République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ABOU BEKR BELKAID TLEMCEN

Faculté des Sciences de l'Ingénieur
Département d'Electronique

MEMOIRE

Pour l'obtention de Diplôme de Magister en API
(Automatique-Productique-Informatique)

Option : Informatique

Thème

Construction d'un entrepôt de métadonnées de « LOM »

Application : e-learning

Présenté par
Mme ILES Nawel

Président	: **M. SARI Zaki**	M.C à l'université de Tlemcen
Examinateur	: **M. CHIKH Amine**	M.C à l'université de Tlemcen
Examinateur	: **M. BESSAID Abdelhafid**	M.C à l'université de Tlemcen
Encadreur	: **M. CHIKH Azeddine**	M.C à l'université de Tlemcen

2004 - 2005

Remerciements

Mes remerciements et ma grande reconnaissance :

- A mon encadreur, Monsieur CHIKH Azeddine, maître de conférence à l'université de Tlemcen, qui m'a encadré avec beaucoup d'efficacité et a été la source de précieux conseils et encouragements ;

- Aux membres du jury, M. Zaki SARI, maître de conférence à l'Université de Tlemcen, qui m'a fait l'honneur de présider mon jury, mais également à M. Amine CHIKH et M. Abdelhafid BESSAID, maîtres de conférence à l'Université de Tlemcen, pour avoir accepter d'examiner et juger mon travail ;

- A tous les enseignants qui nous ont formé durant le cursus de post-graduation à l'université de Tlemcen ;

Je tiens également à remercier toute personne qui m'a soutenu et aidé de prés ou de loin à la réalisation de ce travail.

TABLE DE MATIERES

TABLE DES MATIERES

INTRODUCTION **8**

PARTIE I – INGENIERIE DE DOCUMENTS **11**

1. CHAPITRE - Documents numériques **12**

1.1. Introduction 12

1.2. Qu'est ce qu'un document ? 13

1.3. Différents aspects d'un document 13

1.4. Les différentes structures de documents 14

1.4.1. Structure physique 14

1.4.2. Structure logique 16

1.5. Les différents types de documents 17

1.6. Formats normalisées des documents structurés 18

2. CHAPITRE - Le langage XML **18**

2.1. Introduction 18

2.2. Pourquoi XML ? 19

2.3. Les forces de XML 20

2.4. Avantages de XML 20

2.5. La structure de XML 22

2.6. Standards dérivés de XML et outils 27

2.7. Technologie XML 27

2.7.1 Les langages de lien d'adressage 28

2.7.2 Langages de requêtes structurés XML 29

2.8. Intérêt de XML pour la mémoire documentaire 29

2.9. Le rôle de XML dans les systèmes d'intégration de données 29

2.10. XML ET LES SGBD 31

2.11. XML et les moteurs de recherche 31

2.12. Conclusion 31

3. CHAPITRE - Métadonnées 32

3.1. Définition 32

3.2. Les caractéristiques des métadonnées ? 32

3.3. Objectif des métadonnées 33

3.4. Les différents domaines de métadonnées 34

3.5. Les différentes formes des métadonnées ? 34

3.6. Où se trouvent les métadonnées ? 34

3.7. Les normes et standards de métadonnées 35

3.8. Les standards des méta données 36

3.8.1 Le standard Dublin Core 36

3.8.2 Instructional Management System « IMS » 37

3.8.3 Ressource Description Framework « RDF » 39

3.8.4 Sharable Content Object Reference Model « SCORM » 39

3.8.5 Learning Object Metadata « LOM » 40

3.9. Conclusion 51

PARTIE II – ENTREPOT DE DONNEES 53

4. CHAPITRE - Entrepôt de données 54

4.1. Introduction 54

4.2. Définition et objectifs d'un entrepôt de données 54

4.3. Les entrepôts de données et magasins de données 56

4.4. Architecture d'un entrepôt de données 57

4.5. Modélisation multidimensionnelle de l'entrepôt 64

4.6. Manipulation des données multidimensionnelles 66

4.7. Les vues pour la conception d'entrepôts de données 68

4.8. La problématique des entrepôts face aux données complexes 68

4.9. Conclusion 69

5. CHAPITRE - Etat de l'art sur les entrepôts – travaux existants 70

5.1. Projet Xylème 70
5.2. *Projet e_XMLMédia* 72
5.3. Projet Karina 75
5.4. Projet Lore:«Lightweight Object REpository » 76
5.5. Projet Strudel 77
5.6. Wind : « Warehouse for Internet Data » 78
5.7. Projet DOCWARE 79
5.8. Projet ARIADNE 81
5.9. Approche vers l'entreposage des données complexes 82
Bilan .....

PARTIE III – CONSTRUCTION D'UN ENTREPOT DE METADONNEES..... 88

6. CHAPITRE - Le projet de recherche SABRA 89

6.1. Introduction 89
6.2. La méthodologie ARBRE 90
6.2.1. Les éléments utilisés 90
6.2.2. La qualification des briques dans le modèle ASARD 92
6.2.3. Le Modèle général de composant de document « MCD » 93
6.3. Conclusion 94

7. CHAPITRE - Construction d'un entrepôt de métadonnées pédagogiques..... 95

7.1. Introduction 95
7.2. Le pourquoi de l'entrepôt de documents pédagogiques ? 95
7.3. Le pourquoi d'un entrepôt de métadonnées pédagogiques ? 96
7.4. Définition de l'entrepôt de documents pédagogiques 98
7.5. Objectif de l'entrepôt de document 99
7.6. Identification des acteurs et leurs activités 99
7.7. Architecture de l'entrepôt de document 101
7.8. Prototype 115

CONCLUSION. 119

Références bibliographiques..... 121

Liste des Figures

Figure 1.1 : Vue d'un document 15
Figure 1.2 : Différentes structures de documents 16
Figure 2.1: Différence entre un document HTML et un document XML 19
Figure 2.2: Mécanisme XSL 20
Figure 3.1: Organisation du schéma de métadonnée LOM 24
Figure 3.2: Catégories de LOM 42
Figure 3.3: Descripteur « Général » 42
Figure 3.4: Descripteur « Lifecycle » 43
Figure 3.5: Descripteur « Metametadata » 43
Figure 3.6: Descripteur « Technical » 44
Figure 3.7: Descripteur « Educational » 44
Figure 3.8: Descripteur « Rights » 45
Figure 3.9: Descripteur « relation » 46
Figure 3.10: Descripteur « annotation » 46
Figure 3.11: Descripteur « classification » 46
Figure 4.1: Architecture des magasins de données « Data mart » 47
Figure 4.2: Architecture d'un entrepôt de donnée 56
Figure 4.3: Approche virtuelle 58
Figure 4.4: Approche matérialisée 58
Figure 4.5: Exemple de fait 63
Figure 4.6: Exemple de dimension 63
Figure 4.7: Exemple de modèle en étoile 64
Figure 4.8: Modèle en flocon de neige 65
Figure 4.9: Modèle en constellation 65
Figure 4.10 : Exemple de cube de données 66
Figure 4.11: Exemple d'une table multidimensionnelle 67
Figure 5.1: Architecture de Xylème 71
Figure 5.2: Architecture de e-XMLizer 72
Figure 5.3: Architecture de e-XMLRepository 73
Figure 5.4 : Architecture de e-XMLMediator 75
Figure 5.5: Architecture de Lore 76
Figure 5.6: Architecture de Strudel 77
Figure 5.7: Architecture de Wind 78
Figure 5.8 : Architecture du DOCWARE 80
Figure 5.9 : Le processus d'entreposage et d'analyse des données complexes 82
Figure 5.10: Modélisation multidimensionnelle des données complexes 84
Figure 6.1. Catégories d'annotations du modèle ASARD utilisées pour la qualification 92
Figure 6.2 : Modèle général de composant de document (UML) 94
Figure 7.1: Activités des acteurs 100
Figure 7.2 : Architecture de l'entrepôt 102
Figure 7.3: Architecture du module ENCQ 103
Figure 7.4 : Représentation de l'entrepôt en utilisant un SGBD 106
Figure 7.5 : Architecture du module Serveur Web 111
Figure 7.6 : Cycle de vie d'une requête 112
Figure 7.7 : Etape d'accès à l'entrepôt 115
Figure 7.8 : Etape de construction de l'entrepôt 116
Figure 7.9 : Phase de qualification 117
Figure 7.10 : Etape de recherche de documents 118
Figure 7.11 : Résultat de la recherche 118

Listing

Listing 2.1 : document XML « cours » 20
Listing 3.1 : Exemple d'un document annoté avec le modèle LOM 50
Listing 7.1 DTD de l'entrepôt des métadonnées 107
Listing 7.2 : Instance de l'entrepôt de métadonnée 110
Listing 7.3 : La DTD du résultat d'une requête 113
Listing 7.4: un exemple du résultat d'une requête 113
Listing 7.5 : exemple d'une requête XQuery 114

INTRODUCTION

Face à l'évolution rapide des moyens d'information et de communication dans le domaine du E-Learning, une masse importante de documents pédagogiques est produite chaque jour à travers de nombreuses universités. Par conséquent, la nécessité de structurer ces documents, de les capitaliser et de les rendre utilisables et accessibles de la façon la plus optimale possible ne s'est jamais autant fait sentir.

Cette augmentation exponentielle des documents pédagogiques qui peuvent être des cours, exercices, études de cas, résumés produits par les différentes universités, n'a fait qu'accroître les difficultés de leurs exploitation. Ces difficultés sont en grande partie liées aux volumes à manipuler, à leurs coûts élevés, mais également à leurs hétérogénéités. La recherche de ces documents pédagogiques en utilisant les moteurs de recherche comme Google ou Altavista ne pouvait pas être satisfaisante où les résultats d'une requête de l'utilisateur pourrait engendrer des documents non pertinents retrouvés, ou bien des documents pertinents non retrouvés.

Pour dépasser ces limites et améliorer les résultats de recherche sur nos universités, il devient alors nécessaire, voire indispensable de disposer d'outils d'intégration rendant les documents pédagogiques facilement accessibles et exploitable. Notre solution repose sur la création d'un entrepôt de documents pédagogiques à base de métadonnées de « LOM [1] ». Cet entrepôt doit être vu comme un référentiel documentaire qui pourra capitaliser l'ensemble des documents pédagogiques pertinents et sur lequel l'enseignant ou l'étudiant pourra appliquer les mécanismes d'interrogation et de recherche.

L'originalité de notre approche réside dans sa capacité à l'intégration, de tout type de documents pédagogiques complexes, représentées dans des formats différents (textes, images, son, vidéos, bases de données,etc.), issus de sources diverses (données de production, scanners, satellites, enregistrements vidéos, comptes-rendus médicaux, résultats d'analyse, Web, etc.).

Cette approche permet la capitalisation des documents grâce à l'intégration de quatre technologies complémentaires :

1. le langage *XML*[2] comme solution au problème de représentation et d'échange des données sur le Web vu leurs avantages (structures de données complexes et irrégulières, pas de schéma obligatoire,…).,
2. les *métadonnées de « LOM »* comme moyen permettant de repérer plus efficacement des documents pédagogiques en facilitant la recherche par les descripteurs de « LOM »
3. les entrepôts comme moyen pour rassembler les documents pédagogiques.

---

[1] Learning Object Metadata
[2] eXended Markup Language

Structure de la thèse

Outre l'introduction générale, ce mémoire comporte sept chapitres répartis sur Trois parties :

PARTIE I : L'INGENIERIE DES DOCUMENTS

La première partie composée de trois chapitres, est destinée à la description du concept des documents numériques, la technologie XML et les métadonnées.

Le Chapitre 1 : définit la notion de document numérique ainsi que le document pédagogique, en citant leurs différents aspects, structures et types;

Le Chapitre 2 : décrit le langage XML « structure, standards, outils,….. ». On abordera dans ce chapitre l'intérêt du langage XML dans plusieurs domaines : « SGBD, mémoire documentaire, système d'intégration » ;

Le Chapitre 3 : présente de façon détaillée la notion de métadonnées et son influence sur le domaine e-learning ;

Nous présenterons après les standards actuels de métadonnées élaborés par la communauté du e-Learning et nous essayons de concentrer notre étude sur le LOM du IEEE, du moment qu'il constitue le standard actuel adopté ;

PARTIE II : LES ENTREPOTS DE DONNEES

Cette deuxième partie composée de deux chapitres 4 et 5, va présenter le domaine des entrepôts de données ainsi que quelques travaux réalisés.

Le chapitre 4 : va présenter le concept des entrepôts de données ainsi que les technologie d'analyse et de fouille de données ;

Le Chapitre 5 va exposer un état de l'art sur les travaux dans le domaine de l'entreposage des données. Nous évoquons également une approche d'entreposage des données complexes.

PARTIE III : CONSTRUCTION D'UN ENTREPOT DE METADONNEES

Cette dernière partie va présenter le contexte de notre étude à savoir, le projet de recherche SABRA, ainsi que le résultat de notre approche. Elle est composée de deux chapitres :

Le Chapitre 6 : va représenter le projet « SABRA » sur lequel s'appuie notre approche pour développer un système de construction d'un entrepôt de document pédagogique. Nous le considérons comme un cadre théorique et méthodologique pour développer nos propres concepts.

Le Chapitre 7 : Ce chapitre présente l'architecture que nous avons proposée pour la construction d'un entrepôt de métadonnées basée sur le « LOM », illustrée par un prototype où on retrouvera les composants et les processus de notre architecture. Ainsi, nous allons détaillée chaque composant de l'architecture et en identifiant les différents acteurs intervenant dans le système.

PARTIE I - INGENIERIE DE DOCUMENTS

Partie I – Ingénierie de documents

L'ingénierie des documents désigne un ensemble de ressources documentaires et de services. Elle permet de trouver et mettre en oeuvre les meilleures façons de structurer l'information dans des documents, de la décrire, de la valider, de la conserver, de la repérer et de l'exploiter. La possibilité d'y parvenir est aujourd'hui accrue par le protocole XML (*Extensible Markup Language*) qui, en facilitant la création de documents structurés, permet une interfonctionnalité plus profonde que celle que procurent les seules métadonnées en rendant également partageables les données internes du document au moyen de leur balisage et de leur marquage en éléments logiques.

Ainsi, cette partie vise à introduire les domaines clés qui s'intéressent à la conception de notre approche, à savoir «les documents numériques », « le langage XML » et « le domaine des métadonnées ».

1. CHAPITRE - Documents numériques

1.1. Introduction

Depuis quelques années, les besoins en matière de production, gestion et diffusion d'information sous la forme de documents électroniques ne cessent de croître. La nécessité de gérer et traiter des documents de plus en plus complexes n'a cessé de s'accentuer.

Les documents actuels ne sont pas seulement numériques ou symboliques, mais qu'ils peuvent être représentées dans des formats différents (textes, images, son, vidéos, bases de données, etc.) provenir de sources diverses (données de production, scanners, satellites, enregistrements vidéos, comptes-rendus médicaux, résultats d'analyse, Web, etc.), avoir une sémantique différente (langues différentes, échelles différentes, évolution de la définition d'une donnée dans le temps, etc.) [BOUS03].

La nécessité de disposer de méthodes appropriées pour la conception, la production et la manipulation de documents s'avère essentielle pour gérer efficacement des bases documentaires dans des environnements répartis. Ainsi né, le concept d'ingénierie des documents qui répond actuellement à une grande partie des besoins des applications documentaires.

L'ingénierie des documents est un ensemble de ressources et de services qui supportent diverses activités de travail avec un document. Elle intervient pour structurer les documents et améliorer leur exploitation.

1.2. Qu'est ce qu'un document ?

Selon la définition du Larousse, « Un document est tout renseignement écrit ou objet servant de preuve ou d'information ».

Jusqu'à récemment, au terme document était associé la notion de document papier ; le document représente l'information sous forme de texte ou graphe. A ce jour, selon la définition ISO "Un document est l'ensemble d'un support d'information et des données enregistrées sur celui-ci sous forme en général permanente et lisible par l'homme et la machine".

Mais avec l'évolution de la technologie, nous parlons désormais de « document numérique »

- ***Document numérique***

Un document numérique est un ensemble cohérent d'objets numériques (texte, graphique, photo, images animées et sons) stockés sur des machines informatiques interconnectées, ou stockés sur des supports informatiques amovibles.

Pour lire un document numérique, il est nécessaire, soit de l'imprimer sur du papier, soit de le visualiser sur un écran. Le format électronique a conféré aux documents de nouvelles capacités aussi bien au niveau contenu que de sa forme et de ses traitements possibles [DUPO94].

Le concept de structure est très récent. Il est apparu avec le besoin de réutilisation des informations comprises dans les documents et celui d'indépendance des mécanismes de description, d'interprétation et de production.

Pour qu'un ordinateur soit capable de réutiliser des fragments de documents, pour les assembler et les recomposer avec d'autres, afin de créer de nouveaux documents, il est nécessaire de connaître et d'identifier la structure logique des fragments.

C'est une description d'ordre supérieur à celle des contenus. Elle s'intéresse à la granularité de l'information et à son agencement. C'est par une vue conceptuelle des deux structures de documents, physique et logique, que des informations complémentaires à celles des contenus sont apportées.

1.3. Différents aspects d'un document

Un document peut se présenter sous forme de plusieurs aspects [MAIT01]:

- Editorial
  - présentation du document ;
- Signalétique
  - identification du document : ISBN, titre, auteurs, éditeur, année,...
- Structurel
  - organisation logique du document : découpage en chapitres et en paragraphes, figures, annotations,...
- Sémantique
  - sujet traité par le document ;
- Multimédia
  - type des données véhiculées : textes, images, sons, animation,...

1.4. Les différentes structures de documents

Il est apparu avec le besoin de réutilisation des informations comprises dans les documents et celui d'indépendance des mécanismes de description, d'interprétation et de production. Un document peut être vu de différentes façons. On peut considérer sa forme graphique (ou physique), son organisation logique, le style de son écriture, les informations et les connaissances qu'il véhicule, etc. A chacune de ces différentes visions du document, il est possible d'associer une structure de document. Les structures expriment la connaissance qu'on possède sur l'organisation logique, la présentation voire la sémantique véhiculée par le document. Les différentes structures associées à un document ne sont pas indépendantes les unes des autres [CHIK04].

Pour rendre les différents documents homogènes et leur exploitation automatique, il est important de dissocier le contenu du document de sa structure logique ainsi que de structure physique. Cette dissociation présente plusieurs avantages :

- réutiliser des briques « fragments » de documents pour la création de nouveaux documents ;
- affecter à un même document plusieurs présentations sur des supports physiques différents ;
- interroger le contenu des documents en se basant sur leur structure logique.

La structure d'un document peut être implicite ou explicite. Elle est dite explicite quand elle est accessible aux applications informatiques. Elle est dite implicite quand elle n'est accessible qu'à l'être humain (lecteur du document). La structure d'un document peut être rendue explicite à travers des balises qu'on insère dans le document, des liens entre des objets, etc.

Lorsque les structures d'un document sont rendues explicites, il est possible d'envisager des traitements automatiques et intelligents du document. La richesse d'un modèle de représentation des documents réside dans sa capacité à expliciter les structures d'un document [QUIN94].

On peut donc considérer un document comme un moyen matériel qui permet de véhiculer des idées, des connaissances et de retranscrire les faits passés, présents ou futurs. Il est caractérisé généralement par son contenu « structure logique » et son support de présentation « structure physique.

1.4.1. Structure physique

Le concept de structure physique est lié à la restitution du document sur un support physique (papier, écran, etc.). Cette structure physique permet un découpage de l'information en tenant compte de la présentation que nous voulons donner au document, c'est-à-dire l'agencement sur le support des différents blocs d'information.

Schématiquement, un bloc est représenté par une interface rectangulaire de taille et de coordonnées précises destinée à contenir l'information. Un bloc peut lui aussi être découpé à nouveau en sous blocs. Chaque bloc élémentaire contient un granule logique d'information de type homogène (texte, image, audio, vidéo).

La structure physique se traduit sous forme d'une arborescence de blocs. Sur un support papier, le découpage s'effectuera par exemple page par page, colonne par colonne.

1.4.2. Structure logique

Le concept de structure logique permet un découpage de l'information d'un point de vue hiérarchique selon un principe de décomposition plus ou moins fine. Ce mécanisme impose d'identifier de façon non ambiguë les briques « fragments » composant le document et leur faciliter l'accès à ces informations.

Toute structure logique peut être représentée elle aussi par une arborescence dont la racine correspond au document et les feuilles correspondent aux contenus des briques. La structuration logique d'un document relève du choix de l'auteur. C'est lui qui connaît et introduit la granularité de l'information [VANO00].

Les attributs permettent d'ajouter de la sémantique à la structure logique et contribuent à l'enrichissement de la représentation des documents. Ils peuvent être utilisés dans divers contextes d'applications, par exemple, pour déterminer les droits d'accès associés à certains éléments de la structure pour chaque catégorie d'utilisateurs (pour la prise en compte de la confidentialité).

Un autre exemple d'attributs est la langue. Associé à un élément, l'attribut langue permet d'indiquer dans quelle langue est écrit le passage correspondant du document (pour le traitement des documents multilingues) [CHIK04].

Un exemple sur la figure 1.1 présente ces deux structures :

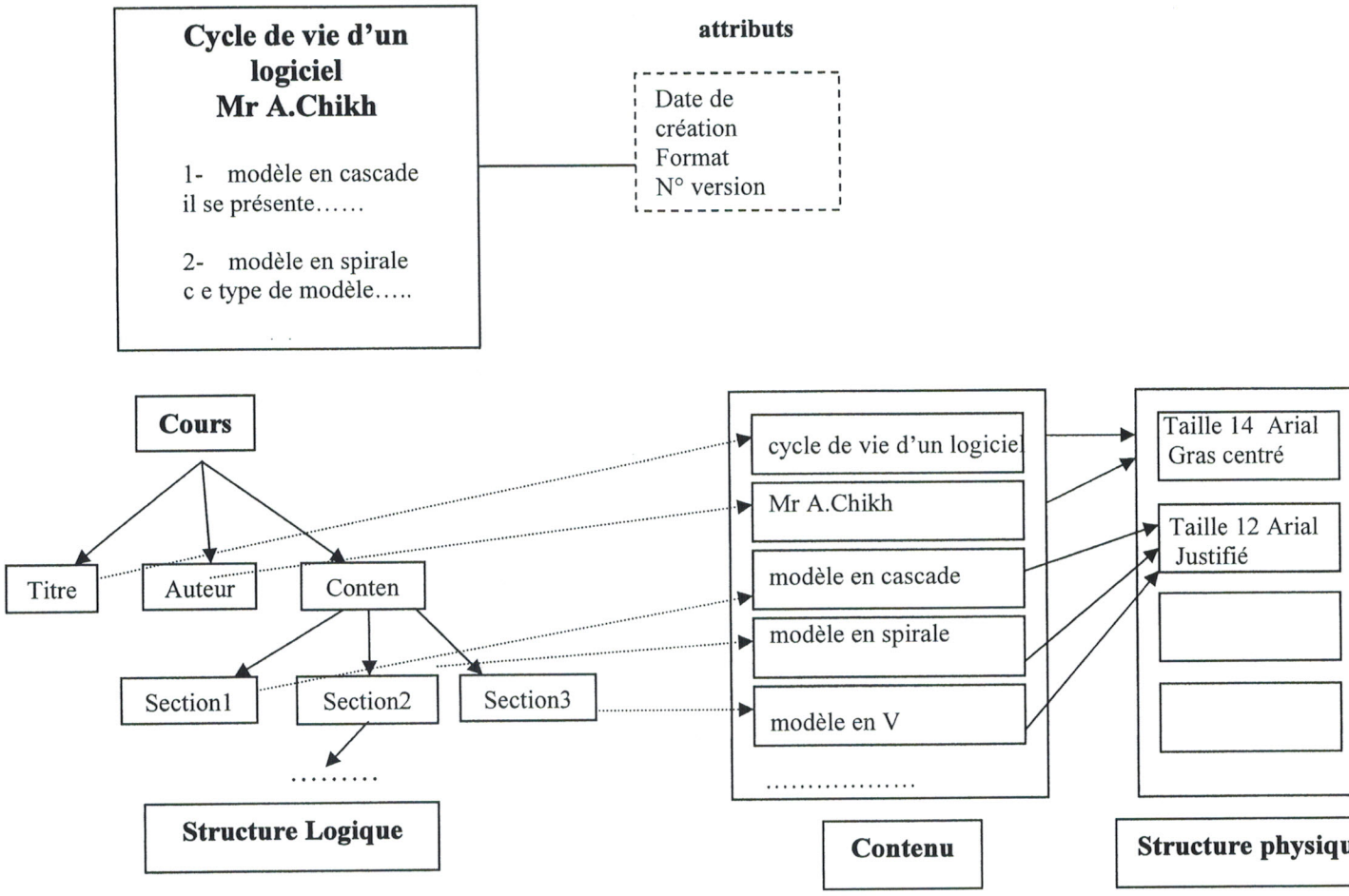


Figure 1.1 Vue d'un document

Le concept de structure logique et physique peut être décomposé en structure générique et structure spécifique. La structure générique exprime l'organisation générique commune à plusieurs documents, c'est une structure qui regroupe toute une classe de documents. La structure spécifique à un document est associée à une instance de la structure générique correspondante. Elle est unique et elle ne concerne qu'un document et un seul. Si la structure générique peut ne pas être définie, la structure spécifique, quant à elle, représente le contenu même du document.

Avec le développement de nouveaux media de communication et d'accès aux informations, il convient d'ajouter aussi le concept de structure hypertexte et de structure spatio-temporelle. Les différentes structures sont articulées entre elles « figure 1.2 ». Ainsi, pour restituer un document, il est indispensable de relier sa structure logique et sa structure physique.

Cette technique nous permet de mettre en correspondance les granules élémentaires logiques et leur situation physique.

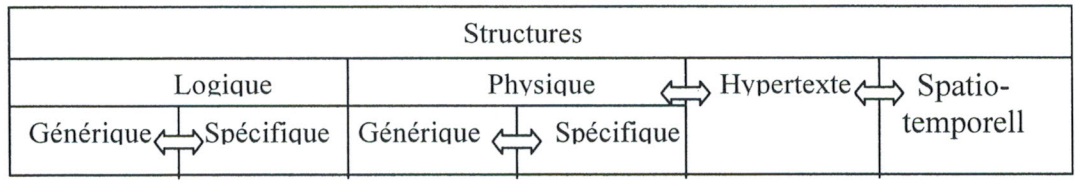


Figure 1.2 différentes structures de documents

Aujourd'hui, un nouveau concept est apparu, « La structure sémantique de documents». Cette structure sémantique est décrite comme « un ensemble de balises sémantiques représentant des concepts associés entre eux par des relations ».

Il s'agit de métadonnées extraites du contenu des documents et reliées entre elles par des relations de type ontologique ou graphe conceptuel. Ce type de structure venant se superposer à tout autre type de structure dans le but d'ajouter une sémantique aux balises ou éléments de document lorsque cela est nécessaire, leur définition est un axe de recherche à part entière que nous allons aborder dans le cadre de ce mémoire.

1.5. Les différents types de documents

Partant de la structure logique, trois classes de documents peuvent être distinguées, selon le niveau (non) structuration de leur contenu : (1) les documents structurés, (2) les documents semi-structurés et 3) les documents non structurés.

- Les documents structurés sont ceux dont la structure logique est explicitement déclarée. Cette structure logique doit être connue à priori avant la création dite du document. Elle doit identifier sans ambiguïté chaque composant utilisé dans le document. Tout document SGML ou XML valide entre dans cette catégorie ;
- Les documents semi-structurés sont ceux qui contiennent des informations structurelles implicitement déclarées ou partiellement définies dans le document. Tout document HTML ou XML bien formé est considéré comme semi-structuré ;
- Les documents non structurés sont ceux qui contiennent peu d'informations sur leur structure logique, comme par exemple un document texte brut.

La manipulation de documents se trouve à la croisée de plusieurs domaines. Dans le domaine de l'enseignement et notamment au service de l'apprentissage en ligne que l'on nomme E_Learning, on est disposé de créer, de réutiliser, d'adapter et d'échanger des documents pédagogiques.

On définit un document pédagogique comme une entité numérique ou non, qui peut être utilisée, réutilisée dans des activités liées à l'apprentissage. Parmi celles-ci, on peut citer l'enseignement traditionnel, l'enseignement à distance, la simulation sur ordinateur, etc. Un document pédagogique parcours un cycle de vie allant de sa création , de son intégration dans des cursus de formation, de son exploitation dans des situations de formation, de sa maintenance et les prises en compte des retours d'information obtenus à par tir de son exploitation [GOIT01].

Les documents pédagogiques peuvent être, par exemple, des transparents, des notes de cours, des pages Web, des logiciels de simulation, des programmes d'enseignement, des objectifs pédagogiques,des exercices, etc.

1.6. Formats normalisées des documents structurés

Les normes ou standard, documentaires sont des moyens mis à la disposition des utilisateurs pour les aider à échanger, archiver et exploiter des documents sur une longue durée quelques soit l'environnement matériel et logiciel dans lequel ils ont été crées.

Les travaux de normalisation et de standardisation effectués au sein de ISO et W3C ont donné naissance à plusieurs standards de documents

Il existait déjà le HTML (*HyperText Markup Language*) conçu pour décrire les pages Web, mais aujourd'hui il s'avère finalement assez rustique. Notamment parce que le but du langage HTML est de mettre des pages Web en forme, et qu'il n'est pas fait pour gérer la structure d'un document. Autrement dit, en HTML une balise présente un élément visuellement plus qu'elle ne le hiérarchise.

Le fait d'échanger des documents structurés est devenu tellement nécessaire que l'on a dû se tourner vers le SGML, un standard exploité pour structurer des documentations complexes.

Le SGML[1] est un métalangage particulièrement puissant, mais en contrepartie très complexe. Malheureusement, sa complexité le rend peu adapté au Web. XML est une version simplifiée de SGML, destinée à rendre plus aisé la définition de documents et le développement d'applications les manipulant. Il omet les parties les moins utilisées de SGML, en contrepartie d'un emploi plus simple.

Actuellement, l'utilisation du standard XML proposé par le W3C[2] connaît une expansion fulgurante. XML est devenu, en peu de temps, le standard pour la modélisation et l'échange de documents structurés. Une étude plus détaillée concernant la technologie XML sera étudiée dans le chapitre suivant.

[1] Standard General Markup Language". normalisé en 1986
[2] Word Wide Web Consortium

2. CHAPITRE - Le langage XML

2.1. Introduction

« XML » e**X**tensible **M**arkup **L**anguage signifie langage de balisage extensible. XML est un langage de description et d'échange de documents structurés. C'est un métalangage dérivé du SGML « Standard Généralized Markup Language » [GOIT01].

XML est le résultat de la coopération d'un grand nombre d'entreprises et de chercheurs partenaires du W3C[1]. La première grande étape a été franchie début février 1998 avec la publication d'une recommandation pour la version 1.0. Leur objectif est de définir un formalisme permettant d'échanger facilement des documents complexes sur le Web, en dépassant les limites imposées par HTML.

Le langage XML[2] permet de décrire la structure logique des documents. A l'aide d'un système de balisage, XML permet de marquer les éléments qui composent la structure et les relations entre ces éléments.

2.2. Pourquoi XML ?

Le Web est confronté à deux problèmes :

1. HTML n'est pas extensible, il ne peut pas répondre aux besoins spécifiques de tous les domaines (mathématiques, chimie, musique, astronomie...) et utilise des balises prédéfinies. Il présente un jeu limité de balises orienté présentation « titre, paragraphe, image, lien hypertexte, etc » ;
2. SGML permet de définir de nouveaux langages de balisage spécifiques et complexe.

XML apporte une réponse à ces problèmes. XML est un métalangage, qui va permettre d'inventer à volonté de nouvelles balises pour isoler toutes les informations élémentaires « titre d'ouvrage, prix article, numéro de sécurité sociale, ...), ou agrégats d'informations élémentaires, que peut contenir une page Web.

---

[1] World Wide Web Consortium

[2] e**X**tensible **M**arkup **L**anguage

2.3. Les forces de XML

- ***Vues multiples des données***- XML a la possibilité d'afficher les documents de différentes façons ;
- ***Séparation du contenu, de la structure et de la présentation*** : l'idée centrale d'XML est qu'il permet d'apporter de la valeur ajoutée si les trois aspects fondamentaux d'un document que sont son contenu, sa présentation et sa structure sont séparés pour celui qui le rédige. L'approche XML consiste à créer un document en se focalisant sur l'information qu'il contient et sur la façon dont elle est structurée, les aspects de présentation sont quant à eux traités séparément ;

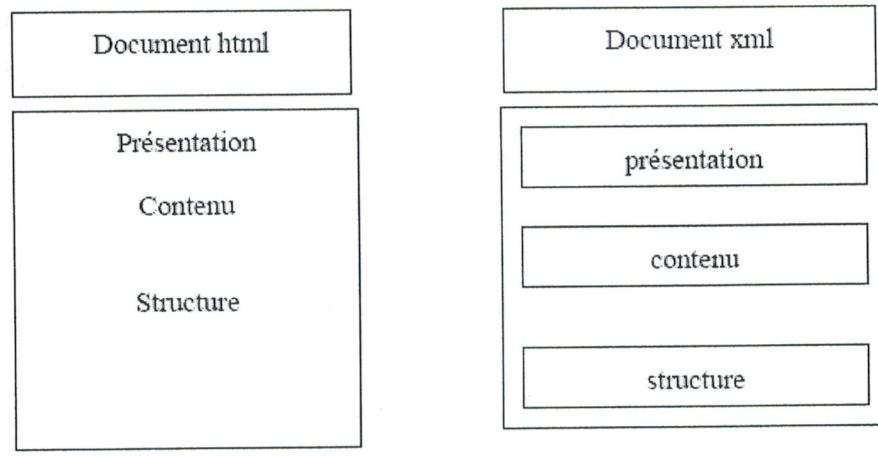


Figure 2.1: différence entre un document HTML et un document XML

- ***Modularité et réutilisation des structures types :*** XML permet de définir librement la structure type d'un document. Il offre la possibilité de définir librement des balises pour marquer les éléments composants un document. Ce qui ouvre des possibilités énormes en termes de traitement automatisé des documents et de réalisation physique [GOIT01] ;
- ***Recherches plus faciles :*** la recherche des documents devient très facile et plus puissantes: on peut par exemple faire une recherche sur un livre écrit par un auteur plutôt que sur un auteur, alors que les méthodes de recherche actuelles mélangent les deux opérations ;
- ***Interopérabilité***- Les documents provenant de plusieurs sources peuvent être intégrées et manipulées par différentes applications.

2.4. Avantages de XML

XML présente les avantages suivants :

- ***Lisibilité :*** les balises utilisent des termes assez intuitifs ;
- ***Indépendance du contenu de sa présentation*** : une balise indique ce que l'information signifie, non pas comment l'afficher. L'information de formatage pour un fichier XML est écrite dans un langage de style et elle est stockée séparément ;
- ***Réduction de la programmation :*** la plupart des vérifications d'erreurs sur la validité des documents sont faites par l'interpréteur. En outre, on peut travailler avec des DTD standard, exemple : XML/EDI ;
- ***Portabilité :*** XML ouvre la programmation Java et Internet aux fonctionnalités portables et indépendantes du navigateur. Java est une excellente plate-forme pour l'utilisation de XML, et XML est une bonne représentation des données pour les applications Java ;
- ***Réutilisabilité*** : notamment celle des DTD et des schémas, mais aussi des fragments de documents par l'utilisation des mécanismes d'entités externes parsées ou non [CHIK04].

2.5. La structure de XML

Le document XML est constitué des instructions de traitement qui permettent de transmettre les informations aux outils chargés de son traitement et d'un ensemble d'éléments délimités par les balises.

Voici un exemple de document XML « cours »

```
< ?xml version ="1.0"  encoding="iso-8859-1" ?>
< ! -- ceci est un commentaire -->
<cours langage ="ar">
    <chapitre>
            <introduction>Introduction 1 </introduction>
            <definition> definition 1 </definition>
            <exemple> Exemple 1 </exemple>
    </chapitre>
    <chapitre>
            <introduction>Introduction 2 </introduction>
            <definition> definition 2 </definition>
            <exemple> Exemple 2 </exemple>
    </chapitre>
</cours>
```

Listing 2.1 : document XML « cours »

Un document XML est composé des éléments suivants :

1. ***Un arbre d'éléments*** qui forme le contenu proprement dit du document ;
2. ***Un prologue*** : éventuellement vide. C'est un ensemble de déclarations dont la présence est facultative mais conseillée ;
3. ***Des commentaires et des instructions de traitement*** dont la présence est facultative et qui peuvent être soit dans le prologue, soit dans l'arbre d'éléments.

Deus types de structures génériques sont utilisés avec XML : la définition type de documents (DTD) et le schéma XML (Xschema).

a- Définition de Types de Documents (DTD)

Une DTD est une Définition de Type de Document, c'est une grammaire permettant de vérifier la conformité du document XML. Elle constitue un mécanisme qui permet de décrire la structure d'une classe de documents XML. La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML. Une DTD peut être une DTD interne, c-à-d, placée dans le DOCTYPE après la déclaration XML et l'autre externe placée dans un fichier séparé. L'appel à ce fichier qui doit avoir le même nom que la racine du document (élément qui englobe tous les autres) est effectué dans le `DOCTYPE` après la déclaration XML. L'avantage des DTD en XML est qu'elles sont flexibles. L'utilisateur peut ajouter de nouveaux éléments et on peut faire référence à plusieurs DTD dans le même document.

b- Le schéma XML « Xschema »

Le XML-Schema est une recommandation du W3C. Les documents XML-Schema permettent de décrire la structure d'un document XML d'une façon plus complète que les DTD. Il est par exemple possible de spécifier la typologie des données (String, decimal, etc..) que va contenir le document XML décrit par le XML-Schema. Cependant, sous le terme XML-Schema se cache plusieurs "normes". Dans la mesure du possible préférer celle du W3C.

** Document valide et document bien formé :*

La norme XML a été conçue pour être utilisée de deux manières distinctes [BILA03] :

- Un document ***bien-formé*** est un document XML dont le balisage est correct, mais qui ne respecte pas nécessairement les règles spécifiques d'une DTD particulière. Dans ce cas, le document doit respecter la syntaxe de la norme XML. Ainsi, il ne peut comporter aucune ambiguïté dans le balisage : tous les éléments doivent posséder une balise ouvrante et fermante, les attributs doivent être entre guillemets ;
- Un document XML ***valide*** est un document conforme à une DTD particulière. Un document valide comporte nécessairement un préambule qui identifie la DTD à laquelle se conforme le document. Les applications de traitement du document peuvent donc récupérer la DTD et l'utiliser pour la validation ou les autres opérations à effectuer sur le document.

2.6. Standards dérivés de XML et outils

a- Standards dérivés

L'intérêt de disposer d'un format commun d'échange d'information dépend du contexte professionnel dans lequel les utilisateurs interviennent. C'est pourquoi, de nombreux formats de données issus de XML apparaissent (il en existe plus d'une centaine) : [ABIT04] [BENJ03]

- **MathML** permet de diffuser aisément sur le Web des documents techniques complexes sans devoir représenter les expressions mathématiques sous forme d'images statiques comme avec HTML.·
- **OFX : Open Financial eXchange** pour les échanges d'informations dans le monde financier
- **PGML, Precision Graphics Markup Language**, décrit les structures de données graphiques complexes avec les primitives du langage Postscript. Il permet la conversion de documents aux formats ps et pdf en XML.
- **RDF(Ressource Description Framework) :** permet de définir des relations arbitrairement complexes entre des documents ou des données,c'est-à- dire de décrire formellement le graphe d'un hypertexte, et de typer les relations entre ces documents ou données(pour l'indexation de documents et la recherche documentaire sur le Web).
- **SMIL(Synchronized Multimedia Integration Language)** est un langage basé sur XML, permettant de représenter et d'échanger des présentations multimédias dynamiques, intégrant sons, images et textes présentés de façon synchrone.·

- **SOAP : Simple Object Access Protocol :** Protocole d'échange de données réseau. Avec ce protocole, il est possible de faire interopérer des applications hétérogènes et distantes à travers le réseau internet. SOAP utilise le protocole http. Son inconvénient est qu'il a des temps de réponse inféieure à celles des autres protocoles de communications.
- **VML, Vector Markup Language,** langage de balisage d'information graphique vectorielle.
- **WML, Wireless Markup Language**, pour l'internet mobile ;
- **XHTML(eXtensible Hypertext Markup Language)** permet de publier sur le Web des documents hypertext qui combinent les balises HTML avec d'autres ensembles de balises

b- Outils

Pour produire un document XML, un utilisateur dispose des outils logiciels suivants :

1- Les outils généraux

La catégorie la plus importante de ces outils est celle des parseurs. Un parseur réalise l'analyse lexicale d'un document XML (reconnaître les unités lexicales qu'il contient), son analyse syntaxique (reconnaître des constructions grammaticalement correctes ou incorrectes du langage), le remplacement des références à des entités internes ou externes par leur valeur

2- Les outils de présentation

Pour utiliser des documents XML sur Internet, tout peut être réalisé en utilisant des éditeurs de feuilles de styles comme CSS, XSLT, XSL etc. On peut aussi trouver des navigateurs XML natifs, mais ils sont encore assez inaccessibles [JOBE03].

Ainsi, l'affichage des documents XML de façon correcte se fait en utilisant du langage des feuilles de style, le CSS (Cascading Style Sheet) ou spécifiquement pour lui le XSL (eXtended Stylesheet Language) que nous allons décrire ci-dessous.

2.1. La visualisation avec une feuille de style CSS « Cascading Style Sheet »

Les feuilles CSS ont été conçues pour le HTML en définissant les propriétés du format telles que la taille des polices, leur type, la couleur .....

2.2. La visualisation avec une feuille de style XSL « eXtensible Style sheet Language »

Le XSL est un langage de description de style plus évolué que le CSS, un document XSL lui-même est un document XML bien formulé, les feuilles XSL peuvent réorganiser les éléments, cacher certains et en montrer d'autres, il permet de choisir le style en fonction du nom des balises, du contenue et de ces attributs.

La présentation d'un document XML grâce à XSL se fait en deux étapes: une étape de transformation réalisée par le mécanisme de transformation de XSL appelé XSLT, et une étape de lecture réalisée grâce au mécanisme de formatage de XSL appelé XSL-FO. La figure suivante illustre ces deux étapes.

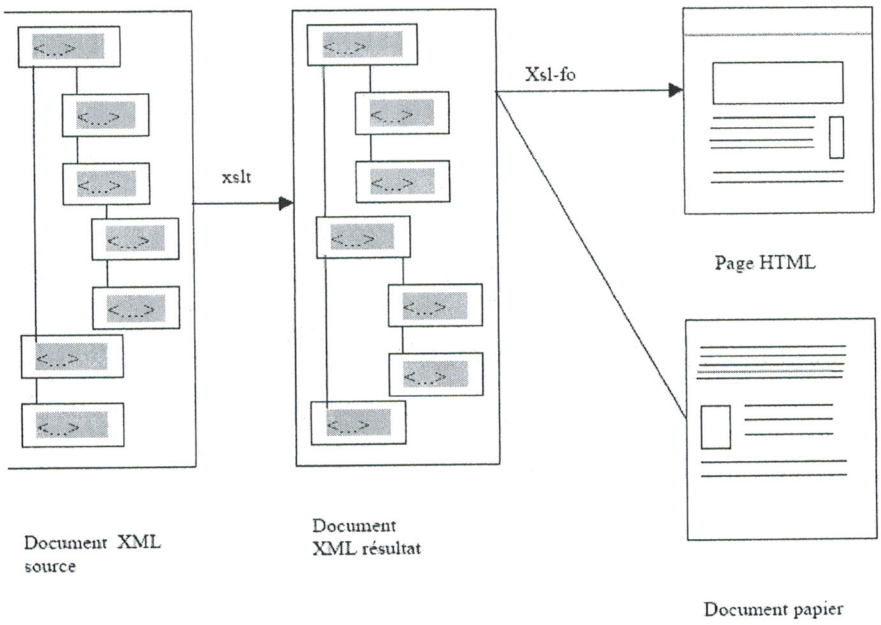


Figure 2.2 : Mécanisme XSL

a- XSLT(eXtensible Stylesheet Language Transformation)

Un document XSL (eXtensible Stylesheet Language) n'est pas toujours structuré d'une façon lisible par les feuilles CSS. Il s'avère, ainsi important de faire recourir à un moyen permettant de transformer ce document en un format de type texte, HTML, XML ou autre, plus adapté à toute application. D'où l'utilisation des feuilles de styles XSLT[3] ;

Le langage XSLT, basé sur les feuilles de style XSL, permet de transformer la structure de documents, par exemple pour correspondre au modèle interne d'une base de données native XML et *vice versa*. Les temps de traitements peuvent cependant être importants si les transformations à effectuer sont nombreuses.

b- XSL-FO (eXtensible Stylesheet Language- Formatting Object)

Un document XML ne contient rien qui spécifie la façon dont il doit être rendu à l'écran ou sur tout autre support (papier, braille…), pas plus que sa DTD. Pour cela, XML utilise un langage de feuilles de styles appelé XSL [4]. Plus précisément, un document XML fait appel à un ou plusieurs documents séparés contenant les styles qui peuvent être appliqués à ses propres éléments : Ces documents sont appelés feuilles de styles et sont rédigés grâce au langage XSL.

---

[3] **eXtensible Stylesheet Language Transformation**

[4] eXtensible Stylesheet Language

XSL -FO permet donc une mise en forme de l'arbre XML résultat. Cette mise en forme est réalisée à l'aide d'un vocabulaire constitué d'objets XML de mise en forme appartenant à un domaine de noms défini. Chaque type d'objet de mise en forme XSL représente un traitement d'affichage dont les paramètres sont définis grâce aux attributs de ces types d'objets.

La visualisation d'un document XML, avec une feuille de style CSS ou XSL, se fait en insérant, après le prologue de ce document, une instruction de traitement précisant la nature de la feuille de style et son chemin.

3- Outil de manipulation des documents XML

- **SAX**

A côté de la manipulation complète de documents, Simple API for XML (SAX) est une spécification très précieuse pour l'écriture des applications. En effet, elle permet de traiter des sous ensembles sans avoir à connaître tout le document. SAX peut permettre à une application de manipuler des vues globales de l'arbre XML d'un document, ou encore de traiter le document en streaming. Cela signifie que le parser ayant une interface SAX et les applications associées vont pouvoir travailler au fur et à mesure, par exemple, de l'arrivée d'un message XML, construisant l'arbre ou traitant des informations dès qu'un élément est correctement constitué [JOBE03] ;

SAX présente un ensemble d'avantages qui, dans certains cas, est la meilleure solution pour écrire son application, dés qu'il s'agit de document de grande taille, de ressources limitées disponibles dans le serveur d'analyse, du besoin d'interrompre l'opération d'analyse du document XML ou même de faire un traitement séquentiel sur le document, SAX est la solution optimale, tandis que le DOM est une interface de programmation orientée document.

- **DOM : « Document Object Model »**

Le DOM est une interface de programmation indépendante du langage et des plates-formes, elle permet d'accéder à n'importe quel document XML, quelque soit sa structure et sa taille, permettant ainsi d'accéder à son contenu ;

Les parseurs DOM niveau 1 et 2 parcourent et reconstruisent sous forme d'un arbre d'objets le document XML. Le modèle DOM permet donc facilement de naviguer et de manipuler la structure objets du document.

- **JDOM**

JDOM permet de travailler avec SAX et DOM. Si nous travaillons en Java, JDOM semble être la solution optimale, car cet outil est optimisé pour Java. De plus, il est impossible de travailler sur une structure de données plus simple que celle imposée par DOM.

4- Format d'interchange de méta données : XMI

L'objet principal de XMI, « XML Metadata Interchange », est de permettre l'échange de métadonnées entre outils de modélisation basés sur UML, Unified Modeling Language, et la communication des répertoires de méta données basés sur le MOF, Meta Object Facility, deux standards de l'OMG, Object Management Group.

XMI se fonde sur les trois standards XML, UML et MOF.

- UML est un formalisme orienté objet de conception et de documentation d'applications ;
- MOF est une technologie de définition et de représentation de métadonnée en tant qu'objets CORBA (Common Object Request Broker Architecture). CORBA est un concept permettant à deux applications de communiquer entre elles sans se soucier de leur localisation ou de leur mode de conception.

Les spécifications XMI consistent à proposer un ensemble de règles de transformation de structures MOF en DTD XML, des DTD pour UML et MOF, un ensemble de règles de production de documents XML pour manipuler des métadonnées MOF.

2.7. ***Technologie XML*** [AMAN01] [ABIT04] [BENJ03]

Autour de la spécification XML, il existe une famille de technologies dont : Xlink, XPointer, XPath, …..

2.7.1 Les langages de lien d'adressage : XPATH, XLINK et XPOINTER

a- XPath

Le XML Path Language est un langage d'adressage destiné à la recherche de nœuds dans un document XML, il utilise une syntaxe semblable à celle de l'adresse d'un système de fichiers ou une URL du Web.

Xpath offre la possibilité de sélectionner des nœuds dans un document XML en fonction de critère simple comme la structure, la position et le contenue. Chaque document XML est perçu sous la forme d'un arbre, ses nœuds peuvent être : racine, éléments, attributs, texte, espaces de noms, instruction de traitement ou commentaires.

b- Xlink : « eXtensible Link Language »

Xlink est le langage de description de liens hypertextes en XML. Il étend les hyperliens définies par HTML, en permettant, entre autres des liens qui pourront être bidirectionnels ou gérés dans un fichier extérieur à l'instance de document, des liens cers des cibles sur Internet non balisées au préalable. Des attributs ajoutés aux liens permettent de définir le type de lien (lien vers une définition,, lien extérieur, etc.

c- XPointer

XLink permet de modéliser des liens plus complexes que ceux qu'il est possible de faire avec HTML. Pour compléter cette spécification, il était nécessaire d'avoir un moyen de pointer sur des éléments spécifiques d'un document XML, d'où l'utilisation de XPointer ;

Xpointer fournit un mécanisme permettant de spécifier un point précis dans un document XML en fonction d'une localisation source pour y sélectionner des éléments. Il opère sur la structure arborescente et les noeuds des documents. Il peut être utilisé par Xlink pour établir des liens au sein même d'un document XML. Les fonctionnalités de Xpointer permettent aux liens Xlink de pointer vers un point précis d'un document qui peut être un élément, un texte ou une partie d'un document.

2.7.2 Langages de requêtes structurés XML

Les langages de requêtes permettent d'extraire des informations de documents XML, à la manière de SQL pour les bases de données. Il y a eu de nombreuses initiatives pour définir un langage qui répond à tous les besoins de recherche dans un document XML : XQL, XML-QL, XQuery

a- XQL : « A Query Language for XML Data »

XQL est une variante qui a été proposée par Oracle et Microsoft pour l'interrogation des documents XML. Il s'agit d'une extension de XSL. XQL propose d'étendre la syntaxe de URL « User Requirement Languages » pour interroger des collections de documents XML avec des expressions Xpath. Il fournit une notation précise et compréhensible pour pointer les éléments spécifiques et rechercher des nœuds ayant des caractéristiques spécifiques. Il présente l'avantage d'étendre les notations URL mais reste loin des possibilités des autres langages tels que : Xquery et XML-QL

b- Lorel : « Lightweight Object Repository Language

Lorel est doté d'une syntaxe qui ressemble à OQL, tout en étant influencé par XSQL « eXtended Structured Query Language » (le premier langage à introduire des requêtes au niveau schéma). Il a été conçu pour interroger des graphes OEM à partir du langage OQL. Son avantage réside dans la flexibilité d'interrogation des données semi-structurées même si nous ne connaissons pas leur structure.

c- XML-QL : "XML Query Language"

XML-QL supporte l'interrogation des données et des métadonnées, les expressions de chemins et les expressions régulières. De plus, il permet la reconstruction des graphes via une clause de définition de résultats très complexe. Le résultat d'une requête est un graphe. Son avantage majeur est la possibilité d'exprimer des transformations de données XML d'un schéma à un autre.

d- XQuery "A Query Language for XML"

XQuery est un langage souple de balises, capable de marquer le conteneur d'informations en provenance de diverses sources de données, dont les documents structurés ou semi-structurés, les bases de données relationnelles et les archives d'objets. XQuery ou XML-Query a été conçu pour permettre des requêtes précises et facilement compréhensibles, tout en étant suffisamment souple pour permettre d'accéder à un grand nombre de types de sources d'informations XML, dont les bases de données et les documents.
XQuery s'impose comme le langage de requêtes:

1) Pour les bases de données XML natives ;
2) Pour les documents XML textuels (XQuery Text) ;
3) Pour l'intégration de données (bases de données virtuelles) ;
4) Le besoin d'interroger les bases relationnelles en XQuery existe ;
5) Pour l'intégration et la publication de données ;
6) Compétition avec les extensions de SQL (SQL/XML).

2.8. Intérêt de XML pour la mémoire documentaire [CHIK04]

L'intérêt de XML pour la gestion des connaissances de l'entreprise réside dans l'intégration des documents et des données structurées sur l'intranet, accédées via les serveurs Web. XML est un format d'échange standard de documents et de données structurés hétérogènes. De par son statut de standard international pour les documents structurés, XML permet d'assurer la pérennité des documents et des informations à long terme, dans le cadre de la mémoire d'entreprise. Les informations existent indépendamment des outils qui les manipulent. [CHIK04]

La possibilité de poser des liens hypertextes depuis l'extérieur d'un document, sans toucher au texte source est fondamentale. Elle permet de mettre en œuvre des annotations multipoints de vue sur un même document. Les outils de recherche syntaxique (XML-QL) et sémantique (RDF) sont également d'un grand intérêt pour la recherche d'information dans la mémoire d'entreprise.

XML permet aussi de construire dynamiquement des documents avec des données en provenance de différentes sources : outils de CAO, base de données, rapport, fiche technique ou d'incident, etc. Il représente ainsi la technologie pivot pour diffuser l'information sur l'intranet de l'entreprise. Enfin, XML va révolutionner la recherche d'information sur le Web (et l'intranet) car les serveurs d'information deviennent des bases de données structurées et non pas seulement des serveurs de pages des textes HTML.

2.9. Le rôle de XML dans les systèmes d'intégration de données

La plupart des systèmes d'intégration de données actuels utilisent XML comme modèle commun. Ainsi, en associant un modèle de données à XML, on peut interroger les données XML sur leur structure et sur leur contenu, ce qui représente une avancée considérable pour la recherche d'informations sur le Web. De plus, les langages de requête pour XML sont un élément important des systèmes d'intégration de données du Web pour deux raisons : [AMAN03]

1. L'utilisateur exprime ses requêtes sur un schéma XML et ignore totalement le format des données ainsi que la localisation de leurs sources ;
2. L'existence de langages de requête sert de support pour la définition de vues sur les sources de données.

2.10. XML ET LES SGBD

Les données semi-structurées telles que les documents XML sont caractérisées par l'absence de schéma fixe, bien que les données possèdent implicitement une certaine structure. La difficulté est alors de pouvoir extraire cette structure. Notons que les données semi-structurées sont souvent stockées dans des systèmes de fichiers pourvus de moyens limités pour l'organisation, la recherche et l'exploitation des données. Par ailleurs, les SGBD classiques sont inappropriés aux informations semi-structurées. En effet, ces dernières ne peuvent être gérées par les SGBD que si elles sont traduites de façon conforme au modèle sous-jacent du système utilisé.

D'autre part, le langage XML offre plusieurs fonctionnalités des SGBD telles que le stockage, les schémas, les langages de requêtes, les interfaces de programmation, etc. Cependant, il manque d'outils importants qui existent dans les SGBD, comme des techniques de stockage efficaces, les index, la sécurité, les transactions, l'intégrité des données, les accès multi-utilisateurs, les déclencheurs, les requêtes basées sur plusieurs documents, etc.

Pour le stockage de documents XML, bien qu'il soit possible de les stocker sous leur forme sérialisée dans des fichiers textes standard, il est évident que cette solution est inefficace quand il s'agit de manipuler des grands volumes d'informations. Le stockage efficace de données XML a fait l'objet d'un grand nombre d'activités de recherches et une multitude de solutions ont été proposées. [AMAN03]. D'une manière générale, on peut distinguer entre deux approches.

- La première approche consiste à exploiter la puissance du modèle relationnel ou relationnel-objet en définissant un ensemble de tables permettant de modéliser la structure de chaque document XML. C'est la technique du « mapping ». parmi les premiers systèmes, il existe Oracle 9i, et DB2 UDB d'IBM [JAOU02].
- La deuxième approche consiste à développer des serveurs natifs XML qui répondent aux besoins de développement d'applications. Les serveurs de données nativement XML, Tamino de XML offrent les possibilités suivantes :

  1- Stockage des données complexes à partir es DTD et Schéma ;

  2- Communication avec d'autres SGBDs via ODBC.

  Cette approche peut également exploiter XML comme standard d'échange de données, mais elle permet surtout de gérer plus efficacement des documents en tant que tels (manuels d'utilisation, pages web statiques, etc.).

Tous ces systèmes ont leurs avantages et inconvénients, mais ils montrent surtout que la technologie des bases de données a bien su s'adapter à ce nouveau type de données.

Le domaine de l'interrogation des documents semi-structurées représente une partie importante de l'effort de recherche en base de données pour le Web. Ainsi, il existe quatre moyens principaux d'effectuer des requêtes sur des documents XML. Ils varient notamment selon le type de stockage employé [BOUS03].

1) Le langage XSLT, basé sur les feuilles de style XSL, permet de transformer la structure de documents, par exemple pour correspondre au modèle interne d'une base de données native XML et *vice versa*. Les temps de traitements peuvent cependant être importants si les transformations à effectuer sont nombreuses ;
2) Les langages les plus courants qui fournissent un résultat sous forme de documents XML sont basés sur des modèles (*templates*). Des requêtes de type SELECT sont encapsulées dans un modèle, qui est ensuite interprété. Le résultat obtenu est un document XML de même structure que le modèle. Ces langages sont très flexibles, mais ne sont quasiment utilisés que pour exporter des données relationnelles dans des documents XML ;

3) Les langages de requêtes basés sur SQL utilisent des instructions SQL modifiées dont les résultats sont ensuite transformés en documents XML. Les extensions XML de SQL sont en cours de standardisation par l'ISO et l'ANSI sous le nom de SQL/XML ;

4) Contrairement aux langages de requêtes basés sur des modèles ou SQL, qui ne sont utilisés que dans un contexte relationnel, les langages de requêtes XML peuvent s'appliquer à tout document XML, y compris ceux qui sont stockés dans une base de données relationnelle, mais suivant un modèle XML. XQuery est par exemple capable d'opérer un *mapping* vers une base de données relationnelle ou relationnelle-objet. XPath est très proche de XQuery, mais s'avère plus limité dans un contexte relationnel car il ne supporte pas les requêtes sur plusieurs tables. En revanche, cette limitation est levée dans un contexte relationnel-objet.

2.11. XML et les moteurs de recherche

Le Web représente actuellement un grand entrepôt contenant des documents hétérogènes où la plupart sont non structurés, ce qui nous pose beaucoup de difficultés lors de la recherche d'information à partir des moteurs de recherche.

Afin de faciliter la recherche et obtenir des résultats pertinents, on doit se baser sur une formalisation de documents de telle manière à prendre en considération la sémantique des informations. Dans cette vision, XML offre un bon support syntaxique et sémantique en se basant sur les métadonnées et les ontologies. Pour traiter la requête d'un utilisateur, XML offre l'avantage par rapport aux moteurs de recherche classique de présenter un document sous forme d'un arbre structuré, ce qui permettra aux parseurs XML d'accéder à chaque élément de l'arbre, et donc fournir une réponse plus pertinente à la requête de l'utilisateur.

2.12. Conclusion

Ainsi, Les métadonnées et XML sont en train de métamorphoser la manière de produire, de gérer et d'utiliser les documents pédagogiques. De plus en plus d'applications sont développées, et des standards se dégage maintenant dans le domaine des métadonnées pédagogiques afin de réponde à tous les besoins des enseignants/étudiant.

Dans le chapitre suivant, nous allons étudier plus en détails les métadonnées, ainsi que les différents standards associés.

3. CHAPITRE – Les Métadonnées

Le volume d'informations constituées sous la forme de données numériques connaît une croissance exponentielle. L'hétérogénéité de ces représentations limite les échanges de données, tout en rendant très difficile une manipulation automatisée. La solution passe par l'utilisation de métadonnées.

3.1. Définition

Les métadonnées peuvent être définies comme étant des données relatives à d'autres données (data about data : données sur des données). Par conséquent, une notice catalographique classique peut-être considérée comme une métadonnée [AMER02].

Le terme métadonnée est utilisé pour désigner l'information concernant des fichiers de données " lisibles par machine" : donc ce terme désigne en quelques sortes une information référentielle sur des données électroniques. Ainsi, si les contenus de livres sont les données, des répertoires de bibliothèque ou des index constituent les métadonnées parce qu'ils contiennent des informations sur ces livres et leurs contenus.

Ces métadonnées peuvent être incluses dans les ressources elles-mêmes ou enregistrées dans un fichier séparé selon le type du contenu. Elles peuvent décrire des ensembles plus petits qu'un document, par exemple, des images, ou des fichiers sonores, à l'intérieur d'un document.

Une métadonnée peut être utilisée à des fins diverses:

- la description et la recherche de ressources
- la gestion de collections de ressources
- la préservation des ressources

Dans le domaine du E-Learning, les métadonnées servent à décrire les documents pédagogiques afin de les rendre plus facilement *identifiables* (accessibles) et plus *manipulables* (interopérables, réutilisables, durables, adaptables). Ces métadonnées sont structurées suivant des catégories ou champs sémantiques « descripteurs ». Chaque champ représente une caractéristique particulière sur la ressource, exemple, son titre ou son résumé [BOURD01].

3.2. Les caractéristiques des métadonnées ?

Parmi les principales caractéristiques des métadonnées [DHER05]:

• Elles sont constituées de contenu structuré ou non qui peut être des mots, des formules, des analyses spectrales... ;

• Elles sont créées de façon automatique ou manuelle ;

• Elles peuvent avoir plusieurs niveaux de complexité ;

• Elles sont faites pour être lues soit par l'humain, soit par la machine.

3.3. Objectif des métadonnées [ALIB03]

Les métadonnées ont pour objectifs de:

1. Permettre une description plus ou moins détaillée des ressources.
2. Faciliter le repérage de l'information : elles permettent une facilité de recherche dans la masse informationnelle du web,
3. Permettre une évaluation rapide de la pertinence du contenu d'un document
4. Multiplier les points d'accès à l'information
5. Permettre la gestion des droits d'accès, d'utilisation, de copie (micro paiement)
6. Rendre l'information plus malléable et personnalisable (différents modes de présentation des résultats)
7. Faciliter l'organisation et la gestion de collections de ressources
8. Faciliter la gestion des différentes versions de document : copie de préservation, copie de diffusion (les métadonnées doivent conserver ces liens et indiquer la différence entre les versions)
9. Donner des informations sur les modes d'entretien à long terme de l'information
10. Certifier une certaine autorité intellectuelle du contenu (métadonnées sur l'auteur, la date de création, l'organisme responsable, la date de mise à jour…)
11. Contribuer à la préservation de l'intégrité des documents électroniques :
    - *L'information contenue* : balises auteur, date de création, mots-clés ;
    - *La fixité* : Accompagnement de toutes modifications par numérotation ou identification rigoureuse des versions, identification de la version originale ;
    - *La référence* : Information sur les modes d'accès à la ressource : Format MARC (champs 856) Métadonnées (TEI, HTML), URI (Uniform Resources Identification) ;
    - *La provenance* : Resituer le document au sein d'une collection ou d'une série de documents (dossier/sous-dossier, chapite/no.page) ;
    - *Le contexte* : logiciel utilisé, format, information sur les modes de diffusion, contexte hypertextuel….

Dans le domaine du e-learning, les métadonnées nous permettent de repérer plus efficacement les diverses ressources éducatives sur l'internet en facilitant la recherche par descripteur ou marqueur. Mais l'objectif principal des métadonnées qui accompagnent les diverses ressources du Web c'est de permettre à divers logiciels ou systèmes dédiés au e_learning de pouvoir « interpréter » la fiche descriptive d'une ressource. [BOUR00]

3.4. Les différents domaines de métadonnées

Les métadonnées sont utilisées dans les systèmes de gestion de contenu (CMS) pour éditer, gérer, rechercher, réutiliser, diffuser (imprimé ou en ligne sur le Web), publier de multiples contenus (textes, images, vidéos, etc.)

Les métadonnées d'ordre technique et administratif (comme l'appartenance à une collection, les informations de copyright, les informations sur l'acquisition, le format de fichier, la résolution, etc.) permettent de gérer, maintenir et préserver des collections digitales.

Dans le domaine de l'éducation, les informations relatives aux documents pédagogiques peuvent être structurées suivant des catégories ou champs sémantiques. Chaque champ représente une caractéristique particulière de la ressource, par exemple, son titre ou son résumé.

3.5. Les différentes formes des métadonnées ?

Les métadonnées peuvent être :

1. ***EXTERNES***
   - Les éléments peuvent être contenus dans une notice séparée du document, comme c'est le cas pour une notice dans un catalogue de bibliothèque.
   - Les métadonnées sont stockées dans une base de données spécifique avec un lien pointant vers la ressource (le lien devrait idéalement être bidirectionnel). Si on bouge la ressource ou on utilise la ressource en dehors de son cadre de référence, on perd les métadonnées. Dans le cas de conservation à long terme, on doit s'assurer de conserver les deux parties.

2. ***INTERNES***
   - Les métadonnées peuvent être intégrées dans la ressource elle-même (implicites ou explicites)
   - Implicites : le logiciel génère automatiquement des informations sur le fichier
   - Explicites : sous forme de balisage des données : on inclut un ou plusieurs jeux de métadonnées dans la ressource.
   - Le document, s'il est déplacé, transporte automatiquement ses métadonnées avec lui. Cependant, il peut y avoir une différence de poids entre un fichier incluant ses métadonnées et un fichier leur faisant référence de manière externe.

3.6. Où se trouvent les métadonnées ?

Elles peuvent être partout. Quelques exemples : [DHER05]

- Externes aux ressources et sous forme papier : Notice papier pour remplir un formulaire dont les données sont saisies par informatique ;
- Externes aux ressources et stockées dans une base de données : Catalogue bibliographique décrivant des ouvrages numérisés ;
- Internes et encapsulées dans la ressource : En-tête d'un document dans une DTD: <ead><eadheader>…</eadhe ader></ead>.

3.7. Les normes et standards de métadonnées

La normalisation des ressources est une tâche fondamentale qu'il faut prendre en compte, surtout dans toute institution éducative. Plusieurs réflexions concernant l'indexation des ressources pédagogiques sont menées au sein des groupes de travail de l'Enseignement Supérieur. La difficulté principale était de savoir quel *système* de métadonnées utiliser.

Norme **:** Ensemble de règles fonctionnelles ou de prescriptions techniques relatives à des produits, à des activités ou à leurs résultats, établies par consensus de spécialistes et consignées dans un document produit par un organisme, national ou international, reconnu dans le domaine de la normalisation. Les normes permettent de fournir une certaine garantie de performance, de qualité, d'interchangeabilité.

Standard **:** Ensemble des règles et des prescriptions techniques établies pour une entreprise et qui servent à fixer les caractéristiques permettant de définir un élément de matériel ou de construction utilisé pour un projet donné.

Spécification **:** Ce terme désigne les exigences techniques auxquelles doit répondre un produit, un processus ou un service. Ces exigences peuvent être indépendantes d'une norme.

A partir de l'usage des métadonnées, il convient de préciser que la description des documents Web par ces éléments n'est pas un objectif final mais plutôt un moyen pour faciliter l'usage de ces documents dans une perspective de recherche d'informations.

Les métadonnées sont des outils importants pour le développement de la description de documents électroniques sur Internet. Les usages de ces éléments peuvent être répartis en :

- *Usage spécifique* pour la description du document lui même.
- *Usage générique* pour faciliter l'affichage des documents par les navigateurs et l'indexation de ces documents par les moteurs de recherche tout en permettant une normalisation de la description des ressources électroniques dans un contexte réseaux.

A partir de l'usage des métadonnées, il convient de préciser que la description des documents Web par ces éléments n'est pas un objectif final mais plutôt un moyen pour faciliter l'usage de ces documents dans une perspective de recherche d'informations. Dans ce cadre, plusieurs projets ont été engagés dans un objectif d'unification de la description des documents web. Parmi ces standards on trouve le Dublin Core, IMS, SCORM, LOM, RDF

3.8. Les standards des méta données

Dans cette section nous allons présenter les standards actuels de métadonnées élaborées par la communauté du E-Learning. Nous allons concentrer notre présentation sur le LOM du IEEE, du moment qu'il constitue le standard actuel adopté pour le domaine.

3.8.1. Le standard Dublin Core

La norme de métadonnées du Dublin Core résulte d'un ensemble de métadonnées communes à diverses communautés. Il s'agit d'une initiative définie en 1995 par le NSCA « National Center for Supercomputing Applications » et l'OLC « Online Computer Library Center ». le dublin Core propose un mode de catalogage sur Internet devant respecter deux objectifs : d'une part, être plus accessible aux usagers que les traditionnels formats, d'autre part faciliter l'interopérabilité des applications. Au sein de ce projet de catalogage, Dublin Core a développé un système de balises de méta-donnée encodables en HTML ou XML. Il fournit un noyau commun de sémantique pour la description des ressources des systèmes de gestion d'information de diverses communautés. Le schéma de métadonnées Dublin Core est composé d'un ensemble de 15 éléments censés décrire une large variété de ressources en réseau. Chaque élément est optionnel et peut être répété

1. **Titre :** Le nom donné à la ressource par le créateur ou l' auteur ;
2. **Auteur ou Créateur** : La personne ou l'organisation principalement responsable de la création du contenu intellectuel de la ressource ;
3. **Sujet et mots-clef :** Le sujet de la ressource. Typiquement, le sujet sera décrit par un ensemble de mots-clefs ou de phrases qui précisent le sujet ou le contenu de la ressource ;
4. **Description :** Une description textuelle du contenu de la ressource, y compris un résumé, dans le cas d'objets tels que des documents,ou une description du contenu dans le cas de ressources visuelles ;
5. **Editeur :** L'entité responsable de la diffusion de la ressource dans sa forme actuelle, telle qu'une maison d'édition, un département universitaire, une entreprise ;
6. **Autre contributeur :** Une personne ou une organisation, non mentionnée dans un élément créateur, qui a fait une contribution intellectuelle significative à la ressource mais dont la contribution est secondaire comparée à celle de toute personne ou organisation spécifiée dans un élément créateur (par exemple un rédacteur, un traducteur, un illustrateur) ;
7. **Date :** La date à laquelle la ressource a été publiée dans sa forme actuelle ;
8. **Type de ressource :** La catégorie de la ressource, telle qu'une page personnelle, un roman, un poème, un document de travail, un rapport technique, une dissertation ou un dictionnaire ;
9. **Format :** Le format de la ressource, utilisé pour identifier le logiciel et, éventuellement, le matériel qui peuvent être nécessaires pour afficher ou traiter la ressource ;
10. **Identificateur de la ressource :** Chaîne de caractère ou nombre utilisé pour identifier de façon unique la ressource ;

11. **Source :** Une chaîne de caractère ou un nombre, utilisé pour identifier de façon unique le travail d'où la ressource est dérivée, si applicable ;
12. **Langage :** Langage(s) du contenu intellectuel de la ressource ;
13. **Relation** : Les relations de cette ressource avec d'autres ressources. Le but de cet élément est de fournir un moyen d'exprimer les relations formelles entre des ressources qui existent aussi en temps que ressources indépendantes ;
14. **Couverture :** Définit les caractéristiques spatiales et/ou temporelles de la ressource ;
15. **Gestion des droits :** Un lien sur les droits de reproduction, les droits d'utilisation, ou renvoi à un service capable de fournir l'information sur les conditions d'accès à la ressource ;

Le Dublin core ne décrit pas la manière selon laquelle les métadonnées doivent être représentés. Le schéma DC a été conçu pour qu'il soit utilisé dans l'univers ouvert du Web caractérisé essentiellement par des utilisateurs qui ne sont pas forcément familiarisés avec l'usage des métadonnées et des ressources qui ressemblent, généralement peu, aux documents textuels traditionnels. Ces conditions d'usage des métadonnées DC ont déterminé d'une certaine manière leurs caractéristiques.

Les concepteurs ont ainsi insisté sur les aspects suivants : simplicité, extensibilité, interopérabilité sémantique, consensus international, flexibilité et procédure d'évolution continue. Par exemple, l'extensibilité est assurée par des éléments répétitifs qui peuvent être qualifiés afin de fournir des définitions plus étoffées. Chacun des aspects évoqués est justifié par les usages possibles des métadonnées.

3.8.2. Instructional Management System « IMS »

Le consortium IMS (*Instructional Management Systems*) a été lancé en 1994. Il rassemble un nombre important d'entreprises du secteur multimédia de formation, ainsi que des organismes et institutions à vocation éducative. La standardisation recherchée concerne les domaines suivants : [AMER00]

- La description des matériaux pédagogiques, afin de faciliter la publication et la recherche sur le Web ;
- L'interopérabilité des matériaux entre eux ;
- L'interopérabilité des plates-formes avec les matériaux et le système d'information des organisations ;
- L'enregistrement des informations sur les apprenants ;
- L'échange des données pour les systèmes d'administration.

IMS a réalisé un ensemble de spécifications pour structurer le contenu pédagogique, modéliser un étudiant, ou modéliser à un niveau général comment décrire, référencer et échanger des connaissances, des compétences, des tâches ou des qualifications.
Les membres de chaque ensemble sont tirés d'un dictionnaire de champ de metadata commun.. Des outils de création de metadata ont été développés, pour faciliter la création des modules.

Le standard de metadata IMS comprend 35 champs tirés du Dictionnaire des Champs IMS :

- Résumé;
- Auteur;
- Catalogue ID;
- Concepts;
- Type de contenant;
- Crédits;
- Date d'expiration;
- Forme;
- Format;
- Guide;
- Niveau d'intéractivité;
- Mots clés;
- Language;
- Niveau d'apprentissage;
- Localisation;
- Version de la metadata;
- Objectifs;
- Pedagogie;
- Plate-forme;
- Pré-requis;
- Presentation;
- Code des prix;
- Relation;
- Rôle;
- Taille;
- Source;
- Distributeur;
- Structure;
- Sujet;
- Titre;
- Droits d'utilisation;
- Support de l'utilisateur;
- Temps d'utilisation;
- Date de la version;
- Version.

La plupart des champs de Metadata sont structurés, permettant des termes multiples, et des hiérarchies. Ceux-ci ont été définis en utilisant le Format de Définition de Ressource du W3C (RDF).

Le système de metadata IMS possède différents types de contenants (objets) et définit un minimum pour chacun. Tous les types incorporent un ensemble de cores de metadata, ***l'IMS core***. Ce dernier comprend quatre (4) types de contenants:

- L'article ;
- Le module ;
- Le profil ;
- L'outil.

Les metadata IMS sont dérivées du Dublin Core. Nous y retrouvons d'ailleurs tous les éléments du Dublin Core, en plus d'autres éléments plus spécifiques.

3.8.3. Ressource Description Framework « RDF »

Une autre norme accroît les possibilités d'exploitation des métadonnées : RDF (*Resource Description Framework*) est une création du W3C4 1 qui facilite le traitement des métadonnées. Ce format fournit l'interopérabilité entre les applications qui échangent de l'information non compréhensible par les machines du Web. RDF permet d'insérer des métadonnées qui permettent aux utilisateurs de prendre d'avantage contrôle de l'utilisation de leurs données personnelles lorsqu'ils visitent des sites Web.

3.8.4. Sharable Content Object Reference Model « SCORM »

Le SCORM (*Sharable Content Object Reference Model*) a été développé par ADL Co Laboratory, l'University of Wisconsin et le Wisconsin Technical College System. Il définit des règles régissant l'instauration d'un modèle de gestion de l'apprentissage par l'utilisation du Web. Ce modèle comprend un format de structure de cours (*Course Structure Format*) qui permet de transférer plus facilement des contenus en définissant les éléments, la structure et les références externes. Cette norme vise à répondre à trois problèmes :

- Le transfert d'un contenu d'une plate-forme à une autre, notamment les informations relatives aux apprenants ;
- La création de matériaux granulaires utilisables dans différents modules ;
- La recherche informatisée de matériaux pour la formation.

Les moyens mis en oeuvre sont la normalisation de la description des matériaux et des fonctionnalités d'échange sur les réseaux de ces matériaux. SCORM a ainsi défini des spécifications permettant de représenter la structure d'un cours, ou les métadonnées destinées à la description documentaire.

L'analyse effectuée dans cette première partie permet de percevoir l'effervescence qui se vit en matière de normalisation de la formation en ligne et la multitude des acteurs qui interviennent dans ce processus. Aujourd'hui, ces acteurs se tournent vers des stratégies de partage et d'échange du matériel pédagogique numérisé. Les viviers de connaissances, ou LOR (*Learning Objects Repositories*), permettent de partager les objets pédagogiques et, avec l'introduction de normes et de standards dans le monde de la formation en ligne, visent l'interopérabilité des systèmes et la portabilité des ressources pédagogiques.

---

1 (*World Wide Web Consortium*)

3.8.5. Learning Object Metadata « LOM »

Dans le domaine de l'éducation et du e-learning, et face à la croissance exponentielle des documents pédagogiques (cours, travaux dirigés, travaux pratiques, définitions, exemples, démonstration, axiomes, ...) sur le Web, il convient de recourir à des procédés de qualification par les métadonnées qui sont utilisés par la majorité. D'où l'utilité du modèle de données LOM [CHRI02].

En ce qui concerne le **Standard for Learning Object Metadata (LOM) de l'IEEE** approuvé en décembre 2002 : il a eu plusieurs versions avant la standardisation officielle qui ont été adoptées par des groupes tels que l'IMS21. Il existe un groupe de travail, le **CEN/ISSS WSLT**, qui travaille sur la traduction des LOM dans les différentes langues européennes. L'objectif principal de ce projet est d'internationaliser les LOM. Il proposera aussi un entrepôt de taxonomies et vocabulaires pour une "Société Européenne Apprenante".

Dans le domaine des technologies éducatives, les travaux de normalisation les plus avancés concernent les métadonnées pédagogiques. L'objectif principal de cette normalisation est de faciliter la réutilisation, la production des documents pédagogiques numériques étant difficile.

Le standard LOM est construit au-dessus du Dublin Core en le complétant par des extensions propres au domaine éducatif. Il spécifie la syntaxe et la sémantique des métadonnées pédagogiques et définit les attributs nécessaires pour une description adéquate et complète des « objets pédagogiques ».

*** Objectif de « LOM »**

- *Coté utilisateur*
  - **Retrouver** des ressources pédagogiques.
  - **Échanger** des ressources pédagogiques.

- *Coté producteur*
  - **Partager** l'information dans un contexte où les ressources sont nombreuses et leur production coûteuse ;
  - **Réutiliser** les ressources ou leurs composants ;
  - **Être interopérable** avec des systèmes de Gestion de la Formation (Learning Management Systems).

Ce standard de métadonnées pédagogique dit « LOM » se limite à l'ensemble minimal de caractéristiques indispensables pour gérer les « documents pédagogiques », les rechercher et les évaluer.

Il permet de décrire les caractéristiques pédagogiques des ressources :

- Type et niveau d'interactivité ;
- Type de ressource : exercice, QCM, examen, diagramme, figure, etc… ;
- Type d'utilisateur : professeur, élève ;
- Contexte (primaire, secondaire, supérieur, formation continue) ;
- Classe d'âge typique ;
- Difficulté ;
- Durée ;
- Description ;
- Langue etc…

- ***Indexation par les métadonnées de LOM***

Le schéma de description proposé par le LOM s'inspire fortement des techniques documentaires avec comme problématique principale la gestion, sous ses différentes facettes, du contenu. Cette démarche d'organisation du contenu ne permet pas de décrire l'activité de l'apprenant et favorise, par conséquent, un mode d'apprentissage axé sur la consultation des ressources pédagogiques structurées en leçons, cours, modules, etc.

Les éléments de LOM pour décrire un document pédagogique sont regroupés en neuf catégories. Un aperçu, général de ces descripteurs est représenté sur la figure suivante : [PASS03]

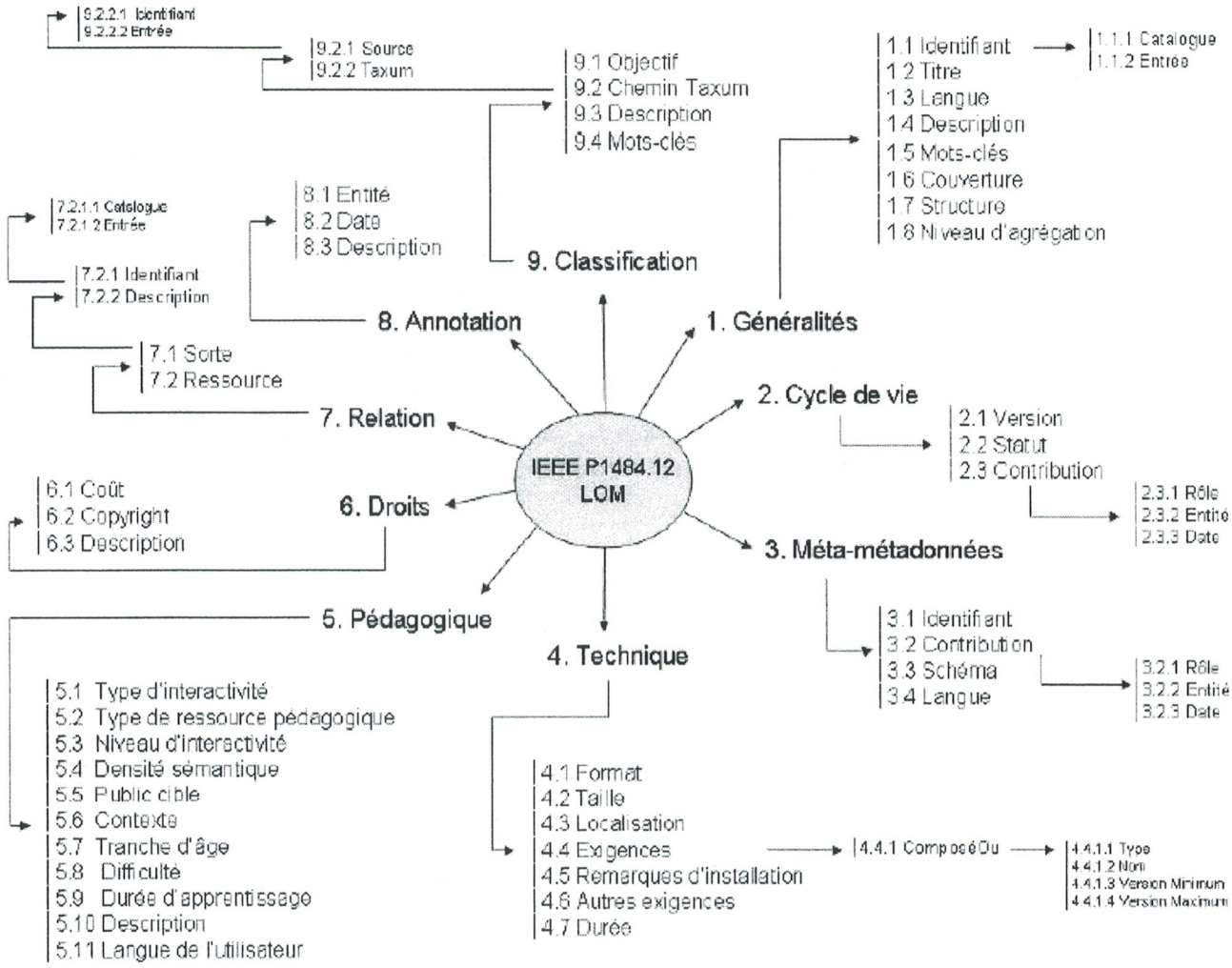


Figure 3.1: Organisation du schéma de métadonnée LOM

On peut détailler ce schéma afin d'expliquer le rôle de chaque descripteur. On aura alors :

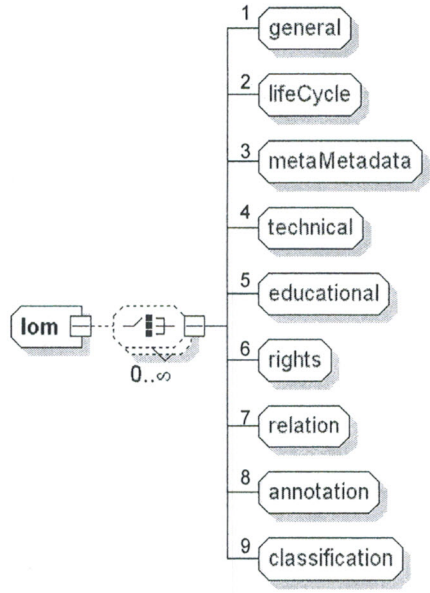


Figure 3.2 : catégories de LOM

a. Description générale : « general »

Dans la catégorie « ***general*** » le document pédagogique l'objet est décrit dans son ensemble. On y trouve des données sur l'identifiant du document, son titre, sa description, la liste des langues utilisées, une liste de mots clés, l'étendue de la ressource (temps, géographie, culture ...), le type de structure (collection, linéaire, hiérarchique ...), son niveau de granularité (de 1 à 4, 1 désignant un cours entier).

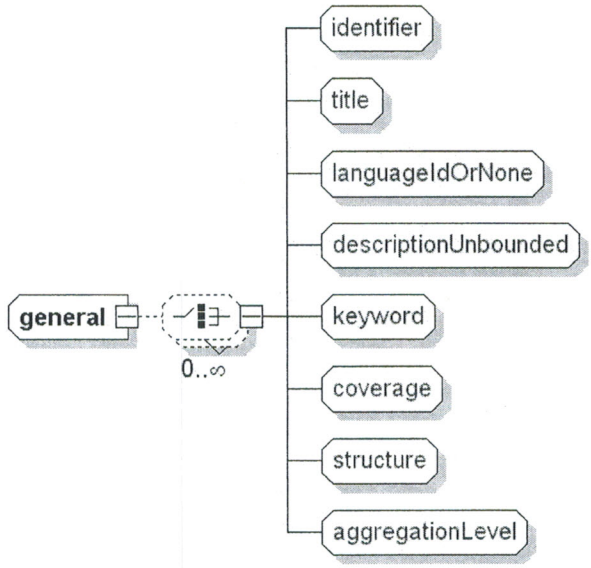


Figure 3.3: descripteur « Général »

b. Cycle de révision : « lifecycle »

Cette catégorie permet de décrire les caractéristiques relatives à l'historique et à l'état courant du document pédagogique (*draft*, *final*...), les personnes qui l'ont modifié, à quelle date ainsi que leur rôle (*author*, *publisher*, *instructional designer*...). Cette partie décrit la liste complète des modifications ou cycle de révision.

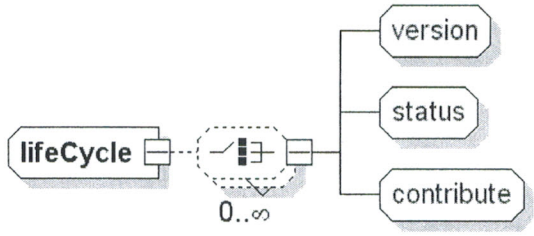


Figure 3.4: descripteur « Lifecycle »

c. Métadonnées sur les métadonnées : « metametadata »

C'est un ensemble de métadonnées sur les métadonnées décrivant le document pédagogique. Cet ensemble décrit le schéma ou la spécification utilisée (*metadatascheme*). Il est possible de satisfaire à plusieurs schémas et de définir des liens dans un système de catalogage connu.

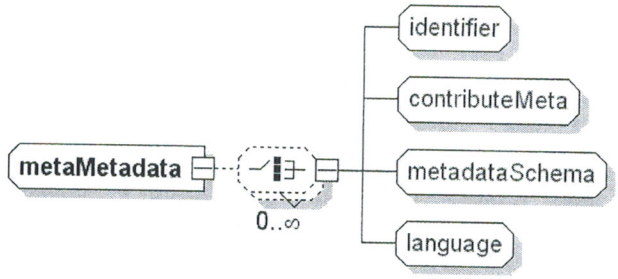


Figure 3.5: descripteur « Metametadata »

d. Les informations techniques : «Technical »

Cette partie définit les exigences techniques en terme de navigateur (type, version), de système d'exploitation ou les caractéristiques comme le type des données ou format (permettant d'identifier les logiciels nécessaires pour les lire), la taille du document numérique (en octets), sa localisation physique (URL *Uniform Resource Locator* ou URI UR *Identifier*), des informations pour installer le document pédagogique et sa durée (en particulier pour les fichiers de type son, animation ou vidéo).

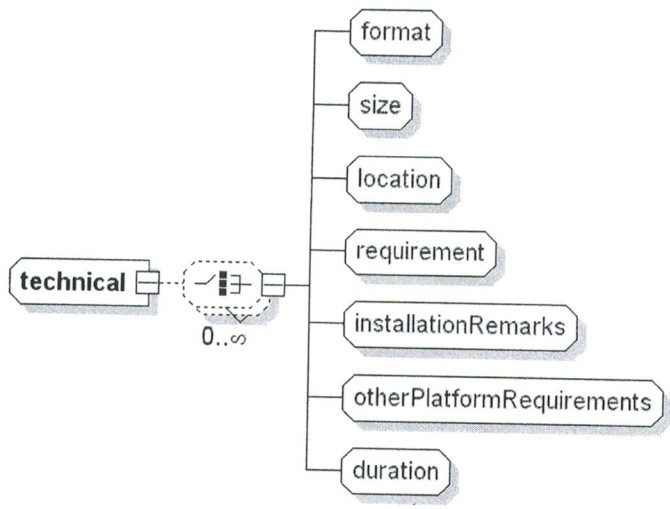
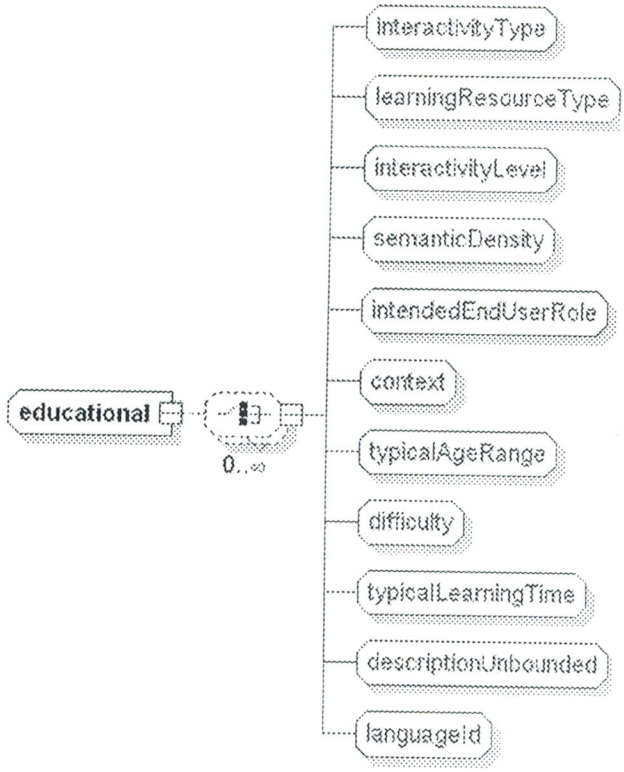


Figure 3.6: descripteur « Technical »

e. La partie pédagogique : « educational »

Cette partie permet de définir les conditions d'utilisation de la ressource « caractéristiques pédagogiques ». C'est souvent par ces caractéristiques que l'on améliore l'exploitation du LOM.

- ***Interactivity Type*** **:** le type d'interaction entre la ressource et l'utilisateur typique *(Active, Expositive, Undefined)* ;
- ***Learning Resource Type*** **:** le type pédagogique *(Exercise, Simulation. . .)*, peut être présent plusieurs fois ;
- ***InteractivityLevel : degré d'interactivité ;***
- **SemanticDensity :** densité sémantique (Very Low, Low, Medium, High, Very High)
- ***Intended end user role*** **:** utilisateur de la ressource ;
- ***Context*** **:** environnement d'utilisation de la ressource ;
- ***Typical Age Range*** : âge de l'utilisateur ;
- ***Difficulty*** **:** difficulté de la ressource ;
- ***Typical Learning Time*** **:** temps approximatif ou typique pour travailler avec la ressource
- ***Description*** **:** commentaires sur l'utilisation de la ressource ;
- ***Language*** **:** la langue de l'utilisateur.



f. La gestion des droits : « rights »

Cette partie concerne les droits (*copyright*) liés à la ressource pédagogique, éventuellement son coût.

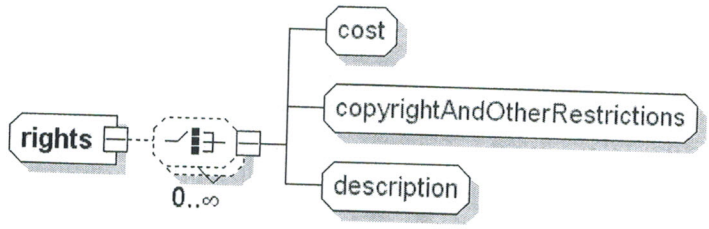


Figure 3.8: descripteur « Rights »

g. L'aspect relationnel : « relation »

Cette catégorie couvre les relations ou liens avec d'autres documents pédagogiques en précisant le type de relation (« *...est requis par...* », « *...est une partie de ...* »...).

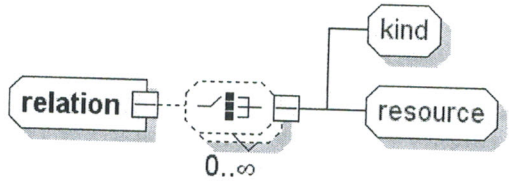


Figure 3.9: descripteur « relation »

h. Annotation

Cette partie regroupe les annotations ou commentaires.

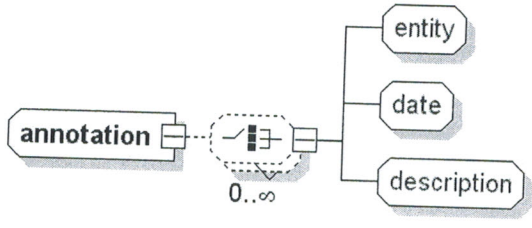


Figure 3.10: descripteur « annotation »

i. Classification

Cet ensemble indique l'appartenance de la ressource à une ou plusieurs instances de classifications, permettant entre autre de définir le type de la ressource « *discipline, idée, prérequis, objectifs pédagogiques, restriction, accessibilité, niveau pédagogique, niveau de compétence, niveau de sécurité* »

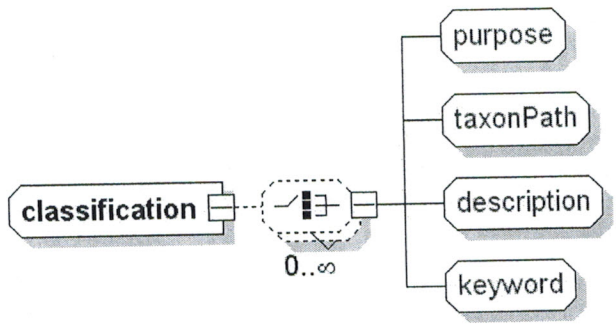


Figure 3.11: descripteur « classification »

Remarque

Ces descripteurs ne sont pas obligatoires mais certains peuvent être répétés. La question qui se pose est la suivante : quelle est la granularité des documents pédagogiques supportée par ce standard ? Parmi tous les descripteurs, on en trouve un (dans la catégorie « *general* ») donnant une indication sur la granularité de la ressource, c'est le descripteur *niveau de granularité*. Il peut prendre les valeurs suivantes :

- Le niveau le plus petit d'agrégation, par exemple : des données brutes ou des fragments ;
- Un ensemble d'atomes, par exemple : un document HTML comprenant des images ou bien une conférence ;
- Un ensemble de ressources de niveau 2 comme un site web avec un sommaire ou un cours entier ;
- Le niveau le plus gros, par exemple un cursus préparant à un diplôme.

Ainsi, ce standard ne prend pas position sur la taille du granule indexable. Bien que les techniques d'indexation soient maîtrisées par les documentalistes, le problème qui se pose ici est différent car il s'agit d'indexer non seulement un livre entier, mais aussi ses chapitres, voire ses paragraphes.

Or, un document pédagogique non indexé est un document pédagogique qui ne sera pas retrouvé et qui ne pourra pas être réutilisé. Cette étape d'indexation est donc indispensable. Fort heureusement, des logiciels peuvent prendre en charge automatiquement une partie des descripteurs (comme le champ auteur, si c'est celui-ci qui indexe) et peuvent aider à la saisie des autres descripteurs. Parmi tous ces descripteurs, certains peuvent être vus comme « objectifs » (titre, auteur, langue. . .) et d'autres comme subjectifs (densité sémantique. . .).

Afin d'illustrer l'usage du schéma LOM, nous avons formaté un ensemble de ressources pédagogiques selon le standard *Learning Object Metadata* (IEEE-LTSC 1484.12.1). Il s'agit d'un cours organisé en leçons permettant de comprendre et de manipuler les fonctions (modèles de document, macros, etc.) avancées de Word. Le cours est présenté sous forme de fichiers html et est destiné aux étudiants en premier cycle. L'exemple ci-dessous présente l'écriture en XML des métadonnées associées au cours. [CHRI02]

**** Exemple d'utilisation du LOM dans un « Content Packaging »***

Pour illustrer ce standard, nous avons formaté un ensemble de ressources pédagogiques selon les spécifications du *Content Packaging* et du LOM :

Pour donner leurs valeurs aux éléments descriptifs, le LOM utilise plusieurs ensembles de balises « *outils* » :

langstring (balise <langstring>) permet de préciser des valeurs selon plusieurs langues (avec l'attribut *xml:lang)*,

vocabulary (balises <source>, <value>) qui permet de les choisir en accord avec un vocabulaire défini,

date (balises <datetime>, <description>)

vcard (balise <vcard>) pour la « carte de visite virtuelle »

```
<lom>
<general>
  <title> <langstring xml:lang="fr"> Initiation à Microsoft Word </langstring>  </title>
   <language>fr</language>
<description>
<langstring xml:lang="fr">Un support de cours présentant les fonctionnalités avancées de Microsoft Word,
c'est  une ressource pédagogique pour l'enseignement de l'informatique de base en ligne </langstring>
</description>

<keyword>
<langstring xml:lang="fr"> Word</langstring>
<langstring xml:lang="fr">Macro</langstring>
</keyword>

<structure>
<source> <langstring xml:lang="xnone"> LOMv1.0</langstring> </source>
<value> <langstring xml:lang="xnone"> Linear</langstring> </value>
</structure>

<aggregationlevel>
<source> <langstring xml:lang="xnone"> LOMv1.0</langstring> </source>
<value> <langstring xml:lang="xnone"> 3</langstring> </value>
</aggregationlevel>
</general>

<lifecycle>
<contribute>
<role> <source> <langstring xml:lang="xnone"> LOMv1.0 </langstring> </source>
<value> <langstring xml:lang="xnone"> Author </langstring> </value>
</role>
    <centity> <vcard>begin:vcard  EMAIL; INTERNET:iles@univ-tlemcen end:vcard </vcard> </centity>
</contribute>
<contribute>
```

```
<role> <source> <langstring xml:lang="xnone"> LOMv1.0</langstring> </source>
<value> <langstringxml:lang="xnone"> Publisher</langstring> </value>
</role>
<date>15-05-2005</date>
</contribute>
</lifecycle>

<metametadata>
  <contribute>
<role> <source> <langstring xml:lang="xnone"> LOMv1.0</langstring> </source>
  <value> <langstring xml:lang="xnone"> Creator</langstring> </value>
</role>
<centity>
  <vcard> begin:vcard fn: Chikh Azzeddine end:vcard
  </vcard>
</centity>
<date>
  <datetime>2005-05-1</datetime>
   <description> <langstring xml : lang="fr">Date description</langstring> </description>
    </date>
 </contribute>
<metadatascheme>LOMv1.0</metadatascheme>
<language>fr</language>
</metametadata>

<technical>
<format>doc </format>
<size>35000</size>
<requirement>
<name>
  <source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
  <value><langstring xml:lang="xnone">Microsoft winword</langstring></value>
    </name>

<minimumversion>7.0</minimumversion>
</requirement>
</technical>

<educational>
<interactivitytype>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">Active</langstring></value>
</interactivitytype>
<learningresourcetype>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">simulation</langstring></value>
</learningresourcetype>

<context>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">Université tlemcen  1er Cycle</langstring></value>
</context>


<interactivitylevel>
```

```
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">high</langstring></value>
</interactivitylevel>

<intendededenduserrole>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">étudiant</langstring></value>
</intendededenduserrole>

<difficulty>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">moyen</langstring></value>
</difficulty>
<language>fr</language>
</educational>

<rights>
<cost>
<source><langstring xml:lang="xnone">LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">non</langstring></value>
</cost>
</rights>

<classification>
<purpose>
<source><langstring xml:lang="xnone"> LOMv1.0</langstring></source>
<value><langstring xml:lang="xnone">but educatif</langstring></value>
</purpose>
<description><langstring xml:lang="fr">Pratique, traitement de texte</langstring> </description>
</classification>
</LOM>
```

Listing 3.1 : Exemple d'un document annoté avec le modèle LOM

Remarque et commentaire

Dans l'exemple ci-dessus, nous nous sommes contentés de renseigner les champs (éléments) recommandés pour une meilleure exploitation des ressources dans un environnement d'apprentissage. Notre description de la ressource n'est donc pas fausse mais incomplète.

L'exemple présenté, illustre la difficulté que rencontre l'auteur des métadonnées pour remplir les différents champs (79) du LOM. Cette difficulté résulte, à notre avis, de la manière selon laquelle le modèle a été conçu. Ce problème sera résolu dans notre projet, après l'utilisation du langage XML où le nombre de champs n'est pas fixe, et non limité

L'analyse des éléments constitutifs du LOM permet de déduire que l'effort principal de ses concepteurs a porté essentiellement sur la détermination des métadonnées permettant une description efficace des objets, leur partage et leur réutilisation. Le schéma de description proposé par le LOM s'inspire fortement des techniques documentaires avec comme problématique principale la gestion, sous ses différentes facettes, du contenu

- ***Apports du learning Object Metadata « LOM »***

La caractérisation des ressources pédagogiques à l'aide du modèle de description "LOM" présente plusieurs intérêts. Tout d'abord, la modularité et la réutilisation des fragments ou briques d'informations se trouvent facilitées à la manière d'un jeu de construction

Le cours devenant assimilable à une collection d'objets à travers une méta description offerte par le « LOM », permet une fabrication facilitée en puisant dans le gisement d'objets pédagogiques, une mise à jour plus simple ainsi qu'une diffusion personnalisée.

Décrire des ressources pédagogiques avec le modèle « LOM » permet de construire des cours qui sont adaptés à un public cible et par la suite, construire des systèmes tutorial simples ou intelligents.

Le LOM apporte aux ressources pédagogiques des caractéristiques et une description à la fois fines et précises pour pouvoir personnaliser l'offre de formation en fonction des différents apprenants mais aussi universelles pour être utiliser par le plus grand nombre, dans des systèmes ouverts comme des sites Web.

3.9. Conclusion

Le grand défi de la recherche d'informations sur le Web est de pouvoir cibler au maximum les résultats de recherche suite à une requête de l'utilisateur. Ce défi peut être franchi par une structuration de la description des ressources électroniques en utilisant les métadonnées. Ces éléments jouent un rôle de plus en plus importants dans la réussite ou l'échec d'un document pédagogique. C'est sur leur qualité que repose la pertinence de la recherche et la satisfaction de l'utilisateur.

Nous venons de passer en revue les principaux standards de metadata existants, même s'ils diffèrent souvent dans leurs structures, il n'en demeure qu'ils poursuivent un objectif principal commun :offrir des éléments de description de l'information pour en faciliter l'accès.

Ainsi, on peut conclure sur l'importance de la normalisation du standard LOM dans une perspective d'unification de la description des ressources pédagogiques. Face à ce phénomène, d'autres projets de métadonnées ont été lancés, notamment avec l'apparition du langage XML (eXtensible Markup Language).

Ce développement montre l'importance de plus en plus croissante des métadonnées dans la description et la diffusion des documents

PARTIE II – ENTREPÖT DE DONNEES

Partie II – Entrepôt de données

Toute université possède actuellement d'importants volumes de documents, stockés le plus souvent dans différents médias (bases de données, documents papiers,...) et a besoin d'outil permettant une exploitation efficace et performante de ces données.

La constitution d'entrepôts de données est une réponse à cette problématique. Dans le domaine du E-learning, l'entrepôt de données regroupe, sous une forme exploitable par des traitements utiles pour l'aide à la décision, les informations extraites de leurs sources et qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine. En quelques années, les entrepôts de données se sont imposés comme une solution rentable pour faire face aux besoins des universités en termes de capitalisation de connaissances et d'aide à la décision.

Ainsi, cette partie va exposer dans le chapitre 4 les concepts de base des entrepôts de données. Le chapitre 5 présentera un état de l'art sur les différents travaux réalisés dans ce domaine. On terminera cette partie par un bilan qui va synthétiser l'ensemble des travaux cités, et on exposera la problématique de notre approche.

4. CHAPITRE – Concepts de base des entrepôts de données

4.1. Introduction

Tout entreprise possède actuellement d'importants volumes de données stockés le plus souvent dans différents médias « bases de données, documents papiers, ... » sui seront manipulées par des utilisateurs. L'intégration et l'exploitation de ces données dans un but d'analyse et de décision s'avère difficile : « elle est réalisée souvent de manière imparfaite par des moyens classiques tel que des requêtes SQL « Structured Query Langage » ». Pour répondre à ces besoins, les entreprises ont besoin d'outils et de modèles pour la réalisation de systèmes décisionnels.

La constitution d'entrepôts de données est une réponse au problème de l'intégration d'une grande quantité de données variées, relatives à un certain domaine d'application, et stockées physiquement dans différentes sources de données. L'entrepôt de données regroupe, sous une forme exploitable par des traitements utiles pour l'aide à la décision, les informations extraites de ces sources et qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine.

4.2. Définition et objectifs d'un entrepôt de données

De nombreuses définitions ont été proposées cherchant à orienter ces définitions dans un sens mettant en valeur leur produit. La définition la plus appropriée est celle de Inmon ou il définit l'entrepôt comme "une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse".

D'après [DEVL97], un entrepôt de données (*data warehouse*) est un ensemble *unique*, *complet* et *consistant* de données obtenues depuis une variété de *sources* et rendu *disponible* à des utilisateurs finaux à des fins de compréhension et de gestion des affaires décrites par ces données.

L'entrepôt de données stocke des données nécessaires à la prise de décision ; il est alimenté et mis à jour via des extractions de données portant sur les bases de production qui sont considérées dans la chaîne décisionnelle comme les "sources de données".

Les données de l'entrepôt doivent respecter les caractéristiques suivantes : [CAST99]

- ***Orientées sujet :*** Les données de l'entrepôt sont organisées par sujets ou thèmes d'analyse, ce qui permettra de rassembler toutes les données pertinentes par rapport à un sujet et nécessaires aux besoins d'analyse ;
- ***Intégrées :*** les données d'un entrepôt sont le résultat de l'intégration de données en provenance de sources différentes et hétérogènes ;
- ***Historisées :*** l'entrepôt doit permettre l'historisation des données ç-à-d garder leurs différentes versions ;
- ***Filtrées :*** l'entrepôt ne doit contenir que les données pertinentes et utiles pour l'avenir ;
- ***Non volatile :*** les données sont matérialisées et ne sont pas modifiables par l'utilisateur. L'entrepôt est mis à jour de façon constante.

Parmi les objectifs d'un entrepôt de donnée :

- Il assure l'accès aux informations de l'entreprise ;
- Ses informations sont cohérentes ;
- Les données d'un entrepôt doivent pouvoir être séparées et combinées au moyen de toutes les mesures possibles de l'activité ;
- C'est un lieu où sont publiées les données qui ont déjà servi ;
- La qualité de l'information d'un entrepôt de données est l'un des ressorts de la réorganisation de l'activité.

L'entrepôt de données offre à l'entreprise les avantages suivants : [BOUL04]

- Il constitue une collection de données centralisée disponible pour l'aide à la décision (OLAP, datamining,...) ;
- Les évolutions des données de l'entrepôt sont conservées (historisation des données) ;
- Il contient un ensemble de données consolidées (données homogènes et fiables) ;
- Il contient des données agrégées permettant une analyse à différents niveaux de détails ;
- Il permet de développer différents thèmes d'analyse (réorganisation en fonction des sujets à analyser) ;
- L'accès direct aux données : facile, rapide, sécurisé ;
- Plus de contrôle sur les données ;
- Possibilité d'ajouter facilement des annotations.

Mais introduit certains problèmes

- Besoin de maintenir à jour les copies ;
- Besoin de savoir quand et comment les données changent ;
- Besoin de recommencer le processus d'annotation en conséquence ;

Le problème majeur des entrepôts de données est du à l'hétérogénéité sémantique entre des données issues de bases différentes. Les données de l'entrepôt proviennent de différentes sources hétérogènes. L'intégration consiste à résoudre ce problème d'hétérogénéité des modèles, de la sémantique, ce qui cause un problème majeur pour les entrepôts de données. Par exemple, un conflit de terminologie peut survenir lorsqu'un même objet est désigné par des noms différents ou lorsqu'un même nom est utilisé pour deux objets différents.

4.3. Les entrepôts de données et magasins de données

L'entrepôt de données consiste à centraliser les informations nécessaires à des fins décisionnelles de l'entreprise en gardant leur évolution (historisation), alors que le magasin de donnée « Data mart » est un sous-ensemble de données et ciblé sur un sujet unique et répondant à un objectif décisionnel précis ou un besoin spécifique.

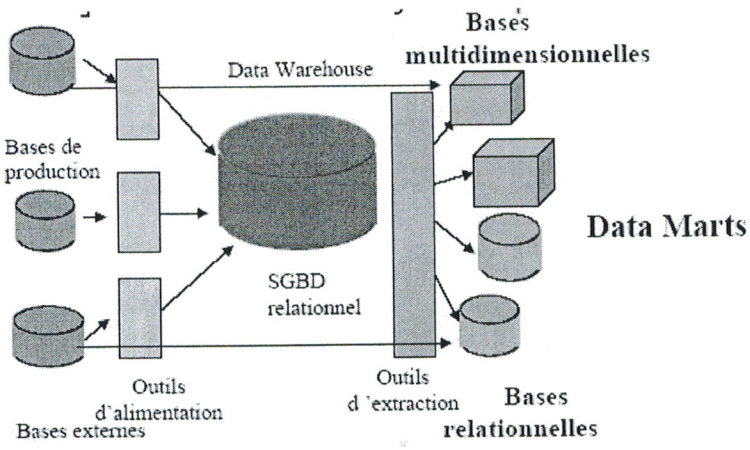


Figure 4.1 : architecture des magasins de données « Data mart »

L'entrepôt de données « Data warehouse » vise à stocker et centraliser l'ensemble des informations pertinentes et utiles d'une entreprise pour la prise de décision. Il a également pour objectif de gérer l'évolution des informations dans le temps. Un entrepôt de données est alors organisé selon un modèle informatique permettant de gérer simplement les données historisées (généralement un modèle relationnel).

Un magasin de données (extrait spécifique de l'entrepôt) est dédié à des analyses décisionnelles particulières. Ces analyses nécessitent la définition des modèles appropriés aux outils des décideurs. La modélisation multidimensionnelle est généralement utilisée à ce niveau puisqu'elle s'avère plus adaptée aux analyses décisionnelles.

4.4. Architecture d'un entrepôt de données

L'architecture d'un entrepôt est classique. On décrive 3 opérations essentielles : l'intégration, le stockage et l'analyse.

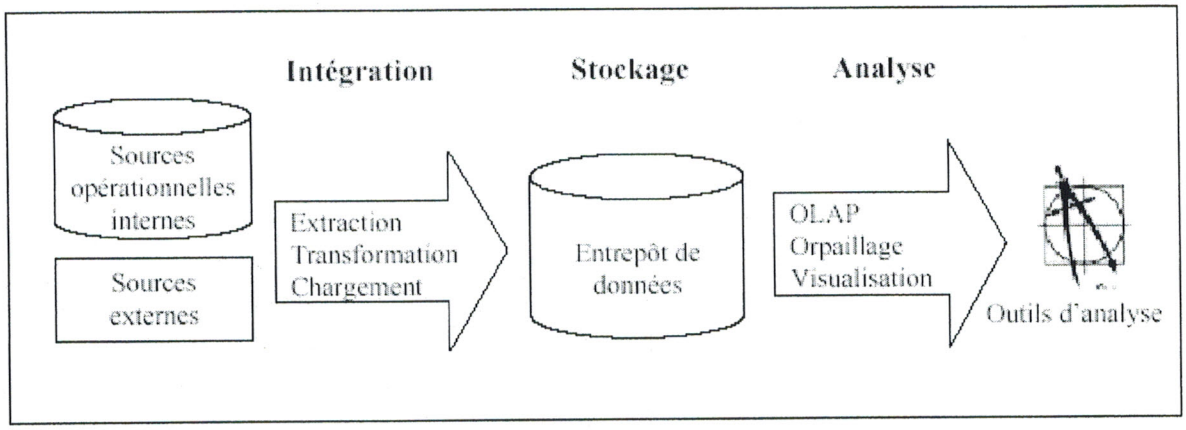


Figure 4.2 : Architecture d'un entrepôt de donnée

1- L'intégration des données

Deux principales approches permettent l'intégration des données: une *approche virtuelle* (souvent appelée approche par médiateur) et une *approche matérialisée* (approche par entrepôt).

Les approches virtuelles sont basées sur une hiérarchie de médiateurs, correspondant à des vues virtuelles, au-dessus des extracteurs. Les données ne sont stockées que dans leur source d'origine. Dans l'approche matérialisée, les données sont effectivement extraites, nettoyées, intégrées et stockées dans un entrepôt. Les requêtes sont posées directement sur les données de l'entrepôt.

Chacune de ces deux approches a des avantages et des inconvénients en termes de traitement des données intégrées. Il existe également des approches hybrides qui essaient de combiner les avantages de ces deux approches en matérialisant les données dans un entrepôt et en utilisant l'approche virtuelle pour leur intégration. Ainsi le problème de disponibilité et de capacités des ressources est diminué tout en gardant la flexibilité de l'approche virtuelle pour l'interrogation. Un exemple d'un tel système est Xylème où les données sources sont stockées dans le format XML (sans être transformées) et intégrées à travers un mécanisme de vues entre les DTD concrètes des sources et des DTD abstraites montrées à l'utilisateur.

a. Approche virtuelle

L'approche virtuelle est basée sur une hiérarchie de médiateurs, correspondant à des vues virtuelles. Les données ne sont stockées que dans leur source d'origine. L'utilisateur a une « vue intégrée » des différentes sources. Dès qu'un utilisateur formule une requête, celle-ci est envoyée à un médiateur, qui décompose la requête vers d'autres médiateurs ou vers des extracteurs. L'intégration se fait vue par vue par chaque médiateur, uniquement en réponse à une requête de l'utilisateur.

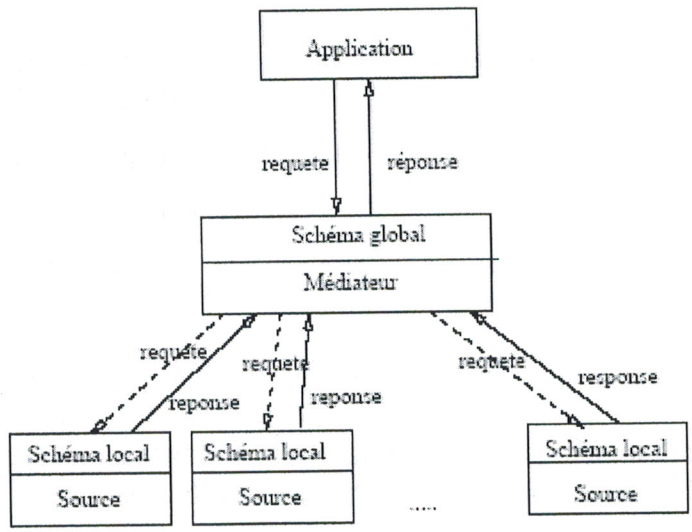


Figure 4.3 : Approche virtuelle

L'avantage de cette approche réside dans le fait qu'elle favorise l'intégration de sources qui sont toujours disponibles avec de mises-à-jour fréquentes, pas de coût de maintenance. Par contre, le temps de réponse est pénalisé par le fait de la distribution des données et de la nécessité de recomposer les résultats des différentes sous-requêtes rendues par les sources avant de présenter un résultat global à l'utilisateur.

b- Approche matérialisée

L'approche matérialisée consiste à stocker localement des données issues des différentes bases, dans un « entrepôt de données ». Les données sont effectivement extraites « par copies sélective », nettoyées, intégrées et stockées dans un entrepôt. Les requêtes utilisateurs sont traitées directement par l'entrepôt de données sans accéder aux sources.

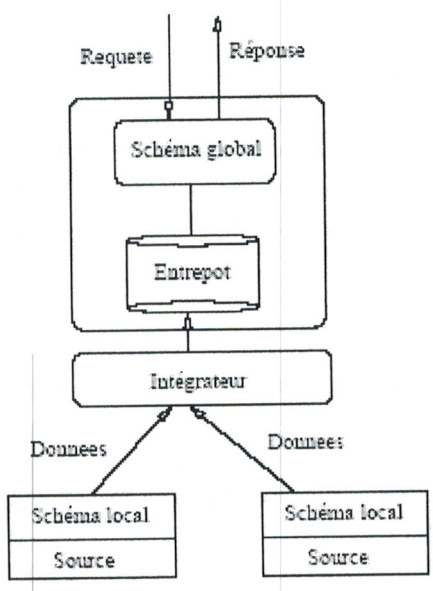


Figure 4.4 : approche matérialisée

b- Modèle ROLAP/OOLAP

Le passage du modèle conceptuel vers un modèle logique adapté au relationnel est connu sous la notion d'approche **ROLAP** (Relational On Line Analytical Processing). Dans ce cas, chaque fait correspond à une table appelée table de fait et chaque dimension correspond à une table appelée table de dimension. Le ROLAP est juste une extension du modèle relationnel dans lequel les opérations sur les données multidimensionnelles sont traduites en opérations relationnelles standards. Cette configuration supporte les gros volumes (gestion par le SGBDR) mais n'est pas facile à manipuler par l'utilisateur.

Une autre approche dite **OOLAP** (Object On Line Analytical Processing) est basée sur l'objet, c'est à dire sur les SGBD orienté objet. Un fait devient une classe de fait et une dimension devient une classe de dimension.

Par rapport à ROLAP, l'intérêt de l'approche OOLAP est sa plus grande richesse de modélisation. En effet, l'objet permet de modéliser facilement certains éléments comme les liens multivalués entre le fait et les hiérarchies par exemple. Cependant, la marché des SGBD est actuellement dominé par les offres relationnelles [CAST99].

c- Les systèmes hybrides OLAP « HOLAP »

En évitant les problèmes des systèmes ROLAP et MOLAP, l'utilisateur formule une requête sur la base de données relationnelle et le résultat s'affiche sous une forme multidimensionnelle. La convivialité est ainsi respectée tout en permettant le stockage des gros volumes d'information. Le cube peut être stocké sur le serveur : il est donc partagé entre l'ensemble des utilisateurs afin de résoudre les problèmes de partageabilité des informations.

3- Analyse

Une fois les données intégrées dans l'entrepôt, l'information doit être présentée et analysées de manière comprise par l'utilisateur, grâce à des outils d'accès, de traitement et de visualisation spécifique. Nous pouvons distinguer trois types d'outils : les outils OLAP, ceux dédiés à la fouille de données « datamining » et d'autres liés à la visualisation. Ces trois types sont détaillés dans ce qui suit :

3.1- les outils OLAP

Le terme OLAP (ON-line Analytical Processing, ou analyse en ligne) est un terme qui désigne un cahier des charges que doivent satisfaire la base de données ou l'entrepôt de données et les outils afférents pour stocker et interroger de façon efficace des volumes de données importants. Le cahier des charges (consistant de douze règles ou déclarations de principe) demande que l'analyse des données puisse se faire de manière interactive et rapide (on-line ou one-pass) pour des données quelconques (distribuées, multidimensionnelles, orientées objets, hiérarchisées, etc.).

Les outils OLAP utilisés consistent à effectuer la synthèse, l'analyse et la consolidation dynamique des données de l'entrepôt. Ces outils doivent aider les décideurs à effectuer des analyses, leur autorisant l'accès aux données de l'entrepôt et leur fournissant de puissants mécanismes d'interrogation. Ces mécanismes doivent comprendre des requêtes qui impliquent des agrégations, des classements et des prévisions.

En d'autres termes, les outils OLAP permettent d'analyser interactivement des indicateurs préalablement structurés par axe d'analyse. L'utilisateur navigue de façon intuitive dans les données par le biais de changement d'axe, de niveau d'agrégation, d'analyse du détail, etc

3.2- les outils de la fouille de données « Datamining »

La fouille de données consiste à extraire et à analyser, par des méthodes statistiques, un large volume de données puisés dans un entrepôt par exemple, en vue de découvrir des corrélations, des tendances ou des règles qui s'avéreront utiles pour déterminer des régularités. Ces modèles peuvent être des modèles de calculs « équations par exemple » ou des modèles logiques « des règles ». Ils visent à décrire le fonctionnement passé ou actuel d'un procédé, pour en prédire l'avenir. Ainsi, l'utilisateur oriente le traitement avec des paramètres initiaux servant à définir le modèle d'analyse. Puis, l'outil de « datamining » analyse le comportement de ces paramètres en fonction des données de la base pour fournir des conclusions allant dans le sens de l'analyse demandé [MORI02].

3.3- les outils de visualisations

La visualisation des données doit faciliter leur analyse et leur interprétation ; Les techniques de visualisation convertissent des données complexes en images, graphiques et en animations qui peuvent être analysées en cherchant des interrelations entre données. Les outils de visualisation autorisent l'utilisateur à explorer de grandes quantités de données.

3.4- Analyse multidimensionnelle

L'analyse multidimensionnelle est la capacité à manipuler des données qui ont été agrégées selon différentes dimensions. a technologie OLAP s'appuie sur des vues multidimensionnelles et fait appel à des cubes de données qu'ils peuvent extraire en choisissant des dimensions différentes, l'élément recherché se situe à l'intersection de chaque dimension à laquelle il se rattache [CAST99].

D'un point de vue conceptuel, la modélisation multidimensionnelle est liée aux concepts de fait et de dimension.

Un fait

Un fait représente un sujet d'analyse. Un fait se compose de mesures représentant les informations de l'activité à analyser. Une mesure est numérique, comme le chiffre d'affaires par exemple, souvent additive, apte à être manipulée via des opérateurs arithmétiques.

Exemple :

Considérons le fait Ventes pouvant être constitué des mesures d'activités suivantes: Quantité de produits vendus et Montant total des ventes.

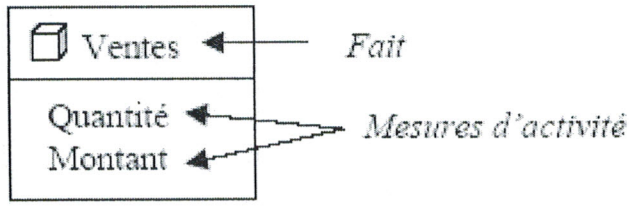


Figure 4.5: exemple de fait

Dimension

Une dimension représente une notion liée au métier, comme le client ou le fournisseur. Le sujet à analyser, en d'autres termes le fait, est analysé selon différents axes appelés dimensions. Une dimension représente donc une perspective d'analyse. Elle se compose de paramètres (ou attributs) correspondants aux informations pour lesquelles sont analysées les mesures.

Exemple :

Le fait Ventes de la figure peut être analysé selon les dimensions suivantes : Produits, Magasins et Temps.

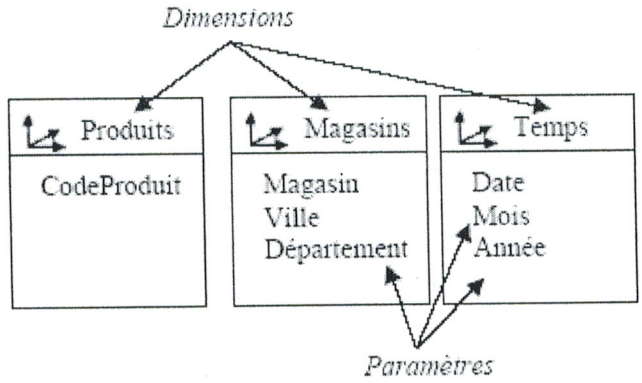


Figure 4.6 exemple de dimension

Hiérarchie

Les paramètres d'une dimension peuvent être organisés selon leur niveau de détail. Pour définir ces différents niveaux, chaque dimension est munie d'une ou plusieurs hiérarchies. Une hiérarchie organise les paramètres d'une dimension selon une relation « est plus fin que » conformément à leur niveau de détail. Par exemple une ville fait partie d'un département qui fait partie d'une région, ….

4.5. Modélisation multidimensionnelle de l'entrepôt

Dans la plupart des projets de recherche sur les entrepôts de données, le modèle de représentation logique des données est le modèle relationnel. Le modèle conceptuel peut être un modèle en étoile, en flocons ou en constellation qui reflète mieux la représentation multidimensionnelle des données [VILL03].

a- Modèle en étoile :

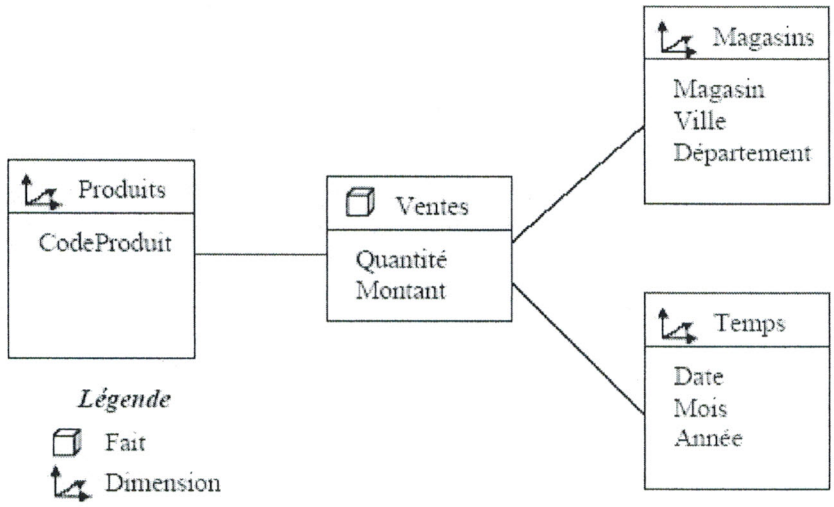


Figure 4.7 : Exemple de modèle en étoile

A partir du fait et des dimensions, il est possible d'établir une structure de données simple qui correspond au besoin de la modélisation multidimensionnelle. Cette structure est constituée du fait central et des dimensions. Cette représentation dite modèle en étoile tire son nom de la position centrale de la table de fait à partir de laquelle partent toutes les références vers les tables de dimension (figure 5.5).

Exemple : le fait Vente stocke la quantité et le montant des produits vendus analysé selon les dimensions : Produits, magasins et temps.

b- Modèle en flocon de neige :

Le modèle en flocon introduit la notion de sous-dimension dans une dimension d'un modèle en étoile. Elle consiste à normaliser les dimensions pour supprimer la redondance. Cette approche a pour but principalement de tenter de gagner de la place pour les dimensions. En effet, ce modèle consiste à décomposer les dimensions du modèle en étoile en sous hiérarchies ; le fait est conservé et les dimensions sont éclatées conformément à leurs hiérarchies des paramètres.

Exemple

Dans cet exemple, nous ajoutons les hiérarchies des paramètres pour les deux dimensions : « magasin & temps »

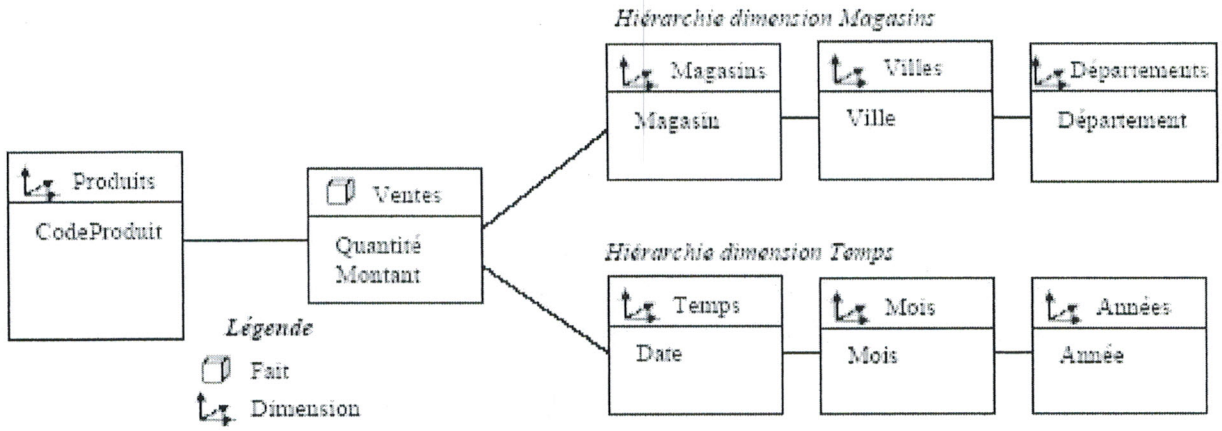


Figure 4.8 : Modèle en flocon de neige

c- Modèle en constellation

Dans ce modèle, plusieurs tables de faits se partagent des tables de dimensions

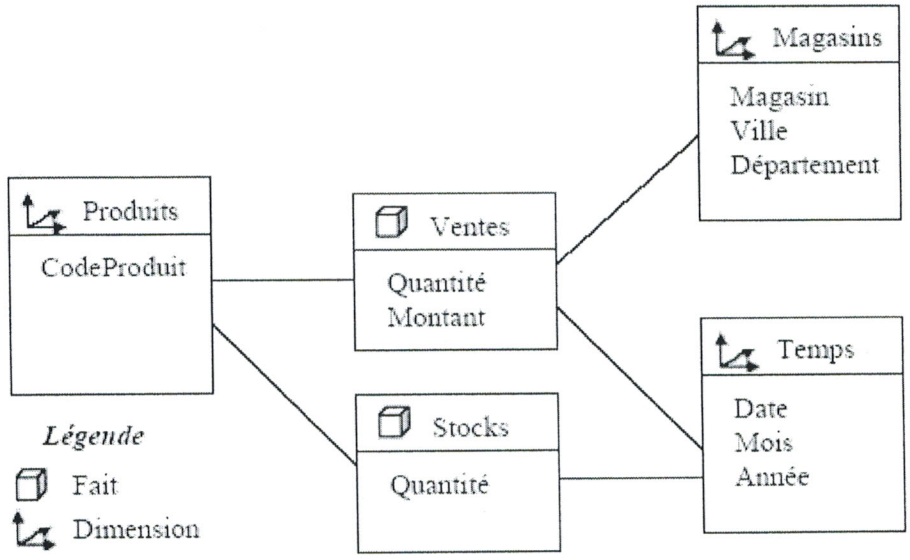


Figure 4.9 : Modèle en constellation

Il est possible d'avoir plusieurs faits pour représenter les situations dans lesquelles les mesures des faits ne sont pas déterminées par le même ensemble de dimensions. Il s'agit de fusionner plusieurs modèles en étoile qui utilisent des dimensions communes. Un tel modèle comprend donc plusieurs faits et des dimensions communes ou non. Le modèle résultant s'appelle constellation de faits.

4.6. Manipulation des données multidimensionnelles

Plusieurs formalismes ont été proposés pour manipuler les données multidimensionnelles : Le cube, la table multidimensionnelle, f-tables, hypercube. Nous allons présenter dans cette section les cubes, et les tables multidimensionnelles [ADIB04] .

1- Le cube

Le constructeur fondamental de la modélisation multidimensionnelle est le cube de données. Un cube organise les données en un ou plusieurs critères (dimensions) qui déterminent une ou plusieurs mesures d'analyse (faits). La représentation sous la forme d'un **cube** est devenue le symbole du décisionnel. Un cube de données représente donc un schéma en étoile comportant trois dimensions (les trois dimensions du cube) et l'intersection dans l'espace de ces axes constitue la mesure analysée. Lorsque le schéma comporte plus de trois dimensions, on devrait dessiner une structure à n dimensions.

Exemple

Soit la modélisation multidimensionnelle suivante concernant une chaîne de magasins. Dans ce cas, la mesure est la Quantité vendue d'un Produit dans un Magasin à un instant de Temps.

On remarque sur ce cube que 100 unités du produit P_1 ont été vendues en 2004 dans le magasin M_3.

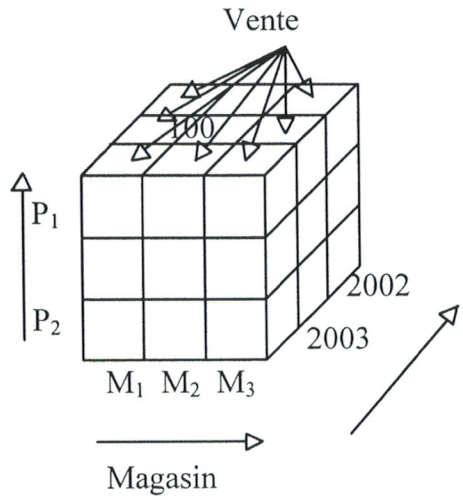


Figure 4.10: Exemple de cube de données

2- La table multidimensionnelle

La table multidimensionnelle est organisée en colonnes, lignes et plans. Les deux premières dimensions sont représentées par les colonnes et les lignes. La troisième colonne est représentée par les plans. L'intersection d'une ligne avec une colonne représente les mesures.

Exemple

On va représenter l'exemple ci-dessus sous forme de tables multidimensionnelle.

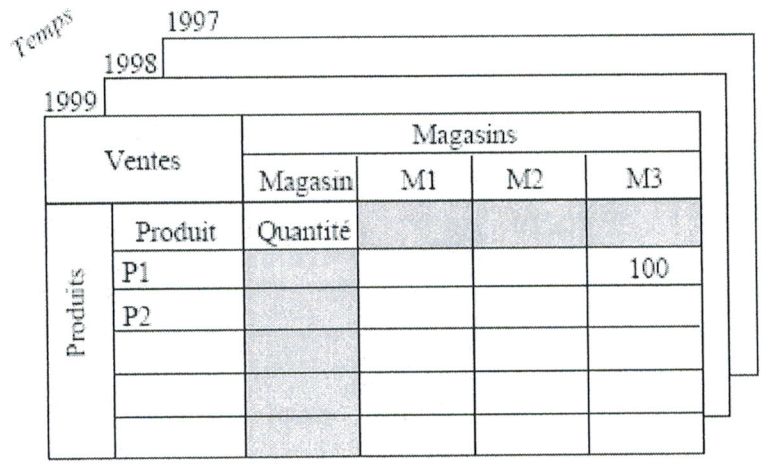


Figure 4.11: exemple d'une table multidimensionnelle

3- Opérateur multidimensionnel

Drill-down & Roll-Up" (zoom avant):

- ***Roll up*** : consolider (résumer) les données, il permet de passer à un niveau supérieur dans la hiérarchie d'une dimension
- ***Drill down*** : l'inverse du Roll-up; il permet de descendre dans la hiérarchie d'une dimension
- **Slice et Dice:** Projection et sélection du modèle relationnel
- ***Pivot (rotate)***: consiste à effectuer une rotation de manière à présenter une face différente. Elle permet ainsi d'avoir accès aux différentes vues des données multidimensionnelles.

4.7. Les vues pour la conception d'entrepôts de données

Le mécanisme de vues est devenue une technique incontournable dans la mise en oeuvre des systèmes d'intégration d'information dans les entrepôts de données

Dans un entrepôt de données, les vues sont utilisées à la fois comme une technique de spécification pour construire l'entrepôt et comme un outil puissant pour modéliser un sous-ensemble de données extrait de l'entrepôt de données et ciblé sur un sujet unique. En effet, une vue est définie par une requête qui produit à la fois le schéma de la vue et les données associées à la vue. Ceci permet l'extraction sélective des données et par le mécanisme de filtrage garantit la confidentialité de certaines données. Ce filtrage de données peut être considéré comme une manière commode de ne montrer à l'utilisateur que ce qui l'intéresse ou/et comme un moyen de garantir une certaine sécurité en cachant certaines données jugées confidentielles [BELL99].

Une des problématiques de recherche concernant l'utilisation des vues dans les entrepôts de données est la sélection de vues à matérialiser. Il s'agit précisément de déterminer l'ensemble de vues à matérialiser en tenant compte d'un certain nombre de paramètres comme les requêtes les plus fréquentes, l'espace de stockage et le coût de maintenance (cf. section 6 de ce chapitre). Le problème à résoudre consiste à optimiser le temps de réponse aux requêtes et l'espace disque.

A ce dernier paramètre, il faut ajouter le coût de maintenance de vues qui dépend de la quantité de données résultant de la matérialisation des vues. Le coût de maintenance correspond au coût de rafraîchissement des données de l'entrepôt pour les maintenir cohérentes suites à des mises à jour opérées sur les données sources. Il en résulte que le coût d'exécution des requêtes est en conflit avec le coût de maintenance des vues car la matérialisation favorise l'optimisation de requêtes mais en contre partie elle entraîne un sur coût de maintenance des données en cas de mise à jour des données sources

4.8. La problématique des entrepôts face aux données complexes

La technologie des entrepôts de données a fait l'objet de nombreuses recherches scientifiques et ont maintenant largement fait leurs preuves dans les domaines de la gestion de données et de l'extraction de connaissances à partir de données dites « simples » (données numériques ou symboliques exprimées dans un tableau de type individus-variables) ***[BOUS03].***

Cependant, les données ne sont pas seulement numériques ou symboliques, mais qu'elles peuvent être représentées dans des formats différents (textes, images, son, vidéos, bases de données,etc.) provenir de sources diverses (données de production, scanners, satellites, enregistrements vidéos, comptes-rendus médicaux, résultats d'analyse, web, etc.), avoir une sémantique différente (langues différentes, échelles différentes, évolution de la définition d'une donnée dans le temps, etc.). De telles données sont désignées par les termes de données complexes.

L'exploration des données complexes dans les entrepôts de données classiques implique de nombreux problèmes, notamment en ce qui concerne leur intégration, structuration, modélisation et leur stockage d'une part et leur analyse d'autre part.

- Le problème de l'intégration des données complexes réside dans le fait que l'intégration basée sur les médiateur ou les entrepôts de données, est valide quand il s'agit de données simples. Mais comment peut-on intégrer des données dont les types sont divers ? Comment peut-on les stocker et les interroger ?
- L'une des difficultés engendrées par la structuration est due à la diversité des formats des données complexes. La description de ces dernières nécessite une certaine précision et un espace de représentation adapté. Le stockage des données complexes est assurés par les bases de données semi-structurées, soit en utilisant le natif XML ou dans des bases de données classiques (relationnelles, orientées objets ou relationnelles-objets), en utilisant les langages de requêtes déjà cités.
- L'intégration des données complexes exige une modélisation permettant de prendre en considération les différents aspects de ces données. Or, il n'existe pas de modèle universel pour les données multimédia, et de manière générale, pour toutes les formes de données complexes. Le recours à un format unifié pour intégrer les diverses sources de données dans une base cible est rendu possible grâce à XML. En effet, ce langage permet non seulement de véhiculer les données, mais aussi de les décrire de façon précise. Ainsi, Il est facile de les gérer, de les mettre à jour ou de les interroger.

4.9. Conclusion

En s'inspirant des méthodes utilisées dans les entrepôts de données classiques, le modèle multidimensionnel, et plus précisément les modèles en étoile, offrent un cadre d'analyse des données. Ils permettent d'observer des faits à travers d'indicateurs (mesures) et d'axes d'analyse (dimensions). L'analyse OLAP va permettre de représenter les données simples dans un espace multidimensionnel (cube de données) dans lequel l'utilisateur peut naviguer et effectuer ainsi une démarche exploratoire.

Par contre, comment peut-on l'appliquer sur des données complexes ? Peut-on affiner encore les magasins (*data marts*) selon le type des données complexes ? Est-il pertinent de construire un cube (*data cube*) dans lequel les données sont issues de différents documents ? Toutes ces interrogations révèlent la nécessité de repenser autrement le processus d'entreposage dans le cas des données complexes.

Face à ces différents problème, nous allons exposer à la suite de ce chapitre, plusieurs travaux existants sur l'intégration des documents à base de XML surtout celles orientés vers les entrepôts de données hétérogènes.

5. CHAPITRE - Etat de l'art sur les entrepôts – travaux existants

Ces dernières années, de nombreux travaux réalisés se sont basées sur XML pour intégrer des données dans des entrepôts de documents. Plusieurs architectures ont été proposées où nous allons présenter les principaux travaux effectués pour la manipulation des documents.

5.1. Projet Xylème [XYLE01]

A l'origine, Xylème créée en septembre 2000 était à l'origine un projet de recherche entre plusieurs équipes « INRIA, U.Mannheim, LRI, laboratoire CEDRIC », afin de développer un produit à partir d'un prototype construit dans Verso. Xylème vise à la création d'un entrepôt extensible capable de stocker toutes les données XML du Web et de fournir une nouvelle génération de technologies de gestion du contenu de données, capable d'exploiter le potentiel du langage XML [SIRO04].

XML offre des solutions d'accès intelligent « recherche précise et intelligente » à des contenus XML agrégés « Volumes importants, multi sources, multi formats XML »

Les principaux thèmes de recherche qui ont été étudiés dans le cadre de Xylème peuvent être résumés comme suit :

- Stockage efficace de gros volumes de données XML ;
- Traitement de requêtes avec indexation de documents XML ;
- Acquisition et maintenance des données XML du Web ;
- Service de notification de mise à jour de données pertinentes ;
- Intégration sémantique des données.

Son objectif :

- Nouvelle génération de moteurs de recherche ;
- Intégration sémantique des données ;
- Requête sur données structurées ;
- Gestion des évolutions ;
- Service d'abonnement et notification ;
- Historique des données ;
- API pour l'écriture d'applications ;
- Hébergement de données et d'applications.

Remarque

XML permet aussi d'enrichir le contenu lui-même et ne pas seulement y travailler au niveau de l'enveloppe « Granularité enrichie de l'information ». Les principaux modules de l'architecture du système Xylème sont illustrés dans la figure ci dessous.

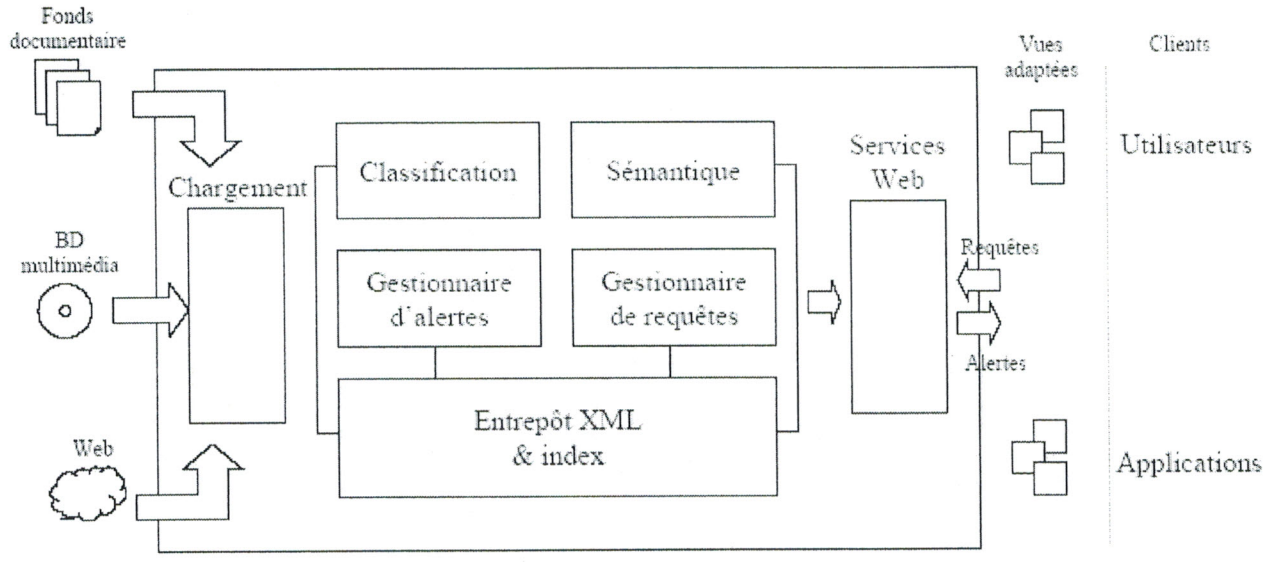


Figure 5.1 : Architecture de Xylème

Les données sont stockées de façon automatique dans un SGBD natif. Ce SGBD indexe à la fois le texte et la structure des documents XML. Il permet aussi de traiter de gros volumes de données. Le module d'indexation est chargé d'indexer le contenu ainsi que la structure des documents XML. Il intègre des fonctions linguistiques « lemmatisation, thesaurus, concepts » qui facilite la recherche dans les documents semi-structurés

Pour interroger l'entrepôt XML, Xylème propose un langage de requête proche de XQUERY. Xylème offre des interfaces de programmation qui permettent de :

- Construire des vues qui intègrent les données du Web par domaine sémantique
- Interroger les données à travers ces vues
- Faire la mise à jour des données

5.2. *Projet e_XMLMédia :* [GARD00]

e-XMLMedia fondée par Gardarin en 1999, propose la suite de composants e-XML pour faire le pont entre les bases de données et XML. Ces composants s'intègrent dans une architecture distribuée, pour publier, échanger, stocker et interroger des documents XML dans un système d'information.

Il est constitué des composants suivants :

a- Le composant XMLizer ;

b- Le composant e-XML Repository ;

c -Le composant e-XML Mediator.

a- Le composant XMLizer

Joue le rôle d'interface XML pour des données stockées dans un SGBD relationnel. Il permet d'extraire, de transformer et d'insérer des données XML dans un SGBD relationnel en définissant des règles de gestion. Le composant e-XMLizer permet de publier une base SQL en XML et inversement

Principe

e-XMLizer permet la transformation des documents XML en tables relationnelles existantes ou nouvelles. En utilisant le langage de requête XQuery, il permet d'interroger les tables relationnelles et de publier des documents XML composé à partir de ces tables.

Architecture

e-XMLizer se compose de deux modules :

- Extractor pour la production de documents XML ;
- Mapper pour l'insertion du contenu de documents XML en base de données relationnelles.

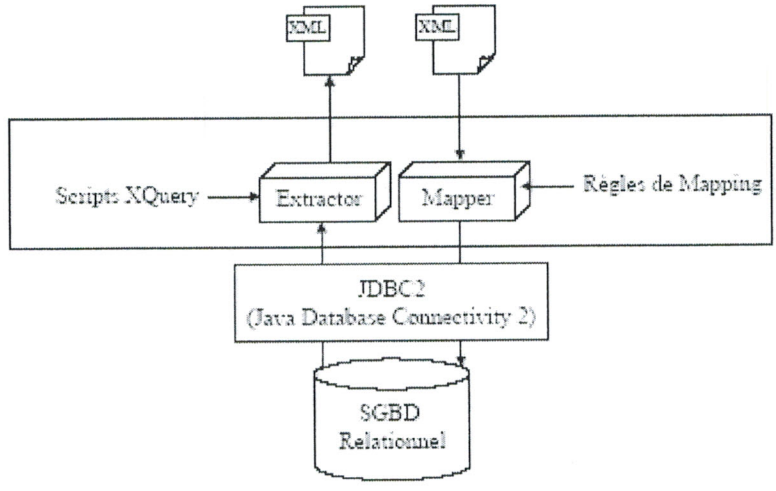


Figure 5.2: Architecture de e-XMLizer

- ***Extractor:*** Pour extraire le contenu et la structure d'un document XML, l'utilisateur fournit des requêtes XQuery pour chaque type de document XML. Extractor fait la connexion à la base de données, exécute les requêtes et reconstruit les documents XML selon le format spécifié.

- ***Mapper :*** L'utilisateur fournit une description des règles de stockage à appliquer au contenu de chaque élément. Ces règles sont spécifiées à l'aide du langage XSL (standard W3C), qui offre une grande flexibilité pour sélectionner des données dans un document XML en se basant sur son contenu et sa structure. Le Mapper utilise ces règles pour décider de l'emplacement de stockage et du traitement à effectuer pour chaque élément XML. Il prend en compte le contrôle des règles de gestion fonctionnelles et techniques avant insertion.

b- *Le composant e-XMLRepository* [BARI03]

Le composant e-XML Repository a été développé par e-XMLMédia, il permet de stocker et d'interroger des documents XML dans un SGBD relationnel. Il étend les fonctionnalités des SGBD relationnels en fournissant un accès dual XML et relationnel aux documents stockés. Il assure un chargement et une restitution rapide des documents stockés dans les tables relationnelles, en conservant et exploitant leur structure pour optimiser leur placement. Le contenu et la structure des documents sont indexés pour permettre l'interrogation via un langage de requêtes XML.

Principe

Ce langage de requêtes offre en particulier la possibilité d'interroger simultanément une collection de documents XML. Pour cela, on utilise des expressions XPATH permettant de naviguer dans la structure et de sélectionner des fragments des documents XML. Il permet aussi un mode de stockage générique utilisé pour stocker tout type de documents XML, y compris ceux dont la structure n'est pas connue a priori, sans configuration préalable. Lorsque la structure des documents stockés est connue, l'utilisateur peut devenir une correspondance entre cette structure (XML schéma) et un modèle relationnel.

Architecture de e-XMLRepository [GARD00]

Sur la figure ci-dessous, on présente les modules les plus intéressant du composant e-XMLRepository

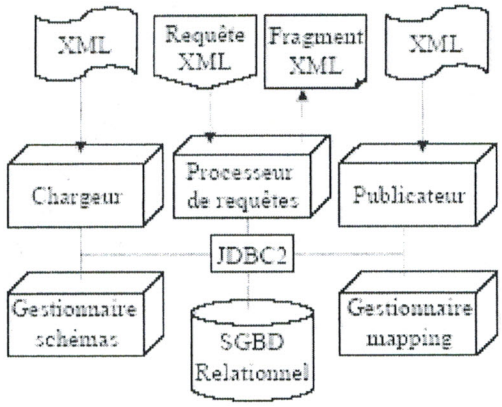


Figure 5.3: Architecture de e-XMLRepository

- ❖ Module gestion de schéma qui assure la manipulation des schémas XML et leur analyse ;
- ❖ Module de gestion de mapping qui assure les manipulations et l'analyse entre schéma XML et schéma relationnel ;
- ❖ Module de chargement qui analyse des documents XML, les valide, et charge les documents en tables suivant le mapping retenu ;
- ❖ Module de reconstruction qui permet de publier tout document XML stocké à partir de son identifiant.

Le processeur de requête qui transforme les requêtes XQuery en requête SQL.

c- Le composant e-XMLMediator

Le composant e-XMLMediator a été développé par e-XMLMédia. C'est un outil de requêtes sur des sources de données hétérogènes. La localisation des données dans les sources est transparente, grâce à l'ajout de métadonnées. En fait, ces métadonnées constituent des vues locales qui définissent les sources pour les intégrer dans un schéma selon une approche LAV.

Le composant e-XML Mediator se connecte aux sources de données via des wrappers (adaptateurs) qui assurent :

- La traduction des données du format natif de chaque source vers XML ;
- La traduction de la requête XML vers le langage de requête natif de la source.

Principe

Ce composant va permettre de :

- Fédérer de sources de données hétérogènes ;
- Donner une vue unique d'un ensemble de sources ;
- Gérer une Grande variété de données accessibles via des wrappers ;
- Réaliser une interrogation multi-base efficace ;
- Manipuler un langage de requêtes basé sur XQuery, retournant du XML
- Gérer des requêtes distribuées performantes grâce à l'utilisation d'un cache de métadonnées et d'une délégation maximale ;
- Utiliser des outils graphiques pour l'administration et les requêtes.

Architecture de e-XMLMediator

e-XMLMediator est composé de plusieurs modules décrits sur la figure ci-dessous.

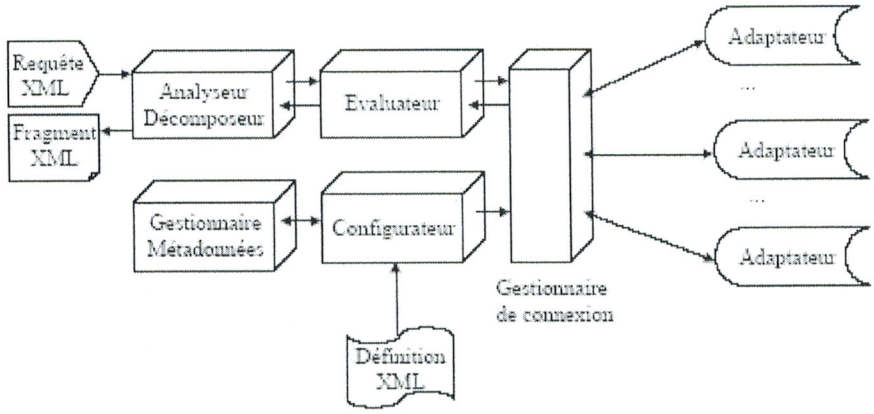


Figure 5.4 : Architecture de e-XMLMediator

Console d'administration : qui permet de configurer le médiateur, enregistrer de nouveaux adaptateurs, et exécuter les requêtes.

Gestionnaire de métadonnées : qui permet de garder les données descriptives de chaque source sous forme de schémas XML Analyseur/décomposeur

Evaluateur : rassemble les résultats et faire l'évaluation

Gestionnaire de connexion : rassure les connexions avec les adaptateurs, la transmission des sous-requêtes, et la réception des résultats.

5.3. Projet Karina [RANW00]

Karina est un projet commun du ministère de l'industrie et du Pôle TIIM86 (Technologies de l'information, Informatique et Multimédia) de Montpellier. Son objectif est la création de cursus pédagogiques personnalisés ou de documents culturels à partir d'éléments provenant de diverses sources (journaux électroniques par exemple). Karina est basé sur quatre points fondamentaux : [RANW00]

- L'utilisation du langage XML pour qualifier des documents ;

- La description sémantique du contenu de ces documents en utilisant un vocabulaire défini dans une ontologie du domaine dont ils traitent ;

- La description globale d'un document mais également sa description par fragments ou bien sa description locale ;

- L'utilisation des descripteurs pour bâtir des parcours guidés par un moteur d'évocation conceptuelle.

Les Briques de documents traitées dans ce projet sont des documents HTML homogènes, annotées à l'aide d'un outil de qualification et une DTD écrite en XML. Leur qualification est stockée dans une base de données relationnelle (Oracle). Le moteur est développé en langage Java. Les requêtes effectuées par le moteur sont écrites en SQL [CHRI02].

5.4. Projet Lore:«Lightweight Object REpository » [AMAN03], [RANW00]

Lore est un système développé par l'université de Stanford , dans le cadre du projet TSIMMIS. Lore assure le stockage et l'interrogation des documents semi-structurés au format OEM « Object Exchange Model ».

LOREL « Ligfhtweight Object Repository Language » est un langage de requêtes qui facilite la recherche des documents ayant des structures irrégulières ou même inconnues.

Architecture

Lore est composé des modules suivants :

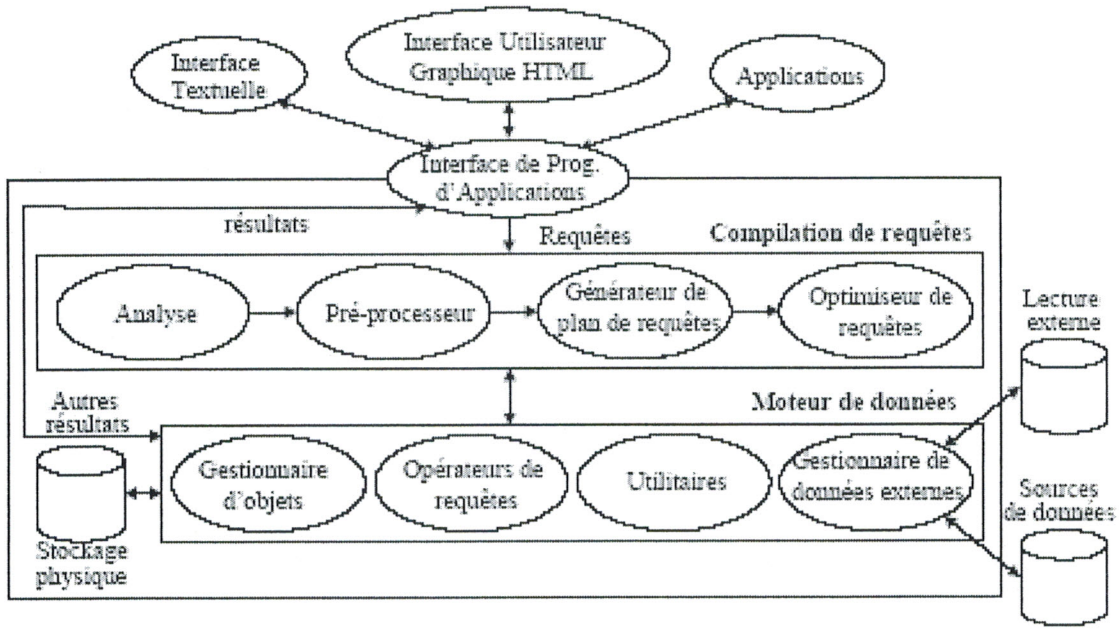


Figure 5.5: architecture de Lore

Les principaux composants

- *Le pré-processeur* : transforme les requêtes en OQL « qui sont faciles à utiliser » ;
- *Générateur de plan de requêtes* : qui décide sur l'indexation du document ;
- *Gestionnaire d'objet OEM ;*
- *Opérateur de requête* pour l'exécution des requêtes ;
- Le gestionnaire de données externes qui récupère des données de sources externes.

5.5. Projet Strudel

Strudel a été développé dans le but d'utiliser les concepts et les techniques des SGBD pour la création des sites Web. Son modèle de données est similaire à celui du OEM. Son objectif est :

- la création de vues intégrées pour construire des sites Web ;
- la spécification de la structure en utilisant des requêtes STRUdel Query.

Son architecture est la suivante :

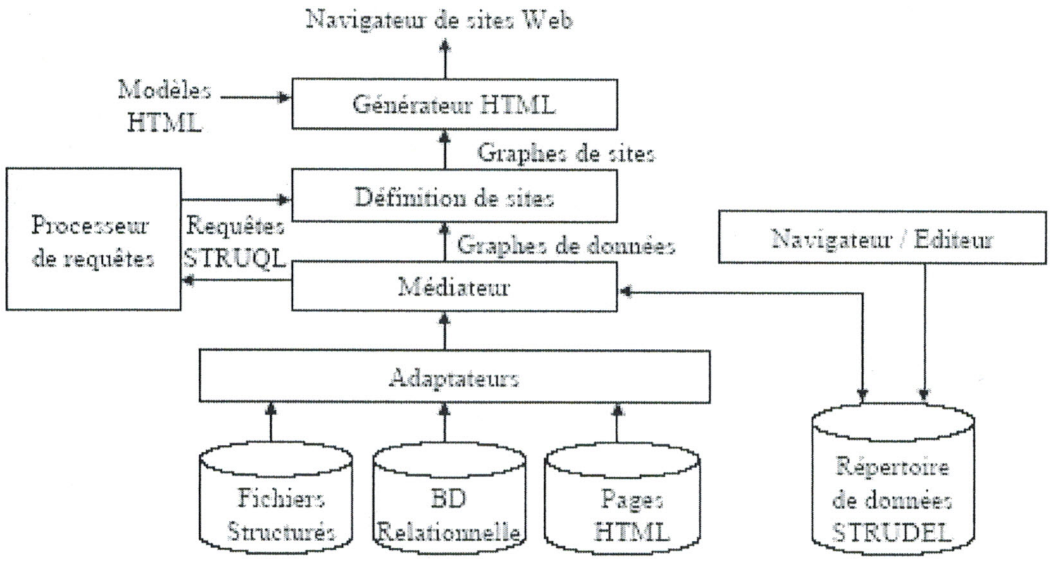


Figure 5.6: Architecture de Strudel

L'architecture de STRUDEL représentée sur la figure ci-dessus, est composée de :

- Répertoire « *Entrepôt* » *de données STRUDEL* : le graphe de données d'un site Web est stocké dans l'entrepôt de données STRUDEL. Les données sont obtenues des adaptateurs qui convertissent les données des sources externes en des données au format interne semi-structuré utilisé par STRUDEL ;
- *Navigateur/éditeur de graphe* : il permet à l'utilisateur de créer, mettre à jour et visualiser les graphes pouvant être utilisés pour le graphe de données et le graphe du site ;
- *médiateur* : il fournit une vue uniforme des données sous-jacentes. Plutôt que d'interroger les sources externes à la demande au moment de l'exécution de la requête, son approche est intégrer les données en stockant préalablement les données des sources externes dans l'entrepôt de données STRUDEL ;

- *Processeur de requêtes* : STRUDEL définit le langage STRUQL (STRUdel Query Language) pour réaliser des requêtes et restructurer des données semi-structurées.
- L'interpréteur de requête de STRUDEL utilise les opérateurs physiques traditionnels (jointure, restriction, sélection) ainsi que des opérateurs nécessaires pour l'interrogation de schéma (ex. trouver tous les noms d'attributs dans un graphe) ;
- *Générateur HTML* : pour produire la représentation graphique de chaque page du site Web, un modèle HTML est associé à chaque noeud du graphe du site. Le résultat est un site Web navigable.

5.6. Wind : « Warehouse for Internet Data »

Est un outil qui permet de construire un entrepôt de données à partir d'informations issues du Web « monde cinématographique » pour traiter des documents hétérogènes.

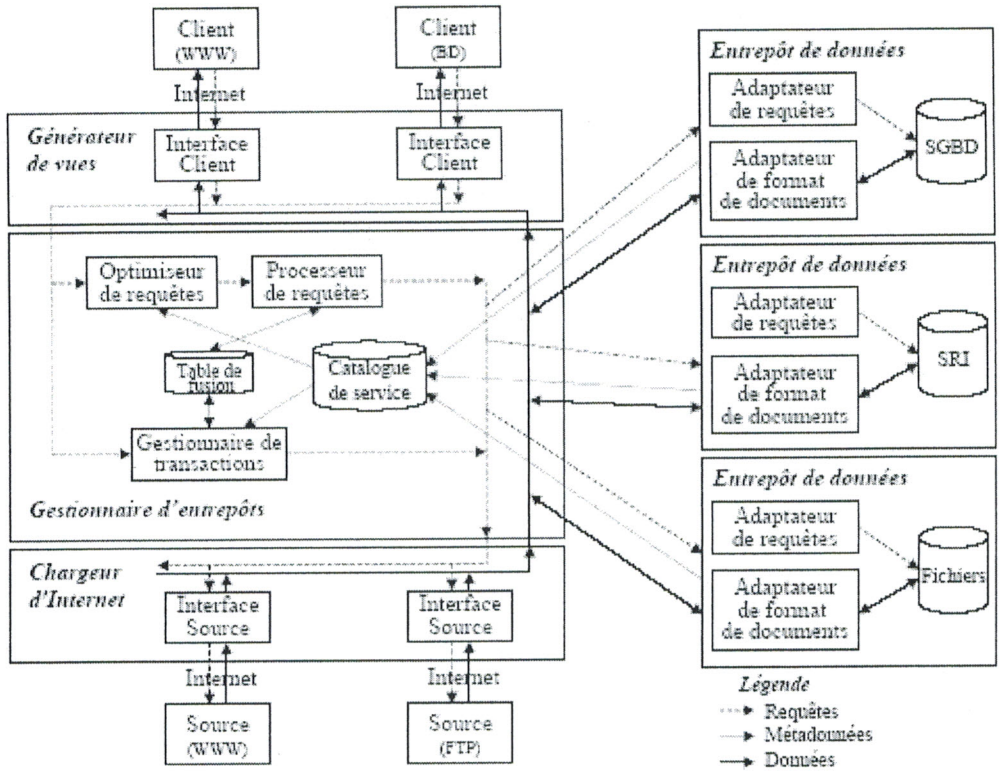


Figure 5.7: Architecture de Wind

Wind collecte un ensemble de documents hétérogènes issus de sources différentes dans des entrepôts de données. Une instance Wind est guidée par le serveur Wind, qui interagit avec chaque entrepôt de données par l'intermédiaire d'un adaptateur. Le serveur Wind se compose des composants suivants :

- Un chargeur d'Internet permettant de récupérer les informations à partir de sources d'informations externes ;
- Un gestionnaire d'entrepôts permettant de traiter les transactions et les requêtes au niveau du serveur et de transmettre les sous-transactions et les sous-requêtes ;
- Un générateur de vues permettant l'interaction avec les utilisateurs.

Chaque entrepôt dispose d'adaptateurs spécifiques :

- Un adaptateur de requêtes qui transforme la requête de l'utilisateur en une requête adaptée à l'entrepôt ;
- Un adaptateur de format de documents qui permet de restituer le résultat dans un format commun quel que soit l'entrepôt interrogé.

5.7. Projet DOCWARE [KHRO05]

Un schéma générique d'entrepôts de documents a été proposé dont le principal intérêt est de permettre le stockage et l'intégration de tout type de documents.

Objectif

Le projet DOCWARE va permettre de :

- Rechercher des documents par leur contenu sémantique ;
- Interroger l'entrepôt en mêlant à la fois les aspects structure et contenu ;
- Réaliser des analystes en se basant sur la structure « analyses multidimensionnelles ».

Les entrepôts de documents qui ont été proposé par Khrouf permettent le stockage de documents hétérogènes, sélectionnés et filtrés, ainsi que leur classification selon des structures logiques génériques. Une telle organisation des entrepôts permet de faciliter l'exploitation des informations documentaires intégrées. Elle est réalisée selon plusieurs techniques :

1. La recherche d'information consiste à restituer des granules de documents en réponse à une requête utilisateur formulée par des mots clefs ;
2. L'interrogation des données en utilisant un langage déclaratif ;
3. L'analyse multidimensionnelle consiste à manipuler les informations contenues dans l'entrepôt.

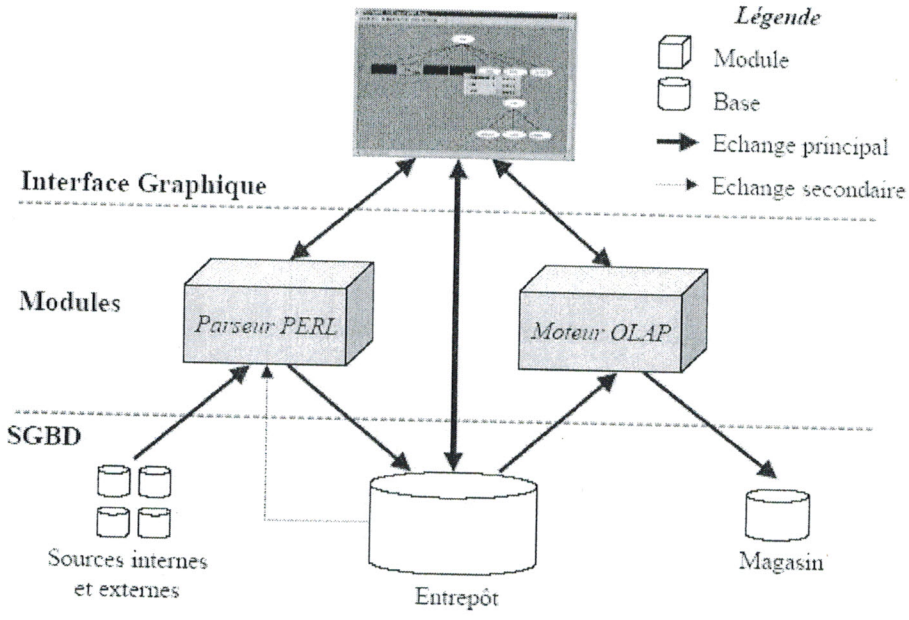


Figure 5.8 : Architecture du DOCWARE

DOCWARE se base sur une architecture modulaire. Il se compose de quatre composants :

1- Interface graphique : il s'agit d'une interface JAVA permettant de dialoguer entre les différents modules du DOCWARE ;

2- Module d'intégration : il s'agit d'un parseur PERL permettant d'intégrer des documents jugés pertinents dans l'entrepôt ;

3- Module d'analyse : il s'agit d'un moteur MOLAP permettant de représenter des extraits de l'entrepôt « magasins de documents » sous formes de tables multidimentionnelles ;

4- Stockage des données : il s'agit du SGBD Oracle8 permettant de stocker les documents intégrés dans l'entrepôt.

5.8. Projet ARIADNE [RANW00]

ARIADNE est un environnement pour la mise en commun de systèmes d'enseignement assisté par ordinateur. Les outils qui le composent peuvent être utilisés dans le cadre de formation classique ou d'enseignement continu. Ce projet a pour objectif le partage et la réutilisation des ressources pédagogiques, il constitue un environnement fermé pour ses membres, ce qui permet un suivi éditorial strict et assure une haute qualité des travaux.

Dans ce but, une base de données distribuées de documents pédagogiques réutilisables a été développée : c'est le *Knowledge Pool System « KPS »*. À notre connaissance, chaque document pédagogique « cours, QCM, exercice, vidéo clips » a été constitué dans un but particulier et est difficilement réutilisable pour un autre objectif [RANW00]

Les documents pédagogiques provenant d'environnements auteur quelconques ou d'outils d'ARIADNE sont indexés et validés et le KPS crée des documents pédagogiques pour les étudiants, ou bien permet une exploitation de ces documents par les auteurs et les pédagogues.

Objectif

ARIADNE a deux objectifs principaux : [BOUR01]

1. La production et la maintenance de matériaux pédagogiques par des spécialistes, basées sur des scénarios sociopédagogiques ;
2. L'utilisation de méthodes, outils et techniques permettant un accès et une utilisation conviviale des cursus présentés. [DELE00]

ARIADNE est caractérisé par trois types d'outils

1. Création des items didactiques : « Authoring reuse » : Le projet ARIADNE a essayé de travailler avec IMS pour définir un ensemble commun de métadonnées. Il en a résulté une norme, nommé LOM « Learning Object Metadata », qui permet d'indexer un document pédagogique grâce à des normes citées dans le chapitre précédant ;
2. Outils de caractérisation des items didactiques ;
3. Outils d'administration et de création de cours.

5.9. Approche vers l'entreposage des données complexes [BOUS03]

Ce projet permet de concevoir des processus décisionnels utilisant des données complexes. Deux grands axes se dégagent : (1) la structuration des données complexes avec la modélisation et l'intégration des données dans une base de données ; (2) l'analyse des données complexes par des techniques de fouille de données, d'analyse en ligne ou par les deux approches combinées.

Vu la complexité du projet, il utilise un système multi agents. Cette complexité est ainsi prise en charge par un travail collaboratif développé par les différents agents du système. Le processus d'entreposage et d'analyse des données complexes est représenté par l'architecture suivante :

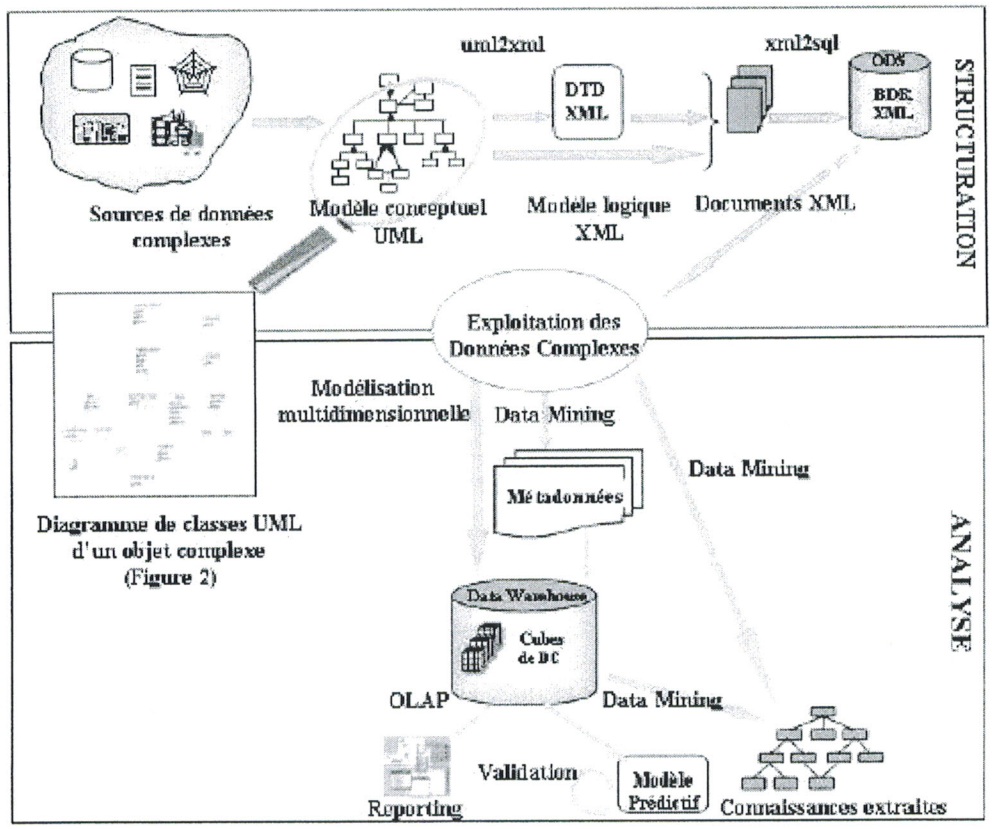


Figure 5.9 : Le processus d'entreposage et d'analyse des données complexes

1- Intégration des données complexes

L'idée de cette approche consiste à modéliser et à intégrer les données complexes dans une source de données structurée comme une base de données relationnelle décisionnelle jouant le rôle d'un entrepôt de données. Le langage XML est choisi en tant que formalisme pivot des volets logique et physique pour le processus de modélisation. Ce dernier est composé de trois phases (modèle conceptuel, logique, puis physique).

Cette approche considère une donnée complexe comme un objet complexe pouvant être décrit par un modèle conceptuel UML. Un objet complexe peut être composé d'un ou de plusieurs sous-documents pouvant avoir chacun une langue et des mots-clefs associés.

Chaque sous-document a un type identifié qui peut être un texte libre ou balisé avec des liens, une vue relation, une image, un son ou une vidéo. Un objet complexe peut permettre de décrire, par exemple, une page Web constituée de différentes portions de textes, d'images et de données issues de bases de données.

Le diagramme de classes UML est traduit en une définition de schéma XML représentant le modèle logique de la donnée complexe. Cette phase de structuration est finalisée par la construction d'un modèle physique représentant des documents XML stockés dans une base de données relationnelle. Ainsi, nous représentons toutes sortes de donnée complexe dans un format unifié par un document XML.

Le modèle conceptuel UML est générique. Il comporte les descripteurs de bas niveau des données complexes (comme la taille, la résolution d'une image ou le nombre de lignes ou de mots d'un texte). Lorsque ce modèle est utilisé dans différentes applications, il est nécessaire de le compléter par des descripteurs sémantiques propres au domaine d'application. L'ensemble des descripteurs de bas niveau et sémantiques va ainsi permettre de construire un modèle de données spécifique.

2- Modélisation multidimensionnelle des données complexes

La modélisation des données complexes permet de mieux préparer les informations à l'analyse. Pour cela, cette approche va créer un référentiel de données qui va réunir l'ensemble des données définies dans les modèles élaborés lors de la phase d'intégration des données complexes. Il liste de façon exhaustive les données nécessaires dans le modèle multidimensionnel, décrit les caractéristiques de chacune des données en précisant son rôle dans le modèle à créer, aide dans le choix des éléments du modèle (descripteurs, indicateurs) conforme aux objectifs d'analyse et enfin vérifie la cohérence des données participant au modèle créé. Il est complété par des informations sur les données indiquant leur origine, leur nature et le rôle qu'elles peuvent jouer.

Les données complexes sont décrites par des attributs de bas niveau et possèdent également des descripteurs sémantiques. Ces derniers peuvent être obtenus par diverses techniques de fouille de données, de statistique, de traitement d'images ou du signal. Une exploration par une technique de fouille des données peut, par exemple, contribuer à l'identification des faits à analyser et peut permettre d'enrichir le référentiel par de nouveaux descripteurs sémantiques.

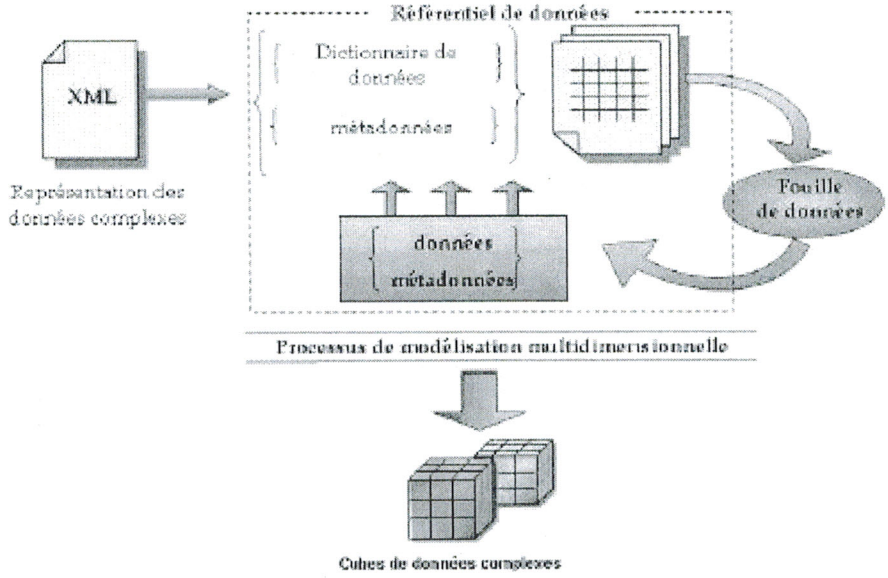


figure 6.10: modélisation multidimensionnelle des données complexes

En fonction des objectifs définis par l'utilisateur, les faits à observer sont identifiés et exprimés à l'aide d'indicateurs (mesures) et d'axes d'analyse (dimensions). Le cube de données construit correspond à une vue des données complexes et représente ainsi un espace d'analyse sur lequel il est possible d'effectuer une analyse en ligne (OLAP) ou d'appliquer des techniques de fouilles de données. La construction des cubes de données complexes se fait à la volée en fonction des besoins d'analyse de l'utilisateur. L'information ou la connaissance extraite lors des ces études est capitalisée sous forme de métadonnées et archivée dans le référentiel de données d'aide à la modélisation multidimensionnelle.

La complexité des données nécessite des opérateurs spécifiques d'agrégation, de navigation ou même d'extraction de connaissance. Dans ce projet, de nouveaux opérateurs d'analyse en ligne spécifiques pour les données complexes ont été construits. Ils sont basés sur la CAH (Classification Ascendante Hiérarchique). Grâce à une API (*Application Programming Interface*), l'utilisateur peut visualiser différents regroupements possibles. Il décide alors du nombre de classes (agrégats) qu'il souhaite en se basant sur un indicateur de qualité des classes que nous avons conçu.

Ainsi, cette approche a permis de décrire un processus complet d'entreposage de données complexes pour l'aide à la décision et leur exploration par des techniques d'analyse en ligne ou de fouille de données.

Le processus d'intégration proposé, permet de modéliser les données complexes dans un format unifié en XML et de les stocker dans une base de données relationnelle. A des fins d'analyse, de nouveaux opérateurs OLAP ont été crées pour faciliter la modélisation multidimensionnelle.

Bilan

Une synthèse des différents travaux cités dans ce chapitre, est résumée dans le tableau « Tableau 5.1 »

A partir des différents travaux étudiés ci-dessus, on remarque les particularités suivantes :

N°	Projet	Types de données	Type d'intégration	Caractéristiques
1	Xylème	Données hétérogènes	Entrepôt de données « matérialisée »	Stockage à base de XML natif Qualification par les ontologies Architecture distribuée. Intégration sémantique de données à travers des vues.
2	e-XMLMédia	Données hétérogènes	Médiateur	Source de données hétérogènes Stockage dans un SGBD relationnel Qualification par les métadonnées Interrogation par le XQuery
3	Karina	Documents HTML homogènes	Médiateur	Stockage au format OEM Qualification par les ontologies Requêtes par SQL
4	Lore	Données hétérogènes	Médiateur	Stockage et interrogation au format OEM
5	Strudel	Données hétérogènes	Médiateur	Analyse à base de notion de vue Stockage dans un SGBD relationnel
6	Wind	Données hétérogènes	Entrepôt de données	Analyse à base de notion de vue
7	DOCWARE	Documents textuels	Entrepôt de données	Recherche par mots clés Stockage dans un SGBD relationnel Utilise l'analyse multidimensionnelle
9	Entreposage des DC	Données complexes	Entrepôt de données	Utilise un système multiagents Stockage dans un SGBD relationnel Utilise de nouveaux opérateurs OLAP

Tableau 5.1 : différents projets et leurs caractéristiques

Ces projets ont tous en commun le stockage des documents dans leurs entrepôts ainsi que leur analyse. Par conséquent, ils se distinguent par la nature des documents à stocker « hétérogène, homogène, complexe, textuel, etc. », le type de qualification « métadonnées/ontologies», ainsi que les différents types d'analyse « OLAP, fouille de données, ...etc ; et il sera difficile de trouver un modèle universel qui pourra stocker tous les documents complexes de formes différents dans un seul entrepôt.

Dans le domaine du E-learning, on peut considérer plusieurs universités, où chacune d'elle possèdera sa propre bibliothèque numérique « cours, exercices, thèses, rapports, etc. ». Il sera intéressant de réaliser un entrepôt de documents pédagogiques qui pourra stocker l'ensemble des bibliothèques numériques de toutes les universités afin de les regrouper et de permettre l'échange des cours, mémoires, etc, ce qui n'est pas possible vu la complexité des documents

Dans ce contexte, nous avons proposé une approche qui permettra de construire non pas un entrepôt de données mais un entrepôt de métadonnées pédagogique «E.M.P » . Cet entrepôt ne va pas contenir les documents, ces derniers seront stockés dans des entrepôts locaux associés à chaque université. Notre entrepôt sera représenté par des catalogues « descripteurs » obtenus après la qualification des documents pédagogiques de toutes les universités par les métadonnées de « LOM ». Les documents pédagogiques peuvent être sous n'importe quel format « PDF ; WAV, JPEG, bases de données, etc. »

Le stockage au niveau de l'entrepôt de métadonnées se fera à base de XML contrairement aux autres travaux des entrepôts de données, où leurs documents sont stockés dans des SGBD. En effet, le langage XML permet non seulement de véhiculer les documents pédagogiques, mais aussi de les décrire de façon précise.

Ainsi, grâce à cet EMP, chaque université pourra :

- Présenter le contenu de sa bibliothèque pédagogique ;
- Assurer la maintenance de son propre entrepôt local ;
- réaliser des mises à jour de ces documents ;
- Permettre l'échange entre les universités et créer une certaine concurrence ;

Au cours de la recherche, l'utilisateur n'aura le droit que de visualiser le catalogue du document recherché, son téléchargement ne sera fait qu'avec l'accord de l'université associée « payement, code, etc ». De cette façon, la sécurité du document sera assurée, et l'université pourra évaluer la pertinence de ce contenu.

Nous allons décrire de façon détaillée notre approche dans la partie III.

PARTIE III – Construction d'un entrepôt de métadonnées de « LOM »

Partie III – Construction d'un entrepôt de métadonnées de « LOM »

Dans cette partie, on exposera le système SABRA sur lequel s'appuie notre travail ; et par la suite on présentera l'architecture principale de notre approche pour la construction d'un entrepôt de métadonnées de « LOM » en impliquant et en identifiant les différents acteurs intervenant dans notre système et en détaillant chaque module composant l'architecture.

Cette partie sera illustrée par un prototype décrivant le système.

6. CHAPITRE - Le projet de recherche SABRA [CHIK04]

6.1. Introduction

Notre travail rentre dans le cadre de la continuité des travaux de recherche sur la réutilisation en ingénierie des documents en général et en e-learning en particulier, et vise à améliorer la qualité des documents produits et à faciliter la recherche.

Dans ce contexte, le système SABRA[1] est une solution d'aide à la rédaction adoptant l'approche de réutilisation ARBRE[2], en définissant les différents concepts relatifs à la méthode. Ainsi, elle sera notre support technique et méthodologique pour construire un entrepôt de métadonnées de « LOM ».

Le système SABRA est organisé en trois modules principaux :

1- Le module « Authoring for reuse » : il consiste à construire un entrepôt de briques de documents. Une brique est représentée par son contenu et par sa qualification par les métadonnées ;

2- Le module « Authoring by reuse » : il consiste à créer un nouveau document (outline) par assemblage de briques de documents contenues dans l'entrepôt.

3- Le troisième module est réservé au traitement de base commun aux deux autres modules « système d'indexation, de stockage, adaptation, qualification »

Notre approche s'intéresse essentiellement par le module « Authoring for reuse » utilisant le modèle ASARD qui représente la qualification et la spécification des documents. Afin de participer à la construction de l'entrepôt du système SABRA, notre système est chargé de :

- Capitaliser les différents documents pédagogiques issus de différentes sources de données existantes « Internet, Intranet, CD, …… » dans un entrepôts ;
- Qualifier, représenter et organiser ces documents ;
- L'exploitation des documents.

Cette qualification des documents se fait en se basant sur la qualification par les métadonnées de LOM, en utilisant le modèle ASARD.

Dans ce chapitre nous présentons le projet SABRA réalisé en adoptant l'approche de réutilisation ainsi que la méthodologie ARBRE en définissant son vocabulaire et ses différents concepts.

[1] System of Authoring By Reuse based on Annotations
[2] Approche d'aide à la Rédaction Basée sur la REutilisation

6.2. La méthodologie ARBRE

La méthode ARBRE « Approche d'aide à la Rédaction Basée sur la REutilisation » vise à faciliter la production de documents, par une réutilisation de documents ou fragments de documents annotés par le modèle ASARD, et organisés et représentés par le modèle MCD.

6.2.1. Les éléments utilisés

6.2.1.1. Les entités fondamentales

a. Le p_document

Le p_document désigne tout objet documentaire produit, ou à produire, durant l'activité de production. La construction d'un p_document, de type « document structuré » n'est pas faites toujours entièrement par réutilisation, certaines parties peuvent être développées par ex'nihilo.

b- Les briques de document

Les briques sont des objets documentaires autonomes pouvant être réutilisées dans la construction d'un p_document. Nous distinguons :

1- les fragments de contenu structuré : « fragments XML »
2- les fragments de contenu semi structuré : « fragments HTML »
3- les fragments de contenu non structuré : « fichiers PDF, PS, DOC, ... »

c- Le o_document

Le document outline (o_document) est une structure logique spécifique au sens XML, qui permet de générer automatiquement une première version du p_document que l'auteur pourra ensuite raffiner et compléter.

d- La qualification des documents : « q_annotations »

Tous les documents qui seront stockés dans l'entrepôt doivent être qualifiées de manière homogène. La qualification est essentielle pour la classification et la recherche des documents.

La qualification des documents dans la méthodologie d'ARBRE est faite avec le modèle ASARD. Ce dernier est basé sur les annotations de qualification q_annotations et offre quatre formes distinctes de qualification, par les métadonnées, par les concepts d'une ontologie, par les associations et par les commentaires. Dans notre système on s'intéresse à la qualification par les métadonnées.

6.2.1.2. Organisation des données

a- le q_composant :

Le q_composant est défini comme une entité organisationnelle bâtie autour d'une brique et servant d'interface à sa qualification, sa réutilisation et sa gestion. Il se présente sous la forme d'un triplet formé de trois types de connaissances : la connaissance réutilisable, la connaissance de réutilisation et la connaissance de gestion.

- *La connaissance réutilisable :* elle constitue la partie intelligible du q_composant. Contient une réréfence à une brique contenue dans l'entrepôt de briques « b_entrepôt ». cette brique est appelée brique pivot
- *La connaissance de réutilisation :* elle se présente sous forme d'un ensemble de q_annotations du modèle ASARD relatives à la brique pivot : META-DESCRIBE, REFERENCE, ASSOCIATE et COMMENT, permettant de la qualifier.
- *La connaissance de gestion :* elle contient des métriques et des données relatives au suivi des q_composant.

b- Le s_composant :

Le s_composant est défini comme l'entité organisationnelle bâtie autour d'une brique et servant d'interface à sa spécification, sa construction et sa gestion. Il se présente sous la forme d'un triplet formé de trois types de connaissances : le sujet de la spécification, la connaissance de spécification et la connaissance de gestion.

- **Le sujet de la spécification**

Le sujet de la spécification correspond à un o_élément donné de type feuille que l'auteur veut développer par réutilisation ou dans le cas échéant par une approche ex nihilo.

- **La connaissance de réutilisation**

La connaissance de réutilisation constitue la partie intelligible du s_composant. Elle se présente sous la forme d'un ensemble de s_annotations du modèle ASARD relatives à la brique pivot : META-DESCRIBE, REFERENCE, ASSOCIATE et COMMENT permettant de spécifier les besoins en briques pour l'élément du o_document, sujet de la spécification du s_composant.

- **La connaissance de gestion**

La connaissance de gestion contient des métriques et des données relatives au suivi des s_composants.

6.2.2. La qualification des briques dans le modèle ASARD

Une annotation dans le modèle ASARD est un prédicat à quatre arguments. Le nom du prédicat, qui va remplacer le nom générique ANNOTATE, exprime le nom de l'action d'annotation. Les arguments dépendent de l'unité d'annotation.

ANNOTATE	(ID ,	Subject,	Object	, A_Text)

a. le premier argument **« ID »** indique l'identifiant de l'annotation

b. **« Subject »** indique le sujet de l'annotation. Dans le cas où ASARD est utilisé comme un langage de qualification, il donne la liste des briques existantes.

c. **« Object »** désigne l'objet utilisé comme moyen de l'annotation

d. **« A_text »** désigne un texte libre, portant sur l'annotation elle-même, et qui peut contenir selon les cas des indications, des orientations, des recommandations, des commentaires ou bien des messages

Une analyse des besoins dans le domaine de la réutilisation, a permis d'identifier quatre catégories d'annotation :

META-DESCRIBE	(ID ,	bricks,	{(Meta-data, Schema,{Value})}	, Q_Text)
REFERENCE	(ID ,	Bricks,	{(Concept, Ontology)}	, Q_Text)
ASSOCIATE	(ID ,	Bricks,	{(Brick | P_document), type)} ,	Q_Text)
COMMENT	(ID ,	Bricks,	Text	, Q_Text)

Figure 6.1. Catégories d'annotations du modèle ASARD utilisées pour la qualification

Ces quatre catégories d'annotations correspondent, dans le cas où ASARD est utilisé comme un langage de qualification, respectivement à quatre formes différentes de qualification des briques: par les méta-données ; par les ontologies ; par les associations ; et par le commentaire

Dans notre cas, les documents pédagogiques sont qualifiés par les métadonnées.

La qualification par les méta-données permet d'améliorer les possibilités de recherche traditionnelle, basée souvent sur une « indexation plein-texte en aveugle ». Elle suppose qu'au préalable aient été définies des méta-données descriptives.

META-DESCRIBE (ID , bricks, {(Meta-data, Schema,{Value})} , Q_Text)

Dans le modèle ASARD, la ressource correspond au second argument (Bricks) de la catégorie META-DESCRIBE Le troisième argument est composé d'un ensemble de métadonnées appartenant à plusieurs schémas (le sous argument Schéma) de description et pouvant prendre plusieurs valeurs (Value).

6.2.3. Le Modèle général de composant de document « MCD »

La modélisation de la dimension statique de la méthodologie ARBRE est le MCD, représenté ci-dessous en UML. Ce modèle est divisé en quatre îlots principaux :

1- L'îlot de production

Cet îlot contient d'une part la classe « p_document » et la classe « document structuré» qu'elle généralise et d'autre part la classe « o_document ».

2- L'îlot de la connaissance réutilisable

La connaissance réutilisable est constituée par l'ensemble de briques représenté par la classe principale « brique » et la classe « fragment de contenu » qu'elle généralise. Cette dernière classe est à son tour spécialisée par les trois classes « frag-contenu structuré », « frag-contenu semi-structuré » et « frag-contenu non structuré » .

3- L'îlot de la connaissance de qualification et de spécification

La connaissance de qualification ou de spécification est représentée par un ensemble d'annotations d'aide à la réutilisation permettant quatre formes d'annotation : par les méta-données ; par les ontologies, par les associations ; et par le commentaire. Chacune de ces formes est représentée par une classe qui est reliée par un lien de généralisation à la classe principale « annotation ».

4- L'îlot d'organisation

Cet îlot contient essentiellement les deux classes principales ou pivots du modèle MCD (Modèle de Composant de Document) : « q_composant » et « s_composant ».

Le modèle MCD (Modèle de Composant de Document)

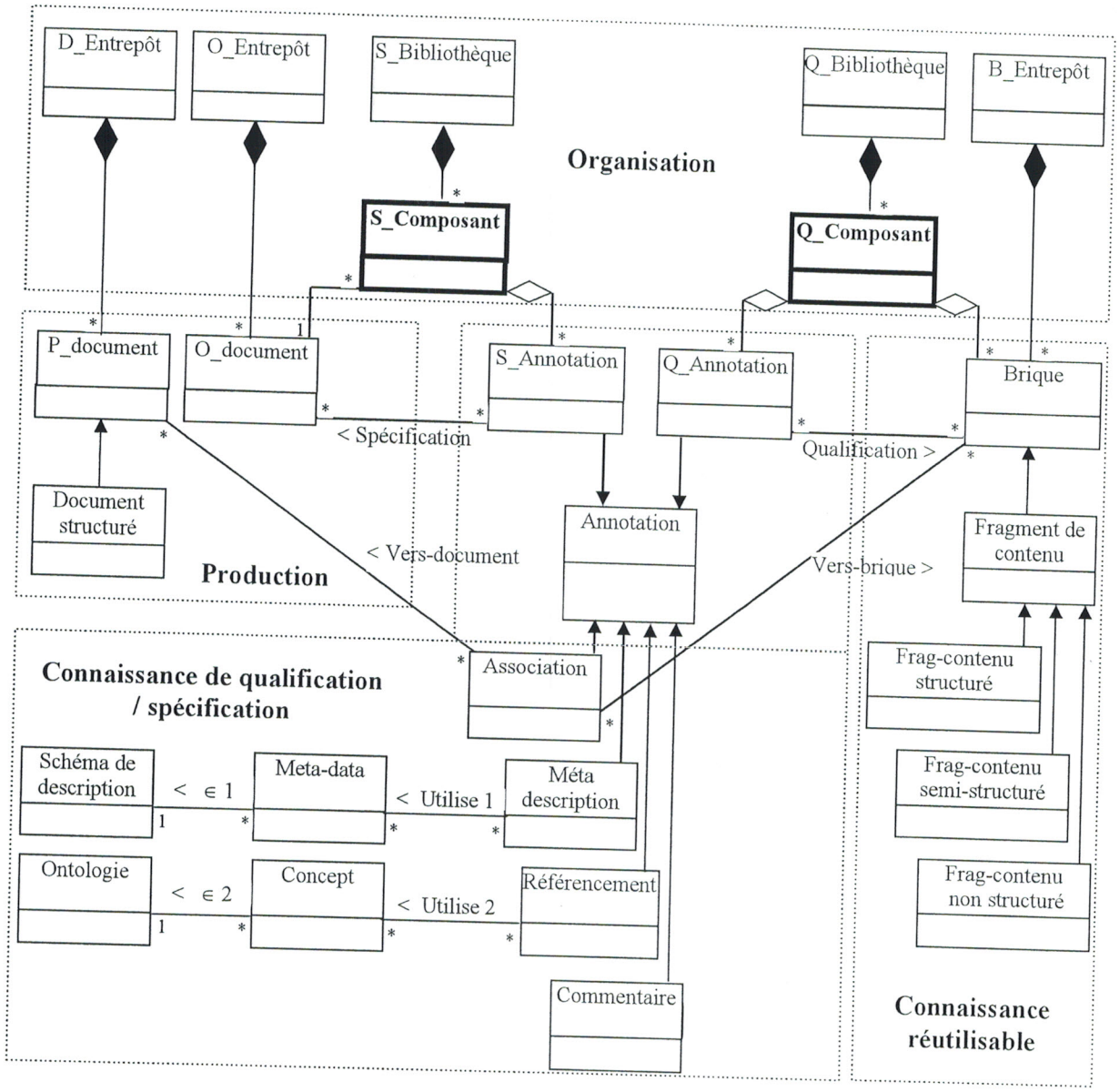


Figure 6.2. Modèle général de composant de document (UML) [CHIK04]

6.3. Conclusion

Dans le chapitre suivant, nous allons voir plus en détail l'apport du modèle ASARD pour la qualification des documents pédagogiques afin de construire notre entrepôt. Ce modèle va nous aider à qualifier les documents par les métadonnées de « LOM ».

7. CHAPITRE - Construction d'un entrepôt de métadonnées pédagogiques « E M P »

7.1. Introduction

Le document pédagogique se met au service du domaine éducatif. En effet, dans le contexte de l'E-learning, qui est définie comme « l'utilisation des nouvelles technologies multimédias et de l'Internet, pour améliorer la qualité de l'apprentissage en facilitant l'accès à des ressources et des services, ainsi que des échanges et la collaboration à distance », la multiplication des solutions logicielles et l'augmentation exponentielle des documents numériques diffusés sur le Web sont devenues des préoccupations importantes.

Cependant, il est actuellement impossible d'accéder de façon rationnelle, et avec de bonnes chances de trouver les documents pertinents, à cette masse grandissante de ressources. Il est donc nécessaire d'étudier les techniques qui permettent d'indexer, d'analyser, de composer, de mémoriser et de rechercher de tels documents.

Dans ce chapitre, nous allons proposer un modèle d'entrepôt qui va nous permettre de résoudre quelques difficultés des entrepôts dans le domaines de l'apprentissage et du E-Learning afin d'améliorer la recherche et l'analyse des documents pédagogiques.

Le concept d'entrepôt de documents doit permettre la gestion et l'exploitation aisée d'une mémoire documentaire constituée à partir d'informations qui doivent être filtrées et nettoyées, en vue d'être partagées et analysées. La construction de notre entrepôt de documents pédagogiques est basée sur la qualification par les métadonnées de LOM.

Les questions qui se posent sont :

- le pourquoi de cet entrepôt de documents pédagogiques ?
- le pourquoi d'une qualification par les métadonnées de LOM ?

7.2. Le pourquoi de l'entrepôt de documents pédagogiques ?

Dans le contexte du développement de l'enseignement supérieur, caractérisé par une compétition accrue, les universités cherchent continuellement à développer et échanger des documents pédagogiques.

On entend par « document pédagogique numérique » tout document qui peut être utilisée, réutilisée ou référencée dans toute activité liée à l'enseignement. Les « documents pédagogiques » peuvent être, par exemple, des transparents, des notes de cours, des pages Web, des logiciels de simulation, des programmes d'enseignement, des objectifs pédagogiques, etc.

Les études supérieures requièrent de la recherche bibliographique. Cependant l'enseignant/étudiant avait beaucoup de problèmes liés aux supports pédagogiques (livres, revues, support de cours, ……), parmi lesquels :

1. Insuffisance des ouvrages en nombre et en genre étant donné leurs coûts élevés ;
2. Supports de cours non capitalisés et difficile à reproduire, par conséquent ils ne sont pas échangeables : Chaque enseignant utilise ses propres outils (ordinateur, logiciel de traitement de texte, etc.) pour rédiger ses textes, produire ses transparents et ses pages Web, indépendamment de ses collègues. Il est souvent très difficile d'échanger ces documents.

Ainsi, la découverte de l'Internet offre d'innombrables ressources d'informations accessibles à travers la recherche sur le *Web*. Ce dernier a véritablement facilité l'accès aux « documents pédagogiques » mais le volume des documents dans le cadre de l'apprentissage et de la fouille de données « Datamart » sont de plus en plus importants.

Le problème qui se pose réside en l'incapacité à retrouver une information pertinente en utilisant des moteurs de recherche. En effet, les plus performants d'entre eux n'indexent que 16% des documents accessibles. Le problème vient du fait que les outils existant ne peuvent pas s'appuyer sur une description du contenu des documents et, la seule manière de rechercher une information est de la retrouver par l'intermédiaire de mots ou de phrases inclus dans les documents.

Donc, l'utilisateur est confronté à une surcharge d'informations (le nombre de documents restitués peut être excessif si la requête n'est pas assez précise), et il risque de perdre un temps considérable avant d'accéder à l'information recherchée.

Le deuxième problème est lié au Web où ce dernier représente un univers de ressources conçu de telle sorte qu'il soit ouvert et accessible à tout le monde. La navigation dans cet univers est facilitée par un mécanisme de lien assurant un certain regroupement «sémantique» de ces ressources. Ce qui a engendré un problème de désorientation où l'utilisateur risque d'être rapidement perdu dans la navigation à travers les documents

Afin d'éviter les problèmes cités ci-dessus et pour répondre aux besoins en terme de solution informatique, on a proposé une architecture d'un entrepôt qui permettra de regrouper et de capitaliser les documents pédagogiques en vue de les exploiter.

7.3. Le pourquoi d'un entrepôt de métadonnées pédagogiques ?

Après la capitalisation des documents pédagogiques dans des entrepôts de documents, plusieurs problèmes sont apparus, dus à la grande quantité d'informations issue des différentes universités.

Parmi ces problèmes :

- *Hétérogénité sémantique*
- *Hétérogénité des formats et volume excessif des document**

Une Solution aux problèmes reposant sur les méta-données et XML devient possible.

❖ ***Aider à la recherche d'information : les méta-données***

La première idée qui vient à l'esprit est d'ajouter une information de nature sémantique aux documents pédagogiques de manière à en obtenir une description plus précise. Les métadonnées sont appropriées pour atteindre ce but. Comme dans une librairie, on doit pouvoir être possible de retrouver nos documents pédagogiques grâce à certaines caractéristiques comme leur titre, leur auteur, leur domaine, etc.

Ainsi, si les contenus de livres sont les données, répertoires de bibliothèque ou index constituent des métadonnées parce qu'ils contiennent des informations sur les livres et leurs contenus.

Dans le domaine de l'éducation, des informations relatives aux auteurs de documents pédagogiques, à leurs champs d'intérêt, à leurs idées, à leurs objectifs pédagogiques, etc. sont des métadonnées. Ces métadonnées peuvent être incluses dans les ressources elles-mêmes ou enregistrées dans un fichier séparé.

L'association des métadonnées aux différents documents pédagogiques présente une amélioration considérable pour la localisation de la ressource : en permettant des recherches basées sur des champs « créateur, titre, etc,), l'indexation et l'accès à une partie du document.

Une recherche sur des champs (données structurées), est plus facile à réaliser qu'une recherche sur le contenu (données « brutes » ou/et semi-structurés), ainsi les métadonnées permettent de repérer de façon efficace les documents pédagogiques et donc d'assurer leur réutilisation, leur accessibilité et leur interopérabilité.

- La réutilisation signifie d'un document pédagogique pourrait être utilisé, suivant le besoin, dans différents contexte de l'enseignement.
- L'accessibilité est traduite généralement par une recherche facilitée par une description précise de la ressource
- L'interopérabilité signifie qu'un document peut être assemblée avec un autre afin de créer de nouveaux documents pédagogiques.

❖ ***Structurer les documents pédagogiques pour les échanges : XML***

La deuxième idée qui vient à l'esprit est de structurer logiquement le contenu de l'entrepôt de documents en utilisant le langage XML de manière à pouvoir séparer le fond de la forme. Ceci se résout par l'utilisation du langage XML et les métadonnées. XML apparaît actuellement comme le moyen le plus adéquat pour décrire, dans la perspective d'une diffusion sur l'intranet, à la fois métadonnées et structure des documents. En effet, ce langage permet non seulement de véhiculer les données, mais aussi de les décrire de façon précise. Le rapprochement entre XML et les entrepôts de données est très prometteur et le passage par XML semble être une voie privilégiée pour entreposer des données complexes.

En résumé, la problématique de ce mémoire est la capitalisation et l'exploitation *des différents documents pédagogiques existants au niveau des universités données.* Ceci est possible grâce à l'intégration de trois technologies : (1) les entrepôts comme moyen pour rassembler les documents, (2) le langage *XML* comme solution au problème de la représentation et l'échange d'informations et (3) les métadonnées comme moyen permettant de repérer plus efficacement des documents pédagogiques sur les différentes universités en facilitant la recherche par des « titre du document, taille , adresse, format, ….. ».

Proposition

Nous proposons alors un solution dans le domaine des entrepôts de données qui permettra d'intégrer et d'exploiter des documents pédagogiques jugés pertinents, représentés sous n'importe quel format, pour nos enseignants et étudiants. On obtiendra un entrepôt dont le contenu est représenté par les métadonnées de « LOM » obtenu après qualification des différents documents pédagogiques et en s'appuyant sur la technologie XML. Le mode de stockage pour les entrepôts locaux peut s'opérer en natif XML ou dans des bases de données classiques (relationnelles, orientées objets ou relationnelles-objets).

- L'association des métadonnées « LOM » aux différents documents pédagogiques offre un potentiel d'amélioration substantiel des possibilités de localisation de documents. Il est clair que l'objectif principal de l'utilisation des métadonnées reste l'amélioration de la qualité de recherche qui est basées sur des champs (titre, format, titre, etc) ;
- Cet entrepôt doit être à la fois facile à gérer et efficace. C'est la raison pour laquelle nous avons pensé à tirer profit du langage XML comme un format d'échange de document ainsi que ses langages de requêtes « Xpath, XQuery, ... » qui permettent d'accéder aux documents en permettant de faire des requêtes plus structurées et complexes.

7.4. Définition de l'entrepôt de documents pédagogiques :

En se basant sur la définition générale de l'entrepôt de donnée citée dans le chapitre 4, nous allons définir notre entrepôt ainsi :

« Notre entrepôt est défini comme une « source de documents pédagogiques, représentés par des métadonnées de «LOM », orientés-sujets, filtrés, intégrés, historisés (versions) et organisés comme support de recherche, d'interrogation ou d'analyse. »

Nous allons décrire dans ce qui suit les termes essentiels de cette définition :

- ***Orientés-sujet :*** grâce à leurs métadonnées, les documents pédagogiques peuvent s'organiser par sujets ou par domaine. Cette spécificité de l'entrepôt permet de faciliter la recherche, l'interrogation et l'analyse.

  Exemple : rechercher des exercices dans le domaine informatique, module « génie logiciel »

- ***Filtrés :*** l'entrepôt ne contient que les documents jugés pertinent « avant le chargement des documents pédagogiques, ils passent par une phase de nettoyage »
- ***Intégrés*** : les documents pédagogiques de l'entrepôt sont le résultat de l'intégration de différents documents pédagogiques issus des sources différentes et sous différents formats.
- ***Historisés*** : l'entrepôt doit permettre l'historisation des documents pédagogiques « archivage » de façon fiable, durable et sécurisé ;

Les documents pédagogiques sont produits en quantités très importantes et qui nécessitent d'être archivées de manière pérennes et indexées pour pouvoir être retrouvées.

7.5. Objectif de l'entrepôt de document

Les entrepôts de documents pédagogiques doivent permettre d'éviter les inconvénients des moteurs de recherches qui ne permettent pas des interrogations précises. La plupart de moteurs permettent une interrogation par mots clés dont le résultat est une liste de référence de documents contenant au moins un de ces mots clés. De plus, ils restituent le document dans son intégralité ce qui s'avère inadapté pour des documents longs qui peuvent ne pas être pertinents.

Parmi les objectifs de notre modèle :

- Fournir un meilleur accès à la documentation pédagogique numérisée;
- Faciliter la réutilisation et les mises à jour;
- Permettre l'édition sous de multiples formats d'un document pédagogique
- Abaisser les coûts de la documentation
- Mettre en place (si nécessaire) des procédures de sécurité et d'authentification dans l'utilisation des « documents pédagogiques »;
- Améliorer la qualité de la documentation pédagogique;
- Faire des recherches avec des critères sémantiques « recherche multicritère »;
- Constituer des bases de données relatives aux « documents pédagogiques ». cexemple : entrepôt des documents d'informatique, électronique, littérature, ...

7.6. Identification des acteurs et leurs activités

Notre étude requiert l'intervention de deux catégories d'acteurs, ceux qui sont chargés d'administrer l'entrepôt et le superviser afin que notre entrepôt soit organiser et bien alimenter par des documents pédagogiques pertinents, ainsi que ceux qui peuvent être soit des enseignants ou des étudiants, qui utilise l'entrepôt à des fin de recherche multicritère ou autres.

1- Les administrateurs (Les experts)

Il existe deux types d'experts :

a- Des experts qui vont superviser l'état des entrepôts locaux, par exemple sélectionner l'entrepôt qui a le plus d'accès à ses documents, superviser les entrepôts qui sont mis à jour de façon régulière, etc. Ces experts vont jouer le rôle de serveur « Serveur E.M » de l'entrepôt de métadonnées. Ils réceptionnent les requêtes des clients « utilisateurs » et leur renvoient les adresses des documents pédagogiques associés en indiquant dans quel entrepôt local se trouve. Ces adresses sont obtenues grâce aux descripteurs des métadonnées de « LOM » ;

b- D'autres experts qui ont une très bonne capacité de connaissance qui leur permettra d'importer des documents pédagogiques qu'ils jugent intéressants à les capitaliser dans les entrepôts locaux, puis les qualifiés par les métadonnées de « LOM ». Ces experts jouent aussi le rôle d'un serveur, où chaque entrepôt local est représenté par son serveur. Une fois le client obtient l'adresse du document recherché à travers le Serveur E.M, il demande son téléchargement à travers le Serveur correspondant à l'entrepôt local. Ce dernier va lui exécuter sa requête, en lui demandant soit par payement ou en lui demandant le login « droit d'accès » ou autres.

Activités de l'administrateur « expert »

En général, l'administrateur est chargé de :

- Alimenter les entrepôts locaux par des documents pédagogiques jugés pertinents ;
- Qualifier chaque document pédagogique stocké dans l'entrepôt local par les métadonnées de LOM pour faciliter leur organisation et leur recherche ;
- Construire l'entrepôt de métadonnées ;
- Gérer la maintenance des entrepôts et leur mise à jour ;
- Gérer les droits d'accès ;

L'expert à la possibilité d'exploiter l'entrepôt « recherche, analyse ..... ».

2- Les utilisateurs (Les auteurs)

Les utilisateurs peuvent être soit des enseignants ou des étudiants. C'est des auteurs qui désirent réaliser des recherches multicritères sur l'entrepôt pédagogique, afin de repérer et d'acquérir des documents adaptés à leurs demandes d'une manière simple, rapide et efficace. Les utilisateurs sont considérés comme des clients de l'entrepôt.

Les **activités** des utilisateurs consisteront à :

1- Réaliser des recherches pertinentes appropriées pour des besoins spécifiques;

2- faire des analyses/requêtes par exemple un utilisateur cherche un support de cours sur le « génie logiciel » niveau 3ème année ingénieur informatique ;

3- Demande de conversion des documents d'un format à un autre ;

4- Donner leur avis pour chaque entrepôt visité afin de pouvoir juger la pertinence de chaque entrepôt local.

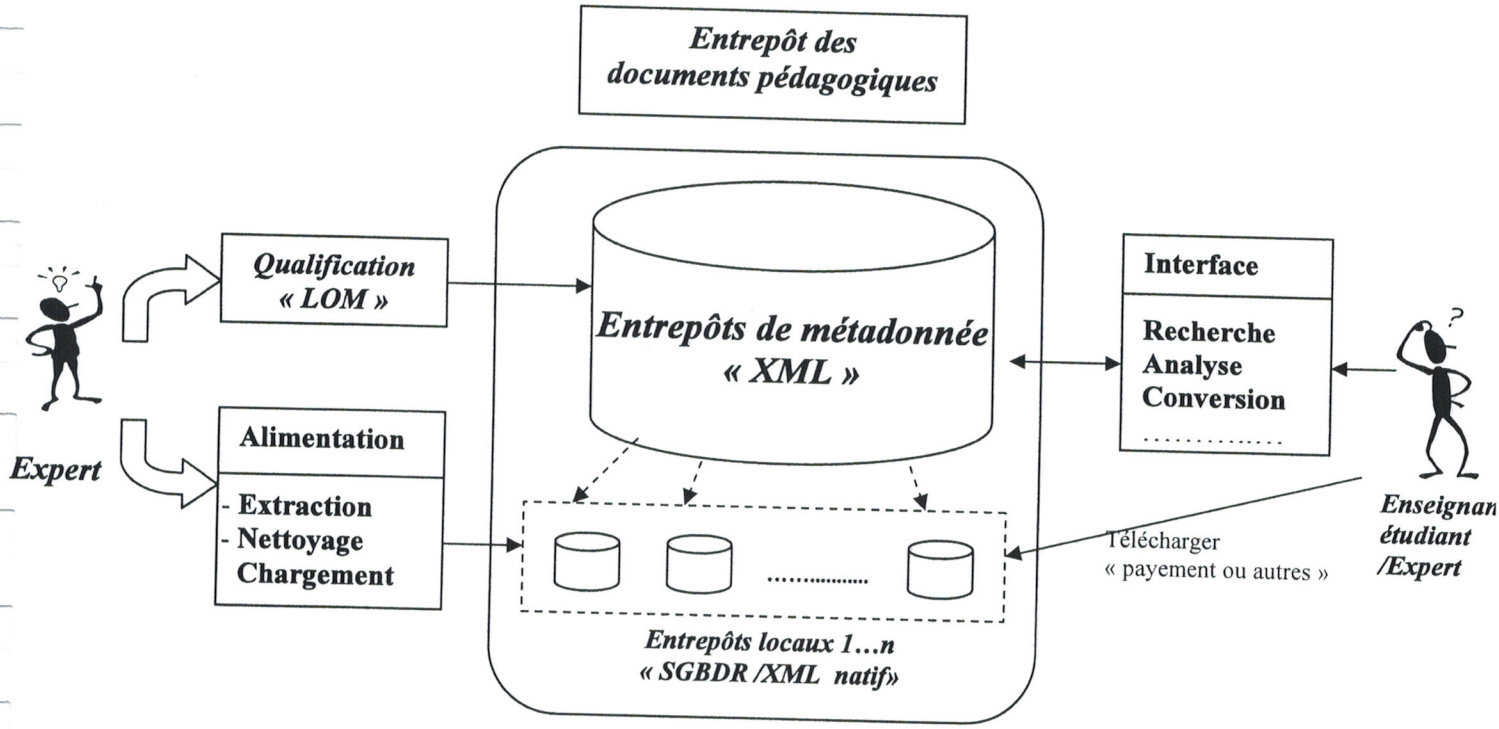


Figure 7.1: Activités des acteurs

7.7. Architecture de l'entrepôt de document

L'architecture choisie pour notre modèle est une architecture client/serveur afin de mettre à la disposition des utilisateurs les différents documents pédagogiques, et leur faciliter l'accès **à** travers un serveur Web.

Il faut se rappeler que notre entrepôt est considéré comme une partie du projet SABRA. Il est composé de deux modules. (1) Le premier module contient la connaissance de réutilisation « entrepôt de métadonnées ». Il est représenté par les différents descripteurs de « LOM », (2) l'autre module contient la connaissance réutilisable « les documents pédagogiques ». Ce module est représenté par un ensemble d'entrepôts locaux où chacun représente son propre site « université » afin de faciliter le chargement et la maintenance.

Ce deuxième module s'appuis sur l'approche matérialisée, où les documents sont capitalisés dans des entrepôts. A partir des requêtes des utilisateurs, et en utilisant les métadonnées et la technologie XML, on pourra accéder aux documents qui se trouvent dans les entrepôts locaux sans accéder à leurs sources.

Notre architecture de stockage offre les avantages suivants :

- nous séparons le stockage des documents pédagogiques de celui des métadonnées. Cela permet d'éviter la redondance des documents dans l'entrepôt et a pour effet de:
    - Améliorer l'espace de stockage ;
    - Faciliter la maintenance des documents pédagogiques ;
    - Faciliter la recherche des documents en se basant sur les métadonnées LOM.

Notre entrepôt de documents est représenté par l'architecture suivante :

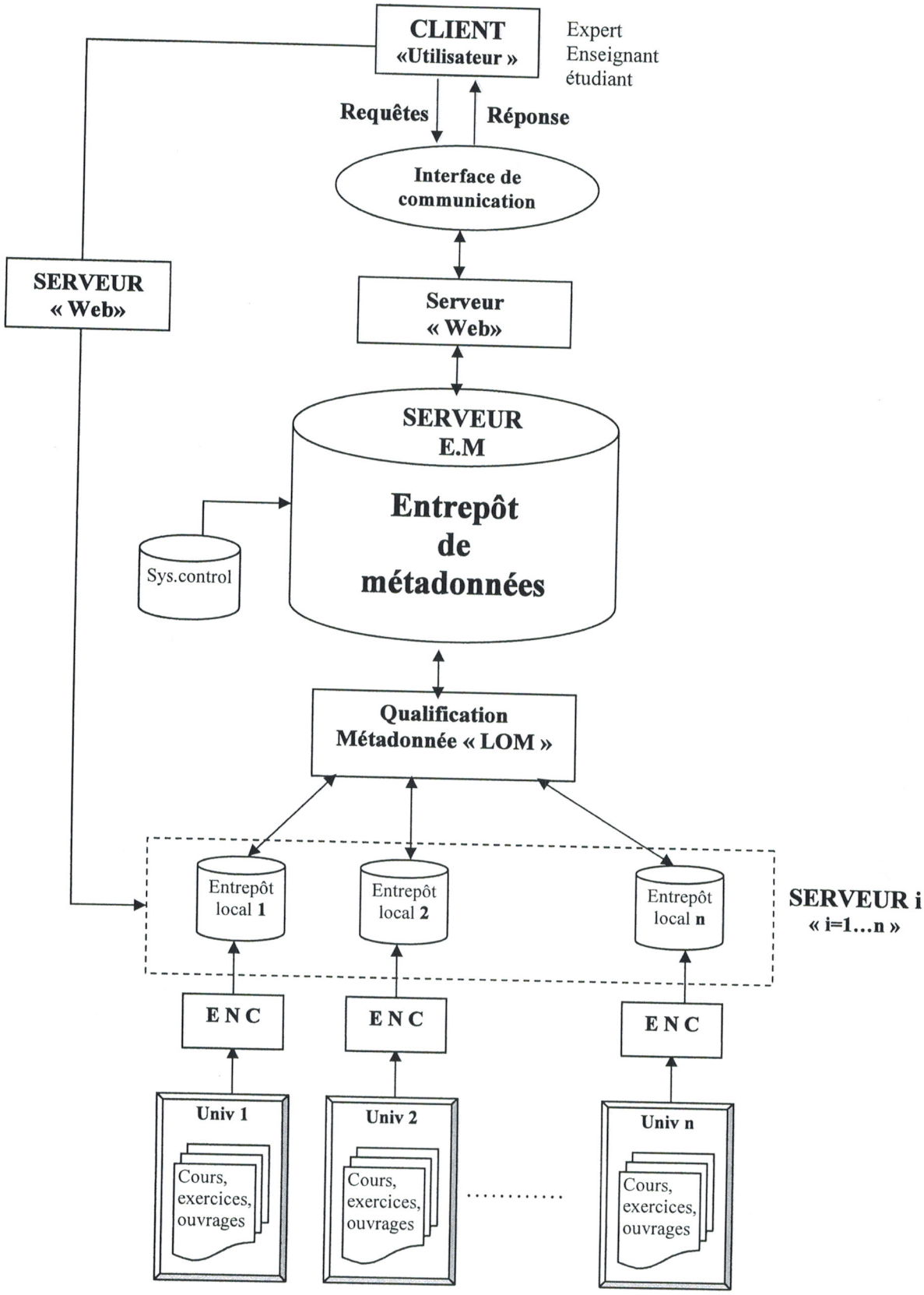


Figure 7.2 : Architecture de l'entrepôt

Nous allons décrire maintenant les différents composants de notre modèle :

1- Le module ENC « Extraction Nettoyage Chargement »

Ce module est destiné aux experts. Ils se connectent à ce module par l'intermédiaire d'une interface sécurisé. Son rôle consiste à l'alimentation de l'entrepôt de chaque université « cours, rapports, ouvrages, ... » .

Le module « ENC » se compose de trois phases : (1) extraction des documents pédagogiques, (2) nettoyage, (3) et le chargement dans l'entrepôt local.

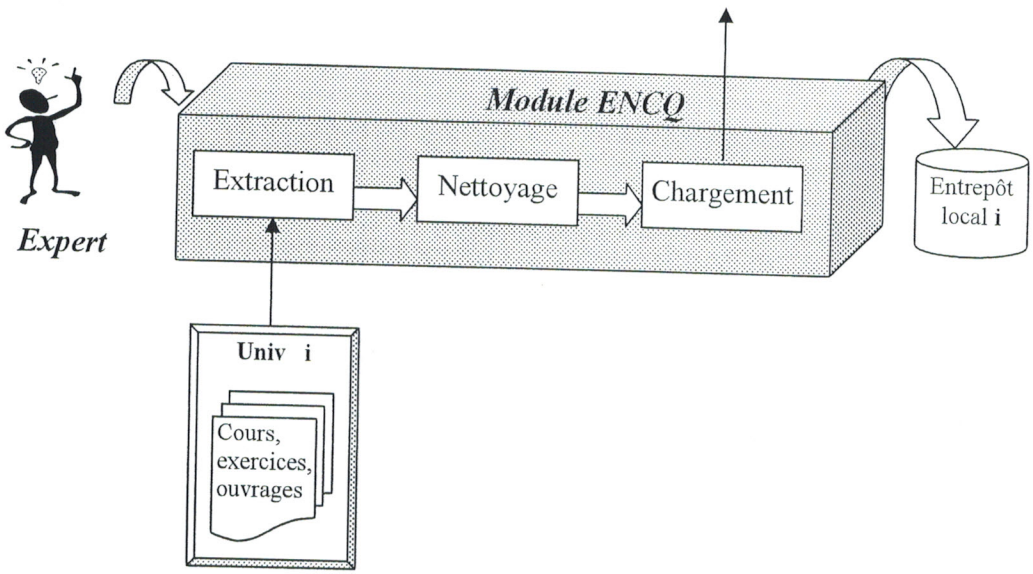


Figure 7.3: Architecture du module ENCQ

a- Extraction

Cette première phase de la construction d'un entrepôt consiste à extraire les documents jugés utiles et pertinents à partir des différentes sources de documents. Pour cela, plusieurs technologies sont utilisables :

- Les passerelles, fournies par les éditeurs de base de données, Ces passerelles sont généralement insuffisantes car elles sont mal adaptées aux processus de transformation complexes ;
- Les utilitaires de réplication, utilisables si les systèmes de production et décisionnel sont homogènes et si la transformation à appliquer aux données est légère ;
- Les outils spécifiques d'extraction (prix élevé), exemple : DataStage, Génio, warehouse Administrator ,..... .

b- Nettoyage

Le "nettoyage" des documents a pour but d'améliorer la qualité des documents. Il permet :

- Identifier et corriger les documents invalides ;
- Traiter les violations de contraintes d'intégrité ;
- Ajouter des heuristiques au module d'extraction des documents pour trouver si deux documents représentent la même information « éviter la redondance ».

De nombreux outils sont disponibles sur le marché pour nettoyer les documents. Exemple ACTAWorks d'ACTA, EDD DataCleanser et GENIO de Leonards Logic appliquent des règles de transformation, pouvant être écrites par l'utilisateur, ou par interfaçage avec d'autres outils de nettoyage. VALITY et ID-CENTRIC de Firstlogic nettoient les documents en intégrant de la logique floue.

c- Chargement

Il constitue la dernière phase d'alimentation. Une fois les documents pédagogiques sont extraits de leurs sources et nettoyés, l'expert va les chargés dans leur entrepôt local. Chaque entrepôt local est représenté soit par un SGBD ou en utilisant le XML natif, qui va contenir l'ensemble des documents pédagogiques de chaque site « université ».

2- Les entrepôts locaux

Chaque entrepôt local va contenir l'ensemble des documents déjà nettoyés, indexés par leurs identifiants « Id ». Le choix de la représentation SGBD/XML natif va permettre à l'entrepôt de contenir une grande quantité d'informations. A chaque mise à jour de document « ajout, suppression, modification », l'expert est chargé de mettre à jour son entrepôt. Chaque entrepôt est représenté par son serveur lui permettant de communiquer avec les utilisateurs et de gérer sa maintenance.

3- Module de qualification

Ce module qui est pris en charge par l'expert va permettre de qualifier chaque document pédagogique issu de l'entrepôt local. Le résultat, qui est représenté par des métadonnées de LOM, sera stocké dans l'entrepôt des métadonnées sur lequel se baseront les requêtes « recherche, analyse, ... » des utilisateurs « enseignants/ étudiants/experts ». La représentation de cet entrepôt est à base du langage XML.

L'utilisation de la norme LOM permet outre une description pertinente des documents, la possibilité d'échange avec d'autres applications utilisant également cette norme. Le modèle LOM définie chaque document pédagogique par ses neufs catégories comprenant au total 79 éléments décrivant un document :

- Généralités « id, titre, description, ... » ;
- Cycle de vie « version, statut, ... » ;
- Méta-métadonnées « schéma utilisé, langue..... » ;
- Informations techniques « format, localisation, ... » ;
- Informations pédagogiques «type interactivité, durée, difficulté, ... » ;
- Droits « coût, description,..... » ;
- Relations « ressource,....» ;
- Commentaires « date, ...» ;
- Classification «objectif, ... » .

Pour la modélisation des documents pédagogiques dans l'entrepôt, on peut utiliser soit :

- Les SGBD,
- Les métadonnées associées à XML
- Les ontologies RDF, RDFs, OWL

3.1 Modélisation par les SGBD

Chaque document pédagogique qualifié par le «LOM » aura au maximum 79 métadonnées. Supposons qu'on va représenter notre entrepôt par un SGBD. L'entrepôt sera représenté par un ensemble de 10 tables : une par descripteur « 9 tables relationnelles », en plus de la dixième table qui va contenir les adresses secondaires du document « car dans le champ « localisation » on peut se retrouver avec un document qui a plusieurs adresse ».et chaque document sera considéré comme un tuplet. Ainsi l'entrepôt aura l'aspect suivant :

1. Général

Id	titre	description	Mots clefs	Contraint techn	Niv agrégat	.....
01	Archirect des ordi	2105k	M1	WinXp		
02	Pédiatrie	513 k	M2	----		
03	Electronique		M3			
05	Cours1	35go	M5			
.....	.....	....	.....	.....		

2. Cycle de vie

Id	Version	Statut	contribution	...
01	6.0			
02	3.1			
.....	.....	....	.....	

4. Technique

Id	Format	localisation	Durée	exigences	.....
01	Pdf	Source1		Facile	
02	Doc	Source2			
04	Mpeg	Source1			
05	Wav	Source5	1h		
.....	.....	....			

3. méta métadonnée

Id	Schéma	langue	.....
01	LOM	Fr	
02		Arabe	
04	DC		
05		Ang	
.....	.....	....	

5. Pédagogique

Id	Type intéractiv	Type ressource pédagog	Durée apprentis	Context	difficul	....
01	active		2h	inform	moyen	
03		simulation			facile	
.....	.....	....				

6. Droit

Id	Coût	Descrip	.....
02		Trt texte	
03		Calcul	
04	Pas cher		
.....	.....	....	

7. relation

Id	Ressour	.....
01	R1	
.....	.....	

8.commentaire

Id	Date	.....
01	22-01-05	
02	01-01-01	
03	13-04-75	
04	08-10-99	
.....	.....	

9 .classification

Id	Objectif	.....
01	educationn	
02		
03		
.....	.....	

10. Table adresse

Id	Adr1	Adr2	Adr3	....
3	Adr 31	Adr_{32}		
78				

Figure 7.4 : Représentation de l'entrepôt en utilisant un SGBD

Critique

D'après la figure, on remarque que l'utilisation d'un SGBD pour cet entrepôt n'est pas intéressant , vu le nombre élevé des champs de LOM , l'existence de plusieurs champs non utilisés « vides » et qui occupent de la place inutile.

3.2 Modélisation par les ontologies

La modélisation de l'entrepôt par les ontologies permet une gestion sémantique des documents pédagogiques. Cela est possible en utilisant un langage de représentation des ontologies comme RDF, RDFs, OWL, etc. Vu le manque de sémantique en utilisant le langage XML, RDF et *RDF Schema* ont essayé de résoudre ce problème en permettant d'associer des sémantiques simples aux identificateurs.

Or le but de notre approche consiste à faire des recherches sur les descripteurs des documents et non pas sur la sémantique des documents, ce qui rend la modélisation difficile. Par conséquent la modélisation de l'entrepôt en utilisant la technologie XML ainsi que le modèle ASARD pourra être une solution pour notre problème.

3.3 Modélisation XML

Comme on a vu dans le projet SABRA cité dans le chapitre précédant, la qualification par les métadonnées est prise en charge dans le modèle ASARD grâce à la première catégorie des q_annotations.

META-DESCRIBE (ID , bricks, {(Meta-data, Schema,{Value})} , Q_Text)

L'élément principal de la DTD, relative aux q_annotation est l'élément <meta_describe> qui représente la qualification d'un document pédagogique et qui est représenté par son identifiant q_id. en outre, chaque <meta_describe> est à son tour composé de deux éléments : l'élement <meta_datas> et l'élément <q_text>.

L'élément <meta_datas> est composé d'un nombre quelconque d'éléments <meta_data>. Chacun de ces derniers est composé à son tour de plusieurs éléments atomiques <value> et est caractérisé par deux attributs ***name*** et ***schema***, désignant respectivement le nom de la métadonnée et le schéma de description auquel elle appartient. La DTD suivante, relative à l'entrepôt des métadonnées, représente la qualification des documents pédagogiques :

```
< ?xml version='1.0' encoding ='UTF-8' ?>
< !-- ********************************************************************************
                                DTD de l'entrepôt des métadonnées
Version 1.0 du 14-07-2005
Auteurs : N. Iles
Département d'informatique –Faculté des sciences de l'ingénieur –
Université de Tlemcen
 ********************************************************************************* -- !>
< !ELEMENT qualifications (meta_describe)>
< !ELEMENT meta_describe (meta-datas, q_text)>
< !ATTLIST meta_describe q_id  ID #REQUIRED >
< !ELEMENT meta-datas (meta_data)*>
< !ELEMENT meta-data (value)*>
< !ATTLIST meta_data name CDATA #REQUIRED schema FIXED " LOM ">
< !ELEMENT value (#PCDATA)>
< !ELEMENT q_text (#PCDATA)>
```

Listing 7.1 DTD de l'entrepôt des métadonnées

Soit Doc1 et Doc2 deux cours que l'expert voudra les qualifier et introduire leurs métadonnées dans l'entrepôt. Voici une instance XML de la DTD :

```
< ?xml version='1.0' encoding ='UTF-8' ? >
< qualifications >
  <meta_describe q_id= « q1 » >
      <meta_datas>
        <meta_data  name = General.title"  schema="LOM" >
            <Value> initiation à Microsoft Word </value>
        </ meta_data >

        <meta_data  name = General.Identifier.Entry" Schema="LOM">
            <Value> http://www.documentserveur1.dz/univ1/a.pdf </value>
        </ meta_data >

        <meta_data  name = General.description"  schema="LOM" >
            <Value> Un support de cours présentant les fonctionnalités avancées
                    de Microsoft Word </value>
        </ meta_data >

        <meta_data  name = General.keyword"  Schema="LOM">
            <Value> icone </value>
        </ meta_data >

        <meta_data  name = General.keyword"  Schema="LOM">
            <Value> saisie </value>
        </ meta_data >

        <meta_data  name = General.language"  Schema="LOM">
            <Value> ar </value>
            <Value> en-GB </value>
        </ meta_data >

        <meta_data   name= "Lifecycle.version "    Schema="LOM">
            <Value> 3.0  </value>
        </ meta_data >

        <meta_data   name= "MetaMetadata.Contribute.role "    Schema="LOM">
            <Value>Createur </value>
        </ meta_data >

        <meta_data   name= "MetaMetadata.Contribute.entity "    Schema="LOM">
            <Value> Mr Dupon </value>
        </ meta_data >

        <meta_data   name= "MetaMetadata.Contribute.date "    Schema="LOM">
            <Value> 08-10-97 </value>
        </ meta_data >
```

```
        <meta_data  name= "Technical.format "   Schema="LOM">
              <Value> Doc </value>
         </ meta_data >

        <meta_data  name= "Technical.size "   Schema="LOM">
              <Value> 4280 </value>
         </ meta_data >

        <meta_data  name= "Educationnal.Interactivity type "   Schema="LOM">
              <Value> Active </value>
         </ meta_data >

        <meta_data  name= "Educationnal.Interactivity level "   Schema="LOM">
              <Value> élevé </value>
         </ meta_data >

        <meta_data  name= "Educationnal.difficulty "   Schema="LOM">
              <Value> facile </value>
         </ meta_data >

        <meta_data  name= "Rights.cost "   Schema="LOM">
              <Value> non </value>
         </ meta_data >

        <meta_data  name= "classification.Educationnal "   Schema="LOM">
              <Value> objectif educationel </value>
         </ meta_data >

        <meta_data  name= "classification.Description"   Schema="LOM">
              <Value> Traitement de texte </value>
         </ meta_data >
    </meta_datas>
</meta_describe>

<meta_describe q_id= « q2 » >
    <meta_datas>
        <meta_data  name = General.title"  schema="LOM" >
              <Value> Internet </value>
         </ meta_data >

        <meta_data  name = General.Identifier.Entry" Schema="LOM">
              <Value> http://www.documentserveur2.dz/univ2/bb.ppt </value>
         </ meta_data >

        <meta_data  name = General.language"  Schema="LOM">
              <Value> fr </value>
         </ meta_data >
        <meta_data  name = General.keyword"  Schema="LOM">
              <Value> Web </value>
```

```
        </ meta_data >
        <meta_data  name= "Lifecycle.version "   Schema="LOM">
            <Value> 5.0  </value>
        </ meta_data >

        <meta_data  name= "MetaMetadata.Contribute.role "   Schema="LOM">
            <Value>Createur </value>
        </ meta_data >

        <meta_data  name= "MetaMetadata.Contribute.entity "   Schema="LOM">
            <Value> Mr Croazt </value>
        </ meta_data >

        <meta_data  name= "MetaMetadata.Contribute.date "   Schema="LOM">
            <Value> 29-05-2000 </value>
        </ meta_data >

        <meta_data  name= "Technical.format "   Schema="LOM">
            <Value> ppt </value>
        </ meta_data >

        <meta_data  name= "Educationnal.Interactivity type "   Schema="LOM">
            <Value> Active </value>
        </ meta_data >

        <meta_data  name= "Educationnal.Interactivity level "   Schema="LOM">
            <Value> moyen </value>
        </ meta_data >

        <meta_data  name= "Educationnal.difficulty "   Schema="LOM">
            <Value> difficile </value>
        </ meta_data >

        <meta_data  name= "Rights.cost "   Schema="LOM">
            <Value> non </value>
        </ meta_data >

        <meta_data  name= "classification.Educationnal "   Schema="LOM">
            <Value> objectif educationel </value>
        </ meta_data >

        <meta_data  name= "classification.Description"   Schema="LOM">
            <Value> Recherche </value>
        </ meta_data >
    </meta_datas>
  </meta_describe>
</qualifications>
```

Listing 7.2 : Instance de l'entrepôt de métadonnée

Remarque : En utilisant le langage XML, il n'est pas obligatoire de remplir tous les champs de « LOM ».

4- Le module Serveur Web

Le système repose sur une architecture classique Client/Serveur qui est représentée ainsi :

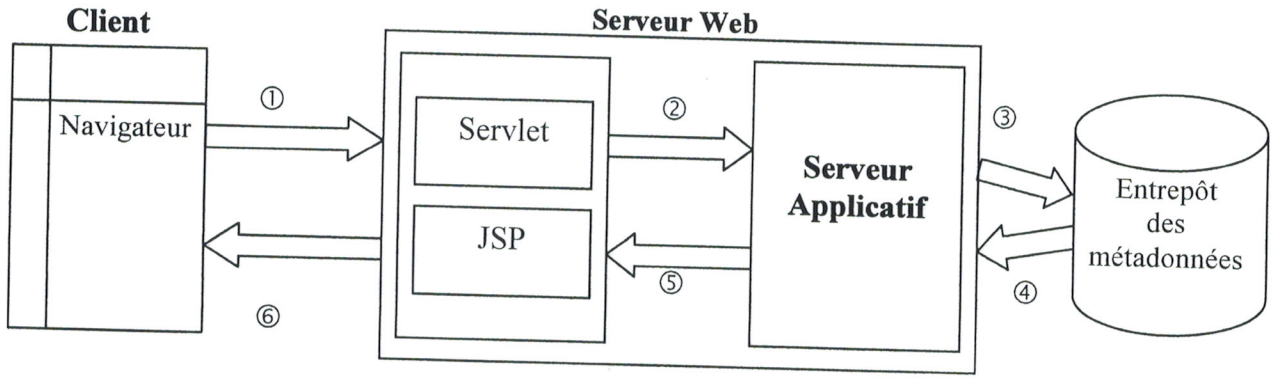


Figure 7.5 : architecture du module Serveur Web

Suivant ce schéma, le cycle de vie d'une requête est le suivant :

1- le client « navigateur Web » envoie une requête à l'application en utilisant le langage XQUERY. La requête est prise en charge par le servlet d'entrée. ;

2- le servlets d'entrée analyse la requête et la réoriente vers le serveur d'application « serveur Web » qui va permettre d'utiliser un contrôleur adapté ;

3- Le controleur sélectionné par le servlets va s'adresser à l'entrepôt et il va exécuter le traitement nécessaire à la satisfaction de la requête. Si la requête est une recherche de document, on aura comme réponse, l'ensemble des documents représentés par leurs ID, URI qui vont satisfaire notre requête ;

4- Les réponses obtenues sont fournies au serveur d'application ;

5- Le serveur d'application transmettra la réponse, sélectionne la JSP qui sera pris en charge de la construction de la réponse ;

6- La JSP (vue) construit la réponse qui lui ont été transmis et l'envoie au navigateur.

Concernant le navigateur Web, il est composé de deux interfaces graphiques : une interface d'alimentation où l'expert va capitaliser les documents pédagogiques dans des entrepôts locaux, et les qualifie à base de métadonnées de LOM. L'autre est destinée à l'utilisateur « enseignant, étudiants » pour l'exploitation de l'entrepôt.

Sur la figure ci-dessous, nous avons présenté le cycle de vie d'une requête depuis la demande de la requête jusqu'à obtenir le résultat :

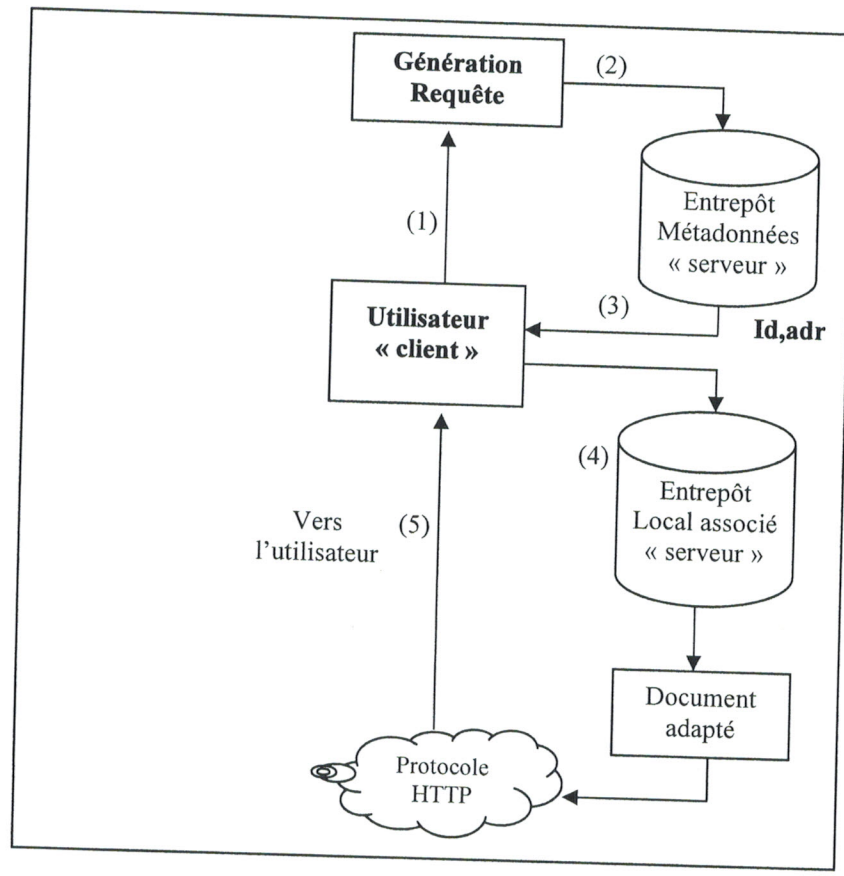


Figure 7.6 : cycle de vie d'une requête

Lors d'une demande de recherche d'un ou de plusieurs documents« cours, mémoire, etc. » (1), la requête sera faite sur l'entrepôt de métadonnée (2). Les différents champs du « LOM » seront remplis (pas obligatoirements tous) et les documents recherchés seront localisés grâce à la modélisation XML et le modèle « ASARD ». L'identifiant, le titre ainsi que l'adresse des documents seront obtenues grâce à travers le champ « General.Identifier.Entry », « General.title » et « Technical.location », et seront transmis à l'utilisateur (3). Ce dernier va accéder directement à l'entrepôt local concerné (4) pour le chargement du contenu du document pédagogique (5).

Pour réaliser une requête d'un client, on utilise un moteur de recherche XQuery. Les résultats obtenus et qui vont satisfaire les critères de la requête suivent la DTD suivante :

```
< ?xml version='1.0' encoding ='UTF-8' ? >
< !-- ********************************************************************************************
                                        DTD résultat requête
Version 1.0 du 14-07-2005
Auteurs : N. Iles
Département d'informatique –Faculté des sciences de l'ingénieur –
Université de Tlemcen
 ********************************************************************************** -- !>
< !ELEMENT results (document)* >
< !ELEMENT document (titre, URI, size >
<ATTLIST document  q_id  ID # REQUIRED >
< !ELEMENT title (#PCDATA)>
< !ELEMENT URI (#PCDATA)>
< !ELEMENT size (#PCDATA)>
```

Listing 7.3 : La DTD du résultat d'une requête

Voici un exemple d'une instance XML de la DTD :

```
< ?xml version='1.0' encoding ='UTF-8' ? >
<results>
  <document  q_id= »q1 »
    <title> initiation à Microsoft Word </title>
    <uri> http://www.documentserveur1.dz/univ1/a.pdf </uri>
     <size> 4280 </size>
  </document>
   <document  q_id= »q7 »
    <title> Architecture des ordinateurs </title>
    <uri> http://www.documentserveur1.dz/univ5/ordi.ppt </uri>
     <size> 5800 </size>
  </document>
</results >
```

Listing 7.4: un exemple du résultat d'une requête

Exemple

Supposons qu'on veut rechercher les documents en langue française dont le mot clef est « Web » et « Internet,

La requête XQuery de l'exemple précédant s'écrit comme suit :

```
<results>
{
        let $doc := doc (« file :c:/client/q_library/q_ibrary.lib »)
        for $brq in ($doc/meta_describe_qualif/meta_describe),
              $mds in $brq/meta_datas
        where  ($mds/meta_data [@name="General.language"]/value="fr")
        and      ($mds/meta_data [@name="General.keyword"]/value="Web")

        and      ($mds/meta_data [@name="General.keyword"]/value="Internet")
        return
        <brique  b_id="{$brq/@b_ids}">
        {
          for £res  in  ($brq/meta_datas/metada),
          $md  in  $res [value]
          return
            if ($md/@name= "General.Identifier.Entry")
             then
              < uri >
                { $md / value / text () }
              < /uri >
   if ($md/@name= "General.Title")
             then
              < title >
                { $md / value / text () }
              < /title >
            else ()
      }
   <brique>
}
</results>
```

Listing 7.5 : exemple d'une requête XQuery

6- Le module Control

Nous avons imposé ce module qui est pris en charge par une soumission composé d'un certains nombre d'expert. Cette soumission va permettre à partir des actions des utilisateurs d'évaluer les entrepôts locaux, les classer et donner leur caractéristiques : date de création de l'entrepôt, nombre de documents qu'il contient, le nombre de fois accéder, date de la dernière mise à jour de l'entrepôt, etc.

Ce module est doté même d'une messagerie qui va permettre le contact direct aux utilisateurs afin de donner leur avis concernant les entrepôts.

2- Alimentation de l'entrepôt

Après que l'expert valide son accès, il lui sera proposé un ensemble d'activités pour l'alimentation de l'entrepôt.

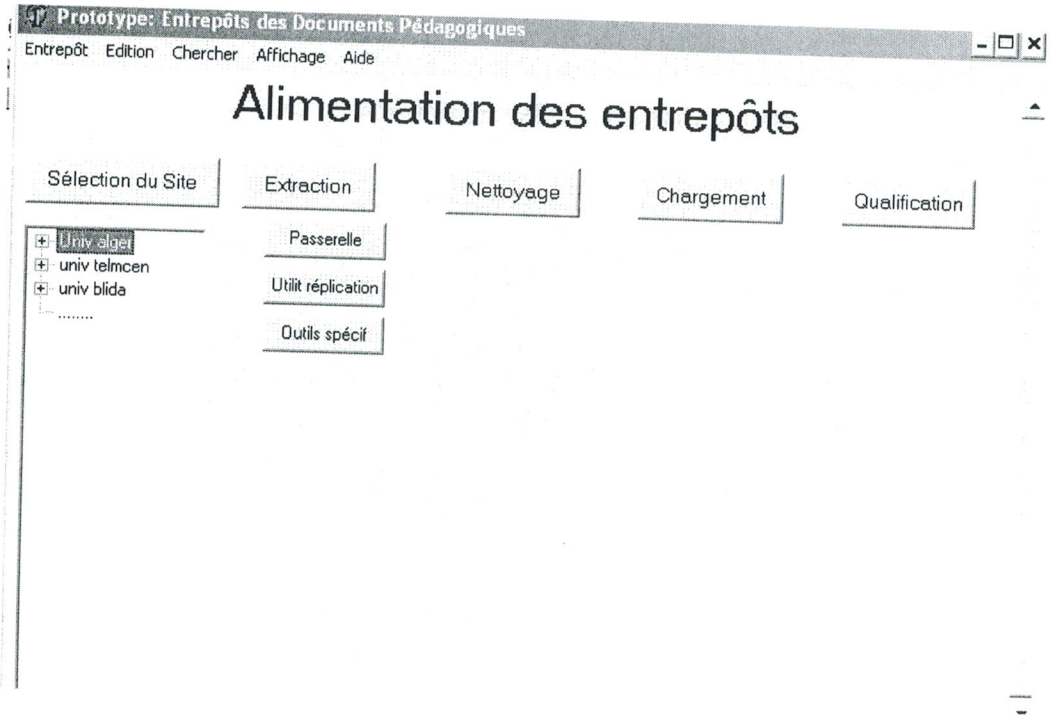


Figure 7.8 : Etape de construction de l'entrepôt

La partie gauche de la figure 7.8 représente la liste des sites « universités contenant leurs documents « pdf, doc, ppt, ... ». Le document est sélectionné de son site d'origine afin qu'il soit traité « extraction & nettoyage » et charger dans son entrepôt local.

La phase d'extraction et de nettoyage va utiliser les outils déjà cités dans la partie ci-dessus, qui sont conçus par d'autres concepteurs. Ainsi, chaque université est représentée par son entrepôt, et l'alimentation de ses documents pédagogiques se fera dans une BDD.

3- La qualification

Cette interface va nous permettre de construire l'entrepôt de métadonnées sur lequel se basera les requêtes :

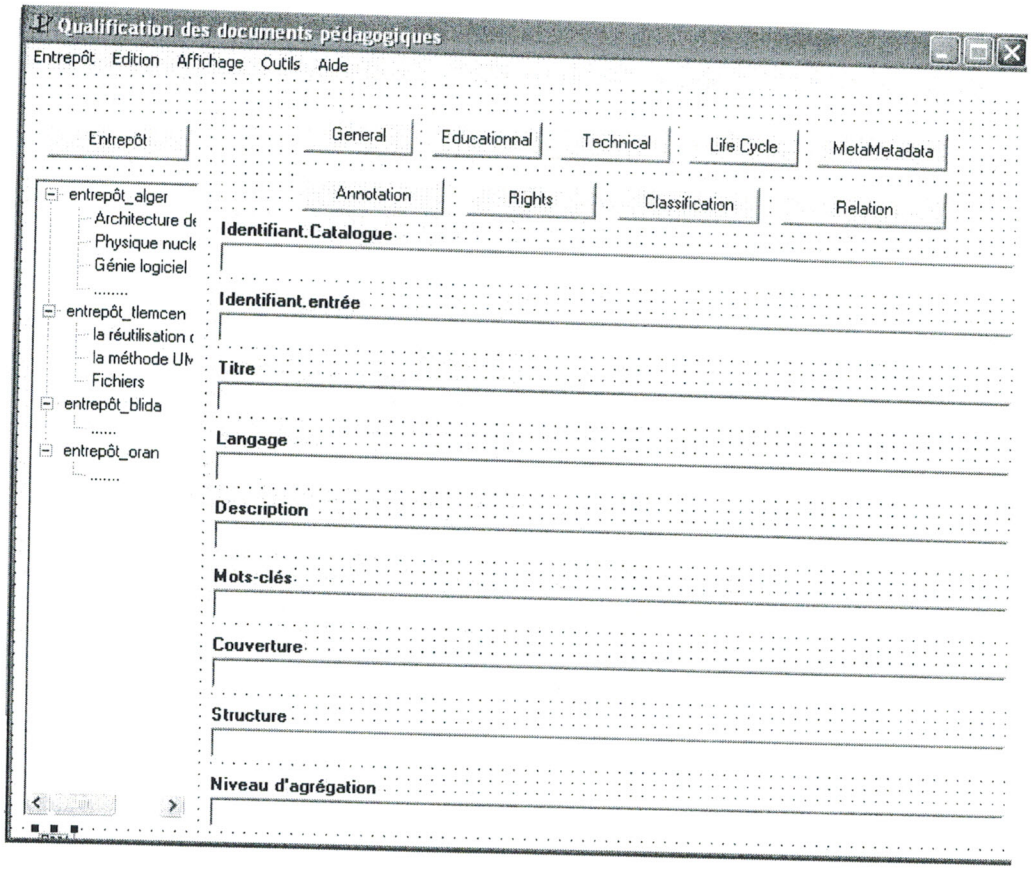


Figure 7.9 : Phase de qualification

La partie gauche de l'interface va contenir l'ensemble des entrepôts locaux déjà alimentés. L'expert va maintenant sélectionné chaque documents issu de cet entrepôt et le qualifié par l'annotation méta_décrire, en se basant sur le schéma de « LOM ».

La partie droite va contenir l'ensemble des descripteurs de LOM.

4- Exploitation de l'entrepôt par l'utilisateur

Lors de l'accès de l'utilisateur, la requête pourra se présentée sous forme d'une recherche. Cette recherche se base sur plusieurs critères ce qui diffère des moteurs de recherche.

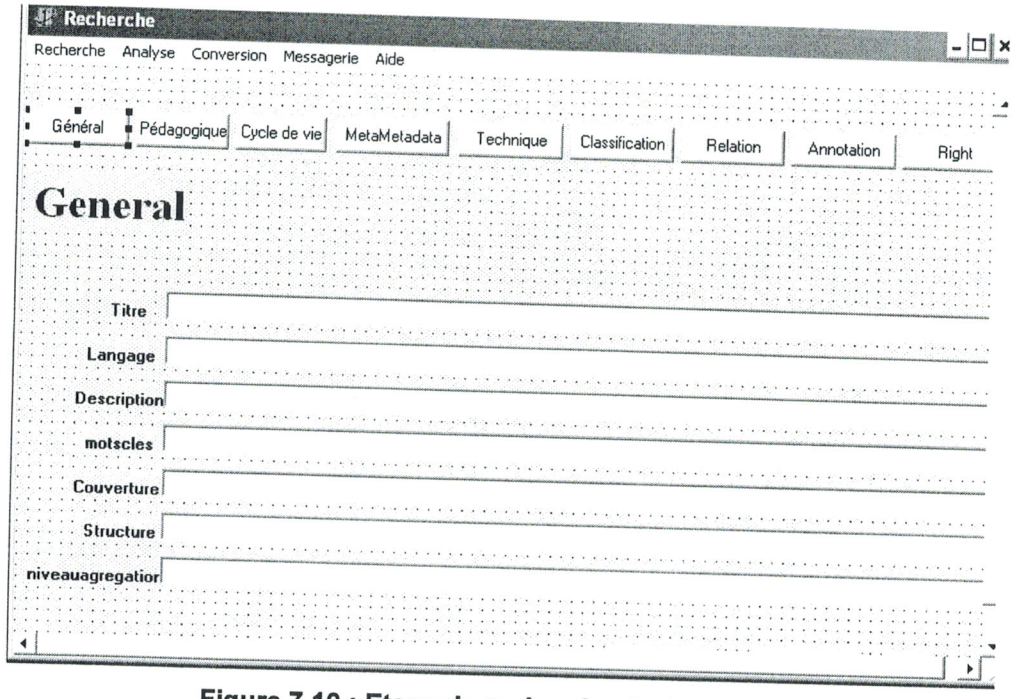


Figure 7.10 : Etape de recherche de documents

A chaque accès d'un utilisateur à l'un des entrepôts, une table existe au niveau du système et qui est gérée par une soumission. Cette table va contenir pour chaque entrepôt, l'ensemble des accès, les documents les plus demandés, ... , ce qui nous permettra après d'évaluer nos entrepôts et de les classer.

Comme réponse à chaque requête demandée, on aura un écran qui va contenir les documents adaptés à notre recherche.

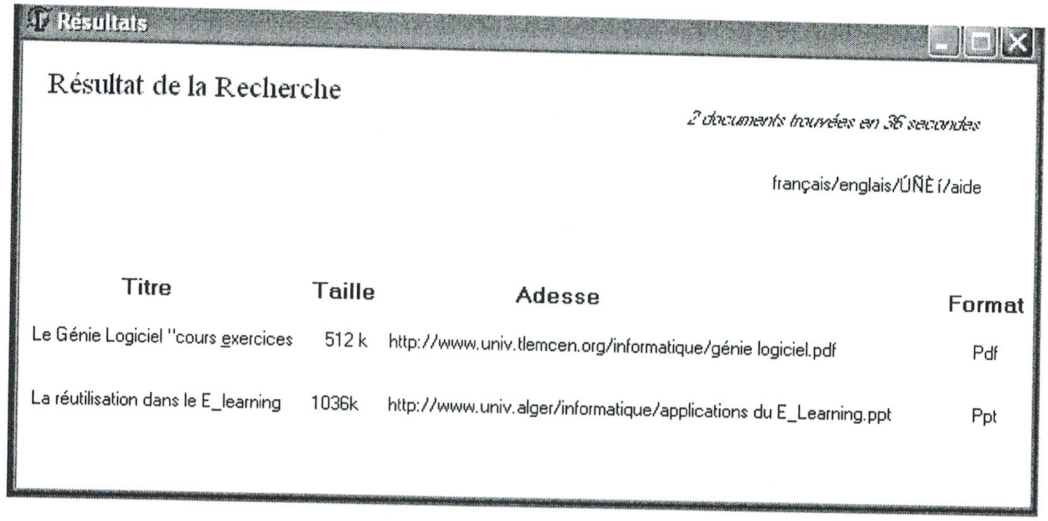


Figure 7.11 : résultat de la recherche

REFERENCES BIBLIOGRAPHIQUES

[JOBE 03] Mlle JOBERT Gabrielle Kahina , « L'apport du langage XML à la formation en ligne » DESS Application des Réseaux et de la Télématique 2002/2003 ;

[KHRO 05] Kais Khrouf , « Entrepôts de documents : de l'alimentation à l'exploitation », Thèse de doctorat ; Université Paul sabatier de Toulouse, 2005 ;

[PASSA 03] Brigitte de La Passardière*, Pierre Jarraud, « ManUeL, un profil d'application de LOM pour C@mpuSciences » ; Université Pierre et Marie Curie, LIP6 , 2003 ;

[QUIN94] Quint V., «Edition de documents structurés», INRIA Rhones-Alpes, Publié dans «le traitement électronique du document», ADBS éditions, p. 11-48, Paris, Octobre 1994 ;

[RANW 00] Sylvie Chabert-Ranwez , « Composition Automatique de Documents Hypermédia Adaptatifs à partir d'Ontologies et de Requêtes Intentionnelles de l'Utilisateur » ; thèse de doctorat ; UNIVERSITE DE MONTPELLIER II, SCIENCES ET TECHNIQUES DU LANGUEDOC ; décembre 2000 ;

[REGI 03] Rosa María Gómez de Regil, « Étude pour la mise en place d'un entrepôt d'objets pédagogiques à l'INSA de Lyon », Rapport de stage | - INSA de Lyon, 2003 ;

[ROUS 02] Daviid ROUSSE, « Entrepôt de données », Uniiverrsiitté Paull Sabattiierr,, TOULOUSE 2001-2002 ;

[SIRO 04] Jean pierre Sirot , « La valorisation de l'information avec Xyleme », Juin 2004 ;

[VANO0] C.Vanoirbeek , « LES MODÈLES DE DOCUMENTS », EPFL Groupe Media – 2000 ;

[VILLE 03] F.-Y. VILLEMIN, « Entrepôts de données (data warehouses) », CNAM-CEDRIC, 2003 ;

[XYLE 01] Lucie Xyleme, « A dynamic warehouse for XML data of the Web", March 2001;

[BOUR 00] Yolaine Bourda et Marc Hélier, « MÉTADONNÉES ET XML: APPLICATIONS AUX OBJETS PÉDAGOGIQUES », Plateau de Moulon, F-91192 Gif-sur-Yvette cedex France, 2000 ;

[BOUR 01] Yolaine Bourda ; « Objets pédagogiques, vous avez dit objets pédagogiques ? », *Supélec, Plateau de Moulon, F91192, GifsurYvette, CEDEX,* Mai 2001 ;

[CAST 99] M. Hervé Chastel, « Les Systèmes Décisionnels », Rapport du Projet système d'information, année 1999;

[CHIKH 04] CHIKH Azzeddine, «La réutilisation en ingénierie de document Outils méthodologiques et logiciels Application : e-learning » ; Thèse de doctorat, Université Alger, juillet 2005 ;

[CHRI 02] Christine Michel, Soufiane Rouissi, « E-learning : normes et spécifications » *CEM-GRESIC, Université Michel de Montaigne Bordeaux 3,* Volume X – n° X/2002 ;

[DELE 00] Delestre N., Métadyne. « Un hypermédia adaptatif dynamique pour l'enseignement »., Thèse de Doctorat, Université de Rouen, 2000 ;

[DEVL 97] Devlin, B. "Data Warehouse from Architecture to Implementation. Addison Wesley Longman", inc.432 p. 1997;

[DHER 05] Catherine Dhérent, « Les métadonnées,à quoi ça sert ? », Journée d'information AFNOR CG 46, 7 juin 2005 ;

[DUPO94] Dupoirier G., «Technologie de la GED», Edition Hermès, Paris, 1994 ;

[GARD 00] Georges GARDARIN, *« Les « Datawebhouses » arrivent »,* Laboratoire PRiSM et e-XMLMedia 45 Avenue des Etats-Unis 78035 VERSAILLES georges.gardarin@e-xmlmedia.fr 2000 ;

[GOIT01] Yacouba GOITA, « Les applications de XML à la production d'objets pédagogiques interactifs », *DEA préparé au sein du Laboratoire CLIPS –IMAG, Université Jospeh Fourier,* 2001 ;

[GRIS 04] E. GRISLIN-LE STRUGEON, « Systèmes d'information décisionnels » , Université de Valenciennes, ISTV, 2004 ;

[JAOU 02] JAOUHARI ouafaa, « Système intégré WADI », Mémoire de projet de fin d'étude , Juin 2002 ;