Appariement d'informations dans les entrepôts de données : quelques approches pour le filtrage flexible

Elisabeth Metais, Florence Sèdes

Appariement d'informations dans les entrepôts de données : quelques approches pour le filtrage flexible1.

Elisabeth Métais①, Florence Sèdes②

① CNAM, laboratoire CEDRIC,
292, rue Saint Martin, 75141 Paris cedex 3, France
metais@cnam.fr

② Institut de Recherche en Informatique de Toulouse, IRIT
118, Route de Narbonne, 31 062 Toulouse cedex 4, France
sedes@irit.fr

Résumé : Les données textuelles possèdent de profondes différences de nature par rapport aux données numériques. Une de leurs particularités est leur polymorphisme dénotationnel : il est impossible de trouver une forme canonique de représentation d'une donnée textuelle, le point de vue de l'utilisateur est toujours primordial dans sa façon de conceptualiser le réel. Cet article propose des techniques pour réconcilier à la fois le point de vue des diverses sources dont sont issues les données et celui de l'utilisateur effectuant la requête ou modélisant sa vue dans le système décisionnel. Après une section exposant le principe d'un entrepôt de données, les sections 3 et 4 s'adressent au problème de l'hétérogénéité linguistique due à une différence de précision dans la saisie des différentes données textuelles, au niveau de l'intégration des schémas puis des données. La dernière section traite de la prise en compte de la différence de structuration de ces données. Dans les deux cas, une flexibilité doit être introduite dans l'appariement, par réécriture ou consultation d'un dictionnaire ontologique. Le problème de la définition par l'utilisateur - ou

[1] Ce papier a pour origine la réflexion entamée à l'occasion de la présentation effectuée dans le cadre de l'école thématique "Le temps, l'espace et l'évolutif" du GDR I3 en septembre 2000 à Marseille, qui nous a amenées à aborder un panorama des approches relatives à l'évolution dans les entrepôts de données et de documents.

de l'adaptation à celui-ci - du degré de flexibilité souhaité est également évoqué.

1. INTRODUCTION

La constitution d'entrepôts de données est une réponse au problème de l'intégration d'une grande quantité de données variées, relatives à un certain domaine d'application, et stockées physiquement dans différentes sources de données. L'entrepôt de données regroupe, sous une forme exploitable par des traitements utiles pour l'aide à la décision, les informations extraites de ces sources et qui sont potentiellement pertinentes pour telle ou telle catégorie de décideurs du domaine. En dix ans d'existence, les entrepôts de données se sont imposés comme une solution rentable pour faire face aux besoins des entreprises en termes de capitalisation de connaissances et d'aide à la décision.

Fondé sur une réutilisation des données de production conservées au cours du temps - "legacy databases" - le problème de l'évolution anarchique de chacune des sources de données est un challenge constant tout au long de la chaîne décisionnelle. Après avoir exposé le principe d'un entrepôt de données en section 2, nous traitons, dans les sections 3 et 4, des problèmes d'évolution des dénominations utilisées pour un même concept du monde réel tant au niveau des structures (noms d'attributs) que des instances (valeurs d'attributs). Une flexibilité existe au niveau du vocabulaire utilisé par l'apport de méta-données inspirée des techniques de dictionnaires linguistiques. Le problème de l'évolution de structure, généralisé à l'interrogation et à la comparaison de documents, est discuté en section 5, à travers de nouvelles approches dans lesquelles la flexibilité est introduite dans le processus de comparaison. L'aspect "évolution" est ici traité du point de vue structurel, dans une dimension temporelle.

2. PRINCIPE D'UN ENTREPÔT DE DONNÉES

Un entrepôt est défini comme "une collection de données intégrées, orientées sujet, non volatiles, historisées, résumées et disponibles pour l'interrogation et l'analyse" [Inmon 96]. L'entrepôt de données stocke des données nécessaires à la prise de décision ; il est alimenté et mis à jour via

des extractions de données portant sur les bases de production qui sont considérées dans la chaîne décisionnelle comme les "sources de données".

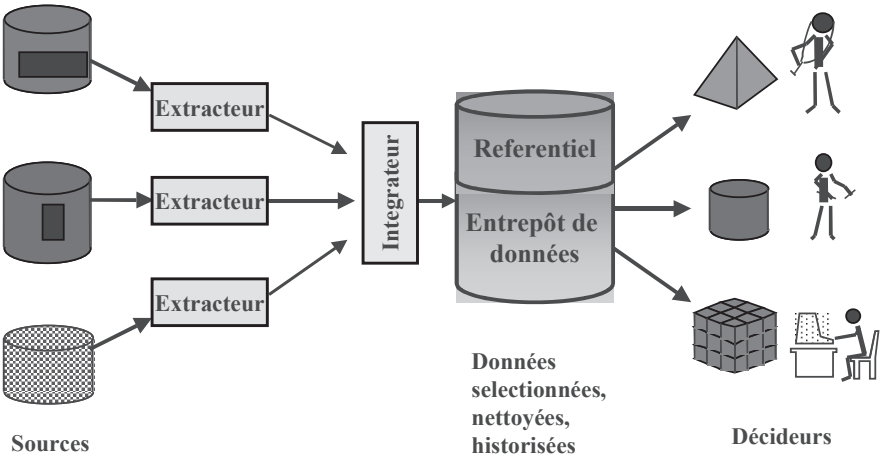

Fig. 1 - *Architecture générale de l'approche par entrepôt de données*

Les données de l'entrepôt proviennent de différentes sources éventuellement hétérogènes. L'intégration consiste à résoudre les problèmes d'hétérogénéité des modèles, des schémas, de la sémantique. La phase d'intégration de schémas dans un entrepôt de données est très particulière pour deux raisons : (i) le schéma résultant n'est pas un schéma conceptuel intégré définitif mais un schéma très évolutif, (ii) l'utilisateur d'un entrepôt de données n'a généralement pas une compréhension globale du schéma (les données proviennent de différentes sources parfois étrangères à l'entreprise) et il n'est pas forcément familier à l'interrogation d'une base de données (c'est avant tout un décideur).

Le schéma obtenu n'est pas quasi-figé comme celui d'une base de données classique mais peut être amené à subir des changements. Cet aspect dynamique est dû au fait que le contenu de l'entrepôt est plus guidé par les requêtes des utilisateurs que par l'étude de l'existant. Les changements consistent principalement en de nouveaux choix dans les relations (ou sous-relations) à extraire des sources, à l'ajout ou à la suppression de sources.

Deux objectifs doivent donc être intégrés dans l'élaboration d'un entrepôt de données : prévoir l'évolutivité du schéma et gérer la traçabilité des évolutions successives.

Un des éléments centraux du système est le "référentiel". Dans le contexte d'un entrepôt de données, "décrire une donnée" consiste principalement à indiquer comment l'obtenir à partir des sources. Les méta-données jouent un rôle important dans les mécanismes d'extraction, de rafraîchissement et d'intégration, et également dans la présentation d'une vision globale des données à l'utilisateur. L'intégration des données nécessite des assertions de correspondance entre les tables des sources et les vues de l'entrepôt.

Une partie importante du référentiel est dédiée aux méta-données nécessaires pour gérer l'hétérogénéité linguistique des sources. Un conflit de terminologie survient lorsqu'un même objet du réel est désigné par des noms différents ou au contraire lorsqu'un même nom est utilisé pour deux objets différents. Ces cas peuvent correspondre à des problèmes de synonymie ou d'homonymie, mais sont le plus souvent dus à une différence de niveau de généricité (ex : "personne" et "client") ou à des converses (ex : "vendre" et "acheter").

L'hétérogénéité de dénomination se situe à deux niveaux : au niveau des schémas c'est-à-dire des noms d'attributs (par exemple, "couleur" et "coloris") et au niveau des données (par exemple, "rouge" et "vermillon"). La façon dont sont saisies les données au cours de la vie d'un objet du monde réel évolue au cours du temps. Par exemple, à l'achat, une voiture a toujours pour coloris un nom spécifiquement créé par le constructeur (ex : "glacier"), puis, dès sa première revente, on ne lui attribue généralement qu'un nom commun de couleur (ex : "bleu ciel"), qui peut se simplifier en "bleu" au moment de sa déclaration à la préfecture. Il est indispensable dans des traitements décisionnels de distinguer cette évolution - ou perte de précision - d'un véritable changement ou d'une incohérence.

3. INTÉGRATION ÉVOLUTIVE DES SCHÉMAS

Peu traitée par les produits du marché, la comparaison de schémas - c'est-à-dire la production d'un ensemble d'assertions interschémas indiquant l'équivalence, l'inclusion ou la dissemblance de deux objets - a fait l'objet de nombreux travaux de recherche.

La seule aide actuellement fournie par les produits est une interface graphique permettant de comparer visuellement deux schémas et de les intégrer à l'aide de la souris, en cliquant sur les objets à rajouter ou à supprimer du modèle global, ou de fusionner deux objets ayant des propriétés différentes.

En recherche, après une première génération d'outils où les assertions devaient être données par l'utilisateur, est apparue dans les années 80 une génération d'approches "expertes" ([de Souza 86], [Bouzeghoub et al. 88], [Sheth et Gala 89]). Les structures d'objets sont comparées grâce à des mesures de similarité souvent complexes qui prennent en compte plusieurs facettes des objets (dénominations, contraintes, structure). Cependant cette deuxième génération traitait des schémas presque exclusivement de façon syntaxique. Dès le début des années 90, une troisième génération d'outils ([Frankhauser 91], [Métais et al. 93], [Johanneson 93], [Mirbel 95]) se caractérise par l'utilisation de techniques et d'outils de compréhension du langage naturel pour comprendre la sémantique des schémas à intégrer. Par exemple, la détection dans un dictionnaire linguistique de type WordNet [Fellbaum 99] d'un concept "personne", générique à la fois de l'objet "étudiant" d'un schéma S1 et de l'objet "professeur" d'un schéma S2, peut introduire la création d'un objet "personne" dans le schéma global, alors qu'il n'était présent ni dans S1 ni dans S2.

La tendance actuelle des travaux de recherche sur l'intégration de schéma est l'utilisation de logiques de descriptions permettant d'inclure les apports des dictionnaires linguistiques (hiérarchie de subsomptions de concepts, graphes canoniques des verbes) dans un raisonnement logique [Calvanese et al. 99], [Franconi 96], [Jarke et al. 99].

Le schéma d'un entrepôt de données étant potentiellement évolutif, l'intégration de nouvelles sources ou parties de sources, est également un souci des chercheurs. Nous avons montré dans [Comyn et Métais 97] la nécessité pour faciliter l'établissement de nouvelles assertions de maintenir deux dictionnaires linguistiques : un dictionnaire correspondant au "modèle du domaine" (c'est-à-dire le schéma global de l'entrepôt, les schémas des sources et leurs assertions), et un dictionnaire général, considéré comme une ontologie.

En effet, deux termes considérés comme synonymes au vu de leur utilisation dans leurs sources respectives, par exemple "client" dans une source et "personne" dans une autre (car ils ont les mêmes instances dans

ces bases de données), ne sont pas pour autant spécifiés comme synonymes (mais comme spécifique/générique) dans l'ontologie car dans l'utilisation habituelle du langage, ils n'ont pas la même définition, ni la même population (la population d"étudiant" est incluse dans la population de "personne"). L'intégration d'une nouvelle source de données (par exemple utilisant la dénomination "personne") nécessitera un positionnement par rapport à ces deux points de vue.

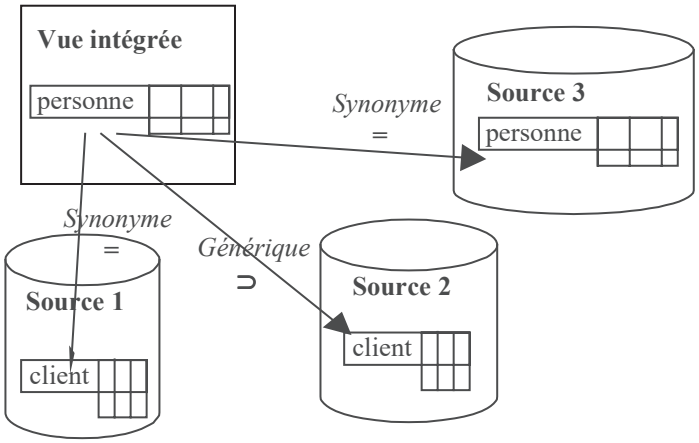

Fig. 2 - *Extrait du "dictionnaire linguistique du domaine"*
(assertions d'équivalences entre sources et entrepôt)

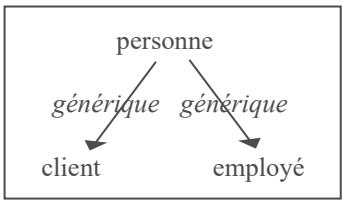

Fig. 3 - *Extrait du "dictionnaire linguistique général" (ontologie)*

Ce mécanisme de double dictionnaire mis en place dans les méta-données de l'entrepôt lui permet de fonctionner avec des terminologies différentes et d'intégrer au cours du temps de nouvelles sources.

Après l'étape de comparaison de schémas qui aboutit à un ensemble d'assertions inter schémas, deux utilisations distinctes peuvent en être faites pour la mise en place d'un entrepôt de données : (1) l'élaboration d'un schéma global de l'entrepôt, appelé modèle d'entreprise, (2) la transposition des assertions au niveau logique afin de permettre la distribution et la réécriture des requêtes sur les sources.

Une fois le problème des assertions inter-sources résolu au niveau des schémas (niveau intentionnel), par exemple une assertion d'équivalence entre la relation "client" dans une source et la relation "personne" dans une autre source, il reste à résoudre le problème des assertions inter-données (niveau extentionnel) ; par exemple, les tuples <213, Dupont, 25, Paris> et <213,Dupont, 25, Paris/France> correspondent-ils à la même personne ? sont-ils redondants ? sont-ils incohérents ? La sous-section suivante traite de ce problème.

4. INTÉGRATION ÉVOLUTIVE DES DONNÉES DE L'ENTREPÔT

Dans un système multisource tel une architecture d'entrepôts, une même donnée du réel peut être codée de différentes façons au cours de ses stockages dans les différentes bases de données ; l'exemple typique est celui de la fluctuation d'une même adresse postale. L'hétérogénéité des dénominations au niveau des données est traitée dans une phase dite de "nettoyage". Le nettoyage des données permet de mettre en conformité les données, par exemple en homogénéisant les codes utilisés (ex : le sexe d'une personne peut être codé "M/F" ou "1/2").

Nous allons illustrer cette section avec le problème des dénominations de couleur, principalement des couleurs de véhicules. Les couleurs sont essentielles dans le domaine du marketing (depuis plusieurs dizaines d'années, des milliers d'études ont montré le changement du consommateur face à un même produit dont on a simplement changé la couleur du "packaging") et sont exploitées par des algorithmes de fouille de données dans de nombreuses analyses de ventes. Une requête floue de type "Quel impact commercial aurait la fabrication de voitures rouges à Paris, au lieu de grises ?" est citée dans [Villacampa 2002] comme typique de ce que l'on attend actuellement d'un entrepôt de données. Or cette exploitation est particulièrement difficile en raison du nombre

énorme de dénominations, dont la plupart sont spécifiques à un constructeur.

4.1. Problématique de la réconciliation des données

Le "nettoyage des données", parfois appelé "épuration des données" ou "analyse de la qualité des données" - en anglais "data cleaning", "data cleansing" ou "data scrubbing" - a pour but de résoudre le problème de la consistance des données réconciliées dans l'entrepôt. Les inconsistances peuvent être locales à un enregistrement (ex : une erreur de frappe), locales à une source (ex : une même personne a deux adresses différentes), ou peuvent survenir lors de la mise en commun de deux sources (ex : une personne a une adresse différente dans chaque source).

Une centaine de types d'inconsistances a été répertoriée. Elles peuvent être dues : (1) à la présence de données fausses dès leur saisie, (2) à la persistance de données obsolètes, (3) à la confrontation de données exactes, sémantiquement identiques, mais syntaxiquement différentes.

Parmi les inconsistances apparaissant le plus fréquemment lors de la confrontation de plusieurs sources, on peut citer :

différents codages pour une même donnée, par exemple "M/F" ou "1/2" pour le sexe d'une personne,

différence d'unités, par exemple un prix en FF dans une source et en € dans une autre,

différence de granularité, par exemple un nombre d'heures travaillées par semaine dans une source et par mois dans une autre,

différence de plages de valeurs, par exemple les tranches d'âge {[11-20], [21-30], [31-40]} et {[15-30], [31-50]},

différence de fraîcheur, par exemple un âge de 25 ans dans une source et de 26 ans dans une autre car la deuxième source a été mise à jour plus récemment,

imprécision, par exemple un poids de 54 kg dans une source et de 54,2 dans une autre,

utilisation de synonymes, par exemple "sans emploi" et "chômeur",

différentes façons d'écrire la même donnée dans un texte libre, par exemple une même adresse peut être "4, av. du Gal. De Gaulle" dans une source et "4, avenue du général de Gaulle" dans une autre,

différence de contenu dans un texte libre, par exemple une adresse contenant dans une source le nom du destinataire ("Père Noël, cercle polaire, Raviemeni, Finlande") et pas dans l'autre ("cercle polaire, Raviemeni, Finlande"),

différence linguistique de niveau de perception dans les textes libres, par exemple pour la couleur d'un même objet : "vermillon" dans une source et "rouge" dans une autre source.

D'après [Jarke et al. 99], les différentes fonctionnalités d'un outil de nettoyage sont les suivantes :

des fonctions de normalisation et de conversion qui rendent standards des codages hétérogènes, par exemple le sexe sera toujours codé "M/F",

des nettoyages particuliers à certains champs, grâce à des tables de conversion (ex : les adresses aux Etats-Unis), de reconnaissance des sous-champs et des dictionnaires de synonymes et d'abréviations (ex : "av" et "avenue"),

des algorithmes de nettoyage indépendants du domaine, qui appliquent des techniques de "matching" pour établir l'équivalence de deux champs (ex : deux champs "matchent" si l'un est sous-chaîne de l'autre [Monge et Elkan 96]),

des règles de nettoyage qui établissent la correspondance de deux enregistrements par une combinaison de "matchings" ou d'égalités de leurs champs (ex : si deux enregistrements ont leurs clés identiques et leurs autres attributs proches par l'application d'une distance, alors ils correspondent).

Les travaux de recherche qui traitent du nettoyage des données concernent principalement les techniques de "matching" entre deux enregistrements. Un point souvent abordé dans le "matching" de tuples est la différence de traitement entre les attributs clés, les attributs non-clés mais sémantiquement discriminants et les attributs moins significatifs ([De

Michiel 89], [Agarwal et al. 95]). Cette échelle conduit à un "matching" strict (égalité de tous les champs) ou flou (égalité des champs clés, similarités entre champs non clés).

Une autre famille de travaux [Calvanese et al. 99] vise à introduire la fonction de "matching" des valeurs inter-sources, de façon formelle et unifiée, dans un algorithme d'intégration de données programmé en logique terminologique.

4.2. Principe général de l'approche proposée pour la réconciliation de données

L'utilisation des opérateurs classiques de l'algèbre relationnelle pour interroger un système multisource peut conduire à des résultats dénués de sens. L'exemple de la figure 4 illustre le problème que nous visons à résoudre. Nous pouvons voir sur la figure 4(a) que l'union classique de deux relations issues de sources différentes peut conduire à un résultat difficile à interpréter et drainer des inconsistances. Nous proposons de traiter ce problème par l'introduction d'opérateurs sémantiques (union sémantique, jointure sémantique, etc.) qui prennent en compte la sémantique des valeurs d'attributs dans leur fonction de "matching" [Kedad et Métais 99]. On peut voir en Figure 4(b) le résultat visé pour ce même exemple.

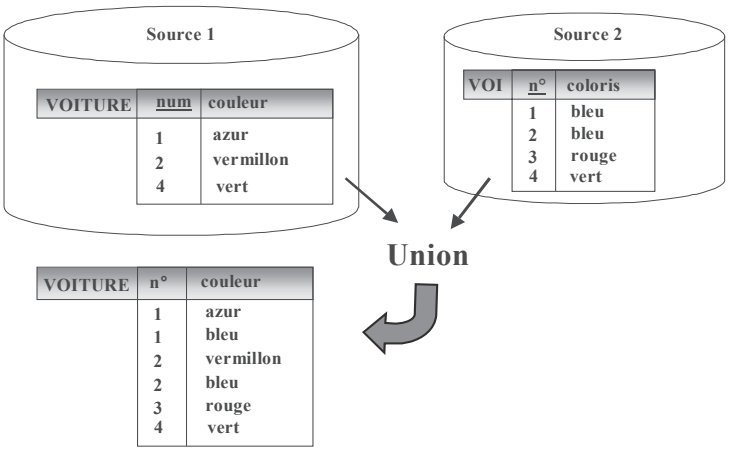

Fig. 4 (a) – *Union de deux sources sans connaissance linguistique*

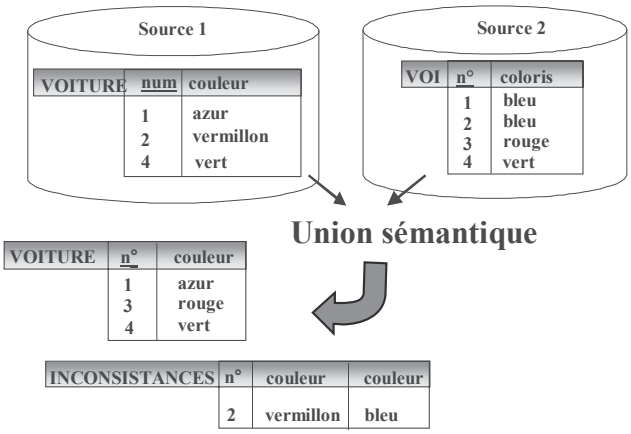

Fig. 4 (b) – *Union de deux sources utilisant des connaissances linguistiques*

Remarque : Dans la figure 4(b) et dans le reste de cette section nous utilisons le terme "inconsistances" pour le cas de deux tuples qui coïncident sur leurs attributs clés mais qui possèdent des valeurs conflictuelles sur un attribut non clé.

Le problème de base est le suivant : étant donnés deux schémas de sources R1(C, A1, A2, A3) et R2(C', A4, A5, A6), une fois que les correspondances entre les noms d'attributs ont été établies au niveau du schéma (par exemple, A1 est équivalent à A4), il est nécessaire d'avoir une connaissance au niveau des instances (la valeur a1 du tuple r1 de R1 est équivalente à la valeur a4 du tuple r2 de R2, en dépit de leur hétérogénéité de codage). Par exemple, les assertions d'équivalence "*bleu* est équivalent à *azur*" et "*bleu* est dissemblable de *rouge*" sont nécessaires à l'union sémantique effectuée figure 4(b).

Au vu du nombre d'instances pouvant se chiffrer en milliers et provenir de dizaines de sources, il n'est pas réaliste de stocker ces assertions comme on peut le faire pour celles des noms d'objets au niveau du schéma.

De plus, nous pensons que ces assertions sont dépendantes de l'utilisateur. Par exemple pour la requête "rechercher les voitures ayant la

même couleur que leurs sièges", certains utilisateurs attendent dans le résultat les voitures bleues à sièges bleus quelle que soit la sorte de bleu, alors que d'autres établissent clairement une distinction entre "bleu clair" et "bleu foncé".

Un nettoyage préalable des données par l'outil d'extraction est également inadéquat car ces dénominations ne sont pas strictement équivalentes comme le seraient par exemple des codages "1" et "M" pour le sexe masculin ou des prix en FF dans une source et en € dans une autre. Le nettoyage de "azur" en "bleu" dans la première relation conduirait à une perte de sémantique. Le nettoyage dans la deuxième relation de "bleu" en "azur" est impossible avant d'avoir la connaissance par l'autre source qu'il s'agit d'un bleu azur et non par exemple d'un bleu marine.

En conséquence, notre démarche est basée sur l'utilisation de connaissances linguistiques stockées dans le référentiel et utilisées pour effectuer un "matching" linguistique au moment de l'exécution des requêtes de construction des vues utilisateur.

4.3 Utilisation de connaissance linguistique pour la réconciliation des données

Le "matching" linguistique proposé consiste à introduire une flexibilité dans la comparaison des valeurs. Il est basé sur l'utilisation d'une ontologie, considérée dans la chaîne décisionnelle comme un ensemble de méta-données. Cette ontologie peut soit être extraite soit d'un outil comme le dictionnaire linguistique WordNet [Fellbaum 99] qui est disponible depuis plusieurs années et actuellement largement utilisé, soit faire partie de la trame d'ontologies qui se mettent actuellement en place dans le cadre du web sémantique [Ding et al. 2002].

L'ontologie

La structure principale d'une ontologie est la hiérarchie de concepts, organisée sous la forme d'une hiérarchie d'arcs "est-un" (i.e. "est subsumé par"). On peut voir sur la figure 5 un extrait de la hiérarchie du domaine des dénominations de couleurs. La principale particularité d'une hiérarchie dévolue à des instances est de mêler dans une même hiérarchie des dénominations du vocabulaire courant (par exemple, "rouge", "bleu") et des dénominations propres aux constructeurs (par exemple, "Séville", "Glacier") qui peuvent parfois être des codes.

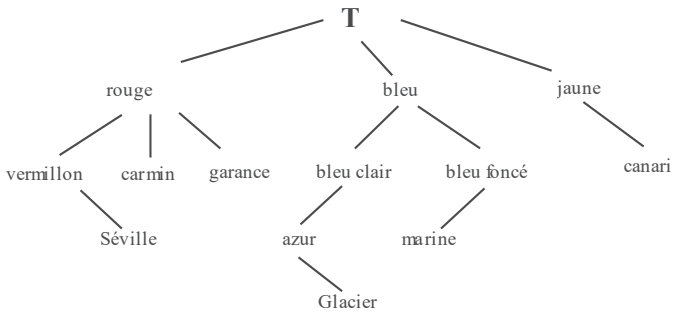

Fig. 5 - *Extrait de la hiérarchie de concepts des dénominations de valeurs*

Chaque nœud correspond à un concept et peut donc regrouper un ensemble de synonymes. En particulier, dans notre approche, cette hiérarchie peut être multilingue ; dans ce cas, le nœud peut regrouper des termes de différents langages correspondant au même concept, par exemple {rouge, red, rojo}. Le problème dû à des informations stockées dans des langues différentes est ainsi traité par la même algorithmique que les synonymes. Cette solution résout également le problème des termes n'ayant pas d'équivalents exacts dans une autre langue, par exemple le terme "glas" qui en breton désigne à la fois le "bleu" et le "vert" (le nœud "glas" est générique immédiat de "bleu" et "vert"). Elle nous paraît donc préférable à l'approche traditionnelle consistant à stocker une hiérarchie par langue puis à établir des lien inter-hiérarchies.

Établissement du niveau de précision pour l'appariement linguistique

Un problème inhérent à l'introduction de flexibilité est de définir sa portée. En d'autres termes, le problème soulevé lors de l'utilisation d'une telle hiérarchie pour l'interrogation est celui de la "distance sémantique" à l'intérieur de laquelle on considérera deux valeurs comme identiques. Par exemple, "bleu ciel" et "marine" doivent-elles être considérées comme des dénominations correspondant à une même couleur ? De nombreux travaux [Resnik 95], [Song 96] ont proposé des distances sémantiques pour des besoins d'intégration ou de recherche d'information. Ces solutions calculent la distance en fonction du nombre et de la nature des arcs qui les séparent ; l'utilisateur (ou le système par défaut) doit fixer une valeur de distance à partir de laquelle le "matching" devra échouer. Cependant, bien

que ces distances puissent tout à fait être appliquées sur un dictionnaire linguistique, nous ne voulons pas demander une telle chose à l'utilisateur car cela ne correspond pas à une manière naturelle de raisonner. Dans notre approche, l'utilisateur va automatiquement définir des classes de valeurs équivalentes par la sélection d'un "niveau de précision".

Lorsque les deux valeurs à comparer sont subsumées l'une par l'autre (relation fils-ancêtre) par exemple "bleu" et "marine", elles correspondent à la même couleur mais donnée avec un niveau de précision différent dans les deux sources. Nous considérons donc qu'il n'y a pas de conflit et qu'il doit y avoir "matching".

Par contre lorsque les deux valeurs correspondent à des nœuds "frères" ou "cousins", par exemple "bleu clair" et "bleu foncé", il y a une différence sémantique et il est impossible de dire *a priori* si le "matching" doit réussir ou échouer, il faut introduire le degré de flexibilité voulu, c'est-à-dire la distance jusqu'à laquelle on va considérer deux valeurs comme similaires.

Nous répondons à ce problème par une interface permettant à l'utilisateur de préciser dans cette hiérarchie un ensemble de "nœuds de précision" correspondant à un "niveau de précision". L'algorithme de "matching" considérera alors comme équivalentes toutes valeurs appartenant au sous-arbre d'un même "nœud de précision". Le choix d'un nœud de précision consiste à construire une classe d'équivalence à l'intérieur de laquelle toutes les valeurs vont matcher. Sur la figure 6, les nœuds de précision choisis sont entourés (il s'agit des nœuds "rouge", "bleu" et "jaune") et les "classes d'équivalences" obtenues sont grisées. Sur cet exemple "bleu" étant considéré comme un nœud du niveau de précision, "bleu ciel" et "marine" seront considérées comme des valeurs pouvant matcher. Sur ce même exemple, "bleu clair" et jaune" ne matchent pas. Cette technique pour déterminer la similarité est beaucoup plus conviviale que l'approche traditionnelle consistant à fixer une distance.

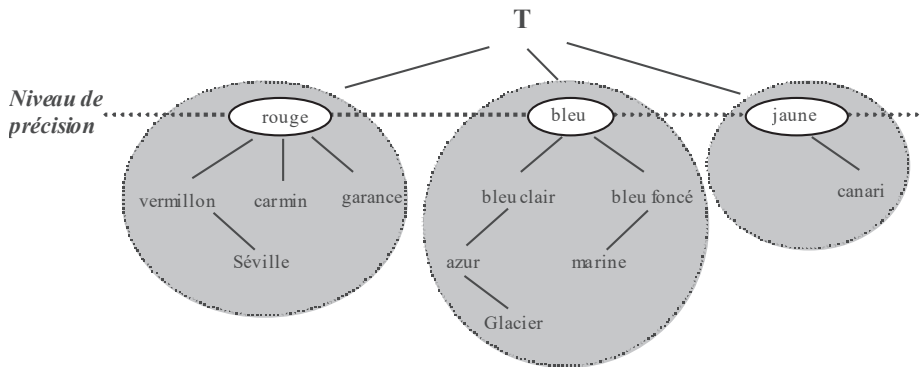

Fig. 6 – *Choix par l'utilisateur des nœuds "rouge", "bleu" et "jaune" comme niveau de précision dans la flexibilité du matching*

Remarque 1 : Les nœuds choisis comme des "nœuds de précision" ne doivent pas obligatoirement être au même niveau. On pourrait par exemple choisir "rouge", "bleu clair", "bleu foncé", et "jaune".

Remarque 2 : Le mécanisme n'est pas affecté par l'existence d'héritages multiples.

Ce "matching" sémantique est introduit dans les opérateurs de l'algèbre relationnelle. Dans l'opérateur d'union, il permet de dédoubler des tuples correspondant au même objet malgré une différence syntaxique, voire linguistique, par exemple une même voiture dont la couleur est stockée avec une précision différente. Dans l'opérateur de jointure, il est par exemple extrêmement utile pour répondre à la requête "quels sont les clients ayant acheté une voiture avec des sièges de la même couleur que la voiture". Il est en effet peu probable que les valeurs entrées dans la base soient exactement les mêmes. Dans l'opérateur SQL "group by", il peut permettre de voir par exemple si les voitures rouges ont plus d'accidents que les autres voitures. Dans l'opérateur "order by", il peut permettre d'ordonner les voitures par grands groupes de couleurs.

Des applications de cette approche ont été réalisées dans le domaine médical, d'une part pour la réconciliation des noms de médicaments, et d'autre part des noms de pathologies qui se prêtent bien à une structuration en hiérarchie de concepts [Réanimatic 2000]. Le nouveau problème rencontré dans ces applications est leur évolution régulière au

cours du temps en raison du manque de consensus et de l'évolution des connaissances.

5. INTÉGRATION DE STRUCTURES

La résolution des hétérogénéités sémantiques, abordée dans la précédente section, lors de l'intégration et de l'interrogation de données textuelles, est une problématique sous-jacente de manière générale aux bases de données [Cholvy et Moral 01], et, avec la généralisation des langages de balisage et standards d'échange tels que XML, à la recherche documentaire en général. En effet, on retrouve les problèmatiques évoquées ci-dessus à propos des attributs et valeurs d'attributs hétérogènes dans des systèmes multisources, pour l'exploitation des documents du web ; citons, par exemple, la disparité des dénominations des balises, des identifications des attributs, l'hétérogénéité des valeurs et contenus, des niveaux de granularité, l'ambiguïté liée à la liberté syntaxique du choix balises/attributs, etc.

A ceci s'ajoute la question de la résolution des hétérogénéités structurelles. En effet, les structures de données manipulées ne sont plus des structures "plates" (i.e. tables relationnelles) mais des modèles gérant des objets complexes, hétérogènes, éventuellement multimédia, en général préexistants à toute norme ou standard ("legacy databases").

L'objectif restant de pouvoir répondre aux requêtes de l'utilisateur sans imposer à celui-ci de pré-requis, et en préservant une qualité de service raisonnable, il paraît indispensable qu'une requête puisse faire référence à un document en terme d'apparence (du point de vue de l'utilisateur ignorant du détail de la structure sous-jacente, cf. section 3), et non, comme dans les approches "classiques" en bases de données, par rapport aux valeurs exactes des attributs (cf. 4.2). Interroger ces données revient dès lors à mettre en correspondance deux structures (ou, du moins, les éléments qui peuvent en être extraits), celle de la requête et celle du document-cible.

Cette préoccupation peut être abordée d'un point de vue plus générique. Ainsi, un des problèmes liés au web est que, outre la possibilité d'accès à de larges collections de données hétérogènes et semi-structurées, il permet à l'information d'être modifiée de manière totalement aléatoire. Ces changements rapides et *a priori* imprédictibles

posent donc le problème de leur détection : un utilisateur visitant des documents de manière répétée peut vouloir être informé de la manière dont le document a été modifié depuis sa dernière visite. Il s'agit donc de pouvoir répondre à une requête consistant à détecter les changements dus à ces évolutions, en comparant ancienne et nouvelles versions du document, et en évaluant combien elles sont similaires.

Compte tenu des hétérogénéités structurelles inhérentes à la nature même des ces données, la comparaison de schémas (éventuellement partiels) élicités à partir de l'extraction de motifs et de traits réguliers, ne peut reposer sur une comparaison stricte, afin d'éviter de restituer un ensemble de réponse vide, mais doit intégrer une certaine flexibilité dans l'exploitation des informations relatives à la structure, via des procédures d'appariement approximatif et de prise en compte des préférences de l'utilisateur.

5.1 Hétérogénéités structurelles et détection de changements

Les travaux sur les données auto-descriptives sont apparus assez récemment dans la communauté Bases de Données. La modélisation de ces données, appelées semi-structurées, est différente des approches adoptées dans les modèles relationnels ou orientés objets. Les données semi-structurées [Abiteboul et al. 97], [Buneman et al. 96], [Fernandez et al. 97] sont caractérisées par une structure semi-explicitée, irrégulière, adaptative ; ce concept recouvre, par exemple, les documents XML "bien formés" (c'est-à-dire syntaxiquement corrects), par opposition aux documents "valides" (c'est-à-dire conformes à une Document Type Definition (DTD) qui régit la structure générique des instances d'une classe de documents, comme le schéma pour une base de données). Ainsi, la plupart des documents du Web, balisés en HTML, XHTML ou XML, qui constituent une classe de documents de plus en plus importante [Chrisment et Sèdes 98], sont considérés comme des prototypes de données semi-structurées. De telles données, irrégulières, incomplètes, ne se conforment donc pas à un schéma prédéfini. Elles peuvent contenir des éléments de structure mais, si ceux-ci existent, ils sont encapsulés dans les instances, et *a priori* inconnus.

La généralisation de leur usage répond aux problèmes de l'intégration de sources de données hétérogènes à travers les systèmes de médiation, de

la modélisation de sources réparties, d'entrepôts et/ou de "dataweb", associant données issues du web, textes structurés ou non, etc.

La prise en compte de la comparaison de structures documentaires est une problématique ancienne. La plupart des travaux antérieurs portent sur la détection des changements, se limitant à des fichiers texte [Myers 86], [Kifer 95], des fichiers "plats" ou des données relationnelles [Abiteboul et al. 95], [Labio et Garcia-Molina 95], en présentant des algorithmes de comparaison d'ensembles d'enregistrements identifiés par des clés. Le système *LaDiff,* par exemple, a été implémenté pour détecter les balises et visualiser les changements dans des documents structurés. Il est basé sur l'analyse de leur structure hiérarchique, prenant deux versions d'un document Latex comme entrée et produisant en sortie un document Latex contenant leurs différences.

Beaucoup de travaux dans le domaine des BD ont été conduits autour de documents structurés, par exemple via SGML [Chawathe et Garcia-Molina 97], [Chawathe et al. 97], [Christophides et al. 94], [Milo et Zohar 98]. Les documents envisagés dans ces contextes sont "valides", donc caractérisés par une DTD à laquelle tout instance doit se conformer ; ceci facilite l'évaluation, puisque, cette DTD constituant l'équivalent du schéma de la base -une structure générique pour la classe de documents-, on se ramène à la problématique d'évolution de schéma évoquée dans la section 3.

La problématique se complexifie avec l'hétérogénéité qui caractérise les données semi-structurées. Peu de systèmes documentaires ont été conçus pour les supporter. L'actualité de cette problématique s'illustre pourtant dans des systèmes tels que Xylème [Marian et al. 00], [Cobéna et al. 01], ou des observations telles que [Broder et al. 00]. Leur objectif est, à partir de l'identification de régularités, de détecter les changements dans le seul but d'assurer la gestion de différences entre structures arborescentes.

5.2 Appariement de structures

Le problème de la comparaison de structures tel qu'il est envisagé dans ces approches s'avère insuffisant compte tenu de la nature des données manipulées et des requêtes à prendre en compte.

La manipulation des documents repose sur la découverte d'éléments de structure "élicités" à partir de processus d'analyse de leur contenu, à l'issue desquels un document pourra être représenté - de manière classique - par un arbre étiqueté, éventuellement ordonné, traduisant la structure hiérarchique dans laquelle les descendants de chaque nœud ont une position désignée. L'une des facettes de la résolution du problème consiste à trouver une mesure de comparaison appropriée pour les arbres à comparer, celle-ci se basant sur les statuts et étiquettes des différents items de marquage, et pas uniquement sur les contenus (cf. 4.1).

En effet, deux documents peuvent être comparés par rapport à leur structure en utilisant les techniques de comparaison d'arbres [Shasha et Zhang 90], [Wang et al. 94], [Zhang et Shasha 89]. Le problème des relations de "sous-arbres" entre deux arbres ne peut être géré correctement sans sauvegarder la relation d'ordre entre les éléments [Kilpelaïnen 92]. Il s'avère que la structure de graphe n'est pas adaptée à des comparaisons autres qu'une égalité exacte. Or, il ne s'agit pas seulement, dans le contexte qui nous intéresse ici, de retrouver des arbres identiques via des techniques de "graph matching" : le résultat du processus d'appariement doit permettre de retrouver des arbres "approximativement égaux", selon un degré de similarité.

Des processus semblant plus adaptés sont ceux basés sur des algorithmes de comparaison, par adaptation successive, par exemple en simulant la suppression d'un niveau de structure si celui-ci est jugé peu pertinent, ou en intégrant un degré qui reflète l'importance de la balise/attribut, de son statut, ou de la valeur du nœud, selon son niveau, celui de ses descendants, etc. Identifier les changements au niveau des relations des nœuds peut consister, par exemple, à détecter que le "delta" entre deux documents se résume au déplacement d'un nœud (et de ses descendants) dans l'arborescence, ou qu'une feuille a été décomposée en nouveaux nœuds, devenant feuilles à leur tour (par ex. pour un niveau *l*, la structure peut avoir été réorganisée en supprimant ce niveau, vide de contenu, et rattachant directement les descendants de niveau *l+1*, à l'ascendant de niveau *l-1*).

5.3 Recherche de similarités

A partir de différents constats illustrés ci-après, corroborés par l'analyse de corpus de requêtes, se pose donc le problème de la recherche des documents "similaires" en terme de structure – et pas uniquement en

terme de contenu comme dans les approches de type "Recherche d'Information" -, à partir d'une requête, d'une structure-exemple (ou d'une description pouvant être considérée comme telle) ou d'une structure pré-existante. La cohabitation possible de vues multiples d'un même document, inhérente à la nature même et à l'hétérogénéité des données, handicape en effet toute approche basée sur une comparaison stricte.

En se replaçant du point de vue de l'utilisateur, on recense différents types de requêtes adressés à une collection de documents semi-structurés.

Un premier, développé dans [Dubois et al. 2001], est du type : "Retrouver le(s) papier(s) incluant impérativement le mot-clé 'x...', et de préférence les mots-clés 'y…' et 'z…', qui référence la plupart des *bons* papiers du domaine". La recherche des papiers pertinents pour cette requête requiert (i) l'identification des mot-clés et des références dans la structure du document, le document lui-même ou toute description (annotation) associée à celui-ci, (ii) la vérification que la plupart des auteurs appartiennent à un ensemble d'auteurs du domaine associé aux mots-clés présents, (iii) la "qualité" des papiers, évaluée via des éléments d'évaluation tels que l'indice de citation, la sélectivité du congrès, etc.

Un autre type de requête déjà évoqué plus haut est celui qui fait référence à la description d'une structure-exemple de document. Par exemple, retrouver un document "déjà vu" à partir d'une structure-exemple constituée de "un titre, un (des) auteur(s), probablement un résumé, peut-être des mots-clés, un corps structuré en sections et sous-sections, certainement suivi de références, parmi lesquelles devraient figurer de préférence les noms des auteurs", exprimant par là même les préférences de l'utilisateur.

L'évaluation de réponses à ces requêtes repose sur la comparaison de deux représentations, celle de la requête et celle du document-cible, intégrant les hétérogénéités potentielles (format, structure, sémantique, etc.). Dans d'autres contextes, il s'agira de pouvoir "mesurer" les changements dus à des évolutions d'un document, en comparant ancienne et nouvelles versions, et en évaluant combien elles sont similaires.

5.4 Degré d'appariement

En l'absence de structure standard des documents à retrouver, des extensions basées sur la flexibilité [Conan et Rocacher 97], [Dubois et al.

97], [Bosc et Prade 98] constituent une alternative prometteuse. Les apports de l'utilisation des ensembles flous aux bases de données en général, discutés par exemple dans [Bosc et Prade 99], et aux systèmes d'information multi-sources plus précisément, sont particulièrement pertinents dans un contexte où les données et leur structure sont a priori inconnues et les requêtes de l'utilisateur basées non plus sur des prédicats booléens, mais plutôt sur des préférences, dans le but d'obtenir un ensemble de réponses non vide.

Le degré d'appariement entre deux items est une notion qui intègre différents degrés dans la mesure de leur ressemblance, basés sur la structure, le contenu et les préférences incluses dans la requête. Le degré de similarité recouvre la distance entre items (conditionnée par un seuil au delà duquel celle-ci n'a plus de sens). Il est éventuellement associé à un degré de synonymie (au sens de celui adopté dans les approches de type "RI"), évalué à partir de thésaurus ou d'ontologies [Loiseau et Prade 02] comme détaillé en section 4.3. Le degré de préférence intervient ensuite pour pondérer les valeurs préférées pour la recherche.

Ce degré doit traduire la proximité des deux items en terme d'appariement. La comparaison devenant une question de degré, par cet appariement, nous entendons le rapprochement de graphes qui correspondent au moins pour leurs parties essentielles, et si possible, diffèrent par leur moins importante sous-partie.

Des travaux allant dans le sens de ces propositions sont récemment apparus [Bordogna et Pasi 02], [Braga et al. 02], [Ciaccia et Penzo 02], basés sur une connaissance sémantique des éléments (nœuds) ou associations (arcs) pour mettre en œuvre une comparaison approximative, par exemple par des requêtes flexibles sur des données XML. Cette connaissance peut être extraite du nom de l'élément (i.e. de la balise), des attributs ou de leur valeur, ou de toute spécification externe de l'utilisateur, puis liée à une comparaison approximative de graphes.

Nous avons proposé une contribution au processus d'appariement, basée sur ce principe. Après élicitation des éléments de structure - implémentée via un outil de réécriture [Lambolez et al. 95]- à partir de la reconnaissance des items dans le contenu du document lui-même, par identification des balises et réécriture automatique au moyen de dictionnaires [Sèdes 98], l'arborescence (ordonnée) est construite. Le processus de réécriture permet d'identifier, dans l'arbre, des arcs qui peuvent être considérés comme plus importants que les autres : à partir de

l'analyse du document, un balisage syntaxique et sémantique peut permettre d'inférer que certains arcs ont un poids moins important que d'autres. La base du calcul du poids repose sur le niveau d'importance de l'arc, qui dépend de la balise reconnue, de son contenu, du texte associé au nœud correspondant, et de la profondeur de l'arbre. La comparaison ne peut pas envisager de similarités structurelles en ignorant le texte associé – ou non : nœuds "vides" - aux nœuds. Le processus de pondération traite donc en premier lieu la comparaison structurelle et la sémantique de balisage, puis le contenu (cf. section 4). Deux structures de document sont d'autant plus similaires qu'il existe moins de sous-parties à négliger pour les rendre identiques. La comparaison d'une structure avec une de ses approximations dans laquelle certains arcs et/ou nœuds ont été supprimés, permet de détecter si le contenu associé au nœud supprimé – à supposer que celui-ci existe - a été "collé" ailleurs dans l'approximation.

La comparaison entre représentations, en impliquant des mécanismes de découverte structurelle, s'intègre naturellement dans les approches de "datawarehousing", pour la traçabilité, l'analyse d'impact, la gestion de versions ou la recherche de sous-structures communes. De telles approches permettent de comparer et d'ordonner des documents, ou encore des documents et des requêtes, a priori hétérogènes mais approximativement similaires du point de vue de la structure (requête, document-exemple, évolution de documents). Le rôle de ce processus de comparaison de l'évolution des structures est complémentaire de ceux présentés en sections 3 et 4 dans la gestion d'un entrepôt dédié au stockage de gros volumes de données, via la comparaison de documents semi-structurés, rendue difficile par l'irrégularité, l'incomplétude et le manque de schéma qui caractérisent ces données. Il peut être envisagé dans différents contextes comme la détection d'inconsistances structurelles (modification de structure involontaire ou anormale, conflits, sécurité) ou de ressemblances (veille technologique, plagiat, duplication), le versionnement de documents XML ou l'interrogation des modifications.

6. CONCLUSION

Les données textuelles possèdent de profondes différences de nature par rapport aux données numériques. Une de leurs particularités est leur polymorphisme dénotationnel : il est impossible de trouver une forme canonique de représentation d'une donnée textuelle, le point de vue de l'utilisateur est toujours primordial dans sa façon de conceptualiser le réel.

Leur deuxième caractéristique est l'érosion au cours du temps. Dans tous les domaines techniques, la précision du vocabulaire s'accroît au fur et à mesure des progrès scientifiques. Dans le domaine décisionnel, les requêtes d'interrogation amènent à effectuer des comparaisons sur des données structurées ou semi-structurées hétérogènes, ayant des structures différentes et des degrés de précision différents.

Nous avons abordé dans ce papier quelques techniques pour répondre à ce problème, principalement basées sur l'introduction d'une flexibilité dans la comparaison de structures et de valeurs. Cette flexibilité est réalisée par l'injection de graphes de méta-données et d'algorithmes de réécriture de graphes. Cet appariement a pour but principal de prendre en compte le point de vue de celui qui a codé les données, au niveau de leur structuration et des termes choisis. Au-delà du cadre applicatif évoqué dans ce papier, la généralisation de ces approches peut conduire à la proposition d'outils automatisables d'audit de sites ou de prédiction d'évolutions. Pour répondre plus complètement au problème, une extension de ces travaux à la prise en compte du temps devrait en effet être introduite.

Remerciements : Nous souhaitons vivement remercier Mokrane Bouzeghoub, Isabelle Comyn-Wattiau, Zoubida Kedad et Marie-Christine Rousset d'une part, Claude Chrisment et Henri Prade d'autre part, pour les nombreuses discussions sur ce sujet et le travail réalisé ensemble.

7. REFERENCES

[Abiteboul et al. 97] Abiteboul S., Quass D., McHugh J., Widom J., Wiener J., The Lorel Query Language for Semistructured Data, Intl Journal on Digital Libraries, vol. 1, no.1, 68-88, 4/1997.

[Abiteboul 97] Abiteboul S., Semi-structured information, Proc. of ICDT'97, International Conference on Database Theory, Invited talk, 1-20.

[Abiteboul et al. 95] Abiteboul S., Cluet S., and Milo T., A database interface for file updates. In Proceedings of the ACM SIGMOD Intl Conference on Management of Data, 1995, 18-26.

[Agarwal 95] Agarwal S., Keller A. M., Wiederhold G., Saraswat K., Flexible relation: an approach for Integrating data From Multiple, Possibly Inconsistent Databases, IEE Int. Conf. On Data Engineering, Taipei (Taiwan), mars 1995.

[Bordogna et Pasi 02] G. Bordogna, G. Pasi, Flexible Querying of Web Documents, ACM SAC 2002, Madrid (Spain), March 2002, pp. 675-680.

[Bosc et al. 99] Bosc P., Buckles B.B., Petry F.E., Pivert O., Fuzzy Databases, in "Fuzzy Sets in Approximate Reasoning and Information Systems", J. Besdeck, D. Dubois and H. Parde eds., Kluwer Ac. Pub., 1999, pp. 403-468.

[Bosc et Prade 98] Bosc P. Prade H., An introduction to the fuzzy set and possibility theory-based treatment of flexible queries and uncertain or imprecises databases, pp. 285-324, chap. 10 in "Uncertainty Management in Information Systems – From needs to solution", A. Motro, Ph. Smets eds., Kluwer Ac. Pub.

[Bouzeghoub et al. 88] Bouzeghoub M., Comyn-Wattiau I., Viallet F., L'intégration de vues par unification sémantique des structures, BDA'88.

[Braga et al. 02] D. Braga, A. Campi, E. Damiani, P.L. Lanzi, G. Pasi, FXPath: Flexible Querying of XML Documents, Proc. of EuroFuse 2002, Workshop on Information Systems, Varenna (Italy), Sept. 2002, B. De Baets, J. Fodor, G. Pasi eds., pp. 115-120.

[Broder et al. 00] Broder A., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., J Wiener J., Graph structure in the web, WWW 9th Conf. Proc., http://www9.org/w9cdrom/160.

[Buneman et al. 96] Buneman P., Davidson S., Hillebrand G., Suciu D., A query language and optimization techniques for unstructured data, Proceedings of ACM-SIGMOD International Conference on Management of Data, 1996.

[Buneman 97] Buneman P., Tutorial: Semistructured data, Proceedings of the ACM SIGMOD Symposium on Principles of Database Systems, 1997.

[Calvanese et al. 99] Calvanese D., de Giacomo G., Lenzerini M., Nardi D., Rosati R., A principle Approach to Data Integration and reconciliation in Data Warehousing, DMDW'99, Heidelberg, Allemagne, 1999.

[Chawathe et al. 97] Chawathe S., Garcia-Molina H., Meaningful Change Detection in Structured Data, Proceedings of the ACM SIGMOD Intl Conference on Management of Data, 1997, 22-32.

[Chawathe et al. 96] Chawathe S., Rajaraman A., Garcia-Molina H. and Widom J., Change detection in hierarchically structured information. Proceedings of the ACM SIGMOD Intl Conference on Management of Data, 1996, 34-43.

[Cholvy et Moral 01] Cholvy L., Moral S., Merging databases: Problems and Examples, Intl J; of Intelligent Systems, vol. 16, pp. 1193-1221, J. Willey & Sons, 2001.

[Chrisment et Sèdes 98] Chrisment C., Sèdes F (1998). Bases d'objets documentaires. Tutoriel, INFORSID 98, Montpellier (France), 1-40.

[Christophides et al. 94] Christophides V., Abiteboul S., Cluet S., Scholl M., From structured documents to novel query facilities. 1994 ACM SIGMOD Intl Conf. on Management of Data, 313-324, Minneapolis, May 1994.

[Ciaccia et Pezzo 02] P. Ciaccia, W. Pezzo, Adding Flexibility to Structure Similarity Queries on XML Data, FQAS'2002, T. Andreasen et al. Eds., LNAI 2522, pp. 124-139.

[Cobéna et al. 01] Cobéna G., Abiteboul S., Marian A., Detecting changes in XML Documents, BDA'2001, Agadir (Maroc), Cepadues ed.

[Comyn-Wattiau et Métais 97] Comyn-Wattiau I., Métais E. View Integration as a Way to Build a Semantic Dictionary for a Data Warehouse. NLDB'97, Vancouver (Canada), 1997.

[Connan et Rocacher 97] Connan F., Rocacher D. (1997) Gradual and flexible Allen relations for querying video data. Proc. 5th Europ. Cong. on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen, Germany, Sept. 8-11, 1132-1136.

[De Michiel 89] De Michiel L.G., Resolving Database incompatibility: an Approach to performing Relational Operations Overmismatched Domains, IEEE trans. Knowledge and Data Engineering (DKE) vol.1,n°4 (1989), pp. 485-493.

[Ding et al. 2002] Ding Y., Fensel D., Klein M. and Omelayenko B., The Semantic Web: Yet Another Hip?, Data And Knowledge Engineering Vol 41, N° 3, 2002, pp. 205-227.

[Dubois et al. 97] Dubois D., Nakata M., Prade H. (1997) Find the items which certainly have (most of) important characteristics to a sufficient degree. Proc. 7th IFSA World Cong., Prague, Academia publ., 243-248.

[Dubois et al. 99] Dubois D., Prade H., Sedes F., Fuzzy logic techniques in Multimedia Databases Querying- A preliminary investigation of the potentials, IFIP DS-8 "Semantics in multimedia", R. Meersman, Z. Tari, S. Stevens Eds, chap. 13, Kluwer, 1999.

[Dubois et al. 2001] Dubois D., Prade H., Sedes F., The use of some fuzzy logic techniques in Multimedia Databases Querying, IEEE TDKE Transaction on Data and Knowledge Engineering, Vol. 13 (3), May-June 2001, pp. 383-393.

[Franconi 96] Franconi E., Logical form and Knowledge representation : toward a reconciliation. In Working notes of the AAAI Fall Symposium on Knowledge Representation Systems based on Natural Language, Cambridge MA, 1996.

[Fellbaum 99] Fellbaum C., WordNet, An Electronic Lexical Database, ISBN 0-262-06197-X.

[Fernandez et al. 97] Fernandez M., Florescu D., Levy A., Suciu D., A Query Language for a Web-Site Management System, SIGMOD Record, vol. 26, no. 3, pp. 4-11, 9/1997.

[Frankhauser 91] Frankhauser P; Kracker M., Neuhold E.J., Semantic vs Structural Ressemblance of Classes, Sigmod Record, 20(4), octobre 1991.

[Inmon 99] Inmon W.H.. Building the Data Warehouse. John Wiley & Sons, INC. Second Edition. ISBN n° 0471-14161-5. 1996.

[Jarke et al. 99] Jarke M., Lenzerini M., Vassiliou M., Vassiliadis P., Fundamentals of Data Warehouses, Springer, 1999.

[Johanneson 93] Johannesson P., Using Conceptual Graph Theory to support Schema Integration, Springer-Verlag, 1999.

[Kedad et Métais 99] Kedad Z, Métais E. Dealing with Semantic Heterogeneity during Data Integration. ER'99, Paris (France), 1999.

[Kifer 95] Kifer M., EDIFF- a comprehensive interface to diff for Emacs 19. Available through anonymous ftp at ftp.cs.sunysb.edu, 1995.

[Kilpelaïnen 92] Kilpelaïnen P., Tree matching problems with application to structured text databases. Tech. Report, Dept. of Computer Science, Univ. of Helsinky, Finland, 1992.

[Labio 95] Labio W., Garcia-Molina H., Efficient algorithms to compare snapshots. Manuscript, available by anonymous ftp from db.stanford.edu in pub/labio/1995/, 1995.

[Lambolez 95] Lambolez P.Y., Queille J.P., Chrisment C., EXREP : a generic rewriting tool for textual information extraction. Revue ISI, "Ingénierie des Systèmes d'Information", vol. 3, n° 4, 471-485, 1995, Hermès.

[Loiseau et Prade 02] Loiseau Y., Prade H., Qualitative pattern matching with linguistic terms, STAIRS 2002, Th. Vidal, P. Liberatore eds., IOS Press, 2002.

[Marian et al. 00] Marian A., Abiteboul S., Mignet L., Change-centric management of versions in an XML Warehouse, BDA'2000, Blois, pp. 281-305.

[Milo 98] Milo T., Zohar S., Using Schema Matching to Simplify Heterogeneous Data Translation, VLDB'98, pp. 122-133, New York, August 1998.

[Mirbel 95] Mirbel I. Semantic Integration of Conceptual Shemes, 1st International Workshop on Application of Natural Language to Data Bases, 1995.

[Monge et Elkan 96] Monge A.E., Elkan C. P. "The field Matching Problem: Algorithms and Applications", KDD'96.

[Myers 86] Myers E., A difference algorithm and its variations. Algorithmica, 1(2):251-266, 1986.

[Réanimatic 2000] http://www.outcomerea.org

[Resnik 95] Resnik P., Using Information Content to Evaluate Semantic Similarity in a Taxonomy, IJCAI'95 (1995).

[Sèdes 98] Sèdes F., Bases d'objets documentaires – Hyperbases, Habilitation à Diriger les Recherches, Université Paul Sabatier, Toulouse 3, 1998.

[Shasha et al. 90] Shasha D., Zhang K., Fast algorithms for the unit cost editing distance between trees. Journal of Algorithms, 11(4):581-621, 1990.

[Sheth et Gala 89] Shet A.P. et Gala S.K., Attribute relationship : An impediment in Automatic shema Integration, Workshop on Heterogeneous Database Systems, decembre 1989.

[Song et al. 96] Song W.W., Johannesson P., Bubenko J.A., Semantic Similarity Relations and Computation in Schema Integration. Revue "Data and Knowledge Engineering", 19(1996).

[de Souza 86] Souza J.M. de, SIS – A Schema Integration System, Proc of the 5th BNCOD Conference, 1986.

[Villacampa 2002] Villacampa F., Analyser pour mieux décider, dans la revue "Décision micro & réseaux", n°499, mars 2002.

[Wang et al. 97] Wang J., Shasha D., Chang G., Relih V., Zhang K., Patel G., Structural Matching and Discovery in Document Databases, Proceedings of the ACM SIGMOD International Conf. on Management of Data, pp. 560-563, 1997.

[Wang et al. 94] Wang J. T. L., Zhang K., Jeong K., Sasha D., A system for approximate tree matching. IEEE Transactions on Knowledge and Data Engineering, 6(4) :559-571, Août 1994.

[Zhang et al. 89] Zhang K., Shasha D., Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal of Computing, 18(6):1245-1262, 1989.