Un état de l'art sur la qualité des données

Laure Berti-Équille

IRISA, Campus Universitaire de Beaulieu
F-35042 Rennes
Laure.Berti-Equille@irisa.fr

RÉSUMÉ. *Cet article propose une synthèse des travaux de recherche dans le domaine de la qualité des données. Il décrit les principales approches préemptives, rétrospectives et correctives jusqu'ici menées pour la modélisation, la mesure, le contrôle et l'amélioration de la qualité des données dans les bases de données relationnelles. Une nouvelle approche adaptative est également présentée. Elle permet d'inclure l'évaluation et la négociation de la qualité des données dans le traitement des requêtes et de proposer une recommandation des données selon leur qualité relative.*

ABSTRACT. *This paper proposes a synthesis of research works in the domain of data quality. It describes the main pre-emptive, retrospective and corrective approaches that have been conducted until now for modelling, measuring, controlling and improving data quality in relational databases. A new approach is also presented. It allows the evaluation and negotiation of data quality within the query processing and also proposes the recommendation of data depending on their relative quality.*

MOTS-CLÉS : *qualité de données, métadonnées, métriques, nettoyage.*

KEYWORDS: *Data Quality, metadata, metrics, cleaning.*

RSTI - ISI – 9/2004. Qualité des systèmes d'information, pages 1 à

1. Introduction

Avec la multiplication des sources d'informations disponibles et l'accroissement des volumes de données potentiellement accessibles, la qualité des données et, au sens large la qualité des informations n'ont cessé de prendre une place de premier plan tant au niveau académique qu'au sein des entreprises (Patterson, 1993 ; Bash, 1995 ; Wang *et al*., 1995 ; Redman, 1996 ; Redman, 2001 ; Wang, 2002 ; Dasu *et al.,* 2003 ; ICIQ1996-2004 ; Scannapieco *et al*., 2004) : il n'est plus question de « laisser-faire »[1] et l'urgence, unanimement reconnue, est de proposer des solutions pratiques aux multiples problèmes de non-qualité des données. Évaluer préalablement la qualité des données stockées dans les systèmes d'information, bases et entrepôts de données est essentiel afin de :

– proposer aux utilisateurs des mesures objectives et une expertise critique de la qualité des données,

– permettre à ceux-ci de relativiser la confiance qu'ils pourraient accorder aux données, et leur permettre ainsi, de mieux en adapter leur usage,

– évaluer finalement la validité et l'intérêt des connaissances extraites en assurant les décisions prises à partir des données.

En effet, une analyse des données et une prise de décision peuvent être réalisées sur des données inexactes, incomplètes, ambiguës et de qualité médiocre. On peut alors s'interroger sur le sens à donner aux résultats de ces analyses et remettre en cause, à juste titre, la qualité des connaissances ainsi « élaborées » et le bien-fondé des décisions prises.

L'objet de l'article est de présenter une synthèse des travaux de recherche menés jusqu'ici sur la qualité des données principalement dans les bases de données relationnelles.

Les principales motivations de l'article sont de souligner que : 1) l'impact de la non-qualité des données comme sa méconnaissance retentit à chaque étape du processus de traitement des données (modélisation des données, collecte/import, stockage, gestion, intégration, interrogation, et analyse des données), 2) de nombreuses techniques doivent être combinées pour mesurer et consolider la qualité des données, 3) de nombreuses voies de recherche restent aujourd'hui « grandes ouvertes » concernant, entre autre, les données complexes (telles que les données multimédias) et l'intégration des données hétérogènes.

Comme tout état de l'art, cet article a ses limites : proposer un fil conducteur qui construit de façon incrémentale l'évolution des techniques conçues pour remédier à la non-qualité des données est d'autant plus difficile que cet exercice est totalement tributaire et spécifique à l'évolution des besoins et aux problèmes de qualité de données rencontrés dans chaque discipline, domaine voire chaque application. La

1. C'est-à-dire, utiliser aveuglément les données sans en connaître la qualité et les laisser se dégrader,

notion de qualité des données n'est pas la même selon qu'elle est prise dans un contexte de conception de base données, de système d'information, d'interrogation de sources de données distribuées, de conception d'entrepôts de données ou de visualisation de données. Et donc, l'ambition de cet état de l'art n'est que d'offrir un parcours descriptif sur la qualité des données structurées.

L'article s'organise de la façon suivante : la suite de la section 1 répertorie les causes de la non-qualité des données et présente les principaux défis à relever dans le domaine de la recherche sur la qualité des données. La section 2 présente un panorama des travaux menés sur la modélisation, la mesure, le contrôle et la correction de la qualité des données selon des approches préemptives, rétrospectives et correctives, telles qu'elles sont proposées principalement par les communautés de recherche en bases de données et en statistiques. La section 3 propose une solution qui permet de négocier la qualité des données dans le traitement adaptatif de la requête. Enfin, la section 4 conclut l'article.

1.1. *Causes et conséquences de la non-qualité des données*

La qualité des données suscite, depuis environ une dizaine d'années, un vif intérêt, et ce thème émerge désormais comme un champ de recherche à part entière[2]. Plusieurs cas décrits dans (Bash, 1995 ; Redman, 1996), révèlent de nombreuses situations alarmantes concernant la qualité des données dans des bases de données commerciales, médicales, du domaine publique ou de l'industrie (Laudon, 1986 ; Celko, 1995) ou encore dans les systèmes d'information géographique (Goodchild *et al*., 1998).

Jusqu'à présent, les approches mises en œuvre sont souvent *ad hoc*, fragmentées et spécifiques à des domaines relativement cloisonnés. Les techniques utilisées sont essentiellement des méthodes statistiques ou des techniques issues des bases de données, de l'ingénierie des connaissances ou de la gestion de processus.

Les causes des problèmes de qualité de données sont d'origines diverses :

– lors de la modélisation conceptuelle des données, lorsque les définitions des attributs n'ont pas été suffisamment bien structurées ou normalisées, que le schéma n'a pas été validé ou qu'il manque des contraintes d'intégrité ou des procédures pour maintenir la cohérence des données,

---

2. Comme le témoigne l'organisation de conférences internationales *International Conference on Information Quality* (*ICIQ* http://web.mit.edu/tdqm/www/iqc) au *Massachusetts Institute of Technology depuis 1996*, de workshops européens tels que *Data Quality in Cooperative Information Systems* (*DQCIS'03* conjointement organisé avec *ICDT'03*) et *International Workshop on Information Quality in Information Systems* (*IQIS'04* conjointement organisé avec *SIGMOD'04*), ainsi que les éditions spéciales que lui consacrent les revues *IEEE Transactions on Data and Knowledge Engineering* (Wang *et al*., 1995 , Ballou *et al*., 2002) et *Communications of ACM* (Tayi *et al*., 1998).

– lorsque les interprétations des données sont incohérentes, notamment à cause de différences nationales ou culturelles dans l'usage de certains codes ou symboles,

– lors du développement logiciel : les besoins peuvent ne pas avoir été complètement spécifiés, ils ont pu évoluer et des erreurs peuvent avoir été introduites lors du processus de développement, notamment entre les phases d'analyse et de conception,

– lorsque la méthode de saisie des données n'a pas été bien conçue et qu'il manque des procédures de vérification systématique ou des techniques de mesure et de détection d'erreurs ou de doublons, car les erreurs humaines sont facilement introduites,

– lorsque la sécurité physique ou logique du système laisse à désirer : les données peuvent être corrompues volontairement,

– lorsque les entreprises n'ont pas les moyens de traquer l'âge de leurs données, ni de les mettre à jour, de les enrichir, ou encore d'éliminer les données devenues obsolètes,

– lors de l'intégration ou de la post-intégration de bases de données hétérogènes : les données des différents systèmes peuvent être contradictoires ou incohérentes, et les heuristiques d'appariement et d'intégration peuvent s'avérer inadaptées pour chacun des systèmes dont la qualité des données n'est d'ailleurs pas uniforme,

– lors de la conversion des systèmes ou de leur migration : les programmes de conversion peuvent introduire de nouvelles erreurs ; la rétro-ingénierie peut ne pas considérer (ou perdre) tout ou partie du contexte de définition, de production ou d'usage des données,

– lorsque des données répliquées sur différents sites ne sont pas correctement ou assez rapidement mises à jour : les copies secondaires ne sont plus cohérentes avec les copies primaires.

Ces raisons (et bien d'autres non mentionnées ici) motivent d'autant plus la prise en compte de la qualité des données à chaque étape de l'analyse, de la conception, du développement et de la maintenance d'un système de gestion des données, ainsi que dans toute la chaîne de traitement des données. Même si de nouvelles fonctionnalités sont ajoutées pour améliorer le processus de prise de décision, ainsi que la qualité des services (sans être trop grandes consommatrices de ressources, ni ajouter une trop grande complexité aux traitements), le besoin de méthodes, de métriques et d'outils pour définir, mesurer, interroger et contrôler la qualité des données dans les bases de données est une réalité tangible.

1.2. *Qualité des données : de nombreux défis à relever*

Les principaux enjeux de la recherche sur la qualité des données consistent à proposer des éléments de solutions pour évaluer et contrôler la qualité :

– face à l'hétérogénéité et la diversité des données à intégrer : les données sont extraites de différentes sources d'informations puis intégrées alors qu'elles possèdent différents niveaux d'abstraction (d'une donnée brute à un agrégat). Les données sont intégrées au sein d'un même jeu de données qui contient donc potentiellement une superposition de plusieurs traitements statistiques. Ceci nécessite une très grande précaution dans l'analyse de ces données. Les jointures entre les différents jeux de données sont difficiles (quand elles ne sont pas faussées) de par le problème de l'identification non-ambiguë des données et les données manquantes sont nombreuses ;

– face au passage à l'échelle et au problème de scalabilité des techniques : si certaines méthodes statistiques sont tout à fait adaptées pour fournir des résumés de la qualité de données numériques en grands volumes, elles s'avèrent inefficaces sur des données multidimensionnelles de grandes dimensions ;

– face aux données complexes telles que les séries temporelles, les données extraites de pages *web* (*web-scraped*), et les données multimédias (audio, vidéo, image) pour lesquelles on ne dispose pas (ou peu) de métriques de la qualité ;

– face aux données en temps réel et au problème du rafraîchissement des métadonnées reflétant la qualité : les bases de données modélisent une portion du monde réel en constante évolution. Or, dans le même temps, la qualité des données peut devenir obsolète. Le processus d'estimation et de contrôle doit être constamment reconduit en fonction de la dynamique du monde réel modélisé ainsi que des changements des centres d'intérêt et de l'attention particulière portée à certaines données (car des données jugées critiques à un instant donné ne le restent pas nécessairement).

Le tableau 1 présente pour chaque étape du traitement des données, les sources de problèmes liés à la qualité des données et les solutions actuellement proposées dans la littérature.

2. Travaux sur la qualité des données

Si la nécessité du contrôle et de la gestion de la non-qualité des données suscite une réelle prise de conscience au sein des différentes communautés de recherche en Bases de Données, Statistiques, Ingénierie des connaissances et Management, la première difficulté réside dans l'absence de consensus sur la notion même de qualité. Tout le monde s'accorde sur le fait que la qualité d'une donnée peut se décomposer en une multitude de dimensions, catégories, critères, facteurs, paramètres ou attributs (Wang *et al*., 1993 ; Fox *et al*., 1994 ; Redman, 1996 ; Wang *et al*., 1995 ; Mihaila *et al*., 2000), mais aucune définition ne fait l'unanimité.

ÉTAPES DE TRAITEMENT	SOURCES DE PROBLEMES	SOLUTIONS POTENTIELLES	REFERENCES
Création des données	Entrée manuelle Absence de standards pour les contenus et les formats Entrée de doublons Approximations Contraintes matérielles ou logicielles Erreurs de mesure	**Approche Préemptive** : Architecture pour la gestion de processus et workflows : (audits, intendance des données - "*data stewards*") **Approche Rétrospective:** – *Nettoyage de données* : élimination des doublons, merge/purge, appariement des noms et adresses, jointure approximative – *Diagnostic* : détection automatiques des erreurs et exceptions	(Wang, 2002 ; Redman, 1996 ; Redman, 2001 ; Tayi *et al.*, 1998 ; Huang *et al.*, 1999 ; Ballou *et al.*, 2002 ; Ballou *et al.*, 1995 ; Loshin, 2001 ; Paradice *et al.*, 1991 ; Delen *et al.*, 1992 ; Carlson, 2002) (Rahm *et al.*, 2000 ; Anathakrishna *et al.*, 2002 ; Winkler, 2003 ; Hernandez *et al.*, 1998 ; Navarro, 2001 ; Navarro *et al.*, 2001 ; Breunig et al., 2001 ; Monge, 2000 ; Pearson, 2002)
Collecte / import des données	Destruction ou mutilation d'information par des pré-traitements inappropriés Perte de données : *buffer overflows*, problèmes de transmission Absence de vérification	Utiliser des techniques de fouille de données pour vérifier que toutes les données ont été correctement transmises Vérifier la transmission, l'intégrité des données, leur format Suivi de données Édition de données (*data publishing*) Agrégation de données et *data squashing*	(Liepins *et al.*, 1990) (Dasu *et al.*, 2003) (DuMouchel *et al.*, 1999)
Stockage des données	Absence de méta-données, de mises à jour Modèles et structures de données inappropriés Modifications *Ad hoc* Contraintes matérielles ou logicielles	– Méta-données – Planification et customisation par domaine – Profilage des données, moniteur de navigation dans les données	(Mihaila *et al.*, 2000 ; Rothenberg, 1996 ; Dasu *et al.*, 2003) (Dasu *et al.*, 2002 ; Zhu *et al.*, 2002)

ÉTAPES DE TRAITEMENT	SOURCES DE PROBLEMES	SOLUTIONS POTENTIELLES	REFERENCES
Intégration des données	Multiple sources de données Synchronisation temporelle Systèmes de données non conventionnels Facteurs sociologiques Jointures *Ad hoc* Appariements aléatoires Heuristiques	– Exiger des estampilles temporelles précises pour assurer la cohérence logique et temporelle des jeux de données, lignage des données – Outils commerciaux pour : la migration des données et le nettoyage de données, le profilage de données – Outils académiques pour faire les appariements et les jointures entre jeux de données	(Cui *et al.*,2001) (Caruso *et al*., 2000) Potter's Wheel (Raman *et al*, 2001), Ajax (Galhardas *et al*., 2001), Bellman (Dasu *et al*., 2002), Arktos (Vassiliadis *et al*., 2000), HIQIQ (Naumann *et al*., 1999 ; Naumann, 2002)
Recherche et analyse des données	Erreur humaine Contraintes sur la complexité de calcul Contraintes logicielles, incompatibilité Problèmes de passage à l'échelle, de performances et de confiance dans les résultats Utilisation de boîtes noires pour l'analyse Attachement à une famille de modèles (statistiques) Expertise insuffisante d'un domaine Manque de familiarité avec les données	– Planification : évaluer le problème par rapport à l'équipement et vice versa – Fouille de données exploratoire (*Exploratory Data Mining* - EDM) – Plus grande responsabilité des analystes – Analyse en continu plutôt que ponctuel – Échantillonnage plutôt qu'analyse complète – Boucle de rétro-action	(Dasu *et al*., 2003 ; Pyle, 1999)

Tableau 1. *Qualité des données : problèmes et solutions proposées*

Chaque domaine d'application possède en fait sa vision spécifique de la qualité de ses données ainsi qu'une batterie de solutions généralement *ad hoc* pour remédier à ses problèmes. Les plus anciens travaux sur la modélisation de la qualité des données ont été menés par (Brodie, 1980) qui propose six concepts pour caractériser la qualité des données gérées par un système d'information (voir tableau A1 en annexes). La terminologie et les concepts fondamentaux sur ce thème ont également été précisés dans (Delen *et al*., 1992 ; Fox *et al*., 1994).

De nombreuses propositions ont depuis été faites et les dimensions les plus fréquemment mentionnées pour caractériser la qualité, sont : l'exactitude, la complétude, l'actualité, la cohérence (Fox *et al*., 1994 ; Wang *et al.*, 1995 ; Redman, 1996 ; Wang, 2002 ; Scannapieco *et al*., 2004) dont des définitions générales sont les suivantes :

– l'exactitude se mesure en détectant le taux de valeurs correctes dans la base de données,

– la complétude se mesure en détectant le taux de valeurs manquantes dans la base,

– l'actualité se mesure en détectant le taux de valeurs obsolètes dans la base de données par rapport à une date fixée ; la fraîcheur se mesure par une comparaison de la date de saisie à la date courante (voir l'article de Bouzeghoub et Peralta dans (Scannapieco *et al*., 2004) pour une étude très complète et détaillée de ces notions),

– la cohérence se mesure par rapport à un ensemble de contraintes en détectant les données de la base qui ne les satisfont pas.

Pour les définitions précises des dimensions de la qualité des données aux niveaux conceptuel et logique, ainsi que les mesures pour les évaluer, nous invitons le lecteur à consulter, entre autre, les articles référencés dans le tableau A1 en annexes, et les livres considérés comme fondamentaux dans le domaine écrits par Redman (Redman, 1996 ; Huang, *et al*., 1998 et Dasu *et al*., 2003.

2.1. *Approches préemptives centrées métadonnées*

2.1.1. *Usage des métadonnées*

Certains travaux intègrent totalement la modélisation et la gestion de la qualité des données dès la conception de la base de données. Parmi ces approches orientées processus et management, le programme *TDQM* (*Total Data Quality Management*) mené par Wang *et al.* au *Massachusetts Institute of Technology* (Wang, 1998 ; Wang *et al*., 1993) propose une méthodologie de modélisation de la qualité pour un modèle conceptuel de type Entité-Association qui guide pas à pas l'ajout de la dimension qualité sur chaque élément du modèle (entités, attributs, associations). Classiquement, la première étape consiste à modéliser le domaine d'application, sous la forme d'un modèle sémantique étendu de type Entité-Association. La figure 1, adaptée de (Wang *et al*., 1993) illustre le rôle des métadonnées et présente un

exemple de l'achat d'un produit par une société. La étape suivante de la modélisation consiste à rendre explicites des paramètres de qualité subjectifs sur chaque type d'entité, attribut ou association (constituant ainsi un premier niveau de méta-données reflétant la qualité). Ces paramètres subjectifs tels que la criticité, le niveau de détail d'une donnée sont ajoutés en tant que qualificatifs aux attributs de premier ordre définis lors de l'étape de modélisation précédente. Des indicateurs de qualité objectifs sont, par la suite, ajoutés. Ils correspondent uniquement à des paramètres subjectifs mesurables et sont des résultats de calculs automatiques (par des méthodes statistiques ou des contraintes). La dernière étape de la modélisation consiste à intégrer, au niveau du schéma conceptuel global les différentes vues-qualité qui ont pu être définies dans les trois phases précédentes (Wang *et al*., 1993).

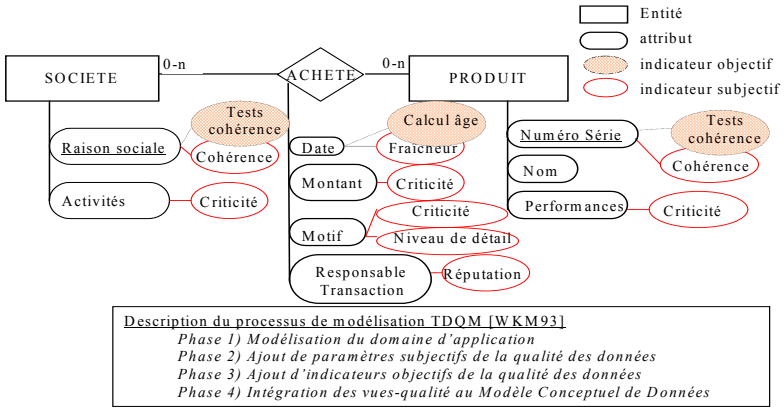

Figure 1. *Ajout des indicateurs subjectifs et objectifs reflétant la qualité des données*

Cette approche semble toutefois difficilement réalisable en pratique (Redman,1996), car l'ajout de métadonnées (labels) sur toutes les entités, associations et leurs attributs est une solution extrêmement coûteuse à la création et à la maintenance d'une telle base. L'utilisation de métadonnées pour l'évaluation et l'amélioration de la qualité des données est cependant préconisée dans (Rothenberg, 1996) et dans (Dasu *et al*., 2003 ; Rothenberg, 1996) incite les producteurs d'informations à assurer la vérification, validation et certification de leur données (*VV&C : Verification, Validation and Certification*) : les métadonnées concernant la qualité des données doivent être fournies avec les données pour permettre le processus d'estimation et de maintenance de la qualité des données.

Dans le contexte des documents et données semi-structurées, la tendance est bien à l'ajout et l'utilisation de métadonnées pour l'annotation qualitative des documents (Calabretto *et al*., 1998 ; TIPS, 1999 ; Hiom *et al*., 2000 , Berti, 2002). Le projet

DESIRE (Hiom *et al*., 2000) a produit une liste détaillée de standards de qualité à utiliser pour choisir des ressources du web selon diverses catégories de critères de qualité : 1) les critères liés à la politique de la diffusion de la ressource, 2) des critères liés au contenu, 3) des critères liés à la forme, 4) des critères liés à la gestion de la qualité de la documentation. Dans le cadre du projet européen *TIPS* (TIPS, 1999), plusieurs services ont été développés pour réutiliser des évaluations faites par des relecteurs humains sur les publications scientifiques : en particulier, l'outil *QCT* (*Quality Control Tools*) vise à rassembler les évaluations des documents afin d'enrichir leur indexation traditionnelle avec des informations qualité (voir tableau A1 pour les descripteurs de qualité utilisés dans *TIPS-QCT*).

Dans le contexte des entrepôts de données, de nombreux travaux ont été menés en particulier concernant la définition d'un modèle pour la qualité des entrepôts (Vassiliadis, 2000 ; Jarke *et al*., 1998 ; Jeusfeld *et al*., 1998 ; Vassiliadis *et al*., 1999), garder la trace du lignage des données permet de garder l'historique des transformations subies par une table ou un tuple, par exemple, sous la forme d'un graphe mentionnant les opérations de transformation et la source des données (Tayi *et al*., 2001 ; Cui *et al*., 2001). Ces métadonnées sont très utiles, par la suite, pour l'analyse et l'interprétation des distributions de probabilités sur les données tronquées ou censurées, le déboguage ou encore pour mettre en place des boucles de rétroaction permettant d'analyser les causes d'erreurs sur les données.

2.1.2. *Approches logiques basées sur des préférences de sources*

Selon une approche logique, (Cholvy, 1994) propose une définition axiomatique pour caractériser une base de données résultant d'une fusion de sources d'information contradictoires, en les classant selon un ordre total de fiabilité relative aux thèmes d'information qu'elles contiennent. La fusion des différentes sources d'information est menée selon deux attitudes :

– l'attitude suspicieuse qui consiste à suspecter toutes les informations fournies par une source qui contredit une source plus fiable. Elle se traduit par une perte de confiance totale en la source la moins fiable,

– l'attitude confiante qui consiste à suspecter seulement l'ensemble minimal des informations fournies par une source qui contredit une source plus fiable qu'elle.

Plus récemment, cette idée se retrouve dans les travaux de De Giacomo *et al.* dans (Scannapieco, 2004) qui exploitent les préférences sur les sources de données pour remédier aux incohérences lors de l'intégration.

L'hypothèse communément admise est de supposer que l'évaluation d'un critère de qualité est uniforme quel que soit le domaine de valeurs d'une source considérée. Ceci est peu réaliste car une source peut être considérée très fiable ou très actualisée sur certains thèmes qu'elle renseigne et non sur d'autres. De nombreux critères de qualité ne sont pas homogènes sur l'ensemble des valeurs proposées par une source. Les travaux de Motro et Rakov (Motro *et al*., 1997) sont intéressants dans la mesure où ils proposent et définissent la notion d'homogénéité de la qualité d'une base de

données, d'une vue ou d'un résultat de requête. Ils proposent une métrique pour évaluer l'homogénéité de la précision des données (voir tableau 3 ci-après). Par ailleurs, le prototype décrit par Sheth *et al.* dans (Sheth *et al*., 1993) vérifie si les données existantes sont correctes : les auteurs parlent de validation et de nettoyage des données au moyen d'un langage de données logique (*LDL++*). Le système emploie des contraintes pour la validation des données et des règles de nettoyage.

2.2. *Approches rétrospectives centrées métriques*

2.2.1. *Approche statistique et fouille de données exploratoire*

Les premières approches adoptées pour mesurer la qualité des données dans une base de données furent des approches basées sur les méthodes statistiques (Liepins *et al*., 1990 ; Parsaye *et al*., 1993), en particulier, pour inférer des données manquantes, prédire la précision des estimations à partir des données disponibles, éditer, suivre les données et détecter automatiquement des données isolées ou des erreurs (Paradice *et al*., 1991 ; Redman, 1996). L'utilisation des techniques d'apprentissage automatique pour la validation et la correction des données sont abordées dans (Parsaye *et al*., 1993). Ce livre montre en particulier comment les règles inférées à partir des instances de la base de données par les méthodes d'apprentissage peuvent être utilisées pour identifier les exceptions et faciliter le processus de validation des données et l'élimination des doublons. D'autres approches similaires peuvent être trouvées dans les travaux de (Schlimmer, 1991). L'utilisation de techniques statistiques pour améliorer la correction des bases de données et l'introduction d'un nouveau type de contraintes d'intégrité ont par ailleurs été proposées par Hou et Zhang (Hou *et al*., 1995). Les contraintes statistiques dynamiques sont dérivées à partir des instances de la base de données en utilisant les méthodes statistiques traditionnelles (échantillonnage, régression...). Chaque mise à jour de la base de données est validée si elle satisfait ces contraintes statistiques.

La fouille de données exploratoire (*Exploratory Data Mining* - *EDM*) (Dasu *et al*., 2003) est un ensemble de techniques statistiques proposant des résumés et des visualisations pour détecter des problèmes dans un ensemble de données avant de réaliser des analyses plus coûteuses. Elle a pour objectifs d'être largement applicable, rapide dans ses temps de réponse, simple à mettre en œuvre et ses résumés sont faciles à stocker, à mettre à jour et à interpréter. Elle révèle des informations qui permettent de faire des hypothèses, notamment sur la distribution des données ou les corrélations entre tel ou tel attribut (par exemple, une distribution gaussienne des valeurs d'un attribut *A* qui est lui-même lié de façon linéaire aux attributs *B* et *C* de la base). La fouille de données exploratoire peut être soit dirigée par les modèles et faciliter l'utilisation de méthodes paramétriques (modèles log-linéaires, par exemple), soit être dirigée par les données. Les résumés seront typiquement des moyennes, écarts types, médianes ou autres quantiles sur plusieurs

échantillons de l'ensemble des données. Ils permettront de caractériser le centre de la distribution des valeurs d'un attribut représentatif de la population, quantifier l'étendue de la dispersion des valeurs de l'attribut autour du centre, ou encore, décrire la dispersion (forme, densité, symétrie, etc.).

Concernant les techniques d'analyse sur des données manquantes, la méthode d'imputation par régression est décrite par (Litlle et Rubin 1987) est très utilisée, d'autres méthodes telles que celle de Monte Carlo par chaînes de Markov (*MCMC*) (Schafer, 1997) permettent de simuler des données en leur supposant une distribution normale multivariée. D'autres références traitant des données manquantes sont décrites dans le tutorial de Pearson (Pearson, 2002).

Concernant les données isolées (*outliers*) : les techniques de détection employées sont des graphes de contrôle, des techniques respectivement basées sur un modèle, une comparaison, sur des méthodes géométriques de mesure de distance à l'ensemble des données, sur la distribution (ou la densité) de la population des données (Knorr *et al*., 1998) avec la notion d'exception locales (*local outliers*) (Breunig *et al*., 2001), ou encore basées sur la déviation dans une série temporelle de données (Dasu *et al*., 2002). D'autres tests de « *goodness-of-fit* » tels que celui du *Chi2* permettent de vérifier l'indépendance des attributs ou encore le test de Kolmogorov-Smirnov de mesurer la distance maximum entre la distribution supposée des données et la distribution empirique calculée à partir des données. Ces tests univariés permettent de valider des techniques d'analyse et des hypothèses sur les modèles employés. D'autres tests plus complexes et multivariés sont également proposés (*DataSphere* : pyramides, hyperpyramides et test de Mahalanobis pour des distances entre moyennes multivariées (Johnson *et al*., 1998). Nous invitons le lecteur à la lecture du livre de Dasu et Johnson (Dasu *et al*., 2003) pour une description détaillée de ces techniques illustrées d'exemples.

2.2.2. *Métriques pour évaluer la qualité d'une modélisation de type entité-association*

Au niveau conceptuel, il est intéressant d'analyser la complexité du modèle à l'origine du schéma d'une base de données relationnelle, et ce, principalement pour en évaluer la maintenabilité. Ces mesures sont des supports quantitatifs afin de comparer des alternatives de conception et permettre l'identification des problèmes de conception qui ont nécessairement un impact plus ou moins direct sur la qualité des données stockées. Quelques-unes des métriques proposées dans la littérature, et en particulier dans l'ouvrage très riche de (Piattini *et al.*, 2002), sont présentées dans le tableau 2. Ce sont des mesures objectives et subjectives de la complexité et des facteurs de qualité d'un modèle conceptuel de type Entité-Association. La validation de ces métriques reste aujourd'hui un problème de recherche ouvert.

2.2.3. *Métriques pour évaluer la qualité d'une base de données relationnelle*

Parmi les nombreuses métriques proposées dans la littérature, le tableau 3 présente plusieurs propositions pour évaluer certaines dimensions de la qualité des données au niveau des instances. Citons les travaux de (Naumann *et al.*, 1999) ; Naumann, 2002) qui définissent plusieurs critères de qualité objectifs et subjectifs permettant d'établir des stratégies d'exécution de requêtes dans une architecture de type médiateur/adaptateurs (incluant la création, la sélection et l'ordonnancement de plans de requêtes selon la qualité des données et des sources). Dans ce contexte, plusieurs sources de données hétérogènes peuvent répondre à tout ou partie d'une requête. Ainsi, la qualité du résultat d'une requête est une fonction (fusion) de la qualité des sources et des données qui le compose.

La qualité des données n'est jamais définitivement acquise et elle évolue à chaque instant. En effet, à chaque action sur les données, peuvent être introduites des erreurs. Ne rien faire et laisser « vieillir » les données engendre également des problèmes de non-qualité. Les travaux tels que ceux de (Labrinidis et Roussopoulos 2001) sur la fraîcheur des données proposent une mesure de probabilité de la fraîcheur des données.

D'autres métriques ont été plus récemment proposées dans (Scannapieco *et al*., 2004) notamment par Motro *et al.*

AUTEURS	METRIQUES	VALIDATION	
		THEORIQUE	EMPIRIQUE
(Gray *et al.* 1991)	Complexité structurelle du modèle : $E=\sum_{i=1}^{n} Ei$ avec n entités Ei Complexité pour l'entité i : $Di=Ri*(a*FDAi+b*NFDAi)$ avec 0 < a <= b, Ri = nombre d'associations, $FDAi$ = nombre d'attributs en dépendances fonctionnelles, $NFDAi$ = nombre d'attributs sans dépendance fonctionnelle	Non	Non
(Moody *et al.* , 1998)	**Complétude** : <nombre d'items du modèle ne correspondant pas aux spécifications des utilisateurs, nombre de spécifications non représentées dans le modèle, nombre d'items du modèle qui correspondent aux spécifications mais qui sont mal définis, nombre d'incohérences dans la modélisation> **Intégrité** : <nombre de contraintes non satisfaites par les données, nombre de contraintes incluses dans le modèle qui ne correspondent pas exactement à la réalité modélisée> **Flexibilité : <** nombre d'éléments dans le modèle sujets à modification, coût estimé des modifications, importance stratégique des modifications> **Compréhensibilité** : <estimation par l'utilisateur du caractère compréhensible et interprétable du modèle> **Correction** : <nombre de violations des conventions du modèle de données, nombre de violations des formes normales, nombre d'instances redondantes dans le modèle> **Simplicité** : <nombre d'entités et associations ; somme pondérée ($aN^E + bN^R + cN^A$), où N^E est le nombre d'entités, N^R le nombre d'associations et N^A le nombre d'attributs> **Intégration** : <nombre de conflits avec le modèle de données commun, nombre de conflits avec les systèmes existants> **Implémentabilité** : <estimation du risque technique, estimation du risque de planification, , estimation du coût de développement, nombre d'éléments du niveau physique inclus dans le modèle>	Non	Non
(Genero *et al.* , 2000)	**NE** : nombre total d'entités dans le modèle ; **NA** : nombre total d'attributs d'entités et d'associations (simples ou composés); **DA** : nombre d'attributs dérivés (i.e., attributs dont la valeur peut être déduite ou calculée) ; **CA** : nombre total d'attributs composés ; **MVA** : nombre total d'attributs multi-valués ; **NR** : nombre total d'associations dans le modèle ; **M:NR** : nombre total d'associations M:N ; **1:NR** : nombre total d'associations 1:N et 1:1 ; **N-AryR** : nombre total d'associations N-aires ; **BinaryR** : nombre total d'associations binaires ; **NIS-AR** : nombre total d'associations IS_A (généralisation/spécialisation) ; **RefR** : nombre total d'associations cycliques ; **RR** : nombre total d'association redondantes	Oui	Partiellement
(Piattini *et al.* , 2000)	**RvsE:** rapport entre le nombre d'associations NR et le nombre d'entités NE du modèle $RvsE=\left(\frac{NR}{NR+NE}\right)^2$ avec $NR + NE > 0$. **DA:** rapport entre le nombre d'attributs dérivés NDA et le nombre maximal d'attributs dérivés possibles NA : $DA=\frac{NDA}{NA-1}$ avec $NA > 1$. **CA:** rapport entre le nombre d'attributs composés NCA et le nombre d'attributs total NA : $CA=\frac{NCA}{NA}$ avec $NA > 0$. **RR:** rapport entre le nombre d'associations redondantes NRR et le nombre d'associations NR : $RR=\frac{NRR}{NR-1}$ avec $NR > 1$. **M:NRel:** rapport entre NM:NR, le nombre d'associations M:N sur le nombre d'associations NR : $M:NRel=\frac{NM:NR}{NR}$ avec $NR > 1$. **FLeaf :** mesure de la complexité des hiérarchies (IS_A) : $FLeaf=\frac{NLeaf}{NE}$ avec $NLeaf$: nombre d'entités filles dans une hiérarchie généralisation/spécialisation et NE nombre d'entités dans chaque hiérarchie, $NE > 0$. **IS_Arel** : nombre moyen de supertypes directs et indirects par entité non racine $ALLSup$: $IS\_ARel=FLeaf-\frac{RLeaf}{ALLSup}$	Non	Partiellement

Tableau 2. - *Métriques utilisées pour évaluer la complexité et la qualité d'un modèle conceptuel de type Entité-Association*

AUTEURS	DEFINITION DES METRIQUES	
(Naumann, Leser, Freytag, 1999)	Spécifiques à la source de données	**Facilité de compréhension** : jugement de l'utilisateur de 1 à 10 basé sur la présentation des données **Réputation** : jugement de l'utilisateur de 1 à 10 basé sur des préférences personnelles et son expérience **Fiabilité** : score de 1 à 10 basé sur la précision des méthodes expérimentales de recueil et/ou de production des données **Actualité** : fréquence de mise à jour mesurée en nombre de jours
	Spécifiques aux requêtes	**Disponibilité** : pourcentage de temps pendant lequel la source de données est accessible **Prix** : prix d'une requête en US dollars **Temps de réponse** : temps d'attente moyen en secondes par requête **Précision** : pourcentage de données sans erreur **Pertinence** : pourcentage d'objets du monde réel représentés par la source de données
	Spécifiques aux attributs	**Complétude** : pourcentage de valeurs non nulles
(Motro, Rakov, 1997)	**Complétude** : proportion des informations correctement stockées D par rapport au monde réel modélisé W: $C=\frac{\lvert D\cap W\rvert}{\lvert W\rvert}$ **Précision** (soundness) : proportion des informations correctement stockées : $S=\frac{\lvert D\cap W\rvert}{\lvert D\rvert}$ **Homogénéité de la qualité d'une vue** : différentiel moyen entre la précision d'une vue v et celle de toutes ses sous-vues possibles v_i $Hs(v)=\frac{1}{N}\sum_{i=1}^{N}\lvert S(v)-S(v_i)\rvert$	
(Labrinidis, Roussopoulos, 2001)	**Fraîcheur** : probabilité d'accéder une version à jour d'une vue v de la base de données sur une période donnée T=[ti,tj] : $Fresh(v)=\frac{1}{T}\times\int_{ti}^{tj} f(d_i)^t$ avec $f(d_i)^t=\begin{cases}0, si la donnée di est obsolète au temps t\\1, si la donnée di n'est pas obsolète au temps t\end{cases}$	

Tableau 3. – *Quelques métriques pour évaluer des dimensions de la qualité des données*

2.3. *Approches correctives par des techniques de base de données*

2.3.1. *Nettoyage des données par extraction et transformation*

L'intérêt des métriques sur la qualité des données est de pouvoir les exploiter à des fins d'amélioration de la qualité de la base. Le nettoyage de données fait partie des stratégies d'amélioration automatique de la qualité des données et se décompose en trois étapes (Rahm et Do, 2000), (Winkler, 2003) (Bouzeghoub et Lenzerini, 2001) : 1) auditer les données afin de détecter les incohérences, 2) choisir les transformations pour résoudre les problèmes de non-qualité, 3) appliquer les transformations choisies au jeu de données.

Pour la mise en œuvre du nettoyage, sont utilisés des outils commerciaux d'audit (tels que *ACR/Data* d'Unitech Systems ou *Migration Architect* d'Evoke) et des outils de transformation *ETL - Extraction/Transformation/Loading* (tels que *Data Junction* ou *DataStage* d'Ascential Software).

Parmi les outils académiques d'extraction et de transformation, citons Potter's Wheel (Raman et Hellerstein, 2001), Arktos (Vassiliadis *et al.* 2000), AJAX (Galhardas *et al.*, 2001), Bellman (Dasu *et al.*, 2002) qui permettent l'extraction de structures et expressions régulières, la transformation de valeurs de données (par application de fonctions de formatage), la transformation de l'ensemble des valeurs de tuples (lignes) et d'attributs (colonnes) d'une base de données relationnelle. Les

opérations de transformation possibles sont récapitulées dans la Figure 2 sur un exemple extrait de (Raman et Hellerstein, 2001).

TRANSFORMATION	DÉFINITION FORMELLE
Formatage	$\phi(R, i, f) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n, f(a_i))\mid (a_1,\dots, a_n) \in R\}$
Ajout	$\alpha(R, x) = \{(a_1,\dots, a_n, x)\mid (a_1,\dots, a_n) \in R\}$
Suppression	$\pi(R, i) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n)\mid (a_1,\dots, a_n) \in R\}$
Copie	$\kappa((a_1,\dots, a_n), i) = \{(a_1,\dots, a_n, a_i)\mid (a_1,\dots, a_n) \in R\}$
Fusion	$\mu((a_1,\dots, a_n,), i, j, glue) = \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_{j-1}, a_{j+1},\dots, a_n, a_i \oplus \text{glue} \oplus a_j)\mid (a_1,\dots, a_n) \in R\}$
Division	$\delta((a_1,\dots, a_n), i, \text{pred}) = \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_n, a_i, \text{null})\mid (a_1,\dots, a_n) \in R \wedge \text{pred}(a_i)\} \cup \{(a_1,\dots, a_{i-1}, a_{i+1}, \dots, a_n, \text{null}, a_i)\mid (a_1,\dots, a_n) \in R \wedge \neg\text{pred}(a_i)\}$
Partage	$\omega((a_1,\dots, a_n,), i, splitter) = \{(a_1,\dots, a_{i-1}, a_{i+1},\dots, a_n, \text{left}(a_i, splitter), \text{right}(a_i, \text{splitter}))\mid (a_1,\dots, a_n) \in R\}$
Repliement	$\lambda(R, i_1, i_2,\dots, i_k) = \{(a_1,\dots, a_{i1-1}, a_{i1+1},\dots, a_{i2-1}, a_{i2+1},\dots, a_{ik-1}, a_{ik+1},\dots, a_n, a_{il})\mid (a_1,\dots, a_n) \in R \wedge 1 <= l <= k\}$
Sélection	$\sigma(R, \text{pred}) = \{(a_1,\dots, a_n)\mid (a_1,\dots, a_n) \in R \wedge \text{pred}((a_1,\dots, a_n))\}$

Notation : R est une relation avec n attributs, i et j sont les indices des attributs ; a_i représente la valeur d'un attribut pour un tuple donné; x et *glue* sont des valeurs, f est une fonction de transformation d'une valeur en une autre ; $x \oplus y$ concatène x et y ; splitter est une position dans une chaîne de caractères ou une expression régulière ; left(x, splitter) est la partie gauche de x après avoir partagé la chaîne de caractères à la position indiquée par la variable splitter ; pred est une fonction retournant un booléen.

Exemple de transformation des données d'une table:

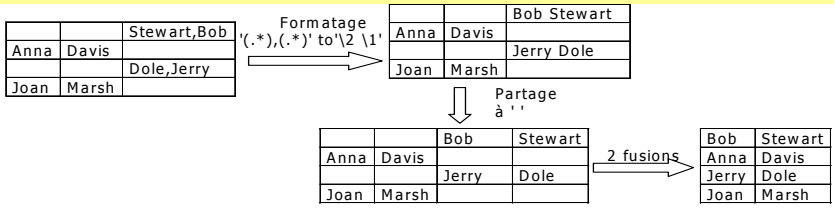

Figure 2. *Opérations de transformation utilisées pour le nettoyage des données*

Potter's Wheel (Raman et Hellerstein, 2001) est un système ETL interactif permettant de visualiser les effets des transformations sur les données. Par un mécanisme d'analyse sur les domaines, il permet également de trouver des incohérences dans les valeurs d'attributs. Bellman (Dasu *et al.*, 2002) fournit une suite d'outils ETL et un outil de profilage de base de données. D'autres opérations, telles que le clustering et l'appariement sont employées pour la détection et l'élimination de doublons : *AJAX* (Galhardas *et al.*, 2001) est une extension du langage déclaratif SQL permettant de spécifier chaque transformation de données (*matching, merging, mapping, clustering*) nécessaire au processus de nettoyage des données. Ces transformations normalisent les formats de données et détectent les paires d'enregistrements qui se rapportent le plus probablement au même objet. Cette étape d'élimination des doublons est appliquée si des données approximativement redondantes sont trouvées et un appariement multitable calcule des jointures par similarité entre des données distinctes ce qui permet leur consolidation (voir les derniers travaux de Galhardas et Carreira dans (Scannapieco

et Naumann, 2004). Nous renvoyons le lecteur au panorama des travaux en ETL proposé par Rahm et Do (Rahm et Do, 2000) ainsi qu'à l'article de Winkler (Winkler, 2003) pour plus d'informations.

2.3.2. Appariement approximatif et élimination des doublons

Dans le cadre d'une intégration de plusieurs bases de données relationnelles, il est nécessaire d'associer plusieurs tables au moyen de jointures pour lesquelles, bien souvent, on ne dispose pas de clés suffisamment précises. La technique de jointure approximative est alors employée lorsque les clés contiennent des informations imprécises (Hernandez et Stolfo, 1998), (Monge, 2000), (Anathakrishna *et al.*, 2002). Par exemple, pour apparier les données des clients d'une base sur les facturations et celles d'une base de données sur les installations, les noms des clients peuvent être écrits de différentes façons ("FT" ou "France Télécom") et il peut être difficile de faire l'appariement sur les noms des clients. Si, en revanche, le numéro de téléphone est le même, on pourra supposer qu'il s'agit bien de la même entité, c'est pourquoi il s'avère nécessaire d'examiner les informations qui corroborent ou non une hypothèse d'appariement. Très spécifique à l'application, la technique de jointure approximative consiste à regrouper et trier les tuples par paquets selon une fonction de hachage sur les valeurs d'un ou plusieurs attributs (par exemple, utilisant les premières lettres ou consonnes des noms propres). Les tuples qui se trouvent dans les mêmes paquets sont candidats à l'appariement et pour chaque paire de candidats, une fonction d'appariement est calculée et seuls les paires de plus haut score sont effectivement appariées (ou assimilées à des doublons). Une implémentation de la technique de jointure approximative est décrite dans (Caruso *et al.*, 2000). Pour des domaines d'attributs textuels, l'appariement des chaînes de caractères (*string matching*) calcule une distance comptabilisant le nombre d'opérations d'édition (telles que l'ajout, la suppression d'un caractère, ou le changement de lettre) nécessaires pour transformer une chaîne de caractères en une autre. Par exemple, "Telecom" a une distance d'édition de 2 de "Telefon". Les chaînes de caractères dont la distance d'édition est inférieure à un seuil fixé seront appariées. L'ensemble des algorithmes d'appariement de chaînes de caractères sont détaillés dans (Navarro, 2001) et (Navarro *et al.*, 2001). Pour des données semi-structurées en XML, l'appariement d'arbres est une technique analogue à l'appariement de chaîne de caractères par le calcul d'une distance d'édition, mais elle est beaucoup plus coûteuse (Koudas *et al.*, 2002) : le calcul d'une distance d'édition sur n caractères est en $O(n^2)$ alors que le calcul d'une distance d'édition sur l feuilles d'arbre est en $O(l^4)$. Toutefois, une approche intéressante d'appariement et de nettoyage de données XML a été récemment proposée par Weiss et Naumann dans (Scannapieco et Naumann, 2004).

3. Une approche adaptative pour négocier la qualité des données

Très peu de projets se sont intéressés particulièrement au traitement et à la planification des requêtes basés sur la qualité des données. Les travaux de Naumann (Naumann *et al.*, 1999) (Naumann, 1999) proposent un algorithme de planification de requêtes distribuées, *HiQ B&B* (*High Quality Branch and Bound Algorithm*), qui inclut une phase de sélection des sources selon leur qualité dans l'énumération des plans de requête, en supposant renseignées, une fois pour toute, les métadonnées sur la qualité définies dans le tableau 3.

Sur cet aspect, notre approche se veut plus flexible et adaptative dans la mesure où les dimensions de la qualité doivent pouvoir être définies dans la requête et le traitement de la requête être adapté en conséquence. Nos travaux présentés dans (Berti-Équille, 2003 et 2004) étendent la syntaxe d'un langage de requêtes de type SQL par une adaptation du langage QML (*Quality of Service Modeling Language*) proposé par Frølund et Koistinen (Frølund et Koistinen, 1998) et dédié à la qualité de service. Cette extension, nommée *XQuaL*, dont la syntaxe est donnée dans le Tableau A2 en annexes est un langage de description de métadonnées sur la qualité et de manipulation de données et métadonnées permettant de déclarer des types de contrat qualité sur les données distribuées ainsi que d'étendre une requête de type SQL avec une clause Qwith qui vérifie si le contrat qualité prédéfini est vérifié et qui, le cas échéant, négocie un autre contrat sur la qualité des données requêtées pour une qualité optimale sur l'ensemble des sources qui peuvent répondre à la requête.

3.1. Extension d'un langage de requête avec QML

3.1.1. Description des métadonnées sur la qualité

Un contrat est un ensemble de contraintes librement définies par l'utilisateur sur toutes les dimensions de la qualité que ce dernier requiert. Chaque contrat est une instance d'un type de contrat. La figure (3a) propose un exemple de déclarations de plusieurs types contrats pour la fiabilité, la fraîcheur, la complétude et la crédibilité des données, respectivement nommées Reliability, Freshness, Completeness, et Credibility. La figure (3b) donne des instances de contrats correspondantes. Le type de contrat Reliability possède quatre dimensions, notées d1,..., d4 en commentaires sur la figure (3a), nommées respectivement failureMasking, serverFailure, numberOfFailures et availability. La dimension failureMasking est définie sur plusieurs valeurs possibles parmi omission, lostResponse, noExecution, etc. et la relation (decreasing) est interprétée telle que les plus petits ensembles de ces valeurs sont préférables pour cette dimension. La dimension serverFailure est définie par des valeurs uniques sans notion d'ordre. Les dimensions numberOfFailures et availability sont numériques. nf/year est l'unité de numberOfFailures en nombre de pannes par

an. La Figure 3b présente quatre instances de ces types de contrats pour la source S1: S1_Reliability, S1_Freshness, S1_Completeness, S1_Credibility.

```
type Reliability = contract {
        failureMasking : decreasing set {omission, lostResponse,
                                noExecution, response, responseValue, stateTransition } ;   //d1
        serverFailure : enum { halt, initialState, rolledBack } ;                           //d2
        numberOfFailures : decreasing numeric nf / year ;                                   //d3
        availability : increasing numeric ;                                                 //d4
} ;
type Freshness = contract {
        dataAge : decreasing numeric seconds;                                               //d5
        lastUpdate : numeric seconds ;                                                      //d6
        updateFrequency : numeric uf / month ;                                              //d7
} ;
type Completeness = contract {
        percentageOfNullValues : percentile number %;                                       //d8
        numberOfNullValuesPerQuery : numeric nnv / query ;                                  //d9
} ;
type Credibility = contract {
        reputation : enum {veryGood, good, bad, veryBad, unknown } ;                        //d10
} ;
```

Figure (3a). *Exemple de types de contrats*

```
S1_Reliability = Reliability contract {
  failureMasking < { noExecution, response };
  serverFailure == initialState ;
  numberOfFailures < 10 nf / year ;
  availability > 0.8 ;
} ;
S1_Freshness = Freshness contract {
  dataAge < 4200 seconds;
  lastUpdate == 4500 seconds;
  updateFrequency == 3 uf / month;
};
 S1_Completeness = Completeness contract {
    PercentageOfNullValues == 8 %;
    NumberOfNullPerQuery == 2 nnv/query;
 };
 S1_Credibility = Credibility contract {
    reputation == veryGood ;
 };
```

FIGURE (3b). *Exemple d'instances de contrats*

Une requête SQL étendue (Qwith-query) contient l'opérateur Qwith utilisé pour déclarer les types de contrats (contractTypeDeclaration), les contrats (contractDeclaration) ou les profils qualité (profileDeclaration) définis ci-après (voir

tableau A2 en annexes). La qualité peut être définie de façon flexible par une combinaison spécifique de contrats ayant une relation d'ordre sur les contraintes de qualité de données ou de sources. La figure 4 présente un exemple de définition de la qualité comme un type complexe de contrat utilisant les types de contrats précédemment définis dans la figure (3a).

```
type Quality contract {
 s : increasing set {Reliability,Freshness,Completeness,Credibility }
     with order {Reliability<Completeness,
                 Completeness< Freshness,
                 Freshness < Credibility }
} ;
```

Figure 4. *Exemple d'une déclaration de contrat sur la qualité des données*

3.1.2. Manipulation des métadonnées sur la qualité

Un profil est un ensemble de contrats requis qui est affecté à une ou plusieurs sources de données ou à une ou plusieurs de leurs fonctionnalités. Les profils sont utilisés pour associer des contrats aux interfaces des sources (par exemple aux adaptateurs dans une architecture médiateur/adaptateurs). Une déclaration de profil (profileDeclaration) est définie avec l'identifiant du profil, l'identifiant de l'interface et une série d'expressions (profileExpression) qui requière la satisfaction des contrats par les interfaces ciblées (requisites) (voir Tableau A2 en annexes).

La déclaration d'un profil (profile) permet, au sein d'une requête, d'associer des contrats qualité à des sources, à leurs interfaces, à leurs services ou à leurs méthodes spécifiques (par exemple, query_answer_method pour la source S1 dans la Figure 5) ci-après.

```
Source_Profile for S1 = profile {
        require S1_Freshness, S1_Completeness, S1_Credibility ;
        from S1.query_answer_method require S1_Reliability contract ;
} ;
```

Figure 5. *Exemple de profil requis pour la source S1*

La clause require lie toutes les interfaces à une liste de contrats, alors que la clause from... require... identifie de façon spécifique une liste d'interfaces à laquelle la liste de définitions de contrats est associée dans la clause require. Le rôle des profils est donc d'étendre la requête par des contraintes sur la qualité en ciblant précisément la (les) ressource(s) et les interfaces qui doivent s'y conformer.

3.2. Négociation de la qualité dans le traitement des requêtes

Trouver un équilibre entre le coût d'une requête et la qualité d'un résultat est le principe de la négociation afin d'obtenir des résultats de "bonne qualité" en un temps acceptable. La négociation peut être menée sur toutes les dimensions de la

qualité. La technique de négociation repose sur les contrats qualité sur les données, les sources ou leurs services. Nous formalisons notre modèle de négociation pour inclure des contraintes liées à la qualité des données dans les requêtes. Soit i ($i \in [1..n]$) représentant les sources et j ($j \in [1..k]$) les dimensions de contrat qualité et les profils pour la négociation (par exemple, dataAge, failureMasking, serverFailure, ... , numberOfNullValuesPerQuery, availability etc. précédemment définis dans la Figure 3). Chaque source possède un score de qualité $score_{ij}$: $[min_{ij}, max_{ij}] \rightarrow [0,1]$ qui représente le score de la source i sur la dimension de la qualité j dans une gamme de valeurs acceptables entre [0,1]. L'importance relative que l'utilisateur pourra affecter à chaque dimension de la qualité pour la négociation est représentée par un poids, noté w_{ij}, qui donne l'importance de la dimension j pour la source i. Nous supposons les poids normalisés.

$$\sum_{1 \le j \le k} w_{ij} = 1 \quad \forall i \in [1..n], \forall j \in [1..k]$$

Une fonction d'agrégation des scores pour la source j dans l'espace multidimensionnel de la qualité définie par $x = (x_1, ..., x_k)$ sur les k dimensions de la qualité est telle que :

$$Score_i(x) = \sum_{1 \le j \le k} w_{ij}.score_{ij}(x_j)$$

La négociation consiste à trouver le meilleur équilibre entre un niveau de qualité requis, noté θ et appelé niveau de qualité aspirationnel, et les contrats qualité effectivement remplis par les sources de données au cours du traitement d'une requête étendue par la clause Qwith. Le module de négociation décide, pour réaliser un équilibre, de renégocier le contrat quand le niveau de qualité aspirationnel ne peut être atteint par les sources. Alors le négociateur calcule le différentiel de qualité entre la qualité de chaque source de données et la qualité requise représentée par la courbe du niveau aspirationnel θ. Et il génère le contrat qui correspond à θ. Le mécanisme trouve ensuite l'instance de contrat la plus proche de la courbe du niveau aspirationnel initial ou la plus préférable (c'est-à-dire celle qui maximise la qualité et minimises le coût de la requête). Compte tenu de la limitation en nombre de pages de l'article, nous renvoyons le lecteur à (Berti-Équille, 2004) pour la description détaillée de l'algorithme de négociation utilisé dans le traitement des requêtes étendues.

4. Conclusion

La plupart des bases de données maintenues actuellement sont entachées d'incertitude et contiennent des erreurs ou des données de qualité "médiocre". Face à cette réalité, aux montants financiers en jeu et à la multiplication des échanges de données, tous les acteurs impliqués dans l'utilisation et l'analyse des données ont pris conscience de la nécessité de mesurer et contrôler la qualité des données. L'objet de cet article a été dans un premier temps, de dresser les causes de la non-qualité des données, et de présenter un panorama des travaux sur la qualité des

données en privilégiant un éclairage "en largeur d'abord" sur les méthodes statistiques et sur les techniques des bases de données utilisées dans les approches préemptives, rétrospectives ou correctives de la qualité des données. Enfin, nous avons brièvement présenté nos travaux sur le traitement adaptatif des requêtes pour inclure une négociation de la qualité des données distribuées.

La qualité des données demeure une notion complexe, multidimensionnelle et polymorphe qui ne peut être capturée par un ensemble de nombres ou règles statiques. Il est nécessaire de concevoir des processus, des systèmes et des techniques d'analyse qui surveillent constamment les données et corrigent celles qui ne se conforment pas à un ensemble de règles qui sont elles-mêmes constamment scrutées, mises à jour et documentées aussi bien que les changements de besoins en termes de qualité des données. Une des complexités du problème est qu'il y a souvent plus d'exceptions que de règles dès qu'il s'agit de concevoir une description des données jugées "de bonne qualité". Une solution générale et réutilisable qui scrute automatiquement les données, crée un ensemble de contraintes sur la qualité des données et isole les données qui ne satisfont pas ces contraintes est loin de pouvoir être d'actualité. En cela, de nombreuses et intéressantes perspectives de recherche sont offertes.

5. Bibliographie

Anathakrishna R., Chaudhuri S., Ganti V., Eliminating fuzzy duplicates in datawarehouses, *Intl. Conf. VLDB*, 2002.

Ballou D.P., Pazer H. Designing information systems to optimize the accuracy-timeliness trade-off, Information Systems Research, 6(1), 1995.

Ballou D.P., Pazer H., Modeling completeness versus consistency trade-offs in information decision contexts. *IEEE TKDE*, 15(1): 240-243, 2002.

Bash R. (Ed.), Electronic information delivery: ensuring quality and value, 1995.

Berti-Équille L., Quality-extended query processing for distributed sources, *Intl. Workshop DQCIS'2003*, Siena, Italy, 2003.

Berti-Équille L., Quality-Adaptive Query Processing over Distributed Sources, *Intl. Conference on Information Quality*, pages 285-296, M.I.T., Boston, U.S., 2004.

Breunig M., Kriegel H., Ng R., Sander J., LOF: Identifying density-based local outliers, *Intl. Conf. ACM SIGMOD*, p. 93-104, 2000.

Brodie M. L., Data quality in information systems, *Information and Management*, vol. 3, p. 245-258, 1980.

Bouzeghoub M., Lenzerini M. (Eds), Introduction to the special issue on data extraction, cleaning, and reconciliation, *J. of Information Systems*, 26(8), 2001.

Calabretto S., Pinon J. M., Poullet L., Richez M.-A., De la qualité de l'information à la qualité de la documentation, *Document Numérique*, 12(1):37-52, 1998.

Carlson D., Data Stewardship in action, *DM Review*, May 2002.

Caruso F., Cochinwala M., Ganapathy U., Lalk G., Missier P., Telcordia's database reconciliation and data quality analysis tool, *Intl. Conf. VLDB*, p. 615-618, 2000.

Cholvy L., A logical approach to multi-source reasoning. *Lecture Notes in Artificial Intelligence*, vol. 808, 1994.

Celko J., McDonald J. Don't warehouse dirty data, *Datamation*, 41(18), 1995.

Cui Y., Widom J., Lineage Tracing for general data warehouse transformation. *Intl. Conf. VLDB*, p. 471-480, 2001.

Dasu T., Johnson T., Exploratory Data Mining and Data cleaning, Wiley, 2003.

Dasu T., Johnson T., Muthukrishnan S., Shkapenyuk V., Mining database structure or, how to build a data quality browser, *Intl. Conf ACM SIGMOD*, 2002.

Delen G., Rijsenbrij D., The specification, engineering and measurement of information systems quality*, J. of Software Systems*, no.17, p. 205-217, 1992.

DQIS, Proceedings of the Intl. Workshop on Data Quality in Cooperative Information Systems, Siena, Italy, 2003.

DuMouchel W., Volinsky C., Johnson T., Cortez C., Pregibon D., Squashing flat files flatter, *Intl. Conf. Knowledge Discovery and Data Mining*, p. 6-16, 1999.

Fox C., Levitin A., Redman T., The notion of data and its quality dimensions, *Information Processing and Management*, vol. 30, no. 1, 1994.

Frølund S. and Koistinen J., QML: A Language for Quality of Service Specification. Tech. Rep. HPL-98-10, Software Technology Laboratory, Hewlett-Packard, 1998.

Galhardas H., Florescu D., Shasha D., and Simon E., Saita C., Declarative Data Cleaning: Language, m odel and algorithms, *Intl. Conf. VLDB*, p.371-380, 2001.

Gray R., Carey B., McGlynn N., Pengelly A., Design metrics for database systems, *BT Technology J.,* 9(4), 1991, 69-79.

Genero M., Piattini M., Calero C., Serrano M., Measures to get better quality databases, *ICEIS 2000,* p. 49-55, July 2000.

Goodchild M., Jeansoulin R. (Ed.), Data quality in geographic information: from error to uncertainty, Hermès, 1998.

Hernandez M., Stolfo S., Real-world data is dirty: data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9-37, 1998.

Hiom D., Belcher M., Place E., People Power and the Web: Building Quality Controlled Portals, *TERENA Networking Conference*, 2000. http://www.desire.org/

Hou W.C., Zhang Z., Enhancing database correctness: a statistical approach, *Intl. Conf. ACM SIGMOD*, 1995.

Huang K., Lee Y., Wang R., Quality Information and Knowledge Management, Prentice Hall, New Jersey, 1999.

ICIQ, Proceedings of the International Conference on Information Quality, MIT, Boston, from 1996 to 2004.

Jarke M., Jeusfeld M. A., Quix C., Vassiliadis P., Architecture and Quality in Data Warehouses, *Intl. Conf. CAiSE ,* p. 93-113, 1998.

Jeusfeld M.A., Quix C., Jarke M., Design and Analysis of Quality Information for Data Warehouses, *Intl. Conf. Conceptual Modeling*, p. 349-362, 1998.

Johnson T., Dasu T., Comparing Massive High-Dimensional Data Sets. *Intl. Conf. KDD*, p. 229-233, 1998.

Knorr E., Ng R., Algorithms for mining distance-based outliers in large datasets. *Intl. Conf. VLDB*, p. 392-403, 1998.

Koudas N., Srivastava D., Jagadish H., Guha S., Yu T., Approximate XML joins, *Intl. Conf. ACM SIGMOD*, 2002.

Labrinidis A., Roussopoulos N., Update propagation strategies for improving the quality of data on the web, *Intl. Conf. VLDB*, 2001.

Laudon K. C., Data quality and due process in large interorganizational record systems, *Com. of the ACM*, p. 4-11, 1986.

Liepins G., Uppuluri V., Data quality control: theory and pragmatics, M. Dekker, 1990.

Little R.J.A., Rubin D.B., Statistical analysis with missing data, Wiley, New-York, 1987.

Loshin D., Enterprise Knowledge Management: The data quality approach, Morgan Kaufmann, 2001.

Mihaila G. A., Raschid L., and Vidal M., Using quality of data metadata for source selection and ranking. *WebDB Workshop*, p. 93-98, 2000.

Monge A., Matching algorithms within a duplicate detection system, *IEEE Data Eng. Bull*., 23(4):14-20, 2000.

Moody L., Shanks G., Darke P., Improving the Quality of Entity Relationship Models - Experience in Research and Practice, *Intl. Conf. on Conceptual Modelling*, p. 255-276, 1998.

Motro A., Rakov I., Not all answers are equally good: estimating the quality of database answers. Flexible Answering Systems, Kluwer Academic Press, p.1-21, 1997.

Navarro G., A guided tour to approximate string matching, *ACM Computer Surveys,* 33(1):31-88, 2001.

Navarro G., Baeza-Yates R., Sutinen E., Tarhio J., Indexing Methods for Approximate String Matching. *IEEE Data Eng. Bull.,* 24(4): 19-27 (2001).

Naumann F., Quality-Driven Query Answering for Integrated Information Systems, *Lecture Notes in Computer Science,* vol. 2261, Springer, 2002.

Naumann F., Leser U., Freytag J., Quality-driven integration of heterogeneous information systems, *Intl. Conf. VLDB*, 1999.

Paradice D.B., Fuerst W.L., An MIS data quality management strategy based on an optimal methodology. *J. of Information Systems*, 5(1):48 - 66, 1991.

Parsaye K., Chignell M. Data quality control with smart databases, *AI Expert*, 8(5): 23 - 27. 1993.

Patterson B., "The need for data quality", *Intl. Conf. VLDB,* 1993.

Pearson R.K., Data Mining in face of contaminated and incomplete records, *SIAM Intl. Conf. Data Mining*, 2002.

Piattini M., Genero M., Calero C., Polo C., Ruiz F., Database Quality, In Chapter 14: *Advanced Database Technology and Design*, Artech House, p. 485-509, 2000.

Piattini, M., Calero C. Genero M. (Eds.), Information and Database Quality, *The Kluwer International Series on Advances in Database Systems*, Vol. 25, 2002.

Pyle D., Data Preparation for Data Mining, Morgan Kaufmann, 1999.

Rahm E., Do H., Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull*. 23(4): 3-13, 2000.

Raman V., Hellerstein J. M., Potter's wheel: an interactive data cleaning system, *Intl. Conf. VLDB*, 2001.

Redman T., Data quality for the information age, Artech House Publishers, 1996.

Redman T., Data quality: The Field Guide, Digital Press (Elsevier), 2001.

Rothenberg J., Metadata to support data quality and longevity, *IEEE Metadata Conf*., 1996.

Scannapieco M., Naumann F. (Eds), *1st Intl. ACM SIGMOD Workshop on Information Quality in Information Systems*, 2004.

Schafer J.L., Analysis of incomplete multivariate data, Chapman & Hall, 1997.

Schlimmer J., Learning determinations and checking databases, *AAAI Workshop on Knowledge Discovery in Databases*, 1991.

Sheth A., Wood C., Kashyap V., Q-Data: Using Deductive Database Technology to Improve Data Quality, Workshop on Programming with Logic Databases, *ILPS*, p. 23-56, 1993.

Tayi G. K., Ballou D. P., Examining Data Quality, *Com. of the ACM*, 41(2):54-57, 1998.

Theodoratos D., Bouzeghoub M., Data Currency Quality Satisfaction in the Design of a Data Warehouse. Special Issue on Design and Management of Data Warehouses, *Intl. J. Cooperative Inf. Syst.*, 10(3): 299-326, 2001.

TIPS Documentation, Quality Control Tools User Requirements, V-Framework Programme IST-1999-10419, 1999. http://tips.sissa.it

Vassiliadis P., Data Warehouse Modeling and Quality Issues, PhD thesis, Department of Electrical and Computer Engineering, National Technical University of Athens (Greece), 2000.

Vassiliadis P., Bouzeghoub M., Quix C., Towards Quality-Oriented Data Warehouse Usage and Evolution. *Intl. Conf. CAiSE,* p. 164-179, 1999.

Vassiliadis P., Vagena Z., Skiadopoulos S., Karayannidis N., ARKTOS: A Tool For Data Cleaning and Transformation in Data Warehouse Environments. *IEEE Data Eng. Bull*. 23(4): 42-47, 2000.

Wang R., Journey to Data Quality, *Advances in Database Systems*, vol. 23, Kluwer Academic Press, Boston, 2002.

Wang R., A product perspective on Total Data Quality Management, *Com. of the ACM*, 41(2): 58-65, 1998.

Wang R., Kon H. B., Madnick S. E., Data quality requirements analysis and modeling, *Intl. Conf. ICDE*, p. 670-677, 1993.

Wang R., Storey V., Firth C., A framework for analysis of data quality research, *IEEE Transactions on Knowledge and Data Engineering*, 7(4): 670-677, 1995.

Winkler W., Data Cleaning Methods, *Intl. Conf. KDD*, 2003.

Zhu Y., Shasha D., StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. *Intl. Conf. VLDB*, p. 358-369, 2002.

Annexes

Auteurs et références	***Dimensions de la qualité des données***	
(Brodie, 1980)	6 concepts	*Intégrité* *Niveau d'abstraction du modèle conceptuel* *Expressivité sémantique* *Validité par rapport à des données de référence* *Maintenance des données* *Efficacité dans l'utilisation des ressources*
(Delen, Rijsenbrij, 1992)	4 dimensions 21 aspects 40 attributs	*Développement et contrôle du S.I.* *Propriétés statiques de maintenance* *Fonctionnement dynamique* *Importance de l'information : donnée correcte, complète, mise à jour, précise, vérifiable*
(Wang et al. *Programme TDQM* 1993, 1995, 2002)	4 catégories 179 attributs	*Qualité intrinsèque* *Qualité d'accessibilité* *Qualité contextuelle* *Qualité de la représentation*
(Redman 1996, 2002)	- 4 dimensions pour les valeurs - 8 dimensions pour le format de représentation	*- Précision, complétude, actualité, cohérence* *- Donnée appropriée, interprétable, portable, précision du format, flexibilité du format, possibilité de représenter les valeurs nulles, utilisation efficace, cohérence*
(Calabretto, Pinon, Poullet, Richez, 1998)	3 critères pour la qualité de l'information	*Disponibilité, Fiabilité, Adaptabilité*
(TIPS-QCT 1999)	8 descripteurs pour la qualité des documents	***Qualité scientifique**: exactitude, précision, originalité, complétude, actualité, qualité de démonstration, qualité de la liste des références, qualité de la méthodologie; **Lisibilité**: qualité du style d'écriture, qualité de la structure logique, adéquation des illustrations, absence des répétitions, clarté de l'expression des idées; **Public visé**: niveau technique; **Fraîcheur**: date de publication, **Autorité**: réputation de l'auteur, réputation du journal ou de la conférence; **Disponibilité**: longévité, imprimabilité; **Popularité**: nombre des lecteurs, des citations; **Qualité d'identification**: citabilité*

TAB. A1 - *Quelques propositions décrivant la notion de qualité de données*

```
Qwith-query ::= query QWITH declarations ;
query ::= ( query ) | query set-op query | query bool-op query | NOT query
 | query constraintop query | query IN query | COUNT ( query )
 | constant | var | sfw-query | EXISTS var query : query | FORALL var query : query
set-op ::= UNION | INTERSECT | MINUS
bool-op ::= AND | OR
constant ::= integer literal | real literal | quoted string literal | true | false
var ::= IDENTIFIER | $IDENTIFIER | @IDENTIFIER
sfw-query ::= sfw-query WHERE query | SELECT projs-list FROM ranges-list
projs-list ::= projs-list, var | * | var
ranges-list ::= ranges-list, one-range | one-range
one-range ::= var query | path-expr
path-expression ::= path-elem.path-expression | path-elem
path-elem ::= [from-to]query | query | path-elem[from-to]
from-to ::= from-to:query | var
declarations ::= declarations declaration ;
declaration ::= contractTypeDeclaration | contractDeclaration | profileDeclaration
contractTypeDeclaration ::= TYPE IDENTIFIER = contractType
contractType ::= CONTRACT { dimensions } ;
dimensions ::= dimensions dimension | dimension
dimension ::= dimName : dimType ;
dimName ::= IDENTIFIER
dimType ::= dimSort | dimSort unit
items ::= items , IDENTIFIER | IDENTIFIER
dimSort ::= ENUM { items } | relSem ENUM { items } WITH order | SET { items }
              | relSem SET { items } | relSem SET { items } WITH order | relSem NUMERIC
relations ::= relations , relation | relation
relation ::= IDENTIFIER < IDENTIFIER
order ::= ORDER { relations }
unit ::= unit / base-unit | base-unit
base-unit ::= % | IDENTIFIER
relSem ::= DECREASING | INCREASING
contractDeclaration ::= IDENTIFIER = contractExpression
contractExpression ::= IDENTIFIER contractDefinition | IDENTIFIER REFINED BY {constraints }
contractDefinition ::= CONTRACT { constraints }
constraints ::= constraints constraint | constraint ;
constraint ::= dimName constraintOp dimValue | dimName { aspects }
constraintOp ::= == | >= | <= | > | < | LIKE | !=
dimValue ::= literal unit | literal
literal ::= IDENTIFIER | { items } | NUMBER
aspects ::= aspects aspect;
aspect ::= PERCENTILE NUMBER constraintOp dimValue | MEAN constraintOp dimValue
        | VARIANCE constraintOp dimValue | FREQUENCY freqRange constraintOp NUMBER %
freqRange ::= dimValue | lRangeLimit dimValue , dimValue rRangeLimit
lRangeLimit ::= [ | (
rRangeLimit ::= ] | )
profileDeclaration ::= IDENTIFIER FOR IDENTIFIER = profileExpression
profileExpression ::= profile | IDENTIFIER REFINED BY { requisites }
profile ::= PROFILE { requisites }
requisites ::= requisites requisite ;
requisite ::= REQUIRE contractList | FROM entityList REQUIRE contractList
contractList ::= contractList , contractElem | contractElem
contractElem ::= IDENTIFIER | contractExpression
entityList ::= entityList , entity | entity
entity ::= IDENTIFIER | IDENTIFIER . IDENTIFIER | RESULT OF IDENTIFIER
```

Tableau A2 – *Syntaxe de XQuaL*