AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur : ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite de ce travail expose à des poursuites pénales.

Contact : portail-publi@ut-capitole.fr

LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10
Loi n°92-597 du 1er juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992
http://www.cfcopies.com/V2/leg/leg-droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par Université Toulouse I Capitole
Discipline ou spécialité : Informatique

Présentée et soutenue par *Ines BEN MESSAOUD Ep DAMMAK*

Le *Vendredi 18 Décembre 2014*

Titre : *Approche de construction d'entrepôts de documents XML*

JURY

Rafik BOUAZIZ	*Professeur, FSEG – Sfax Tunisie*	*Président*
Jamel FEKI	*Maître de conférences habilité, FSEG – Sfax Tunisie*	*Directeur de thèse*
Gilles ZURFLUH	*Professeur, Université Toulouse I Capitole France*	*Directeur de thèse*
Yahya SLIMANI	*Professeur, ISAMM – Mannouba Tunisie*	*Rapporteur*
Fadila BENTAYEB	*Maître de conférences habilité, Université Lyon 2 France*	*Rapporteur*

Ecole doctorale : *Mathématiques, Informatique et Télécommunications de Toulouse*
Unité de recherche : *Institut de Recherche en Informatique de Toulouse*
Directeur(s) de Thèse : *Gilles ZURFLUH et Jamel FEKI*

APPROCHE DE CONSTRUCTION D'ENTREPOTS DE DOCUMENTS XML

DIRECTEURS DE THESE :
JAMEL FEKI : MAITRE DE CONFERENCES HABILITE A LA FACULTE DES SCIENCES ECONOMIQUES ET DE GESTION DE SFAX (FSEGS)
GILLES ZURFLUH : PROFESSEUR A L'UNIVERSITE TOULOUSE I CAPITOLE (UT1)

Résumé

Les entrepôts de données stockent une grande volumétrie de données et permettent de les analyser par des traitements analytiques en ligne ("OLAP : On Line Analytical Processing"). Néanmoins, des études récentes affirment que seuls 20% des données d'un système d'information sont numériques et peuvent être traitées par un système OLAP ; les 80% restants correspondent à des documents (rapports, articles, etc.). Comme le volume de ces documents en prolifération considérable au cours du temps, les décideurs n'arrivent plus à les explorer facilement, rapidement et efficacement. En conséquence, certains documents pertinents peuvent être ignorés alors que d'autres moins pertinents peuvent être retenus par intuition, ce qui dégrade la qualité des résultats. Dans cette thèse nous avons proposé une approche de construction du schéma de l'entrepôt de documents XML permettant d'offrir une vision globale et homogène pour un ensemble de documents initialement hétérogènes. Cette approche se compose de deux méthodes : une méthode d'unification des structures des documents XML et une seconde méthode de modélisation multidimensionnelle de ces documents.

La méthode d'unification permet de définir une structure commune pour décrire les documents XML hétérogènes et appartenant au même domaine. Elle comporte quatre étapes : i) la Représentation arborescente des structures des documents XML, ii) la Génération des arbres unifiés qui traite les synonymes et les acronymes en utilisant respectivement la base de données lexicale Wordnet et un dictionnaire des acronymes ; elle génère des arbres unifiés à partir des arbres issus de l'étape précédente et ceci en appliquant un ensemble de trois opérateurs que nous avons définis, iii) l'Approbation des arbres unifiés en tenant compte des besoins analytiques des décideurs, et iv) la Vérification des arbres qui garantit la bonne formation des arbres générés en vérifiant un ensemble de quatre contraintes que nous avons défini. Pour valider cette méthode d'unification, un outil logiciel baptisé ***USD*** (***U****nification of* ***S****tructures of XML* ***D****ocuments*) est développé.

La deuxième méthode, la modélisation multidimensionnelle, a pour but de concevoir semi-automatiquement le schéma du magasin de documents, selon le modèle multidimensionnel en galaxie, à partir d'une structure XML unifiée. Cette méthode s'articule autour de quatre étapes : *i)* le Prétraitement, il améliore la lisibilité conceptuelle de l'arbre en l'enrichissant par des cardinalités qui aident ultérieurement à l'extraction des éléments multidimensionnels, ii) la Génération du modèle en galaxie, il identifie les éléments de la galaxie moyennant dix règles que nous avons proposé, iii) l'Approbation donne la main au décideur/concepteur pour confirmer la galaxie

obtenue, et iv) la Vérification de la galaxie contrôle la validité syntaxique du modèle générée par rapport à un ensemble de onze contraintes de bonne formation. Afin de valider cette méthode, un outil nommé ***Galaxy-Gen*** (**Galaxy Gen**eration) est développé.

Pour évaluer notre approche d'entreposage de documents XML, nous avons réalisé un ensemble d'expérimentations sur deux corpus : *un corpus académique* défini manuellement et *un corpus médical*. Le premier corpus se compose de 20 documents XML, du domaine académique, décrits par quatre DTDs. Alors que le corpus médical est constitué de 1691 documents XML de la collection médicale *Clef 2007* et est décrit par trois DTDs.

Pour le corpus académique, l'application des étapes de la méthode d'unification a produit un arbre unifié après trois itérations. Après, nous avons traduit cet arbre en une DTD et nous avons constaté que les documents de ce corpus sont valides par rapport à la DTD unifiée. Cette vérification est réalisée via l'outil *XMLSpy*. Puis, la méthode de modélisation en galaxie nous a permis de générer un modèle en galaxie composé d'un nœud central, une dimension temporelle et trois autres dimensions nommées : *D-Article, D-References* et *D-Writer*.

En ce qui concerne le corpus médical, la méthode d'unification a généré un arbre unifié que nous avons traduit en une DTD et nous avons vérifié, avec l'outil *XMLSpy,* que les documents du corpus sont valides par rapport à cette DTD unifiée. Puis, la méthode de modélisation a produit un modèle en galaxie constitué de cinq dimensions compatibles reliées par un nœud.

Nous avons proposé un ensemble de requêtes décisionnelles pour les galaxies générées pour les corpus académique et médical, et nous avons utilisé les opérateurs multidimensionnels des galaxies définis dans la littérature pour exprimer ces requêtes. Nous avons constaté que les modèles générés permettent aux décideurs de définir des requêtes qui les aident dans le processus décisionnel.

Mots clés

Entrepôt de documents XML, Unification, Arbre unifié, Modélisation multidimensionnelle, Modèle en galaxie.

A mes parents, mon mari, mon frère et ma sœur

Qu'ils retrouvent ici mes sincères remerciements pour leurs encouragements et leur soutien dans tous les moments de cette thèse. Qu'ils soient assurés de mon amour et mon profond respect.

Remerciements

J'adresse mes remerciements les plus sincères à Messieurs Jamel FEKI Maître de conférences à la Faculté des Sciences Economiques et de Gestion de Sfax (FSEG-Sfax) et Gilles ZURFLUH Professeur à l'Université Toulouse I Capitole (UT1), pour avoir dirigé et encadré mes travaux de thèse, pour leurs confiances, leurs rigueur scientifique, ainsi que pour leurs conseils et leurs critiques constructives, pour leurs disponibilités et pour l'aide qu'ils m'ont accordé. Qu'ils soient assurés de ma profonde gratitude et de mon très grand respect.

Mes remerciements vont également à Messieurs Franck RAVAT, Olivier TESTE, Ronan TOURNIER et Kais KHROUF pour leurs conseils judicieux. Leurs remarques pertinentes et les nombreuses discussions que nous avons eues, ont contribué à améliorer la qualité de mes travaux présentés dans ce mémoire. Qu'ils soient assurés de ma reconnaissance pour leurs soutiens et leurs nombreux encouragements, ainsi que du plaisir que j'ai à travailler avec eux.

Je remercie très sincèrement Messieurs les rapporteurs : Monsieur Yahya SLIMANI et Madame Fadila BENTAYEB pour avoir accepté d'évaluer ce mémoire de thèse et pour leur participation au jury. Je tiens à remercier également monsieur le président Rafik BOUAZIZ pour tout l'intérêt qu'il a manifesté envers mon travail et pour l'honneur qu'il m'accorde en participant au jury.

Mes remerciements s'adressent à Monsieur Faiez GARGOURI le responsable du laboratoire Mir@cl pour m'avoir accueilli au sein de son laboratoire. Aussi, je remercie Messieurs Claude CHRISMENT et Gilles ZURFLUH, responsables de l'équipe Systèmes d'Informations Généralisées (SIG) du l'institut de recherche IRIT pour m'avoir accueilli au sein de leur équipe afin que je puisse mener à bien cette thèse.

Je tiens aussi à remercier tous les membres des deux laboratoires Mir@cl et IRIT pour leur accueil chaleureux et leur gentillesse. Je remercie mes amis pour leur présence, leur aide et leur collaboration. Mes remerciements vont aussi à l'ensemble du personnel de Mir@cl et de l'IRIT, pour leur disponibilité, leur aide généreuse et leur gentillesse.

Je remercie mes parents à qui je dédie cette thèse. Je leur suis très reconnaissante de leurs encouragements et de leur soutien sans limite au cours de ce si long cursus universitaire. J'espère rester un sujet de fierté à leurs yeux. J'ai également une pensée affectueuse envers mon époux Slim pour l'amour qu'il m'a toujours consenti, son soutien et sa patience. De plus, j'ai une pensée affectueuse envers mon petit Rayan. Également, je remercie mon frère Issam et ma sœur Sameh et son mari Wassim pour leurs encouragements. Parallèlement, je remercie mes grands-parents, mes oncles, mes tantes et ma belle-famille pour leurs encouragements.

Je ne peux pas conclure ces quelques lignes sans adresser mes remerciements les plus sincères à tous mes amis Salma, Jihen, Fatma, Hela, Wafa, Manel, Yesser, Nabil, Saïd et Mahdi. Vous étiez la bouffée d'oxygène qui me ressourçait dans les moments difficiles. Également, je remercie mes collègues de l'Institut Supérieur d'Informatique et de Multimédia de Sfax (ISIM-Sfax) et mes collègues de l'Institut Supérieur de Gestion de Gabès (ISG-Gabès).

Il serait trop long de toutes les nommer, mais je remercie chaleureusement toutes les personnes qui ont contribué de prêt ou de loin à l'aboutissement de ce travail.

Ines Ben Messaoud

Sommaire général

Liste des figures

Liste des tableaux

Introduction générale

Contexte

Au cours de la dernière décennie, les technologies des entrepôts de données et des traitements analytiques en ligne (« OLAP : On-Line Analytical Processing ») se sont imposées comme solutions quasiment incontournables pour aider les décideurs dans leurs processus décisionnels. En effet, ces technologies permettent l'analyse, en un temps opportun, de grands volumes de données que les organisations détiennent dans les bases de données transactionnelles (Pérez-Martínez, et al., 2008). L'examen de ces données a été un support important pour l'évaluation et l'analyse des performances des activités, ou encore des processus métiers ; ceci dans une perspective de prise de décisions permettant de prévoir et planifier, voire d'anticiper le futur. C'est dans cette perspective que nombreuses entreprises ont investi dans cette nouvelle technologie en développant des projets décisionnels.

Cependant, face à la progression rapide des technologies de l'information, les données numériques ne constituent plus les principales informations pertinentes à la prise de décisions. En fait, les documents représentent aussi une source importante de connaissances qui pourraient désormais jouer un rôle non négligeable pour le décisionnel : Par exemple, les emails de clients ainsi que les rapports de crédit sont enregistrés dans des documents XML et peuvent renfermer des éléments utiles à la prise de décisions. De ce fait, Inmon (Inmon, 1994) insiste sur l'importance des informations contextuelles pour interpréter les opérations analytiques et historiques des organisations. Il en résulte que l'intérêt des informations contenues dans les documents est devenu essentiel pour le système de pilotage. Ceci a suscité l'intérêt de nombreux chercheurs de la communauté d'entreposage de données (« *Data warehousing* ») à accorder une attention particulière au contenu des documents en proposant des solutions appropriées pour les analyser. Ainsi, afin d'exploiter efficacement et plus facilement les documents, les auteurs de (McCabe, et al., 2000) et (Sullivan, 2001) recommandent l'entreposage des documents, d'où le concept d'entrepôt de documents (« *Document Warehouse* »).

L'entreposage de documents est une tâche complexe et très peu maîtrisée en l'état actuel en raison de difficultés multiples : les documents à entreposer se présentent généralement dans des formats hétérogènes, comme par exemple HTML, XML, PDF ou même des textes

bruts. Même s'ils adoptent le même formalisme de représentation, leurs structures ne sont pas identiques ; par exemple, les documents XML suivent généralement un schéma propre. Particulièrement, nous nous intéressons à l'entreposage de documents XML (*« eXtensible Markup Language »*) dans la mesure où il s'agit d'un des formats les plus utilisés pour la représentation et l'échange des données sur le Web. Généralement, les documents XML sont décrits par des structures de type DTD (« *Document Type Definition* ») ou, sous une forme étendue : les XSD (« *XML Schema Definition* »). En conséquence, l'entreposage des documents nécessite des efforts d'*homogénéisation* en vue de leur *interrogation* par des traitements analytiques en ligne. L'homogénéisation vise à *unifier leurs structures* afin d'aboutir à une structure commune permettant une vue globale de ces documents. Quant à l'interrogation analytique en ligne (OLAP), elle nécessite une *modélisation multidimensionnelle* pour mettre en évidence « *ce qui est analysable* », communément appelé *Sujet d'analyse*, et les axes selon lesquels le sujet est analysable (i.e., Dimensions). Ces efforts de modélisation sont souvent réalisés de manière manuelle, nous comptons contribuer à les automatiser au moins partiellement afin d'assister le concepteur du système d'information décisionnel (SID).

Objectifs et proposition

Ce mémoire de thèse aborde la problématique d'entreposage de documents XML (« *XML Document Warehousing* »). Principalement, notre thèse traitera de la conception et s'étendra à l'interrogation des entrepôts de documents. Du fait que les documents XML sont généralement décrits par des structures hétérogènes, nous avons besoin de construire une structure commune pour les documents XML à entreposer. De même, une représentation multidimensionnelle des documents à entreposer s'avère indispensable ; elle permettrait de mettre en évidence les sujets et les axes d'analyses. Pour ce faire, nous proposons une approche qui s'articule autour de deux méthodes : *(i)* L'*Unification des structures des documents XML* et *(ii) La Modélisation multidimensionnelle des documents*.

La première méthode vise à élaborer une structure unifiée pour une collection de documents XML. Cette structure unifiée se fonde sur le formalisme des arbres ; elle génère un ou plusieurs arbres unifiés à partir des structures des documents XML appartenant à un même domaine ; ces arbres seront remis au concepteur en vue de leur approbation.

Quant à la deuxième méthode, elle permet d'élaborer un modèle conceptuel multidimensionnel en *galaxie* préparant ainsi à l'interrogation. En fait, un modèle en galaxie

est une variante du modèle en étoile qui convient mieux aux entrepôts de documents (Tournier, 2007).

Organisation du mémoire

Pour présenter nos travaux et le domaine dans lequel ils s'inscrivent, nous avons retenu pour ce rapport une organisation en six chapitres.

Dans le premier chapitre, nous définissons les concepts de base des systèmes d'information décisionnels (SID) et nous introduisons les deux types de structure des documents XML : DTD « *Document Type Definition* » et XSD « *XML Schema Definition* ». De plus, nous soulignons les différences entre entrepôts de données XML et entrepôts de documents XML.

Le deuxième chapitre étudie les travaux de la littérature les plus populaires en matière d'unification des structures des documents XML et de modélisation multidimensionnelle des documents. Aussi, il présente une synthèse des travaux d'unification et de modélisation ; de plus, il donne un aperçu de notre proposition pour la construction semi-automatique du schéma de l'entrepôt de documents.

Le chapitre 3 développe notre méthode d'unification des structures des documents XML. Cette méthode, composée de quatre étapes, génère la structure unifiée des documents sous forme d'arbre.

Dans le quatrième chapitre, nous détaillons la méthode semi-automatique de modélisation multidimensionnelle proposée. Cette méthode traduit le résultat de la méthode d'unification en un modèle multidimensionnel en galaxie. Elle comporte quatre étapes successives.

Le cinquième chapitre traite de l'implantation et détaille les fonctionnalités des deux outils développés : ***USD*** « ***U****nification of* ***S****tructures of XML* ***D****ocuments* » pour l'unification des structures des documents XML et ***Galaxy-Gen*** « ***Galaxy Gen****eration* » qui supporte la méthode de modélisation multidimensionnelle.

Le chapitre 6 décrit les résultats des expérimentations réalisées dans le cadre de l'unification des structures des documents XML et de la modélisation en galaxie des documents XML de deux corpus académique et médical.

Finalement, dans la conclusion générale, nous récapitulons l'ensemble de nos contributions et nous énonçons quelques perspectives.

CHAPITRE 1 : CONTEXTE ET PROBLEMATIQUE

Résumé du chapitre :
Ce chapitre expose le contexte de ce mémoire de thèse. Il introduit les concepts « Système d'information décisionnel », « Entrepôt de données » et « Magasin de données ». De plus, il présente les entrepôts de données XML et les entrepôts de documents XML. Il introduit également la problématique de ce mémoire de thèse.

Sommaire du chapitre 1

1.1. Introduction

De nos jours, le texte demeure le moyen le plus répandu pour communiquer. En fait, les données textuelles aident les décideurs à mieux comprendre l'évolution des activités de l'organisation au cours du temps. Par exemple : les e-mails, les lettres d'information sont enregistrés dans des documents. Généralement, les documents sont de divers formats. Parmi ces formats, XML constitue un format simple et flexible. Il joue un rôle important dans l'échange des données via le Web[1].

Des études récentes (Tseng, et al., 2006) affirment que seulement 20% de l'information décisionnelle peut être extraite à partir des bases de données conventionnelles (*i.e.*, des systèmes opérationnels), alors que les 80% restants se présentent sous forme de données non numériques (*i.e.*, des documents) et ne sont pas intégrés au sein du système d'information décisionnel. Vu la quantité de l'information qu'ils contiennent, les documents méritent d'être entreposés ; ainsi le concept d'*Entrepôt de documents* (« *Document Warehousing* ») est né.

Dans ce chapitre, nous présentons les concepts inhérents aux documents XML et aux entrepôts de documents. Tout d'abord, nous définissons les composants d'un système d'information décisionnel : l'entrepôt et le magasin de données. Ensuite, comme nous nous focalisons sur les documents XML, nous présenterons les deux types de documents XML : Documents XML orienté-données (« *data-centric document* ») et Documents XML orienté-documents (« *document-centric document* ») ; le premier type est pour décrire les documents bien structurés. Alors que le second type est pour les documents XML moins structurés. Ensuite, nous présentons les deux structures de documents : DTD (« *Document Type Definition* ») et XSD (« *XML Schema Definition* »). Ces deux structures permettent de décrire la grammaire selon laquelle un document XML est produit et est interprété. Notons qu'un XSD permet de définir de nouveaux types. Finalement, comme il existe deux types d'entrepôts XML : entrepôt de données XML et entrepôt de documents XML, nous illustrons chaque type par son architecture. Le premier type d'entrepôt permet l'entreposage des documents XML orienté-données. Alors que le deuxième type est relatif aux documents XML orienté-documents.

[1] Http://www.w3.org/xml/, W3C-XML Extensible Markup Language (XML) 1.0.

1.2. Système d'information décisionnel

Un système d'information décisionnel (SID) offre aux décideurs une vision globale et transversale des informations qui circulent au sein de leur organisation. Il a pour objectif d'aider les décideurs dans le processus décisionnel. Un SID est composé de l'ensemble de matériels et logiciels informatiques permettant l'analyse des données provenant du système d'information opérationnel (SI) des organisations. Les données du SI sont extraites, préparées et transformées en un format compréhensible par les décideurs. La *Figure 1* illustre l'architecture du système d'information décisionnel.

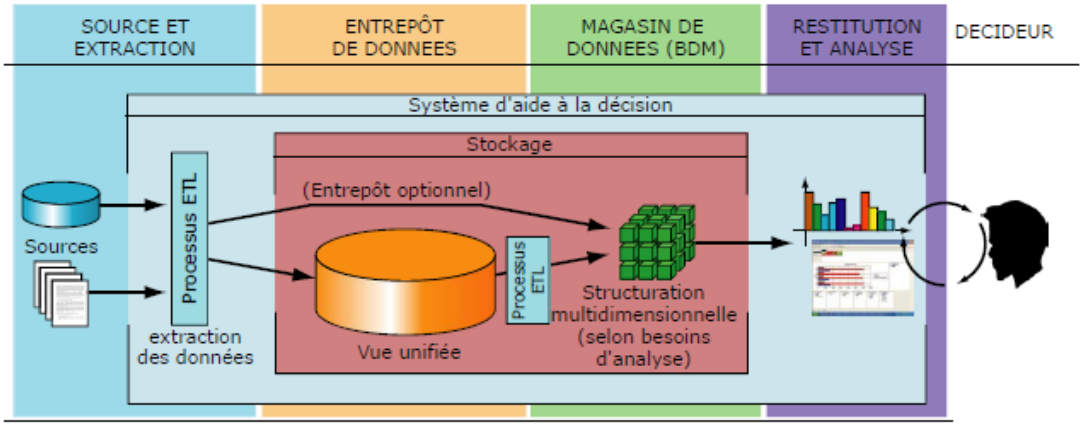


Figure 1 : *Architecture du système d'information décisionnel* **(Tournier, 2007)**.

Cette architecture comporte quatre niveaux :

- *Extraction des données* : Elle résout les problèmes liés à l'hétérogénéité et à la distribution des sources pour intégrer les données hétérogènes. De plus, elle extrait les données pertinentes pour la prise de décisions afin de générer l'entrepôt de données.
- *Entrepôt de données* : C'est le premier espace de stockage du système d'information décisionnel. Il est organisé selon un modèle facilitant la gestion des données.
- *Magasin de données* : C'est le second espace de stockage du système d'information décisionnel. Il correspond à un extrait des données de l'entrepôt destiné à une classe d'utilisateurs, et structure ces données via une modélisation spécifique permettant des analyses multidimensionnelles.
- *Restitution et analyse* : C'est l'exploitation des données contenues dans un magasin afin de restituer au décideur des résultats, généralement agrégés, à travers des outils spécifiques d'analyse.

1.2.1. Entrepôt de données

L'entrepôt de données est le lieu de stockage des données issues d'un ou de plusieurs systèmes d'information opérationnels. Inmon (Inmon, 1994) le définit comme : « *Une collection de données orientées sujet, intégrées, non volatiles, historisées, organisées pour le support d'un processus d'aide à la décision* ».

- *Orientées sujet* : Les données de l'entrepôt proviennent de différents services de l'entreprise. Elles sont organisées par sujet afin de permettre des analyses thématisées ; *i.e.,* sujet par sujet.
- *Intégrées* : Les données de l'entrepôt concernent les différents services de l'entreprise. Par conséquent, l'intégration de ces données dans l'entrepôt nécessite une bonne maîtrise de la sémantique des données.
- *Non volatiles* : Les données de l'entrepôt sont stables, c'est-à-dire qu'elles ne peuvent être ni supprimées ni modifiées puisqu'elles décrivent des résultats de traitements (*i.e.,* transactions) validés dans la source opérationnelle ; seules les opérations de rafraîchissement incrémental sont autorisées.
- *Historisées* : L'historisation des données permet le suivi de l'évolution des différentes valeurs des indicateurs dans le temps. En effet, les données d'un entrepôt traduisent l'activité d'une entreprise pendant une longue durée.
- *Organisées pour le support d'un processus d'aide à la décision* : Les données de l'entrepôt doivent être présentées au décideur d'une manière qui lui facilite leur exploitation sans qu'il soit connaisseur des techniques informatiques, et sans efforts de programmation. Pour ce faire, la modélisation multidimensionnelle est alors préconisée.

1.2.2. Magasin de données

Le magasin de données est un sous-ensemble de l'entrepôt ; il est dédié à des besoins d'analyse particuliers et organisé selon un modèle multidimensionnel mettant en évidence le sujet et les axes d'analyses. Généralement, il est modélisé sous forme d'un modèle en étoile ou en flocon. La *Figure 2* illustre un exemple de modèle en étoile permettant les analyses des montants (*Montant*) et le nombre de jours (*Nb_Jours*) de location des véhicules en fonction des quatre dimensions : *Agence*, *Client*, *Temps* et *Véhicule*.

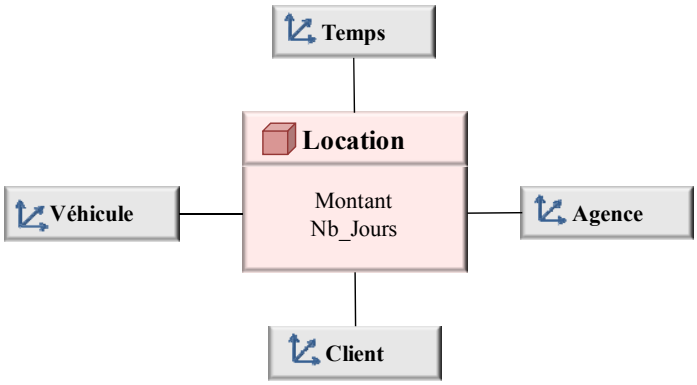


Figure 2 : *Exemple d'un modèle en étoile.*

1.3. Les documents XML

XML « *eXtensible Markup Language* » est un langage de balises permettant d'organiser les documents d'une manière structurée ; ces balises forment la structure hiérarchique du document XML. XML joue un rôle important dans l'échange de données sur le Web et se caractérise par une flexibilité lui permettant le stockage de données de diverses natures.

Il existe deux types de documents XML, nous présentons, dans ce qui suit, ces types ainsi que les formalismes utilisés pour décrire la structure de ces types.

1.3.1. Les types des documents XML

Il existe deux types de documents XML (Fuhr, et al., 2001) (Kamps, et al., 2004) :

Document XML orienté-données (« *data-centric document* ») : Ce type de document est composé de données majoritairement numériques et fortement structurées (*e.g.*, Commande, Facture). Il est utilisé par les applications du e-commerce. La *Figure 3* illustre un exemple d'un document XML orienté-données décrivant les données d'une facture.

Document XML orienté-documents (« *document-centric document* ») : Il est décrit par des structures hétérogènes et est composé principalement de texte. La *Figure 4* montre un exemple de document XML orienté-documents décrivant un article de recherche.

Bien que ces deux types de documents XML (*i.e.*, documents XML orienté-données et orienté-documents) permettent la représentation des données, néanmoins ils diffèrent par l'importance de l'ordre de leurs éléments. En effet, cet ordre dans un document XML orienté-documents est crucial pour sa compréhension (*e.g.*, séquencement des paragraphes dans un texte). Alors que, dans un document XML orienté-données, l'ordre des éléments n'est pas

significatif (*e.g.*, les données du deuxième produit peuvent précéder celles du premier produit).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Facture Numéro="F112">
  <Client>
    <Prénom>Michel</Prénom>
    <Nom>Dupont</Nom>
    <Adresse>
      <Ville>Paris</Ville>
      <Pays>France</Pays>
    </Adresse>
    <Téléphone>0144572972</Téléphone>
    <Date>15/03/2013</Date>
  </Client>
  <Contenu>
    <Produit Référence="P199">
      <Désignation>PC Portable</Désignation>
      <Prix_Unitaire>500</Prix_Unitaire>
      <Quantité>1</Quantité>
      <Montant>500</Montant>
    </Produit>
    <Produit Référence="P50">
      <Désignation>Souris</Désignation>
      <Prix_Unitaire>6</Prix_Unitaire>
      <Quantité>1</Quantité>
      <Montant>6</Montant>
    </Produit>
    <Total_H.T>506</Total_H.T>
    <TVA>19,6</TVA>
    <Total_T.T.C>605,176</Total_T.T.C>
  </Contenu>
</Facture>
```

Figure 3 : *Exemple d'un document XML orienté-données (Facture).*

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Article>
   <Titre> Entrepôt de données </Titre>
   <Auteur>
      <Nom> Claude Chrisment</Nom>
      <Affiliation> IRIT</Affiliation>
   </Auteur>
   <Auteur>
      ……
   </Auteur>
      ……
   <Section>
      <Titre> Bases de données décisionnelles </Titre>
      <Paragraphe>
        La gestion et le pilotage des entreprises, dans le cadre d'une mondialisation
        croissante de l'économie, nécessitent des systèmes d'information performants. Les
        décideurs, quel que soit leur niveau de responsabilité, doivent pouvoir accéder aux
        informations qui leur sont utiles le plus rapidement possible. Les entrepôts de
        données tentent de répondre à cette nécessité.
      </Paragraphe>
      <Paragraphe>
        ……
      </Paragraphe>
        ……
    </Section>
      ……
    </Section>
   ……
</Article>
```

Figure 4 : *Exemple d'un document XML orienté-documents (Article de recherche).*

1.3.2. Les structures des documents XML

Le langage XML permet le stockage des données selon un formalisme auto descriptif conformément à une *structure* (*i.e.*, grammaire) appelée DTD (« *Document Type Definition* ») ou XSD (« *XML Schema Definition* ») dite aussi schéma XML ou simplement *Xschema*.

Une DTD est une grammaire décrivant les règles selon lesquelles un document est produit. Elle est aussi utile pour la vérification de la conformité du document XML par rapport à cette grammaire. Une DTD est constituée d'éléments et d'attributs ; ces éléments peuvent imbriquer d'autres éléments. Les attributs définissent des informations supplémentaires sur un élément de la DTD. Ils constituent une paire (nom, valeur) représentant une propriété de l'élément. Notons que des cardinalités peuvent être imposées sur les éléments de la DTD. Ces cardinalités décrivent le nombre d'occurrences des balises (des éléments) dans le document XML. La *Figure 5* montre une DTD décrivant la structure de l'article de recherche de la *Figure 4*.

```
<!ELEMENT Article (Titre, Auteur+, Section+)>
<!ELEMENT Auteur (Nom, Affiliation)>
<!ELEMENT Section (Titre, Paragraphe)>
<!ELEMENT Titre (#PCDATA)>
<!ELEMENT Paragraphe (#PCDATA)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Affiliation (#PCDATA)>
```

Figure 5 : *Exemple d'une DTD.*

Un XSD est un standard pour la description des structures des documents XML. Il définit le type de contenu, la syntaxe et la sémantique d'un document XML. Aussi, un XSD est utilisé pour valider un document XML. La *Figure 6* est un exemple d'un XSD régissant les documents XML relatifs à des fiches client.

En examinant les deux structures des documents XML, nous constatons qu'un XSD propose des nouveautés en plus des fonctionnalités fournies par une DTD. En effet, un XSD est plus complexe et plus expressif qu'une DTD. Il donne la possibilité de créer de nouveaux types à partir de types existants. De plus, il donne la main pour préciser le nombre d'occurrences exacte d'un élément ; les indicateurs d'occurrences des éléments peuvent être tout nombre non négatif (dans une DTD, les indicateurs d'occurrences sont limités à 0,1 ou un nombre infini * ou +).

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Client">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nom" type="xs:string" />
            <xs:element name="prenom" type="xs:string" />
            <xs:element name="date_naissance" type="xs:date" />
            <xs:element name="adresse" type="xs:string" maxOccurs=2 />
            <xs:element name="num_tel" type="xs:string" />
            <xs:element name="num_fax" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Figure 6 : *Exemple d'un XSD pour les fiches client.*

1.4. Les entrepôts XML

La divergence des structures de données a donné naissance à deux types d'entrepôts : *Entrepôt de données* et *Entrepôt de contenu*. Les *entrepôts de données* s'intéressent aux données transactionnelles. Tandis que, l'entrepôt de contenu archive un grand volume de données textuelles issues du Web : c'est l'*entrepôt Web*. Il est divisé en deux sous-types d'entrepôts : *entrepôt de textes* lorsqu'il existe peu de structures de documents disponibles et *entrepôt de documents* dans le cas contraire.

L'intégration des documents XML dans l'entrepôt produit deux types d'entrepôts : *Entrepôt de données* permettant d'intégrer des documents XML orienté-données et *Entrepôt de documents* permettant d'intégrer des documents XML orienté-documents.

De l'autre côté, l'entreposage des documents XML distingue deux types d'entrepôts : *Entrepôt de données XML* (Il permet l'entreposage des documents XML orienté-données (*cf.* section 1.4.1)) et *Entrepôt de documents XML* (Il permet l'entreposage des documents XML orienté-document (*cf.* section 1.4.2)).

Dans le contexte de l'entrepôt XML, l'entrepôt permet des analyses OLAP non seulement des documents orienté-données, mais aussi des documents orienté-documents (Ravat, et al., 2010).

La *Figure 7* illustre l'évolution de l'utilisation de la technologie XML avec les approches d'entreposages.

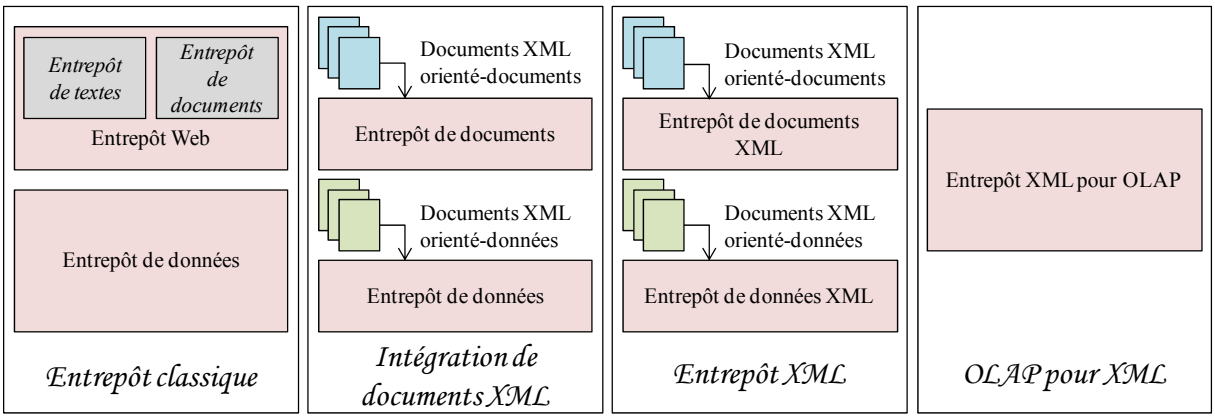


Figure 7 : *Progression de l'usage de XML avec les approches d'entreposage* **(Ravat, et al., 2010)**.

Dans ce travail, nous nous sommes intéressés aux entrepôts XML pour lesquels nous étudions les différents types.

1.4.1. Les entrepôts de données XML

L'intégration des documents XML *orienté-données* dans un entrepôt XML produit un entrepôt de données XML. Dans la littérature, il existe deux catégories d'approches d'intégration : *physique* et *logique*.

Intégration physique : Les données sources des documents XML sont stockées dans l'entrepôt.

Intégration logique : Elle ne conserve pas les documents XML dans l'entrepôt.

Khrouf (Khrouf, 2004) définit l'entrepôt de données XML comme un espace de stockage des documents XML structurés (*i.e.,* document XML orienté-données). Cet entrepôt ressemble à l'entrepôt de données classique du fait que ces deux se partagent les mêmes caractéristiques :

- données orientées sujet,
- données intégrées,
- données historisées, et
- données organisées pour le support d'un processus d'aide à la décision.

La *Figure 8* illustre l'architecture d'un entrepôt de données XML.

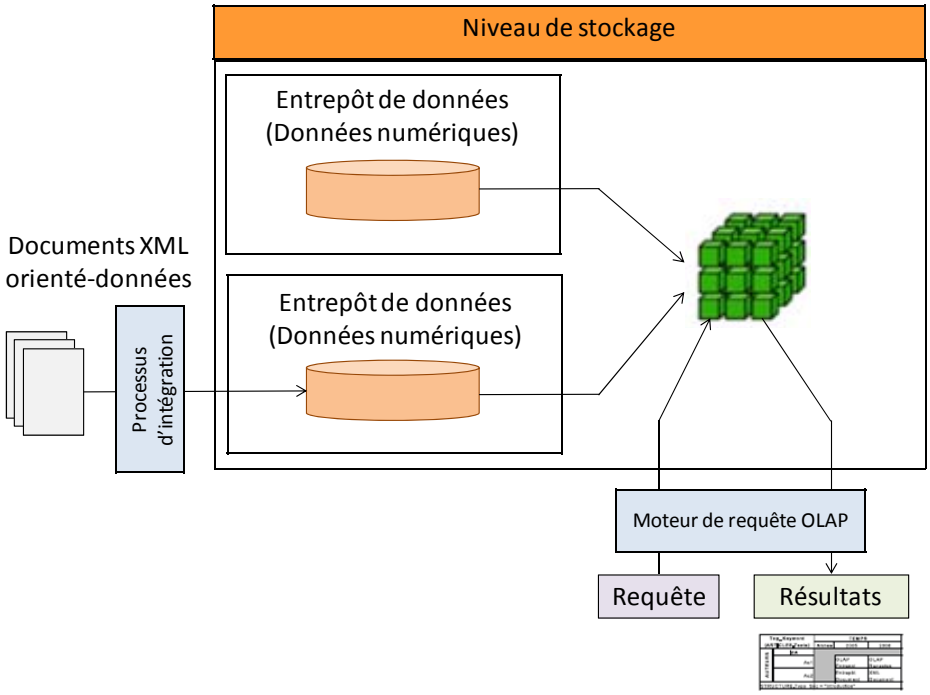


Figure 8 : *Architecture d'un entrepôt de données XML (intégration logique)* **(Ravat, et al., 2010)**.

1.4.2. Les entrepôts de documents XML

Un entrepôt de documents XML fournit un environnement de stockage des données peu structurées : c'est le lieu d'entreposage des documents XML orienté-documents issus des sources de données externes et internes. Il permet d'organiser les données pour une analyse efficace afin de permettre une intelligence économique (« Business intelligence ») réussie (Tseng, et al., 2006). Ses préoccupations majeures sont : *(i)* le stockage uniforme des données, et *(ii)* la restitution des fragments de textes jugés pertinents par l'utilisateur (Tournier, 2007).

Il existe deux catégories d'approches pour l'utilisation d'un document XML orienté-documents au sein de l'entrepôt de documents (Ravat, et al., 2010) :

- *Contextualisation de l'entrepôt de données avec les documents XML orienté-documents, et*
- *Construction de magasins de documents à partir des métadonnées des documents XML orienté-documents.*

1.4.2.1. Contextualisation de l'entrepôt de données avec les documents XML

L'objectif de la contextualisation des entrepôts est de fournir au décideur un complément d'information pour ses analyses. Dans ce cadre, nous trouvons les travaux de (Pérez-Martinez, 2007) et (Pérez-Martínez, et al., 2008) où ils proposent d'intégrer d'une part l'entrepôt de données et, d'autre part, l'entrepôt de documents XML. L'entrepôt résultat est un *entrepôt contextualisé* : c'est un système d'aide à la décision combinant des sources de données structurées et non structurées et permettant d'analyser les données dans des contextes différents. Il permet de relier chaque fait de l'entrepôt de données à son contexte.

La *Figure 9* présente l'architecture d'un entrepôt contextualisé.

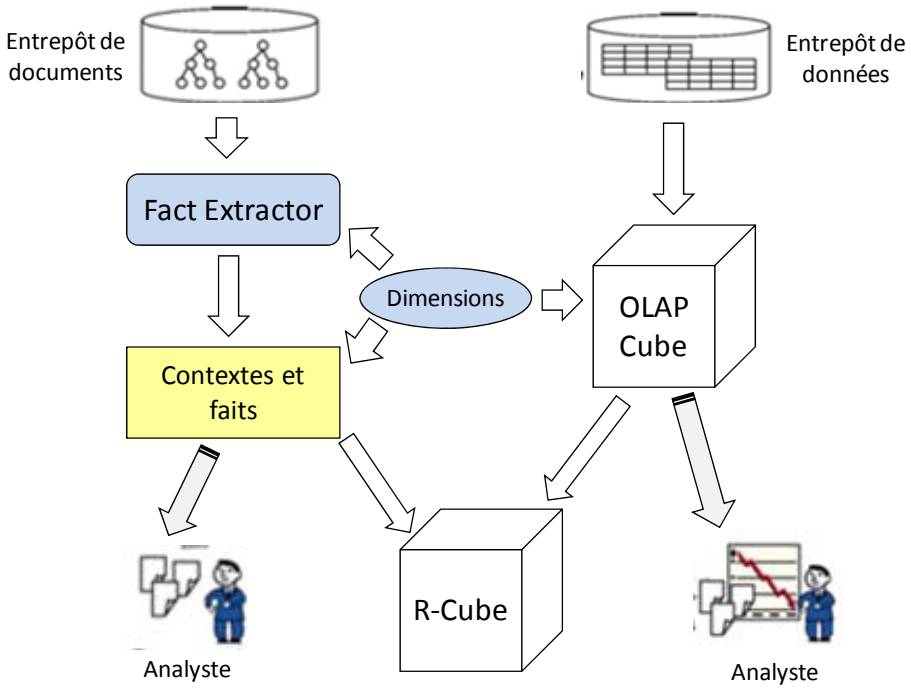


Figure 9 : *Architecture d'un entrepôt contextualisé* **(Pérez-Martínez, et al., 2008)**.

Les composants de cette architecture sont :

- *L'entrepôt de données* : stocke les données structurées.
- *L'entrepôt de documents* : stocke les données non structurées (*e.g.,* les documents XML) issues des sources externes et internes.
- *Le module "Fact Extractor"* : relie les faits de l'entrepôt de données avec les documents qui décrivent son contexte.
- *Le cube OLAP-Cube* : permet la représentation abstraite des données multidimensionnelles numériques.

- *Le cube R-Cube* : relie chaque fait avec l'ensemble des documents importants décrivant son contexte, et est caractérisé par deux nouvelles dimensions :
  - Pertinence : utilisée pour explorer les parties du cube les plus pertinentes puisqu'elle représente l'importance de chaque fait dans le contexte sélectionné.
  - Contexte : présente les documents de l'entrepôt détaillant le contexte du fait.

1.4.2.2. Construction de magasin de documents à partir des métadonnées des documents

Dans cette catégorie, les métadonnées décrites dans les documents orienté-documents issus de l'entrepôt sont extraites pour construire le magasin. La *Figure 10* présente l'architecture d'analyse des métadonnées des documents orienté-documents.

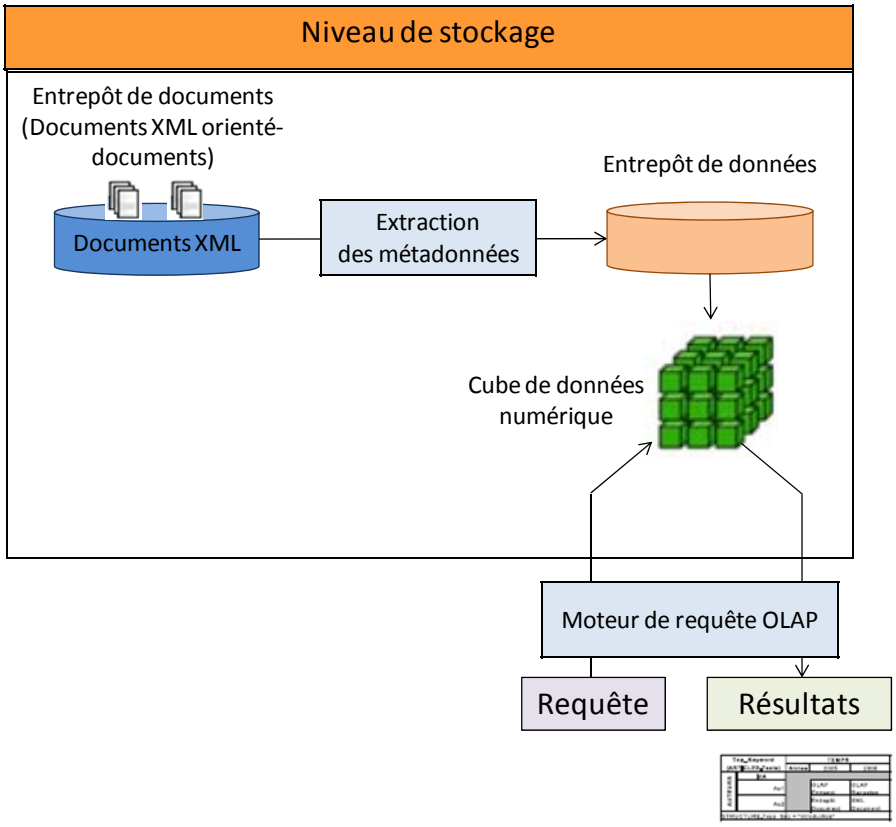


Figure 10 : *Architecture d'analyse des métadonnées des documents XML orienté-documents* **(Ravat, et al., 2010)**.

Dans ce contexte, (Khrouf, 2004) propose, d'une part, de grouper les structures des documents pour alimenter l'entrepôt et, d'autre part, d'analyser la structure et les métadonnées des documents pour exploiter les données de l'entrepôt.

Dans (Tseng, et al., 2006), les auteurs utilisent le modèle en étoile pour analyser les métadonnées des documents extraites par référence au *Dublin Core Metadata Initiative*[2]. Cette analyse permet à l'utilisateur de consulter les documents pertinents.

1.5. Problématique

Les documents constituent une capitalisation importante de connaissances dans un système d'information opérationnel. De plus, ils sont utiles pour les acteurs du système de pilotage. Généralement, ces documents sont caractérisés par un contenu peu structuré et il est difficile de les intégrer dans les systèmes d'information décisionnels des organisations (Tournier, 2007). Par conséquent, lors d'une prise de décisions, les analystes-décideurs n'arrivent pas à accéder et manipuler facilement, rapidement et efficacement ces documents. Cette situation risque d'entraîner l'oubli de certaines informations pertinentes ou même l'utilisation de données non pertinentes au cours du processus décisionnel. Le résultat pourrait alors conduire à des décisions inadaptées (Tseng, et al., 2006).

Afin de permettre des analyses plus consistantes, le système d'aide à la décision doit autoriser l'exploitation de l'ensemble des données (numériques et textuelles) du système d'information de l'entreprise et, au-delà, des données de la toile.

Dans le cadre de ce mémoire de thèse, nous proposons des modèles et outils qui permettent aux décideurs d'entreposer et d'analyser des documents XML orienté-documents. Ces documents sont décrits par des structures hétérogènes qui conduisent à écrire des requêtes multiples pour les manipuler. Nous proposons une solution pour résoudre ce problème.

Ainsi, notre problématique s'articule autour de l'entreposage de documents XML. Elle aborde les points suivants :

- Comment construire une description unifiée d'un ensemble de documents XML hétérogènes, *i.e.,* une vue globale ?
- Vu que les documents XML appartenant à un même domaine sont généralement décrits par des structures hétérogènes, alors une structure commune de ces documents est indispensable. Elle cache l'hétérogénéité structurelle de ces documents.
- Comment modéliser ces documents en vue d'être exploités par un processus décisionnel ?

[2] Http://www.dublincore.org/: C'est un vocabulaire comportant un ensemble de métadonnées ou de propriétés pour décrire un document.

- Lors d'une prise de décisions, le décideur a besoin d'un modèle multidimensionnel pour exprimer ses besoins décisionnels. De ce fait, les documents méritent d'être modélisés pour les exploiter, plus tard, dans un processus de prise de décisions.

1.6. Conclusion

Les documents représentent une source importante des informations et des connaissances (Tournier, 2007). Ils sont souvent présentés sous format XML et contiennent des données qui aident les décideurs dans le processus décisionnel. De ce fait, ces documents méritent d'être entreposés. Dans ce chapitre, nous avons présenté les deux types de documents XML orienté-données (« *Data-centric document* ») et orienté-documents (« *Document-centric document* ») ainsi que les deux structures DTD (« *Data Type Definition* ») et XSD (« *XML Schema Definition* »). Nous avons également exposé le concept d'entrepôt XML permettant de stocker les documents XML.

La construction d'un entrepôt de documents XML est une tâche délicate surtout que, généralement, ces documents possèdent des structures hétérogènes même au sein d'un même domaine. De ce fait, l'unification de ces structures est nécessaire pour masquer leur hétérogénéité. De même, la modélisation des documents est essentielle afin de présenter les documents sous forme de modèle compréhensible par les preneurs de décision.

Nous nous intéressons dans le chapitre suivant à présenter les travaux relatifs à l'unification des structures des documents XML et à la modélisation multidimensionnelle des documents. Aussi, nous présentons un aperçu de l'approche que nous proposons pour la construction du schéma de l'entrepôt de documents.

CHAPITRE 2 : ÉTAT DE L'ART

Résumé du chapitre :
Ce chapitre présente, d'une part, les travaux de la littérature les plus pertinents traitant l'unification des DTDs et des XSDs et, d'autre part, les travaux relatifs à la modélisation multidimensionnelle des documents. Il expose parallèlement une étude comparative de ces travaux. De même, il donne un aperçu de notre approche proposée pour la construction du schéma de l'entrepôt de documents XML.

Sommaire du chapitre 2

2.1. Introduction

Rappelons que les documents XML appartenant à un même domaine d'activité et manipulés au sein d'une même organisation (*e.g.,* les articles de recherche scientifiques) sont décrits par des structures généralement hétérogènes. Par ailleurs, l'exploitation de ces documents dans un processus décisionnel nécessite leur homogénéisation en vu de leur entreposage. Pour mettre en œuvre cet entreposage, deux tâches principales s'avèrent indispensables : tout d'abord, résoudre le problème d'hétérogénéité par l'élaboration d'une structure unifiée (*i.e.,* commune) pour ces documents, ce qui permettra de les décrire et de les voir d'une manière homogène ; ensuite, dériver à partir de cette structure unifiée une représentation permettant d'exploiter ces documents dans un processus décisionnel, ceci peut se faire par le passage vers un modèle multidimensionnel afin de profiter des outils logiciels d'implantation et de manipulation OLAP disponibles sur le marché. Ce modèle se veut compréhensible par les décideurs et d'usage simple.

Ce chapitre s'intéresse aux travaux de l'état de l'art relatifs à la problématique d'entreposage des documents. Nous y présentons, dans la section 2, les travaux les plus pertinents touchant l'unification des structures des documents XML. Puis, une synthèse de ces travaux sera donnée dans la section 3. Quant à la section 4, elle étudie les travaux les plus pertinents traitant la modélisation multidimensionnelle des documents XML. La section 5 présente une synthèse de ces travaux. Finalement, nous esquissons dans la section 6 un aperçu de notre méthode proposée pour la construction du schéma de l'entrepôt de documents XML.

2.2. Unification des structures des documents XML : État de l'art

La technique d'unification des structures des documents XML a pour objectif de créer une structure unifiée pour un ensemble de documents XML de structures hétérogènes tout en préservant les éléments de ces structures et en minimisant la perte de sémantique.

Dans cette section, nous présentons les travaux qui nous semblent les plus pertinents traitant le problème d'unification. Dans ces travaux, nous distinguons l'unification des DTDs (« *Document Type Definition* ») et l'unification des XSDs (« *XML Schema Definition* »).

2.2.1. LES TRAVAUX DE (LEE, ET AL., 2002)

Dans (Lee, et al., 2002), les auteurs proposent une stratégie d'intégration qui nécessite le regroupement ("clustering") des DTDs d'un ensemble de documents XML. Cette stratégie

reçoit en entrée un ensemble de DTDs et génère une DTD globale. Elle schématise les DTDs en entrée sous forme d'arbres. La *Figure 11* illustre cette stratégie d'intégration.

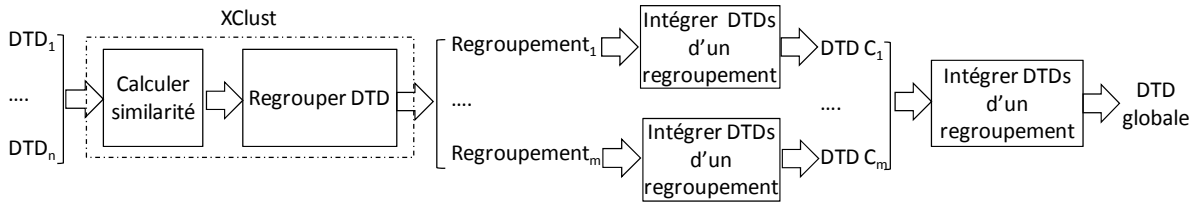


Figure 11 : *Stratégie d'intégration des DTDs* **(Lee, et al., 2002)**.

Dans cette stratégie, premièrement les degrés de similarité des DTDs en entrées sont calculés. Pour calculer la similarité entre deux arbres de DTD, un degré de similarité est calculé pour chaque paire de nœuds de ces deux arbres. Ce degré tient compte des informations linguistiques, structurelles et sémantiques des nœuds des arbres avec des poids différents définis par l'utilisateur. La similarité sémantique considère la similarité des noms des éléments, leur cardinalité et leur chemin. Elle utilise une table pour gérer les acronymes et la base de données lexicale *Wordnet* pour déterminer les synonymes. La similarité linguistique prend en considération les éléments descendants immédiats de l'élément d'une DTD, alors que la similarité structurelle traite le contexte d'élément feuille qui est défini par l'ensemble des nœuds dans le chemin de l'élément de la DTD vers l'élément feuille.

Deuxièmement, des regroupements de DTDs similaires seront formés en se basant sur les degrés de similarité calculés précédemment. Finalement, une DTD globale est générée à partir de chaque regroupement ("cluster") de DTDs.

Néanmoins, le calcul de similarité peut être coûteux en termes de temps. En effet, pour intégrer un nombre important de DTDs présentées par des arbres complexes, plusieurs degrés de similarité devront être calculés ; le calcul du degré de similarité entre deux arbres A1 et A2 nécessite le calcul d'autant de similarités que de combinaisons des nœuds de A1 par tous nœuds de A2 (|A1|x|A2|).

2.2.2. LES TRAVAUX DE (MELLO, ET AL., 2002)

Les auteurs de (Mello, et al., 2002) proposent une méthode d'unification de DTDs pour des documents XML sémantiquement équivalents qui génère une ontologie de référence. Cette méthode comporte deux étapes : *(i) Conversion des DTDs* en un schéma conceptuel canonique, et *(ii) Intégration des schémas* canoniques. La *Figure 12* trace l'enchaînement de ces étapes.

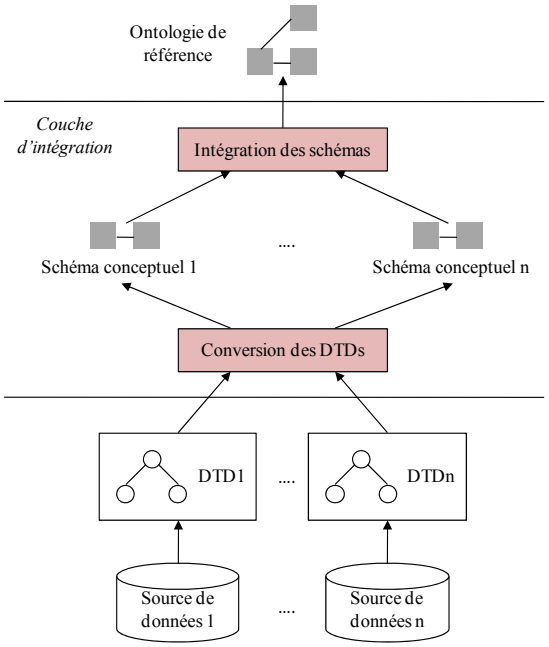


Figure 12 : *Processus d'intégration de DTDs* **(Mello, et al., 2002)**.

La première étape convertit chaque DTD de la source de données en entrée en un schéma conceptuel canonique basé sur le modèle ORM/NIAM « ***O**bject with **R**oles **M**odel / **N**atural language **I**nformation **A**nalysis **M**ethod* » et décrit par les notations graphiques du modèle entité-association. Ce schéma présente une abstraction conceptuelle des DTDs. Alors que, la deuxième étape (*i.e.*, Intégration des schémas) intègre les schémas conceptuels résultats de l'étape précédente. Cette intégration comporte cinq sous-étapes : *(i) Regroupement des concepts des schémas conceptuels basé sur les synonymes*, *(ii) Unification des concepts d'un même cluster*, *(iii) Regroupement des concepts de ces schémas basé sur les hyperonymes*[3], *(iv) Génération des relations d'héritage*, et *(v) Restructuration des relations entre les concepts du résultat*.

Tout d'abord, des regroupements de concepts synonymes sont déterminés, à partir des schémas conceptuels générés, grâce à l'outil ARTEMIS. Ensuite, l'unification gère les conflits de nommage et de relations entre les concepts du même regroupement. Elle génère un concept ontologique représentatif et les relations de "mapping" pour chaque regroupement en appliquant un ensemble de règles sur les concepts du regroupement. A la fin de cette sous-étape, une ontologie préliminaire est construite. Puis, des regroupements de concepts avec des relations d'hyperonymes sont déterminés avec ARTEMIS. Ensuite, ces regroupements seront utilisés pour déterminer les relations d'héritage dans l'ontologie préliminaire. Finalement, la

[3] C'est une relation qui caractérise des termes génériques dont le sens inclut celui d'autres termes spécifiques (*e.g.,* animal est l'hyperonyme de mammifère).

restructuration contrôle les relations entre les concepts en supprimant les relations redondantes pour générer une ontologie de référence pour le résultat d'unification. En fait, cette ontologie est présentée sous forme d'un schéma conceptuel. Elle agit comme un frontal des requêtes des utilisateurs exprimées sur les documents XML sources.

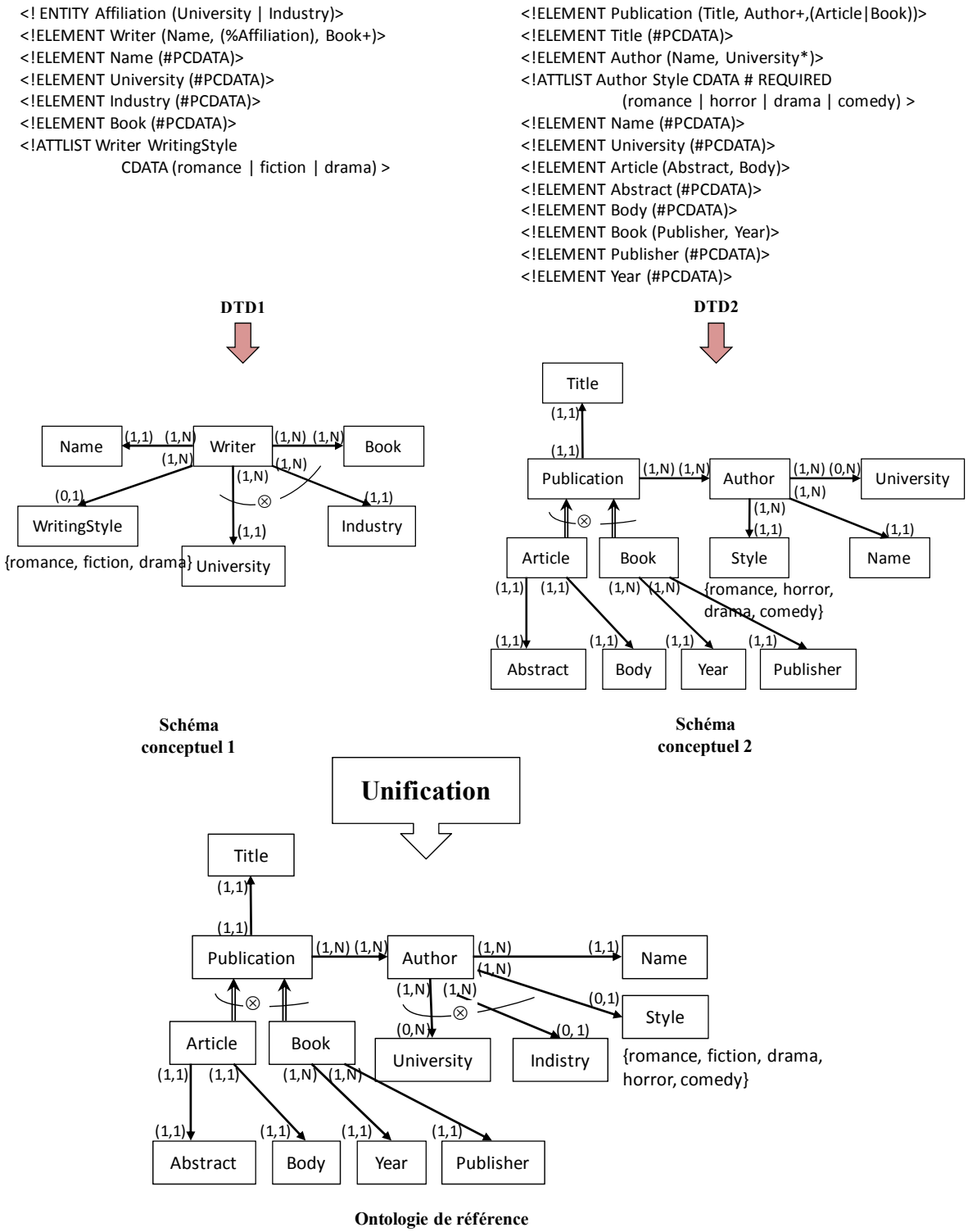


Figure 13 : *Exemple d'unification de deux DTDs* **(Mello, et al., 2002)**.

La *Figure 13* montre un exemple d'unification de deux DTDs : *DTD1* et *DTD2*. Ces deux DTDs sont transformées respectivement en *schéma conceptuel 1* et *schéma conceptuel 2* exprimés selon le modèle ORM/NIAM. Leur unification produit *l'ontologie de référence* illustrée en bas de la même *Figure 13*.

Toutefois, la méthode d'unification proposée ne traite pas les acronymes des noms des éléments et des attributs de la structure XML. Aussi, les règles d'unification proposées ne traitent pas les informations structurelles des concepts (*e.g.*, concept père du concept à unifier). Nous pouvons rencontrer dans deux schémas différents deux concepts distincts décrits par deux synonymes. Par conséquent, la qualité du résultat d'unification de ces concepts risque d'être détériorée.

2.2.3. LES TRAVAUX DE (YOO, ET AL., 2005)

Les auteurs de (Yoo, et al., 2005) présentent un algorithme pour l'unification des DTDs. Cet algorithme reçoit en entrée un ensemble de DTDs de documents appartenant à un même domaine et décrits par des structures similaires (*e.g.*, les DTDs des articles de revues scientifiques) et génère une DTD commune jouant le rôle d'un schéma conceptuel global. L'algorithme proposé comporte quatre étapes : *(i) Prétraitement des DTDs*, *(ii) Représentation des DTDs*, *(iii) Génération d'une DTD uniforme,* et *(iv) Post-traitement*. La *Figure 14* montre l'enchaînement de ces quatre étapes.

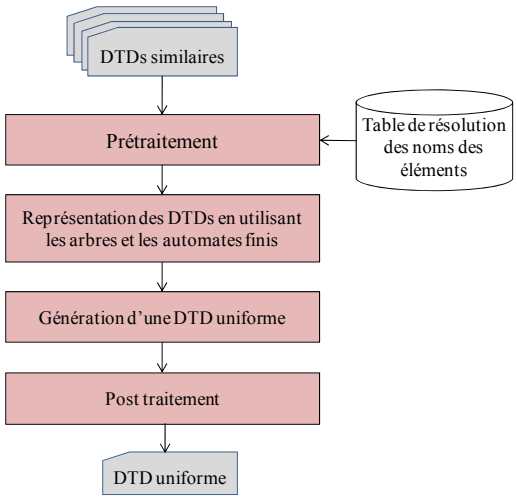


Figure 14 : *Étapes d'unification des DTDs* **(Yoo, et al., 2005)**.

Initialement, l'étape de prétraitement résout les ambiguïtés des noms des éléments des DTDs. Elle remplace les éléments synonymes des DTDs par un nom commun et ceci en se basant sur une table construite par un expert de domaine. Ensuite, la représentation des DTDs

traduit les DTDs sous forme d'arbres et d'automates finis[4]. En fait, l'arbre traduit la structure arborescente des documents XML. Tandis que l'automate fini représente les éléments et les connecteurs des DTDs. Puis, les arbres et les automates résultats de cette étape sont fusionnés pour générer une seule DTD unifiée représentée sous forme d'arbre et d'automate. Pour ce faire, les auteurs proposent un ensemble d'algorithmes. Finalement, l'étape de post-traitement transforme l'arbre et l'automate fini résultat en une DTD unifiée qui sera vérifiée syntaxiquement via un parseur.

Cependant, lors de l'étape de prétraitement les auteurs utilisent une table pour gérer les synonymes. Cette table risque d'être incomplète du fait que sa construction nécessite un échantillon bien représentatif des documents du domaine. Le risque de ne pas couvrir tous les synonymes des noms affecte la qualité du résultat d'unification. Cette étape pourrait être améliorée par le recours à une ontologie.

2.2.4. LES TRAVAUX DE (KHROUF, ET AL., 2003)

Dans le cadre de l'alimentation de l'entrepôt de documents avec des documents XML, les auteurs de (Khrouf, et al., 2003) proposent une méthode de comparaison et de fusion des DTDs. Cette méthode compare la structure arborescente du document à celles stockées dans le référentiel de l'entrepôt. Elle comporte les six étapes suivantes : *(i) Filtrage*, *(ii) Pondération*, *(iii) Conservation de l'ordre*, *(iv) Ajout d'éléments*, *(v) Calcul de similarité*, et *(vi) Décision finale.* La *Figure 15* illustre l'enchaînement de ces six étapes.

Initialement, un coefficient de filtrage est calculé par comparaison de la structure logique[5] d'un document en entrée avec toutes les structures logiques décrivant les documents de l'entrepôt. Ce coefficient sert à sélectionner les structures de l'entrepôt qui ressemblent à la structure du document. Chaque structure respectant le seuil subit, ultérieurement, une comparaison avec la structure du document en entrée. Puis, la *Pondération* fournit des poids aux nœuds de la structure du document en entrée. Ces poids tiennent compte de la profondeur et de l'ordre des fils d'un élément père. Ensuite, la *Conservation d'ordre* traite le placement des nœuds des structures à comparer, à savoir l'ordre des éléments d'un point de vue hiérarchique (*i.e.*, l'ordre des ancêtres) et l'ordre des éléments fils. En fait, uniquement les structures préservant l'ordre peuvent être étudiées à l'étape suivante (*i.e.*, étape d'ajout d'éléments). Puis, dans le but d'homogénéiser les deux structures (*i.e.*, celle du document en

[4] Un automate est constitué d'un ensemble d'états et de transitions. Il est dit fini s'il possède un nombre fini d'états.
[5] La structure logique décrit l'ensemble des informations structurées contenues dans un document.

entrée et de la structure de l'entrepôt), des éléments peuvent être ajoutés dans l'une des deux structures. Si les deux structures sont devenues identiques, alors le document est rattaché à la structure de l'entrepôt. Autrement, le *Calcul de similarité* est déclenché, il détermine un degré de similarité pour décider si les deux structures peuvent être fusionnées. Finalement, l'étape *Décision finale* permet de déterminer parmi les structures de l'entrepôt la structure qui mérite d'être fusionnée avec la structure du document en entrée et ceci en se basant sur un seuil.

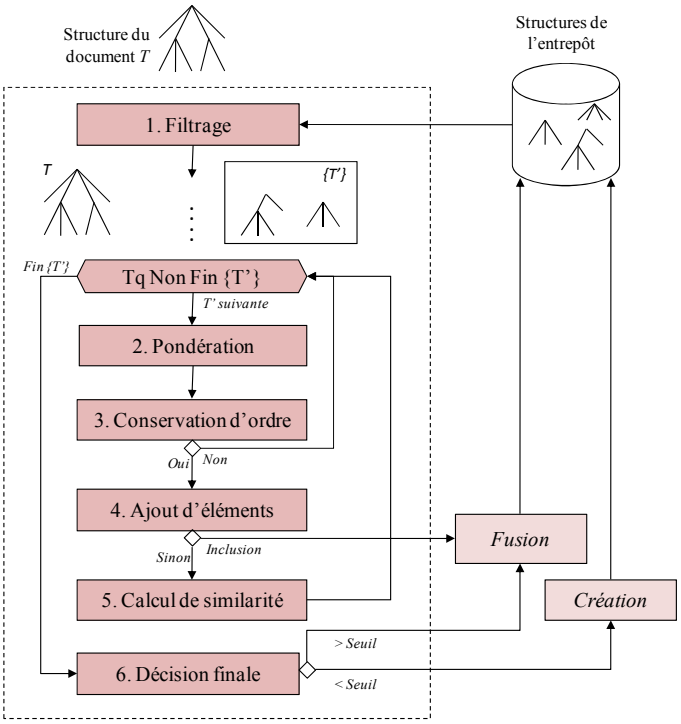


Figure 15 : *Etapes de comparaison de la structure d'un document avec les structures de l'entrepôt* **(Khrouf, et al., 2003)**.

Néanmoins, cette méthode compare deux structures et ne traite ni les synonymes ni les acronymes. Par conséquent, pour comparer plusieurs structures, il faut réitérer ce processus pour chaque paire de structures. De plus, pour fusionner les structures, l'auteur a proposé un ensemble de règles dont l'application exige un certain ordre des éléments des structures ce qui risque d'engendrer la génération de plusieurs structures.

2.2.5. LES TRAVAUX DE (ZHANG, ET AL., 2002)

Les auteurs de (Zhang, et al., 2002) définissent un processus d'intégration des XSDs. Ce processus reçoit en entrée un ensemble de XSDs et génère un modèle conceptuel global modélisé sous forme d'un diagramme de classes étendu nommé EUML « ***Extended UML*** *class diagram* ». Ce processus comporte trois étapes : *(i) Regroupement ("Clustering") des concepts, (ii) Unification des concepts,* et *(iii) Restructuration des relations.*

Notons que le diagramme EUML prend en compte les propriétés sémantiques des XSDs (*e.g.*, les cardinalités et l'ordre des données dans le document XML).

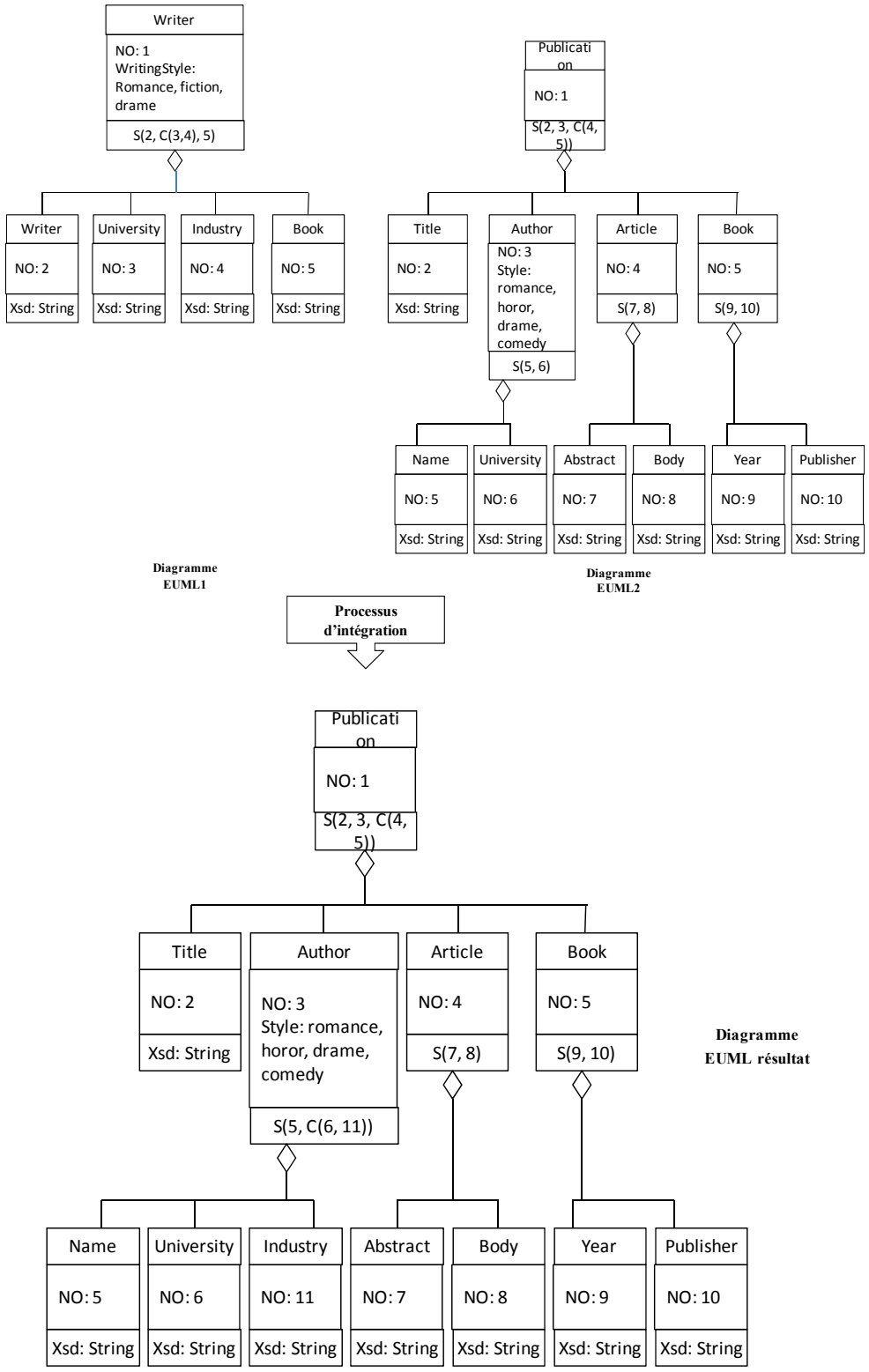


Figure 16 : *Exemple d'intégration de deux schémas XSDs* **(Zhang, et al., 2002)**.

L'étape de regroupement a pour but de résoudre les conflits de nommage via la base lexicale *Wordnet*. Elle génère des regroupements de concepts synonymes à partir des diagrammes EUML, ensuite sélectionne le nom le plus utilisé au sein de chaque regroupement pour représenter ses concepts. Puis, l'étape d'unification intègre les concepts de chaque cluster et génère un concept global[6] et les informations de mapping. Au cours de cette étape, les conflits de typage et de structure sont résolus. Les conflits de typage sont remédiés par l'utilisation des points de vue ou l'union des types de données disjoints. Alors que, les conflits de structure sont réglés par le biais d'un ensemble de règles. Enfin, la troisième étape restructure les relations entre les différentes classes du diagramme EUML résultat et supprime celles qui sont redondantes. Notons que l'utilisateur peut intervenir durant les étapes du processus d'intégration.

La *Figure 16* illustre un exemple d'intégration de deux XSDs qui seront convertis en deux diagrammes nommés EUML1 et EUML2. Le processus de leur intégration produit un digramme EUML représenté dans la même figure.

Notons que ce processus ne prend pas en considération les acronymes des noms des éléments. De plus, l'unification des éléments des diagrammes EUML ne considère pas la hiérarchie des éléments dans le diagramme.

2.2.6. LES TRAVAUX DE (DE-MEO, ET AL., 2003)

Les auteurs de (De-Meo, et al., 2003) suggèrent une approche d'intégration des XSDs. leur approche reçoit en entrée deux XSDs et un degré de sévérité ("*severity degree"*) spécifié par l'utilisateur. Ce degré est un seuil à partir duquel l'intégration des éléments des XSDs peut être réalisée. Le résultat de l'approche est un XSD intégré.

Pour déterminer les éléments synonymes des XSDs en entrée, les auteurs proposent, d'une part, d'utiliser *Wordnet* et, d'autre part, de vérifier le degré de sévérité et ceci en examinant les éléments en voisinage. Les éléments qui ne satisfont pas le degré de sévérité présentent les éléments homonymes[7]. Ensuite, ces propriétés (*i.e.*, synonymie et homonymie) sont exploitées pour modifier les XSDs afin de les uniformiser structurellement et sémantiquement. Puis, les éléments sémantiquement similaires des deux XSDs en entrée sont fusionnés et les éléments redondants et/ou ambigus sont supprimés pour obtenir un XSD

[6] Un concept global est le résultat d'intégration des concepts appartenant à un même regroupement.
[7] L'homonyme caractérise un mot qui se prononce et s'écrit de la même manière qu'un autre mot qui n'a pas le même sens.

global résultat. Dans cette approche, l'utilisateur peut intervenir pour approuver le résultat obtenu.

Cependant, cette approche ne traite pas les acronymes. De plus, elle est conçue pour l'intégration de deux XSDs. Subséquemment, pour intégrer plusieurs XSDs, il faut réitérer le processus d'intégration plusieurs fois.

2.2.7. LES TRAVAUX DE (MELLO, ET AL., 2005)

Dans le cadre d'étendre leur travail antérieur (Mello, et al., 2002) (*cf.* section 2.2.2), (Mello, et al., 2005) proposent un processus ascendant et semi-automatique nommé BInXS « ***B**ottom-up **In**tegration of **X**ML **S**chemata* » pour l'intégration des structures des documents XML. Ce processus reçoit en entrée un ensemble de structures XML (DTDs et/ou XSDs) appartenant à un même domaine et génère un schéma conceptuel global basé sur le modèle conceptuel ORM/NIAM (« ***O**bject with **R**oles **M**odel /**N**atural language **I**nformation **A**nalysis **M**ethod* »). Leur processus est ascendant du fait qu'il part d'un ensemble de schémas XML, il est semi-automatique vu que l'utilisateur intervient dans le processus d'intégration. Il comporte deux étapes : *(i) Conversion de schémas*, et *(ii) Unification* (*cf. Figure 17*).

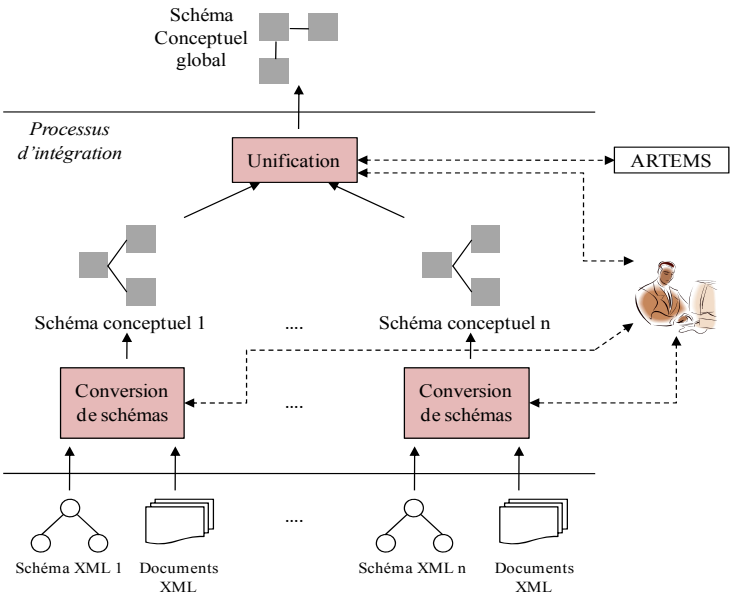


Figure 17 : *Processus d'intégration BInXS* **(Mello, et al., 2005)**.

L'étape de conversion produit pour chaque structure XML un schéma conceptuel. Elle se décompose en trois sous-étapes : *Prétraitement*, *Conversion* et *Restructuration*. Le prétraitement modifie la définition du schéma XML en supprimant les éléments non pertinents. Cette sous-étape nécessite l'intervention de l'utilisateur. Ensuite, la conversion génère un schéma conceptuel préliminaire et les informations de mapping et ceci en

appliquant un ensemble de règles de conversion sur les schémas résultats du prétraitement. Finalement, la restructuration génère un schéma conceptuel définitif en effectuant des modifications manuelles (*e.g.*, définition des noms pour les concepts générés automatiquement) et automatiques (*e.g.*, suppression des relations redondantes) sur le schéma conceptuel préliminaire.

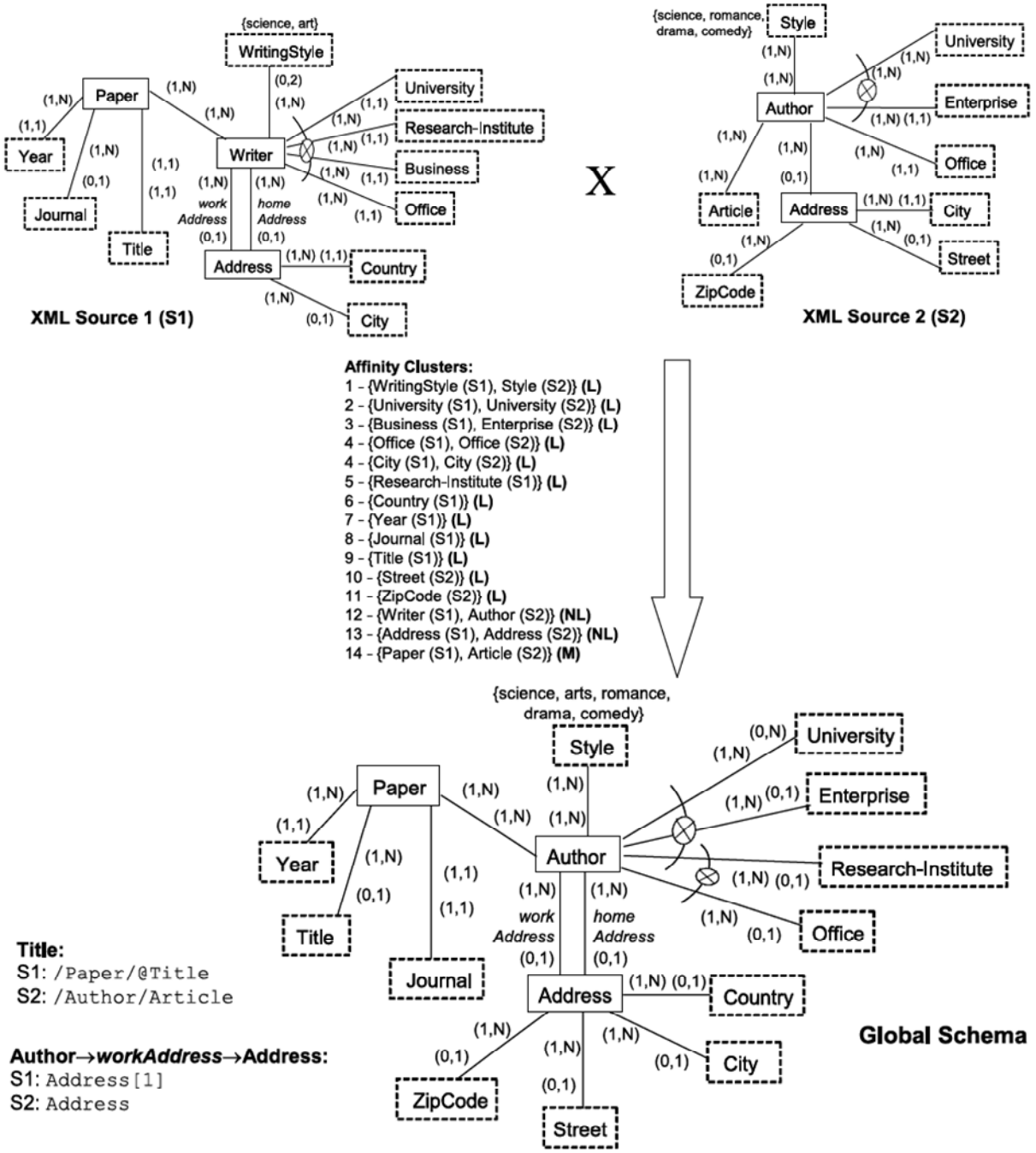


Figure 18 : *Exemple d'unification* **(Mello, et al., 2005)**.

La deuxième étape du processus (*i.e.*, unification) génère un schéma conceptuel global à partir des schémas conceptuels produits dans l'étape précédente, elle suit les étapes classiques

d'intégration des schémas de base de données : *Comparaison des schémas*, *Fusion*, et *Restructuration*. Tout d'abord, la comparaison des schémas définit des regroupements de synonymes des concepts à partir des schémas en entrée. Ensuite, les concepts de chaque regroupement sont fusionnés afin de générer un schéma global préliminaire et ceci en se basant sur un ensemble de règles d'unification (*i.e.,* la sous-étape : fusion). Finalement, la restructuration génère un schéma global final en validant le schéma global préliminaire défini dans l'étape précédente. La *Figure 18* schématise deux schémas conceptuels et le résultat de leur unification (*i.e.*, schéma global). Cependant, lors de l'unification, les auteurs ne prennent pas en considération les informations structurelles des éléments des XSDs ce qui risque d'aboutir à un résultat d'unification imparfait.

Dans cette section, nous nous sommes limités à étudier les travaux qui nous semblent les plus pertinents traitant l'unification des structures des documents XML et plus particulièrement aux travaux qui ont proposé des approches pour l'unification des DTDs et des XSDs. Nous avons alors constaté que certaines approches de la littérature traduisent les structures des documents XML dans un modèle approprié (*e.g.*, arbre, diagramme EUML), ce qui facilite, ultérieurement, le traitement des éléments des structures (*i.e.*, l'application des règles d'unification). En effet, la traduction des structures dans un modèle rend la détermination des éléments liés à l'élément à unifier plus aisée ; ceci améliore la qualité du résultat d'unification.

2.3. Comparaison des travaux d'unification des structures des documents XML

Nous synthétisons les travaux de la littérature relatifs à l'unification des structures des documents XML dans le *Tableau 1* où les lignes représentent les approches étudiées et les critères d'évaluation de ces approches sont portés en colonnes. Les critères que nous établissons sont les suivants :

- C1 : L'approche unifie des DTDs.
- C2 : L'approche unifie des XSDs.
- C3 : L'approche traduit les structures des documents XML en un modèle approprié (*e.g.*, arbre, diagramme).
- C4 : L'approche traite les noms des éléments des structures des documents XML afin de résoudre les problèmes de *Synonymies* ou des *Acronymes*.
- C5 : L'approche calcule un degré de similarité afin de déterminer les structures qui méritent d'être fusionnées.

- C6 : L'approche calcule une matrice de similarité entre les documents en entrée. Ce calcul peut être :
    - Itératif : La matrice est recalculée jusqu'à ce que ce le calcul devienne sans intérêt.
    - Interactif : L'utilisateur intervient dans ce calcul, par exemple pour arrêter le processus.
- C7 : L'approche définit des règles d'unification pouvant être :
    - Syntaxiques : Elles traitent les noms des éléments des structures des documents XML ainsi que leurs cardinalités.
    - Structurelles : Elles prennent en considération la hiérarchie des éléments (*i.e.*, pères et fils).
- C8 : Le concepteur intervient dans le processus d'unification pour approuver le résultat.

Critères / Approches	**C1** : Unification des DTDs	**C2** : Unification des XSDs	**C3** : Traduction de la structure XML dans un modèle approprié	**C4** : Traitement des noms des éléments des structures		**C5** : Calcul de degré de similarité	**C6** : Calcul de matrice de similarité		**C7** : Définition des règles d'unification		**C8** : Participation du décideur.
				Synonymie	Acronyme		Itératif	Interactif	Syntaxiques	Structurelles	
(Lee, et al., 2002)	✓	-	✓	✓	✓	✓	-	-			-
(Mello, et al., 2002)	✓	-	✓	✓	-	-	-	-	-	-	-
(Yoo, et al., 2005)	✓	-	✓	✓		-	-	-	-	-	-
(Khrouf, et al., 2003)			-	-	-	✓	-	-			-
(Zhang, et al., 2002)	-	✓	✓	✓	-	-	-	-	✓	-	✓
(De-Meo, et al., 2003)	-	✓	-	✓	-	-	-	-	✓	-	✓
(Mello, et al., 2005)	✓	✓	✓	✓	-	-	-	-	✓	-	✓

Légende :
✓ : Critère supporté. - : Critère non supporté.

Tableau 1 : *Tableau comparatif des travaux d'unification des structures des documents XML.*

Suite à cette étude, nous constatons que la plupart des approches traitent les noms des éléments des structures (synonymie et/ou acronyme). Ce traitement permet de résoudre les ambiguïtés des noms des éléments. Aussi, ces approches définissent un degré de similarité pour mesurer la pertinence d'intégration des structures des documents XML. Généralement, ce calcul du degré de similarité est non optimisé : les auteurs n'utilisent pas un calcul itératif

de matrice de similarité. En outre, les règles d'unification ne traitent pas les relations structurelles des éléments des structures des documents XML.

2.4. Modélisation multidimensionnelle des documents : État de l'art

Rappelons que la modélisation multidimensionnelle vise à concevoir des modèles multidimensionnels reflétant les besoins analytiques des décideurs. Dans la littérature, il y a des travaux qui s'intéressent à la modélisation des documents XML orienté-données comme (Hachaichi, et al., 2010) et (Boussaid, et al., 2006), et d'autres travaux qui traitent la modélisation des documents XML orienté-documents. Nous étudions dans cette section les travaux traitant la modélisation des documents XML orienté-documents.

2.4.1. Les travaux de (McCabe, et al., 2000)

Dans (McCabe, et al., 2000), les auteurs utilisent le modèle en étoile pour la modélisation multidimensionnelle des documents. Ce modèle se compose d'un fait et de cinq dimensions : *Temps*, *Terme*, *Document*, *Catégorie* et *Localisation*. Le *Fait* comporte la mesure *Nombre d'occurrence* du terme. La *Figure 19* montre l'allure d'un tel modèle en étoile.

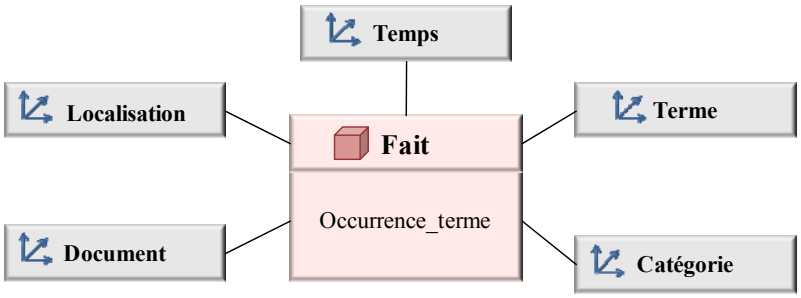


Figure 19 : *Modèle en étoile pour l'analyse des documents* **(McCabe, et al., 2000)**.

En fait, la dimension *Document* contient les informations structurelles et bibliographiques du document. Alors que, la dimension *Terme* est définie comme suit : < *Terme*, *Catégorie*, *poids du terme* >. La dimension *Catégorie* décrit la catégorie du terme (*e.g.*, nom, verbe) ou une hiérarchie comme de *Wordnet*.

Cependant, le modèle proposé comporte des axes d'analyses prédéterminé ; par exemple, des analyses par auteur ne sont pas possibles du fait de l'absence de la dimension *Auteur*. Par conséquent, les analyses sur ce modèle seront limitées. Ainsi, seules les analyses quantitatives peuvent être réalisées du fait que le modèle en étoile mesure le nombre d'occurrences de chaque terme dans un document.

2.4.2. Les travaux de (Khrouf, 2004)

Dans (Khrouf, 2004), l'auteur propose un processus pour l'analyse multidimensionnelle des données de l'entrepôt de documents. Ce processus est basé sur un modèle en étoile pour la modélisation des documents. Il comporte les trois étapes suivantes : *(i) Construction du schéma du magasin*, *(ii) Génération du magasin*, et *(iii) Visualisation de la table multidimensionnelle résultat*. La *Figure 20* montre l'enchaînement de ces étapes.

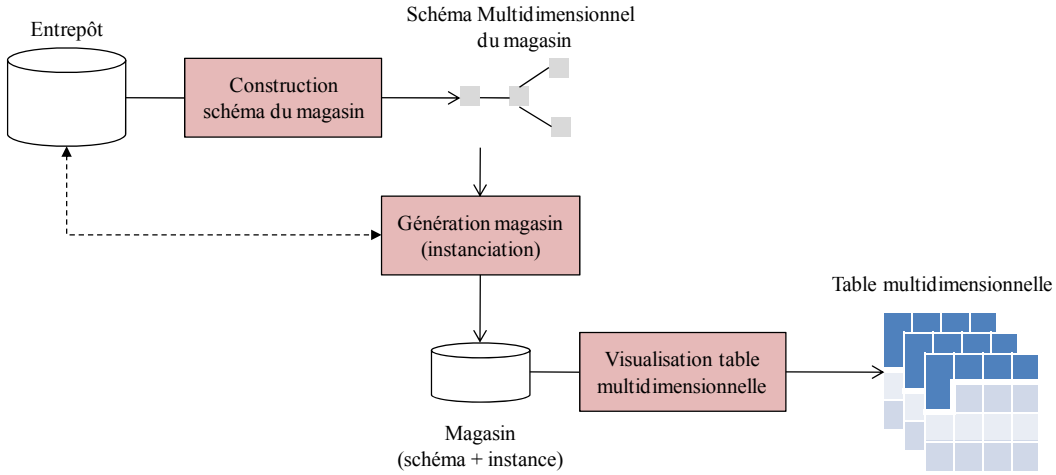


Figure 20 : *Processus d'analyse multidimensionnelle* **(Khrouf, 2004)**.

Dans ce processus, l'étape de construction d'un schéma de magasin donne la main au décideur pour spécifier les composants d'analyse, c'est-à-dire un fait et des dimensions, la position d'affichage des dimensions (en ligne ou en colonne), et la fonction d'agrégation de la mesure du fait. Ensuite, le magasin est généré et instancié automatiquement. Finalement, le résultat est visualisé sous forme d'une table multidimensionnelle. La *Figure 21* présente un exemple de modèle en étoile permettant de calculer le nombre des livres édités en fonction des trois dimensions : *Auteur*, *Editeur* et *Année*.

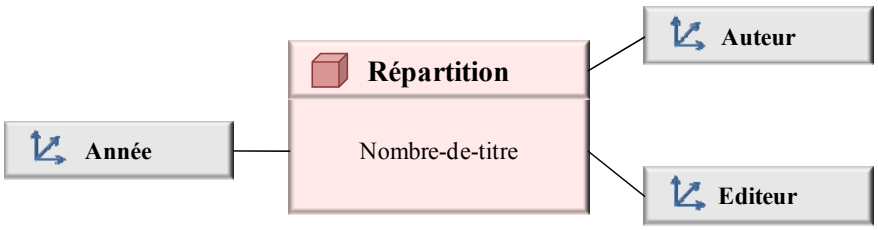


Figure 21 : *Exemple de modèle multidimensionnel* **(Khrouf, 2004)**.

Cependant l'élaboration du modèle conceptuel multidimensionnel (*i.e.*, détermination des concepts fait, dimensions, hiérarchie) est complètement manuelle.

2.4.3. Les travaux de (Tseng, et al., 2006)

Les auteurs de (Tseng, et al., 2006) utilisent le modèle en étoile pour modéliser de manière multidimensionnelle des documents. Ils spécifient trois types de dimensions : *Ordinaires*, *Métadonnées* et *Catégorie*. Une *dimension ordinaire* est déterminée par le bais des données extraites du contenu des documents à entreposés ; un exemple de dimension ordinaire est la dimension mots-clés. La *dimension Métadonnées* modélise les métadonnées extraites des documents (*e.g.*, Auteur, Editeur). L'initiative des métadonnées *Dublin Core* fournit ces métadonnées. Alors que la *dimension Catégorie* constitue les données externes du document qui permettent sa catégorisation. La *Figure 22* illustre un exemple d'un modèle en étoile modélisant les papiers publiés dans des journaux de recherche.

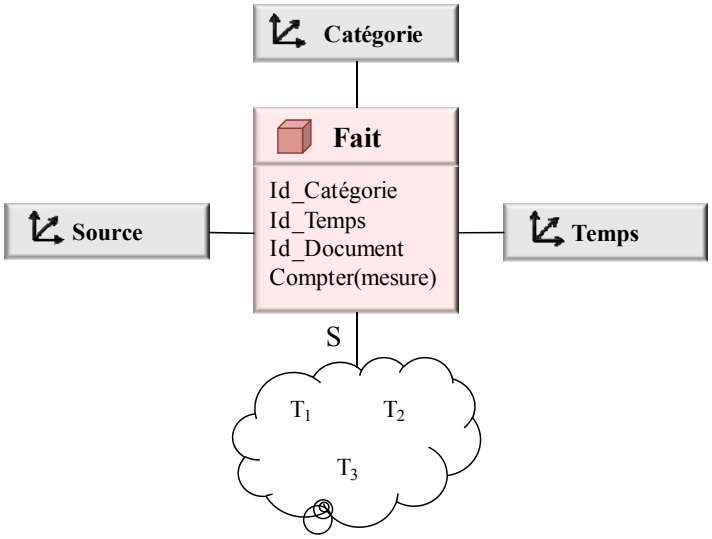


Figure 22 : *Modèle en étoile des papiers journal de recherche* **(Tseng, et al., 2006)**.

Néanmoins, un tel modèle en étoile ne réalise que des analyses quantitatives parce que les mesures sont numériques et donc ne peuvent être que comptées. De plus, les analyses sont limitées puisque le sujet d'analyse est prédéterminé.

2.4.4. Les travaux de (Ravat, et al., 2007)

Dans (Ravat, et al., 2007), les auteurs proposent de modifier le modèle en constellation pour réaliser l'analyse multidimensionnelle d'une collection de documents de structures homogènes correspondant à un besoin d'analyse. En fait, ils suggèrent d'ajouter un nouveau type de mesure (*i.e.*, textuelle) et deux nouvelles dimensions nommées *Structure* et *Complémentaire*.

Leur modèle proposé distingue deux types de mesures : *numérique* et *textuelle*. Une mesure numérique peut être soit additive, soit semi-additive. Alors que, la mesure textuelle peut représenter un mot, un paragraphe, un document, etc. Elle peut être *brute* (contenu complet d'un document) ou *élaborée* (issue d'une mesure textuelle brute après prétraitement). Ainsi, ce modèle différencie cinq types de dimensions : *Ordinaire*, *Métadonnées*, *Catégorie* (*cf.* section 2.4.3), *Structure,* et *Complémentaire*. La dimension structure décrit la *Structure* commune des documents d'une collection. Tandis que la dimension *Complémentaire* est déterminée par des sources de données complémentaires telles que les données de l'état civil des auteurs des articles.

La *Figure 23* schématise un exemple de modèle en étoile pour les articles d'un laboratoire de recherche. Ce modèle est composé d'un fait nommé *Article* et de quatre dimensions : *Auteur, Mots_clefs, Structure* et *Temps*. Le fait est décrit par les deux mesures *Nb_mots_clefs* et *Texte*.

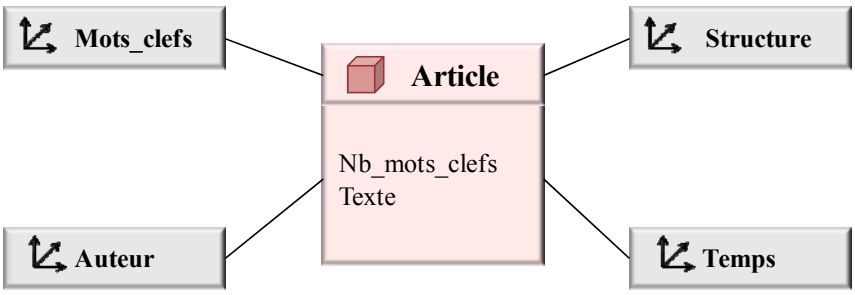


Figure 23 : *Modèle en étoile pour l'analyse multidimensionnelle des articles scientifiques* **(Ravat, et al., 2007)**.

Dans ce travail, les auteurs ne proposent pas une méthode semi-automatique/automatique de conception du modèle multidimensionnel ; cette conception est réalisée manuellement. En effet, les auteurs n'ont pas proposé des règles ou des algorithmes pour assister le concepteur décisionnel à identifier les faits et les dimensions.

2.4.5. Les travaux de (Tournier, 2007) et (Pujolle, et al., 2011)

Dans (Tournier, 2007), l'auteur propose un nouveau modèle multidimensionnel appelé *modèle en galaxie* pour l'analyse des documents XML orienté-documents. Ce modèle est décrit par l'unique concept « *Dimension* » ; le concept fait n'est pas explicité. Les dimensions de la galaxie sont liées entre elles par un ou plusieurs nœuds. Chaque nœud réunit les dimensions compatibles pour une même analyse, c'est-à-dire pouvant être utilisées ensemble dans une même requête. Quant à la modélisation des données textuelles, l'auteur distingue des

attributs et des dimensions documentaires. En effet, les attributs documentaires représentent les données issues des documents textuels (*e.g.*, paragraphe) ; alors que, les dimensions documentaires décrivent la structure du document.

Afin de construire un modèle en galaxie, (Tournier, 2007) et (Pujolle, et al., 2011) proposent un processus hybride. Ce processus accepte un ensemble de besoins des utilisateurs et un ensemble de documents XML orienté-documents.

La *Figure 24* illustre un exemple de modèle en galaxie ; il est composé de six dimensions : *Conférences, Dates, Rapports, Articles, Auteurs* et *Instituts,* et de deux nœuds. Le premier nœud relie les quatre dimensions *Conférences, Dates, Articles,* et *Auteurs*. Il associe les articles publiés au sein d'une conférence et écrits par des auteurs à une date donnée. Tandis que, le deuxième nœud relie les quatre dimensions : *Dates, Rapports, Auteurs,* et *Instituts.* Il traduit les *Rapports* de projets dirigés par des *Instituts* et encadrés par des personnels scientifiques *Auteurs*) à une certaine *Date*.

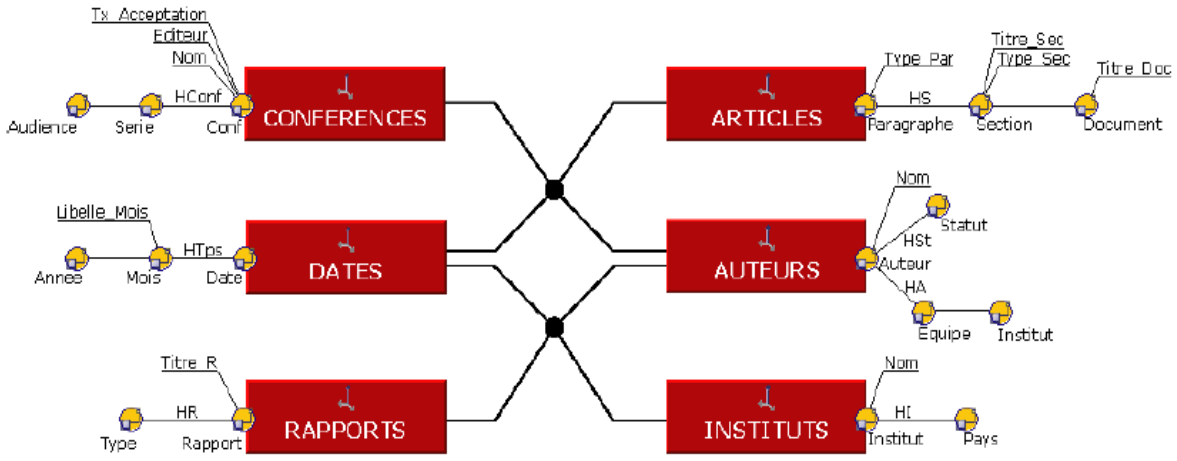


Figure 24 : *Exemple d'un modèle en galaxie* **(Tournier, 2007)**.

Encore une fois, ces auteurs ne proposent pas des règles permettant le passage des documents XML orienté-documents au modèle en galaxie.

A travers cette section, nous avons passé en revue les travaux de la littérature qui nous semblent les plus intéressants relatifs à la modélisation multidimensionnelle des documents. Nous nous sommes focalisés sur les deux modèles en *étoile* et en *galaxie*. Cette étude nous a permis de savoir les points forts et les points faibles de ces travaux. Nous continuons à les comparer dans la section suivante.

2.5. Comparaison des travaux de modélisation multidimensionnelle des documents XML

Nous synthétisons les travaux traitant la modélisation multidimensionnelle des documents dans le *Tableau 2*. Les lignes représentent les travaux examinés et les colonnes sont les cinq critères d'évaluation que nous avons pu dégagés.

- C1 : L'approche modélise les documents XML orienté-données.
- C2 : L'approche modélise les documents XML orienté-documents.
- C3 : L'approche utilise le modèle en étoile pour la modélisation multidimensionnelle des documents.
- C4 : L'approche utilise le modèle en galaxie pour la modélisation multidimensionnelle des documents.
- C5 : L'approche détermine les concepts multidimensionnels :
    - Manuellement.
    - Automatiquement.
    - Semi-automatiquement.

Critères / Approches	**C1** : Modélisation des documents XML orienté-données	**C2** : Modélisation des documents XML orienté-documents	**C3** : Utilisation du modèle en constellation pour la modélisation multidimensionnelle	**C4** : Utilisation du modèle en galaxie pour la modélisation multidimensionnelle	**C5** : Détermination des concepts multidimensionnels		
					Manuelle	Automatique	Semi-automatique
(McCabe, et al., 2000)	-	✓	✓	-			
(Khrouf, 2004)	-	✓	✓	-	✓	-	-
(Tseng, et al., 2006)	-	✓	✓	-	✓	-	-
(Ravat, et al., 2007)	-	✓	✓	-	✓	-	-
(Tournier, 2007) et (Pujolle, et al., 2011)	-	✓	-	✓	✓	-	-

Légende :
✓ : Critère supporté. - : Critère non supporté.

Tableau 2 : *Tableau comparatif des travaux de modélisation multidimensionnelle des documents.*

Nous remarquons que, quel que soit le modèle multidimensionnel en étoile ou en galaxie utilisé pour les documents XML orienté-documents, aucun travail n'a proposé des règles

permettant d'assister le concepteur de l'entrepôt à élaborer même de manière semi automatique en un modèle multidimensionnel.

Devant cette situation, et afin de contribuer à alléger la tâche fastidieuse et difficile du concepteur décisionnel, nous avons vu utile de proposer une approche semi-automatique pour la construction d'un schéma d'entrepôt de documents.

2.6. Aperçu de l'approche proposée

Notre objectif est d'assister le concepteur décisionnel dans la phase de conception de modèles multidimensionnels pour les documents XML orienté-documents. Pour atteindre cet objectif, nous proposons une approche semi-automatique qui reçoit en entrée un ensemble de structures (DTD et/ou XSD) de documents et génère un (des) modèles(s) en galaxie. Cette approche comporte deux méthodes automatisables complémentaires :

- *Une méthode d'unification des structures des documents XML, et*
- *Une méthode de conception de modèles en galaxie*

La méthode d'unification des structures des documents XML vise à produire la ou les structures communes pour un ensemble de documents. Elle est indispensable vu l'hétérogénéité des structures des documents XML à entreposer. Elle traduit chaque structure en un arbre ; ces arbres seront ensuite fusionnés en appliquant un ensemble de règles que nous définissons. Finalement, les arbres résultats seront présentés au concepteur pour approbation (suppression et/ou modification). Cette traduction vers des arbres nécessiterait la définition d'un ensemble de règles de transformation.

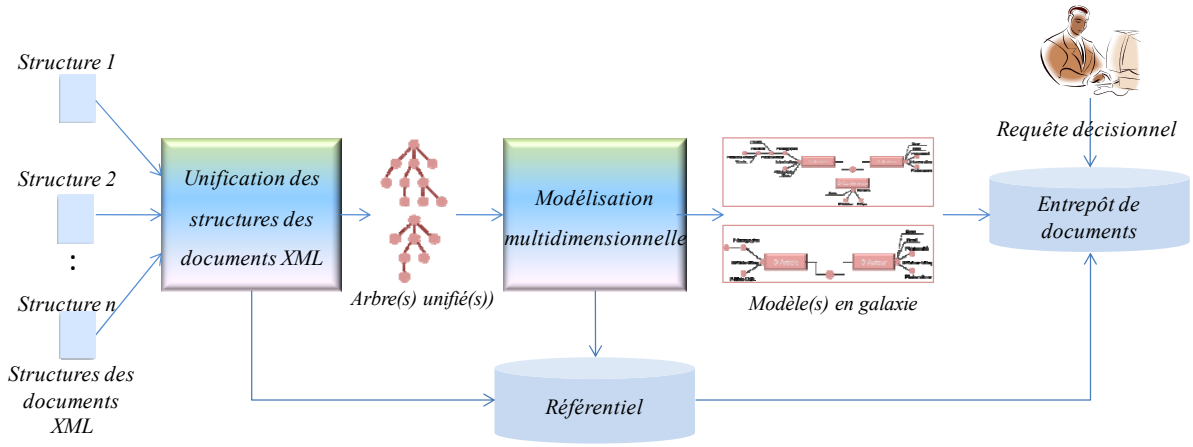


Figure 25 : *Approche de construction d'un schéma d'entrepôt de documents* **(Ben Messaoud, et al., 2010)**.

La méthode de conception vise à générer des modèles en galaxie à partir des arbres résultats de l'unification. Pour réaliser cette génération, nous définirons un ensemble de règles permettant d'identifier les éléments multidimensionnels du modèle. La *Figure 25* illustre cette approche proposée.

2.7. Conclusion

A travers ce chapitre, nous avons étudié les travaux de la littérature les plus pertinents relatifs à l'unification des structures des documents XML et à la modélisation multidimensionnelle des documents.

Concernant les travaux d'unification, nous avons constaté que la plupart propose des méthodes qui se basent sur des règles ne traitant pas les informations structurelles des éléments à unifier.

Pour ce qui est de la modélisation multidimensionnelle, toutes les méthodes étudiées ne suggèrent pas de règles assurant le passage des documents XML vers une conception de modèle multidimensionnel.

Pour atteindre notre objectif d'entreposage de documents XML centrés-documents, nous avons esquissé une proposition d'approche articulée autour de deux méthodes : une méthode pour l'unification des structures des documents XML hétérogènes et appartenant à un même domaine, et une seconde méthode pour la modélisation multidimensionnelle de ces documents. La première méthode a pour but de générer la structure commune pour un ensemble de documents en entrée ; alors que la deuxième vise à traduire quasi-automatiquement le résultat de la première méthode sous forme d'un modèle en galaxie. Nous détaillons ces deux méthodes dans les chapitres 3 et 4.

CHAPITRE 3 : PROPOSITION D'UNE METHODE D'UNIFICATION DES STRUCTURES DE DOCUMENTS XML

Résumé du chapitre :
Ce chapitre se focalise sur la présentation de notre méthode proposée pour l'unification des structures des documents XML et détaille minutieusement ses étapes. Cette méthode utilise une structure d'arbre pour la représentation des structures XML. Elle se base sur un ensemble de trois opérateurs pour l'unification et quatre contraintes pour vérifier la validité syntaxique du résultat d'unification.

Sommaire du chapitre 3

3.1. Introduction

Les documents XML, même appartenant à un même domaine, sont souvent décrits par des structures multiples. Cette situation augmente les difficultés lors des analyses OLAP puisque le décideur sera contraint de considérer l'hétérogénéité structurelle de ces documents. En effet, il se trouve dans l'obligation de gérer lui-même cette hétérogénéité à différents niveaux : *Ecriture de nombreuses requêtes*, soit autant de requêtes que de structures distinctes à interroger, il obtient ainsi des résultats partiels ; *Synthèse des résultats partiels* des requêtes afin de construire le résultat final. Pour remédier à ces inconvénients et faciliter les analyses, le décideur a besoin d'une structure offrant une vue globale de l'ensemble des documents XML de l'entrepôt et utiliser un langage ensembliste.

C'est dans cette perspective que nous proposons, dans ce chapitre, une méthode pour l'unification des structures de documents XML. La méthode est articulée autour de quatre étapes : *Représentation arborescente des structures XML*, *Génération des arbres unifiés, Approbation des arbres unifiés,* et *Vérification des arbres* à travers un ensemble de contraintes que nous définissons.

Ce chapitre est structuré comme suit : la section 2 donne un aperçu général de notre méthode d'unification. La section 3 détaille sa première étape *Représentation arborescente des structures XML*. Quant à la section 4, elle présente la deuxième étape *Génération des arbres unifiés*. La section 5 s'intéresse à l'étape d'*Approbation des arbres unifiés*. Finalement, la section 6 définit et illustre les contraintes de vérification des arbres générés.

3.2. Méthode d'unification des structures des documents XML

Dans la littérature, il y a des travaux qui ont proposé des approches pour déterminer la structure commune des documents XML (*cf.* chapitre 2 Section 2.2) et d'autres qui s'intéressent à définir la structure sémantique des documents XML comme (Ben Mefteh, et al., 2013). Dans le cadre de nos travaux, nous nous intéressons à définir la structure commune des documents XML.

En se référant à notre étude de l'état de l'art (*cf.* chapitre 2 Section 2.2), il existe quatre façons pour représenter la structure arborescente des DTDs et des XSDs : schéma conceptuel canonique, automate fini, diagramme de classes étendu (*i.e.*, EUML), et arbre.

Le schéma conceptuel canonique est basé sur les notations graphiques du modèle entité/association (E/A). Alors que le diagramme EUML est basé sur des notations

spécifiques. De ce fait, la compréhension de ces deux modèles (*i.e.*, schéma conceptuel canonique et diagramme EUML) par un décideur non informaticien nécessite une bonne connaissance du modèle E/A et du diagramme de classes. En ce qui concerne l'automate, il est composé d'un ensemble d'états et de transitions. Cependant, ce modèle risque d'être compliqué lorsqu'il comporte un ensemble assez important d'états et de transitions. Cette complexité engendre des difficultés de compréhension par le décideur. D'autre part, un arbre est composé de nœuds et d'arrêtes et telles qu'il existe un chemin unique pour relier deux nœuds quelconques. Il a l'avantage de représenter la nature arborescente des structures des documents XML. Aussi, il est graphique et facile à comprendre par les décideurs. Vu ces propriétés de l'arbre, nous optons pour ce dernier pour représenter la structure arborescente des DTDs et des XSDs.

Dans cette section, nous présentons la méthode d'unification des structures des documents XML introduite dans (Ben Messaoud, et al., 2011a) et complétée dans (Ben Messaoud, et al., 2012). Cette méthode reçoit en entrée un ensemble de structures de documents XML et génère un ou plusieurs arbres unifiés. Elle comporte les quatre étapes suivantes :

a. Représentation arborescente,
b. Génération des arbres unifiés,
c. Approbation des arbres unifiés, et
d. Vérification des arbres.

La *Figure 26* montre l'enchaînement de ces étapes que nous détaillons dans les sections suivantes.

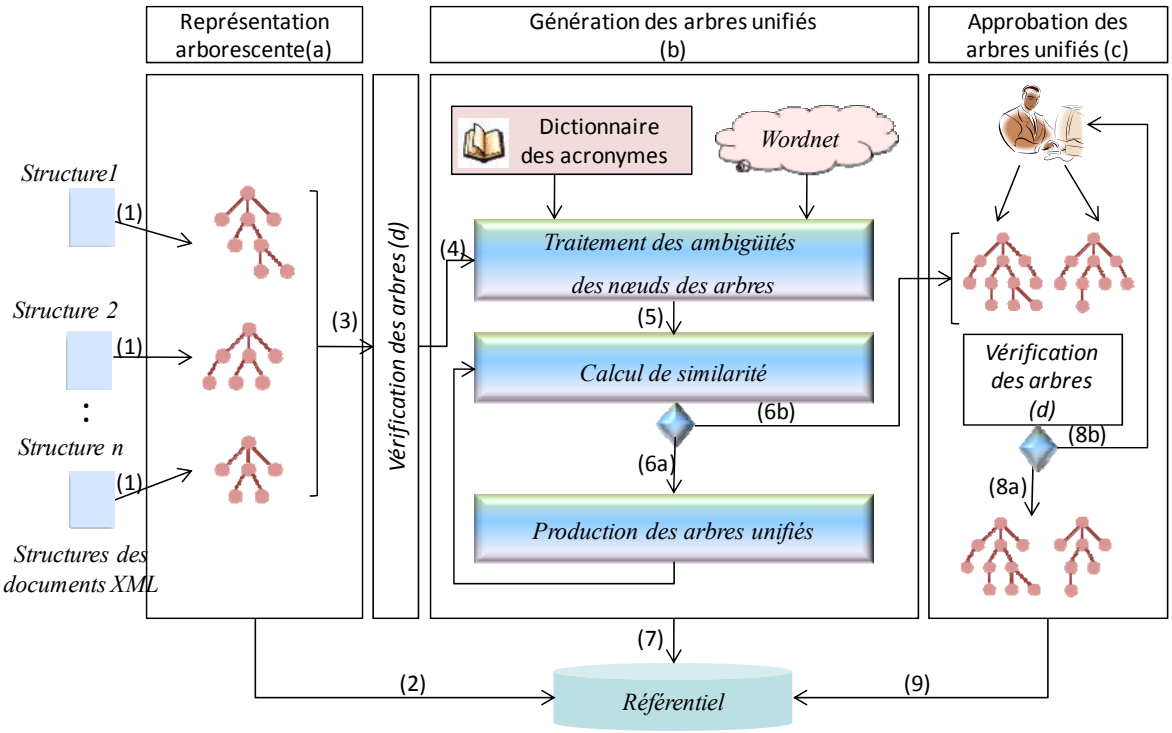


Figure 26 : *Méthode d'unification des structures des documents XML.*

Du fait que les arbres résultats d'unification seront utiles pour la détermination des concepts multidimensionnels (*i,e.,* méthode de modélisation multidimensionnelle), et que les structures des documents XML en entrée et les résultats intermédiaires sont nécessaires pour l'interrogation des documents XML, nous stockons ces éléments dans un référentiel. La *Figure 27* illustre le diagramme de classes de ce référentiel.

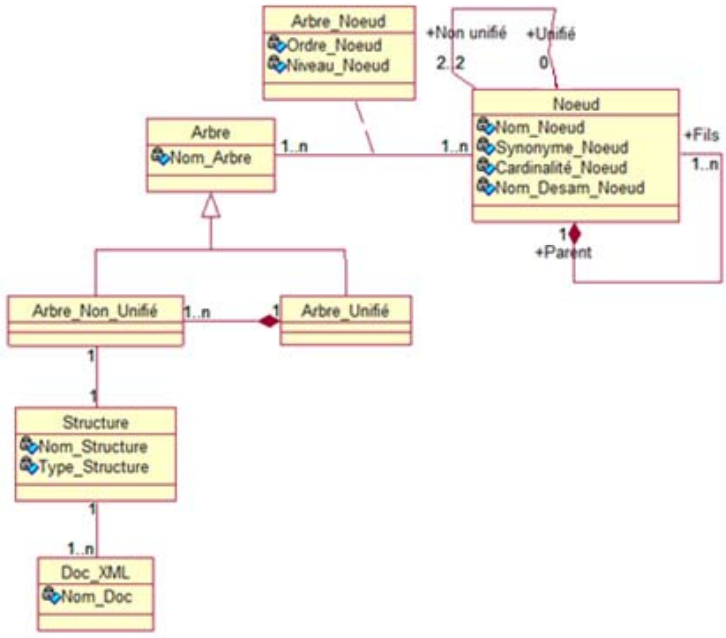


Figure 27 : *Référentiel des arbres.*

3.3. Représentation arborescente

Dans cette étape, chaque structure XML est transformée sous forme d'un arbre.

La transformation d'une structure XML en un arbre est réalisée en appliquant deux règles et ceci afin de permettre la bonne transformation d'une structure de document XML en un arbre. Ces deux règles sont les suivantes :

Règle 1 :

Chaque élément ou attribut de la structure XML est transformé en un nœud annoté par une cardinalité extraite de sa balise XML.

Règle 2 :

Chaque paire d'éléments XML définies dans la même balise se transforme en un arc traduisant la relation entre ces éléments.

Ces deux règles permettent de dégager tous les éléments constituant un arbre. En effet, la première règle identifie les nœuds des arbres ; alors que la deuxième règle détermine les relations entre ces nœuds. Nous définissons un arbre comme suit :

Définition 1 :

Un arbre A est défini selon le triplet (E, r, N) :

- $E = \{e_1, e_2 ..., e_m\}$: un ensemble non vide de m nœuds.
- $r \in E$: le nœud racine de l'arbre A.
- $N = \{n_1, n_2 ..., n_k\}$: un ensemble de k arcs de A.

Chaque nœud est identifié par son nom et son type.

Définition 2 :

Un nœud $e_i \in E$ est défini par le couple *(nom, type)* :

- *nom* : le nom du nœud.
- *type* : le type du nœud qui peut être racine[8], feuille[9] ou fils[10].

[8] Nœud racine est un nœud origine et n'a pas de père.
[9] Nœud feuille est un nœud terminal, c'est-à-dire, n'ayant pas de fils.
[10] Nœud fils est un nœud non terminal lié à un nœud père.

Chaque arc est identifié par un nœud source et un nœud destination annotés par des cardinalités.

Définition 3 :

Un arc $n_i \in N$ est défini par le quadruplet $(e_1, card_1, e_2, card_2)$:

- e_1 : le nœud source de l'arc.
- $card_1$: la cardinalité du nœud e_1.
- e_2 : le nœud destination de l'arc.
- $card_2$: la cardinalité du nœud e_2.

La *Figure 28* illustre un exemple d'une DTD (DTD 1) avec son arbre correspondant (Arbre 1). Cet arbre est composé de neuf nœuds {*Article*, *Title*, *Writer*, ...} et de huit arcs {*Article-Title*, *Article-Writer*, *Article-Section*, ...}. Certains arcs sont annotés par des cardinalités (? ou +) extraites de la DTD1.

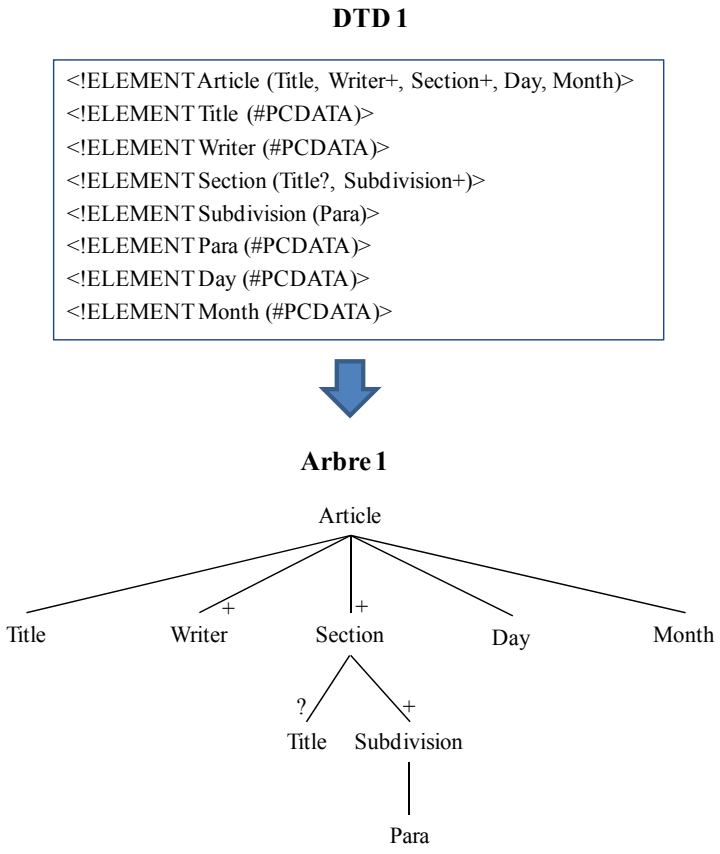


Figure 28 : *Exemple d'une DTD avec son arbre.*

Pour les documents XSD, lorsqu'un arc est annoté par une cardinalité numérique (*>1*), nous remplaçons cette cardinalité par (*).

3.4. Génération des arbres unifiés

L'étape de génération des arbres unifiés reçoit en entrée les arbres résultats de l'étape précédente, applique un ensemble d'opérateurs et génère un ou plusieurs arbres unifiés (Ben Messaoud, et al., 2011a) et (Ben Messaoud, et al., 2012). Elle comporte les trois sous-étapes suivantes :

- Traitement des ambiguïtés des noms des nœuds.
- Calcul de similarité, et
- Production des arbres unifiés.

Ces étapes sont illustrées dans la *Figure 26.b*.

3.4.1. TRAITEMENT DES AMBIGUÏTES DES NOMS DES NŒUDS

L'objectif de cette sous-étape est de résoudre les ambiguïtés des noms des nœuds des arbres. Pour ce faire, nous utilisons la base lexicale *Wordnet* et un dictionnaire des acronymes que nous élaborons.

Initialement, chaque nom acronyme d'un nœud est déterminé puis remplacé par un nom complet et ceci en utilisant un dictionnaire des acronymes. Nous avons utilisé l'algorithme de *Jaro*[11] (Jaro, 1989) pour développer un algorithme de construction de ce dictionnaire (*cf.* Section Annexe annexe1). Ensuite, les nœuds synonymes sont identifiés en se référant à la base lexicale *Wordnet* et sont remplacés par un nom commun. Ce nom est le plus fréquent dans l'ensemble des nœuds synonymes. Finalement, les nœuds d'un même arbre ayant des noms identiques sont renommés en préfixant le nom du nœud par celui de son nœud père afin d'obtenir des arbres étiquetés par des noms uniques.

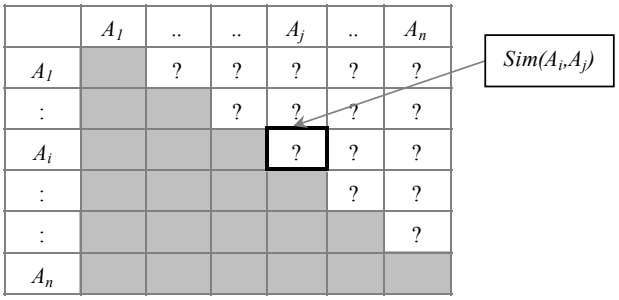
Nous générons ainsi des arbres étiquetés par des noms uniques et standards ce qui permet de calculer plus tard des degrés de similarités plus précis entre les arbres à unifier.

3.4.2. CALCUL DE SIMILARITE

Le calcul de similarité reçoit en entrée les arbres résultats de l'étape précédente et détermine un degré de similarité entre chaque paire d'arbres. Ce calcul permet de décider des

[11] L'algorithme de Jaro calcule la similarité entre deux chaînes de caractères.

arbres qui méritent d'être fusionner, c'est-à-dire, ceux ayant des structures assez similaires. Afin d'optimiser le calcul de similarité, nous définissons une matrice de similarité (MS) triangulaire avec *n* arbres en ligne et en colonne. Elle est inspirée de la matrice proposée dans (Feki, 2004) pour l'intégration des modèles multidimensionnels. Chaque cellule de la matrice MS évalue le degré de similarité *Sim* (A_i, A_j) de l'arbre en ligne *i* avec l'arbre en colonne *j*.



Pour simplifier le calcul de similarité entre deux modèles multidimensionnels, l'auteur de (Feki, 2004) a proposé une variante simplifiée de la méthode de calcul de similarité entre deux entités utilisée dans l'intégration de schémas de bases de données (Akoka, et al., 1998) (PRISM, 2000). Nous avons adapté cette variante pour le calcul de similarité entre deux arbres.

Nous calculons la similarité selon la définition 4.

Définition 4 :

Le degré de similarité des deux arbres A_i (E_i, r_i, N_i) et A_j (E_j, r_j, N_j) est :

$$Sim\left(A_i, A_j\right) = \begin{cases} 0.75 \text{ si } n_i = c_{i,j} \text{ et } n_i < n_j \\ \dfrac{c_{i,j}}{q} \text{ sinon} \end{cases}$$

où :

$n_i = |\ E_i\ |$, $n_j = |\ E_j\ |$, $c_{i,j} = |\ E_i \cap E_j\ |$ et $q = n_i + n_j - c_{i,j}$

Pour calculer la matrice de similarité, nous avons développé la fonction *Calculer-MS*.

```
Fonction Calculer-MS (
n : entier (nombre d'arbres),
T = {T1, T2 …, Ti …, Tn} un ensemble non vide de n arbres avec
Ti = (Ei, ri, Ni))
Retourne MS (n,n): une matrice de similarité ayant n lignes
et n colonnes
Variables
i, j, c, n, m : entier
Début
Pour i de 1 à n faire
  Pour j de (i+1) à n faire
```

```
    n := |Ei|
    m := |Ej|
    c := |Ei ∩ Ej|
    Si (n = c et n < m) alors
      MS(i,j) := 0.75
    Sinon
      MS(i,j) := c/(n + m – c)
    Fin Si
  Fin Pour
Fin Pour
Fin.
```

Notons que la fusion de deux arbres est réalisée lorsque leur degré de similarité calculé dépasse un seuil qui peut être déterminé expérimentalement.

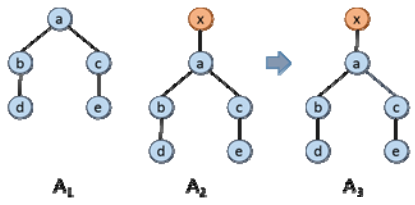
3.4.3. PRODUCTION DES ARBRES UNIFIES

Elle fusionne chaque paire d'arbres possédant un degré de similarité supérieur ou égal au seuil et génère un arbre unifié. Pour cette fusion, nous avons proposé les trois opérateurs suivants : *fusion par inclusion*, *fusion par union des sous-arbres* et *fusion par union des nœuds* (Ben Messaoud, et al., 2011a).

La *fusion par inclusion* est utile quand l'un des deux arbres en entrée est inclus dans l'autre. L'arbre résultat est alors celui qui inclut tous les nœuds. La définition 5 décrit cet opérateur.

Définition 5 :

F-Inclusion $(A_1, A_2) = A_3$



Entrées :

- $A_1 (E_1, r_1, N_1)$
- $A_2 (E_2, r_2, N_2)$

Conditions :

- $A_1 \subseteq A_2$ *ou* $A_2 \subseteq A_1$

Sortie :

- $A_3 = A_2$ *si* $A_1 \subseteq A_2$
- $A_3 = A_1$ *si* $A_2 \subseteq A_1$

La *Figure 29* montre un exemple d'unification des deux arbres *Arbre1* et *Arbre2*. Vu que l'*Arbre2* est inclus dans l'*Arbre1*, alors l'arbre résultat d'unification est *Arbre1* (nommé *Arbre3* dans la *Figure 29*).

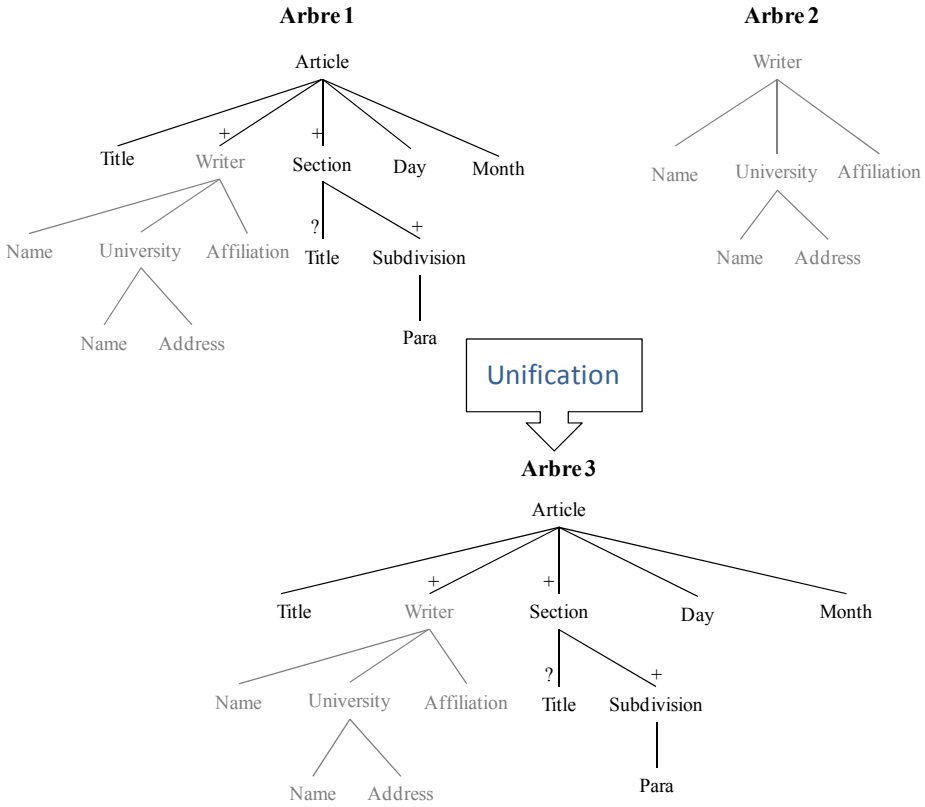
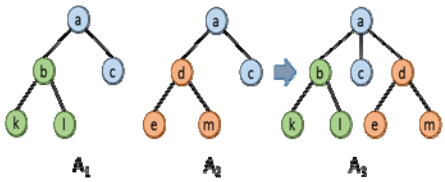


Figure 29 : *Exemple d'unification par inclusion.*

La *fusion par union des sous arbres* est opérée lorsque les nœuds communs des deux arbres en entrée ne partagent pas les mêmes nœuds fils ; dans ce cas, l'arbre résultat est composé de l'union des sous-arbres des arbres en entrée. La définition 6 présente cet opérateur.

Définition 6 :

F-Union-Sous-Arbres $(A_1, A_2) = A_3$



Entrées :

- $A_1 (E_1, r_1, N_1)$
- $A_2 (E_2, r_2, N_2)$

Conditions :

- $E_1 \cap E_2 \neq \varnothing$.
- $\exists e_i \in E_1$ *et* $e_j \in E_2$, $\forall e_i = e_j$, *Parent* (e_i) = *Parent* (e_j) et *Fils* $(e_i) \neq$ *Fils* (e_j).
  /* Parent (e_i) retourne le nœud père du nœud e_i */
  /* Fils (e_i) détermine les nœuds fils du nœud e_i */

Sortie :

- $A_3 (E_3, r_3, N_3)$

Avec $E_3 = E_1 \cup E_2$, $r_3 = r_1 = r_2$ et $N_3 = N_1 \cup N_2$

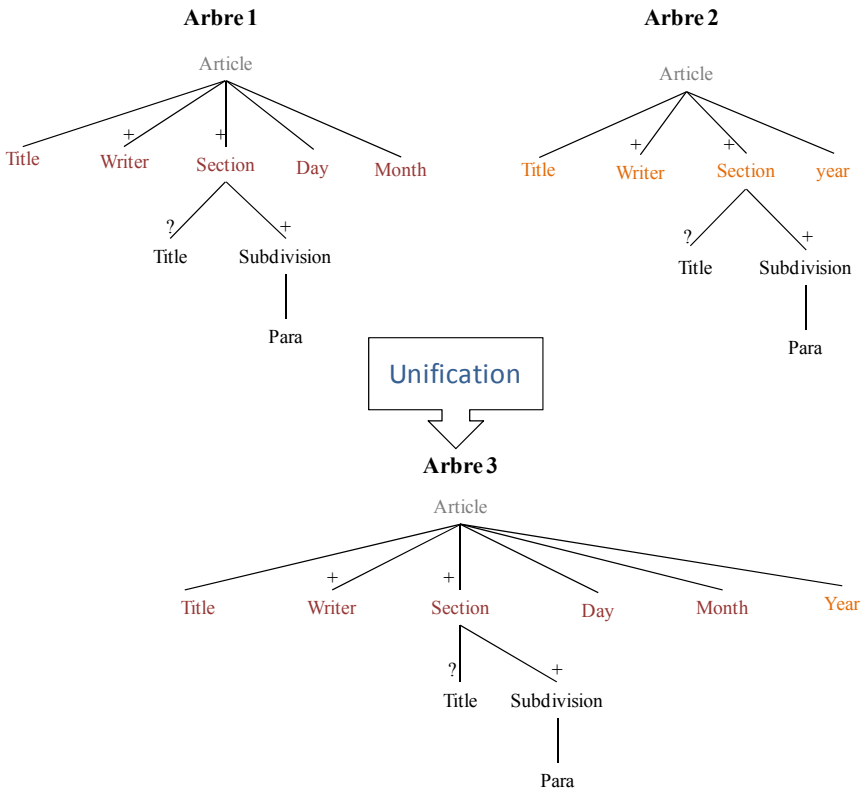


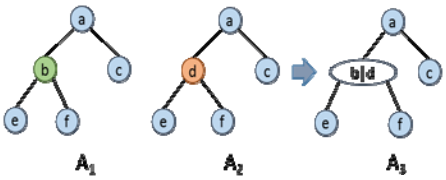
Figure 30 : *Exemple d'unification d'arbres fusionnés par union des sous-arbres.*

La *Figure 30* illustre un exemple d'unification des arbres *Arbre1* et *Arbre2*. Le nœud racine *Article* de ces deux arbres ne partage pas les mêmes nœuds fils. Subséquemment, le nœud racine dans *Arbre3,* résultat de l'unification, englobe tous les nœuds fils du nœud *Article* des deux Arbres.

La *fusion par union des nœuds* est opérée lorsque deux sous-arbres identiques ont des nœuds pères différents. Dans ce cas, l'arbre résultat est caractérisé par un nœud spécifique *ou* qui remplace les nœuds parents distincts des deux arbres en entrée. La définition 7 décrit l'opérateur *fusion par union des nœuds*.

Définition 7 :

F-Union-Nœuds $(A_1, A_2) = A_3$



Entrées :

- $A_1 (E_1, r_1, N_1)$
- $A_2 (E_2, r_2, N_2)$

Conditions :

- $E_1 \cap E_2 \neq \varnothing$
- $\exists e_i \in E_1$ *et* $e_j \in E_2$ $\forall e_i \neq e_j$, *Fils* (e_i) = *Fils* (e_j) et *Parent* (e_i) = *Parent* (e_j).

Sortie :

- $A_3 (E_3, r_3, N_3)$

Avec $E_3 = (E_1 - \{e_i\}) \cup (E_2 - \{e_j\}) \cup (e_i | e_j)$, $r_3 = r_1 = r_2$ et $N_3 = N_1 \cup N_2$.
/* $(e_i | e_j)$ est le nœud ou */

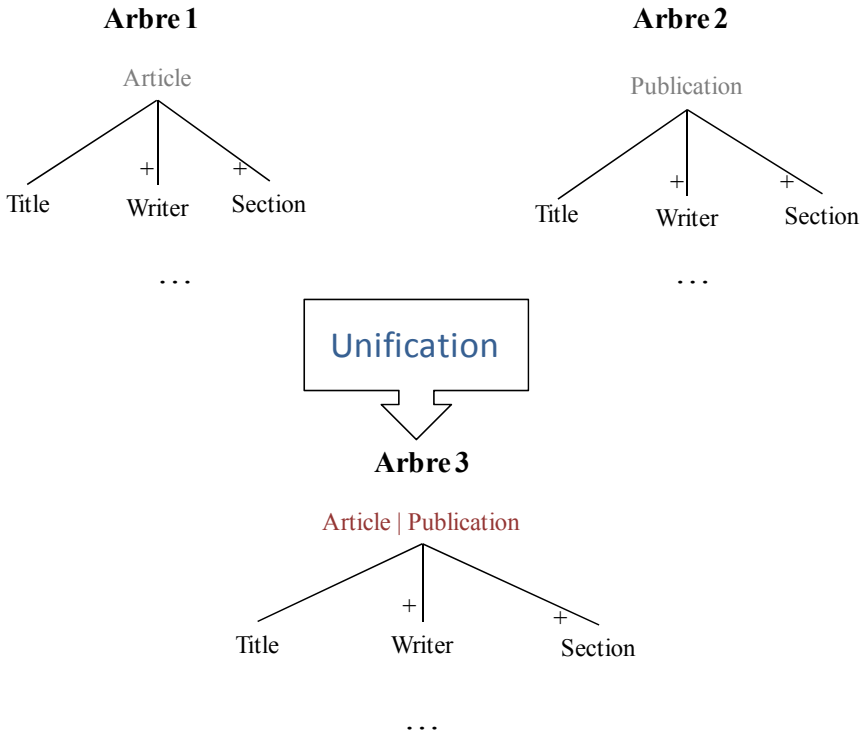


Figure 31 : *Exemple d'unification d'arbres fusionnés par union des nœuds.*

La *Figure 31* montre un exemple d'unification par union des nœuds. Les arbres en entrée : *Arbre1* et *Arbre2* sont décrits par des nœuds racines étiquetés par des noms différents et partageant le même nœud père. Par conséquent, l'arbre résultat est caractérisé par le nœud spécifique *ou* noté '|'.

Nous avons défini un algorithme nommé *Fusion-Arbre* pour produire des arbres unifiés. Cet algorithme utilise la fonction *Calculer-MS* pour évaluer la matrice de similarité, identifie les arbres à unifier et fusionne les arbres identifiés moyennant les trois opérateurs : *fusion par inclusion*, *fusion par union des sous-arbres* et *fusion par union des nœuds.*

```
Algorithme Fusion-Arbre
Entrée
n : Entier (nombre d'arbres)
T = {T1, T2 …, Ti …, Tn} : un ensemble non vide de n arbres
avec Ti = (Ei, ri, Ni) i=1..n
Seuil : Réel ∈ ]0,1[
Sortie
R = {R1, R2, Rm}: un ensemble non vide de m (m≤n) arbres
fusionnés.
Variables
MS (n, n) : Matrice de similarité
Max : Valeur maximale de la matrice
```

```
Début
R := T
MS := Calculer-MS (n, T) /* Calculer la matrice de similarité
*/
Pour i de 1 à (n-1) faire
  Max := Déterminer-Max (MS(i)) /* Déterminer la valeur
  maximale de la matrice */
  Si (Max ≥ Seuil) alors
    Pour j de i à n faire
      Si (MS(i, j) = Max) alors
        Marquer la ligne i et la colonne j
        Pour k de i à j faire
          Si (MS(k, j) = max) alors
            Marquer la ligne k
          Fin Si
        Fin Pour
      Fin Si
    Fin Pour
    R := R – {Arbres correspondants aux lignes et colonnes
    marquées}
    /* Fusion des arbres correspondants aux lignes et
    Colonnes marquées*/


    Si (Ti ⊆ Tj or Tj ⊆ Ti) Alors




      R := R ∪ F-Inclusion (Ti, Tj)


    Sinon


      Si (Ei ∩ Ej ≠ ∅ et (∃ ei ∈ Ei et ej ∈ Ej et ei = ej


      et Parent(ei) = Parent(ej) et Fils(ei) ≠ Fils(ej) ))
      Alors


        R := R ∪ F-Union-Sous-Arbres(Ti,Tj)


      Sinon


        Si (Ei ∩ Ej ≠ ∅ et ( ∃ ei ∈ Ei et ej ∈ Ej et
```

```
        Fils(ei) = Fils (ej) et Parent(ei) = Parent(ej) et
        ei ≠ ej)) Alors


          R := R ∪ F-Union-Nœuds(Ti,Tj)




        Fin Si
      Fin Si
    Fin Si
    Supprimer la ligne et la colonne marquées
    Calculer-MS (R)
  Sinon
    Arrêt
  Fin Si
Fin Pour
Fin.
```

Notons que la fusion des arbres prend en considération les cardinalités des nœuds des arbres et ceci en appliquant les dix règles définies dans (Hachaichi, et al., 2010). Ces règles remplacent les apparitions multiples d'un même élément par un seul élément ayant la cardinalité maximale. La *Figure 32* énumère ces règles.

e1*, e1*→ e1*
e1*, e1?→ e1*
e1?, e1*→e1*
e1?, e1?→ e1*
e1, e1→ e1+
e1+, e1+→ e1+
e1+, e1?→ e1+
e1?,e1+→ e1+
e1+,e1*→ e1+
e1*, e1+→ e1+

Figure 32 : *Règles traitant les cardinalités des arbres.*

3.5. Approbation des arbres unifiés

Dans cette étape, le décideur/concepteur intervient pour approuver les arbres unifiés issus de l'étape précédente de génération et ceci en tenant compte de ses besoins analytiques. En effet, il peut supprimer les nœuds qui ne présentent pas un intérêt ; il peut également modifier les noms des nœuds des arbres (*e.g.,* renommer le nœud *Article* par *Publication*). Les arbres résultats sont enregistrés dans le référentiel de la *Figure 27*.

3.6. Vérification des arbres unifiés

Afin de générer des arbres syntaxiquement corrects, nous avons défini dans (Aouabed, et al., 2012) quatre contraintes qui assurent la bonne formation des arbres résultats et qui aident ultérieurement à construire des modèles multidimensionnels corrects. Ces contraintes sont :

- Ct1 : Connexité,
- Ct2 : Hiérarchie,
- Ct3 : Existence du nœud racine, et
- Ct4 : Acyclicité.

Ct1. *Connexité* : La connexité garantit que chaque nœud appartenant à un arbre est lié à au moins un nœud.

Cette contrainte contrôle que chaque arbre ne comporte aucun nœud isolé. Ainsi, lors d'une analyse OLAP, le décideur peut utiliser tout élément multidimensionnel dans l'expression de son besoin. Rappelons que chaque nœud sera traduit comme étant un élément multidimensionnel dans la méthode de modélisation multidimensionnelle.

Ct2. *Hiérarchie* : La hiérarchie garantit que chaque nœud est lié à un et un seul nœud père à l'exception du nœud racine.

Cette contrainte assure l'obtention de structures arborescentes.

Ct3. *Unicité du nœud racine* : Elle contrôle que chaque arbre doit comporter un et un seul nœud racine.

Cette contrainte garantit qu'un arbre n'ayant pas des sous-arbres déconnectés. Elle garantit de générer plus tard des modèles multidimensionnels corrects, c'est-à-dire, ayant des éléments multidimensionnels tous connectés.

Ct4. *Acyclicité* : L'acyclicité garantit l'absence de cycles dans un arbre ; c'est-à-dire qu'un nœud ne peut pas être père et fils du même nœud par transitivité.

Cette contrainte permet la génération des modèles multidimensionnels ne comportant pas de cycles dans les hiérarchies et pouvant, par conséquent, être exploités par des opérations de forage.

3.7. Conclusion

Dans ce chapitre, nous avons présenté une méthode d'unification qui reçoit en entrée les structures des documents XML (DTD et/ou XSD) et génère un ou plusieurs structures unifiés.

Cette méthode utilise le formalisme de l'arbre. En fait, elle transforme chaque structure en un arbre et lève les ambiguïtés des noms des nœuds des arbres en se référant à un dictionnaire des acronymes et à la base de données lexicale *Wordnet*. En ce qui concerne l'unification des arbres, elle est réalisée en appliquant un ensemble de trois opérateurs que nous avons proposé : *fusion par inclusion*, *fusion par union des sous-arbres* et *fusion par union des nœuds*. Ces opérateurs prennent en considération la hiérarchie des nœuds des arbres. Afin d'aboutir à des structures unifiés (*i.e.,* arbres) valides, nous avons défini un ensemble de quatre contraintes : *Connexité*, *Hiérarchie*, *Existence du nœud racine*, et *Acyclicité*. Ces contraintes garantissent la bonne formation des structures unifiées et permettent de concevoir ultérieurement des modèles multidimensionnels valides.

Dans le chapitre suivant, nous proposons notre méthode semi-automatique pour la modélisation multidimensionnelle. Elle reçoit en entrée les arbres unifiés résultats de la méthode d'unification et génère pour chaque arbre un modèle multidimensionnel en galaxie.

CHAPITRE 4 : PROPOSITION D'UNE METHODE SEMI-AUTOMATIQUE DE MODELISATION EN GALAXIE

Résumé du chapitre :
Ce chapitre présente d'une part le modèle en galaxie et d'autre part notre méthode proposée de modélisation semi-automatique en galaxie et explique ses étapes. Cette méthode se base sur un ensemble de dix règles permettant l'extraction des éléments multidimensionnels (e.g., dimension, paramètre) à partir d'arbre unifié. De plus, elle vérifie la validité syntaxique des galaxies générées en utilisant le Dublin Core Metadata et un ensemble de onze contraintes.

Sommaire du chapitre 4

4.1. Introduction

La modélisation multidimensionnelle a pour objectif de concevoir des modèles multidimensionnels à des fins de traitements analytiques en ligne (« OLAP »). Dans ce chapitre, nous présentons notre méthode semi-automatique de modélisation multidimensionnelle en galaxie composée de quatre étapes : *Prétraitement des arbres*, *Génération de modèles en galaxie*, *Approbation de modèles en galaxie,* et *Vérification de* ces *modèles*. Notre processus reçoit en entrée les arbres unifiés (résultat de la méthode d'unification) et génère pour chaque arbre un modèle en galaxie.

Initialement, l'étape de prétraitement est déclenchée : il s'agit d'enrichir les extrémités des arcs de chaque arbre unifié par des cardinalités. Ensuite, des modèles en galaxie sont générés automatiquement en appliquant un ensemble de dix règles que nous avons définies. Puis, la validité de ces galaxies est vérifiée conformément à des contraintes. Finalement, ces modèles sont approuvés par le décideur.

Nous exposons, dans ce chapitre, notre méthode de modélisation. La section 2 est un aperçu des étapes de la méthode. La section 3 présente le modèle multidimensionnel en galaxie utilisé pour les documents. Les sections 4 à 7 détaillent les différentes étapes de la méthode.

4.2. Méthode de modélisation multidimensionnelle

Dans la littérature et conformément à notre étude de l'état de l'art (*cf.* section 2.4 du chapitre 2), il existe deux principaux modèles multidimensionnels pour les documents : le modèle en étoile et le modèle en galaxie.

Le modèle en étoile est basé sur la dualité fait-dimension où un fait modélise un sujet d'analyse préalablement défini. Alors que, le modèle en galaxie est plus simple puisqu'il est basé sur l'unique concept de dimension qui représente à la fois un axe et un sujet d'analyse plausible. Par conséquent, le sujet d'analyse n'est pas prédéfini à priori et sera spécifié au moment de l'interrogation du modèle. Vu les avantages du modèle en galaxie, nous le retenons pour modéliser l'entrepôt de documents.

Dans la suite de ce chapitre, nous décrivons notre méthode proposée pour générer des modèles en galaxie pour les documents XML. Cette méthode reçoit en entrée un ou plusieurs arbres unifiés et approuvés qui sont les résultats de la méthode semi-automatique d'unification des structures des documents XML. Puis elle génère, pour chaque arbre, un

modèle en galaxie (Ben Messaoud, et al., 2011b) (Feki, et al., 2013). Elle s'articule autour des quatre étapes suivantes :

- Prétraitement des arbres.
- Génération des modèles en galaxie.
- Approbation des modèles en galaxie.
- Vérification des modèles en galaxie.

La *Figure 33* illustre cette méthode.

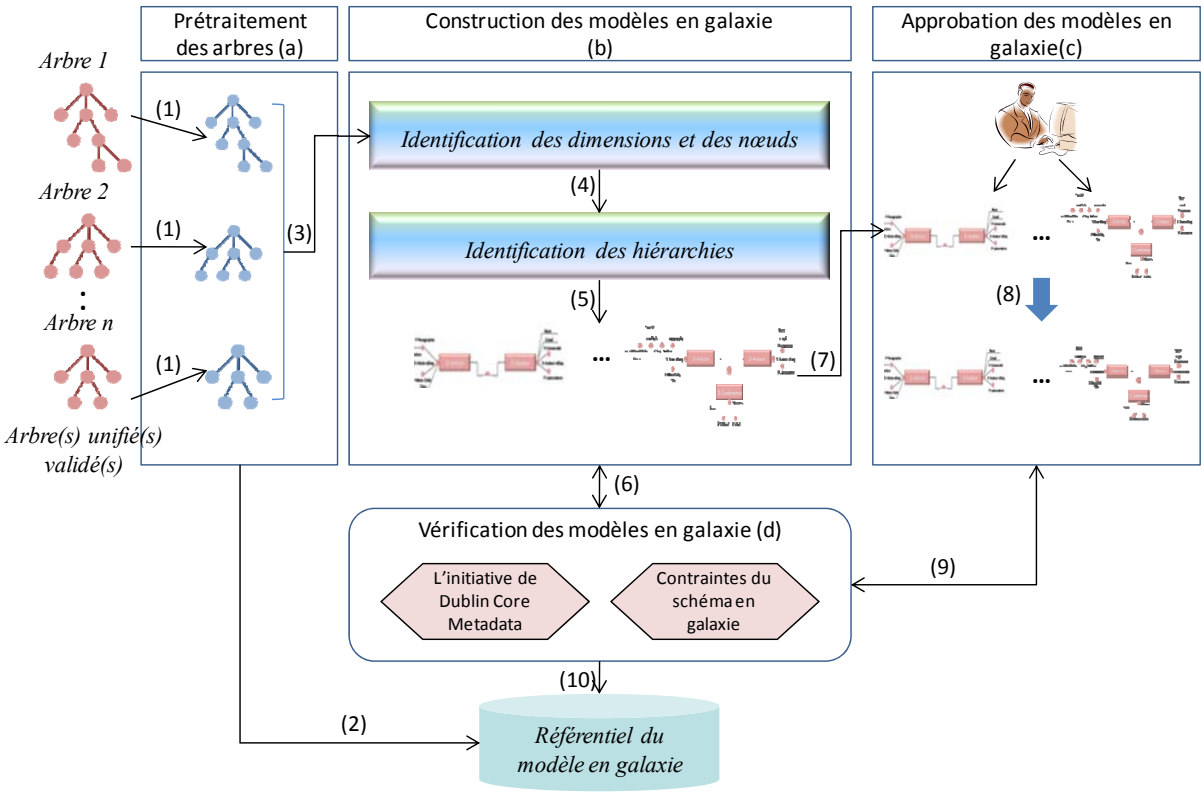


Figure 33 : *Méthode de construction de modèles en galaxie.*

Nous signalons que les éléments multidimensionnels des modèles en galaxie générés sont stockés dans le référentiel de la *Figure 34.b*. Ces éléments seront utiles, plus tard, pour l'interrogation des documents XML.

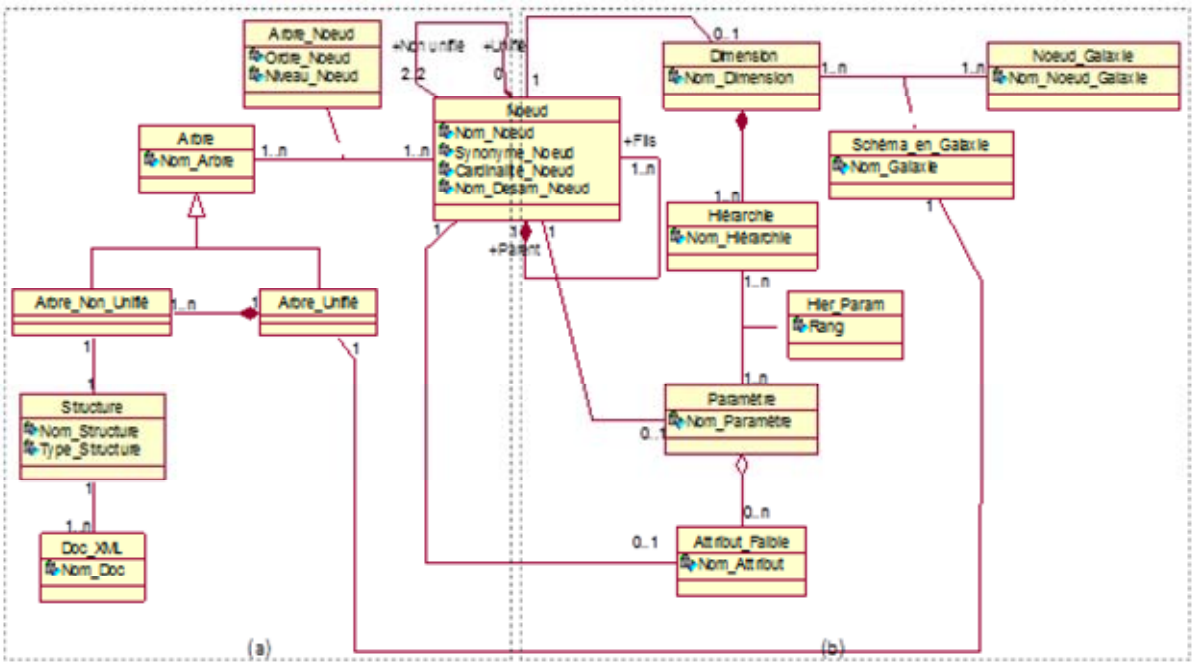


Figure 34 : *Référentiel des arbres et des modèles en galaxie.*

Pour la clarté de ce chapitre, nous présentons en premier lieu le modèle en galaxie, et nous détaillons en deuxième lieu les étapes de notre méthode proposée.

4.3. Modèle en galaxie

Le modèle en galaxie représente une généralisation du concept de constellation défini par (Kimball, 1997). Il est décrit par l'unique concept *« Dimension »* ; le concept *« fait »* est disséminé (*cf. Figure 24* du chapitre 2) parmi les dimensions. Ce modèle peut être vu comme un regroupement de dimensions interconnectées par un ou plusieurs nœuds où chaque nœud présente les dimensions compatibles pour une même analyse (Tournier, 2007) (Pujolle, et al., 2011).

Dans ce qui suit, nous présentons les concepts de la galaxie, c'est-à-dire, le concept dimension et le concept lien.

4.3.1. Concept de dimension

Une dimension (ou un axe d'analyse) modélise une perspective d'analyse. Elle est définie par un ensemble de paramètres organisés d'une manière hiérarchique, où chaque paramètre spécifie un indicateur d'analyse potentiel.

Dans une galaxie, deux types de dimensions peuvent être distingués :

- *Dimension non partagée :* connectée à un nœud.
- *Dimension partagée :* connectée à $n \geq 2$ nœuds.

Exemple : Dans la *Figure 35*, la dimension *Articles* est une dimension non partagée du fait qu'elle est liée à un seul nœud. Alors que, la dimension *Dates* est connectée à deux nœuds ; c'est une dimension partagée.

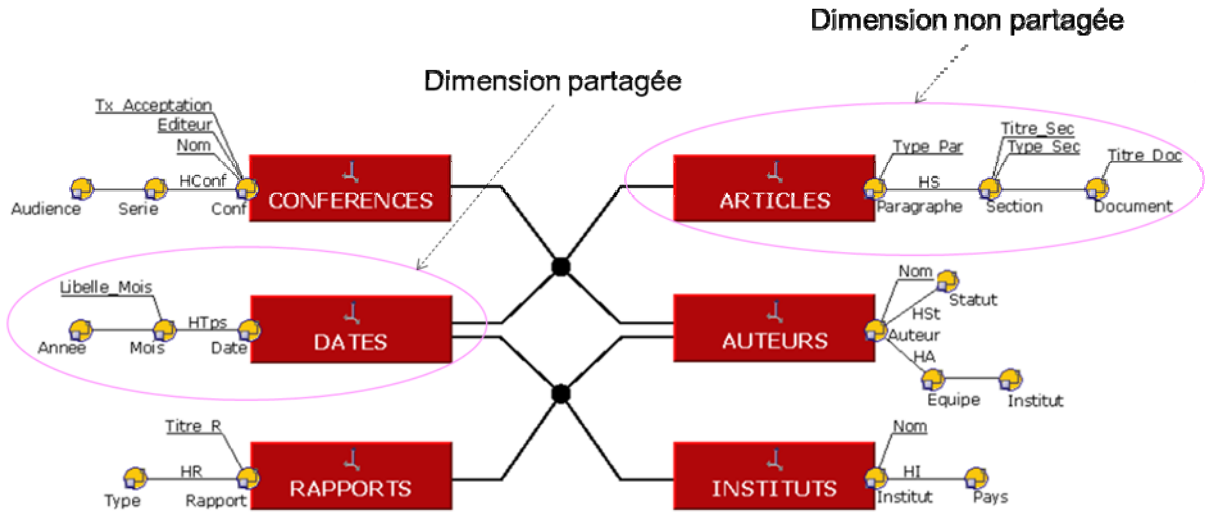


Figure 35 : *Exemples de dimensions partagée et non partagée.*

Chaque dimension de la *Figure 35* englobe un ensemble de paramètres (attributs) organisés de la granularité la plus fine vers la granularité la plus générale. Ces paramètres ordonnés forment une ou plusieurs hiérarchies. De plus, certains paramètres sont accompagnés par des descripteurs appelés *attributs faibles.* Ces attributs ont un rôle informationnel (Teste, 2000).

Toutes les hiérarchies des dimensions de la galaxie débutent par le paramètre le plus fin (c'est le paramètre identifiant de la dimension nommé paramètre racine) et se terminent par le paramètre de plus forte granularité ; le paramètre *All* est appelé paramètre extrémité. Comme ce paramètre est un paramètre artificiellement ajouté dans toute hiérarchie, il ne sera pas alors représenté graphiquement pour des raisons de simplification du modèle (Malinowski, et al., 2006).

Par exemple, la dimension *INSTITUTS* est formée des deux paramètres *Institut* (identifiant de la dimension) et *Pays*, et de l'attribut faible *Nom* associé au paramètre *Institut.* Les paramètres dans l'ordre *Institut* puis *Pays* forment une hiérarchie.

4.3.2. Concept de lien

Dans une galaxie, un lien est une liaison entre deux attributs de deux hiérarchies différentes ou de la même hiérarchie selon la relation *« correspond à »* entre les valeurs de ces deux attributs. Dans certains cas, ce lien peut être non matérialisé : Par exemple, il est très rare de trouver des liens qui relient les références d'un article scientifique et le contenu de

chaque article. Graphiquement, ces liens sont représentés par une flèche qui relie les deux paramètres en question. Un lien est annoté par un libellé descriptif. Exemple : Dans la *Figure 36*, le lien qui relie le paramètre *Institut* de la dimension *Auteurs* et le paramètre *Institut* de la dimension *INSTITUTS* est annoté par le libellé *Institut.* Il existe deux types de liens :

- *Lien intra-dimension :* relie deux paramètres d'une même dimension.
- *Lien inter-dimension :* connecte deux paramètres de deux dimensions différentes.

La *Figure 36* présente un exemple d'un lien inter-dimension. Ce lien relie les instituts des auteurs aux instituts qui pilotent des projets.

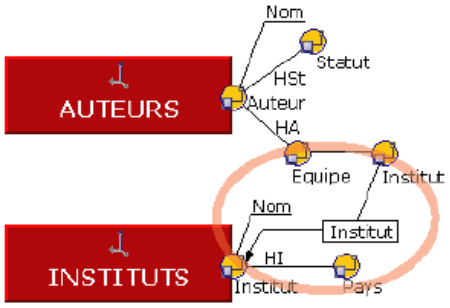


Figure 36 : *Exemple de lien inter-dimension.*

4.4. Prétraitement des arbres

Afin de faciliter la détermination des concepts multidimensionnels pour la génération de modèles en galaxie, l'étape de prétraitement génère, pour chaque arbre en entrée, un arbre correspondant à l'arbre source complété par des cardinalités pour chaque nœud père. Ces cardinalités sont ajoutées en examinant une collection de documents conformes aux structures initiales.

La *Figure 37* montre un exemple d'un arbre prétraité où des cardinalités sont ajoutées.

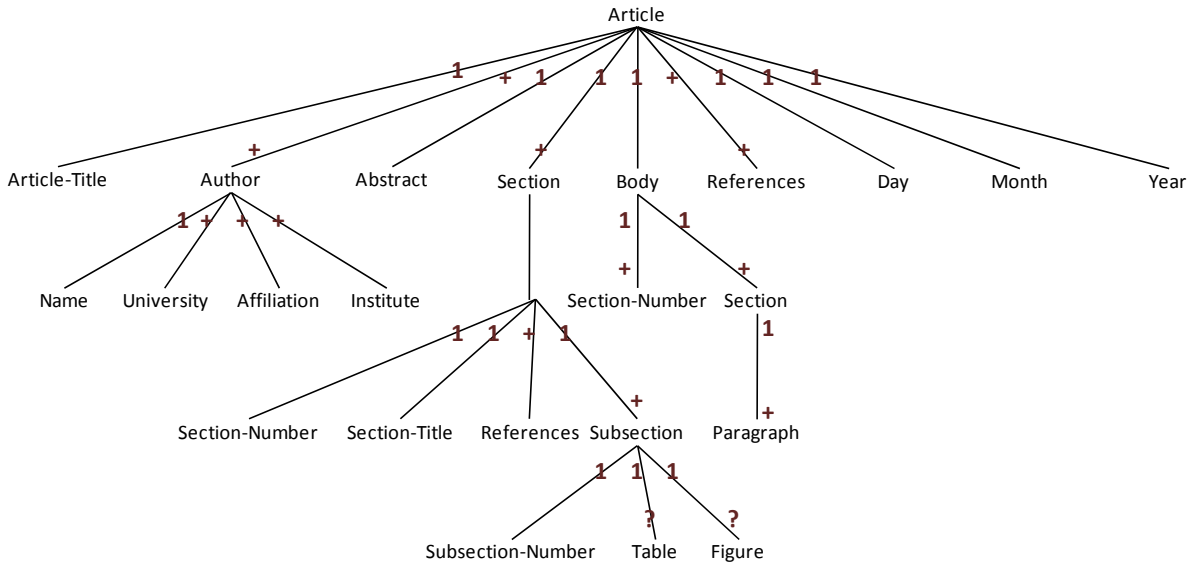


Figure 37 : *Exemple d'un arbre prétraité.*

4.5. Construction des modèles en galaxie

Cette construction transforme chaque arbre prétraité en un modèle en galaxie et ceci en appliquant un ensemble de dix règles. Elle est composée de deux sous-étapes :

- Identification des dimensions et des nœuds inter-dimensions.
- Identification des hiérarchies.

4.5.1. IDENTIFICATION DES DIMENSIONS ET DES NŒUDS INTER-DIMENSIONS

Détermination des dimensions : Nous avons défini trois règles ***Rd1*** à ***Rd3*** pour identifier les dimensions :

Rd1 :

Le nœud racine *r* d'un arbre devient une dimension nommée *D-r*.

Naturellement, le nœud racine est le nœud le plus générique d'un arbre. Par la suite, il ne peut représenter qu'une dimension.

La *Figure 38* montre un exemple de l'application de la règle *Rd1* où le nœud racine *Article* est transformé en une dimension nommée *D-Article.*

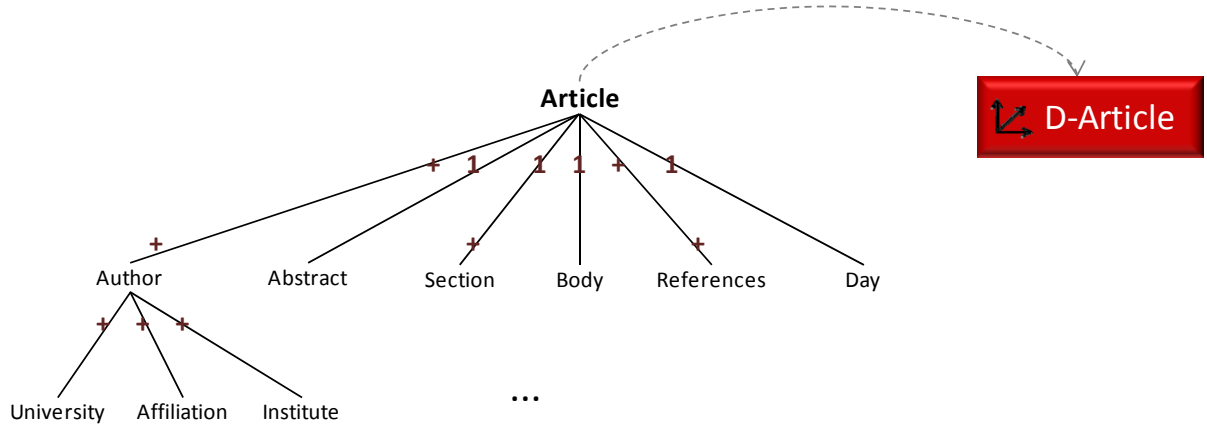


Figure 38 : *Exemple de l'application de la règle Rd1.*

Nous continuons à extraire les dimensions à partir des autres nœuds d'un arbre en définissant les deux règles ***Rd2*** et ***Rd3***.

Rd2 :

Chaque paire de nœuds non terminaux *M* et *N* telle que l'arc *M-N* est annoté avec les cardinalités + ou * des deux côtés, constituent deux dimensions nommées *D-M* et *D-N*.

Les cardinalités + ou * des deux côtés de l'arc *M-N* dénotent l'existence d'une association entre ces deux nœuds. Alors, nous considérons ces nœuds comme des dimensions.

La *Figure 39* présente un exemple de l'application de la règle *Rd2* : Les deux nœuds *Article* et *Author* sont liés par un arc annoté par les cardinalités + des deux côtés, par conséquent ils sont convertis respectivement en *D-Article* et *D-Author*. Ce qui permet, par exemple d'analyser le nombre d'articles pour chaque auteur.

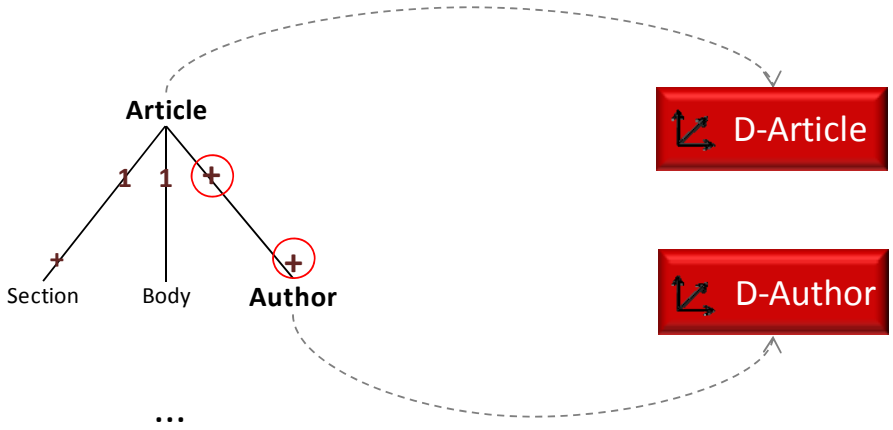


Figure 39 : *Exemple de l'application de la règle Rd2.*

Rd3 :

L'ensemble des nœuds décrivant le composant Date (*e.g.*, jour, mois) et ayant un même nœud père constitue une dimension temporelle nommée *D-Date* où ces nœuds seront ses paramètres.

Dans le contexte des entrepôts de données, la dimension temporelle est systématiquement présente (Kimball, 1997). Par conséquent, les nœuds décrivant une *date* sont organisés en hiérarchies et ceci en se référant à une dimension date standard.

Dans la *Figure 40,* les trois nœuds *Day*, *Month* et *Year* décrivent la composante *Date* ce qui engendre la détermination de la dimension *D-Date*.

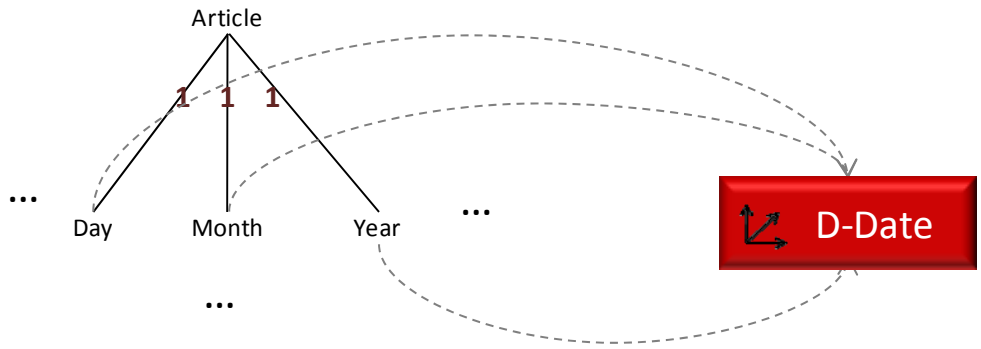


Figure 40 : *Exemple d'application de la règle Rd3.*

Détermination des dimensions compatibles : Dans un modèle en galaxie, un nœud dénote les dimensions compatibles pour une même analyse. Nous définissons la règle suivante pour identifier un nœud.

Rn :

Chaque paire de nœuds-dimensions *M* et *N* directement liés par un arc *M-N* dans l'arbre prétraité constituent deux dimensions compatibles.

Naturellement, puisque les nœuds identifiés comme dimensions sont liés dans l'arbre prétraité, alors ces dimensions méritent d'être connectées dans le modèle en galaxie.

4.5.2. IDENTIFICATION DES HIERARCHIES

Les paramètres d'une dimension sont organisés en une ou plusieurs hiérarchies selon la relation « *est plus fin* » conformément à leur niveau de détail (Teste, 2000). Toutes les hiérarchies d'une dimension *D* partent obligatoirement de l'identifiant de *D* qui représente le paramètre le plus fin. Dans nos travaux, nous définissons un identifiant de substitution (« Surrogate key ») pour chaque dimension générée.

Détermination des paramètres : Nous avons défini quatre règles pour identifier les paramètres d'une hiérarchie. Chaque paramètre est décrit par son nom et son rang.

Rp1 :

Tout nœud terminal *N* connecté à un nœud père *M* identifié comme une dimension dont l'arc *M-N* n'est pas de cardinalités 1 des deux côtés, représente un paramètre nommé *P-N* de rang 2.

Un nœud terminal *N* lié à un nœud-dimension *M* constitue un nœud paramètre. Nous écartons les nœuds terminaux tels que l'arc *M-N* est de cardinalités 1-1 parce que ces nœuds représentent des données descriptives du nœud père *M*.

La *Figure 41* est un exemple d'application de la règle *Rp1* ; le nœud *Institute* est connecté au nœud *Author* (identifié comme dimension) via un arc non annoté avec les cardinalités 1-1. Il est transformé en un paramètre de rang 2 nommé *P-Institute*.

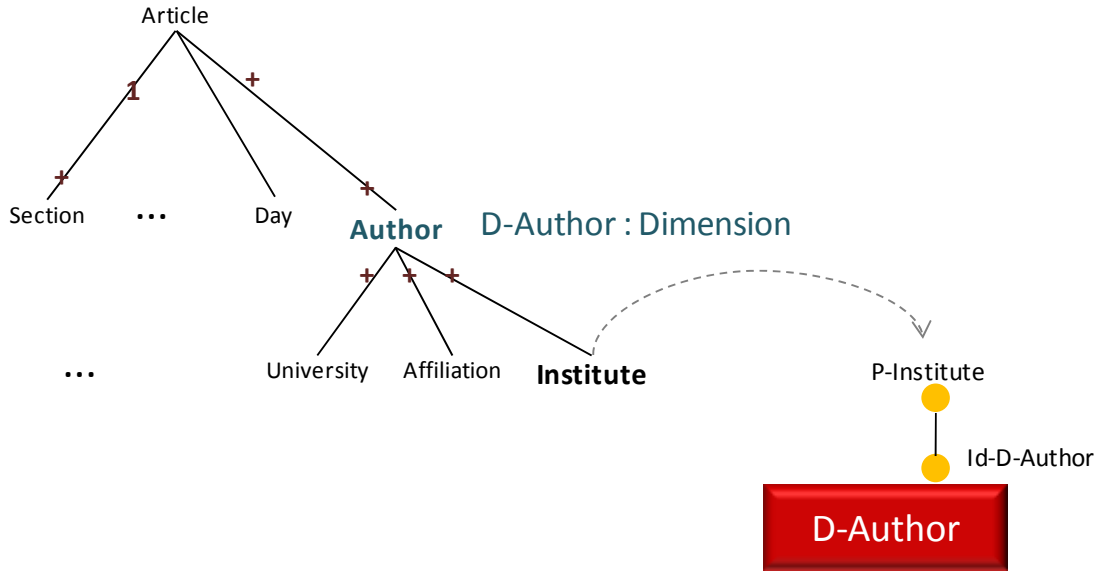


Figure 41 : *Exemple d'application de la règle Rp2.*

Rp2 :

Tout nœud terminal *N* connecté à un nœud père *M* identifié comme un paramètre de rang *i* dont l'arc *M-N* n'est pas de cardinalités 1 des deux côtés, représente un paramètre nommé *P-N* de rang (*i* – *1*).

Un nœud terminal *N* lié à un nœud paramètre *M* par l'arc *M-N* non annoté avec les cardinalités 1-1 représente un paramètre. Puisque, dans un arbre, chaque nœud père regroupe

les données élémentaires qui le décrivent, alors le nœud le plus profond dans l'arbre représente le paramètre de niveau le plus fin.

Nous poursuivons à dégager les paramètres des hiérarchies à partir des nœuds non terminaux avec les règles ***RP3*** et ***RP4***.

Rp3 :

Chaque nœud non terminal *N* connecté à un nœud père *M* identifié comme un paramètre de niveau *i* telle que l'arc *M-N* de cardinalités 1-(+ ou *), constitue un paramètre *P-N* de rang (*i-1*).

Rp4 :

Chaque nœud non terminal *N* connecté à un nœud père *M* identifié comme une dimension telle que l'arc *M-N* n'est pas de cardinalités + ou * des deux côtés, constitue un paramètre qui s'ajoute à la fin de la hiérarchie.

La *Figure 42* présente le résultat des règles *Rp2* à *Rp4*. La règle *Rp2* identifie le paramètre *P-Paragraph* de rang 2, *Rp3* détermine le paramètre *P-Section* de rang 3. Finalement, la règle *Rp4* extrait le paramètre *P-Body* comme un paramètre de rang 4.

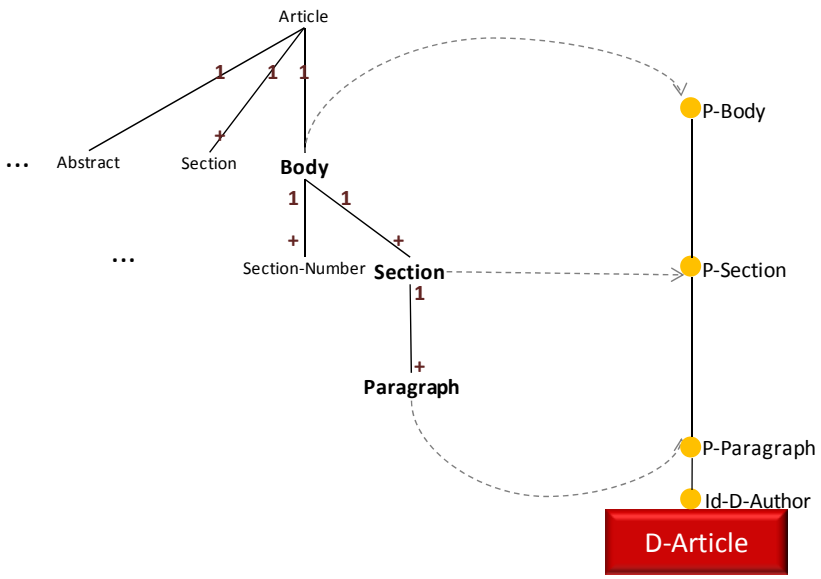


Figure 42 : *Exemple de l'application des règles Rp2, Rp3 et Rp4.*

Détermination des attributs faibles : Dans une dimension, les paramètres peuvent être accompagnés par des attributs faibles. Nous avons défini deux règles pour identifier les attributs faibles. Chaque attribut faible est caractérisé par son nom.

Ra1 :

Tout nœud terminal *N* connecté à un nœud père *M* identifié comme une dimension *D* avec l'arc *M-N* et de cardinalités 1-1 ou 1-0, constitue un attribut faible, du paramètre identifiant de *D,* nommé *W-N.*

Ra2 :

Tout nœud terminal *N* connecté à un nœud père *M* identifié comme un paramètre *P* avec l'arc *M-N* et de cardinalités 1-1 ou 1-0, constitue un attribut faible du paramètre *P* nommé *W-N.*

Naturellement, un nœud terminal *N* connecté à un nœud *M* avec l'arc *M-N* annoté avec les cardinalités 1-(0 ou 1) constitue une information descriptive du nœud *M*. Par conséquent, ces nœuds représentent des attributs faibles.

La *Figure 43* schématise l'application des règles de détermination des attributs faibles. La règle *Ra1* extrait l'attribut faible *W-Name* pour l'identifiant de la dimension *D-Author*, et la règle *Ra2* détermine *W-Section-Number* comme un attribut faible du paramètre *P-Section*.

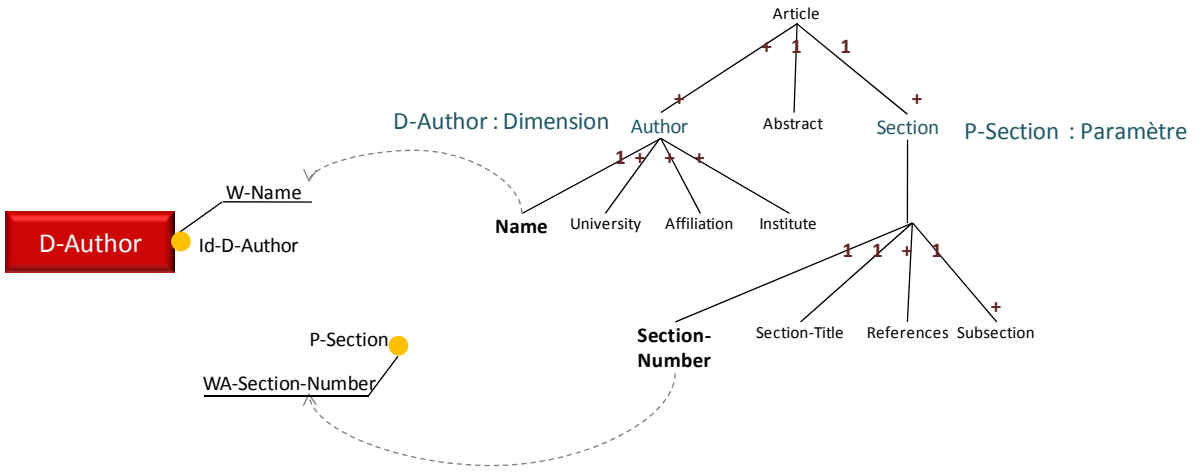


Figure 43 : *Exemple d'application des règles Ra1 et Ra2.*

L'application de l'ensemble des règles de détermination des concepts multidimensionnels sur l'arbre prétraité de la *Figure 37* produit le modèle en galaxie de la *Figure 44*.

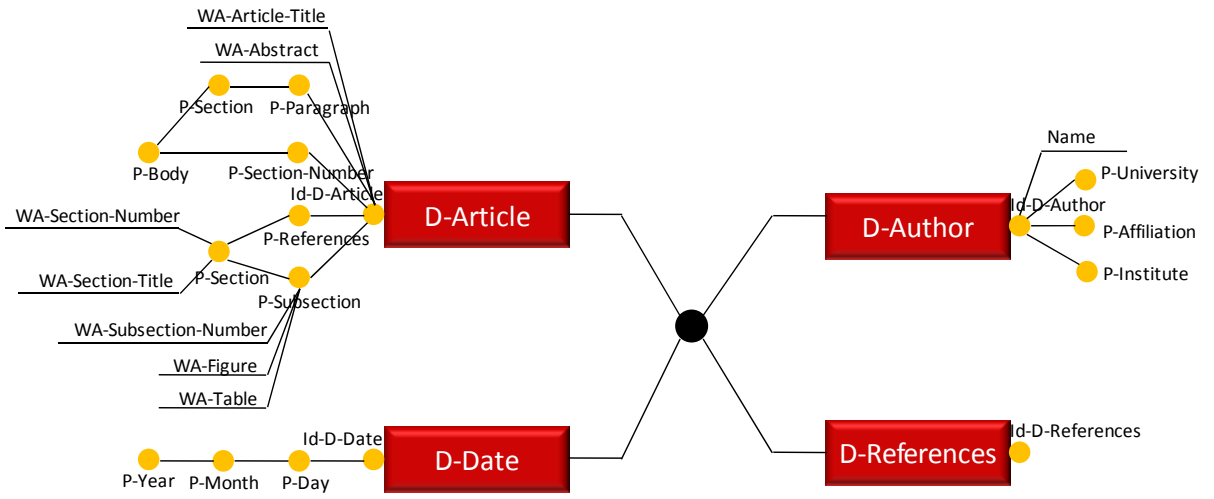


Figure 44 *: Modèle en galaxie correspondant à l'arbre de la Figure 37.*

A partir de ce modèle en galaxie, le preneur de décisions peut, par exemple, analyser les articles écrits par les auteurs et publiés au sein d'une conférence particulière.

4.6. Approbation du modèle en galaxie

Afin d'aboutir à des modèles en galaxie qui traduisent les besoins analytiques des décideurs, une étape de confirmation est indispensable. Pour ce faire, nous donnons la main au concepteur décisionnel pour effectuer les changements nécessaires permettant de prendre en considération les besoins analytiques conformément aux exigences des décideurs. Ainsi, il peut supprimer et/ou modifier les éléments multidimensionnels. Nous gardons trace de toutes les modifications qui seront enregistrées dans le référentiel de la *Figure 34.b*.

4.7. Vérification des modèles en galaxie

Dans certains cas, nous pouvons rencontrer, parmi les modèles en galaxie générés, certaines imperfections syntaxiques *(e.g.*, un modèle en galaxie comportant une seule dimension). Par conséquent, nous examinons ces modèles résultats issus des deuxième et troisième étapes. Pour ce faire, nous avons défini un ensemble de contraintes (Feki, et al., 2013). En outre, nous avons remarqué que parfois une dimension générée peut décrire à la fois la structure et les métadonnées des documents. Pour pallier à ce problème, nous utilisons le *Dublin Core Metadata Initiative* comme référence pour déterminer les hiérarchies décrivant les métadonnées du document et créer une nouvelle dimension appropriée pour ces hiérarchies.

Dans la littérature, certains travaux définissent des contraintes pour les modèles en étoile et en constellation (Ben Abdallah, et al., 2008) (Carpani, et al., 2001) (Ghozzi, 2004)

(Hurtardo, et al., 2002). Néanmoins, à notre connaissance, aucun travail ne s'est intéressé à définir des contraintes pour le modèle en galaxie.

En examinant les modèles en étoile et en galaxie, nous constatons quelques similitudes entre ces deux modèles. De ce fait, huit contraintes parmi celles définies pour un modèle en étoile (Ben Abdallah, et al., 2008) (Carpani, et al., 2001) (Ghozzi, 2004) (Hurtardo, et al., 2002) peuvent être adoptées pour la galaxie, et trois nouvelles contraintes seront spécifiées pour la galaxie. Nous classons l'ensemble de ces contraintes en trois classes :

- Contrainte de dimension.
- Contrainte de nœud.
- Contrainte de hiérarchie.

Dans ce qui suit, nous détaillons ces contraintes.

Contraintes de dimension :

Cd1. *Contrainte d'identifiant :* Toute dimension doit avoir un identifiant qui peut être :

- Une clé provenant de la source, ou
- Une clé de substitution (Hurtardo, et al., 2002) (Carpani, et al., 2001).

L'identifiant d'une dimension *D* est son attribut le plus fin. Pour garantir que les opérations de forages haut et bas puissent s'exécuter sur une dimension, toute dimension doit posséder au moins une hiérarchie. D'où la contrainte suivante.

Cd2. *Dimension non vide :* Toute dimension doit avoir au moins une hiérarchie.

Dans une galaxie, il existe deux types de dimensions :

- *Dimension non partagée :* connectée à n=1 nœud.
- *Dimension partagée :* connectée à n≥2 nœuds.

Il s'ensuit la contrainte suivante :

Cd3. *Dimension non isolée :* Toute dimension doit être liée *n* ($n \geq 1$) nœuds.

Cette contrainte est spécifique au modèle en galaxie. Elle garantit que toutes les dimensions connectées peuvent participer dans l'expression d'une analyse multidimensionnelle. En effet, une dimension non connectée est sans intérêt.

Par ailleurs, les nœuds d'une galaxie sont sujets à des contraintes.

Contraintes de nœud :

Cn1. *Nœud non isolé :* Tout nœud doit être connecté à au moins deux dimensions distinctes.

L'expression d'un besoin décisionnel nécessite au moins deux dimensions liées ; l'une joue le rôle du sujet d'analyse et l'autre joue le rôle d'axe d'analyse.

Cn2. *Disjonction des nœuds :* Aucun lien direct ne peut exister entre deux nœuds quelconques.

Un nœud relie les dimensions compatibles pour une même analyse. Par conséquent, les nœuds d'une même galaxie ne peuvent pas être connectés entre eux.

Rappelons que les hiérarchies permettent le forage haut et bas. Nous leurs associons des contraintes.

Contraintes de hiérarchies :

Ch1. *Racine hiérarchique :* Toutes les hiérarchies d'une même dimension partent du même paramètre le plus fin : identifiant de la dimension (Ben Abdallah, et al., 2008).

Cette contrainte garantit que toutes les opérations de forage vers le bas suivant toute hiérarchie d'une dimension *D* peuvent atteindre le même niveau le plus détaillé.

Ch2. *Exclusivité de la hiérarchie minimale :* Toute dimension ayant une hiérarchie minimale ne doit pas avoir d'autre hiérarchie (Ben Abdallah, 2010).

Cette contrainte assure la construction de dimension n'ayant pas une hiérarchie minimale conjointement avec d'autres, autrement elle sera inutile.

Ch3. *Attribut non isolé :* Chaque attribut d'une dimension *D* doit appartenir à au moins une hiérarchie de *D* (Ben Abdallah, 2010).

Cette contrainte assure que chaque attribut a un niveau d'analyse.

Ch4. *Hiérarchie non vide :* Une hiérarchie d'une dimension *D* doit connecter au moins deux paramètres : *Id* et *All* (Ben Abdallah, 2010).

Une telle hiérarchie est dites minimale ; elle assure l'agrégation sur sa dimension.

Ch5. *Connexion vers le haut :* Tous les paramètres d'une hiérarchie, sauf *All*, possèdent au moins un père (Hurtardo, et al., 2002) (Ghozzi, 2004).

A travers cette contrainte, lors d'une analyse, nous pouvons passer d'un niveau de détail à un autre.

Ch6. *Acyclicité :* Un paramètre ne peut pas être père et fils du même paramètre par transitivité, sauf *All* (Hurtardo, et al., 2002) (Ghozzi, 2004).

Cette contrainte inhibe les opérations de forage haut et bas récursifs et infinis.

4.8. Conclusion

Dans ce chapitre, nous avons présenté une méthode de modélisation multidimensionnelle. Cette méthode a le mérite de traduire automatiquement chaque arbre issu de la méthode d'unification en un modèle en galaxie validé par rapport à des contraintes. Le choix du modèle en galaxie est motivé par sa simplicité et sa flexibilité puisque le sujet d'analyse est déterminé au moment de l'interrogation de la galaxie et non prédéterminé (comme dans le cas des modèles en étoile).

Pour les besoins de la construction automatique de modèles en galaxie, nous avons défini un ensemble de *dix règles d'extraction* des concepts multidimensionnels : trois règles pour l'extraction des dimensions, une pour les nœuds inter-dimensions, quatre règles pour l'extraction des paramètres (*i.e.*, niveaux d'analyses) et deux autres pour les attributs faibles.

Afin de permettre à ces règles de produire de bons résultats, nous avons enrichi chaque arbre (résultat de l'unification) par des cardinalités. Ces cardinalités sont extraites en examinant une collection de documents XML conformes aux DTDs et/ou XSDs initiales. Finalement, ces modèles sont vérifiés syntaxiquement via un ensemble de onze contraintes dont huit sont héritées de la littérature du domaine multidimensionnel et trois que nous avons définies spécifiquement pour le modèle en galaxie (Feki, et al., 2013). L'ensemble des modèles ainsi obtenus sont soumis au décideur pour approbation.

CHAPITRE 5 : OUTILS DEVELOPPES

Résumé du chapitre :
*Ce chapitre décrit les fonctionnalités des deux outils développés dans ce mémoire de thèse : USD (**U**nification of **S**tructures of XML **D**ocuments) et Galaxy-Gen (**Galaxy Gen**eration) supportant respectivement les deux méthodes d'unification des structures des documents XML et la modélisation en galaxie.*

Sommaire du chapitre 5

5.1. Introduction

Afin de valider notre approche de construction du schéma de l'entrepôt de documents, nous avons développé deux outils logiciels : ***USD*** « *Unification of Structures of XML Documents* » pour l'unification des structures des documents XML, et ***Galaxy-Gen*** « *Galaxy Generation* » pour la génération semi-automatique de modèles en galaxie.

L'outil ***USD*** reçoit en entrée un ensemble de structures de documents XML appartenant à un même domaine et génère une ou plusieurs structures unifiées. Pour ce faire, il transforme chaque structure en arbre. Ensuite, il résout les problèmes de synonymie et d'acronyme en utilisant respectivement la base lexicale *Wordnet* et un dictionnaire des acronymes. Puis, il calcule les similarités entre ces arbres et les fusionne jusqu'à atteindre un seuil. Finalement, il génère la structure unifiée arborescente et la visualise graphiquement.

L'outil ***Galaxy-Gen*** a pour objectif de générer, à partir de la structure unifiée, un modèle multidimensionnel en galaxie. Pour ce faire, il identifie automatiquement les éléments de la galaxie : les axes d'analyse, les nœuds inter-dimensions et les paramètres des hiérarchies. De plus, il permet d'approuver et de visualiser graphiquement le modèle en galaxie.

Le reste de ce chapitre décrit ces deux outils. La section 2 présente l'environnement de réalisation de ces deux outils. La section 3 détaille l'architecture et les différentes interfaces de l'outil ***USD*** traitant la méthode d'unification. Quant à la section 4, elle s'intéresse à l'architecture de l'outil ***Galaxy-Gen*** et à ses interfaces.

5.2. Environnement et outils de réalisation

Pour le développement des deux outils *USD* et *Galaxy-Gen*, nous avons utilisé :

Eclipse : C'est un EDI (Environnement de Développement Intégré) qui simplifie la programmation. Il permet de créer, compiler, exécuter et déboguer les programmes Java. Il présente pareillement des techniques avancées pour la visualisation du code et la création des interfaces[12].

Oracle : Nous avons utilisé le Système de Gestion de Base de Données Relationnel Oracle 10g pour la :

- définition et la manipulation des données,
- cohérence des données,

[12] http://www.wikipedia.org/.

- confidentialité des données,
- intégrité des données, et
- sauvegarde et restitution des données.

Wordnet : C'est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton (A. Miller, 1995). Elle permet de répertorier, classifier et mettre en relation, de diverses manières, le contenu sémantique et lexical de la langue anglaise. Il existe des versions de *Wordnet* pour d'autres langues, mais la version anglaise est la plus complète[13].

XMLSpy : Edité par la société *Altova, XMLSpy* est un environnement de développement des documents XML[14]. Il permet l'édition et la validation des documents XML, ainsi que leurs XSDs et DTDs associés. De plus, il offre la fonctionnalité de génération automatique des structures (DTDs et XSDs) pour des documents XML. Nous l'exploitons pour cette fonctionnalité

Les parseurs DTDParseur et XSOM : Pour contrôler et vérifier la validité syntaxique des structures des documents XML, nous avons utilisé :

- *DTDParser :* qui est un composant exploitant une API *(« Application Programming Interface »).* Nous l'avons utilisé afin de vérifier qu'une DTD est syntaxiquement conforme à la norme W3C.
- *XSOM (« XML Schema Object Model ») :* C'est un composant assurant les mêmes fonctionnalités que *DTDParser* mais opérant sur les XSDs.

5.3. *USD* : Un outil d'unification des structures des documents XML

*USD « **U**nification of **S**tructures of XML **D**ocuments »* met en œuvre les quatre étapes de la méthode d'unification des structures des documents XML : a) *Représentation arborescente*, b) *Génération des arbres unifiés,* c) *Approbation des arbres unifiés,* et d) *Vérification des arbres* (Ben Messaoud, et al., 2011a) (Ben Messaoud, et al., 2012). (*cf.* section 3.2 du chapitre 3).

5.3.1 ARCHITECTURE DE *USD*

USD comporte les cinq modules suivants :

[13] http://wordnet.princeton.edu/
[14] http://www.altova.com/xml-editor/.

- *Représentation arborescente* : Ce module transforme chaque structure XML en entrée en un arbre. Les nœuds de l'arbre représentent soit les éléments, soit les attributs de la structure XML et les arcs indiquent les relations existantes entre ces nœuds.
- *Génération des arbres unifiés* : Ce module génère un ou plusieurs arbres unifiés à partir des arbres issus du module précédent et ceci en appliquant un ensemble de trois opérateurs que nous avons proposé : *fusion par inclusion*, *fusion par union des sous-arbres* et *fusion par union des nœuds* (*cf.* section 3.4.3 du chapitre 3).
- *Approbation des arbres unifiés* : Il donne la main au concepteur décisionnel/décideur pour approuver les arbres unifiés résultats, selon ses besoins d'analyses.
- *Vérification des arbres* : Il vérifie la validité syntaxique des arbres résultats des deux modules *Représentation arborescente* et *Vérification des arbres* à travers les quatre contraintes que nous avons définies : *Connexité*, *Hiérarchie*, *Existence du nœud racine*, et *Acyclicité (cf. section 3.6 du chapitre 3)*.
- *Affichage* : Il permet de visualiser graphiquement les arbres unifiés résultats.

La *Figure 45* décrit l'architecture de l'outil *USD*.

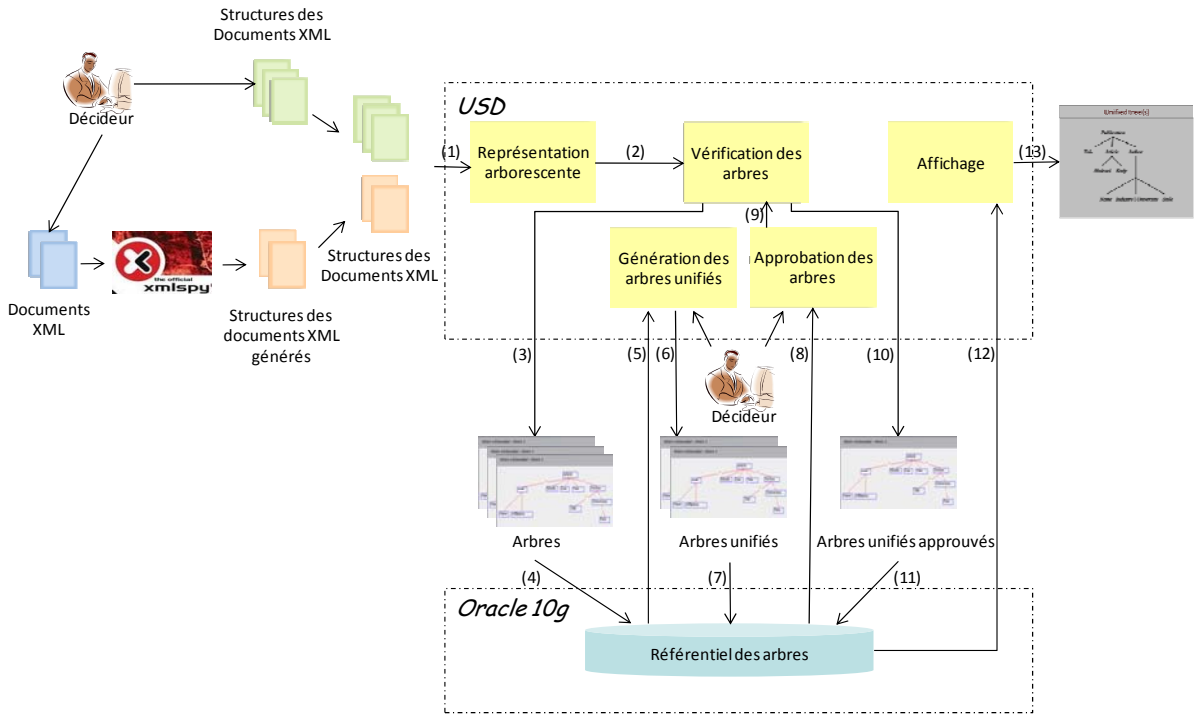


Figure 45 : *Architecture de l'outil USD.*

5.3.2 INTERFACES DE *USD*

Dans cette sous-section, nous présentons les différentes fonctionnalités de l'outil *USD* en les illustrant à travers l'unification de quatre structures de documents XML : deux DTDs

inspirées de (Mello, et al., 2005) (*cf. Figure 46* et *Figure 47*) et deux XSDs (*cf. Figure 48* et *Figure 49*) qui ressemblent aux XSDs présentés dans (Zhang, et al., 2002).

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Author (Address, (University|Entreprise|Office), Article+)>
<!ELEMENT Article (#PCDATA)>
<!ELEMENT University (#PCDATA)>
<!ELEMENT Entreprise (#PCDATA)>
<!ELEMENT Office (#PCDATA)>
<!ELEMENT Address (ZipCode, Street, City)>
<!ELEMENT ZipCode (#PCDATA)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ATTLIST Author Style (Science|Romance|Drama|Comedy) #REQUIRED>
```

Figure 46 : *Exemple de DTD "DTD 1".*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Paper (Year, Journal, Title, Writer+)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Journal (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Writer (Address+, (University|Research-
Institute|Business|Office))>
<!ELEMENT University (#PCDATA)>
<!ELEMENT Research-Institute (#PCDATA)>
<!ELEMENT Business (#PCDATA)>
<!ELEMENT Office (#PCDATA)>
<!ELEMENT Address (Country, City)>
<!ELEMENT Country (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ATTLIST Writer WrittingStyle (Science | Art) #REQUIRED>
```

Figure 47 : *Exemple de DTD "DTD 2".*

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
      <xs:element name="Writer">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Name"/>
                        <xs:element ref="Address"/>
                        <xs:choice>
                              <xs:element ref="University"/>
                              <xs:element ref="Industry"/>
                        </xs:choice>
                  </xs:sequence>
                  <xs:attribute name="Style" use="required">
                        <xs:simpleType>
                              <xs:restriction base="xs:NMTOKEN">
```

```
                                    <xs:enumeration value="drama"/>
                                    <xs:enumeration value="romance"/>
                                    <xs:enumeration value="fiction"/>
                              </xs:restriction>
                        </xs:simpleType>
                  </xs:attribute>
            </xs:complexType>
      </xs:element>
      <xs:element name="Address">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="ZipCode"/>
                        <xs:element ref="City"/>
                  </xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="ZipCode">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="City">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Name">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="University">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Industry">
            <xs:complexType mixed="true"/>
      </xs:element>
</xs:schema>
```

Figure 48 : *Exemple de XSD "XSD 1".*

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
      <xs:element name="Paper">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Tit"/>
                        <xs:element ref="Writer" maxOccurs="unbounded"/>
                        <xs:element ref="Article"/>
                        <xs:element ref="Year"/>
                        <xs:element ref="Journal"/>
                  </xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Tit">
            <xs:complexType mixed="true"/>
      </xs:element>
```

```
      <xs:element name="Writer">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Name"/>
                        <xs:element ref="University" minOccurs="0"
maxOccurs="unbounded"/>
                  </xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Name">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="University">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Article">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Abstract"/>
                        <xs:element ref="Body"/>
                  </xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Abstract">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Body">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Year">
            <xs:complexType mixed="true"/>
      </xs:element>
      <xs:element name="Journal">
            <xs:complexType mixed="true"/>
      </xs:element>
</xs:schema>
```

Figure 49 : *Exemple de XSD "XSD 2".*

Initialement, l'utilisateur choisit les structures des documents XML qu'il souhaite unifier à travers l'interface de la *Figure 50* ; au moins deux structures sont nécessaires pour lancer l'unification. Notons que nous permettons à l'utilisateur de corriger son choix en supprimant une ou plusieurs structures de l'ensemble des structures choisies. De plus, il peut sélectionner un document XML. Dans ce cas, USD appelle automatiquement l'outil *XMLSpy* pour générer une DTD ou un XSD pour ce document et ceci selon la préférence de l'utilisateur.

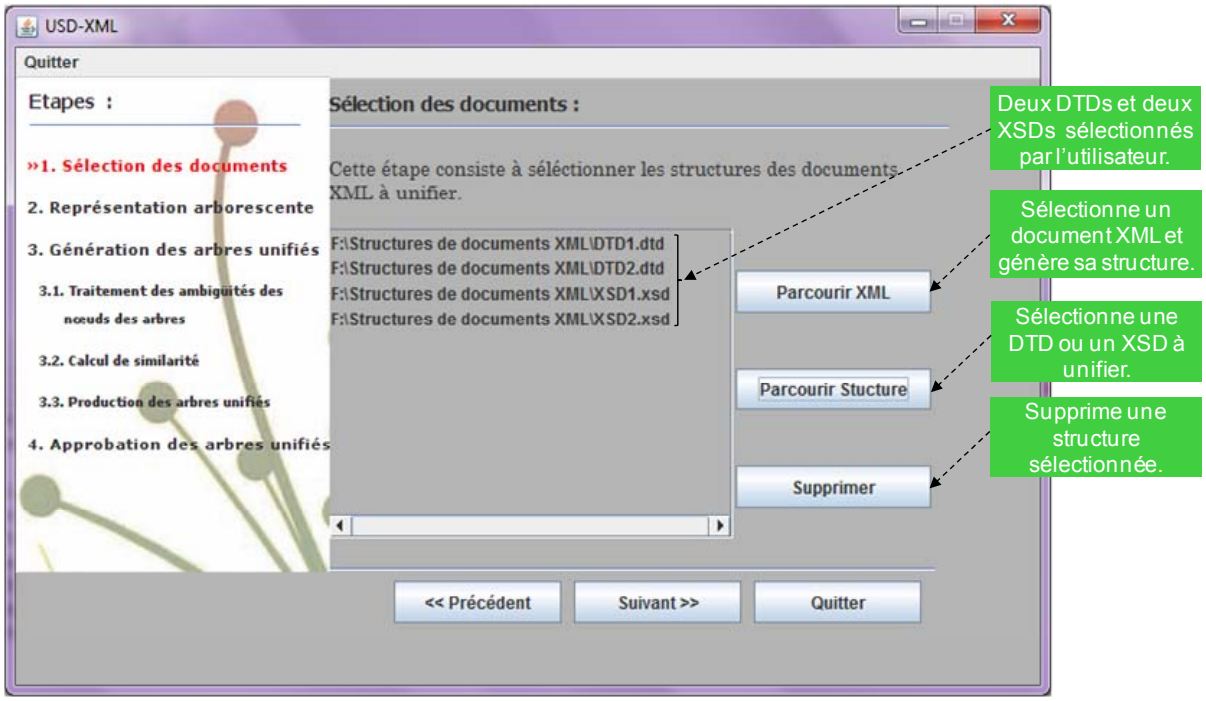


Figure 50 : *Interface de sélection des structures des documents XML à unifier.*

L'unicité des structures sélectionnées est systématiquement contrôlée et l'utilisateur est averti de l'erreur éventuelle pour corriger.

Représentation arborescente :

La sélection des structures s'enchaîne par une représentation graphique sous forme d'arbre. Les *Figure 51* à *Figure 54* montrent respectivement les arbres *Arbre1* à *Arbre4* correspondants aux quatre structures des *Figure 46* à *Figure 49* respectivement.

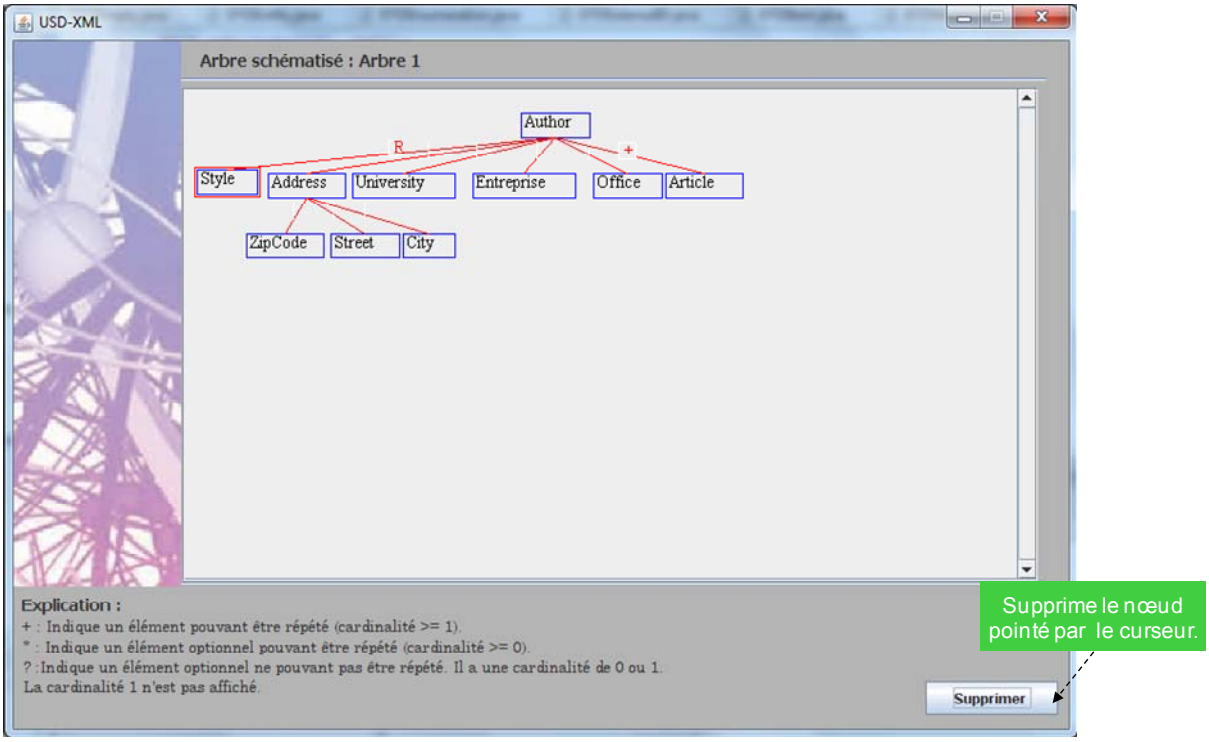


Figure 51 : *Arbre1 : Représentation arborescente de la DTD1.*

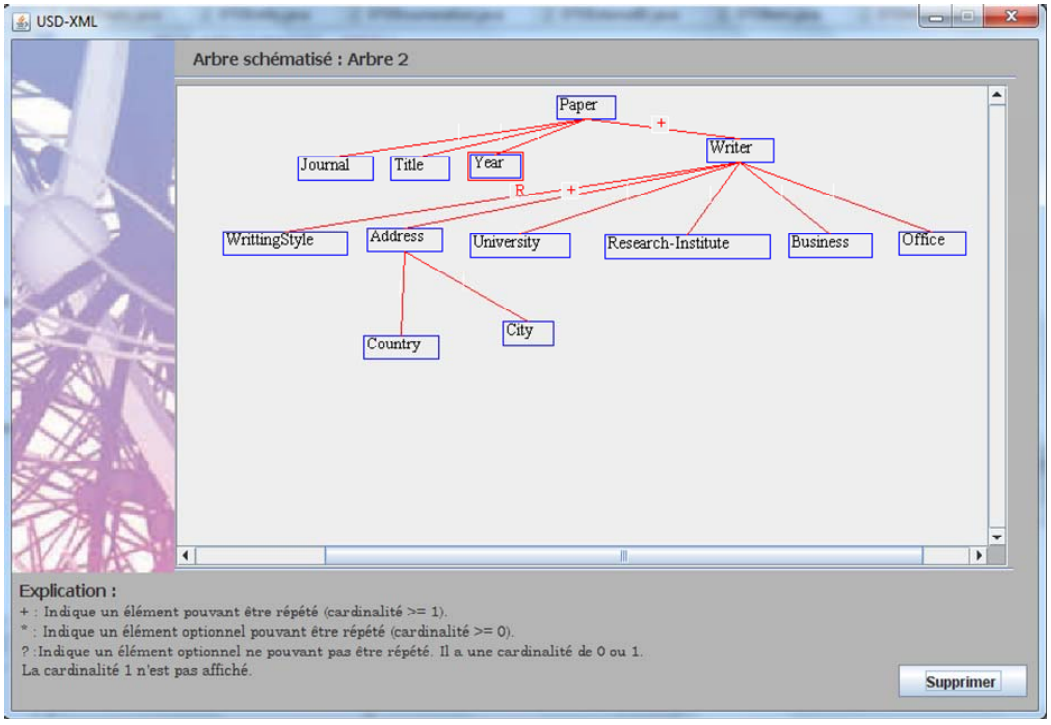


Figure 52 : *Arbre2 : Représentation arborescente de la DTD2.*

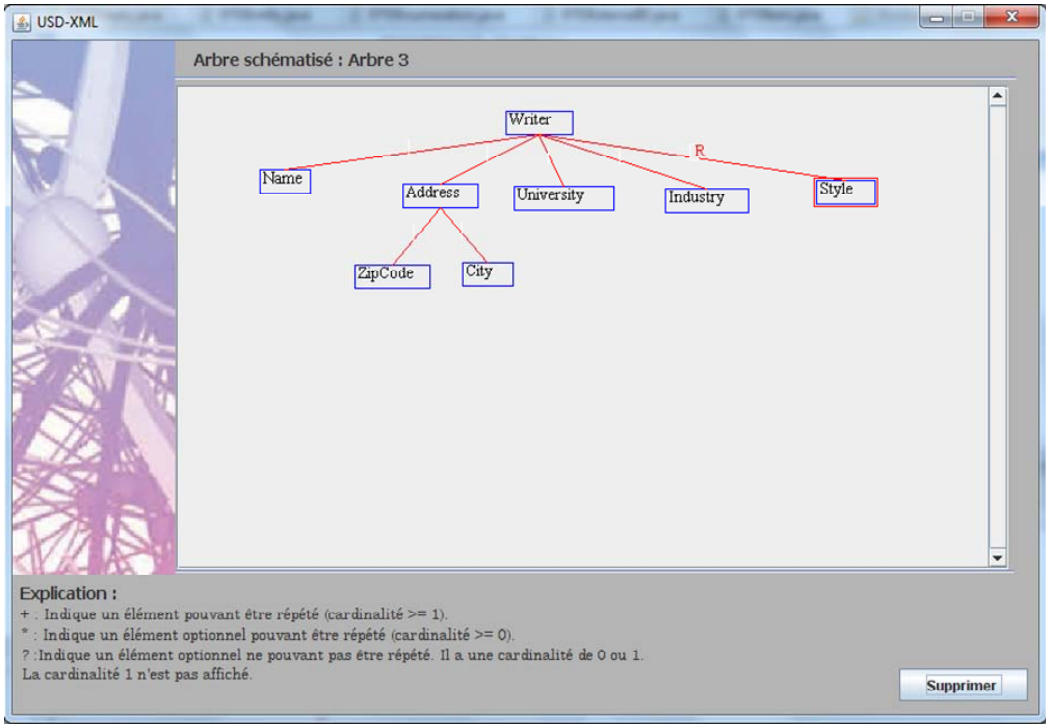


Figure 53 : *Arbre3 : Représentation arborescente du XSD1.*

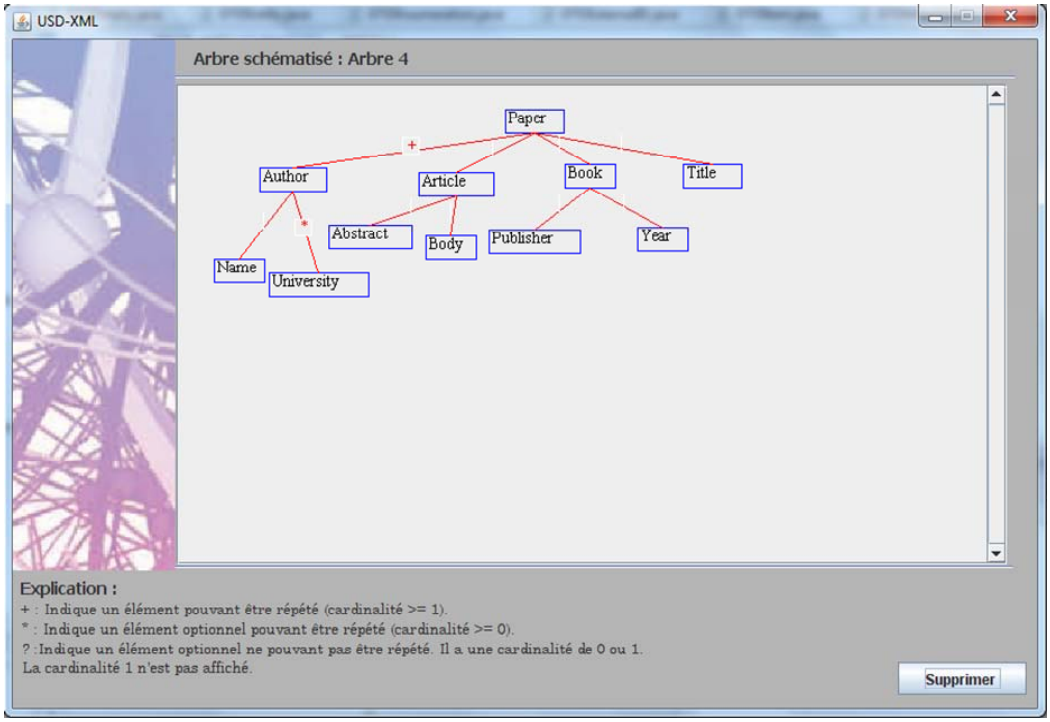


Figure 54 : *Arbre4 : Représentation arborescente du XSD2.*

Signalons qu'*USD* donne la main à l'utilisateur pour supprimer les nœuds jugés inutiles, c'est-à-dire les nœuds qui ne décrivent pas ses besoins analytiques.

Génération des arbres unifiés :

Après représentation graphique et enregistrement des arbres dans le référentiel de la *Figure 27*, le traitement des ambiguïtés des noms est déclenché.

Premièrement chaque nœud dont le nom est un acronyme est remplacé par sa forme complète, c'est-à-dire son nom correspondant et ceci en consultant un dictionnaire des acronymes. La *Figure 55* est l'interface résultat du traitement des acronymes des quatre arbres. En fait, le nœud *Tit* de l'*Arbre4* est remplacé par *Title*.

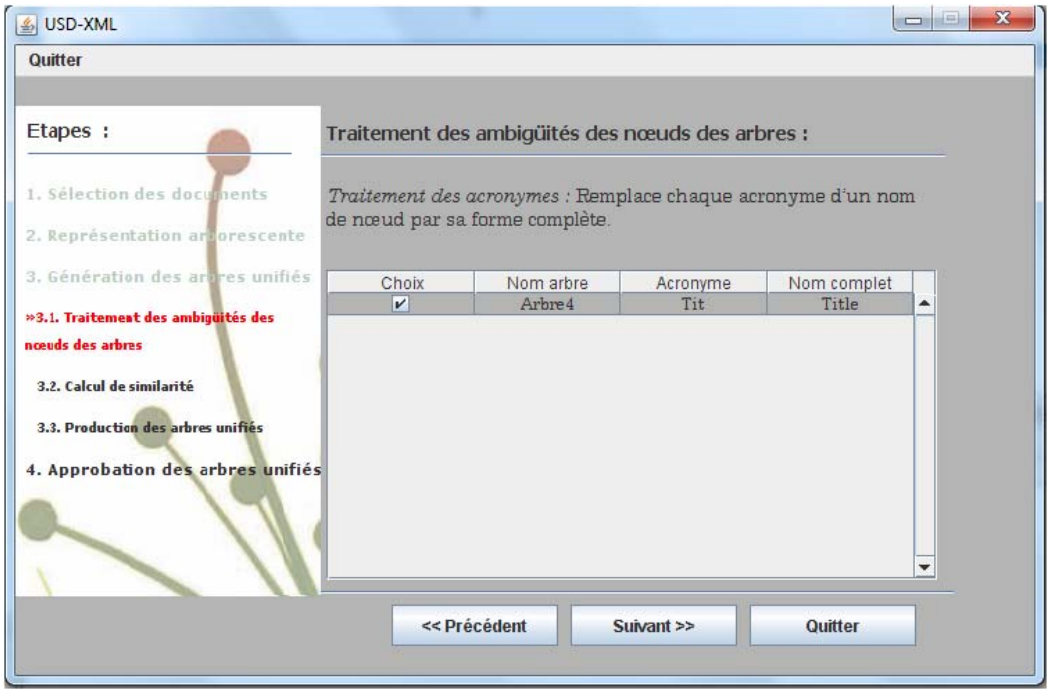


Figure 55 : *Résultat du traitement des acronymes des nœuds.*

Les synonymes sont déterminés en accédant à la base lexicale *Wordnet*. Le nœud *Author* de l'*Arbre1* est remplacé par son synonyme *Writer* qui représente le mot le plus fréquent dans les structures des documents XML en entrée. Le résultat de ce traitement est illustré par l'interface de la *Figure 56*.

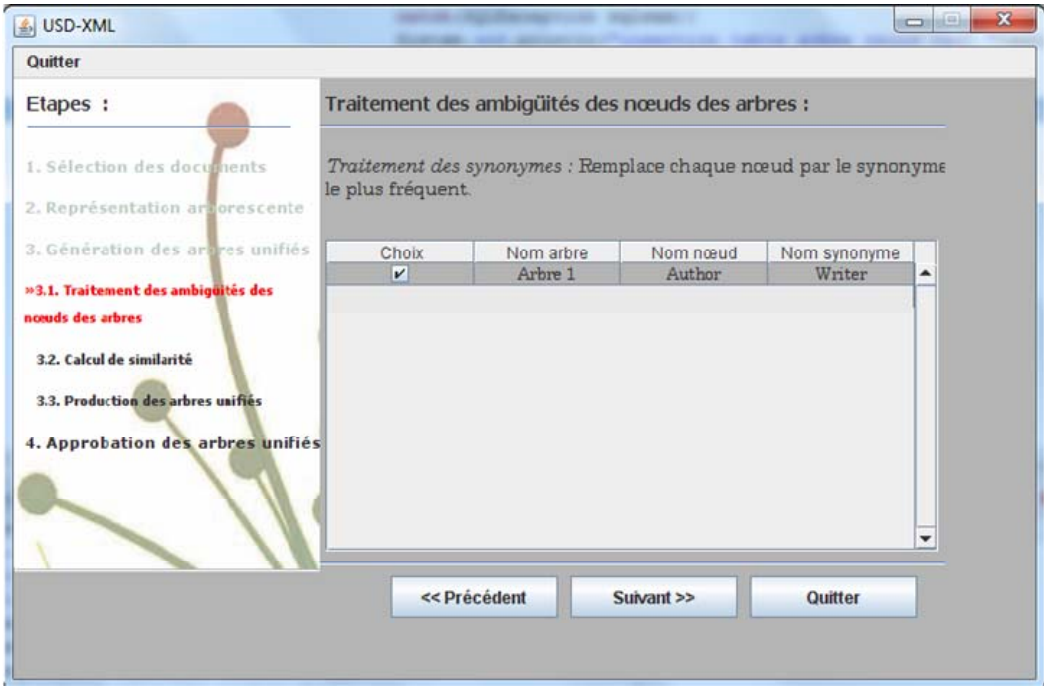


Figure 56 : *Résultat du traitement des synonymes des nœuds.*

Pour garantir l'unicité des noms des nœuds, les ambiguïtés des noms sont résolues. En fait, *USD* renomme les nœuds ayant un même nom et appartenant à un même arbre et ceci en préfixant le nom du nœud par le nom de son nœud père. Dans notre exemple courant, les nœuds des arbres possèdent des noms tous uniques, en conséquence cette étape ne modifie le nom d'aucun nœud. L'interface de la *Figure 57* montre le résultat du traitement des noms ambigus.

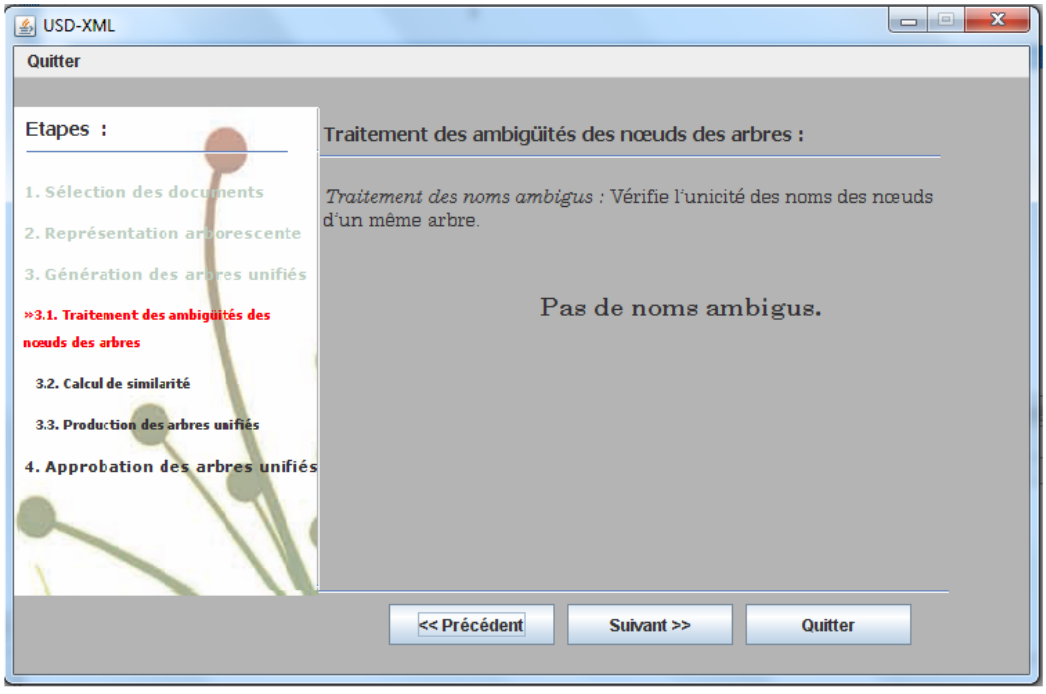


Figure 57 : *Résultat du traitement des noms ambigus.*

Ensuite, afin de déterminer les arbres qui peuvent être fusionnés, la matrice de similarité est calculée. Rappelons que la fusion de deux arbres n'est réalisable que lorsque le degré de similarité entre ces deux arbres est supérieur ou égal à un seuil. *USD* donne la main à l'utilisateur pour fixer son propre seuil.

Dans notre exemple, nous avons fixé arbitrairement le seuil à 0,3. La première itération produit la matrice montrée dans l'interface de la *Figure 58*.

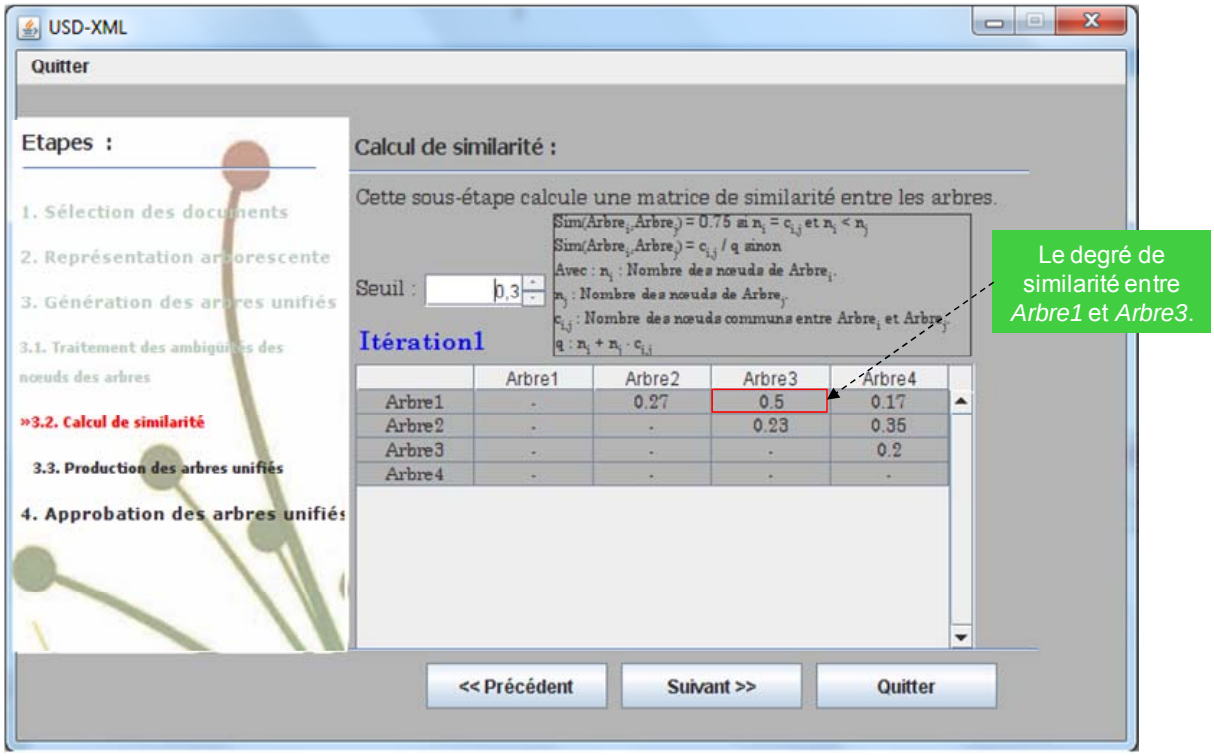


Figure 58 : *Matrice1 : Matrice de similarité résultat de la première itération.*

Nous constatons que le degré de similarité entre *Arbre1* et *Arbre3* dépasse le seuil 0,3 (c'est la valeur la plus élevée) en conséquence ces arbres seront fusionnés moyennant l'opérateur *Fusion par union des sous-arbres*. L'arbre résultat de la fusion (*Arbre5*) est visualisé dans la *Figure 59*.

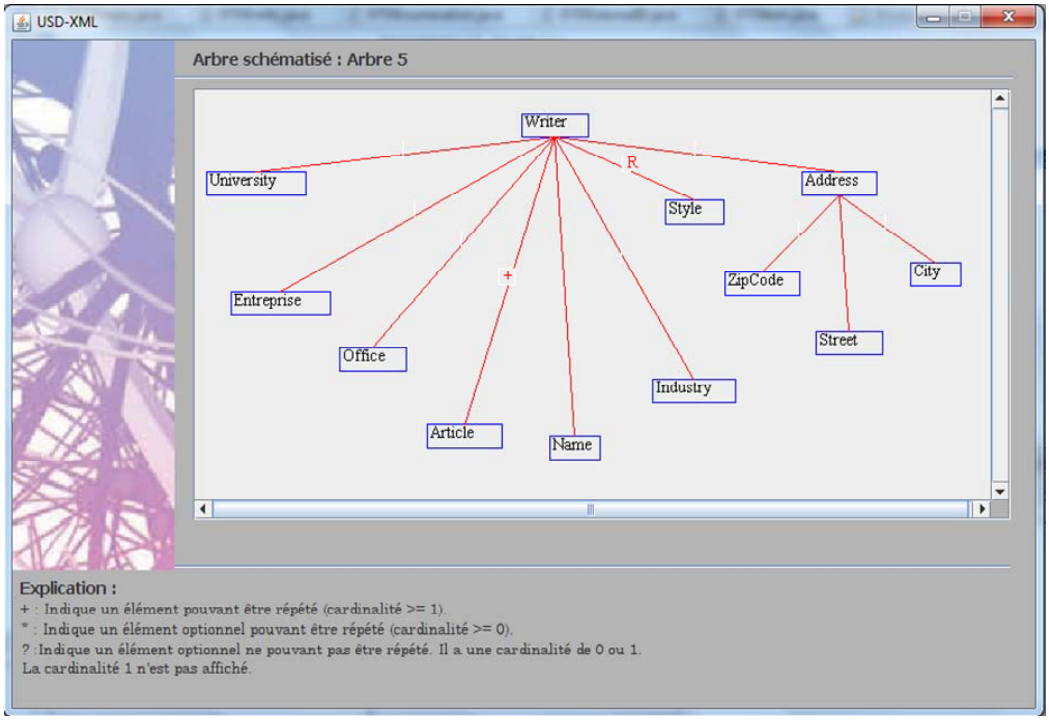


Figure 59 : *Arbre5 : Arbre résultat de la fusion des arbres 1 et 3.*

La *Figure 60* illustre la liste des arbres produits suite à la première itération.

Notons que l'utilisateur peut intervenir pour arrêter le processus d'unification. Dans le cas de notre exemple courant, les arbres *2*, *4* et *5* représenteront le résultat d'unification.

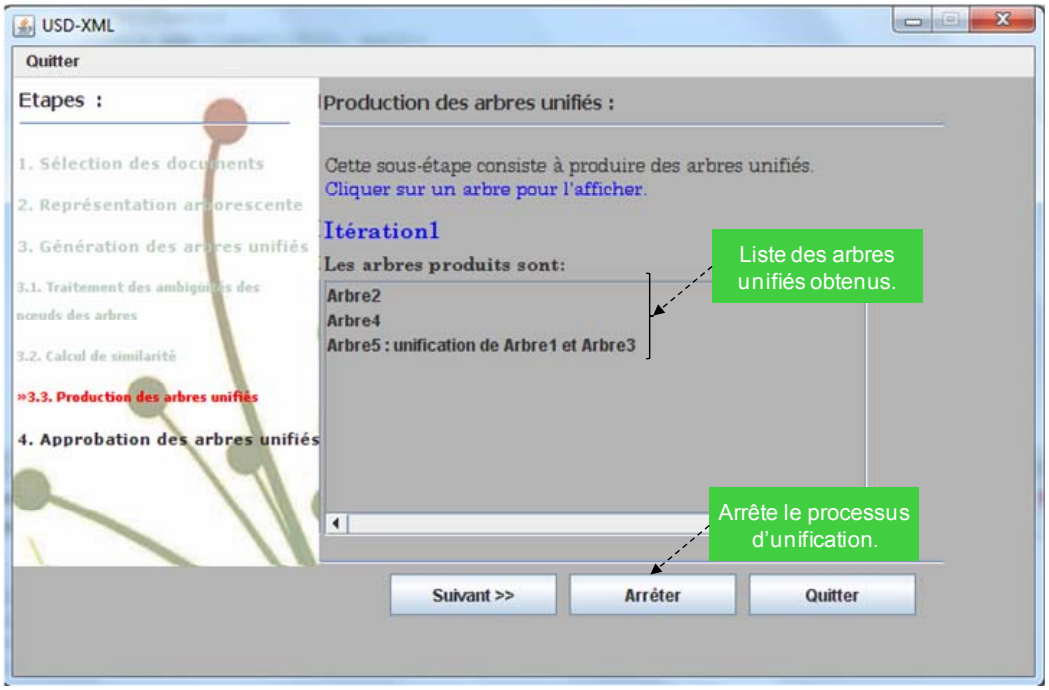


Figure 60 : *Liste les arbres unifiés résultats de la première itération.*

Nous remarquons que cette itération produit trois arbres ; une deuxième itération est alors déclenchée. Elle produit la matrice illustrée par la *Figure 61*.

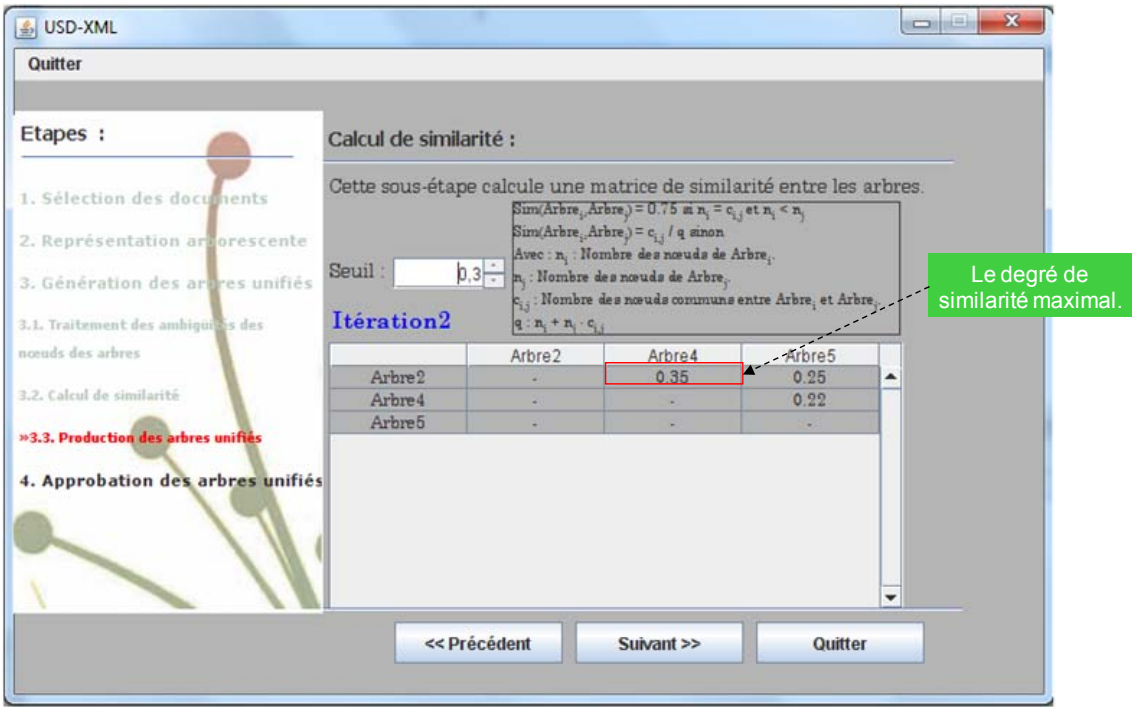


Figure 61 : *Matrice 2 : Matrice de similarité résultat de la deuxième itération.*

Le degré de similarité entre les arbres 2 et 4 dépasse le seuil 0,3. En conséquence, ces arbres seront fusionnés par l'opérateur *Fusion par union des sous arbres*. L'arbre résultat de la fusion (*Arbre6*) est montré dans la *Figure 62*.

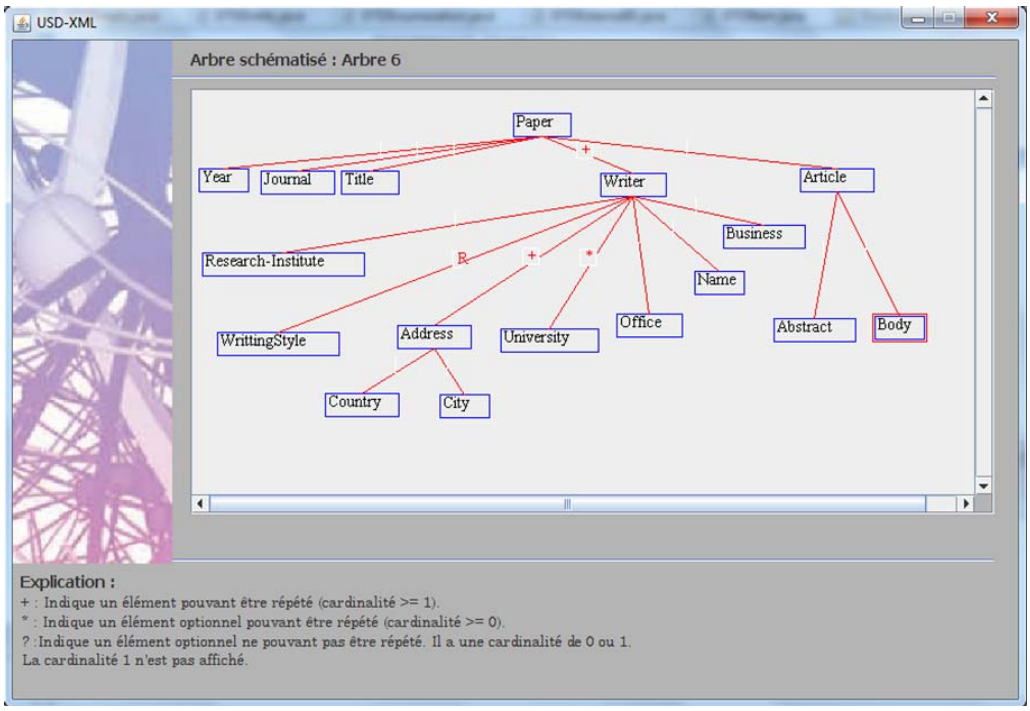


Figure 62 : *Arbre6 : Arbre résultat de la fusion des arbres 2 et 4.*

Cette deuxième itération produit deux arbres : *Arbre5* et *Arbre6*. Par conséquent, une troisième itération est déclenchée. La matrice résultat de la troisième itération est illustrée par la *Figure 63*.

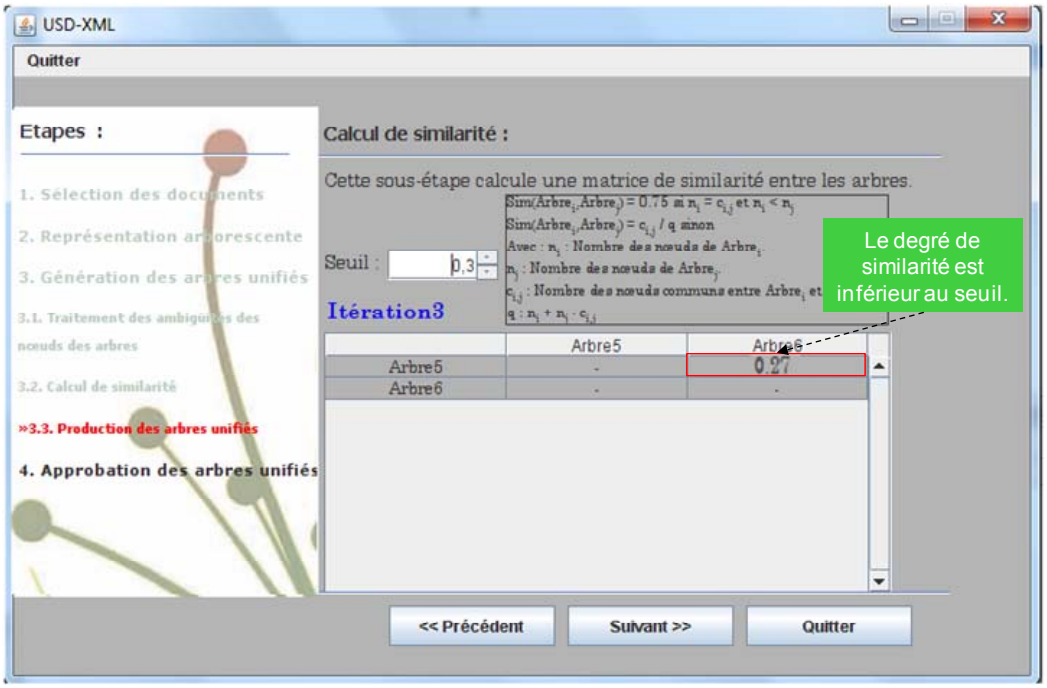


Figure 63 : *Matrice 3 : Matrice de similarité résultat de la troisième itération.*

Maintenant, du fait que le degré de similarité des arbres 5 et 6 est inférieur au seuil, le processus d'unification s'arrête ; ces deux arbres constituent le résultat final de l'unification.

Approbation des arbres unifiés :

Dans cette étape, le ou les arbres unifiés résultats de l'étape de génération sont fournis au décideur qui sera assisté par le concepteur pour les ajuster selon ses besoins d'analyse ; il peut supprimer et/ou renommer des nœuds. La *Figure 64* est l'*Arbre6* approuvé où l'utilisateur a renommé le nœud racine *Paper* par *Publication.*

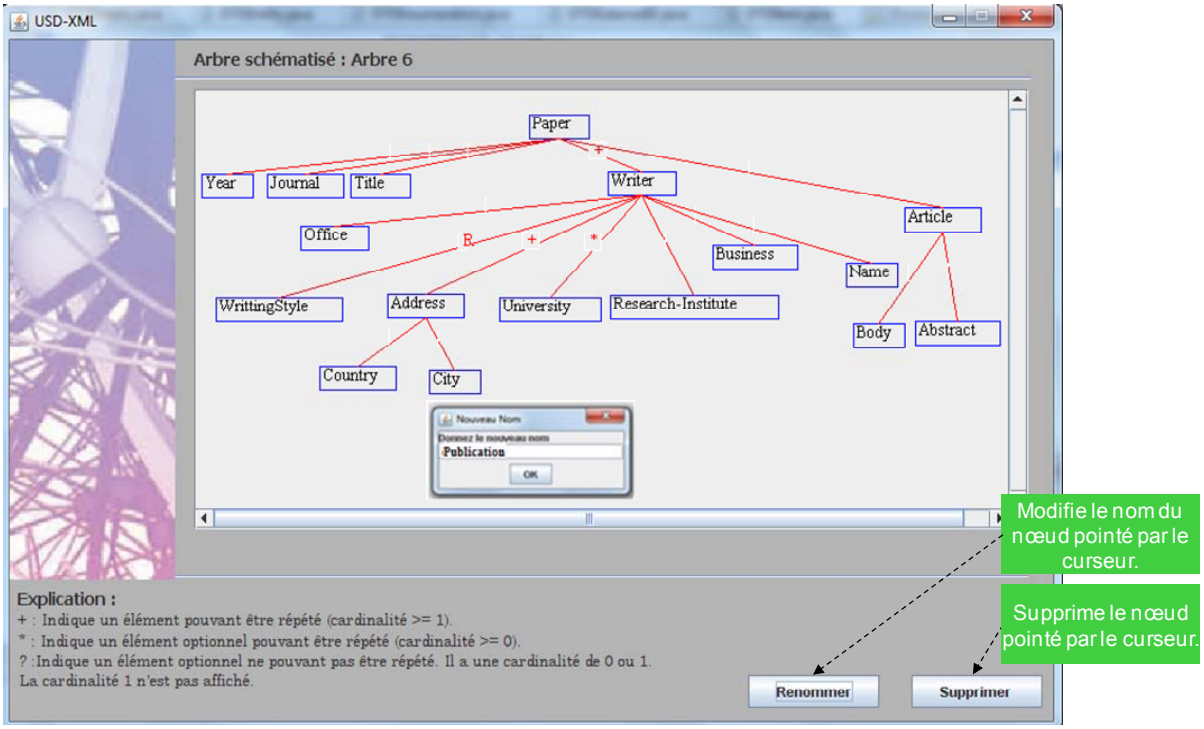


Figure 64 : *Exemple d'approbation de l'arbre6.*

Vérification des arbres unifiés :

L'étape de vérification des arbres a pour rôle de vérifier la validité syntaxique des arbres non unifiés et des arbres unifiés et ceci conformément aux quatre contraintes définies dans la section 3.6 du chapitre 3.

Dans notre exemple, les quatre arbres non unifiés (Arbre 1 à 4) sont valides ; l'interface de la *Figure 65* montre qu'elles vérifient les quatre contraintes.

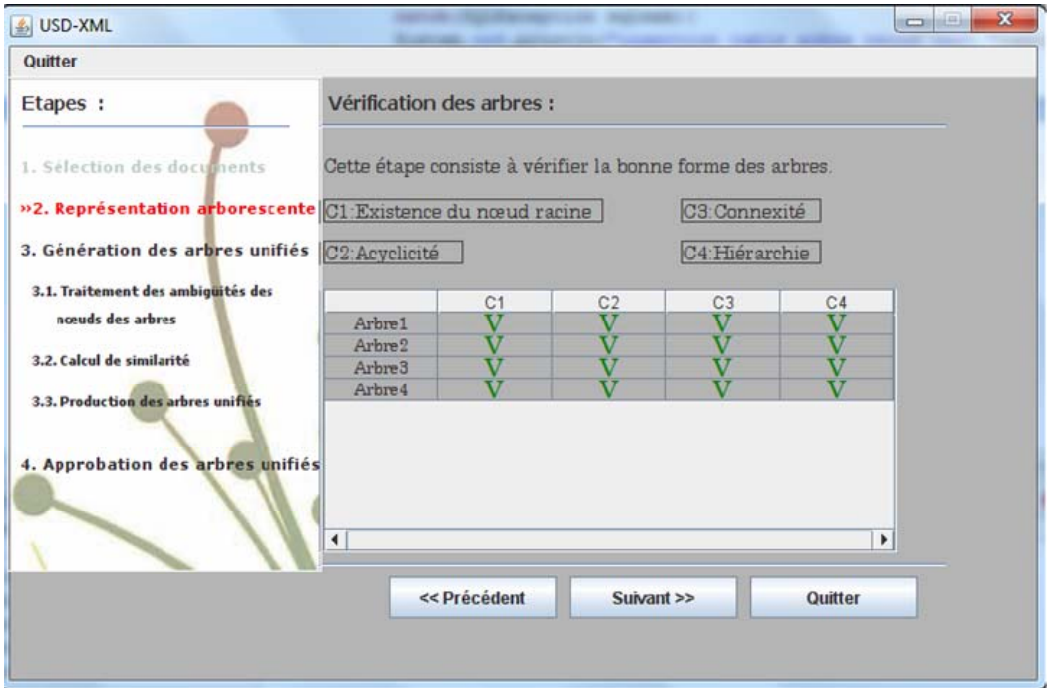


Figure 65 : *Résultat de la vérification syntaxique des arbres non unifiés.*

Les arbres résultats du processus d'unification sont aussi vérifiés syntaxiquement au cours de l'étape de l'approbation. Par exemple, nous avons programmé un contrôle interdisant la suppression du nœud racine.

En résumé, l'unification des deux DTDs 1 à 2 et des deux XSD 1 à 2 a produit les arbres 5 et 6. Ces arbres seront transformés en modèles multidimensionnels en galaxie et ceci en appliquant la méthode décrite dans le chapitre 4.

Dans la section suivante, nous présentons l'outil *Galaxy-Gen* automatisant la méthode proposée de modélisation multidimensionnelle.

5.4. *Galaxy-Gen* : Un outil de génération de modèles en galaxie

Galaxy-Gen implante les quatre étapes de la méthode de génération de galaxie (*cf.* Chapitre 4, Section 4.2) : a) *Prétraitement des arbres*, b) *Génération de modèles en galaxie*, c) *Approbation de modèles en galaxie*, et d) *Vérification de modèles en galaxie* (Ben Messaoud, et al., 2011) (Feki, et al., 2013).

5.4.1 ARCHITECTURE DE *GALAXY-GEN*

Galaxy-Gen comporte les cinq modules suivants :

- *Prétraitement d'arbre* : Ce module transforme un arbre en entrée en un arbre prétraité où chaque nœud père est enrichi par une cardinalité ajoutée en observant la collection de documents XML conformes aux structures initiales.
- *Génération de modèle en galaxie* : Ce module traduit l'arbre obtenu à partir du module précédent en un modèle multidimensionnel en galaxie. Ce traitement est réalisé en appliquant l'ensemble des dix règles que nous avons proposées.
- *Approbation de modèle en galaxie* : Il donne la main au concepteur/décideur pour approuver le modèle en galaxie obtenu par rapport à ses besoins d'analyse.
- *Vérification de modèle en galaxie* : Ce module vérifie la validité syntaxique du modèle en galaxie obtenu.
- *Affichage* : C'est le module de visualisation graphique des galaxies.

La *Figure 66* décrit l'architecture de l'outil *Galaxy-Gen* :

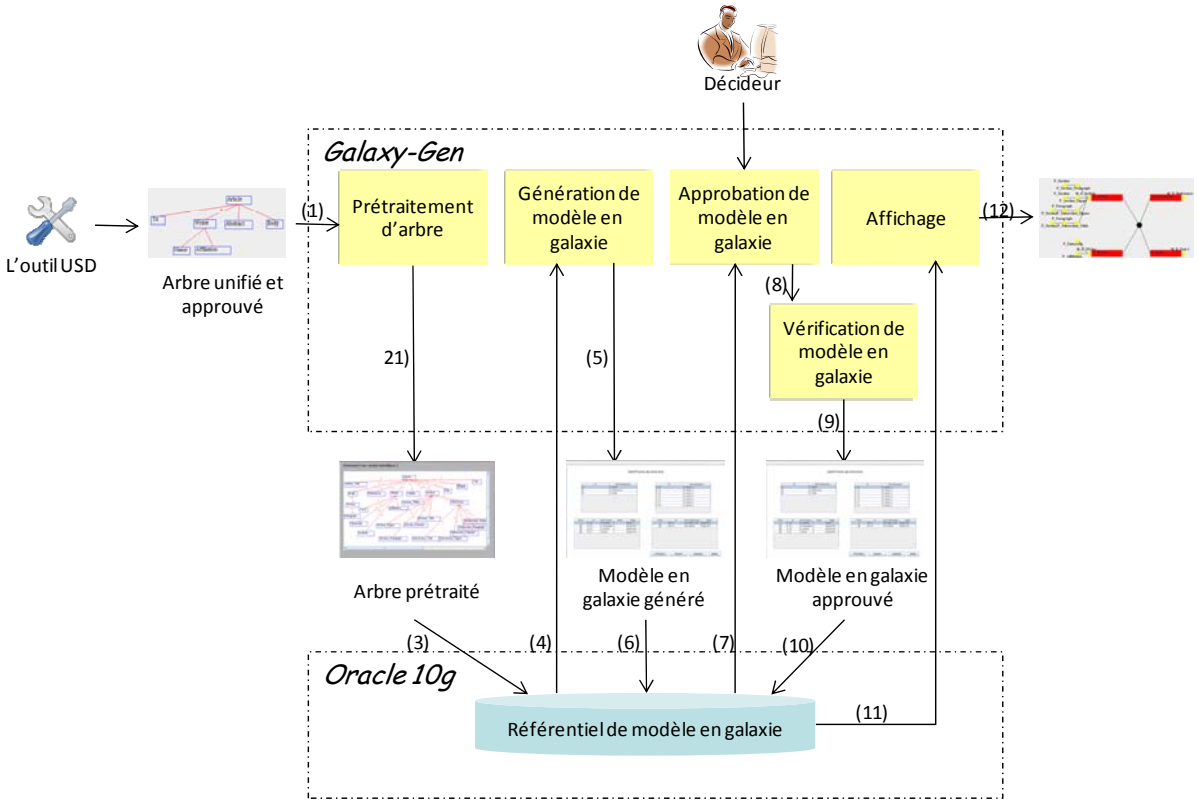


Figure 66 : *Architecture de l'outil Galaxy-Gen.*

5.4.2 INTERFACES DE *GALAXY-GEN*

Nous exposons, tout au long de cette section, les différentes interfaces de *Galaxy-Gen* via la génération d'un modèle en galaxie pour l'*Arbre6* (*cf. Figure 62*).

Prétraitement d'arbre :

Initialement, *l'Arbre6* subit un prétraitement qui consiste à enrichir chaque nœud père par des cardinalités ajoutées en examinant les seize documents XML conformes aux quatre structures initiales (deux DTDs et deux XSDs). L'interface de la *Figure 67* montre l'arbre résultat de l'étape de prétraitement de *l'Arbre6*.

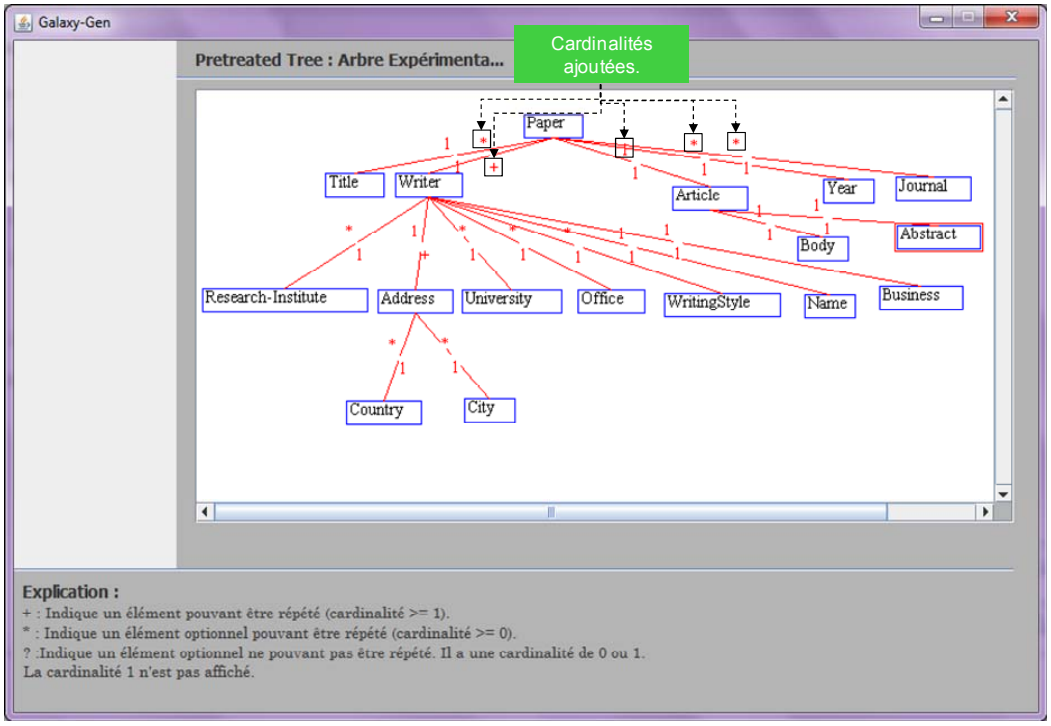


Figure 67 : *Interface de prétraitement de l'Arbre6.*

Génération de modèle en galaxie :

Après le prétraitement, la modélisation enchaîne sur l'extraction des axes d'analyses. Cette extraction est réalisée en appliquant les trois règles d'extraction des dimensions définies dans la section 4.5.1 du chapitre 4.

L'interface de la *Figure 68* affiche la liste des dimensions extraites. Pour chaque dimension, nous affichons la ou les règles utilisées pour son identification. Par exemple, la dimension *D-Paper* est extraite en appliquant les deux règles *Rd1* et *Rd2*.

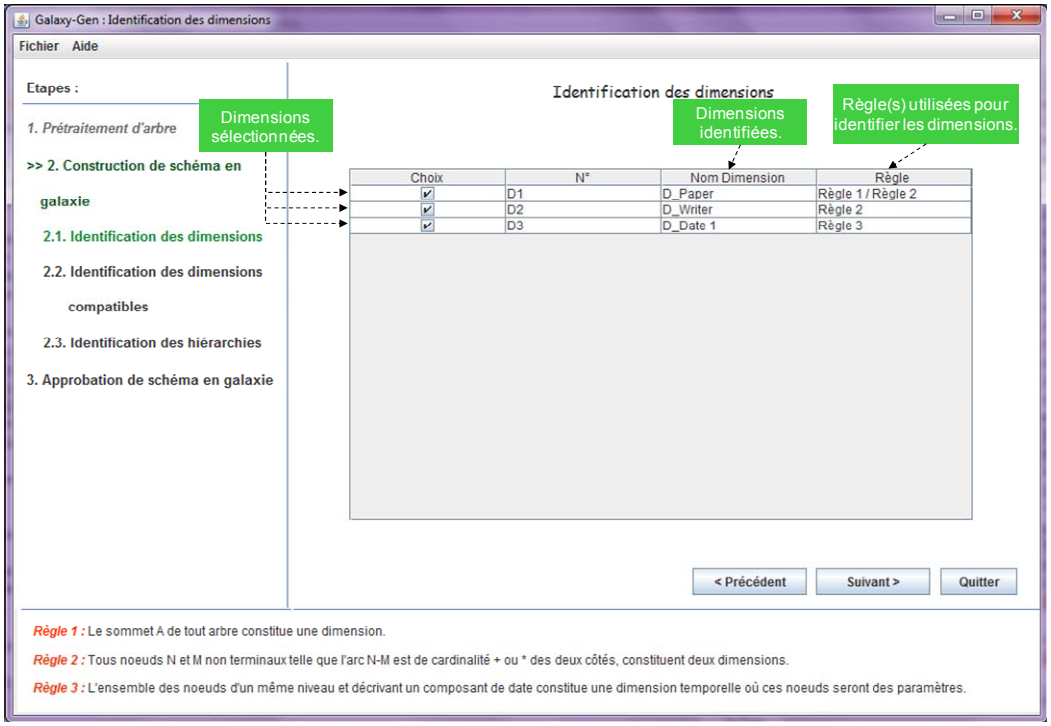


Figure 68 : *Interface d'extraction des dimensions.*

Le logiciel vérifie que le décideur a bien sélectionné des axes d'analyses. En cliquant sur le bouton *Suivant*, l'interface de la *Figure 69* est affichée pour visualiser les nœuds inter-dimensions identifiés.

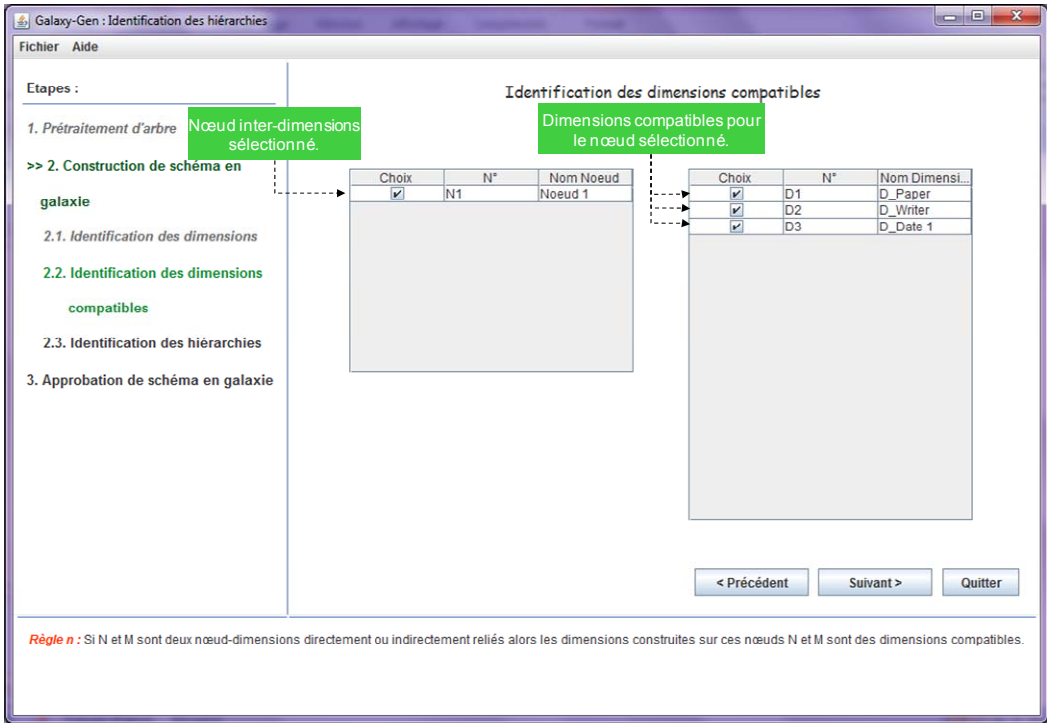


Figure 69 : *Interface d'identification des nœuds inter-dimensions.*

Puis les paramètres des hiérarchies seront extraits. La *Figure 70* indique les hiérarchies extraites pour la dimension *D-Paper*.

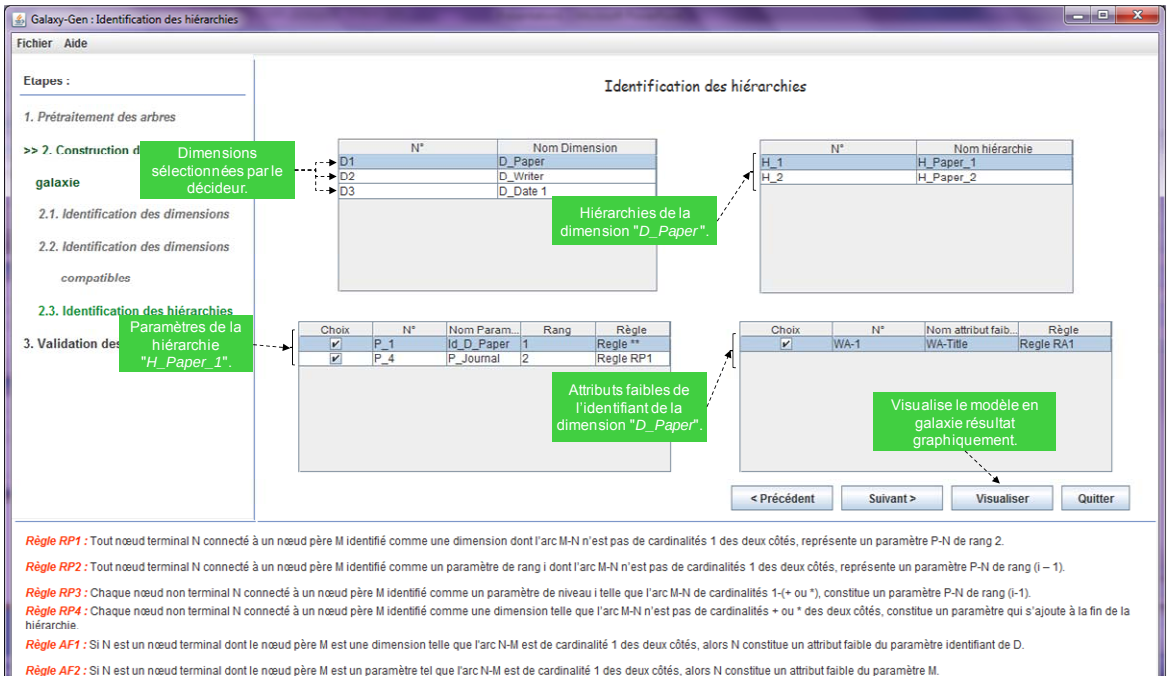


Figure 70 : *Interface d'identification des hiérarchies.*

Galaxy-Gen offre la visualisation graphique du modèle en galaxie. La *Figure 71* est le modèle en galaxie résultat.

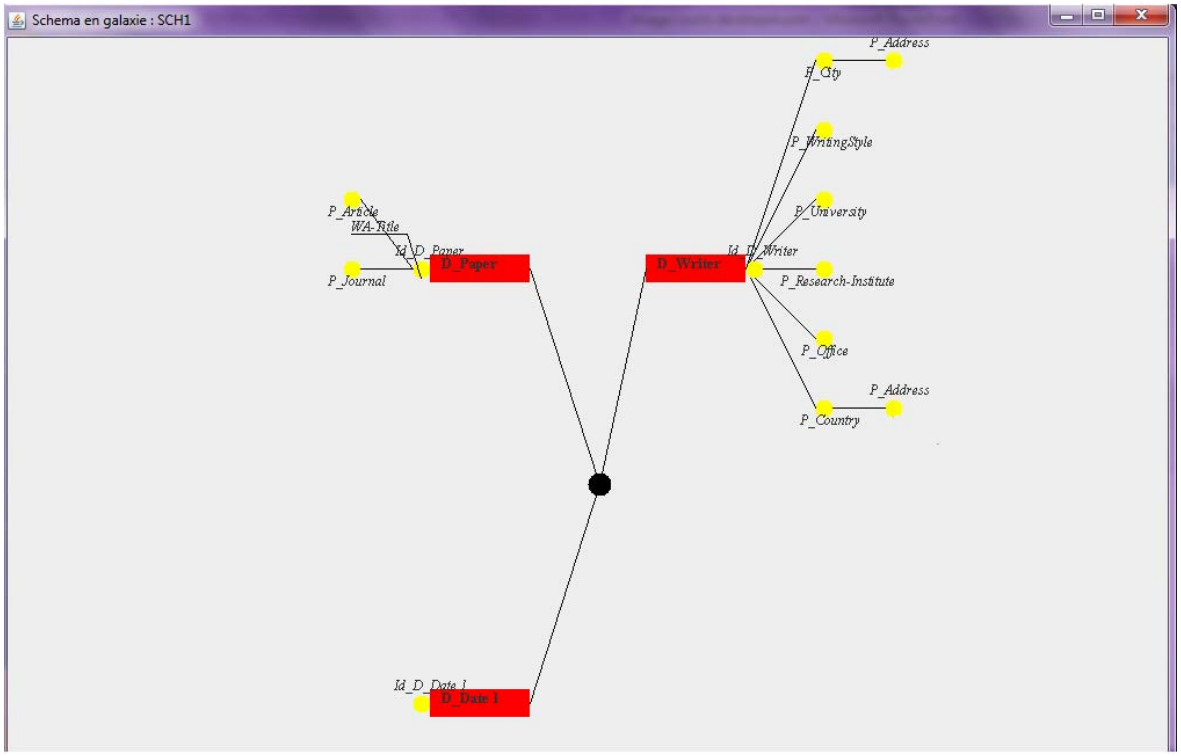


Figure 71 : *Modèle multidimensionnel en galaxie obtenu avec l'outil Galaxy-Gen pour l'Arbre6.*

A partir de ce modèle en galaxie, un décideur peut analyser, par exemple, le nombre d'articles par auteur. De même, il peut analyser le nombre d'article par université.

5.5. Conclusion

A travers ce chapitre, nous avons introduit les principales fonctionnalités des deux outils développés pour la construction du schéma de l'entrepôt de documents : ***USD*** « ***U****nification of* ***S****tructures of XML* ***D****ocuments* » et ***Galaxy-Gen*** « ***Galaxy Gen****eration* ». Les interfaces de ces deux outils sont illustrées à travers un exemple de quatre structures de documents XML issues des travaux de la littérature : deux DTDs et deux XSDs.

USD permet l'unification des structures des documents XML. Il transforme les structures des documents XML et les présente graphiquement sous forme d'arbre. De plus, il résout les ambiguïtés des noms des éléments : les synonymes et les acronymes. Il calcule également des matrices de similarité pour déterminer les structures qui méritent d'être unifiées. Finalement, il affiche l'arbre graphique de la structure unifiée.

Galaxy-Gen génère un modèle en galaxie à partir de l'arbre unifié généré par l'outil *USD*. Il détermine automatiquement les concepts multidimensionnels en appliquant un ensemble de dix règles. Aussi, il vérifie la validité des galaxies par rapport à un ensemble de onze contraintes. Cet outil permet de visualiser le modèle en galaxie résultat sous forme tabulaire et graphique.

CHAPITRE 6 : EXPERIMENTATION ET EVALUATION

Résumé du chapitre : *Ce chapitre présente un ensemble d'expérimentations des deux méthodes : (i) d'unification et (ii) de modélisation en galaxie. Ces expérimentations sont réalisées sur deux corpus ; le premier est composé de 20 documents XML du domaine académique définis manuellement, et le deuxième corpus est constitué de 1691 documents XML de la collection médicale Clef 2007. Ce chapitre discute également l'évaluation de ces expérimentations.*

Sommaire du chapitre 6

6.1. Introduction

Dans le but d'évaluer notre approche, nous avons réalisé un ensemble d'expérimentations sur deux corpus. Le premier corpus est défini manuellement. Il se compose de 20 documents XML du domaine académique et est décrit par quatre DTDs. Alors que le deuxième corpus se compose de 1691 documents XML de la collection médicale *Clef 2007*. Ce corpus est décrit par trois DTDs.

Dans ce chapitre, nous présentons l'expérimentation et l'évaluation des deux méthodes d'unification et de modélisation en galaxie sur ces deux corpus. La section 2 décrit ces deux corpus. La section 3 expose les expérimentations d'unification et de modélisation réalisées ainsi que les résultats obtenus. Finalement, la section 4 présente l'évaluation des expériences réalisées.

6.2. Description des corpus

Nous commençons par décrire les deux corpus que nous utilisons dans l'expérimentation ainsi que dans l'évaluation des deux méthodes de l'approche.

6.2.1 CORPUS DU DOMAINE ACADEMIQUE

Ce corpus est composé de 20 documents XML du domaine académique; il est décrit par les quatre DTDs présentées par les *Figure 72* à *Figure 75*. Nous avons construit ce corpus afin de tester les opérateurs et les règles que nous avons définis. Notons que chaque document appartenant à ce corpus comprend en moyenne 12 nœuds répartis sur un nombre maximum de 4 niveaux ; chaque nœud père possède au plus 7 nœuds fils.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Auth ((Name, Affiliation))>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Subsection ((Para))>
<!ELEMENT Section ((Title?, Subsection+))>
<!ELEMENT Para (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Month (#PCDATA)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Article ((Title, Auth+, Section+, Day, Month))>
<!ELEMENT Affiliation (#PCDATA)>
```

Figure 72 : *DTD1 du corpus académique.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT References (#PCDATA)>
<!ELEMENT Paragraph (#PCDATA)>
```

```
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Month (#PCDATA)>
<!ELEMENT Institute (#PCDATA)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Mots_Clefs (#PCDATA)>
<!ELEMENT Abstract (#PCDATA)>
<!ELEMENT Body ((Paragraph+))>
<!ELEMENT Writer ((Name, Institute))>
<!ELEMENT Article ((Title, Writer+, Abstract?, Mots_Clefs+, Body,
References+, Day, Month, Year))>
```

Figure 73 : *DTD2 du corpus académique.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT University (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Subsection_Number (#PCDATA)>
<!ELEMENT Subsection (Subsection_Number, Title, Paragraph+, Figure*,
Table*)>
<!ELEMENT Section_Number (#PCDATA)>
<!ELEMENT Section (Section_Number, Title, Paragraph+, Figure*, Table*,
Subsection*)>
<!ELEMENT Paragraph (#PCDATA)>
<!ELEMENT Outline (#PCDATA)>
<!ELEMENT Mots_Clefs (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Figure (#PCDATA)>
<!ELEMENT Table (#PCDATA)>
<!ELEMENT Writer ((Name, University))>
<!ELEMENT Article ((Title, Writer+, Outline, Mots_Clefs+, Section+,
Year))>
```

Figure 74 : *DTD3 du corpus académique.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Tit (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Body (#PCDATA)>
<!ELEMENT Mots_Clefs (#PCDATA)>
<!ELEMENT Writer ((Name, Affiliation))>
<!ELEMENT Article ((Tit, Writer+, Abstract, Mots_Clefs*, Body))>
<!ELEMENT Affiliation (#PCDATA)>
<!ELEMENT Abstract (#PCDATA)>
```

Figure 75 : *DTD4 du corpus académique.*

Le *Tableau 3* montre la description quantitative du corpus académique.

Nombre	DTD1	DTD2	DTD3	DTD4
- de documents XML par DTD	6	7	3	4
- total de documents	20			

Tableau 3 : *Caractéristiques du corpus académique.*

6.2.2 CORPUS MEDICAL

Nous avons veillé à mener une évaluation expérimentale plus significative de notre approche en utilisant un deuxième corpus réel composé de 1691 documents XML issus de la collection médicale *Clef 2007* qui est décrit par 3 DTDs (*cf. Figure 76* à *Figure 78*). La moyenne des nœuds par document est de 44, répartis sur 3 niveaux. Un nœud père peut avoir jusqu'à 36 nœuds fils.

Cependant, il y avait quelques lacunes dans ces documents : par exemple, les mots clés sont regroupés dans une même balise textuelle. Ceci empêche de les intégrer dans le modèle multidimensionnel en tant qu'axe d'analyse. Afin de remédier à cette insuffisance, nous avons amélioré les DTD initiales, en ajoutant la cardinalité + pour certains de leurs éléments, pour obtenir des documents XML plus précis.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WEBURL (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT State (#PCDATA)>
<!ELEMENT Sex (#PCDATA)>
<!ELEMENT Reviewer (#PCDATA)>
<!ELEMENT References (#PCDATA)>
<!ELEMENT Order (#PCDATA)>
<!ELEMENT OPolytrauma (#PCDATA)>
<!ELEMENT OPathologic (#PCDATA)>
<!ELEMENT OOperation (#PCDATA)>
<!ELEMENT OOpen (#PCDATA)>
<!ELEMENT OLocation (#PCDATA)>
<!ELEMENT OJoint (#PCDATA)>
<!ELEMENT OImplant (#PCDATA)>
<!ELEMENT OGraft (#PCDATA)>
<!ELEMENT ODislocation (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT KeyWords (#PCDATA)>
<!ELEMENT ImageThumbnaillID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Hospital (#PCDATA)>
<!ELEMENT Diagnosis (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT DateTime (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Creation (#PCDATA)>
<!ELEMENT Commentary (#PCDATA)>
<!ELEMENT ClinicalPresentation (#PCDATA)>
<!ELEMENT Chapter (#PCDATA)>
<!ELEMENT CaseID (#PCDATA)>
<!ELEMENT CASIMAGE_CASE ((ID, Description, Diagnosis, Sex, CaseID,
ClinicalPresentation+, Commentary, KeyWords+, Anatomy, Chapter, ACR,
References+, Author+, Reviewer+, Hospital, Department, State, Date,
Language, Title, Birthdate, Age, ImageThumbnaillID, Creation, DateTime,
Order, OJoint, OLocation, OImplant, ODislocation, OPolytrauma, OOpen,
OPathologic, OOperation, OGraft, WEBURL))>
<!ELEMENT Birthdate (#PCDATA)>
```

```
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Anatomy (#PCDATA)>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT ACR (#PCDATA)>
```

Figure 76 : *DTD1 de la collection médicale Clef 2007.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WEBURL (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT State (#PCDATA)>
<!ELEMENT Sex (#PCDATA)>
<!ELEMENT Reviewer (#PCDATA)>
<!ELEMENT References (#PCDATA)>
<!ELEMENT QUESTION (#PCDATA)>
<!ELEMENT QCM ((QUESTION, ANSWERA, ANSWERB, ANSWERC, ANSWERD,
COMMENTARY))>
<!ELEMENT Order (#PCDATA)>
<!ELEMENT OPolytrauma (#PCDATA)>
<!ELEMENT OPathologic (#PCDATA)>
<!ELEMENT OOperation (#PCDATA)>
<!ELEMENT OOpen (#PCDATA)>
<!ELEMENT OLocation (#PCDATA)>
<!ELEMENT OJoint (#PCDATA)>
<!ELEMENT OImplant (#PCDATA)>
<!ELEMENT OGraft (#PCDATA)>
<!ELEMENT ODislocation (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT KeyWords (#PCDATA)>
<!ELEMENT ImageThumbnaillID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Hospital (#PCDATA)>
<!ELEMENT Diagnosis (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT DateTime (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Creation (#PCDATA)>
<!ELEMENT Commentary (#PCDATA)>
<!ELEMENT ClinicalPresentation (#PCDATA)>
<!ELEMENT Chapter (#PCDATA)>
<!ELEMENT CaseID (#PCDATA)>
<!ELEMENT COMMENTARY (#PCDATA)>
<!ELEMENT CASIMAGE_CASE ((ID, Description, Diagnosis, Sex, CaseID,
ClinicalPresentation+, Commentary, KeyWords+, Anatomy, Chapter, ACR,
References+, Author+, Reviewer+, Hospital, Department, State, Date,
Language, Title, Birthdate, Age, ImageThumbnaillID, Creation, DateTime,
Order, OJoint, OLocation, OImplant, ODislocation, OPolytrauma, OOpen,
OPathologic, OOperation, OGraft, QCM+, WEBURL))>
<!ELEMENT Birthdate (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Anatomy (#PCDATA)>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT ANSWERD (#PCDATA)>
<!ELEMENT ANSWERC (#PCDATA)>
<!ELEMENT ANSWERB (#PCDATA)>
<!ELEMENT ANSWERA (#PCDATA)>
<!ELEMENT ACR (#PCDATA)>
```

Figure 77 : *DTD2 de la collection médicale Clef 2007.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WEBURL (#PCDATA)>
<!ELEMENT WEBLINK ((URL, DESCRIPTION))>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT State EMPTY>
<!ELEMENT Sex EMPTY>
<!ELEMENT Reviewer EMPTY>
<!ELEMENT References EMPTY>
<!ELEMENT QUESTION (#PCDATA)>
<!ELEMENT QCM ((QUESTION, ANSWERA, ANSWERB, ANSWERC, ANSWERD,
COMMENTARY))>
<!ELEMENT Order (#PCDATA)>
<!ELEMENT OPolytrauma (#PCDATA)>
<!ELEMENT OPathologic (#PCDATA)>
<!ELEMENT OOperation (#PCDATA)>
<!ELEMENT OOpen (#PCDATA)>
<!ELEMENT OLocation EMPTY>
<!ELEMENT OJoint EMPTY>
<!ELEMENT OImplant EMPTY>
<!ELEMENT OGraft (#PCDATA)>
<!ELEMENT ODislocation (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT LINK ((ID, COMMENTARY))>
<!ELEMENT KeyWords EMPTY>
<!ELEMENT ImageThumbnaillID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Hospital (#PCDATA)>
<!ELEMENT Diagnosis (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT DateTime (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT Creation (#PCDATA)>
<!ELEMENT Commentary EMPTY>
<!ELEMENT ClinicalPresentation (#PCDATA)>
<!ELEMENT Chapter (#PCDATA)>
<!ELEMENT CaseID EMPTY>
<!ELEMENT COMMENTARY (#PCDATA)>
<!ELEMENT CASIMAGE_CASE ((ID, Description, Diagnosis, Sex, CaseID,
ClinicalPresentation, Commentary, KeyWords, Anatomy, Chapter, ACR,
References, Author, Reviewer, Hospital, Department, State, Date,
Language, Title, Birthdate, Age, ImageThumbnaillID, Creation, DateTime,
Order, OJoint, OLocation, OImplant, ODislocation, OPolytrauma, OOpen,
OPathologic, OOperation, OGraft, QCM, LINK*, WEBLINK+, WEBURL))>
<!ELEMENT Birthdate (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Anatomy (#PCDATA)>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT ANSWERD (#PCDATA)>
<!ELEMENT ANSWERC (#PCDATA)>
<!ELEMENT ANSWERB (#PCDATA)>
<!ELEMENT ANSWERA (#PCDATA)>
<!ELEMENT ACR EMPTY>
```

Figure 78 : *DTD3 de la collection médicale Clef 2007.*

La description quantitative de ce corpus est illustrée par le *Tableau 4*.

Nombre	DTD1	DTD2	DTD3
- de documents XML par DTD	1623	66	2
- total de documents	1691		

Tableau 4 : *Caractéristiques du deuxième corpus.*

6.3. Expérimentation

Afin d'évaluer notre approche d'entreposage de documents XML, nous testons les deux méthodes proposées avec les deux corpus présentés dans la section précédente6.2.

6.3.1 APPLICATION DE L'UNIFICATION

Unification pour le corpus académique

L'application de la méthode d'unification (outil *USD*) sur les documents de ce corpus a généré, après trois itérations, un arbre unifié illustré par l'interface de la *Figure 79*.

Pour la génération de cet arbre, *USD* a détecté trois nœuds nommés par des acronymes (les nœuds nommés : *Auth*, *Para* et *Tit*). Il a remplacé ces acronymes par leurs formes complètes en se référant au dictionnaire des acronymes. Aussi, il a identifié des nœuds synonymes et ceci en accédant à la base lexicale *Wordnet* (les noms *Writer* et *Author* sont synonymes). D'autre part, *USD* a également résolu les ambiguïtés des noms (de onze nœuds : les nœuds nommés *Title*, *Paragraph*, *Figure*, *Table*, etc.) en préfixant le nom d'un nœud ambigu par celui de son nœud père, et ceci afin de garantir l'unicité des noms des nœuds.

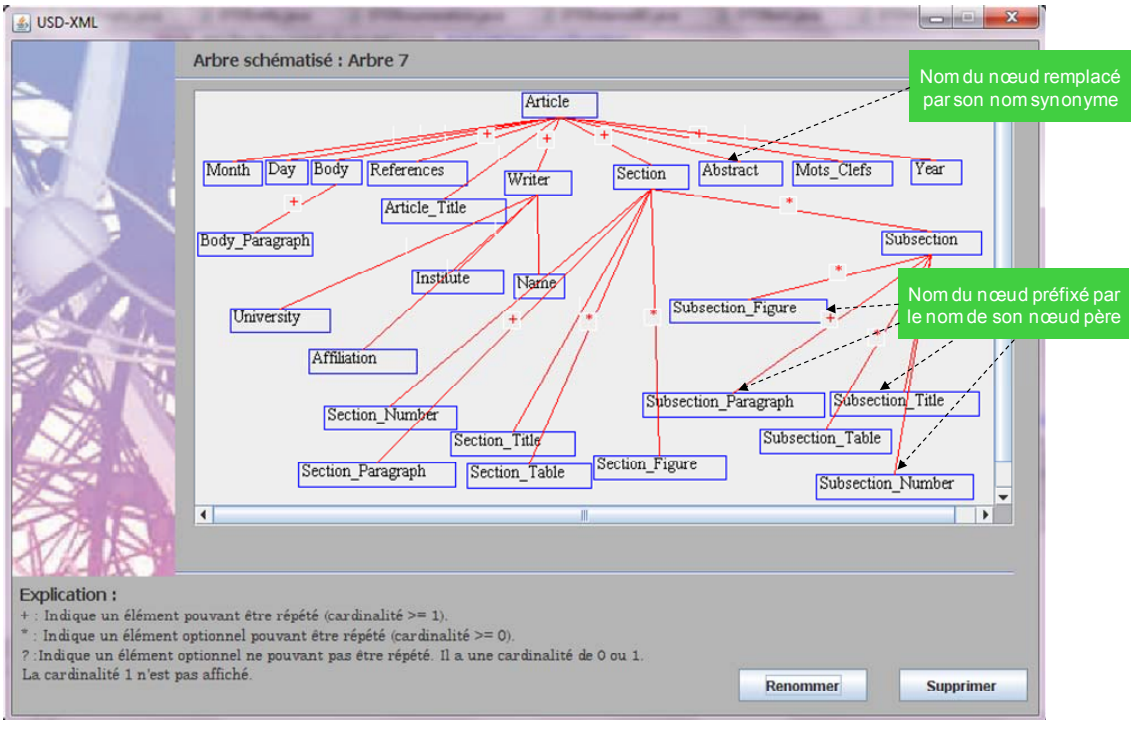


Figure 79 : *Arbre unifié résultat pour le corpus académique.*

Unification pour le corpus médical

Notre deuxième expérience d'unification est réalisée sur le corpus de la collection médicale *Clef 2007* (*cf.* section 6.2.2) ; elle a généré un arbre unifié après deux itérations. La *Figure 80* présente l'arbre unifié résultat. Notons que pour générer cet arbre, *USD* a résolu les ambiguïtés des noms des nœuds : 2 nœuds nommés *COMMENTARY* et 2 nœuds nommés *ID*.

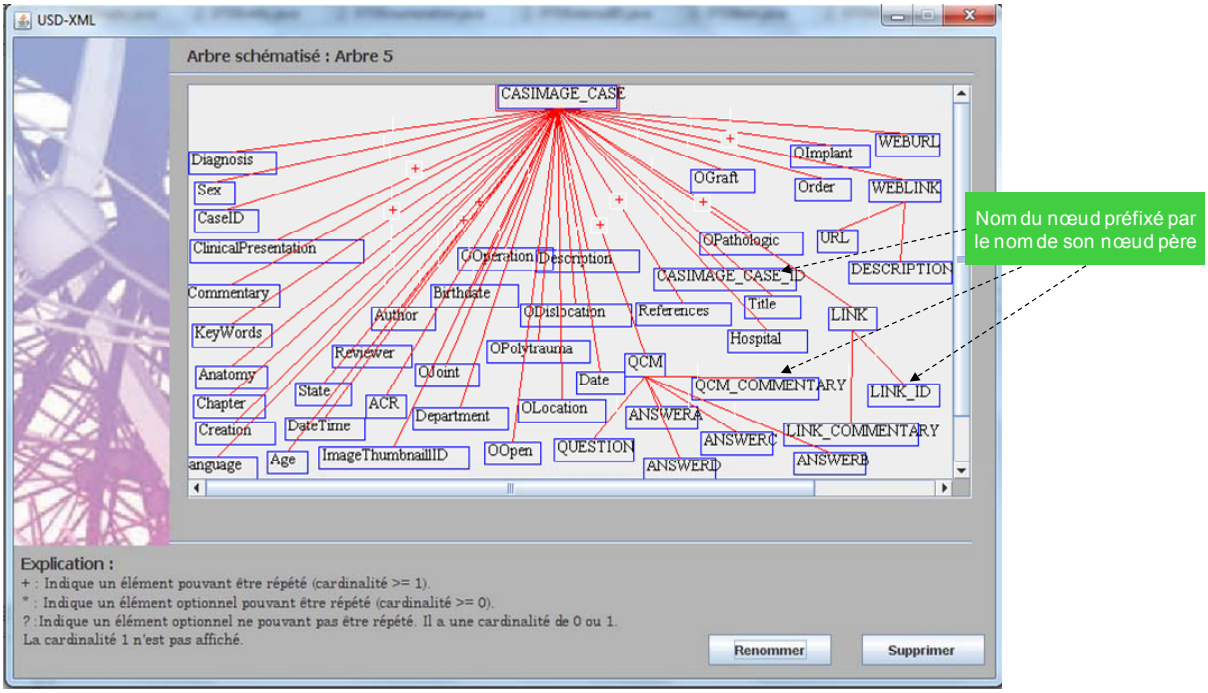


Figure 80 : *Arbre unifié résultat pour le corpus médical.*

6.3.2 APPLICATION DE LA MODELISATION EN GALAXIE

Génération de galaxie pour le corpus académique

Cette génération réalisée sur vingt documents XML du domaine académique a produit un modèle en galaxie comportant une dimension temporelle ainsi que quatre autres dimensions nommées *D-Article, D-References, D_Mots_Clefs* et *D-Writer* ; l'ensemble de ces cinq dimensions est connecté par un nœud.

La dimension *D-Article* décrit la structure d'un article de recherche. Elle est composée de six hiérarchies. La dimension *D-Writer* se rapporte aux auteurs des articles et compte trois hiérarchies. Finalement, les trois dimensions *D-References, D_Mots_Clefs* et *D-Date* décrivent respectivement les références, les mots clefs et la date de publication d'un article de recherche, chacune possédant une seule hiérarchie (Ben Messaoud, et al., 2014).

La *Figure 81* montre le modèle en galaxie généré par l'outil *Galaxy-Gen* pour le corpus académique.

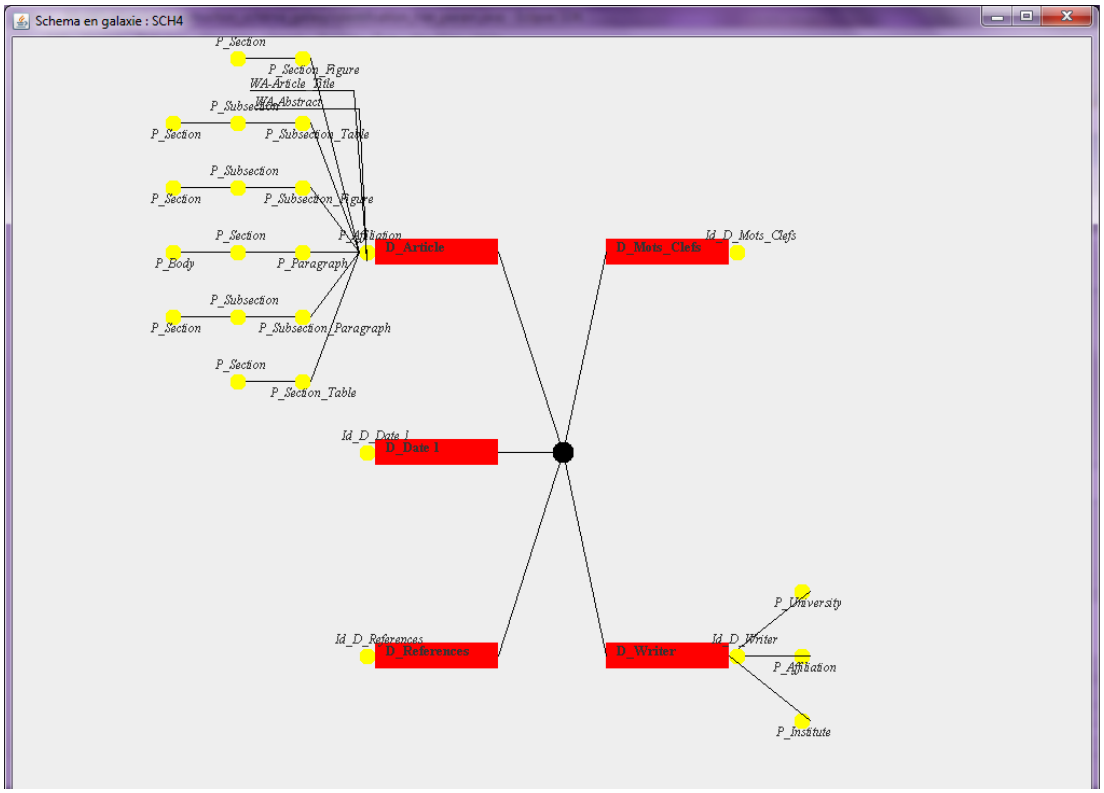


Figure 81 : *Modèle en galaxie résultat pour le corpus académique.*

Génération de galaxie pour le corpus médical

L'application des étapes de la méthode de génération de modèle en galaxie pour ce corpus nous a permis de produire un modèle en galaxie constitué de cinq dimensions : *D-CASIMAGE-Case, D-Author, D-References, D-KeyWords* et *D-Reviewer* toutes connectées par un seul nœud. La dimension *D-CASIMAGE-Case* est composée de quatre hiérarchies, tandis que chacune des quatre autres dimensions en possède une seule (Ben Messaoud, et al., 2014). La *Figure 82* illustre le modèle en galaxie ainsi généré par *Galaxy-Gen* (seuls les paramètres sont présentés dans le modèle résultat).

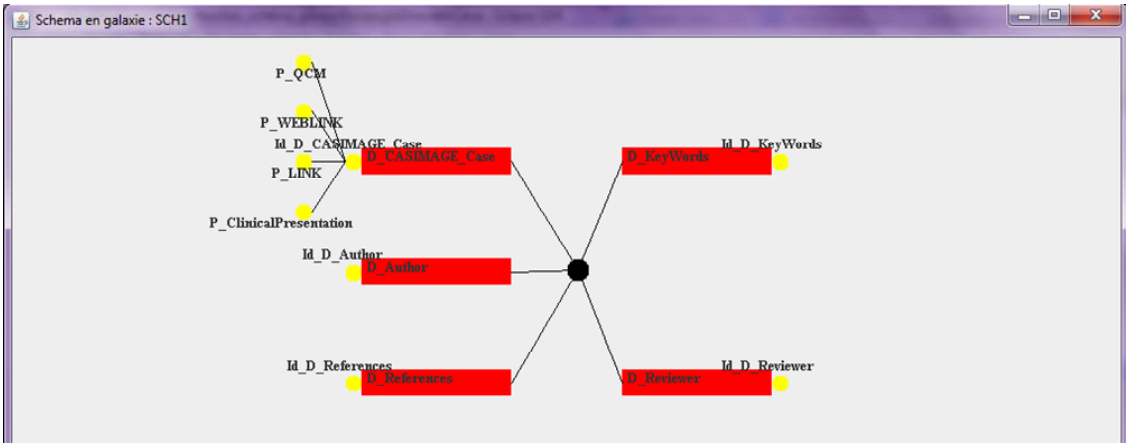


Figure 82 : *Modèle en galaxie résultat pour le corpus médical.*

6.4. Evaluation

Dans cette section, nous évaluons en premier lieu la méthode d'unification et nous exposons en deuxième lieu l'évaluation de la méthode de modélisation en galaxie.

6.4.1 EVALUATION DE LA METHODE D'UNIFICATION

Pour évaluer le résultat de la méthode d'unification, nous vérifions que les documents XML de chaque corpus restent encore valides par rapport à leur DTD unifiée. Pour cette vérification syntaxique, nous avons utilisé *XMLSpy* et nous avons suivi les étapes suivantes :

- Traduire chaque arbre résultat de l'unification en une DTD unifiée,
- Remplacer la cardinalité de chaque élément par une cardinalité plus générale conformément aux quatre règles suivantes :

+ → *
* → *
1 → ?
? → ?

- Remplacer les noms des éléments et des attributs de la DTD unifiée présentant des noms désambiguïsés, des noms synonymes ou des noms acronymes par leurs correspondants et ceci en consultant le référentiel des arbres.

Unification pour le corpus académique

La traduction de l'arbre unifié résultat de la première expérience (*cf. Figure 79*) et l'application des traitements nous ont permis de déterminer la DTD de la *Figure 83*.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Article ( (Title?|Tit?), (Author*|Auth*|Writer*),
(Abstract?|Outline?), Mots_Clefs*,Section*, Body?, References*, Day?,
Month?, Year?)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Tit (#PCDATA)>
<!ELEMENT Author (Name?, Affiliation?, Institute?, University?)>
<!ELEMENT Auth (Name?, Affiliation?, Institute?, University?)>
<!ELEMENT Writer (Name?, Affiliation?, Institute?, University?)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Affiliation (#PCDATA)>
<!ELEMENT Institute (#PCDATA)>
<!ELEMENT University (#PCDATA)>
<!ELEMENT Mots_Clefs (#PCDATA)>
<!ELEMENT Abstract (#PCDATA)>
<!ELEMENT Outline (#PCDATA)>
<!ELEMENT Section (Section_Number?, Title?, Paragraph*, Figure*, Table*,
Subsection*)>
<!ELEMENT Section_Number (#PCDATA)>
<!ELEMENT Paragraph (#PCDATA)>
<!ELEMENT Figure (#PCDATA)>
```

```
<!ELEMENT Table (#PCDATA)>
<!ELEMENT Subsection (Subsection_Number?, Title?, (Para?|Paragraph*),
Figure*, Table*)>
<!ELEMENT Subsection_Number (#PCDATA)>
<!ELEMENT Para (#PCDATA)>
<!ELEMENT Body (#PCDATA|Paragraph)*>
<!ELEMENT References (#PCDATA)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Month (#PCDATA)>
<!ELEMENT Year (#PCDATA)>
```

Figure 83 : *DTD unifiée résultat pour le corpus académique.*

Nous avons vérifié la validité des vingt documents XML du corpus académique par rapport à leur DTD unifiée en utilisant l'outil *XMLSpy*. Ainsi, la *Figure 84* montre la validité de quatre documents du corpus académique.

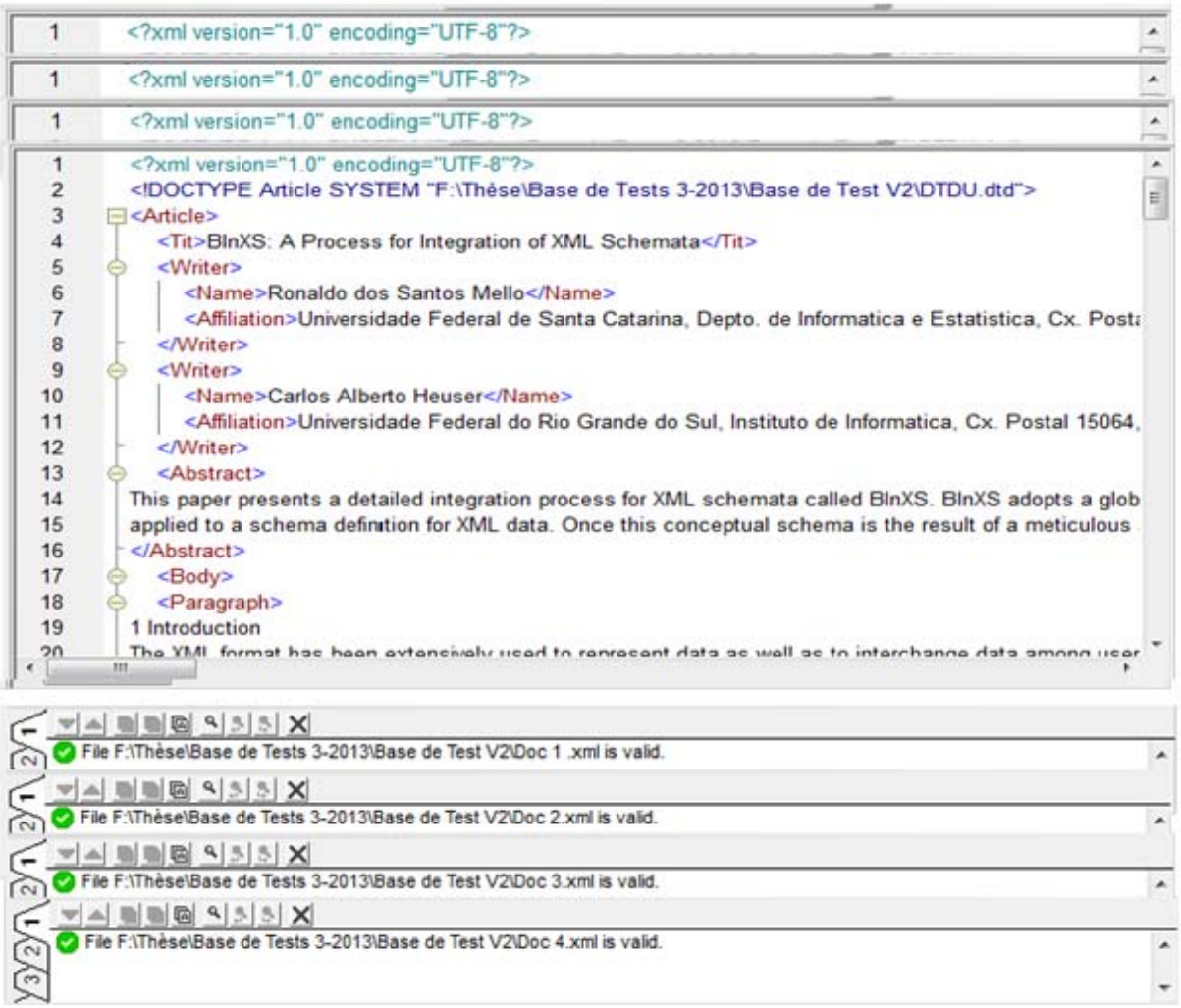


Figure 84 : *Validité de quatre documents XML du corpus académique par rapport à leur DTD unifiée.*

Unification pour le corpus médical

La traduction de l'arbre résultat d'unification du corpus médical (*cf. Figure 80*) en une DTD et la réalisation des traitements déjà cités a produit la DTD unifiée de la *Figure 85*.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT WEBURL (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT State (#PCDATA)>
<!ELEMENT Sex (#PCDATA)>
<!ELEMENT Reviewer (#PCDATA)>
<!ELEMENT References (#PCDATA)>
<!ELEMENT Order (#PCDATA)>
<!ELEMENT OPolytrauma (#PCDATA)>
<!ELEMENT OPathologic (#PCDATA)>
<!ELEMENT OOperation (#PCDATA)>
<!ELEMENT OOpen (#PCDATA)>
<!ELEMENT OLocation (#PCDATA)>
<!ELEMENT OJoint (#PCDATA)>
<!ELEMENT OImplant (#PCDATA)>
<!ELEMENT OGraft (#PCDATA)>
<!ELEMENT ODislocation (#PCDATA)>
<!ELEMENT Language (#PCDATA)>
<!ELEMENT KeyWords (#PCDATA)>
<!ELEMENT ImageThumbnaillID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Hospital (#PCDATA)>
<!ELEMENT Diagnosis (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Department (#PCDATA)>
<!ELEMENT DateTime (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT Creation (#PCDATA)>
<!ELEMENT Commentary (#PCDATA)>
<!ELEMENT ClinicalPresentation (#PCDATA)>
<!ELEMENT Chapter (#PCDATA)>
<!ELEMENT CaseID (#PCDATA)>
<!ELEMENT CASIMAGE_CASE ((ID, Description, Diagnosis, Sex, CaseID,
ClinicalPresentation+, Commentary, KeyWords+, Anatomy, Chapter, ACR,
References+, Author+, Reviewer+, Hospital, Department, State, Date,
Language, Title, Birthdate, Age, ImageThumbnaillID, Creation, DateTime,
Order, OJoint, OLocation, OImplant, ODislocation, OPolytrauma, OOpen,
OPathologic, OOperation, OGraft, QCM*, LINK*, WEBLINK*, WEBURL))>
<!ELEMENT QCM ((QUESTION, ANSWERA, ANSWERB, ANSWERC, ANSWERD,
COMMENTARY))>
<!ELEMENT Birthdate (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Anatomy (#PCDATA)>
<!ELEMENT Age (#PCDATA)>
<!ELEMENT ACR (#PCDATA)>
<!ELEMENT ANSWERD (#PCDATA)>
<!ELEMENT ANSWERC (#PCDATA)>
<!ELEMENT ANSWERB (#PCDATA)>
<!ELEMENT ANSWERA (#PCDATA)>
<!ELEMENT COMMENTARY (#PCDATA)>
<!ELEMENT QUESTION (#PCDATA)>
<!ELEMENT LINK ((ID?, COMMENTARY?))>
<!ELEMENT WEBLINK ((URL?, DESCRIPTION?))>
<!ELEMENT URL (#PCDATA)>
```

```
<!ELEMENT DESCRIPTION (#PCDATA)>
```

Figure 85 : *DTD unifiée résultat pour le corpus médical.*

Par ailleurs, nous avons également utilisé l'outil *XMLSpy* pour vérifier la validité des documents XML par rapport à cette DTD unifiée et nous avons constaté que les documents de la collection médicale *Clef 2007* sont valides. Ainsi, la *Figure 86* montre la validité d'un échantillon de quatre documents du corpus médical par rapport à leur DTD unifiée.

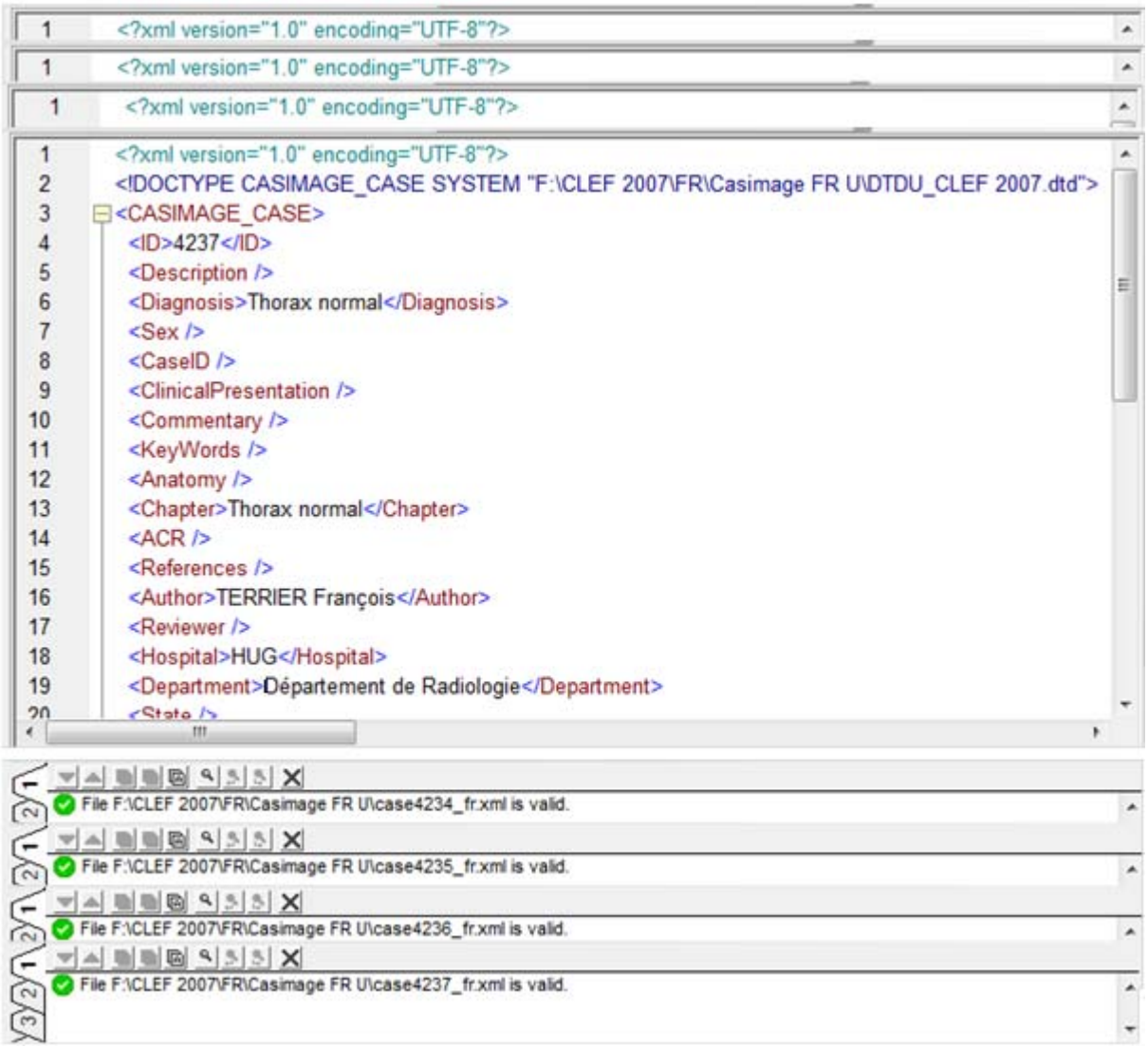


Figure 86 : *Validité de quatre documents XML du corpus médical par rapport à leur DTD unifiée.*

En conclusion, la méthode proposée pour l'unification des structures des documents XML a permis de déterminer la structure commune des documents XML en entrée. Cette structure décrit le contenu d'un ensemble de documents appartenant à un même domaine. Plus particulièrement, elle définit la grammaire permettant de vérifier la conformité des documents

XML. Notons que cette structure unifiée est obtenue après la résolution des noms acronymes, synonymes et ambigus, et un calcul de similarité.

6.4.2 EVALUATION DE LA METHODE DE MODELISATION EN GALAXIE

Pour évaluer la méthode de modélisation en galaxie, nous avons utilisé les opérateurs de manipulation multidimensionnelle des galaxies définis dans (Tournier, 2007) et ceci afin de justifier que le modèle en galaxie résultat permet au décideur d'exprimer des requêtes décisionnelles significatives qui l'aident dans le processus décisionnel.

Dans ce qui suit, et pour la clarté de cette section, nous présentons tout d'abord le langage de manipulation multidimensionnelle des galaxies. Ensuite, nous suggérons un ensemble de requêtes décisionnelles et nous appliquons les opérateurs de ce langage pour exprimer ces requêtes.

6.4.2.1. Langage de manipulation multidimensionnelle

Dans (Tournier, 2007), l'auteur a proposé un langage de manipulation multidimensionnelle pour le modèle en galaxie. Il a présenté une fonction nommée *AVG_KW* et un ensemble d'opérateurs permettant la manipulation des concepts multidimensionnels de la galaxie.

- Fonction d'agrégation *AVG_KW* :

L'agrégation permet de restreindre le volume des données à visualiser par un utilisateur lors d'une même analyse. Dans ce cadre, la fonction *AVG_KW* permet de résumer un ensemble de mots-clefs en un ensemble restreint de mots-clefs généraux et ceci en utilisant une ontologie de domaine. Notons que les mots-clefs à synthétiser sont issus d'une ontologie dont le domaine est voisin de celui des documents à analyser.

- Opérateur de focalisation *FOCUS* :

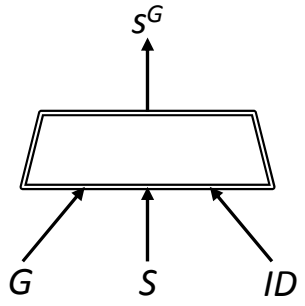
Cet opérateur permet de spécifier une analyse multidimensionnelle et ceci en choisissant un sujet d'analyse et de projeter les données de ce sujet sur des axes d'analyse.

L'opérateur FOCUS :

FOCUS (G, S, ID)=s^G

où :

- *G* : un modèle en galaxie en entrée.
- *S = (D_s, H_s, $Param_s$)* : le sujet d'analyse focalisé avec :
  - D_s : la dimension qui joue le rôle d'un sujet d'analyse,
  - H_s : une hiérarchie de D_s, et
  - $Param_s$= (*fagg*($Param_{s\_i}$[, $Param_{s\_inf}$, …, $Param_{s\_sup}$]). Les données de la dimension sont agrégées par la fonction *fagg* en fonction d'une liste d'attributs facultatifs ($Param_{s\_inf}$, …, $Param_{s\_sup}$).
- *ID = ((D_x, H_x, $Param_x$), (D_y, H_y, $Param_y$), …)* : un ensemble ordonné des axes de projection, telle que
  - D_x est la première dimension sélectionnée,
  - H_x est la hiérarchie courante de D_x, et
  - $Param_x$ = ($Param_{x\_inf}$, …, $Param_{x\_sup}$) est un ensemble ordonné de paramètres de H_x.
  - D_y est la deuxième dimension sélectionnée,
  - H_y est la hiérarchie courante de D_y, et
  - $Param_y$ = ($Param_{y\_inf}$, …, $Param_{y\_sup}$) est un ensemble ordonné de paramètres de H_y.



Représentation graphique

- Opérateur de sélection *SELECT* :

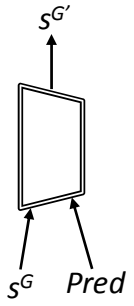
L'opérateur *SELECT* a pour rôle de réduire le volume de données à analyser d'un modèle en galaxie s^G et ceci en spécifiant un prédicat de restriction *Pred* exprimé sur un axe ou un sujet d'analyse de s^G.

L'opérateur SELECT :

SELECT (s^G, pred)= $s^{G'}$

où :

- s^G : un modèle en galaxie en entrée.
- *pred* : un prédicat de restriction sur un attribut d'une dimension.



Représentation graphique

Afin de changer le niveau de détail utilisé pour examiner les données, deux opérateurs de forage peuvent être employés.

- Opérateur de forage vers le bas *DRILL-DOWN* :

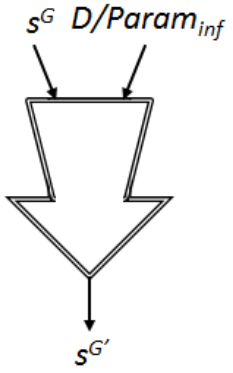
Il permet d'accéder à des données plus détaillées.

L'opérateur DRILL-DOWN :

DRILL-DOWN (s^G, *D*, $Param_{inf}$) = $s^{G'}$

où :

- s^G : un modèle en galaxie en entrée.
- *D* : une dimension de l'ensemble des dimensions de projection de s^G.
- $Param_{inf}$: un paramètre de granularité plus fine que le paramètre déjà sélectionné.



Représentation graphique

- Opérateur de forage vers le haut *ROLL-UP* :

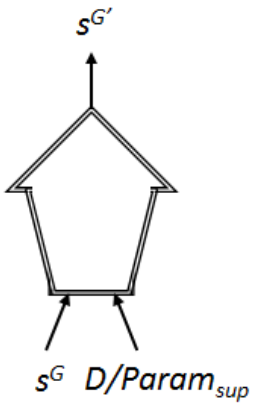
Contrairement à l'opérateur *DRILL-DOWN,* cet opérateur permet d'obtenir une vision plus agrégée des données analysées.

L'opérateur ROLL-UP :

ROLL-UP (s^G, *D*, $Param_{sup}$) = $s^{G'}$

où :

- s^G : un modèle en galaxie en entrée.
- *D* : une dimension de l'ensemble des dimensions de projection de s^G.
- $Param_{sup}$: un paramètre de niveau supérieur que le paramètre déjà sélectionné.



Représentation graphique

Afin de réorienter une analyse c'est-à-dire changer le sujet ou l'axe d'analyse, l'opérateur *ROTATE* est utilisé.

- Opérateur de réorganisation d'analyse *ROTATE* :

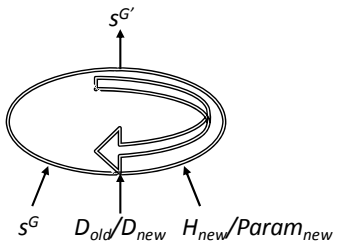
Il modifie l'un des axes d'analyse du modèle en galaxie par un nouvel axe. Notons que la dimension modifiée peut être le sujet d'analyse ou l'une des dimensions de l'ensemble des dimensions de projection.

L'opérateur ROTATE :

ROTATE (s^G, D_{old}, $D_{new}.H_{new}$, $Param_{new}$)=$s^{G'}$

où :

- s^G : un modèle en galaxie en entrée.
- D_{old} : l'axe d'analyse à modifier.
- D_{new} : le nouvel axe d'analyse.
- H_{new} : une hiérarchie de l'axe d'analyse D_{new}.
- $Param_{new}$: un paramètre appartenant à la hiérarchie H_{new}.



Représentation graphique

Dans les sections suivantes, nous utilisons ces opérateurs pour illustrer l'interrogation des modèles en galaxie issus de notre méthode de modélisation.

6.4.2.2. Expression de requêtes sur la galaxie du corpus académique

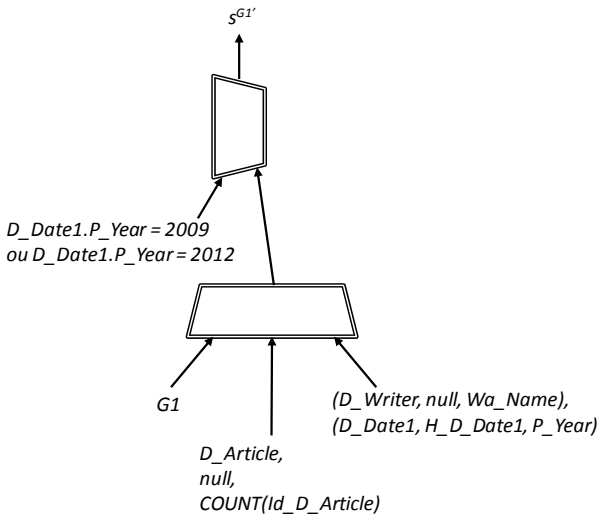
Pour montrer la faisabilité du modèle en galaxie généré pour le corpus académique, nous définissons deux requêtes décisionnelles et nous appliquons les opérateurs du langage proposé pour la galaxie (*cf. Figure 81*) afin de résoudre ces requêtes.

Requête 1 : Nombre d'articles par nom d'auteur pour les années 2009 et 2012.

Pour exprimer cette requête, nous utilisons tout d'abord l'opérateur *FOCUS* pour mettre en avant le sujet d'analyse *D_Article*. Ensuite, nous employons l'opérateur *SELECT* pour restreindre la portée de l'analyse en cours aux années 2009 et 2012. Cette analyse est représentée par le *Tableau 5*.

```
SELECT
(
  FOCUS
  (
  G1,
  (D_Article, null, COUNT(Id_D_Article),
  (D_Writer, null, Wa_Name),
  (D_Date1, H_D_Date1, P_Year)
  ),
 (D_Date1.P_Year = 2009 ou
 D_Date1.P_Year = 2012)
)
```

Notation textuelle



Notation graphique

COUNT (Id_D_Article)		D_Date 1		
		P_Year	2009	2012
D_Writer	Wa_Name			
	Jamel Feki		2	1
	Fahmi Bargui		2	0
	Hanene Ben-Abdallah		2	0
	Gilles Zurfluh		0	1
	Ines Ben Messaoud		0	1

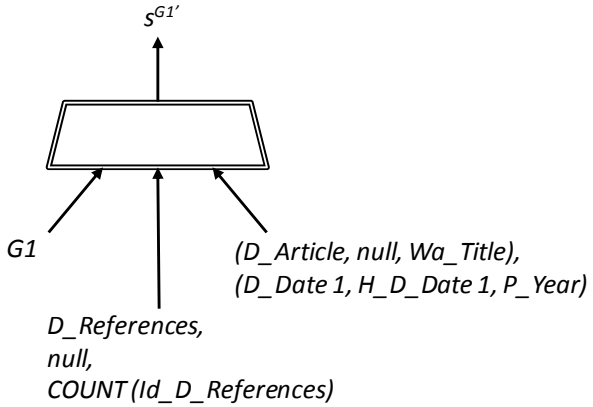
Tableau 5 : *Nombre d'articles par nom d'auteur pour les années 2009 et 2012.*

Requête 2 : Nombre de références bibliographiques par titre d'article et par année de publication.

Pour répondre à ce type d'analyse, il faut sélectionner le sujet d'analyse *D_References* et projeter les données de ce sujet sur les deux axes d'analyse *D-Article* et *D-Date 1*. Pour ce faire, nous utilisons l'opérateur *FOCUS*. Cette analyse est représentée par le *Tableau 6*.

FOCUS
(
G1,
(D_References, null, COUNT
(Id_D_References)),
(D_Article, null, Wa_Title),
(D_Date 1, H_D_Date 1, P_Year)
)

Notation textuelle



Notation graphique

COUNT (Id_D_Refences)		D_Date 1					
		P_Year	2002	2005	2008	2009	2012
D_Article	Wa_Title						
	Semantic integration of XML schema		11				
	Arbres de décisions			64			
	Patrons multidimensionnels constraints				27		
	Un modèle distribué d'entrepôt pédagogique Utilisation de métadonnées LOM et d'annotations sémantiques				9		
	A hybrid approach for data mart schema design from NL-OLAP requirements					6	
	Multidimensional Concept Extraction and Validation from OLAP Requirements in NL					18	
	A First Step for Building a Document Warehouse						21

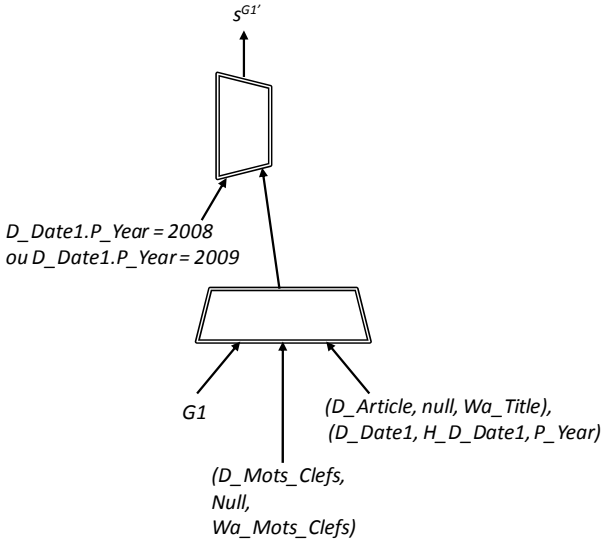
Tableau 6 : *Nombre de références bibliographiques par titre d'article et par année de publication.*

Requête 3 : Analyse des mots clefs par titre d'article pour les années 2008 et 2009.

Pour exprimer cette requête, nous employons l'opérateur *FOCUS* pour mettre en avant le sujet d'analyse *D_Mots_Clefs*. Finalement, nous utilisons l'opérateur *SELECT* pour limiter l'analyse en cours aux années 2008 et 2009. Cette analyse est représentée par le *Tableau 7*.

```
SELECT
(
  FOCUS
  (
  G1,
  (D_Mots_Clefs, H_D_Mots_Clefs,
   P_Mot_Clef),
  (D_Article, null, Wa_Title),
  (D_Date1, H_D_Date1, P_Year)
  ),
 (D_Date1.P_Year = 2008 ou
 D_Date1.P_Year = 2009)
)
```

Notation textuelle



Notation graphique

P_Mots_Clefs		D_Date 1		
		P_Year	2008	2009
D_Article	Wa_Title			
	A hybrid approach for Data Mart schema design for NL_OLAP requirements			OLAP
	Multidimensional concept extraction and validation from OLAP requirements in NL			OLAP, Design, Processing, Database, Logical
	Un modèle distribué d'entrepôt pédagogiques utilisation de métadonnées LOM et d'annotations sémantiques		Conception, Etoile, OLTP, OLAP, Processing, Logique	
	Patrons multidimensionnels contraints		Architecture, Traitement	

Tableau 7 : *Analyse des mots clefs par titre d'article pour les années 2008 et 2009.*

Afin d'apporter une vision plus globale du résultat de la requête précédente, nous utilisons la fonction d'agrégation *AVG_KW* pour résumer et synthétiser les mots clefs.

Requête 4 : Analyse des mots clefs synthétisés par titre d'article pour les années 2008 et 2009.

La réponse à cette requête nécessite l'utilisation de la fonction *AVG_KW*. Egalement, nous avons utilisé l'ontologie de domaine sur les systèmes d'information définie dans (Tournier, 2007). Notons que nous avons apporté quelques modifications à cette ontologie (nous avons défini une version française pour cette ontologie). La *Figure 87* illustre l'ontologie utilisée.

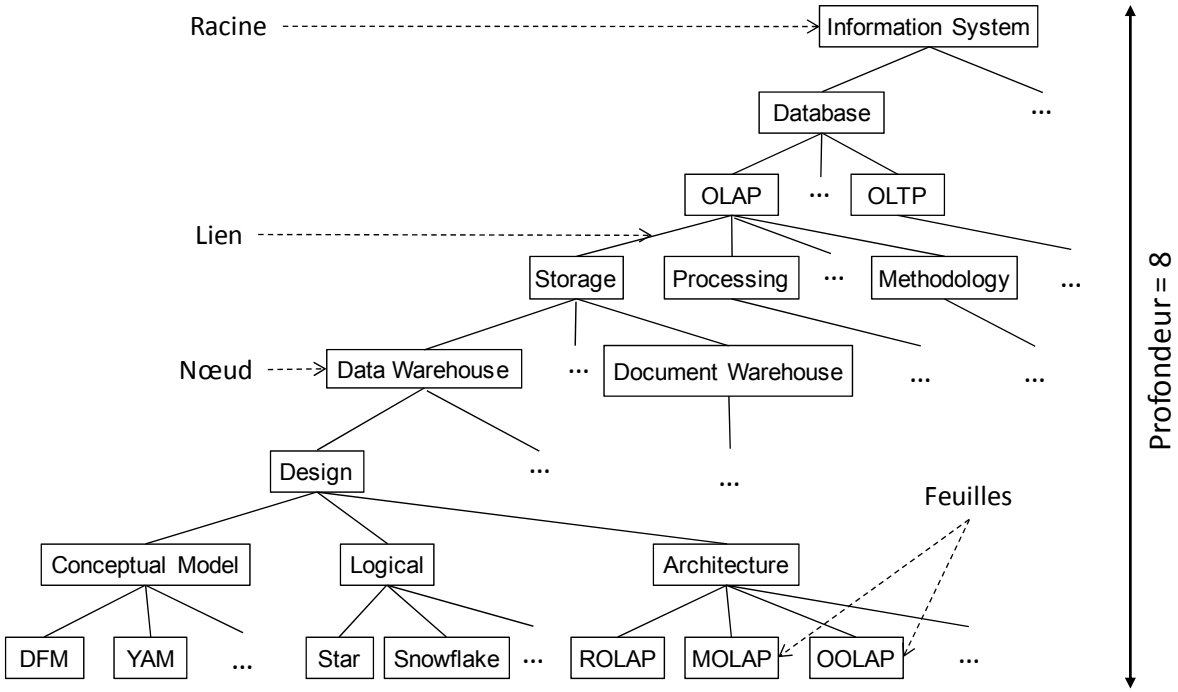


Figure 87 : *Extrait de l'ontologie de domaine sur les systèmes d'information* **(Tournier, 2007)**.

Le *Tableau 8* représente l'analyse de la requête 4.

AVG_KW (P_Mots_Clefs)		D_Date 1		
		P_Year	2008	2009
D_Article	Wa_Title			
	A hybrid approach for Data Mart schema design for NL_OLAP requirements			OLAP
	Multidimensional concept extraction and validation from OLAP requirements in NL			Design, OLAP, Database
	Un modèle distribué d'entrepôt pédagogiques utilisation de métadonnées LOM et d'annotations sémantiques		Logique, OLAP, Database	
	Patrons multidimensionnels contraints		Architecture, Traitement	

Tableau 8 : *Analyse des mots clefs synthétisés par titre d'article pour les années 2008 et 2009.*

6.4.2.3. Evaluation de la méthode de modélisation pour le corpus médical

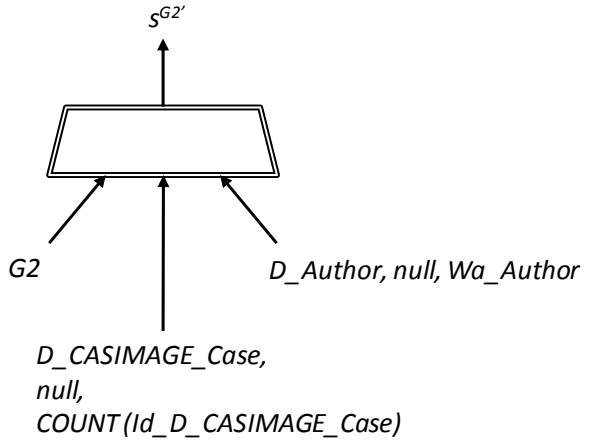
Afin d'évaluer le modèle en galaxie généré pour le corpus médical (*cf. Figure 82*), nous choisissons deux requêtes décisionnelles et nous utilisons les opérateurs du langage proposé pour la galaxie afin de répondre à ces requêtes.

Requête 1 : Nombre de cas cliniques (casimage_case) par auteur.

La réponse à cette requête nécessite l'utilisation de l'opérateur *FOCUS.* En effet, il permet de sélectionner le sujet d'analyse *D_CASIMAGE_Case* et de projeter les données de ce sujet sur l'axe : *D_Author.* Le *Tableau 9* présente le résultat de cette analyse.

```
FOCUS
 (
  G2,
  (D_CASIMAGE_Case, null, COUNT
  (Id_D_CASIMAGE_Case)),
  (D_Author, null, Wa_Author),
 )
```

Notation textuelle



Notation graphique

		COUNT (Id_D_Casimage_Case)
D_Author	Wa_Author	
	Frank Kolo	10
	BERIS Photis	6
	TERRIER François	37
	ROSSET Antoine	51
	Natalia Dfouni	50
	Jean Garcia	36
	Martins Martina	95
	BERCKER Minerva	1
	MP Bianchi	102
	J Garcia	56
	PETITPIERRE Nicolas	1
	Pkindynis	1
	DIDIER Dominique	26
	⋮	⋮

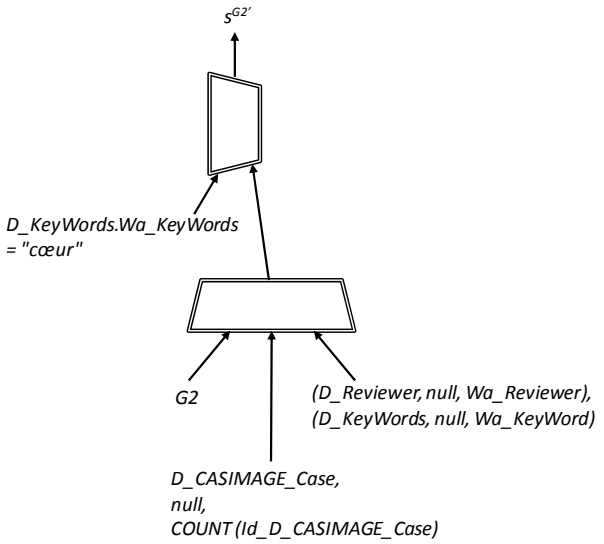
Tableau 9 : *Analyse du nombre de cas cliniques par auteur.*

Requête 2 : Nombre de cas cliniques par nom de reviewer et pour le mot clé "cœur".

La réponse à cette requête nécessite l'utilisation des opérateurs *FOCUS* et *SELECT*. En effet, *FOCUS* précise le sujet d'analyse *D_CASIMAGE_Case* et projette les données de ce sujet sur les axes : *D_Author* et *D_KeyWords ; a*lors que l'opérateur *SELECT* permet de spécifier le prédicat de restriction (mot clé = "cœur"). Le *Tableau 10* présente cette analyse.

```
SELECT
 (
  FOCUS
   (
    G2,
    (D_CASIMAGE_Case, null,
    COUNT (Id_D_CASIMAGE_Case)),
    (D_Reviewer, null, Wa_Reviewer),
    (D_KeyWords, null, Wa_KeyWord)
   ),
   (D_KeyWords.Wa_KeyWords =
   "cœur")
 )
```

Notation textuelle



Notation graphique

COUNT (Id_D_Casimage_Case)		D_KeyWords	
		Wa_KeyWords	cœur
D_Reviewer	Wa_Reviewer		
	F Terrier		0
	N Dfouni		0
	Howarth Nigel		2
	Jean Garcia		0
	⋮		⋮

Tableau 10 : *Nombre de cas cliniques par nom de reviewer et pour le mot clé "cœur".*

A partir de ces requêtes, nous constatons que notre méthode proposée pour la modélisation en galaxie nous a permis de définir des modèles en galaxies qui aident les preneurs de décisions dans le processus décisionnel. Ces galaxies permettent d'exprimer des requêtes décisionnelles significatives. La résolution de ces requêtes a été réalisée en utilisant les opérateurs spécifiques à ces modèles.

6.5. Conclusion

Les expérimentations, exposées dans ce chapitre, sont réalisées sur deux corpus : *un académique* et un autre *médical*. Le premier est défini manuellement. Il est composé de 20 documents XML du domaine académique, et est décrit par 4 DTDs. Tandis que le deuxième corpus est constitué de 1691 documents XML de la collection médicale *Clef 2007*. Ces documents sont décrits par 3 DTDs.

Les expériences de la méthode d'unification sur ces deux corpus nous a permis de définir une DTD unifiée pour chaque collection de documents. Cette DTD présente la structure commune des documents XML en entrée. Nous avons vérifié que les documents XML de chaque corpus sont valides par rapport à leur DTD unifiée.

Les expériences de la méthode de modélisation en galaxie ont généré *(i)* un modèle en galaxie composé de cinq dimensions et un nœud inter-dimension pour les documents XML du corpus académique et *(ii)* un modèle en galaxie formé de cinq dimensions liées par un nœud pour les documents de la collection médicale *Clef 2007*. Pour montrer l'utilité de ces modèles quant à la prise de décisions, nous avons défini des requêtes et nous avons utilisé les opérateurs du langage de manipulation multidimensionnelle des galaxies pour les exprimer.

Conclusion générale

Bilan

Les travaux de cette thèse se situent dans le contexte de l'entreposage de documents XML. Plus précisément, nous avons élaboré et décrit une approche de construction d'un schéma d'entrepôt de documents XML, comportant deux méthodes : *Unification des structures des documents XML* et *Modélisation multidimensionnelle* (Ben Messaoud, et al., 2010).

La *méthode d'unification* proposée permet de définir une structure commune pour décrire les documents XML hétérogènes appartenant au même domaine. Elle s'articule autour de quatre étapes que nous rappelons : *(i) la représentation arborescente* traduit les structures des documents XML (DTDs et XSDs) en arbre ; *(ii) la génération des arbres unifiés* traite les synonymes et les acronymes, et génère des arbres à partir des arbres issus des différentes structures. Pour effectuer cette génération, nous avons défini une métrique de similarité qui évalue la ressemblance entre arbres ; elle utilise une matrice de similarité qui facilite l'identification des arbres les plus prioritaires à fusionner moyennant un ensemble d'opérateurs ; *(iii) l'approbation* permet au décideur d'approuver les arbres unifiés afin de garantir la bonne forme des arbres générés ; *(iv) la vérification des arbres résultats* conformément à un ensemble de contraintes (Ben Messaoud, et al., 2011a) (Ben Messaoud, et al., 2012). Pour valider cette méthode, nous avons développé un outil logiciel baptisé **USD** (« **U**nification of **S**tructures of XML **D**ocuments »).

La *méthode de modélisation multidimensionnelle* permet de concevoir semi-automatiquement un modèle *en galaxie* à partir des arbres unifiés. Ce modèle est plus approprié aux entrepôts de documents. Il a l'avantage de la simplicité puisqu'il repose sur l'unique concept de *Dimension* et offre ainsi une bonne flexibilité lors de la spécification d'une analyse. En effet, le sujet d'analyse n'est pas prédéterminé mais sera choisi par le décideur, parmi les dimensions, lors de l'expression de son besoin. Nous avons proposé une méthode de conception semi-automatique qui vise à élaborer des modèles en galaxie à partir des arbres unifiés. Cette méthode s'articule autour de quatre étapes : *(i) le prétraitement des arbres* améliore leur lisibilité conceptuelle en les enrichissant par des cardinalités qui aident ultérieurement à l'extraction des éléments multidimensionnels ; *(ii) la construction de modèles en galaxie* identifie les éléments du modèle, elle est fondée sur des règles que nous

avons proposées ; *(iii) l'approbation* donne la main au décideur/concepteur pour approuver les modèles multidimensionnels obtenus. Finalement, *(iv) la validité syntaxique* des modèles en galaxie est vérifiée via un ensemble de règles (Ben Messaoud, et al., 2011b) (Feki, et al., 2013). Pour valider la méthode de modélisation multidimensionnelle proposée, nous avons mis en œuvre un outil logiciel nommé ***Galaxy-Gen***.

En ce qui concerne l'expérimentation des deux méthodes proposées, nous avons utilisé deux corpus : ***un corpus académique*** et ***un corpus médical***. Le corpus académique est composé de 20 documents XML du domaine académique décrits par quatre DTDs. Le corpus médical est constitué de 1691 documents XML de la collection médicale *Clef 2007*, est décrit par trois DTDs. Nous avons réalisé des expériences d'unification et de modélisation en galaxie sur les documents de ces deux corpus (Ben Messaoud, et al., 2014).

Pour le corpus académique, l'application de la méthode d'unification a généré une DTD unifiée et celle de la méthode de modélisation en galaxie a produit un modèle en galaxie composé de cinq dimensions reliées par un nœud central.

Concernant le corpus de la collection médicale *Clef 2007,* l'unification a produit une DTD unifiée et la méthode de modélisation a permis l'obtention d'un modèle en galaxie constitué de cinq dimensions interconnectées par un nœud.

L'évaluation de la méthode d'unification nous a permis de prouver que le résultat d'unification des documents de chacun des deux corpus, est transformé en une DTD respectant la validité des documents XML. Pour cette vérification nous avons utilisé *XMLSpy*.

En ce qui concerne l'évaluation de la méthode de modélisation, nous avons utilisé un ensemble de requêtes décisionnelles ; cette expérimentation a permis de montrer que le modèle en galaxie généré est utile pour exprimer des requêtes décisionnelles sur le contenu textuel des documents entreposés. Pour répondre à ces requêtes, nous avons utilisé les opérateurs du langage de manipulation multidimensionnelle défini pour les galaxies dans (Tournier, 2007).

Perspectives de recherche

Parmi les perspectives de recherche immédiates, nous estimons que la proposition d'un langage de requête convivial facilitera l'interrogation d'un entrepôt par les décideurs. Quant au long terme, les perspectives devraient être normalement plus ambitieuses et nombreuses.

Nous nous contentons de nous poser des questions dans le contexte des nouvelles tendances de recherches comme les *Big Data*, le *Cloud computing* et les technologies qui s'y rapportent.

Proposition d'un langage de requêtes pour l'entrepôt de documents. Dans la littérature, (Tournier, 2007) a proposé un langage de manipulation multidimensionnelle permettant la manipulation des galaxies via un ensemble d'opérateurs. Ce langage aide le décideur à spécifier ses besoins analytiques. Cependant, l'usage d'une syntaxe bien précise oblige le décideur à comprendre le langage pour pouvoir l'utiliser. Pour alléger cette tâche au décideur, nous envisageons de proposer une grammaire en langage quasi-naturel pour aider le décideur dans l'expression de ses besoins. Le choix du langage naturel est motivé par sa facilité de compréhension par le décideur.

Gestion du volume croissant des documents. La production croissante des documents et le partage des informations entre les décideurs engendrent de très gros volumes de documents disponibles et utiles à analyser. Il devient difficile de stocker ces données très volumineuses (*Big data*), les interroger, les modéliser et les analyser. Par conséquent, nous pensons qu'il serait intéressant de redéfinir l'architecture d'un entrepôt de documents dans un contexte de *Big Data*.

Les documents dans les nuages. Avec l'avènement du *Cloud Computing* ou *l'informatique dans les nuages* pour les systèmes informatiques, les systèmes décisionnels sont entrain de bénéficier de ce nouveau paradigme. Ce dernier se caractérise par une délocalisation des données / des documents dans les nuages. Il est intéressent de délocaliser l'entrepôt de documents XML dans le *Cloud.* Ceci soulève un ensemble de verrous scientifiques tels le temps de réponse des requêtes analytiques, la sécurité des documents entreposés, la répartition et la distribution des données sur le *Cloud*.

Liste des publications de la thèse

Articles de revues indexés

1) Jamel Feki, Ines Ben Messaoud, Gilles Zurfluh, "*Building an XML Document Warehouse*", Journal of Decision Systems, Ed. Taylor & Francis, Vol. 22, n° 2/2013, pp. 122-148, DOI: 10.1080/12460125.2013.780322.
2) Ines Ben Messaoud, Jamel Feki, Gilles Zurfluh, "*Modélisation multidimensionnelle des documents XML"*, Revue des Nouvelles Technologies de l'Information (RNTI), Ed. Cépaduès, vol. B-7, pp. 55-70, 2011, ISBN 97827056 8127 2.

Articles de conférences avec comité de lecture

3) Ines Ben Messaoud, Jamel Feki, Gilles Zurfluh, *"Galaxy-Gen: A Tool for Building Galaxy Model from XML Documents"*, International Conference on Knowledge Engineering and Ontology Development (KEOD'14), 21-24 October 2014, Rome, Italy.
4) Ines Ben Messaoud, Jamel Feki, Gilles Zurfluh, "*A First Step for Building a Document Warehouse: Unification of XML Documents"*, International Conference on Research Challenges in Information Science (RCIS'12), 16-18 Mai 2012, pp. 59-64, IEEE 2012 ISBN 978-1-4577-1938-7, Valence, Spain.
5) Haithem Aouabed, Ines Ben Messaoud, Jamel Feki, Gilles Zurfluh, "*USD : Un outil d'unification des structures des documents XML"*, Atelier des Systèmes Décisionnels (ASD'12), 1-3 Avril 2012, pp. 83-94, Blida, Algérie.
6) Ines Ben Messaoud, Jamel Feki, Kais Khrouf, Gilles Zurfluh, "*Unification of XML Document structures for Document Warehouse (DocW)",* International Conference on Entreprise Information Systems (ICEIS'11), 8-11 June 2011, pp. 85-94, ISBN 978-989-8425-53-9, Beijing, China.

7) Ines Ben Messaoud, Jamel Feki, Gilles Zurfluh, "*Unification des structures des documents XML pour l'entreposage de documents"*, Atelier des Systèmes Décisionnels (ASD'10), 5-6 Novembre 2010, pp. 1-12, Sfax, Tunisie.

BIBLIOGRAPHIE GENERALE

A, B

[*A. Miller*, 1995] A. Miller George, *WordNet: A Lexical Database for English*, Magazine Communications of the ACM, New York, NY, USA, Novembre 1995, Vol. 38, pp. 39 - 41.

[*Akoka, et al., 1998*] Akoka Jacky, Comyn-Wattiau Isabelle et Kaded Zoubida, *Combining View Integration and Schema Clustering to Improve Database Design.* Actes XIVèmes Journées Bases de Données Avancées, Hammamet, Tunisie, 1998.

[*Aouabed, et al., 2012*] Aouabed Haithem, Ben Messaoud Ines, Feki Jamel et Zurfluh Gilles, *USD: Un outil d'unification des structures des documents XML*. 6ème Atelier des Systèmes Décisionnels ASD'12, Blida, Algérie, 2012, pp. 83-94.

[*Ben Abdallah, 2010*] Ben Abdallah Mounira, *Un cadre de conception d'entrepôts de données à base de patrons multidimensionnels.* Thèse de doctorat en Informatique, Université de Sfax, Sfax, Tunisie, 2010.

[*Ben Abdallah, et al., 2008*] Ben Abdallah Mounira, Feki Jamel et Ben-Abdallah Hanen, *Patrons multidimensionnels contraints*. Conférence Systèmes d'Information et Intelligence Economique SIIE'08, Hammamet, Tunisie, 2008.

[*Ben Mefteh, et al., 2013*] Ben Mefteh Salma, Khrouf Kais, Feki Jamel, Ben Kraiem Maha et Soule-Dupuy Chantal, *Semantic Structure for XML Documents: Structuring and Pruning.* Journal of Information Organization (JIO), Vol. 3., N°1, Mars 2013, pp. 37-46.

[*Ben Messaoud, et al., 2014*] Ben Messaoud Ines, Feki Jamel et Zurfluh Gilles, *Galaxy-Gen: A Tool for Building Galaxy model from XML documents*. 6th International Conference on Knowledge *Engineering* and Ontology Development KEOD'14, Rome, Italie, 21-24 October 2014.

[*Ben Messaoud, et al., 2012*] Ben Messaoud Ines, Feki Jamel et Zurfluh Gilles, *A First Step for Building a Document Warehouse: Unification of XML Documents*. Proceeding of Sixth International Conference on Research Challenges in Information Science RCIS'12, Valencia, Spain, 16-18 Mai 2012, pp. 59-64.

[*Ben Messaoud, et al., 2011a*] Ben Messaoud Ines, Feki Jamel, Khrouf Kais et Zurfluh Gilles, *Unification of XML Document Structures for Document Warehouse (DocW)*. 13th

International Conference on Entreprise Information Systems ICEIS'11, Beijing, Chine, 2011, pp. 85-94.

[*Ben Messaoud, et al., 2011b*] Ben Messaoud Ines, Feki Jamel et Zurfluh Gilles, *Modélisation multidimensionnelle des documents XML*. Septièmes journée francophones sur les Entrepôts de Données et d'Analyse en ligne EDA'11, Clermont Ferrand, France, 2011, Hermann, Vol. B-7, pp. 55-70.

[*Ben Messaoud, et al., 2010*] Ben Messaoud Ines, Feki Jamel et Zurfluh Gilles, *Unification des structures des documents XML pour l'entreposage de documents*. 5ème Atelier sur les Systèmes Décisionnels ASD'10, Sfax, Tunisie, 2010, pp. 1-12.

[*Boussaid, et al., 2006*] Boussaid Omar, Ben Messaoud Riadh, Choquet Rémy, Anthoard Stéphane, *Conception et construction d'entrepôts XML*. *2ème journée francophone sur* les Entrepôts de Données et l'Analyse en ligne HYPERLINK "http://www.informatik.uni-trier.de/~ley/db/conf/eda/eda2006.html" \l "BoussaidMCA06" EDA'06, Versailles, France, 2006 , pp. 3-22.

C, D

[*Carpani, et al., 2001*] Carpani Fernando et Ruggia Raul, *An Integrity Constraints Language for a Conceptual Multidimensional Data Model.* 13th International Conference on Software Engineering & Knowledge Engineering SEKE'01, Argentina, 2001.

[*De-Meo, et al., 2003*] De-Meo Pasquale, Quattrone Giovanni, Terracina Giorgio et Ursino Domenico, *"Almost automatic" and semantic integration of XML Schemas at various "severity" levels.* Procedings of the International Conference on Cooperative Information Systems (CoopIS), Taormina, Italy, 2003.

F, G

[*Feki, 2004*] Feki Jamel, *Vers une conception automatisé des entrepôts de données : Modélisation des besoins OLAP et génération de schémas multidimensionels*. 8th Maghrebian Conference on Software Engineering and Artificial Intelligence, MCSEAI'O4, Sousse, Tunisie, 2004, pp. 473-485.

[*Feki, et al., 2013*] Feki Jamel, Ben Messaoud Ines et Zurfluh Gilles, *Building an XML Document Warehouse*. Journal of Decision System JDS'13, Vol. 22, The DOI is 10.1080/12460125.2013.780322.

[*Fuhr, et al., 2001*] Fuhr Norbert et Grobjohann Kai, *XIRQL: a query language for information retrieval in XML documents*. 24th International ACM Conference on Research and Developement in Information Retrieval (SIGIR), ACM Press, 2001, pp. 172-180.

[*Ghozzi, 2004*] Ghozzi Faiza, *Conception et manipulation de bases de données dimensionnelles à contraintes*. Thèse de doctorat en Informatique, Université Paul Sabatier, Toulouse, France, 2004.

H, I, J

[*Hachaichi, et al., 2010*] Hachaichi Yesser, Feki Jamel et Ben-Abdallah Hanen, *Modélisation multidimensionnelle de documents XML centrés-données.* Journal of Decision Systems, JDS'10, Vol. 19/3, 2010, pp. 313-345.

[*Hurtardo, et al., 2002*] Hurtardo Carlos A. et Mendelzon Alberto O., *OLAP Dimension Constraints*. 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems PODS'02, Madison, USA, 2002, pp. 169-179.

[*Inmon, 1994*] Inmon William H, *Building the data warehouse*. John Wiley&Sons, 1994.

[*Jaro, 1989*] Jaro M. A. Advances in record linking methodology as applied to the 1985 census of Tampa Florida. Journal of the American Statistical Society, 1989, pp. 414-420.

K, L

[*Kamps, et al., 2004*] Kamps Jaap, Marx Maarten, De Rijke Maarten et Sigurbjornsson Borkur, *Best-Match Querying from Document-Centric XML*. Proceedings of the Seventh International Workshop the Web and Databases, 2004, pp. 55-60.

[*Khrouf, 2004*] Khrouf Kais, *Entrepôts de documents : De l'alimentation à l'exploitation*. Thèse de doctorat en Informatique, Université Paul Sabatier, Toulouse, France, 2004.

[*Khrouf, et al., 2003*] Khrouf Kais, Ravat Frank et Soulé-Dupuy Chantal, *Comparaison et fusion de structures logiques de documents semi-structurés*. Lavoisier, 2003, Paris, France, Vol. 8., pp. 127-151.

[*Kimball, 1997*] Kimball Ralph, *The Data Warehouse Toolkit*, John Wiley and Sons, 1997.

[*Lee, et al., 2002*] Lee Mong Li, Yang Liang Huai, Hsu Wynne et Yang Xia, *XClust: Clustering XML Schmas for Effective Integration*. The ACM Internationl Conference on Information and knowledge Management, Mclean, Virginia, 2002, pp. 292-299.

M, P

[*Malinowski et al., 2006*] Malinowski Elzbieta et Zimànyi Esteban Hierarchies, A multidimensional model : From concrptual modeling to logical representation. Data & Knowledge Engineering (DKE), Elsevier, Novembre 2006. Vol. 59(2), pp. 348-377.

[*McCabe, et al., 2000*] McCabe Catherine, Lee Jinho, Chowdhury Abdur, Grossman David et Frieder Ophir, *On the design and evaluation of a multi-dimensional approach to information retrieval*. 23rd International Conference on Research an Development in Information Retrieval, SIGIR, 2000, pp. 363-365.

[*Mello, et al., 2005*] Mello Ronaldo Dos Santos et Heuser Carlos Alberto, *BInXS: A Process for Integration of XML Schemata*. 17th International Conference on Advanced Information Systems Engineering, CAiSE'05, Porto, Portugal, 2005, pp. 151-166.

[*Mello, et al., 2002*] Mello Ronaldo Dos Santos, Castano Silvana et Heuser Carlos Alberto, *A method for the unification of XML schemata*, Information and Software Technology 44, 2002, pp. 241-249.

[*Pérez-Martinez et al., 2008*] Pérez-Martínez Juan Manuel, Berlanga-Llavori Rafael, Aramburu-Cabo María José et Pedersen Torben Bach, *Contextualizing data warehouses with documents*. Decision Support Systems (DSS), 2008, pp. 77-94.

[*Pérez-Martinez, 2007*] Pérez-Martinez Juan Manuel, *Contextualizing a data warehouse with documents,* Thèse de doctorat, Université Jaume I, Spain, 2007.

[*PRISM, 2000*] PRISM, *Conception, intégration et évolution des systèmes d'information*. Rapport du Laboratoire PRISM, 2000.

[*Pujolle et al., 2011*] Pujolle Geneviève, Ravat Franck, Teste Olivier, Tournier Ronan et Zurfluh HYPERLINK Gilles, *Multidimensional Database Design from Document-Centric XML Documents*. DaWaK, 2011, pp. 51-65.

R, S, T

[*Ravat et al., 2010*] Ravat Franck, Teste Olivier, Tournier Ronan et Zurfluh Gilles, *Finding an application-appropriate model for XML data warehouses*. Information Systems 35, 2010.

[*Ravat et al., 2007*] Ravat Franck, Teste Olivier et Tournier Ronan, *Analyse multidimensionnelle de documents via des dimensions OLAP*. Document numérique, Hermès, Numéro spécial Entreposage de documents et données semi-structurées, 2007, pp. 85-104.

[*Sullivan, 2001*] Sullivan D., *Document Warehousing and Text Mining: Techniques for Improving Business Operations, Marketing and Sales*. John Wiley & Sons, 2001.

[*Teste, 2000*] Teste Olivier, *Modélisation et manipulation d'entrepôts de données complexes et historisées*. Thèse de doctorat en informatique, Toulouse, France, 2000.

[*Tournier, 2007*] Tournier Ronan, *Analyse en ligne (OLAP) des documents.* Thèse de doctorat en Informatique, Universtié Toulouse III, Paul Sabatier, Toulouse, France, 2007.

[*Tseng et al., 2006*] Tseng Frank S. C. et Chou Annie Y. H., *The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence.* Decision Support Systems, 2006, pp. 727-744.

Y, Z

[*Yoo et al., 2005*] Yoo Chun-Sik, Woo Seon-Mi et Yong-Sung Kim, *Unification of XML DTD for XML Documents with Similar Structure*. Computational Science and its Applications, ICCSA'05, 2005, pp. 954-963.

[*Zhang et al., 2002*] Zhang Yan-Feng et Liu Wei-Yi, *Semantic integration of XML Schema*. First International Conference on Machine Learning and Cybernetics, Beijing, 2002.

ANNEXE

Annexe 1 : Etapes de construction du dictionnaire des acronymes

Pour construire le dictionnaire des acronymes, nous avons suivi les étapes suivantes pour chaque ensemble d'arbres à unifier :

1- Extraire les noms des nœuds à partir de tous les arbres : $M = \{mot_1, ..., mot_n\}$

2- Pour chaque mot $\in M$, accéder à la base lexicale *Wordnet* pour lui trouver un synonyme

Construire un ensemble M_s des mots ayant un synonyme

A la fin de cette étape, on obtient un sous ensemble $M_a = M - M_s = \{mota_1, ..., mota_m\}$.

3- Pour chaque $mota_i \in M_a$

a- Calculer la distance de *Jaro Winkler* entre $mota_i$ et chaque *mot* entier ($mot \in M_s$).

b- Trier les valeurs des distances de Jaro Winkler obtenues : la distance maximale entre $mota_i$ et *mot* nous permet de déduire si *mot* peut être la forme complète de $mota_i$.

c- Ajouter une nouvelle entrée au dictionnaire des acronymes pour indiquer que $mota_i$ est l'acronyme de *mot*.

Annexe 2 : Quelques documents XML du corpus académique

Document XML n°1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Article SYSTEM "F:\ DTD1.dtd">
<Article>
 <Title>Unification of XML DTD for XML Documents with Similar Structure</Title>
  <Auth>
   <Name>Chun-Sik Yoo</Name>
   <Affiliation>Division of Electronics and Information Engineering, Chonbuk National University,
   664-14 1ga Duckjin-Dong, Duckjin-Gu, Jeonju, Jeonbuk, 561-756, Republic of korea</Affiliation>
  </Auth>
  <Auth>
   <Name>Seon-Mi Woo</Name>
   <Affiliation>Division of Electronics and Information Engineering, Chonbuk National University,
   664-14 1ga Duckjin-Dong, Duckjin-Gu, Jeonju, Jeonbuk, 561-756, Republic ofKorea</Affiliation>
  </Auth>
  <Auth>
   <Name>Yong-Sung Kim</Name>
   <Affiliation>Division of Electronics and Information Engineering, Chonbuk National University,
   664-14 1ga Duckjin-Dong, Duckjin-Gu, Jeonju, Jeonbuk, 561-756, Republic ofKorea</Affiliation>
  </Auth>
  <Section>
   <Title>Abstract</Title>
   <Subsection>
    <Para>There are many cases that XML documents have different DTDs in spite of having a
similar structure and being logically the same kind of document.
...
...
 And we apply a proposed algorithm to unify DTDs of science journals.
    </Para>
   </Subsection>
 </Section>
 <Section>
  <Title>1 Introduction</Title>
  <Subsection>
   <Para>XML documents declare DTD (Document Type Definition) to define the structure of
document, and perform the strict document structure validation using this DTD.
…
…
Also, the users can get more effective management and administration environment for XML
document database.
    </Para>
  </Subsection>
 </Section>
  ...
  …
 <Day>9-12</Day>
 <Month>May</Month>
</Article>
```

Document XML n°2

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Article SYSTEM "F:\ DTD2.dtd">
<Article>
 <Title>SEMANTIC INTEGRATION OF XML SCHEMA</Title>
 <Writer>
  <Name>YAN-FENG ZHANG</Name>
  <Institute>Department of Computer science, Kunming, Yunnan University, China</Institute>
 </Writer>
 <Writer>
  <Name>WEI-YI LIU</Name>
  <Institute>Department of Computer science, Kunming, Yunnan University, China</Institute>
 </Writer>
 <Abstract>The availability of large amounts of heterogeneous distributed web data necessitates
the integration of XML data from multiple XML sources.
…
…
Our integration process includes three steps: clustering of concepts, unification of concepts, and
restructuring of relationships. Finally, a global conceptual model is provided for users.</Abstract>
 <Mots_Clefs>XML</Mots_Clefs>
 <Mots_Clefs>XML schema</Mots_Clefs>
 <Mots_Clefs>Integration</Mots_Clefs>
 <Mots_Clefs>UML</Mots_Clefs>
 <Body>
  <Paragraph>1 Introduction
XML [1] is a common standard for semi-structured and structured data representation and exchange
over the web.
…
…
The remainder of this paper is organized as follows. Section 2 comments some related work.
Section 3 presents how to convert XML Schema to UML diagram. Section 4 describes and
exemplifies the process of semantic integration and section 5 is dedicated to the conclusion.
  </Paragraph>
…
…
 </Body>
 <References>[1] W3C Extensible Markup Language (XML). Available at: http://www.w3.org/xml
 </References>
 <References>[2] W3C XML Schema, Available at: http:/lwww .w3.org/xml/schema.html
 </References>
…
…
 <References>[11] Chuang-Hue Moh, Ee-Peng Lim, Wee-Keong Ng. Reengineering structures from
web documents. Proceeding of the fifth ACM conference on digital libraries. San Antonio USA (June
2-7,2000)
 </References>
 <Day>4-5</Day>
 <Month>November</Month>
 <Year>2002</Year>
</Article>
```

Document XML n°3

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Article SYSTEM "F:\ DTD3.dtd">
<Article>
 <Title>Modelling Stars Using XML</Title>
 <Writer>
  <Name>Jaroslav Pokorný</Name>
  <University>Charles University Dep. of Software Engineering Praha, Czech Republic</University>
 </Writer>
 <Outline>
We suppose collections of XML data described by Document Type Definitions (DTDs). This data has
been generated by applications and plays a role of OLTP database(s).
…
…
 </Outline>
 <Mots_Clefs>Data warehouse</Mots_Clefs>
 <Mots_Clefs>XML</Mots_Clefs>
 <Mots_Clefs>dimension</Mots_Clefs>
 <Mots_Clefs>star schema</Mots_Clefs>
 <Section>
  <Section_Number>1</Section_Number>
  <Title>INTRODUCTION</Title>
  <Paragraph>With the recent popularity of the WWW, an enormous amount of heterogeneous
information is now available in enterprises. Such data stores may be classical formatted databases
but also data collections coming from e-mail communication, e-business, or from inner digital
documents that are produced by applications in enterprise.
  </Paragraph>
…
…
  <Paragraph>The paper is organized as follows. Section 2 introduces the main concepts and
notions of DM based on tables, and states some restrictions chosen for the approach in the paper. In
Section 3 we give a brief overview over XML and present a model for XML. Section 4 defines notions
needed for characterization of XML collections, for specifying XML-referential integrity, and for
establishing dimensions over XML data. We define XML-star schemes with explicit dimension
hierarchies. Finally, we summarize the approach and point out further research issues.
  </Paragraph>
  <Figure>Figure 1: XML-star schema</Figure>
 </Section>
…
…
 <Year>2001</Year>
</Article>
```

Document XML n°4

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Article SYSTEM "F:\ DTD4.dtd">
<Article>
  <Tit>BInXS: A Process for Integration of XML Schemata</Tit>
  <Writer>
    <Name>Ronaldo dos Santos Mello</Name>
    <Affiliation>Universidade Federal de Santa Catarina, Depto. de Informatica e Estatistica, Cx.
    Postal 476, Florianopolis, SC, Brasil 88040-900</Affiliation>
  </Writer>
  <Writer>
    <Name>Carlos Alberto Heuser</Name>
    <Affiliation>Universidade Federal do Rio Grande do Sul, Instituto de Informatica, Cx. Postal
    15064, Porto Alegre, RS, Brasil 91501-970</Affiliation>
  </Writer>
  <Abstract>
This paper presents a detailed integration process for XML schemata called BInXS. BInXS adopts a
global-as-view integration approach that builds a global schema from a set of heterogeneous XML
schemata related to a same application domain.
…
…
In addition, BInXS supports a mapping strategy based on XPath expressions in order to maintain
correspondences among global concepts and data at the XML sources.
  </Abstract>
  <Body>
1 Introduction
The XML format has been extensively used to represent data as well as to interchange data among
users and applications, specially through the Web [7].
…
…
This paper is organized as follows. Section 2 gives an overview of the integration process followed
by BInXS. Section 3 describes the conversion of an XML schema to a conceptual schema. Section 4
describes how the global schema is defined from the unification of conceptual schemata. Section 5
discusses some related work. Section 6 is dedicated to the conclusion.

2 BInXS Overview
BInXS is a semi-automatic and bottom-up process for semantic integration of XML schemata [25].
…
…
 An external tool, called ARTEMIS, is used to find out semantic affinities between concepts in
different schemata. User intervention is considered again to eventually choose one among several
alternative semantic meanings for a global concept or relationship representation, or to validate an
automatic-generated preliminary global schema. Section 4 details this phase.
…
…
References
1. CXML.org. Available at: lt http://www.cxml.org gt, mar 2005.
…
…
32. X. Yang, M. L. Lee, and T. W. Ling. Resolving Structural Conflicts in the Integration of XML
Schemas: A Semantic Approach. In 22th International Conference On Conceptual Modeling (ER),
pages 520–533, Chicago, USA, 2003. Springer-Verlag.
  </Body>
</Article>
```

Annexe 3 : Quelques documents XML de la collection médicale Clef 2007

Document XML n°1

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE CASIMAGE_CASE SYSTEM "F:\CLEF 2007\FR\casimage_FR\DTD1_Clef 2007.dtd">
<CASIMAGE_CASE>

 <ID>2163</ID>
 <Description>ECHOGRAPHIE ABDOMINALE DU 29/6/1998: La vésicule biliaire présente une paroi très épaissie, hyperéchogène et à bords irréguliers avec présence de petites plages hypoéchogènes
intrapariétales et d' une lame liquidienne dans le lit vésiculaire. Le contenu vésiculaire montre une bande hyperéchogène d'avant en arrière sans net nodule individualisé et sans évidence de cône d' ombre, pouvant correspondre à de la bile calcique (images : us 1 à 3). Ces différentes images parlent en faveur d' une cholécystite aiguÎ gangréneuse.</Description>
 <Diagnosis>Cholécystite aiguÎ gangréneuse</Diagnosis>
 <Sex />
 <CaseID />
 <ClinicalPresentation>PATIENTE DE 69 ANS AVEC DOULEURS ABDOMINALES DANS L'HYPOCHONDRE DROIT. Il s'agit d'une patiente de 69 ans se présentant pour des douleurs abdominales situées au niveau de l' hypocondre droit</ClinicalPresentation>
…
…
 <ClinicalPresentation>l' état de la patiente se péjore. Elle devient confuse et agitée. Elle se plaint de frissonner et d' avoir des sudations froides. Sur le plan cardio-pulmonaire elle est tachypnéique et présente une hypotension artérielle avec une valeur systolique à 85 mmHG et une fréquence cardiaque à 110/minutes.</ClinicalPresentation>
 <Commentary>En ce qui concerne la prise en charge des patients il convient dans un premier temps de les hospitaliser, de leur assurer un support hémodynamique sous la forme d'une perfusion de liquide et de leur administrer des antibiotiques systémiques.
…
…
La mortalité de la cholécystite aiguÎ est de 5 à 10% et pratiquement entièrement confinée aux patients de plus de 60 ans souffrant de comorbidités importantes et à ceux qui développent les complications de la cholécystite aiguÎ.</Commentary>
 <KeyWords>Cholecystitis</KeyWords>
 <KeyWords>gallbladder</KeyWords>
 <KeyWords>stone</KeyWords>
 <Anatomy />
 <Chapter>Foie et VB</Chapter>
 <ACR>762.285</ACR>
 <References />
 <Author>Frank Kolo</Author>
 <Reviewer>F Terrier</Reviewer>
 <Reviewer>N Dfouni</Reviewer>
 <Hospital>HUG</Hospital>
 <Department>Département de Radiologie</Department>
 <State />
 <Date>21.03.2003</Date>
 <Language>French</Language>
 <Title>AMC Techniques d'Urgence/Abdomen</Title>
 <Birthdate>18.08.2003</Birthdate>
 <Age>69</Age>
```

```
  <ImageThumbnailID>8729</ImageThumbnailID>
  <Creation>23.11.2001</Creation>
  <DateTime>18:04:37</DateTime>
  <Order>1011</Order>
  <OJoint />
  <OLocation />
  <OImplant />
  <ODislocation>0</ODislocation>
  <OPolytrauma>0</OPolytrauma>
  <OOpen>0</OOpen>
  <OPathologic>0</OPathologic>
  <OOperation>00.00.00</OOperation>
  <OGraft>0</OGraft>
  <WEBURL>http://129.195.254.38:5000/4DMETHOD/_HTML_MCase/2163</WEBURL>
</CASIMAGE_CASE>
```

Document XML n°2

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE CASIMAGE_CASE SYSTEM "F:\CLEF 2007\FR\casimage_FR\DTD2_Clef 2007.dtd">
<CASIMAGE_CASE>
  <ID>1683</ID>
  <Description/>
  <Diagnosis>Frottis sanguin périphérique normal (leucocytes)</Diagnosis>
  <Sex/>
  <CaseID/>
  <ClinicalPresentation>Il s'agit d'une donneuse de sang de 44 ans. Son hémogramme est le suivant
: Hb 142 g/L</ClinicalPresentation>
…
…
  <ClinicalPresentation>les plaquettes sont visualisées. La coloration du frottis est une coloration
panoptique de Wright.</ClinicalPresentation>
  <Commentary/>
  <KeyWords>globules blancs</KeyWords>
  <KeyWords>leucocytes</KeyWords>
  <KeyWords>globules rouges</KeyWords>
  <KeyWords>érythrocytes</KeyWords>
  <KeyWords>plaquettes</KeyWords>
  <Anatomy/>
  <Chapter/>
  <ACR/>
  <References/>
  <Author>BERIS Photis</Author>
  <Reviewer/>
  <Hospital>HUG</Hospital>
  <Department>Hématologie</Department>
  <State/>
  <Date>21.03.2003</Date>
  <Language>French</Language>
  <Title>Hématologie-APP-2ème</Title>
  <Birthdate>18.08.2003</Birthdate>
  <Age>44</Age>
  <ImageThumbnailID>6356</ImageThumbnailID>
  <Creation>15.01.2003</Creation>
  <DateTime>18:05:05</DateTime>
  <Order>0</Order>
  <OJoint/>
  <OLocation/>
  <OImplant/>
  <ODislocation>0</ODislocation>
  <OPolytrauma>0</OPolytrauma>
  <OOpen>0</OOpen>
  <OPathologic>0</OPathologic>
  <OOperation>00.00.00</OOperation>
  <OGraft>0</OGraft>
  <QCM>
    <QUESTION>Le MCV (volume corpusculaire moyen) de la donneuse en question est :
REMARQUE : Pour calculer le MCV, vous devez connaître l'hématométrie (globules rouges,
hémoglobine, hématocrite). Attention : ne pas confondre les unités. Vous devez
rendre le résultat en fL.</QUESTION>
    <ANSWERA>86 fL</ANSWERA>
```

```
   <ANSWERB>66 fL</ANSWERB>
   <ANSWERC>108 fL</ANSWERC>
   <ANSWERD>34 fL</ANSWERD>
    <COMMENTARY/>
  </QCM>
…
…
   <QCM>
    <QUESTION>Le lymphocyte stimulé se trouve sur l'image :</QUESTION>
    <ANSWERA>FSP-6</ANSWERA>
    <ANSWERB>FSP-4</ANSWERB>
    <ANSWERC>FSP-5</ANSWERC>
    <ANSWERD>FSP-7</ANSWERD>
    <COMMENTARY/>
   </QCM>
   <WEBURL>http://129.195.254.38:5000/4DMETHOD/_HTML_MCase/1683</WEBURL>
</CASIMAGE_CASE>
```

Document XML n°3

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE CASIMAGE_CASE SYSTEM "F:\CLEF 2007\FR\casimage_FR\DTD3_Clef 2007.dtd">
<CASIMAGE_CASE>

 <ID>2973</ID>
 <Description>Diverticule de Meckel, avec juxta-posé un petit abcès.</Description>
 <Diagnosis>Diverticule de Meckel</Diagnosis>
 <Sex />
 <CaseID />
 <ClinicalPresentation>Douleurs abdominales</ClinicalPresentation>
 <Commentary />
 <KeyWords />
 <Anatomy>jambe</Anatomy>
 <Chapter>Digestif</Chapter>
 <ACR />
 <References />
 <Author>ROSSET Antoine</Author>
 <Reviewer />
 <Hospital>HUG</Hospital>
 <Department>Département de Radiologie</Department>
 <State />
 <Date>21.03.2003</Date>
 <Language>French</Language>
 <Title>Demo Collection</Title>
 <Birthdate>18.08.2003</Birthdate>
 <Age>37</Age>
 <ImageThumbnailID>11675</ImageThumbnailID>
 <Creation>29.03.2002</Creation>
 <DateTime>18:04:24</DateTime>
 <Order>7</Order>
 <OJoint>Foot</OJoint>
 <OLocation>Wrist</OLocation>
 <OImplant>Plate - Other</OImplant>
 <ODislocation>0</ODislocation>
 <OPolytrauma>0</OPolytrauma>
 <OOpen>0</OOpen>
 <OPathologic>0</OPathologic>
 <OOperation>02.02.2002</OOperation>
 <OGraft>0</OGraft>
 <QCM>
  <QUESTION>Diagnostic?</QUESTION>
  <ANSWERA>Diverticule de Meckel inflammé</ANSWERA>
  <ANSWERB>Diverticulite</ANSWERB>
  <ANSWERC>Appendicite</ANSWERC>
  <ANSWERD>Perforation colique</ANSWERD>
  <COMMENTARY>Superbe cas de diverticule de Meckel surinfecté par un petit abcès!
</COMMENTARY>
 </QCM>
 <WEBLINK>
  <URL>http://www.mc.vanderbilt.edu/peds/pidl/gi/meckel.htm</URL>
  <DESCRIPTION>Le diverticule de Meckel (anglais)</DESCRIPTION>
 </WEBLINK>
 <WEBLINK>
```

```
    <URL>http://www.sante.univ-nantes.fr/decas/certif00/Abdomen/Meckel_brute.htm</URL>
    <DESCRIPTION>Un diverticule de Meckel à l'entéroclyse</DESCRIPTION>
  </WEBLINK>
  <WEBURL>http://129.195.254.38:5000/4DMETHOD/_HTML_MCase/2973</WEBURL>
</CASIMAGE_CASE>
```

Résumé :

Les documents constituent une capitalisation importante des connaissances. Généralement, ces documents sont caractérisés par un contenu peu structuré et il est alors difficile de les intégrer dans les systèmes d'information décisionnels. En conséquence, les décideurs ne peuvent pas tirer profit de ces documents. Pour répondre à cette problématique, nous proposons une approche de construction du schéma de l'entrepôt de documents XML. Cette approche se compose de deux méthodes : une *méthode d'unification* des structures des documents XML et une *méthode de modélisation multidimensionnelle* de ces documents. La méthode d'unification permet de définir une structure commune pour décrire les documents XML hétérogènes et appartenant au même domaine. Pour valider cette méthode, un outil logiciel baptisé **USD** (**U**nification of **S**tructures of XML **D**ocuments) est développé. La méthode de modélisation multidimensionnelle a pour but de concevoir semi-automatiquement le schéma du magasin de documents, selon le modèle multidimensionnel en galaxie, à partir d'une structure XML unifiée. Afin de valider cette méthode, un outil nommé **Galaxy-Gen** (**Galaxy Gen**eration) est développé.

Mots-clés :

Entrepôt de documents, Document XML, Modélisation multidimensionnelle des documents, Unification de documents XML, Modèle en galaxie.

Abstract

Documents represent an important knowledge capitalization. In general, these documents are characterized by unstructured content, and therefore it is difficult to integrate them in the decision information systems. As a result, decision-makers are unable to exploit these documents easily and efficiently. To alleviate this problem, we propose an approach to build the schema of the XML documents warehouse. This approach consists of two methods: a *method for unification* of the structures of XML documents and a *method for multidimensional modeling* of these documents. The unification method defines a common structure to describe heterogeneous XML documents belonging to the same domain. To validate this method, a software tool called **USD** (**U**nification of **S**tructures of XML **D**ocuments) is developed. While the method of multidimensional modeling builds semi-automatically the schema of the documents mart as a galaxy model. To validate this method, the tool called **Galaxy-Gen** (**Galaxy Gen**eration) is developed.

Keywords

Documents warehouse, XML document, Multidimensional modeling of documents, Unification of XML documents, Galaxy model.

الخلاصة

تتضمن الوثائق قيمة معرفية مهمة. وهي عادة تتميز بمحتوي غير منظم وبالتالي فمن الصعب دمجها في نظم معلومات اتخاذ القرارات. ونتيجة لذلك لا يمكن لصانع القرارات أن يستفيد من هذه الوثائق. لمعالجة هذه المشكلة، اقترحنا منهجا لبناء رسم بياني لمخازن الوثائق. هذا المنهج يتكون من طريقتين : طريقة لتوحيد هياكل الوثائق و طريقة للتصميم المتعدد الأبعاد للوثائق. طريقة التوحيد تسمح للإيجاد هيكل مشترك لوصف وثائق غير متجانسة و منتمية لنفس المجال. أنجزنا برمجيّة للتحقق من صحة هذه الطريقة. أما طريقة التصميم متعدد الأبعاد فهي تهدف للتصميم الشبه التلقائي للرسم بياني لمخازن الوثائق وذلك وفقا للنموذج المتعدد الأبعاد الكوكبي. لقد أنجزنا برمجية **(Galaxy-Gen)** لتقييم المنهجية المقترحة.

الكلمات المفاتيح

مخازن الوثائق، وثيقة اكس ام ال، التصميم المتعدد الأبعاد للوثائق، توحيد الوثائق، نموذج كوكبي.