurs qui font la recherche d'information audiovisuelle dans le domaine des journaux télévisés, afin d'exploiter les bénéfices offerts par ce logiciel. Ces bénéfices sont répartis sur différents types par exemple un journaliste a comme objectif de réaliser un reportage sur une célébrité ou un phénomène, un documentaliste spécialisé dans la diffusion des documents perçoit ces bénéfices dans la conception d'un documentaire de nature ou d'histoire, de la même manière un producteur exploite l'outil pour réaliser un film en adoptant quelques séquences, etc.

Dans le cadre du projet «IJT», nous avons développé une solution complète (du serveur au client) de recherche et d'accès aux documents audiovisuels tout en apportant des alternatives aux problèmes non résolus dans le domaine d'indexation audiovisuel. Notre logiciel permet une visualisation de la séquence visée d'une façon directe. L'utilisateur décide de l'importance de cette séquence selon son objectif personnel.

Grâce à une interface ergonomique, l'utilisateur peut annoter un journal télévisé sans avoir besoin d'une grande connaissance technique en informatique.

La manipulation du logiciel est soutenue par des interfaces graphiques qui facilitent l'utilisation des diverses fonctionnalités via des composants graphiques simples, bien définis, qui donnent à l'utilisateur l'impression de la simplicité d'utilisation.

Les résultats obtenus par le projet IJT suscitent un sérieux intérêt auprès des équipes de production et d'archivage des entreprises de télévision. La plate forme IJT



représente un environnement intéressant de teste et de validation d'expériences d'indexation et de recherche des données audiovisuelles. Cependant notre expérience avec l'indexation des journaux télévisés a révélée quelques perspectives afin d'enrichir la plate forme. Ces perspectives sont synthétisées dans les points suivants :

1. Élargir le centre d'intérêt (pas seulement les journaux télévisés mais aussi d'autres thèmes comme la météo, le football, ...etc ) d'une façon automatique via l'apprentissage des ontologies.
2. Opter pour une approche d'indexation automatique (traitement du signal, reconnaissance de la parole et de la forme, analyse du langage naturel).
3. Réduire la charge du réseau en intégrant la technologie des agents mobiles et des agents intelligents.
4. Développer le logiciel en utilisant un SGBDR puissant comme ORACLE.

### **Critique personnelle :**

Le serveur vidéo du projet «IJT» est un prototype simple, qui a comme objectif de satisfaire les besoins utilisateurs et de fournir des video par la technique streaming<sup>1</sup>. l'API JMF de Sun microsystems présente un certain nombre de problèmes, en effet cette API est encore en cours de développement, et l'on peut espérer l'apparition de nouvelles versions plus stables dans un future proche. Les différents problèmes liés à cette API ont engendré un certain nombre d'anomalies au niveau du serveur vidéo (comme par exemple une mauvaise qualité de réception, la nécessité de redémarrer le serveur dans certains cas ..etc.).

La base de données présente l'inconvénient d'une faible tolérance aux fautes et d'une performance limitée. En effet, si la base de données est endommagée, toutes les données comprises seront inaccessibles. D'un autre coté si les données de la base deviennent très volumineuses, cela conduit à une dégradation des performances de recherche et d'accès.

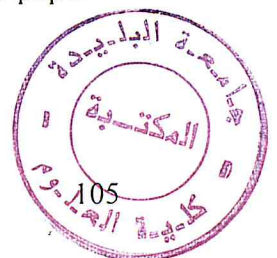
---

<sup>1</sup> Voir annexe C



# Bibliographie :

- [SICH02] Chapitre 6 sirsale :un Système d'Indexation et de recherche de séquences audiovisuelles à large échelle., Ahmed Mostfaoui de Franche Comté à Montbéliard & Lionel Brunie LIRIS, INSA de Lyon
- [ChFr00]: Chabane Djeraba \_ François Pachet « Objet et Multimédia volume 6 n° 2/2000 » Edition HERMES Science
- [walda01] Walide Dabbous « Systèmes multimédias communicants » Edition HERMES Science »
- [DecF99] Declochez W., Duluc F., « Conception d'un environnement multimédia pour la documentation technique aéronautique », Rapport de stage ,Aérospatiale Matra airbus, Université Paul Sabatier-Toulous III, Toulous juin 1998.
- [Most00] Ahmed Mostefaoui, Conception et mise en oeuvre d'un serveur parallèle de séquences audiovisuelles, Mémoire de Doctorat 2000, laboratoire de l'informatique du parallélisme, Lyon
- [MocF97] Mocellin F., Gestion des données et de présentation multimédia par un SGBD à objets, Doctorat d'informatique , université Joseph Fourier, 1997
- [Bmsr03]: Ben Mokhtar Sonia Ratiba, La génération automatique de code dans le cadre d'un système d'indexation et de recherche de séquences audiovisuelle à large échelle. Mémoire d'ingénieur 2003,INI
- [LayM99] ] Layaida N, Madeus : Un système d'édition et de présentation de documents structurés multimédias, *Doctorat d'informatique*, Université Joseph Fourier, janvier 1999
- [Sab199] Sabry – Ismael L. Schéma d'exécution pour les documents multimédias distribués.Doctorat d'informatique, Université Joseph Fourier, Grenoble,janvier 1999.
- [BacB99] Bachimount, B., «Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en Ingénierie des connaissances », In J, Charlet, M. Zacklad,& G . Kassel (Eds), *Ingénierie des connaissances* : Eyrolles,1999
- [ChaM00] Chabin, M.-A., Le management de l'archivage, PARIS , Hermès, 2000
- [RogC96] : Le grand poche JAVA 1.1  
Roger CODENHEAD - Edition : Simon & Schuster Macmillan,1996
- [CSHG00] :Au cœur de JAVA 2 Notions fondamentales  
édition CompusPress juin 2000
- [Mull97] Pierre-Alain Muller, modélisation objet avec UML, EYROLLES, 1997
- [Mull01] Pierre-Alain Muller, modélisation objet avec UML, EYROLLES, 2001.
- [Pill02] Jean-François Pillou, Introduction à la programmation orienté objet, 2002disponible sur  
l'adresse :<http://www.commentcamarche.net/poo/poointro.php3>.
- [BerF02] F. Bernardi, Méthode d'analyse orienté objet UML, 2002



- [FaHe00] Rémy Fannader, Hervé Lerroux, UML principes de modélisation, DUNOD, 2000.
- [kett01] kettani N, Mignit D.Paré P.,Rosenthal-Sabroux C, De Merise à UMI, Ed. Eyrolles, Paris,2001.
- [Bres93] J.bres, ateliers de genie logiciel, masson, 1993
- [SomL88] Lan Sommerville, le génie logiciel et ses applications, paris, 1988
- [GabJ98] Joseph Gabay, Merise vers OMT et UML, Masson 1998
- [Jguy99] J. Guyot. Les classes et les objets . Université de Genève 1999
- [DjMs03] Messaouda Farah, Fatima Boumahdi ,Analyse et développement d'un outil de définition de workflow basé sur le langage EXPRESS, application au commerce électronique.mémoire ingénieur Blida 2003
- [Anmi] Antoine Mirecourt Le développeur JAVA 2 édition 1999 fichier pdf
- [PaAl] Pascal André, Alain Vailly ,Spécification des logiciels, Ellipses 2001

### Liste des ouvrages:

Titre de livre	Masson d'édition	L'écrivain	L'année
Pensez en JAVA			
Programmer en java :	École nationales des sciences géographiques		septembre 2002
Programmation par objets en JAVA :	Université de rennes	Pascale le certen , Lucien ungaro	janvier 1998.
JAVA et le Multimédia	Ellipses 2001		juin 2003
The Java(tm) Tutorial :A practical guide for programmers	<a href="http://www.java.sun.com">www.java.sun.com</a>		Mars 2002
Analyse et conception des Systèmes d'Information objets :	Université de Bourgogne.		
Le Modèle Entité-Association et les Bases de Données Relationnelles :	Université Joseph Fourier	Didier DONSEZ	
RMI Remote Method Invocation:	Université de Valenciennes	Hafid Bourzoufi, Didier Donsez	
JDBC and DATABASE CONNECTION POOLING	<a href="http://www.coreservlets.com">www.coreservlets.com</a>		
Distrebuted Video	Frenche comity	Ahmed Mestfaoui	

Documents Indexing and content-Based :	science lab montbélard , france	and loyce favory	
La recherche d'information dans : les archives du téléjournal à la TSR ,le projet COALA	Flash Informatique Université EPLF	Nastaran.Fatemi. Omar.AbouKhaled Giovanni.Coray	avril 2003

## Les sites web:

[www.experts-exchange.com](http://www.experts-exchange.com)  
 [01] [www.commentcamarche.net](http://www.commentcamarche.net)  
[www.mhp-interactive.org](http://www.mhp-interactive.org)  
[www.moteurprog.com](http://www.moteurprog.com)  
[www.developpez.com](http://www.developpez.com)  
[www.java.sun.com](http://www.java.sun.com)  
[www.javaworld.com](http://www.javaworld.com)  
[www.wanadoo.fr](http://www.wanadoo.fr)  
[www.onjava.com](http://www.onjava.com)  
[www.eteks.com](http://www.eteks.com)  
[www.javafr.com](http://www.javafr.com)  
[www.umlfr.com](http://www.umlfr.com)  
[www.ina.fr](http://www.ina.fr)  
[www.mysql.com](http://www.mysql.com)



# Annexe A : Manuel d'utilisation du logiciel (IJT).

## A. Définition :

L' IJT (Indexation des Journaux Télévisé) est un logiciel d'indexation audiovisuelle des journaux télévisés qui comporte une interface graphique, cette dernière permet aux utilisateurs d'annoter des séquences audiovisuelles et d'interroger une base de données à propos des besoins des utilisateurs, la manipulation de ce logiciel n'est pas du tout compliquée, cependant elle nécessite un certain nombre de connaissances à travers l'utilisation, coté annotation, l'utilisateur doit être un journaliste (en premier lieu ), un archiviste, un documentaliste...etc

Coté interrogation n'importe quel type d'utilisateur peut bénéficier de l'IJT, et peut exploiter ces bénéfices à sa manière (l'utilisateur) exemple : composer un document vidéo sur une célébrité, un documentaire d'un événement,...etc.

## B. support du logiciel :

L'IJT nécessite une plate forme pour un bon fonctionnement la liste est comme suit :

- .un traducteur java (j2sdk1.4.0), pour vérifier la version former la commande suivant :

```
C:\> java -version
C:\>java(tm) 2 Runtime Environement , Standard Edition (build1.4.0-b92)
      Java HostSpot (tm) Client VM (build 1.4.0-b92, mixed mode)
C:\>
```

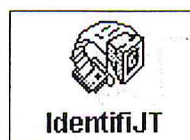
Fig. obtention de la version du traducteur java :j2sdk1.4.0

- .l'API java : JMF(Java Media Framework) sous forme un package "media" de la bibliothèque Swing ,pour cela il faut installer le logiciel JMFStudio de SUNmicrosystems 2003 la version JMF2.1.2.e 2003.
- L'API java : JDBC (Java DataBase Connectivity) avec l'installation du pilot MySQL la type : Le pilote mm : Un pilote de type 4 JDBC par Mark Matthews exactement la version : mm.mysql-2.0.2. (recommandé).

## C. scénario d'utilisation :

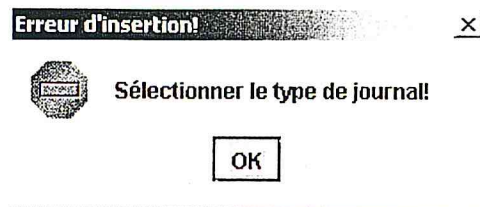
### **C.1 tâche d'identification :**

l'interface responsable de cette tâche est la fenêtre « identification des journaux télévisé »,l'accès a cette fenêtre se fait par le menu indexation / identification\_JT, où en appuient sur le bouton « identification\_JT »qui se trouve dans l'explorateur d'outil.



FigA.1 identification des journaux

Il faut visualiser un journal à laide de menu fichier/ouvrir à travers une boite de dialogue ensuite, l'utilisateur choisit un document multimédia, l'utilisateur remplit les champs d'identification et dans la dernière étape il valide le choix, dans cette interface il existe des champs obligatoires et d'autres sont optionnels, pour des raisons de recherche structurelles. Si un des champs obligatoires n'est pas rempli, un message d'erreur est généré qui affirme le nom du champ manqué, comme la présente la figure suivante :



FigA.2 message d'erreur :manque du type du journal

### C.1.1 Les champs obligatoires :

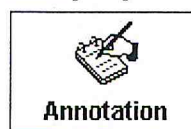
1. Type du journal ;
2. les Animateurs ;
3. La date du journal.

### C.1.2 Les champs optionnels :

1. Les invités du journal ;

## C.2 tâche d'annotation:

L'IJT fournit une interface dénommée « annotation du journal télévisé », et l'accès à cette fenêtre se fait par deux manières soit par le menu indexation / annotation, ou par l'explorateur d'outil avec le bouton « annotation », de la même manière dans cette interface il y a des champs obligatoires et des champs optionnels.



FigA.3 annotation

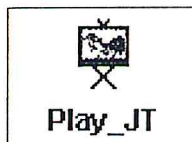
### C.2.1 Les champs obligatoire :

1. Le type du journal ;
2. La date du journal ;
3. La description ;
4. Les mots clé (pas moins de trois caractères) ;
5. Le début de la séquence ;
6. La fin de la séquence ;

### C.2.2 Les champs optionnels :

1. Le lieu (pays, région, plateau) ;
2. Les acteurs.

Dès le choix du type du journal, la date est configuré automatiquement affichant toutes les dates du type du journal sélectionné existant dans la base de données, ensuite en appuyant sur le bouton « Play\_JT » le journal s'affiche automatiquement dans le lecteur vidéo.



FigA.4 le bouton Play\_JT qui lance le vidéo dans la fenêtre annotation

Et en fin il faut valider les informations via le bouton valider.

### **C.3 tâche d'interrogation:**

Cette tâche engendre deux sous tâches selon des besoins de l'utilisateur, la première tâche est l'interrogation par journal télévisé, la deuxième tâche est l'interrogation par événement. L'IJT fournit les services de ces deux tâches sous forme deux fenêtres.

#### **C.3.1 tâche d'interrogation par journal télévisé :**

L'accès à cette interface se fait par deux manières, soit par le menu indexation / interrogation par JT. Le résultats de la recherche se traduit par un journal complet. Tout les champs de cette interface sont optionnels c'est à dire que l'utilisateur peut remplir n'importe quel champs sans aucune exigence, et la recherche se lance via le bouton « chercher ».

Le résultatst présenté sous forme une fenêtre en bas dans une table, et l'affichage du document vidéo se fait par un simple click de souris sur la ligne du tableau spécifié.

#### **C.3.2 tâche d'interrogation par évènement :**

De la même manier L'accès à cette interface se fait par deux manière, soit par le menu indexation / interrogation par événement soit par un bouton dans l'explorateur, cette interface est presque identique à l'interface d'interrogation par journal télévisé, le seul différence se résume dans le résultats, où dans l'interrogation par Journal télévisé le résultat est un journal télévisé, et dans l'interrogation par événement la résultats est une séquence vidéo .

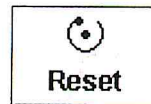


FigA.5 interrogation par événement



### C.4 la tâche « Reset » :

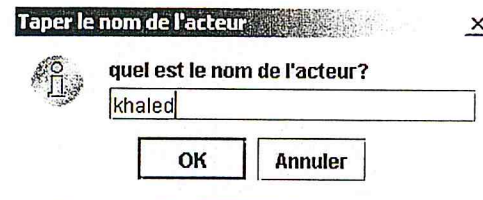
Cette tâche présente dans tout les fenêtre via le bouton « RESET », qui se trouve dans l'explorateur d'outil qui permet de réinitialiser les champ de la fenêtre active.



FigA6 le bouton RESET

### C.5 la tâche Nouveau :

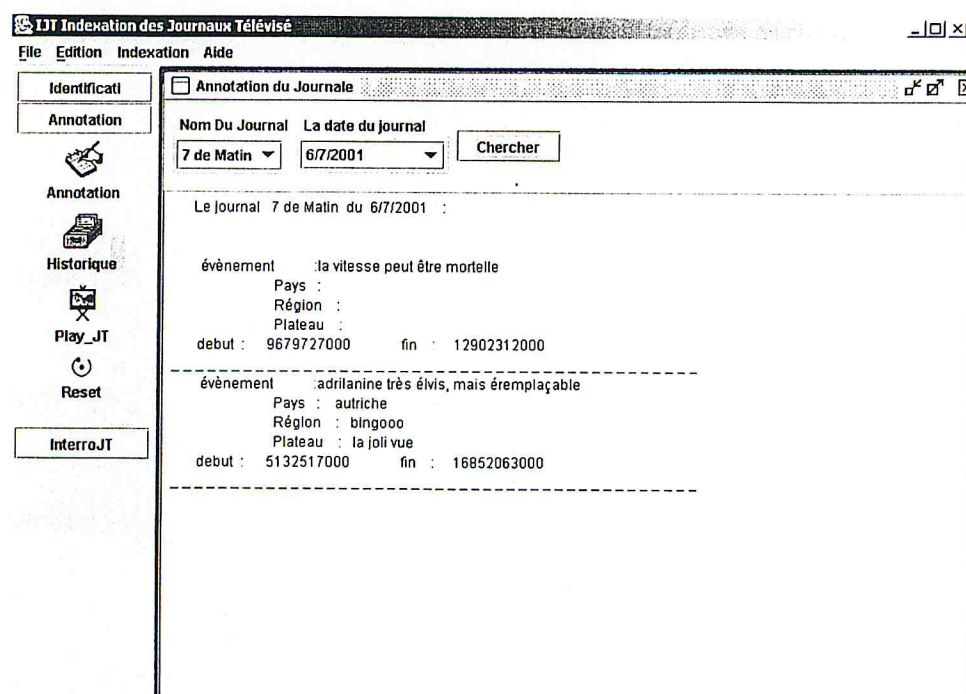
Cette opération existe dans des champs spécifiques comme les acteurs, les animateur,...etc. son rôle consiste à insérer une nouvelle donnée qui peut être répétitive comme le nom d'un invité, au lieu de remplir à chaque fois le nom, l'utilisateur n'a pas à glisser le glisseur de la liste associée et choisir le nom associé s'il existe si non il sélectionne « Nouveau », une boîte de saisie s'affiche comme le montre la figure suivante ou l'utilisateur saisit la nom associé pour que le nom puisse être utilisable ultérieurement.



FigA.7 une boîte de saisie de l'acteur

## L'interface des annotations contenues dans un journal

Cette interface permet d'afficher les événements d'un journal télévisé choisi à partir d'une liste glissée qui fournit le type du journal et d'une autre liste qui retourne la date de tous les journaux du type choisi auparavant, le résultat s'affiche dans une zone de texte, cette interface est présentée sous forme d'une interface graphique, l'accès à cette interface se fait par un bouton dans l'explorateur.



FigA.8 les événements d'un journal

## Annexe B :

## PRESENTATION DE JAVA

### Définition :

JAVA est un langage de programmation dont le squelette principal est constitué du langage C++. Son atout majeur réside dans l'efficacité de son utilisation sur le WEB et la réalisation de jeux fortement basée sur le graphisme. En effet, il permet la création de graphismes élaborés, animés, ainsi que la présentation de texte évolué (hypertext). Le développeur utilisant JAVA, acquiert une technique méthodique car le langage est très exigeant sur la structure générale du code (contrainte et rigueur d'écriture du code). JAVA est sans aucun doute incontournable pour tous les amateurs du WEB, de même que pour la demande croissante de la connaissance du langage par des entreprises désirantes être à la pointe de la technique.

### Historique :

JAVA a été inventé, il y a une vingtaine d'années par James Gosling, ingénieur chez Sun Microsystems. Il s'appelait à l'époque OAK (chêne) à cause de l'arbre qu'il voyait de la fenêtre de son bureau. Sun Microsystems avait prévu de gagner beaucoup d'argent en destinant ce nouveau langage à des programmes pour appareils intelligents (télévision interactif, fours omniscients..). Mais malheureusement, les prévisions furent désastreuses jusqu'au jour où le WEB prit son envol. Le langage OAK est adapté par l'équipe de développeurs de Sun Microsystems afin de permettre l'exécution des programmes en toute sécurité dans un environnement réseau.

### LE JDK

Le JDK (JAVA Développeur's Kit) est un ensemble d'outils permettant d'écrire et de mettre au point des programmes en JAVA, mais ce n'est pas un logiciel à environnement graphique. En effet, l'exécution des programmes se fait à partir de la ligne de commandes Ms-dos même si votre environnement de travail se situe sous windows. Le JDK n'est pas indispensable, il existe d'autres outils de développement JAVA. Il est actuellement disponible chez JAVASOFT pour des systèmes tels que :

- \* Microsoft Windows NT.
- \* Windows 95.

Actuellement, il n'est pas disponible pour des systèmes tels que APPLE Macintosh.

Le JDK fournit un ensemble de programmes dont les principaux sont :

- \* Le compilateur JAVAC qui traduit un fichier code JAVA en fichier de code directement utilisable par le micro.
- \* L'interpréteur JAVA exécute les programmes créés par le compilateur.
- \* L'outil de navigation WEB appletviewer permet d'exécuter des programmes JAVA conçus pour tourner sur le WEB. [RogC96]

### les avantages de JAVA :





L'un des avantages évidents de ce langage est une bibliothèque d'exécution qui se veut indépendante de la plate-forme : en théorie il vaut est possible d'utiliser le même code pour Windows95 / w98 / wNT ...etc, Solaris, Linux, Mac,..etc. cette propriété est indispensable pour une programmation sur Internet ( cependant , par rapport à la disponibilité sur Windows et Solaris, les implémentations sur d'autres plates-formes ont toujours un léger décalage)

Un autre avantage de ce langage de programmation réside dans le fait que la syntaxe du java est analogue a celle de C++, ce qui le rend économique et professionnel, les programmeurs visual Basic risquent néanmoins de trouver cette syntaxe déplaisante, et regretteront certaines constructions de visual Basic telles que SELECT CASE.

### Les propriétés du java :

La simplicité :

Java est un système de programmation simple, qui ne nécessite pas un apprentissage ésotérique et qui tire parti de l'expérience standard actuelle. En conséquence java a été conçu de façon relativement proche du langage C++ dans le dessein de faciliter la compréhension du système, de nombreuses fonctions compliquées, mal comprises, rarement utilisées de C++, qui nous semblaient par expérience apporter plus d'inconvénients que d'avantages [RogC96], ont été supprimées de java.

L'orienté objet :

Au cours des trente dernières années la programmation orienté objet a prouvée ses avantages, et il est inconcevable qu'un langage de programmation moderne n'en tire pas parti, en fait , les caractéristiques orientées objet de java sont comparables à celle de C++.

Les déférences majeurs résident dans l'héritage multiple( pour le quelle java à trouver une solution plus appropriée), et le modèle objet de java (méta classe). Le mécanisme de réflexion, ainsi la fonction de sérialisation des objets facilite énormément la mise en ouvre des objets persistent et des générateur GUI capables d'intégré des composants réutilisables.

La distribution :

Java possède une importante bibliothèque de routine permettent de gérer les protocole TCP-IP tell que http et ftp .les applications java peuvent charger et accéder à des fichier sur Internet via des URL avec la mêmè facilité qu'elles accèdent à un fichier local sur le système.

La Fiabilité :

Java a été conçu pour que les programmes qu'ils utilisent soient fiables sous différents aspects, sa conception encourage à traquer préventivement les éventuels problèmes, à lancer des vérifications dynamiques en cours d'exécution et à éliminer les situations génératrices d'erreurs. La seule et unique grosse différence entre C++ et java réside dans le fait ce dernier intègre un modèle de pointeur qui écart les risques d'écrasement de la mémoire et d'endommagement des données.

La sécurité :



Java a été conçu pour être exploité dans des environnements serveur et distribué dans ce but, la sécurité n'a pas été négligée, java permet la construction des systèmes inaltérables et son virus.

La neutralité de l'architecture :

Le compilateur génère un format de fichier objet dont l'architecture est neutre – le code compilé est exécutable sur de nombreux processeurs, à partir du moment où le système d'exécution de java est présent pour ce faire, le compilateur java génère des instructions en ByteCode qui n'ont de lien avec aucune architecture d'ordinateur particulière. Au contraire, ces instructions ont été conçues pour être à la fois faciles à interpréter, quelle que soit la machine, et faciles à traduire à la volée en code machine natif.

La portabilité :

A la différence du C et de C++, on ne trouve pas les aspects de dépendance de la mise en œuvre dans la spécification. Les tailles de type de données primaires sont spécifiées, ainsi que le comportement arithmétique qui leur est applicable.

L'interprétation :

L'interpréteur de java peut exécuter les ByteCode directement sur n'importe quelle machine sur la quelle il a été porté. Dans la mesure où la liaison est un processus incrémentiel et léger, le processus de développement peut se révéler plus rapide et exploratoire.

Performances élevées :

En règle générale, les performances des bytecode interprétées sont tout a fait suffisant ; Il existe toute fois des situations dans les quelles des performances plus élevées sont nécessaires les bytecode peuvent être traduites à la volée ( en cours d'exécution) en code machine pour l'unité centrale destinée à accueillir l'application.

La multithread :

Les avantages de multithread sont une meilleure interréactivité et un meilleur comportement en temps réel.

Java, langage dynamique :

Sur plusieurs points, Java est un langage plus dynamique que C ou C++. Il a été conçu pour s'adapter à un environnement en évolution constante. Les bibliothèques peuvent ajouter librement de nouvelles méthodes et variables sans pour autant affecter leurs clients. La recherche des informations de type exécution dans Java est simple. [CSHG00]

## Annexe C : API JMF

### Définition de l'API - JMF:

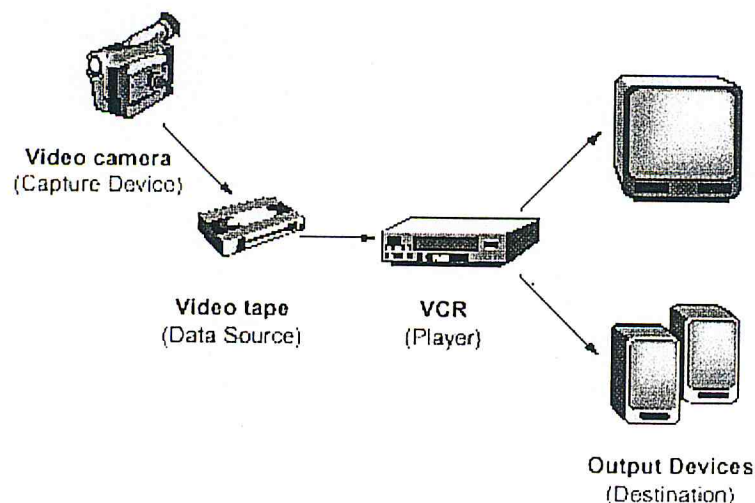
JMF (Java Media Framework) est une technologie développée par Sun Microsystems, JMF fournit une architecture unifiée ainsi qu'un protocole de messages pour réaliser l'acquisition, le traitement ainsi que la distribution de données media temporelles. JMF supporte la plupart des formats standard, En fait il est même possible *via* un mécanisme de plugins, de lire quasiment tous les formats existants. Mais les formats supportés par défaut satisfont déjà largement aux besoins de leur utilisation.

Le paquetage JMF répond aux exigences suivantes :

- Faire abstraction des médias utilisés grâce aux classes de haut niveau de l'API JMF.
- Pouvoir répondre à des besoins plus spécifiques grâce aux classes de bas niveau: il est ainsi possible de personnaliser les échanges RTP ou encore intégrer de nouveaux formats (exemple : pour le MP4).

### Architecture de JMF :

Les appareils tel que les radios cassettes ou les magnétoscopes fournissent un modèle familier pour l'enregistrement, le traitement et la présentation des medias temporels. Quand vous diffusez une vidéo en utilisant un magnéscope, vous fournissez le flux d'informations au magnéscope en insérant une vidéo cassette. Le magnéscope lit et interprète les données sur la cassette et envoie les données appropriées à votre téléviseur ou votre micro.

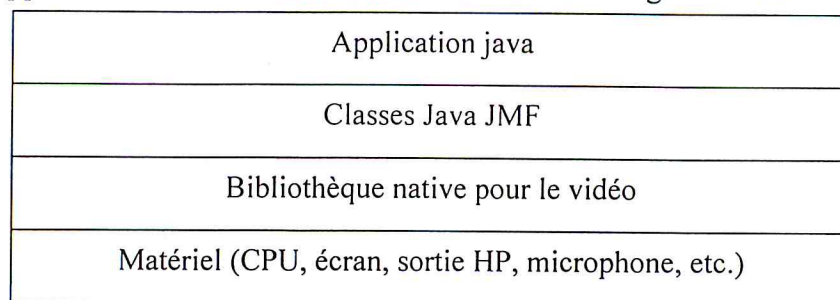


JMF utilise le même modèle de base. Le lecteur qui traite les données audio ou vidéo est repéré dans Java Media Framework par des références de l'interface `javax.media.Player`. Il prend en entrée des données audio ou vidéo et les envoie en sortie vers les périphériques adéquats ( carte son, écran ). Il le chargement,

l'acquisition des ressources multimédia, l'exécution ( démarrage, arrêt, vitesse d'exécution ...) et les sorties sonores et visuelles d'un document multimédia.

Une DataSource (source de données) enregistre le flux de données un peu à la manière d'un magnéto-scope et un *player* (lecteur) fournit le traitement et les contrôles de mécanismes similaires à ceux du magnéto-scope. Diffuser et capturer les données audio et vidéo avec JMF nécessite les sorties et entrées appropriées comme des microphones, des caméras, des haut-parleurs et un écran.

Les DataSource et les *players* font partie intégrante de l'API de haut niveau de JMF pour réussir la capture, la présentation, et le traitement de médias temporels. JMF fournit aussi une API de niveau inférieur qui supporte l'intégration de nouveaux composants de traitement et d'extension. Cette couche inférieure permet aux développeurs Java d'insérer, grâce à une API facile d'utilisation, des médias dans leurs programmes Java tout en maintenant la flexibilité et l'extensibilité requises pour supporter les applications média avancées et les futures technologies médias.



FigC.1 Pile matériel et logicielle d'un programme JMF

### Lire un média avec l'interface de visualisation

Pour présenter un média audio ou vidéo avec JMF on utilise un *player* (lecteur). La réécoute peut être contrôlée par la programmation, où on affiche un panneau de contrôle (panel control) qui permet à l'utilisateur de contrôler la réécoute de manière interactive.

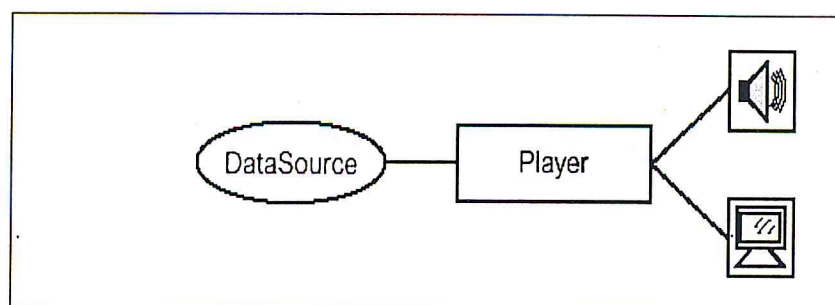


Fig C.2 Modèle du Player JMF

Un *player* traite un flux entrant de données et le rend à un temps précis. Une DataSource est utilisée pour délivrer le flux au *player*. La destination rendue dépend du type de média que l'on est en train de diffuser.



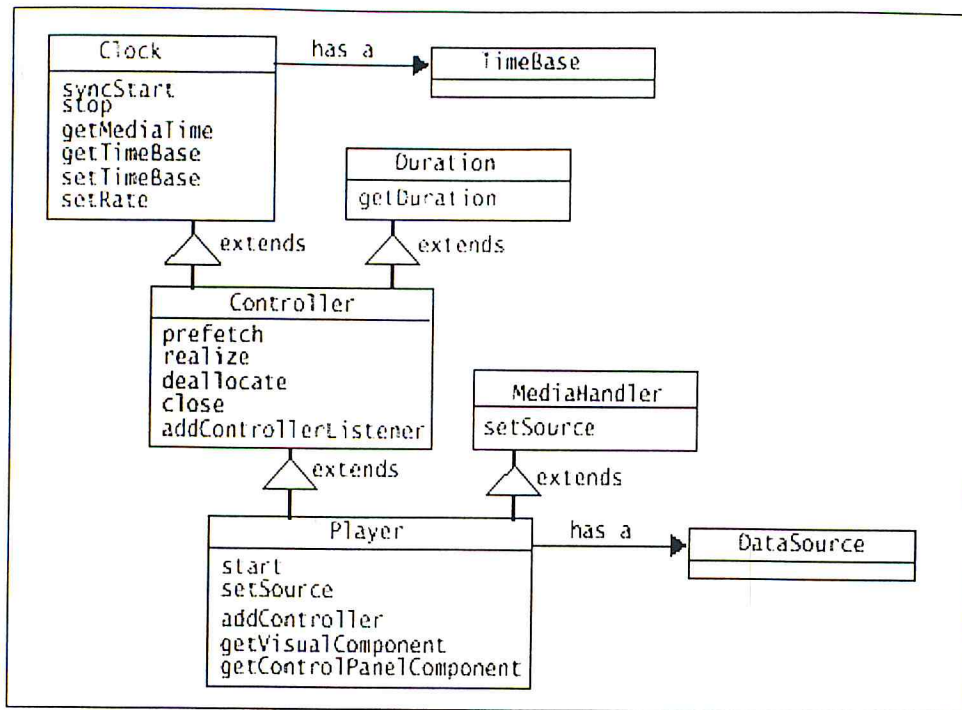


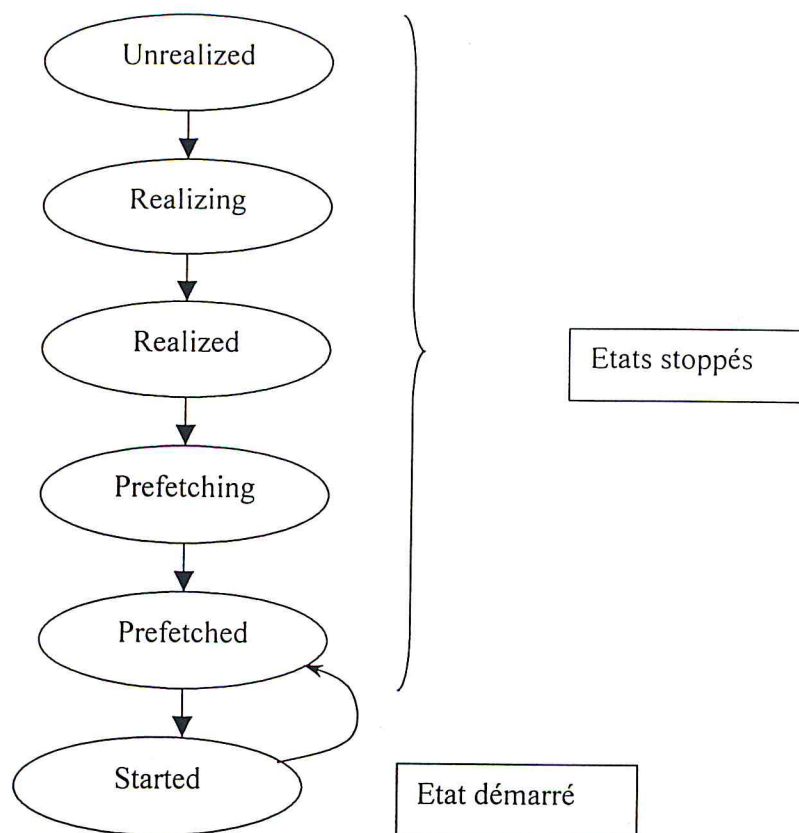
Fig C.3 Représentation UML du player[www.sun.com]

### Les états d'un Player :

Durant d'une utilisation d'un Player, un Player passe dans différents états. Ses états fondamentales sont les états Unrealized, Realized, Prefetched et Started :

- dans l'état *Unrealized*, le lecteur (Player) est un objet qui a été construit mais qui ne connaît rien de son environnement et des ressources qu'il devra traiter, aucune ressource ne lui a été attribuée ;
- dans l'état *Realized*, le Player connaît le type de document multimédia qu'il devra traiter et sait quelles ressources il devra acquérir. De ce fait, il peut d'ores et déjà préparer les composants visuels et sonores ainsi que l'interface graphique des boutons de commandes pour manipuler le document multimédia ;
- dans l'état *Prefetched*, le Player a acquis toutes les ressources pour traiter convenablement le document multimédia. Dans cet état, il est prêt à le présenter correctement à l'utilisateur ;
- dans l'état *Started*, l'affichage du document multimédia est en cours.

En fait, le passage d'un état fondamental à un autre peut prendre du temps, aussi il a été défini des états intermédiaires. Entre les états Unrealized et Realized, le lecteur (Player) est dans l'état Realizing. Entre les états Realized et Prefetched, le Player est dans l'état Prefetching. Dans l'état Realizing, un lecteur acquiert les ressources non exclusives. Ce sont principalement des ressources réseau. Dans l'état Prefetching, un Player acquiert toutes les ressources qui lui manquent, entre autres les ressources qu'il doit être seul à utiliser, dites ressources exclusives.



FigC.4 Etat d'un Player

### Définition de « Streaming » :

C'est une technique de transfert de flux multimédia en temps réel, la lecture d'un document vidéo peut s'effectuer avant la totalité de leur chargement. Pour cela nous avons besoin d'un protocole réseau adéquat.

Le premier protocole testé c'était le TCP :

Ce protocole n'est pas fiable avec la nature ou la structure des données multimédia, le TCP transporte les paquets et même les paquets corrompu, sachant qu'un vidéo se compose de 25 image /seconde, ce qui implique qu'une image défectueuse reflète sur la qualité du vidéo, TCP est un protocole qui peut servir dans d'autres domaines de données comme les données statistiques ou les images indépendamment avec leur affichage avec le facteur de temps.

L'autre protocole testé c'était l'UDP :

L'UDP est un protocole rapide, il assure la transmission de tous les paquets par contre il ne garantit pas l'arrivée de tous les paquets, en plus c'est un protocole de couche de transport.

Cependant, les besoins demandent un protocole qui manipule des couches supérieures à cause de la nature des données vidéo, la solution était d'utiliser le protocole RTP (Réal Time Protocole), puisque il garantit la notion de temps réel ainsi l'identification des données transportées, il transporte les paquets en bon ordre, ainsi permettant la synchronisation de flux.

Il peut se prouver sur n'importe quelle couche de transport mais il ne garantit pas la qualité de service, la raison d'utiliser le RTCP (Réal Time contrôle Protocole)

Le RTP et RTCP est utilisé pour le *Streaming* des flux vidéo depuis le serveur vidéo vers l'interface des utilisateurs.

La mise en œuvre du RTP, RTCP nécessite des API JMF afin d'implémenter la circulation des documents vidéo entre le serveur vidéo et les utilisateurs.

L'utilisation des API JMF/RTP a été mise en œuvre pour des besoins de compatibilité avec les fonctionnalités de JMF tel que : la capture, la restitution. Et voici une figure qui montre à quel niveau JMF intervient l'API RTP.

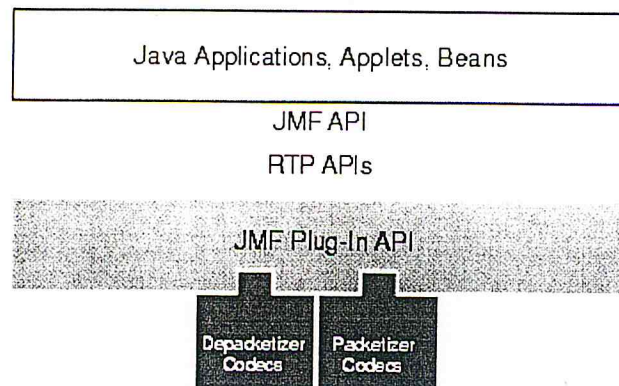


Fig.C.5 Architecture à niveau élevé de JMF RTP

#### Gestion des sessions RTP :

Le *SessionManager* maintient l'état de la session comme vu du participant local. en effet, le *SessionManager* est une représentation locale d'une entité distribuée, la session de RTP. Le *SessionManager* manipule également *le canal* de commande de RTCP, et soutient RTCP pour des expéditeurs et des récepteurs.

*SessionManager* L'interface définit les méthodes qui permettent à une application d'initialiser et commencer à participer à une session, pour enlever différents jets créés par l'application, et clôturer la session entière.

#### Transmission/Réception RTP :

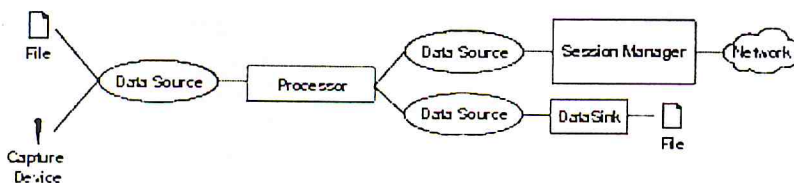


Fig C.6 Transmission du flux RTP en JMF

Au début, on capture un flux d'un document stocké, puis on encode le flux via l'objet *Processor* ensuite on l'envoie sous forme de flux RTP sur le réseau à travers l'objet *sessionManager*. A l'initialisation, le *SessionManager* se comporte comme un récepteur et envoie des paramètres RTCP Récepteur. une fois que le canal d'émission est initié via l'objet *SendStream*, il commence à envoyer des paramètres RTCP émetteur tant qu'au moins un canal reste ouvert, si tous les canaux sont fermés, le *SessionManager* reprend le statut récepteur passif. [SICH02]

La procédure à suivre pour créer un canal d'émission est la suivante :



- Créer, initialiser et démarrer un *SessionManager* pour la session RTP.
- Construire un *Processor* utilisant le *DataSource* de capture approprié
- Choisir un format de sortie du *Processor* spécifique au RTP. Pour cela un codec approprié doit être disponible pour le format de données désiré.
- Récupérer le flux de sortie du *Processor* dans un *DataSource*

Appeler la méthode *CreateSendStream* du *SessionManager* avec le *DataSource* en paramètre

A la réception du côté client le flux RTP est capturé par le *SessionsManager* qui alimente un *DataSource* afin que l'information soit restituée.

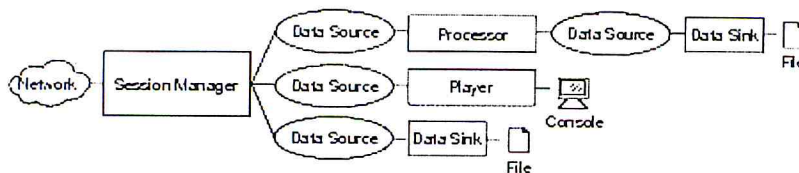


Fig C.7 Réception du flux RTP en JMF

C'est le *Player* qui se charge de restituer le contenu du flux RTP entrant ,pour recevoir et restituer un flux provenant d'une session RTP, il faut utiliser un objet *MediaLocator* qui spécifie les paramètres de construction du *Player*. [SICH02]

## **Annexe D : La Base de Données**

### **Définition de MySQL :**

MySQL est un Système de Gestion de Bases de Données (SGBD) fonctionnant sous Linux et Windows. Depuis la version 3.23.19, MySQL est sous Licence GPL (aussi bien sous Linux que Windows), ce qui signifie qu'il peut être utilisé gratuitement. *MySQL* est un véritable serveur de base de données SQL multi utilisateur et multi-threaded. Par exemple, vous pouvez utiliser le SQL pour récupérer des informations concernant un produit ou bien stocker des informations. Les principaux objectifs de *MySQL* sont la rapidité, la robustesse et la facilité d'utilisation. *MySQL* a été originellement développé pour puisse gérer des grandes bases de données de manière plus rapide que ce que pouvaient offrir les distributeurs de bases de données.

La base sur laquelle *MySQL* est construite est un ensemble de routines qui ont été largement éprouvées pendant des années dans un environnement de production exigeant. Même si *MySQL* est encore en développement, il propose déjà un ensemble de fonctionnalités riches et extrêmement utiles.

### **Définition du JDBC :**

Le JDBC (Java Database Connectivity) est un ensemble de classes (regroupées dans une interface) qui permettent d'accéder à des bases de données et d'exécuter des traitements sur ces données.

Le but de cette API (Application Programming Interface) est de fournir un niveau commun de connexion à toutes les bases de données, libre ensuite à chaque Editeur (MySQL, ORACLE, etc...) de fournir des API de plus haut niveau. Ceci nous permet d'écrire un programme de connexion à une base de données qui est indépendant de celle-ci.

Le JDBC supporte les fonctionnalités SQL de base, SELECT, INSERT, UPDATE, DELETE, Clause WHERE etc....

### **Le pilot de la base de données :**

Le pilot MySQL est de type : mm : c'est un pilote de type 4 JDBC par Mark Matthews exactement la version : mm.mysql-2.0.2.  
[<http://www.worldserver.com/mm.mysql/>].

### **La création de la base de données :**

Dans l'IJT le programme qui suit génère une chaîne d'opération afin de créer la base de données du système.

```
import java.sql.*;
```

```
public class DBCreate
```

```
{
```

```
static String[] identificationJT = {"drop table identificationjt","create table
identificationjt " +
```

```
"( idJT integer unsigned not null AUTO_INCREMENT , " +
```

```
    "nom_jt VARCHAR(15) not null, " +
    "url varchar (50) not null , "+
    "annee INTEGER(4) not null, "+
    "mois INTEGER(2) not null, "+
    " jour INTEGER(2) not null, "+
    "PRIMARY KEY(idJT))"
```

```
    } ;
```

```
static String[] animer = {"drop table animer","create table animer " +
```

```
    "(nom_animateur VARCHAR(30) not null, " +
    "idJT integer unsigned not null,"+
    "PRIMARY KEY
```

```
(nom_animateur,idJT))"
```

```
    } ;
```

```
static String[] invite = {"drop table invite","create table invite " +
```

```
    "(nom_inviti VARCHAR(30) not null, " +
    "idJT integer unsigned not null,"+
    "PRIMARY KEY (nom_inviti,idJT))"
```

```
    } ;
```

```
static String [] annotation = {"drop table annotation","create table annotation"+
```

```
    "(id integer unsigned not null AUTO_INCREMENT , " +
    "description varchar (100) not null,"+
    "pay    varchar(30) ,"+
    "region  varchar (30),"+
    "plateau  varchar (50),"+
    "annee_evt INTEGER(4) not null, "+
```

```
    "mois_evt INTEGER(2) not null, "+
```

```
    " jour_evt INTEGER(2) not null, "+
    "begin BIGINT (30) not null, "+
```

```
    " end BIGINT (30) not null, "+
```

```
    "idJT integer unsigned not null, "+
```



```

        "PRIMARY KEY (id))"
    };
    static String [] acte = {"drop table acte","create table acte" +
        "(nom_acteur VARCHAR(50) not null, " +
        "annotation_id integer unsigned not null, " +
        "PRIMARY KEY (nom_acteur,annotation_id))"
    };
    static String [] keyword={"drop table keyword","create table keyword"+
        "(motcle VARCHAR(15) not null, " +
        "annotation_id integer unsigned not null, " +
        "PRIMARY KEY (motcle,annotation_ID))"
    };
// Constructeur

    public DBCreate()throws
    SQLException,ClassNotFoundException,IllegalAccessException
    {

        Class.forName("org.gjt.mm.mysql.Driver"); // le pilot JDBC MySQL

        Connection c = DriverManager.getConnection("jdbc:mysql://localhost/dbjt");

        Statement s = c.createStatement();
        // // créer la table identificationJT

        for(int i = 0; i < identificationJT.length; i++)
        {
            System.out.println(identificationJT[i]);

            try {
                s.executeUpdate(identificationJT[i]);
            }
            catch(SQLException sqlEx)
            {
                System.err.println("la requette SQL n'a pas pu être executé
");
            }
        }

        // // créer la table animer
        for(int i = 0; i < animer.length; i++)
        {
            System.out.println(animer[i]);

            try {
                s.executeUpdate(animer[i]);
            }
            catch(SQLException sqlEx)
            {

```

```
        System.err.println("la requette SQL n'a pas pu être executé
");
    }
}

// créer la table invite

for(int i = 0; i < invite.length; i++)
{
    System.out.println(invite[i]);

    try {
        s.executeUpdate(invite[i]);
    }
    catch(SQLException sqlEx)
    {
        System.err.println("la requette SQL n'a pas pu être executé
");
    }
}

// créer la table annotation
for(int i = 0; i < annotation.length; i++)
{
    System.out.println(annotation[i]);

    try {
        s.executeUpdate(annotation[i]);
    }
    catch(SQLException sqlEx)
    {
        System.err.println("la requette SQL n'a pas pu être executé
");
    }
}

// créer la table acte

for(int i = 0; i < acte.length; i++)
{
    System.out.println(acte[i]);

    try {
        s.executeUpdate(acte[i]);
    }
    catch(SQLException sqlEx)
    {
        System.err.println("la requette SQL n'a pas pu être executé
");
    }
}
```

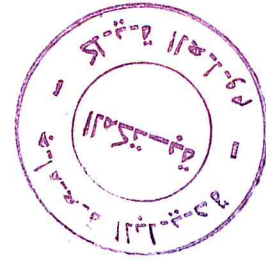
```
// créer la table keyword
for(int i = 0; i < keyword.length; i++)
    {
        System.out.println(keyword[i]);

        try {
            s.executeUpdate(keyword[i]);
        }
        catch(SQLException sqlEx)
        {
            System.err.println("la requette SQL n'a pas pu être executé
");
        }
    }
} // Fin du Programme:~
```



# Annexe E: RMI

(Remote Method Invocation)



## Introduction à RMI (Invocation de méthodes distantes) :

Les approches traditionnelles pour exécuter des instructions à travers un réseau sur d'autres ordinateurs étaient confuses qu'ennuyeuses et sujettes aux erreurs. La meilleure manière d'aborder ce problème est de considérer qu'en fait un objet donné est présent sur une autre machine, on lui envoie un message et l'on obtient le résultat comme si l'objet était instancié sur votre machine locale. Cette simplification est exactement celle que Java 1.1 *Remote Method Invocation* (RMI - Invocation de méthodes distantes) permet de faire.

RMI permet à des objets Java d'invoquer des méthodes sur des objets localisés dans des JVM (Java Virtual Machine) différentes, et même distantes sur le réseau, et ceci de façon quasi transparente.

**RMI** (*Remote Method Invocation*) est une API Java permettant de manipuler des objets distants (c'est-à-dire un objet instancié sur une autre machine virtuelle, éventuellement sur une autre machine du réseau) de manière transparente pour l'utilisateur, c'est-à-dire de la même façon que si l'objet était sur la machine virtuelle (JVM) de la machine locale.

Ainsi un serveur permet à un client d'invoquer des méthodes à distance sur un objet qu'il instancie. Deux machines virtuelles sont donc nécessaires (une sur le serveur et une sur le client) et l'ensemble des communications se fait en Java.

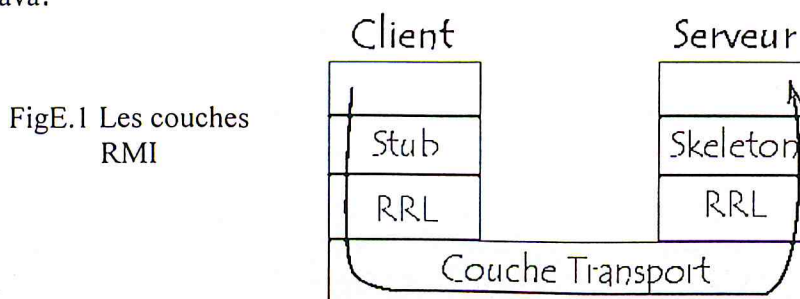
On dit généralement que RMI est une solution "tout Java", contrairement à la norme Corba de l'OMG (Object Management Group) permettant de manipuler des objets à distance avec n'importe quel langage. Corba est toutefois beaucoup plus compliqué à mettre en oeuvre, c'est la raison pour laquelle de nombreux développeurs se tournent généralement vers RMI.

## Structure des couches RMI :

Les connexions et les transferts de données dans RMI sont effectuées par Java sur TCP/IP grâce à un protocole propriétaire (**JRMP**, *Java remote Method Control*) sur le port 1099.

A partir de Java 2 version 1.3, les communications entre client et serveur s'effectuent grâce au protocole RMI-IIOP (*Internet Inter-Orb Protocol*), un protocole normalisé par l'OMG (*Object Management Group*) et utilisé dans l'architecture CORBA.

La transmission de données se fait à travers un système de couches, basées sur le modèle OSI afin de garantir une interopérabilité entre les programmes et les versions de Java.



- Le **stub** (traduisez *souche*) et le *skeleton* (traduisez *squelette*), respectivement sur le client et le serveur, assurent la conversion des communications avec l'objet distant.
- La couche de référence (*RRL, remote Reference Layer*) est chargée du système de localisation afin de fournir un moyen aux objets d'obtenir une référence à l'objet distant. Elle est assurée par le package *java.rmi.Naming*. On l'appelle généralement **registre RMI** car elle référence les objets.
- La couche de transport permet d'écouter les appels entrants ainsi que d'établir les connexions et le transport des données sur le réseau par l'intermédiaire du protocole TCP. Les packages *java.net.Socket* et *java.net.SocketServer* assurent implicitement cette fonction.

Ainsi, une application client-serveur basé sur RMI met ainsi en oeuvre trois composantes :

- **une application cliente** implémentant le **stub**
- **une application serveur** implémentant le **skeleton** (*squelette*)
- **une application médiatrice** (le **registre RMI**) servie par un processus tiers (*rmiregistry*)

#### Architecture de RMI :

L'architecture de RMI est schématisée ci-dessous :

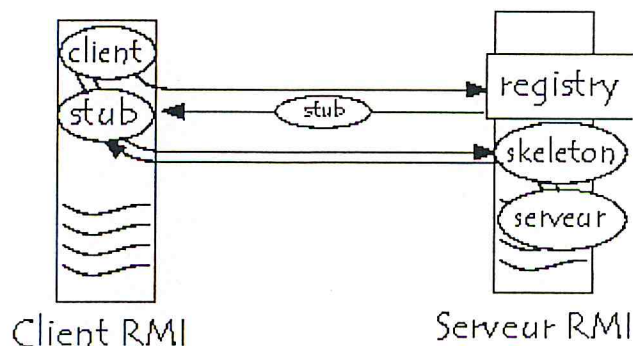


Fig. architecture de RMI

Lorsqu'un objet instancié sur une machine cliente désire accéder à des méthodes d'un objet distant, il effectue les opérations suivantes :

1. il localise l'objet distant grâce à un service de désignation : le **registre RMI**
2. il obtient dynamiquement une image virtuelle de l'objet distant (appelée **stub** ou *souche* en français). Le stub possède exactement la même interface que l'objet distant.
3. Le stub transforme l'appel de la méthode distante en une suite d'octets, c'est ce que l'on appelle la sérialisation, puis les transmet au serveur instanciant l'objet sous forme de flot de données. On dit que le stub "*marshalise*" les arguments de la méthode distante.
4. Le **squelette** instancié sur le serveur "désérialise" les données envoyées par le stub (on dit qu'il les "*démarshalise*"), puis appelle la méthode en local
5. Le squelette récupère les données renvoyées par la méthode (type de base, objet ou exception) puis les marshalise

6. le stub démarshallise les données provenant du squelette et les transmet à l'objet faisant l'appel de méthode à distance

### Mise en oeuvre de RMI :

Pour créer une application avec RMI il suffit de procéder comme suit :

1. définir la classe distante. Celle-ci doit dériver de *java.rmi.server.UnicastRemoteObject* (utilisant elle-même les classes *Socket* et *SocketServer*, permettant la communication par protocole TCP)
2. définir l'interface pour la classe distante. Celle-ci doit implémenter l'interface *java.rmi.Remote* et déclarer les méthodes publiques globales de l'objet, c'est-à-dire les méthodes partageables. De plus ces méthodes doivent pouvoir lancer une exception de type *java.rmi.RemoteException*.
3. créer les classes pour le stub et le squelette grâce à la commande *rmic*
4. Lancer le registre RMI et lancer l'application serveur, c'est-à-dire instancier l'objet distant. Celui-ci lors de l'instanciation créera un lien avec le registre
5. Créer un programme client capable d'accéder à aux méthodes d'un objet sur le serveur grâce à la méthode *Naming.lookup()*
6. Compiler l'application cliente
7. Instancier le client