Intégration holistique et entreposage automatique des données ouvertes

Imen Megdiche

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *10/12/2015* **par :**

IMEN MEGDICHE EP BOUSARSAR

Intégration holistique et entreposage automatique des données ouvertes

---

JURY

JACKY AKOKA	Professeur, CNAM	Examinateur
ALAIN BERRO	Maître de conférence, Université Toulouse 1	Co-directeur
JÉRÔME DARMONT	Professeur, Université Lyon 2	Rapporteur
BERNARD ESPINASSE	Professeur, Université Aix-Marseille	Rapporteur
FRANÇOIS PINET	Directeur de Recherche, Irstea Clermont-Ferrand	Président de jury
FRANCK RAVAT	Professeur, Université Toulouse 1	Examinateur
OLIVIER TESTE	Professeur, Université Toulouse 2	Directeur
GILLES ZURFLUH	Professeur, Université Toulouse 1	Examinateur

École doctorale et spécialité :
MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)

Directeur(s) de Thèse :
Olivier TESTE et *Alain BERRO*

Rapporteurs :
Jérôme DARMONT et *Bernard ESPINASSE*

Résumé

L'émergence de nombreuses sources de données ouvertes (Open Data) a encouragé la communauté scientifique ainsi que les entreprises à développer des outils permettant leur exploitation. En effet, les données statistiques présentes dans les Open Data constituent très souvent des informations précieuses dans un système d'aide à la décision. L'intégration de ces données dans un entrepôt, qui constitue l'espace de stockage d'un système décisionnel, se fait à travers des processus d'Extraction, Transformation et Loading (ETL). Ceux-ci demandent une expertise et s'avèrent également chronophage ce qui ralentit la mise en place d'un entrepôt de données. A l'ère de l'information décisionnelle ouverte (Open BI ou self-service BI), les utilisateurs souhaitent de plus en plus intégrer et analyser eux-mêmes les données sans l'aide d'experts. Les processus ETL classiques sont ainsi remis en cause.

Pour intégrer les données ouvertes, les processus ETL font face à plusieurs problèmes :
- Les données ouvertes sont très hétérogènes structurellement et sont très souvent présentées sous forme tabulaire, représentation visuelle très utilisée.
- Les sources tabulaires n'ont pas de schémas, ce qui remet en cause l'approche classique des ETL où le schéma des sources est toujours disponible.
- Les données ouvertes sont rarement significatives individuellement ; il est généralement plus intéressant de croiser plusieurs sources.
- Les données ouvertes sont dispersées et proviennent de plusieurs fournisseurs, ce qui aboutit à une forte hétérogénéité sémantique en particulier dans les vocabulaires utilisés.

Pour répondre à ces problématiques, nous proposons une démarche ETL permettant d'automatiser le plus possible l'entreposage des données ouvertes tabulaires. Cette démarche comprend trois étapes basées sur une représentation commune des données en graphes.

La première étape permet de découvrir le contenu des sources tabulaires et l'extraction de leurs schémas. Nous définissons un modèle de représentation des données tabulaires sur lequel nous nous appuyons pour la détection et l'annotation automatique des composants. Nous nous sommes également focalisés sur la découverte de relations hiérarchiques entre les données pour faciliter l'obtention de hiérarchies dans le schéma multidimensionnel de l'entrepôt. Nos propositions permettent de remédier au problème d'hétérogénéité structurelle et d'absence de schéma grâce à un modèle de tableau commun et générique. A l'issue de cette étape chaque source de données ouvertes est modélisée par un graphe annoté.

La deuxième étape consiste à intégrer simultanément et automatiquement plusieurs graphes. Cette intégration simultanée, appelée intégration holistique, automatise la phase de transformation des données dans le processus ETL. Nous proposons une nouvelle modélisation, sous la forme d'un programme linéaire, qui permet d'inférer plusieurs contraintes

sur la structure des graphes et sur le type de correspondances recherchées. Nous répondons au problème d'hétérogénéité sémantique en combinant plusieurs mesures de similarité. Notre modèle met l'accent sur la structure hiérarchique des graphes intégrés afin de préparer et faciliter la découverte de schémas multidimensionnels de l'entrepôt.

La troisième étape permet de définir le schéma multidimensionnel pour l'alimentation d'un entrepôt de données. Parallèlement le graphe intégré est augmenté par des annotations multidimensionnelles.

Pour valider nos propositions, nous avons développé un prototype couvrant chaque étape et nous avons évalué expérimentalement l'efficacité de ces propositions. La détection du contenu des tableaux a été évaluée sur des données ouvertes disponibles sur data.gouv.fr et la proposition d'intégration holistique a été évaluée sur la qualité des correspondances en l'appliquant sur deux bancs d'essais de référence [Melnik *et al.*, 2002] [Duchateau et Bellahsene, 2014].

Abstract

THE emergence of several Open Data, rich in information, urges the scientific community as well as corporates to develop tools allowing their exploitation. The statistics present into tabular Open Data are very useful for decision support systems (DSS). Their integration in a data warehouse, which is the storage space in DSS, is achieved through Extraction, Transformation and Loading (ETL) processes. These later require an expertise and turn out expensive, what slows down the implementation of data warehouses. In the ear of open business intelligence (Open BI or Self-service BI), users expect to integrate and analyse themselves data without experts assistance, hence classical ETL processes are called into question.

ETL processes have to deal with several problems spanned by the integration of tabular Open Data :

- Open Data are structurally heterogeneous and they are often presented in tables.
- Tabular Open Data lack schemes, which shakes classical ETL processes where schemes are always available.
- Open Data are rarely significant individually so it is more interesting to cross several sources.
- Open Data are scattered over several suppliers leading to a highly semantic heterogeneity.

To meet these issues, we propose a new ETL approach automating as much as possible the warehousing of tabular Open Data. This approach encompasses three steps based on a common representation of data in graphs.

The first step is about discovering the contents of tabular sources and the extraction of their schemes. We define a table model which supports several automatic activities detecting the table components. We also focused on the discovery of hierarchical relationships between the data in order to prepare hierarchies of the multidimensional schema of the data warehouse. The results of the detection activities are transformed into a graph. Our propositions lead to a homogenous and common representation of data which resolves the problems of structural heterogeneity and lack of schema.

The second step consists of integrating simultaneously and automatically several graphs. This is known as holistic integration which is able to automate the transformation phase of the ETL process. We propose a new linear program encompassing different constraints on the graph structure and the tuning of correspondences. This model emphasizes the hierarchical structure of the integrated graphs in order to facilitate the discovery of the multidimensional schema. We combine also several similarity measures to face out the problem of semantic heterogeneity. The third step is devoted to the definition of the multidimensional schema of the data warehouse. At the same time, the integrated graph is increased by

multidimensional annotations.

In order to validate our proposals, we have implemented a prototype covering every step and we have evaluated them experimentally. The detection of the tables'content was evaluated on Open Data available on data.gouv.fr. The holistic integration was experimented on two benchmarks [Melnik *et al.*, 2002] [Duchateau et Bellahsene, 2014] to evaluate the quality of correspondences. It was also evaluated on the performance of the resolution time.

A la mémoire de ma mère Lilia,
A mes deux trésors Yessine et Salma,

Remerciements

UN vrai challenge, c'est ainsi comment je résume cette expérience. Un challenge qui a aboutit grâce à beaucoup d'amour, de soutien, d'encouragements de la part de mes directeurs, de mes amis et surtout de mon mari et de ma famille. Je suis profondément reconnaissante à tous ceux qui m'ont entouré. Je suis également fière d'avoir pu relevé ce challenge.

Mes remerciements vont d'abord à mon directeur de thèse Mr. Olivier Teste, professeur à l'université Toulouse 2, pour la qualité de son encadrement, pour sa disponibilité, pour sa rigueur scientifique et pour la confiance qu'il a accordé à ces travaux de thèse. Soyez rassuré de ma plus grande gratitude envers tout ce que vous m'avez apporté et appris scientifiquement et pédagogiquement. Je vous remercie également pour votre bonne humeur, votre gaieté naturelle, votre soutien et vos encouragements sans limites qui m'ont toujours poussé à faire les choses au mieux et qui m'ont beaucoup aidé à surmonter les difficultés.

Je tiens à remercier également mon co-directeur de thèse Mr. Alain Berro, maître de conférence à l'université Toulouse 1, pour la qualité de son encadrement, sa disponibilité et son aide précieuse à plusieurs reprises. Mais aussi pour votre bonne humeur, vos qualités d'écoute et votre grande patience. Soyez rassuré de ma très grande reconnaissance. Je suis tout particulièrement touchée par votre gentillesse en me procurant un point de chute à côté des "vortexs".

J'exprime mes remerciements à mes deux rapporteurs, Mr. Jérôme Darmont, professeur à l'université Lyon 2 et Mr. Bernard Espinasse, professeur à l'université Aix-Marseille, d'avoir accepté d'évaluer mes travaux de thèse. Vous me faites l'honneur d'être membres de mon jury.

Je tiens à remercier Mr. Franck Ravat, professeur à l'université Toulouse 1, d'avoir accepté d'être membre de mon jury de thèse. Je le remercie vivement pour ses lectures et ses remarques très pertinentes qui ont contribué à améliorer la qualité de ce mémoire.

Je tiens à remercier également Mr. François Pinet, directeur de recherche à l'IRSTEA, Mr. Jacky Akoka, professeur au CNAM et Mr. Gilles Zurfluh, professeur à l'université Toulouse 1, pour avoir accepté d'être examinateurs de cette thèse. Vous me faites l'honneur d'être membres de mon jury.

Je remercie Mme. Josiane Mothe responsable de l'équipe SIG de m'avoir accueilli au sein de cette grande "famille". J'apprécie beaucoup l'ambiance chaleureuse qui règne que ce soit entre permanents ou non-permanents. Un grand merci va de soi à mes collègues sig : Mohammed, Bilel, Rafik, Baptiste, Sirinya, Liana, Manel, Arlind, Thomas, Eya, Mariem, Ameni,... J'ai passé avec vous de très bons moments.

Je remercie mon père qui m'a toujours poussé depuis toute petite à persévérer dans ce que je fais. Merci papa pour ton amour et ton soutien sans failles !

Je remercie la famille Megdiche notamment ma grande-mère, mes oncles et mes tantes qui m'ont énormément soutenu et aidé sur plusieurs années.

Je remercie mon beau-père et ma belle-mère pour leur amour, leur bonne humeur et leur aide très précieuse depuis que je suis parmi eux.

Je garde le meilleur pour la fin, je remercie mon compagnon de route, mon mari Anouar, pour son amour, son soutien, ses conseils et ses encouragements. C'est grâce à un grand engagement de ta part que j'ai pu faire un doctorat. Mais aussi grâce à ta très grande patience et amour qu'on est entrain de faire pousser nos deux fleurs Yessine et Salma.

Table des matières

Liste des figures

Liste des tableaux

I Cadre d'étude et approche globale

L'émergence de nombreuses sources de données ouvertes (Open Data) pousse plusieurs communautés de recherche ainsi que les entreprises à développer des outils permettant leur exploitation. Les données ouvertes présentées sous la forme de tableaux sont particulièrement intéressantes dans le cadre d'analyses décisionnelles [Chaudhuri *et al.*, 2011]. Elles contiennent en effet des statistiques qu'il est intéressant d'exploiter notamment dans les systèmes d'information décisionnels (SID). Ces outils d'analyse nécessitent une vue intégrée des données, généralement matérialisée au sein d'un entrepôt de données sur lequel sont appliqués des processus d'analyses en ligne (OLAP). L'intégration des données se fait par des processus d'Extraction-Transformation-Chargement (ETL) [Vassiliadis, 2009]. Toutefois, les processus ETL actuels s'avèrent inadaptés aux données ouvertes. Ces processus nécessitent des schémas représentatifs des données sources tandis que les données ouvertes tabulaires ne disposent généralement pas de schémas. Les processus ETL ont souvent besoin d'un schéma global pour intégrer les différentes sources [Abello *et al.*, 2015]. Ce schéma nécessite une expertise et est particulièrement difficile à concevoir à partir des données ouvertes largement dispersées sur le web. Cette dispersion entraine notamment une forte variabilité dans les données. Enfin, les processus ETL sont souvent spécifiques, manquant de généricité et leur utilisation est manuelle. En particulier la définition des processus d'intégration des données sources est réalisée manuellement, attribut par attribut.

Nous proposons dans cette thèse une démarche ETL pour l'intégration des données ouvertes tabulaires dans les systèmes d'information décisionnels. L'originalité de notre démarche réside dans : (1) l'automatisation de l'extraction des schémas des tableaux et (2) l'automatisation de l'intégration de différents schémas sans utiliser un schéma global. Notre démarche repose sur une formalisation à base de graphes n'étant rattachés à aucun formalisme. La simplicité de ces graphes assure une meilleure généricité de notre approche. A notre connaissance, notre démarche est la première qui exploite la réutilisation des données ouvertes tabulaires dans les systèmes décisionnels.

Dans ce chapitre introductif, nous définissons tout d'abord le cadre d'étude de nos travaux qui se situe au croisement entre les données ouvertes et les systèmes d'informations décisionnels. Nous décrivons ensuite globalement notre approche d'entreposage automatique des données ouvertes.

1 Cadre d'étude

1.1 Les données ouvertes tabulaires

Les données ouvertes (Open Data) sont des données produites par les organismes publics, disponibles sous licence libre et destinées à la réutilisation et à la redistribution par n'importe quelle personne [Mazón *et al.*, 2012] [Coletta *et al.*, 2012] [Eberius *et al.*, 2012]. Ces données sont très souvent encodées dans des formats tels que CSV, XML ou HTML [Mazón *et al.*, 2012].

Tim Berners-Lee, fondateur du web sémantique, a attribué des étoiles aux différents types de données ouvertes disponibles sur le web, voir Figure I.1. Les données ouvertes disponibles en n'importe quel format (images, pdf, HTML etc) et qui ne peuvent pas être lues par des machines sont des données ouvertes à 1 étoile. Les données ouvertes 1 étoile disponibles en format propriétaire comme Excel et qui peuvent être lues par les machines sont des données ouvertes à 2 étoiles. La troisième étoile est attribuée aux données qui sont dans un format non propriétaire tel que CSV. La quatrième étoile concerne les données ouvertes décrites par le standard RDF [1]. La cinquième étoile est réservée aux données RDF liées à d'autres données RDF constituant les données ouvertes liées ou le web de données liées [Bizer *et al.*, 2009].

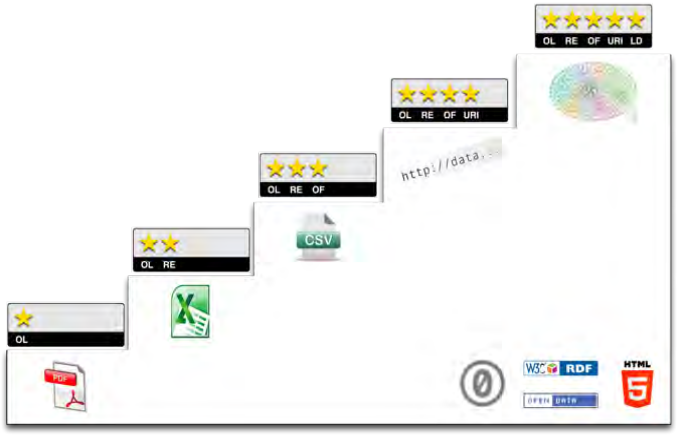


Figure I.1 — Les étoiles des données ouvertes

Les données ouvertes tabulaires sont des données ouvertes 2/3 étoiles dans les formats Excel ou CSV contenant un ou plusieurs tableaux. Un tableau est une forme de communication omniprésente [Embley *et al.*, 2006]. Ceci explique la quantité importante de données ouvertes tabulaires publiées par les gouvernements d'après la récente étude de Qunb [2], en 2012. D'après cette étude, le nombre de données ouvertes tabulaires est de 85% en France, 65% en Amérique et 66% en Angleterre.

Les données ouvertes tabulaires sont difficiles à exploiter et à acquérir vu les caractéristiques suivantes :

1. https ://www.w3.org/RDF/
2. http ://fr.slideshare.net/cvincey/opendata-benchmark-fr-vs-uk-vs-us

– *Absence de schémas*. Il s'agit de tableaux qui peuvent être lus par des machines, néanmoins seuls les utilisateurs ont la capacité d'analyser l'organisation compacte des données dans un tableau. Afin de permettre à la machine d'exploiter les données il est nécessaire d'identifier un schéma représentatif (attributs) du contenu (valeurs) du tableau.
– *Hétérogénéité structurelle*. L'organisation des données tabulaires est très variable. Elle est influencée par la personne ou le groupe de personnes qui l'ont produite. L'origine de ces tableaux, notamment s'ils proviennent de différents organismes publics, induit une hétérogénéité dans les niveaux de détails des données et dans la présentation de la structure des tableaux.
– *Hétérogénéité sémantique*. Les données ouvertes tabulaires utilisent des concepts issus des systèmes d'informations des producteurs de ces données. Il est quasi systématique que ces données soient hétérogènes sémantiquement puisque les producteurs de données n'utilisent pas les mêmes vocabulaires pour décrire l'information.
– *Données imparfaites en termes de qualité*. En effet, nous retrouvons des données manquantes, soit en raison d'un oubli humain ou de la non disponibilité de la donnée, soit en raison d'une limitation sur les données ouvertes puisque les producteurs peuvent masquer certaines données. Des données erronées peuvent également être présentes.

Dans la littérature, nous pouvons clairement constater que la majorité des travaux se focalisent sur la production de données ouvertes liées notamment au travers du projet LOD (Linked Open Data) [Ferrara *et al.*, 2011][3]. Diverses problématiques sont étudiées sur les données ouvertes liées : l'interopérabilité [Governatori *et al.*, 2014], l'exploitation analytique [Colazzo *et al.*, 2014], la visualisation [Tschinkel *et al.*, 2014]. Ces travaux supposent que la représentation des données en RDF est disponible [Abello *et al.*, 2015]. Or, la grande majorité des données ouvertes tabulaires est à ce jour sans encodage RDF.

Nos travaux de thèse traitent le problème difficile d'exploiter les données ouvertes tabulaires quelconques, hétérogènes structurellement et sémantiquement, sans schémas et pauvres en méta-données. En particulier, nous souhaitons les intégrer dans un système d'information décisionnel afin de les rendre accessibles aux outils d'analyse OLAP.

1.2 Les systèmes d'aide à la décision

Un système d'aide à la décision se présente comme étant un ensemble de techniques et d'outils permettant la collecte, l'extraction, le stockage et l'analyse de données. Les systèmes d'aide à la décision supportent les processus de prise de décision des organisations, leur permettant ainsi d'induire de l'intelligence dans leur métier, communément connu sous le terme *Business Intelligence*.

L'architecture typique d'un système décisionnel est illustrée par la Figure I.2. Ce système collecte des données issues de différentes sources via des processus d'extraction, de transformation et de chargement dits processus ETL (Extract-Transform-Load). Ces données sont stockées dans un entrepôt de données [Teste, 2000]. Enfin, des outils d'analyse en ligne (OLAP) sont appliqués sur un cube de données multidimensionnel. Les analyses fournissent

3. Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak.`http://lod-cloud.net/`

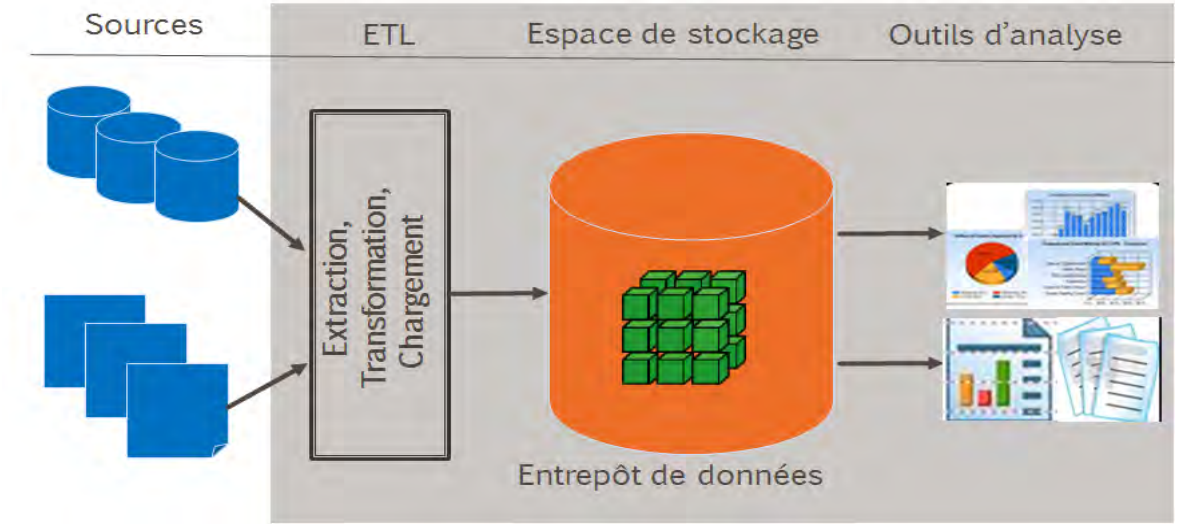


Figure I.2 — Architecture d'un système décisionnel pour les données d'organisation

aux décideurs une vision synthétique et facilement compréhensible des données.

Définitions

- *Un entrepôt de données (ED)* est une collection de données orientées sujets, intégrées, variant selon le temps et non volatiles, qui sert de support au processus de prise de décision [Inmon, 1992]. L'entrepôt de données contient une copie des données transactionnelles [Kimball, 1996] structurées en cube multidimensionnel. Du point de vue conceptuel, une modélisation en cube correspond à un modèle en étoile [Kimball et Ross, 2002], ou des modèles étendus dans la littérature [Teste, 2009]. Dans le modèle étoile, les données sont organisées selon la vision fait-dimension. Un fait est un ensemble de données numériques représentant un sujet d'analyse observable depuis différents axes d'analyses, appelés dimensions. Du point de vue logique, les entrepôts de données sont hébergés dans une base de données multidimensionnelles le plus souvent "Relationnelle-OLAP" (ROLAP) [Chaudhuri et Dayal, 1997].
- *Un processus ETL* est un enchaînement de programmes servant au traitement et à l'homogénéisation des données des sources afin d'alimenter un entrepôt de données. Selon [Vassiliadis et Simitsis, 2009], la description d'un processus ETL se résume dans les étapes suivantes :

  1. Les données sont extraites des sources qui peuvent être structurées (relationnelles) ou non-structurées (des pages web, des fichiers tabulaires, des flux de données, etc..). Cette étape est nommée l'*Extraction*.
  2. Les données sont propagées dans un espace de stockage temporaire appelé Data Staging Area dans lequel des opérations de transformation, d'homogénéisation, de nettoyage et de filtrage sont mises en place. L'objectif est de définir les règles de transformation entre un schéma global et les schémas locaux des sources. Cette étape est nommée *Transformation*.
  3. Les données sont chargées dans l'entrepôt de données ; on parle de matérialisation des données. Cette étape est nommée *Chargement*.

L'essor des données du web entre autres les données ouvertes a fait émerger de nouvelles définitions dans l'informatique décisionnelle : *OpenBI* [Mazón *et al.*, 2012] [Schneider *et al.*, 2011], *Self-service BI* [Abello *et al.*, 2013] et ETQ [Abello *et al.*, 2015].

- L'OpenBI ou le Self-service BI a pour objet de permettre à des utilisateurs non-experts d'exploiter eux-mêmes des données [Mazón *et al.*, 2012] [Schneider *et al.*, 2011].
- L'ETQ (Extract-Transform-Query) [Abello *et al.*, 2015] constitue une évolution de l'ETL. Dans un processus ETQ, la phase de chargement n'est plus indispensable ; les données issues de la phase de transformation peuvent être directement interrogées par des requêtes SPARQL-OLAP [Saad *et al.*, 2013] [Azirani *et al.*, 2015].

Conception d'un entrepôt de données

La conception d'un entrepôt de données a fait l'objet de nombreux travaux [Romero et Abelló, 2009]. Ces travaux peuvent être décomposés en deux catégories [Khouri, 2013] :

- La première catégorie considère un entrepôt de données comme un système d'intégration de sources de données par le biais des processus ETL [Vassiliadis, 2009] [Vassiliadis et Simitsis, 2009]. Les travaux de cette catégorie sont focalisés soit sur la modélisation des transformations entre un schéma global et les schémas des sources [Bergamaschi *et al.*, 2011] [Khouri *et al.*, 2013], soit sur la modélisation des processus ETL [Khouri, 2013] [Atigui *et al.*, 2012].
- La deuxième catégorie se concentre sur l'analyse des besoins des utilisateurs et la modélisation conceptuelle [Golfarelli *et al.*, 1998]. Les travaux de cette catégorie sont notamment axés sur la modélisation multidimensionnelle. La plupart des travaux [Romero et Abelló, 2009] traitent des sources de données structurées (UML, entité/association, etc). En contrepartie, très peu de travaux [Romero et Abelló, 2007] [Nebot *et al.*, 2009] ont exploité la conception d'un schéma multidimensionnel à partir de sources semi-structurées [Ravat *et al.*, 2007a] voire non-structurées.

Notre objectif est d'intégrer les données ouvertes tabulaires dans un système d'information décisionnel afin de les rendre accessibles aux outils d'analyse OLAP. Pour cela, nous aurions besoin de processus ETL pour l'intégration de ces données. Nous aurions également besoin de définir le schéma multidimensionnel pour pouvoir appliquer les analyses OLAP.

Les processus ETL actuels s'avèrent inadaptés aux données ouvertes. Ces processus nécessitent des schémas représentatifs des données sources tandis que les données ouvertes tabulaires ne disposent généralement pas de schémas. Les processus ETL ont souvent besoin d'un schéma global pour intégrer les différentes sources [Abello *et al.*, 2015]. Ce schéma nécessite une expertise et est particulièrement difficile à concevoir à partir des données ouvertes largement dispersées sur le web. De plus, les approches ETL actuels supposent que les correspondances entre le schéma global et les schémas des sources sont disponibles [Abello *et al.*, 2015]. Or, les correspondances entre des données ouvertes disséminées sur le web ne sont pas disponibles. La recherche de ces correspondances relève d'un problème difficile dans la littérature connu sous le nom de problème d'appariement [Euzenat et Shvaiko, 2013]. Enfin, les processus ETL sont souvent spécifiques, manquant de généricité et leur utilisation est manuelle. L'automatisation des processus ETL reste un verrou à résoudre [Laborie *et al.*, 2015].

Par ailleurs, les travaux de conception d'un schéma multidimensionnel à partir de données semi-structurées manquent de maturité [Abello *et al.*, 2015]. Ceci est dû à la complexité des sources traitées. Les approches actuelles nécessitent des experts pour pouvoir résoudre ce problème. Or, avec la nouvelle génération d'informatique décisionnelle dédiée à des utilisateurs non-experts il faudrait simplifier le plus possible cette tâche afin de la rendre plus accessible.

2 Problématiques

Le cadre d'étude dans lequel se positionnent nos travaux pose différents problèmes :

- Nous nous heurtons à la complexité des données ouvertes tabulaires qui n'ont pas de schémas, qui sont hétérogènes structurellement et sémantiquement et qui sont imparfaites en terme de qualité.
- Les processus ETL classiques sont inadaptés : (1) ils exigent en pré-requis un schéma global et les correspondances entre le schéma global et les schémas des sources ; (2) ils sont souvent spécifiques et non-automatisés.
- La conception d'un schéma multidimensionnel est souvent réalisée par des experts. A l'ère de la nouvelle génération de systèmes décisionnels, elle doit être simplifiée pour la rendre accessible à des non-experts.

L'objectif de nos travaux de thèse est d' : **intégrer automatiquement des données ouvertes tabulaires dans un système d'information décisionnel de type self-sevice BI.**

Nous ambitionnons d'automatiser et de simplifier le plus de tâches possible dans le processus ETL. Nous prenons aussi en considération l'enjeu de rendre notre solution réutilisable.

3 Une approche ETL basée sur les graphes

Nous proposons une nouvelle approche ETL pour l'entreposage des données ouvertes tabulaires. A notre connaissance, cette approche est la première qui exploite la réutilisation des données ouvertes tabulaires dans les systèmes décisionnels. Notre approche repose sur une formalisation à base de graphes n'étant rattachés à aucun formalisme particulier. La simplicité de ces graphes assure une meilleure généricité de notre approche.

Notre approche, illustrée dans la Figure I.3, est composée de trois étapes :

1. *Détection et reconnaissance*. Cette première étape permet de transformer automatiquement des données ouvertes tabulaires en graphes dans lesquels nous distinguons clairement les données de structures des valeurs (données statistiques). Nous proposons un ensemble d'activités pour la détection des composants d'un tableau suivant un modèle représentatif de l'anatomie d'un tableau. Parmi les activités proposées, nous mettons l'accent sur la découverte de relations hiérarchiques entre les données structurelles. Les résultats des différentes activités sont organisés dans un graphe de propriétés (qui pourrait être facilement étendu à un graphe RDF). Dans cette étape, nos propositions permettent de pallier le problème d'hétérogénéité structurelle et l'absence de schéma

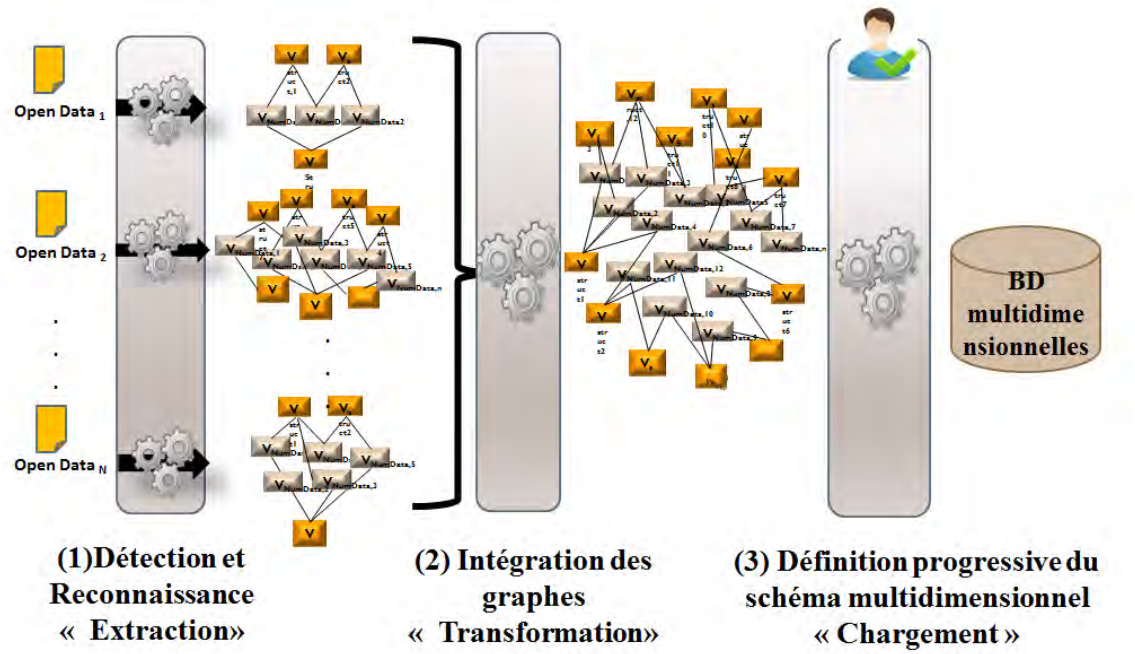


Figure I.3 — Une approche ETL basée sur les graphes

grâce à un modèle de tableau commun et générique. Cette phase résout le problème de manque de schémas nécessaires pour l'intégration.

2. *Intégration des graphes*. Cette deuxième étape intègre simultanément différentes données structurelles provenant de plusieurs graphes. Cette intégration simultanée de plusieurs graphes s'appelle une intégration holistique. Nous proposons un programme linéaire qui permet une généralisation de l'appariement des graphes par paire à de multiples graphes. Nous avons choisi la technique de programmation linéaire puisqu'il a été prouvé, dans le domaine de l'optimisation combinatoire, que le problème de couplage (un problème similaire au problème d'appariement) se résout en temps polynomial avec cette technique [Almohamad et Duffuaa, 1993]. Nous avons également choisi cette technique puisqu'elle retourne automatiquement une solution unique et globalement meilleure dans l'espace des solutions. Notre programme linéaire repose sur un modèle à base de contraintes notamment sur la structure hiérarchique des graphes intégrés afin de préparer et faciliter la découverte de schémas multidimensionnels. Nous faisons face au problème d'hétérogénéité sémantique en combinant plusieurs mesures de similarité dans le programme linéaire. Nous garantissons également avec cette étape la possibilité d'intégrer n'importe quelle combinaison de sources de données ouvertes sans besoin de définir un schéma global. Notre proposition permet d'automatiser la phase de transformation des données dans le processus ETL.

3. *Définition progressive du schéma multidimensionnel*. Cette dernière étape est principalement destinée à la définition de schémas multidimensionnels pour l'alimentation d'un entrepôt de données ROLAP. Parallèlement, le graphe intégré est augmenté par des annotations multidimensionnelles. Cette étape est semi-automatique : l'utilisateur conçoit progressivement à partir du graphe intégré le schéma multidimensionnel à un niveau

conceptuel. Le graphe intégré est directement transformé par interactions successives pour obtenir une vision conceptuelle du schéma multidimensionnel. L'utilisation de graphes n'étant rattachés à aucun formalisme en plus de la préparation de structures hiérarchiques dans les deux phases précédentes simplifient aux non-experts cette dernière étape. De plus, l'augmentation du graphe intégré par des annotations multidimensionnelles permet d'interroger directement les données sans besoin de les matérialiser conformément à l'approche ETQ.

4 Organisation du manuscrit

Ce manuscrit s'articule en trois chapitres de propositions qui correspondent aux trois étapes de notre approche d'entreposage, et un chapitre de validation (prototypage) et d'évaluations. Dans les trois premiers chapitres, nous résumons d'abord les travaux pertinents du domaine étudié puis nous présentons notre contribution. Le chapitre validation illustre à travers une étude de cas les différents modules développés ainsi que plusieurs résultats des évaluations de notre approche.

Le **Chapitre II** présente notre contribution pour la détection et la reconnaissance de tableaux dans les données ouvertes. Nous définissons un modèle de représentation des tableaux. Ce modèle est utilisé pour définir sept activités de détection et d'annotation des tableaux contenus dans les données ouvertes. Ce processus génère des graphes annotés.

Le **Chapitre III** présente notre contribution pour l'intégration holistique de plusieurs graphes. Nous détaillons notre proposition centrée sur un programme linéaire composé de contraintes dédiées à la structure des graphes et de contraintes dédiées au problème d'appariement.

Le **Chapitre IV** présente notre contribution pour la conception d'un schéma multidimensionnel à partir d'un graphe intégré. Un processus progressif pour la définition des dimensions et des faits est décrit. Il est complété par un processus de génération de données RDF.

Le **Chapitre V** détaille la validation expérimentale de notre approche ETL. Nous donnons un aperçu sur les modules implémentés. Puis nous analysons les résultats d'évaluations de notre proposition sur la détection des tableaux et sur l'intégration des graphes. La phase d'intégration est expérimentée sur deux bancs d'essai de référence dans la littérature scientifique du domaine.

II Détection et reconnaissance du contenu des données ouvertes

LES tableaux sont des sources d'informations riches en statistiques et universellement utilisés. Ces sources majoritairement présentes dans les données ouvertes [Coletta *et al.*, 2012] sont notre cible d'étude. En particulier, nous nous intéressons à la détection et à la reconnaissance automatique des données ouvertes tabulaires afin de produire les schémas nécessaires pour leur intégration dans un système d'information décisionnel. La reconnaissance des données ouvertes tabulaires est un défi important en raison de l'hétérogénéité structurelle et sémantique des données.

Ce chapitre présente le domaine de recherche concernant la détection et la reconnaissance des tableaux et les approches proposées dans la littérature. Nous identifions leurs limites puis nous présentons nos propositions.

1 Introduction

Les tableaux intéressent les scientifiques depuis plusieurs années pour la richesse de leur contenu. Ces tableaux se résument en un ensemble de données structurelles et d'informations relationnelles disposées dans un espace à deux dimensions [Liu *et al.*, 2007]. Ils sont omniprésents dans les documents scientifiques, les rapports financiers des entreprises, les pages web, les librairies digitales, etc. L'organisation variable des données tabulaires dans ces divers supports engendre des problèmes d'hétérogénéité structurelle. De plus, la diversité des domaines et l'utilisation du langage naturel non-standardisé induisent une hétérogénéité sémantique des données. Ces deux types de problèmes d'hétérogénéité rendent difficile la détection et la reconnaissance automatique des tableaux. La détection des tableaux a pour objectif de repérer l'emplacement du tableau et de segmenter ses différents composants [Zanibbi *et al.*, 2004]. La reconnaissance permet d'analyser le contenu du tableau détecté et d'expliciter les relations entre les composants de ce dernier.

Plusieurs domaines d'application sont concernés par la détection et la reconnaissance des tableaux [Embley *et al.*, 2006] :

- L'intégration des tableaux ; à titre d'exemple [Embley *et al.*, 2006] évoque la création individuelle de base de données à partir de tableaux provenant de différentes sources telles que les e-mail, le web, etc.
- L'interrogation et la navigation dans les données du tableau ; à titre d'exemple si un tableau est transformé en un modèle relationnel alors il peut être interrogé en SQL [Embley *et al.*, 2006] tandis que s'il est transformé en RDF il peut être interrogé en SPARQL [Scharffe *et al.*, 2012].
- La manipulation des tableaux ; à titre d'exemple le nettoyage et le formatage (le projet

OpenRefine [1]) ou la fusion des tableaux (le projet Google Table Fusion [2]).
- L'extraction d'informations en utilisant des ressources externes comme des ontologies ou des schémas existants ; à titre d'exemple le projet @web [Hignette, 2007].
- L'apprentissage d'ontologies à partir des tableaux ; à titre d'exemple le projet TANGO [Tijerino *et al.*, 2005].

Selon un chiffre publié par l'organisme Qunb en 2012 [3], les données ouvertes tabulaires (en format XLS ou CSV) représentent plus de la moitié des données ouvertes gouvernementales disponibles sur le web. Ces sources sont très prometteuses pour les systèmes d'information décisionnels puisqu'elles contiennent des informations riches en statistiques. Il est alors possible d'envisager une variété de scénarii d'analyse à partir de ces données.

Notre objectif, dans ce chapitre, est de proposer une approche de détection et de reconnaissance automatique des données ouvertes tabulaires afin de pouvoir générer des schémas permettant leur intégration dans les systèmes d'information décisionnels.

Ce chapitre sera découpé en deux sections. Dans la première section, nous présentons une étude des travaux de la littérature dans le domaine de reconnaissance des tableaux. Dans cette étude, nous décrivons et comparons les différentes approches proposées. Dans la deuxième section, nous abordons en détail notre proposition appliquée aux données ouvertes tabulaires.

2 État de l'art : Détection et reconnaissance des tableaux

La détection et la reconnaissance sont deux étapes complémentaires [Hu *et al.*, 2002] qui rendent le contenu des tableaux accessible, compréhensible et réutilisable. La détection des tableaux a pour objectif de repérer l'emplacement du tableau et de segmenter ses différents composants [Zanibbi *et al.*, 2004]. La reconnaissance permet d'analyser le contenu du tableau détecté et d'expliciter les relations entre les composants de ce dernier. Les résultats de ces deux étapes peuvent être transformés vers un schéma de tableau. Le schéma peut être sous la forme d'un fichier de méta-données (xml) attaché à un fichier des données du tableau. Il peut être aussi sous la forme d'un graphe qui englobe les méta-données et les données.

La détection d'un tableau se base soit sur ses données soit sur un modèle de tableau [Lopresti et Nagy, 2000]. Un modèle décrit l'organisation des composants d'un tableau. Parmi les modèles, proposés dans la littérature, nous pouvons illustrer les principaux composants d'un tableau à travers le modèle de [Wang, 1996]. Ce modèle est l'un des plus complet [Zanibbi *et al.*, 2004] de la littérature.

[Wang, 1996] a distingué la structure logique de la structure physique dans un tableau, correspondant au contenu et à la présentation de ce dernier.
- La structure logique est définie par les composants et leurs relations. Un tableau contient des composants élémentaires qui peuvent être du texte (labels), des nombres, des images, etc. Les **éléments numériques** sont souvent indexés par les **éléments tex-**

1. http ://openrefine.org/
2. https ://developers.google.com/fusiontables/
3. http ://fr.slideshare.net/cvincey/opendata-benchmark-fr-vs-uk-vs-us

tuels représentant la relation logique entre eux.

– La structure physique est définie par un ensemble de règles topologiques déterminant l'emplacement des composants du tableau et par un autre ensemble de règles de style permettant la génération de l'apparence du tableau. Selon les règles topologiques, un tableau contient des lignes et des colonnes. L'intersection entre ligne et colonne est une **cellule**. Une cellule peut appartenir à un **bloc** de cellules (une collection rectangulaire [Wang, 1996] ou un groupe de cellules contiguës [Zanibbi *et al.*, 2004]). Le tableau se divise en quatre régions : 3 régions d'entêtes et le corps. Nous avons l'**entête de colonnes**, l'**entête de lignes** et l'entête des entêtes de lignes. Ces trois types d'entêtes contiennent des éléments textuels. Quant au **corps** du tableau, il contient les éléments numériques. Le corps est une matrice de valeurs indexées par l'entête de lignes et l'entête de colonnes [Zanibbi *et al.*, 2004] ou l'une des deux.

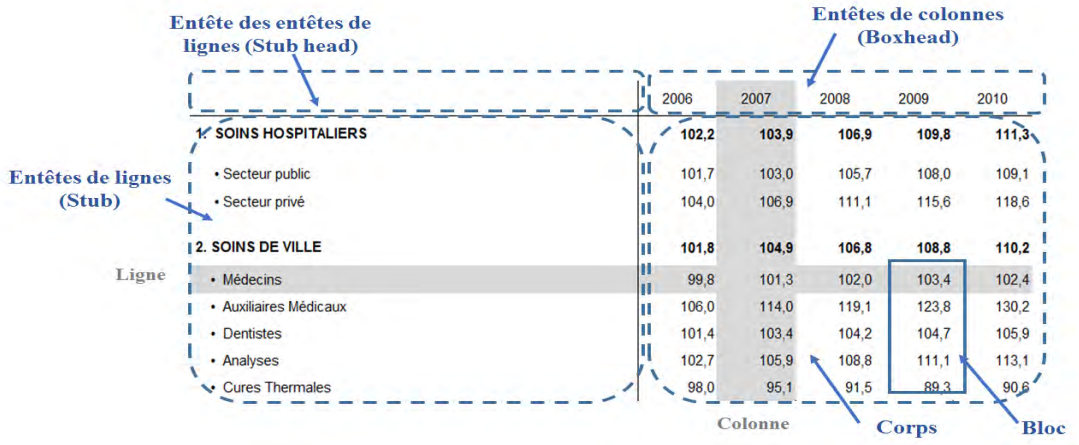


Figure II.1 — Anatomie d'un tableau

Exemple 1. *la FigureII.1 illustre cette terminologie. Il s'agit d'un extrait d'un tableau* [4] *concernant des statistiques sur la consommation des soins et biens médicaux entre 2006 et 2010 en France.*

Pour la structure logique, nous avons par exemple un élément numérique 102,2 qui est indexé par l'élément textuel Soins Hospitaliers de l'entête de lignes et par l'élément textuel 2006 de l'entête de colonnes.

Pour la structure physique, nous avons une entête de lignes qui est composée par les éléments textuels {Soins Hospitaliers, Secteur public, Secteur privé, Soins de ville, Médecins, Auxiliaires Médicaux, Dentistes, Analyses, Cures Thermales} et une entête de colonnes qui est composée par les éléments textuels {2006, 2007, 2008, 2009, 2010}.

Un tableau se caractérise par : (1) son type, (2) la nature de ses données et (3) le support qui le contient (HTML, XLS, CSV, etc.)

Nous distinguons des tableaux de type relationnel (R) ou non-relationel (NR). Ces types sont déterminés en fonction de la présence des entêtes de lignes et de colonnes :

– Les tableaux relationnels sont ceux qui ont soit une entête de colonnes (dits aussi relationnel horizontal), soit une entête de lignes (dits aussi relationnel vertical).

4. https ://www.data.gouv.fr/fr/datasets/comptes-nationaux-de-la-sante-2010-consommation-de-soins-et-de-biens-medicaux-en-volume-bas-30378565/

– Les tableaux non-relationnels sont ceux qui ont des entêtes de lignes et des entêtes de colonnes qui indexent le corps du tableau.

Nous distinguons des tableaux de nature de données statistiques (S) ou de nature de données non-statistiques (NS). La nature est déterminée par le type des données contenues dans le corps du tableau.

– Les tableaux de nature non-statistique ont des corps formés majoritairement par des éléments textuels ou alpha-numériques.
– Les tableaux de nature statistique ont des corps formés exclusivement par des éléments numériques.

Nous étudions dans les sections suivantes les travaux de la littérature dans le domaine de détection et de reconnaissance des tableaux. Nous avons classé ces travaux en trois catégories : les travaux dédiés uniquement à la détection, les travaux dédiés uniquement à la reconnaissance qui supposent que la détection ait déjà été faite et les travaux qui font la détection et la reconnaissance.

2.1 Les travaux de détection des tableaux

Les recherches autour du problème de détection des tableaux ont émergé avec le traitement des images de documents scannés. Avec l'avènement du web, plusieurs travaux se sont orientés vers la détection des tableaux contenus dans les pages web. Nous étudions, dans cette section, quelques travaux dans le domaine de détection des tableaux en s'appuyant sur les critères de comparaison suivants :

– Critères relatifs au tableau :
  – Le type de tableau : Relationnel, Non-relationnel (R/NR).
  – La nature des données : Statistiques, Non-statistiques (S/NS).
  – Le support contenant le tableau.
– Critères relatifs à la détection :
  – Le type de détection : Automatique, Semi-automatique, Manuelle (A/SA/M).
  – Les techniques utilisées.
  – La stratégie de détection adoptée. Elle peut être dirigée par les données et ascendante / dirigée par les données et descendante comme elle peut être dirigée par un modèle et ascendante / dirigée par un modèle et descendante d'après [Lopresti et Nagy, 2000]. Une stratégie est ascendante si elle commence par la détection des composants du tableau et finit par sa détection. Inversement, elle est descendante si elle commence par la détection du tableau et finit par la détection de ses composants. En absence de modèle de tableau, la stratégie est considérée comme étant dirigée par les données.

2.1.1 Étude des travaux

[Laurentini et Viada, 1992] ont étudié la détection des tableaux textuels (NS) dans l'image d'un document scanné. Il s'agit d'une approche descendante dirigée par un modèle logique de tableau. Ce modèle de tableau est composé d'éléments indexés par des entêtes de lignes (relationnel horizontal) ou des entêtes de colonnes (relationnel vertical). Les au-

teurs utilisent la technique de reconnaissance optique de caractères (OCR) pour identifier les données textuelles connectées. Ceci permet de déduire l'emplacement du tableau dans l'image de document. Les données textuelles connectées sont ensuite analysées afin d'identifier les caractères, les mots et les phrases. En parallèle, différents processus sont lancés pour détecter des séquences importantes de pixels noirs qui constituent les lignes du tableau. La disposition des lignes et des phrases permet la déduction des entêtes de lignes et de colonnes.

[Hurst et Douglas, 1997] ont proposé un système à deux phases pour la détection des tableaux dans des sources codées en ASCII. Les auteurs distinguent deux types de données : les données textuelles et les données numériques. Le modèle de tableau utilisé est un ensemble de **template**. Un template est un triplet [largeur, profondeur, type de données] qui permet d'indiquer la géométrie de données de même domaine. Chaque template peut être composé par de sous-templates. Les auteurs ont défini des restrictions sur les templates correspondants aux entêtes de lignes, aux entêtes de colonnes, au corps du tableau et à des colonnes de valeurs numériques. La première phase du système consiste à identifier des aires rectangulaires de type numérique ou textuel. Pour cela, les auteurs ont transformé les sources ASCII en un corpus SGML où la donnée de chaque cellule est marquée par un outil spécialisé. Ensuite, des experts ont spécifié manuellement les aires rectangulaires de données numériques et textuelles. La deuxième phase consiste à mesurer par des fonctions de cohésion s'il y a une correspondance entre les aires rectangulaires et les templates. Il s'agit alors d'une approche dirigée par un modèle et ascendante.

[Ng *et al.*, 1999] ont proposé d'utiliser des algorithmes d'apprentissage pour détecter des composants de tableaux présents dans des fichiers textes codés en ASCII. La détection du cadrage, des lignes et des colonnes s'appuie sur les algorithmes d'apprentissage C4.5 [Quinlan, 1993] et la propagation arrière [Rumelhart *et al.*, 1988]. Ces algorithmes sont appliqués sur les proportions de caractères et leur positions dans une ligne et entre les lignes. Il s'agit d'une approche dirigée par les données et descendante.

[Chen *et al.*, 2000] se sont focalisés sur des tableaux relationnels statistiques ou non-statistiques dans des pages HTML. Les auteurs proposent un processus à quatre phases dirigé par les données et ascendant. Premièrement, les tableaux entre les balises <table></table> sont extraits des pages HTML. Deuxièmement, un filtrage de formulaire ou de tableau de moins de deux colonnes est effectué. Troisièmement, les auteurs combinent les similarités des labels, les similarités des entités nommées et les similarités entre les numériques pour identifier les cellules similaires. Ceci permet de détecter les lignes ou les colonnes du corps du tableau qui sont indexées par un attribut d'une entête de lignes ou une entête de colonnes. Enfin, ils proposent un algorithme heuristique avec un raisonnement sans et avec la présence des cellules fusionnées pour interpréter la présentation des entêtes de lignes et de colonnes par rapport aux cellules similaires.

[Cafarella *et al.*, 2008b] ont proposé un système pour la détection des tableaux relationnels dans un large corpus de tableaux HTML. Son objectif est de pouvoir interroger efficacement ces tableaux dans un cadre applicatif de recherche d'informations [Cafarella *et al.*, 2008a]. Les auteurs ont utilisé des analyseurs pour écarter des tableaux spécifiques en HTML tels que les formulaires ou les calendriers. Ensuite, deux utilisateurs iden-

tifient manuellement l'ensemble de tableaux relationnels dans un échantillon de plusieurs tableaux. Enfin, le classificateur statistique proposé par les auteurs fait de l'apprentissage sur l'échantillon puis il est appliqué sur la totalité du corpus. La stratégie de cette approche est dirigée par les données. Par contre, elle ne peut pas être catégorisée comme ascendante ou descendante.

2.1.2 Synthèse et limites des travaux

Le Tableau II.1 synthétise les différentes caractéristiques des approches que nous avons décrites ci-dessus. Ces travaux montrent plusieurs limites par rapport à notre contexte. En effet, les trois approches [Laurentini et Viada, 1992], [Chen *et al.*, 2000] et [Cafarella *et al.*, 2008a] ne s'appliquent que sur des tableaux relationnels. De plus, pour [Chen *et al.*, 2000] et [Cafarella *et al.*, 2008a], les tableaux situés entre les balises <table></table> dans des pages HTML sont faciles à détecter automatiquement par rapport à des tableaux dans des fichiers XLS ou CSV.

[Laurentini et Viada, 1992] recherchent des composants connectés de type textuel tandis que nous recherchons à différencier les données numériques des données textuelles. Les approches de [Hurst et Douglas, 1997] et [Ng *et al.*, 1999] sont plus génériques que les autres approches puisqu'elles considèrent les différents types de tableaux et les différentes natures de données. La détection d'un tableau dans une source ASCII et la détection d'un tableau dans une source XLS sont du même ordre de difficulté. Toutefois, [Hurst et Douglas, 1997] font appel à des humains pour détecter les différentes aires rectangulaires tandis que notre objectif est de détecter ces aires automatiquement. L'approche de [Ng *et al.*, 1999] est la plus automatique et générique parmi toutes les autres approches mais elle ne peut pas pallier l'hétérogénéité structurelle des données ouvertes, en particulier lorsqu'un tableau est composé de sous-tableaux. Nous pensons que pour ce cas une approche ascendante serait plus efficace qu'une approche descendante.

Tableau II.1 — Comparaison des approches de détection de tableaux

	Tableau			**Détection**		
	Type	**Nature données**	**Support**	**Stratégie**	**Type**	**Techniques utilisées**
[Laurentini et Viada, 1992]	R	NS	Images de documents	dirigée par un modèle et ascendante	A	Heuristiques, OCR et templates
[Hurst et Douglas, 1997]	R/NR	S/NS	Texte(ASCII)	dirigée par un modèle et ascendante	SA	Fonction de cohésion et templates
[Ng *et al.*, 1999]	R/NR	S/NS	Texte(ASCII)	dirigée par les données et ascendante	A	Algorithmes d'apprentissage
[Chen *et al.*, 2000]	R	S/NS	HTML	dirigée par les données et ascendante	A	Heuristiques, Similarité
[Cafarella *et al.*, 2008a]	R	NS	HTML	dirigée par les données	SA	Classification et heuristiques

2.2 Les travaux de reconnaissance de structure de tableaux

Nous étudions dans cette section les approches qui se sont uniquement focalisées sur la reconnaissance de structure de tableaux. Cette tâche a pour objectif d'analyser le contenu sémantique et structurel du tableau. Nous parlerons dans cette section de modèle d'analyse et

d'annotation de contenu. En effet, le contenu des tableaux est d'abord comparé avec un modèle d'analyse puis il est annoté par des annotations issues de ce modèle. Avec l'émergence du web sémantique, nous remarquons que le modèle d'analyse est souvent une ontologie et les annotations sont des concepts issus de cette ontologie. Nous nous appuyons sur les critères suivants pour la comparaison des travaux :

- Critères relatifs au tableau :
  - Le type du tableau : relationnel, non-relationnel (R/NR).
  - La nature des données : statistiques, non-statistiques (S/NS).
  - Le support contenant le tableau.
- Critères relatifs à la reconnaissance de structure :
  - Les modèles d'analyse utilisés.
  - Le type d'approche de reconnaissance : Automatique, Semi-automatique, Manuelle (A/SA/M).
  - Les techniques utilisées.
  - Limitation ou non à un domaine d'étude.

2.2.1 Étude des travaux

[Embley *et al.*, 2002] ont proposé la reconnaissance de tableaux relationnels S/NS dans les pages HTML en utilisant comme modèle d'analyse une "ontologie d'extraction". Cette ontologie est construite manuellement par un expert pour un domaine donné. Elle est composée d'un ensemble de concepts ayant des attributs et d'un ensemble de relations entre les concepts et les attributs. Chaque concept est décrit par un "frame" qui contient ses mots clés représentatifs, son contexte et des règles de reconnaissance de ses attributs. Dans ce travail, les auteurs supposent que la première ligne du tableau est un objet d'intérêt et que les différentes cellules de cette ligne sont les attributs de cet objet. Ils génèrent donc à partir de chaque enregistrement en-dessous de la première ligne un vecteur de la forme <attribut de la première ligne, valeur de l'attribut dans l'enregistrement>. En appliquant les règles de reconnaissance, les auteurs cherchent s'il y a des correspondances entre chaque couple <attribut, valeur> du tableau et chaque instance <attribut, valeur> de l'ontologie. De cette façon, ils arrivent à reconnaître les annotations des attributs et l'objet du tableau. Ils appliquent ensuite des règles d'inférences, en s'appuyant sur les relations dans l'ontologie, entre les attributs annotés pour extraire les relations possibles entre ces attributs.

[Tenier *et al.*, 2006] ont proposé une approche pour l'annotation des pages web avec les concepts et les relations d'un domaine particulier. Les pages web contiennent des tableaux codés entre les balises <table></table> qui sont de type relationnel et non-statistique. Les auteurs ont construit leur propre ontologie qui comporte des relations binaires entre des concepts hiérarchisés. Les utilisateurs doivent sélectionner manuellement les concepts de l'ontologie pour annoter les termes dans les tableaux. Ensuite les auteurs exploitent la technique d'inférence pour déduire les relations entre les concepts sélectionnés. Ceci permet de reconnaître les relations entre les concepts du tableau.

[Hignette, 2007] a proposé dans ses travaux de thèse une méthode pour l'annotation des tableaux guidée par une ontologie de domaine (la microbiologie) construite manuellement par des experts. La méthode permet l'annotation des cellules, des colonnes et des relations

dans des tableaux en format XTAB [Saïs *et al.*, 2005] (des tableaux relationnels matérialisés en XML). Pour annoter les données textuelles dans les cellules, dites termes, elle calcule des similarités lexicales (différents types de distances sont utilisées) entre ces termes et les termes de l'ontologie. Pour annoter les colonnes, l'auteur compare les intervalles, les unités de mesure et les valeurs des données numériques avec celles de l'ontologie de domaine ce qui lui permet de détecter s'il s'agit de données purement numériques (ex. mesures de résultats d'expériences...) ou de données symboliques (ex. réponse d'un micro-organisme à un traitement...). Les relations sont identifiées en utilisant le titre du tableau ainsi que les relations de l'ontologie. L'auteur a choisi de produire des annotations floues puisque plusieurs valeurs peuvent être attribuées à chaque type d'annotation.

[Van Assem *et al.*, 2010] ont proposé une approche pour l'annotation de tableaux relationnels statistiques en format XLS. La reconnaissance cible les données quantitatives. Elle utilise comme modèle d'analyse l'ontologie OUM (Ontology of Units of Measure and related concepts). Cette ontologie a été développée par les auteurs et elle comporte les principaux concepts : Quantité, Unité de mesure, Dimension, Domaine d'application. L'approche se déroule en cinq phases. Premièrement, les valeurs des cellules de la première ligne du tableau sont transformées en sacs de termes. Deuxièmement, ils calculent les similarités entre les sacs de termes et les noms de quantités et d'unités de OUM, en utilisant la distance de similarité Jaro-Winkler-TFIDF. Dans la troisième et quatrième phase, ils calculent les similarités entre la composition d'unités de termes et les unités composées appartenant ou pas à OUM. Finalement, des règles heuristiques définissent des stratégies de désambiguïsation si le même terme se réfère à différentes quantités.

[Scharffe *et al.*, 2012] ont proposé le projet Datalift [5]. Il s'agit d'une plateforme pour la publication et la liaison des données du web. Datalift propose plusieurs fonctionnalités telles que la transformation des données en RDF, l'annotation sémantique des données avec des ontologies, la liaison des données avec SILK [6][Jentzsch *et al.*, 2010], la visualisation, etc. Pour transformer les données tabulaires (format XLS ou CSV) en RDF, Datalift effectue la reconnaissance de tableaux relationnels. Ils supposent que les cellules de la première ligne représentent des propriétés et à partir de la seconde ligne chaque ligne représente un sujet en RDF. En suivant ce principe, ils transforment le tableau en plusieurs triplets RDF tel que chaque triplet (s,p,o) correspond à (l'identifiant de l'objet de la ligne i, la valeur de première ligne à la colonne j, la valeur de la cellule de la ligne i et colonne j). Les données tabulaires transformées en RDF sont ensuite annotées par des vocabulaires sélectionnés du LOV (Linked Open Vocabulary) [7]. L'utilisateur doit lui même effectuer et valider la recherche des concepts du vocabulaire pouvant le mieux annoter les propriétés. En utilisant ces résultats, le système pourrait générer automatiquement les annotations des différents triplets.

[Buche *et al.*, 2013] ont proposé le système ONDINE (ONtology based Data INtEgration) pour l'intégration floue de tableaux guidée par une ressource ontologique et terminologique (OTR) d'un domaine donné (la microbiologie) construite manuellement par des experts du domaine. ONDINE est composé de deux sous-systèmes : (1) @web qui a pour rôle d'annoter des tableaux du web dans le domaine de microbiologie en se basant sur la

5. http://datalift.org/
6. http://silk-framework.com/
7. http://lov.okfn.org/dataset/lov/

ressource OTR et de produire des ontologies pour ces tableaux (2) MIEL++ qui a pour rôle d'interroger simultanément l'entrepôt des ontologies de tableaux et une base de données relationnelle locale par des requêtes SPARQL en utilisant aussi la ressource OTR. Le sous-système @web est la partie concernée par la reconnaissance des tableaux. @web est une extension des travaux de [Hignette, 2007] avec la nouvelle ressource OTR. L'originalité de la ressource utilisée réside dans sa composition en deux parties. La première partie est une ontologie du domaine de microbiologie, elle est constituée de différents concepts qui sont soit des concepts d'unités, soit des concepts simples. Les concepts simples se répartissent en concepts symboliques et concepts quantitatifs. La deuxième partie de l'OTR est une terminologie du domaine de microbiologie et chaque terme dénote un concept de l'ontologie. La démarche de [Hignette, 2007] a été reprise pour @web. En effet ils calculent les similarités entre les termes de chaque colonne et les termes de la ressource OTR pour annoter la colonne. Comme les termes de l'OTR dénotent des concepts symboliques ou quantitatifs, ces derniers seront les annotations des colonnes. Par déduction, des relations n-aires entre les concepts de l'ontologie annotent les relations entre les colonnes du tableau.

2.2.2 Synthèse et limites des travaux

Nous avons synthétisé les différentes caractéristiques des approches ci-dessus dans le Tableau II.2. Un premier constat est qu'aucune approche ne traite les tableaux non-relationnels. Un deuxième constat est que toutes les approches se basent sur des ontologies comme modèle d'analyse. Parmi ces ontologies, certaines sont dépendantes à un domaine par exemple celles utilisées par [Embley *et al.*, 2002], [Tenier *et al.*, 2006], [Hignette, 2007] et [Buche *et al.*, 2013]. Certes, utiliser une ontologie de domaine permet de générer des annotations pertinentes et peut aider dans l'intégration des données mais se procurer ou construire ces ontologies est un problème en soi. Nous avons relevé que ces ontologies de domaine ont été développées par des spécialistes : auteurs ou experts. Face à la variété des données ouvertes, les ontologies de domaine semblent peu appropriées car il faut en produire autant que de domaines étudiés. Par opposition, les approches de [Van Assem *et al.*, 2010] et de [Scharffe *et al.*, 2012] proposent des modèles indépendants du domaine. Le problème avec l'approche de [Van Assem *et al.*, 2010] est qu'elle suppose que la première ligne doit contenir des termes sur la nature des quantités et des unités des données numériques du tableau. Or, cette supposition n'est pas valide pour les données ouvertes statistiques non-relationnelles. En effet, les quantités et les unités de mesure sont soit dispersées dans le titre ou les notes du tableau, soit absentes (nous rappelons que les données ouvertes sont imparfaites).

L'approche de [Scharffe *et al.*, 2012] n'est pas complètement automatisée alors que nous envisageons une reconnaissance automatisée. Par ailleurs, nous avons constaté qu'à l'exception de [Tenier *et al.*, 2006] qui peut inférer des relations hiérarchiques entre les concepts, les autres approches n'abordent pas la possibilité d'avoir des relations hiérarchiques entre les concepts de l'entête de lignes. Ces approches supportent uniquement les relations n-aires ou binaires. Dans une perspective de construction d'un schéma multidimensionnel à partir de plusieurs tableaux, nous accordons une importance à la reconnaissance des relations hiérarchiques entre les concepts des entêtes.

Tableau II.2 — Comparaison des approches de reconnaissance de structure de tableau

	Tableau			Reconnaissance			
	Type	**Nature données**	**Support**	**Modèle d'analyse**	**Type**	**Techniques utilisées**	**Appliquée domaine**
[Embley *et al.*, 2002]	R	S/NS	HTML	une ontologie d'extraction	SA	règle de correspondance et d'inférence	oui
[Tenier *et al.*, 2006]	R	NS	HTML	une ontologie de domaine	SA	interaction et inférence	oui
[Hignette, 2007]	R	S/NS	XTAB	une ontologie de domaine	A	calcul de similarité et annotation floue	oui
[Van Assem *et al.*, 2010]	R	S	XLS	l'ontologie OUM	A	calcul de similarité	non
[Scharffe *et al.*, 2012]	R	S/NS	CSV/XLS	LOV	SA	mise en correspondance entre propriétés	non
[Buche *et al.*, 2013]	R	S	HTML	une ressource terminologique et ontologique OTR	A	distances de similarité et annotation floue	oui

2.3 Les travaux de détection et de reconnaissance des tableaux

Dans cette section, nous étudions des travaux qui ont développé les deux étapes de détection et de reconnaissance des tableaux. De manière analogue aux deux sections précédentes, nous relevons les différentes caractéristiques des tableaux, de l'approche de détection et de l'approche de reconnaissance pour les travaux étudiés.

2.3.1 Étude des travaux

[Pivk *et al.*, 2004] propose une méthodologie en quatre couches pour la transformation de tableau relationnel non-statistique dans des pages HTML en frame F-Logics [Kifer *et al.*, 1995]. La première couche permet le nettoyage et la normalisation des tableaux en format DOM (Document Object Model). La deuxième couche détecte la structure entre les cellules. En effet, ils transforment le tableau en une matrice où chaque cellule est soit un I-cell (cellules d'instances), soit A-Cell (cellules d'attributs) en fonction du type des termes contenus dans les cellules. Les types des termes sont identifiés à l'aide d'une hiérarchie de types (date, alapha, punct,...). Puis, ils proposent un algorithme heuristique pour découper le tableau en blocs unitaires selon la disposition des cellules fusionnées. Ensuite, ils calculent, à l'aide d'une formule, la meilleure région d'un bloc unitaire. Il s'agit alors d'une approche de détection dirigée par un modèle et ascendante. Les troisième et quatrième couches sont dédiées à la reconnaissance des relations et des concepts du tableau. Dans la troisième couche, les auteurs construisent un modèle fonctionnel FTM qui représente les relations entre les données du tableau. C'est un modèle en arbre dont les feuilles sont des blocs de I-Cell et les noeuds intermédiaires sont des cellules A-Cell. La dernière couche a pour rôle l'enrichissement sémantique du modèle FTM. Pour cela, ils ont utilisé les deux sources externes Wordnet et GoogleSets pour déterminer la classe des données appartenant à un même bloc I-Cell. Ils utilisent la distance IDF pour calculer la similarité entre les concepts des sources externes et les données des blocs d'I-Cell.

[Tijerino *et al.*, 2005] ont proposé le système TANGO (Table Analysis for Generating Ontologies) pour la construction d'une ontologie commune à plusieurs tableaux. TANGO comporte quatre phases : (1) la transformation de tableaux extraits de pages HTML en tableau relationnel (dit aussi canonique), (2) la construction de mini-ontologie à partir de ces tableaux, (3) la découverte de correspondances entre les différentes mini-ontologies deux

à deux et (4) la fusion itérative des différentes mini-ontologies. Dans cette section, nous détaillons uniquement les deux premières phases de cette approche qui sont relatives à la détection et la reconnaissance des tableaux. Nous détaillerons les deux dernières phases dans l'état de l'art du chapitre 3. Les auteurs traitent des tableaux codés entre les balises <table></table> dans des pages HTML. Ils appliquent des patrons [Crescenzi *et al.*, 2001] pour identifier les colonnes. Ensuite sur les colonnes, ils essayent différents patrons lexicaux (data frame) pour identifier des données géographiques, temporelles, pourcentages. Ils utilisent aussi des heuristiques pour reconnaître les concepts représentatifs d'un ensemble de valeurs dans une colonne. Par la suite, ils construisent un tableau relationnel par les différentes colonnes détectées et annotées. A partir de ce dernier, ils combinent d'autres patrons de données et heuristiques pour découvrir les dépendances fonctionnelles et les relations entre les concepts de l'entête de colonnes du tableau. Les différentes annotations et leurs instances formeront la mini-ontologie représentative du tableau.

[Liu *et al.*, 2006] [Liu *et al.*, 2007] ont proposé le système TableSeer qui est un moteur de recherche de tableaux. C'est un système complet qui aspire des librairies digitales, détecte les tableaux, extrait les méta-données des tableaux, indexe et note ces derniers pour pouvoir appliquer la recherche d'information dans les tableaux. Nous nous focalisons uniquement sur la partie détection et reconnaissance. D'abord, les auteurs transforment les données d'un document PDF vers un document TXT qu'il nomme Document Content File (DCF). Celui-ci est une suite de lignes où chaque ligne contient les coordonnées du dernier mot, la largeur et la hauteur de la ligne, le style du texte et le texte extrait du document PDF. Pour la détection des tableaux, [Liu *et al.*, 2007] ont proposé la méthode page box-cutting. Cette méthode se déroule en plusieurs phases. D'abord ils construisent des *page-box* qui sont des rectangles de lignes connectées dans une même page ayant la même taille du style. Puis, ils les classifient en trois catégories suivant la taille du texte (petit, ordinaire, grand). Ensuite, ils parcourent chaque groupe de *page-box* et cherchent s'il y a un page-box qui commence par un mot d'une liste K (table, Figure, Form..), si c'est le cas ils vérifient si la structure du *page-box* contient des espaces pour décider s'il s'agit d'un tableau. Pour la détection, il s'agit d'une approche dirigée par les données et descendante. Concernant la reconnaissance des tableaux, [Liu *et al.*, 2006] proposent un algorithme qui parcourt le tableau détecté et extrait en même temps ces annotations en se repérant par les indices du DCF. Les auteurs proposent d'annoter le tableau par sa propre structure. Les méta-données concernent : l'environnement/géographie du tableau, le cadre du tableau, le texte en dehors du tableau (titre, notes..), le traçage du tableau (nombre de lignes, nombre de colonnes, la longueur..), le contenu de cellules (position (i, j) et contenu) et le type de cellules (numérique, symbolique).

[Coletta *et al.*, 2012][Castanier *et al.*, 2013] ont proposé un environnement web appelé WebSmatch pour l'intégration et la visualisation de données ouvertes tabulaires en format XLS. L'environnement WebSmatch rassemble des outils tiers tels que les outils de visualisation de Data Publica [8] et de Google Data Explorer [9]. Il est défini par un processus à trois phases : (1) détection et reconnaissance de tableaux, (2) intégration des données et (3) visualisation. Pour la détection des tableaux, WebSmatch combine des algorithmes de vision par ordinateur. En effet, les fichiers XLS sont transformés en une matrice binaires (0 pour les

8. http ://www.data-publica.com/
9. http ://www.google.com/publicdata/directory

cellules vides et 1 pour les cellules non-vides). Un algorithme de détection des composants connectés est appliqué sur cette matrice afin de partitionner la matrice en zones d'éléments. Des algorithmes de vision par ordinateur sont ensuite appliqués pour fusionner ces zones et tracer le cadre du tableau. Ensuite, ils classifient les données en corps, entêtes, notes, en utilisant la technique d'apprentissage sur des règles établies à partir des habitudes des utilisateurs. Par exemple "si une cellule dans la première ligne formée par des composants connectés de type textuel et que la deuxième ligne contient des éléments numériques alors la première ligne constitue une entête". Il s'agit donc d'une approche de détection dirigée par les données et descendante. Pour la reconnaissance des données du tableau, ils ont utilisé l'outil YAM++ [Ngo et Bellahsene, 2012] qui permet de déduire des descriptions DSPL (Data Set Publishing Language) en résolvant la tâche d'alignement des instances par rapport à une liste prédéfinie de description DSPL. Ils proposent aussi dans l'environnement web, la possibilité de sélectionner interactivement des descriptions DSPL prédéfinies.

2.3.2 Synthèse et limites des travaux

Nous avons synthétisé dans le Tableau II.3 les caractéristiques des quatre approches décrites ci-dessus. L'approche de [Pivk *et al.*, 2004] a trois limites : (1) la présence de cellules fusionnées est obligatoire pour l'application de l'algorithme de détection de la structure du tableau alors que ces cellules ne sont pas toujours présentes, (2) aucune conclusion ne peut être tirée sur l'applicabilité de cette approche sur des tableaux non-relationnels ou sur plusieurs tableaux simultanément, (3) les cellules fusionnées peuvent aussi signifier la présence d'une hiérarchie entre les cellules ce qui n'est pas exploité par les auteurs.

Nous considérons que l'approche TANGO [Tijerino *et al.*, 2005] est exhaustive puisque plusieurs patrons sont utilisés pour la reconnaissance et l'annotation des concepts du tableau. Mais la transformation de colonnes en tableau relationnel réduit d'emblée le type de relations qui peuvent être déduites, chose que nous pouvons constater dans le résultat de recherche de dépendances fonctionnelles entre les entêtes de colonnes. En effet, il y a forcément un unique concept central et autour de lui soit des sous-concepts soit des attributs. L'hypothèse qu'un tableau peut être analysé par plusieurs concepts centraux non-connectés est éliminé. Dans l'approche de [Liu *et al.*, 2006][Liu *et al.*, 2007], nous partageons l'initiative d'annoter un tableau par ses méta-données qui le caractérisent non pas pour rechercher des tableaux mais pour capitaliser et réutiliser ces informations. Toutefois, il nous semble que leur algorithme est très lié au format PDF puisque le style de texte est primordial dans ce dernier. Le manque de cette information dans XLS ou CSV par exemple peut poser un problème pour la détection des tableaux. Enfin l'approche de [Coletta *et al.*, 2012] [Castanier *et al.*, 2013] qui est la plus proche de notre contexte a deux limites selon notre point de vue. Tout d'abord, la détection des entêtes en se basant sur les habitudes est informelle ce qui peut dégrader la qualité de détection des composants du tableau. Ensuite, les annotations sont restreintes aux types que Google exige dans le format DSPL.

Tableau II.3 — Comparaison des approches de détection et de reconnaissance des tableaux

	Tableau			Détection			Reconnaissance		
	Type	**Nature Données**	**Support**	**Stratégie**	**Type**	**Techniques utilisées**	**Modèle d'analyse**	**Type**	**Techniques utilisées**
[Pivk *et al.*, 2004]	R	S/NS	HTML	dirigée par un modèle et ascendante	A	algorithmes heuristiques	Wordnet GoogleSets	SA	calcul similarité
[Tijerino *et al.*, 2005]	R	S/NS	HTML	dirigée par les données et ascendante	A	patrons	patrons et frames	A	apprentissage
[Liu *et al.*, 2006] [Liu *et al.*, 2007]	R/NR	S/NS	PDF	dirigée par les données et descendante	A	méthode page box-cutting	méta-données	A	algorithme d'extraction
[Coletta *et al.*, 2012] [Castanier *et al.*, 2013]	R/NR	S/NS	XLS	dirigée par les données et ascendante	A	vision par ordinateur apprentissage	DSPL	SA	alignements d'instances

3 Contribution à la détection et à la reconnaissance des données ouvertes tabulaires

Dans cette section, nous décrivons notre approche de détection et de reconnaissance des données tabulaires. La détection vise à identifier l'emplacement et le type des composants du tableau. La reconnaissance vise à décrire le contenu du tableau. Dans cette section, nous employons le terme annotation (attachement d'une étiquette décrivant le composant) qui représente la technique utilisée pour la reconnaissance des composants.

Un aperçu global de notre approche est illustré dans la Figure II.2 :

- en entrée, nous avons des données ouvertes tableaux de nature *statistique* (S) et de type *relationnel* ou *non-relationnel* (R/NR). Ces tableaux se trouvent dans des sources en format XLS ou CSV.
- en sortie, nous avons des graphes (graphes de propriétés [Rodriguez et Neubauer, 2010] ou graphes RDF). Ces graphes représentent les schémas des tableaux qui seront utilisés pour l'intégration des données.

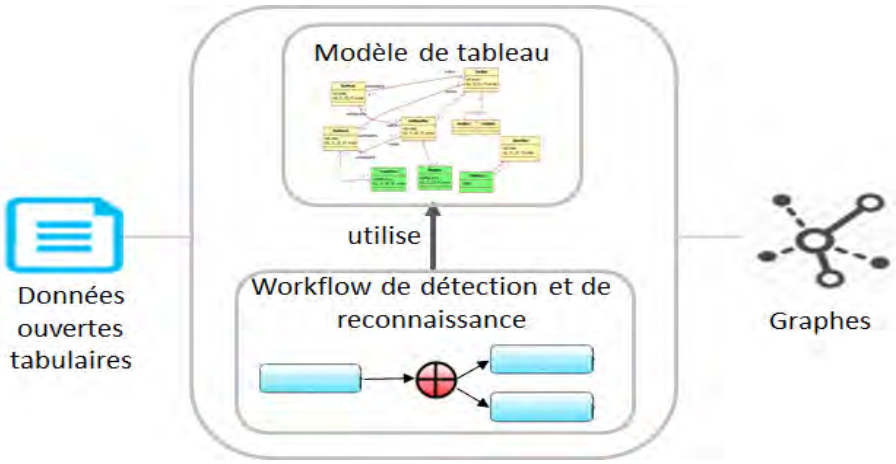


Figure II.2 — Un aperçu global de notre approche de détection et de reconnaissance des tableaux

Notre approche comporte deux propositions :

- La première proposition est un modèle de tableau. Ce modèle permet de décrire d'une façon précise et homogène les composants de chaque tableau et les relations entre eux. Ce modèle est également utilisé dans les annotations qui vont être attribuées aux

composants du tableau.
- La deuxième proposition est un workflow de détection et de reconnaissance. Le workflow est composé de différentes activités réparties sur trois niveaux et dépendantes fonctionnellement. Chaque activité s'appuie sur le modèle de tableau pour détecter le type et l'emplacement du composant et pour produire les annotations du composant.

Notre proposition fait partie des approches dirigées par un modèle et ascendantes. En effet, la stratégie adoptée consiste à découvrir les plus petits composants du tableau puis les composants les plus complexes. Par rapport aux travaux de la littérature, notre approche se distingue par des activités destinées à la découverte automatique de relations hiérarchiques sans faire appel à des ressources externes. Ces activités s'appliquent d'une façon générique à n'importe quel domaine d'étude puisqu'elles ne dépendent que du contenu des tableaux. Elles permettent aussi à un stade avancé dans notre démarche ETL, la préparation de l'organisation hiérarchique des données tabulaires afin de faciliter la découverte du schéma multidimensionnel.

Notre contribution se résume dans les points suivants :

- Un modèle de tableau qui fournit une vision homogène sur les composants du tableau. Il permet de résoudre le problème d'hétérogénéité structurelle des tableaux.
- Des annotations qui s'appuient sur le modèle de tableau. Ces annotations permettent de capitaliser les résultats de détection des composants. Elles permettent également d'être informé sur le contenu sans avoir recours à des ressources externes.
- Des activités de détection et de reconnaissance automatiques. Ceci permet d'automatiser l'étape d'extraction du processus ETL.
- Une hiérarchisation des concepts, sans avoir recours à des ressources externes, applicable sur n'importe quel domaine d'étude. Ceci permet de pallier le problème de diversité et d'hétérogénéité sémantique des données ouvertes tabulaires.
- Une transformation des données tabulaires en graphes fournit les éléments nécessaires pour l'intégration des données. La transformation en graphes RDF favorise également la réutilisation dans le contexte du web sémantique.

3.1 Description formelle d'un modèle de tableau

Nous nous intéressons aux tableaux de nature statistique, de type relationnel ou non relationnel et contenus dans des sources en format XLS ou CSV. Ces sources sont disposées géométriquement dans des grilles à deux dimensions. Le contenu de ces sources peut être transposé en matrices $S_{n,m}$ de taille $n \times m$.

Un tableau statistique $T_{nbL,nbC}$ est une sous matrice de $S_{n,m}$ que nous définissons par le tuple $\langle C, P, R \rangle$ où :

- C est l'ensemble des composants du tableau. Un composant peut être une cellule ou un bloc de cellules.
- P est une fonction qui renvoie les délimitations du composant C dans la matrice $S_{n,m}$. $P : C \rightarrow \mathbb{N}^4$. D'une façon générale, $P(C) = (DL, FL, DC, FC)$ où DL (Début de Ligne), FL (Fin de Ligne), DC (Début de Colonne), FC (Fin de Colonne). Pour une cellule c située en ligne i et colonne j, $P(c) = (i, i, j, j)$ et pour un bloc de cellules b situé entre la

ligne i et i' et entre la colonne j et j', $P(b) = (i, i', j, j')$.
– R est la relation entre les composants C du tableau.

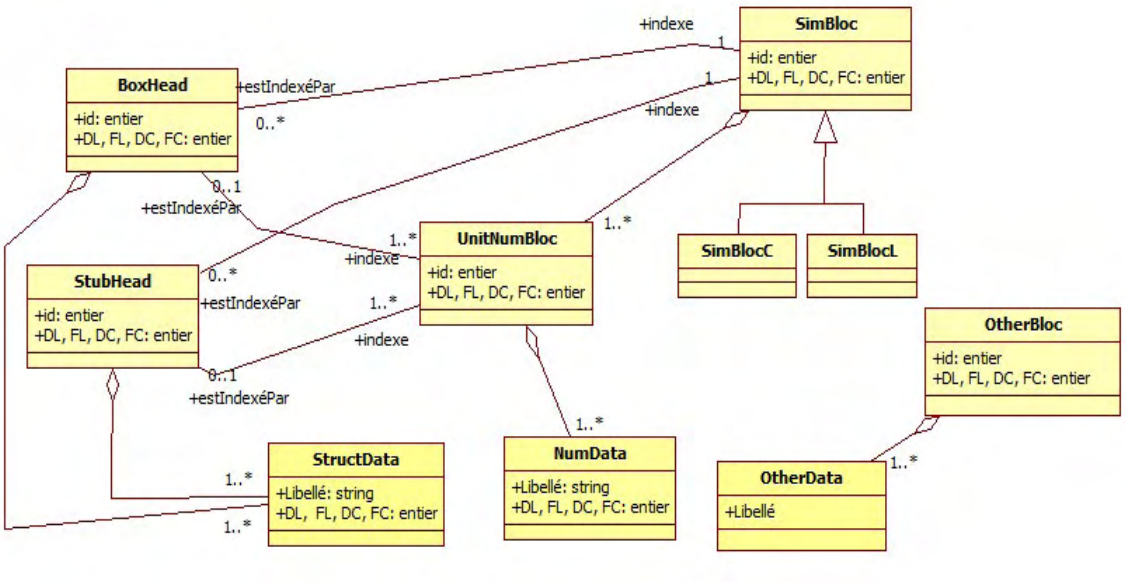


Figure II.3 — Une représentation en UML du modèle de tableau

Nous illustrons, dans la Figure II.3, un diagramme de classe des composants C et des relations R d'un tableau $T_{nbL,nbC}$.

– Une cellule $c \in \{StructData, NumData, OtherData\}$. Les cellules $StructData$ interviennent dans le schéma du tableau alors que les cellules $NumData$ contiennent les données statistiques du tableau.
– Un bloc de cellules $b \in \{Stubhead, BoxHead, UnitNumBloc, SimBloc, SimBlocC, SimBlocL\}$.
  – Un bloc de type $BoxHead$ représente l'entête de colonnes d'un tableau. Ce bloc est composé de données structurelles $StructData$.
  – Un bloc de type $StubHead$ représente l'entête de lignes d'un tableau. Ce bloc est composé de données structurelles $StructData$.
  – Un bloc de type $UnitNumBloc$ représente le corps d'un tableau composé par un ensemble contiguë de cellules $NumData$. Ce bloc pourrait être indexé soit par un $BoxHead$, soit par un $StubHead$, soit par les deux.
  – Un bloc de type $SimBloc$ représente un ensemble de blocs numériques unitaires similaires dans le sens où ils sont indexés soit par un même $BoxHead$, soit par un même $StubHead$.
    – Un bloc de type $SimBlocC$ représente un ensemble de $UnitNumBloc$ indexé par le même $StubHead$ et différents $BoxHead$.
    – Un bloc de type $SimBlocL$ représente un ensemble de $UnitNumBloc$ indexé par le même $BoxHead$ et différents $StubHead$

Exemple 2. *Dans la FigureII.4, nous illustrons les composants de deux tableaux*[10]*. Le premier tableau contient des statistiques sur le nombre de campings par catégorie de campings. Il est composé de trois UnitNumBloc indexés par différents StubHead et le même BoxHead, ces blocs ont été*

10. disponibles sur le fournisseur data.gouv.fr

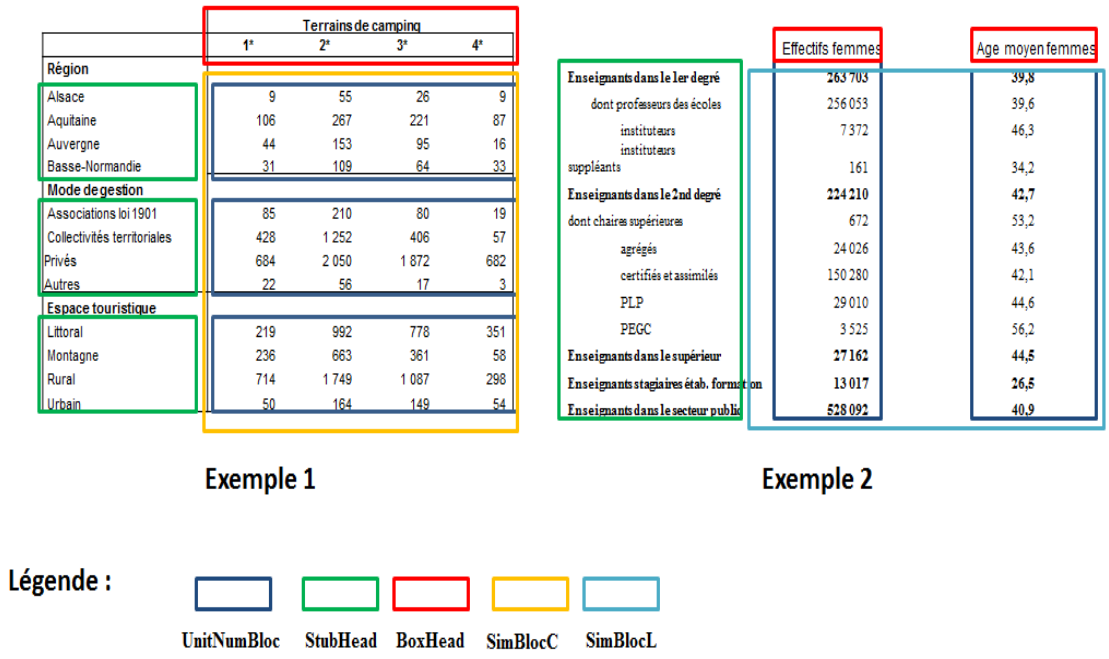


Figure II.4 — Exemple d'illustration des composants d'un tableau

rassemblés dans un SimBlocC. Le deuxième tableau contient des statistiques sur la féminisation du personnel dans l' enseignement supérieur et secondaire. Il contient deux UnitNumBloc indexés par le même StubHead et deux BoxHead différents. Ces blocs sont rassemblés dans un SimBlocL.

Nous avons choisi d'annoter les tableaux en s'appuyant sur les noms des composants décrits dans le modèle de tableau. Notre solution à trois avantages : (1) nous capitalisons toutes les informations d'un tableau, (2) nous pouvons annoter des tableaux indépendamment de leur domaine d'étude ce qui permet de couvrir la diversité des données ouvertes tabulaires et (3) nous pouvons réutiliser ces annotations pour proposer automatiquement de nouvelles annotations à des tableaux structurellement similaires ou provenant de même fournisseur.

Chaque composant C aura trois types d'annotations qui seront les propriétés de chaque composant dans le graphe de propriétés :

- Annotation Intrinsèque (AI) qui décrit le type des données contenues dans le composant, nous distinguons essentiellement des données numériques, des données textuelles, des formules et des dates. Les valeurs de AI sont des labels de l'ensemble $\{Numérique, Label, Formule, Date\}$.
- Annotation Topologique (AT) qui décrit l'espace topologique auquel appartient le composant. Les cellules *StructData* appartiennent à des *StubHead* ou à des *BoxHead*. Les *StubHead* (resp. les *BoxHead*) peuvent appartenir au *StubHead* (resp. *BoxHead*) qui indexe les *SimBloc*. Les cellules *NumData* appartiennent à des *UnitNumBloc*. Les *UnitNumBloc* appartiennent à des *SimBlocC* ou à des *SimBlocL*. Les cellules *OtherData* appartiennent à des blocs *OtherBloc*. Les valeurs de *AT* sont les identifiants des composants de type

$\{StubHead, BoxHead, UnitNumBloc, SimBlocC, SimBlocL, OtherBloc\}$.

– Annotation Sémantique (AS) qui décrit la classe sémantique d'un composant. Nous nous sommes restreints aux classes temporelle (Année, Mois,...) et géographique (Région, département,...) puisque d'une part ce sont des ressources sémantiques faciles à obtenir et d'autre part elles sont primordiales pour une analyse multidimensionnelle des données. Les valeurs de AS sont des labels de l'ensemble $\{Temporel.*, Géophraphique.*, AutreSem\}$.

3.2 Un workflow pour la détection et la reconnaissance des tableaux

Notre proposition de détection et de reconnaissance se représente sous la forme d'un workflow d'activités. Ce workflow prend en entrée des données ouvertes tabulaires et renvoie en sortie un graphe de propriétés qui peut être sérialisé en différents formats.

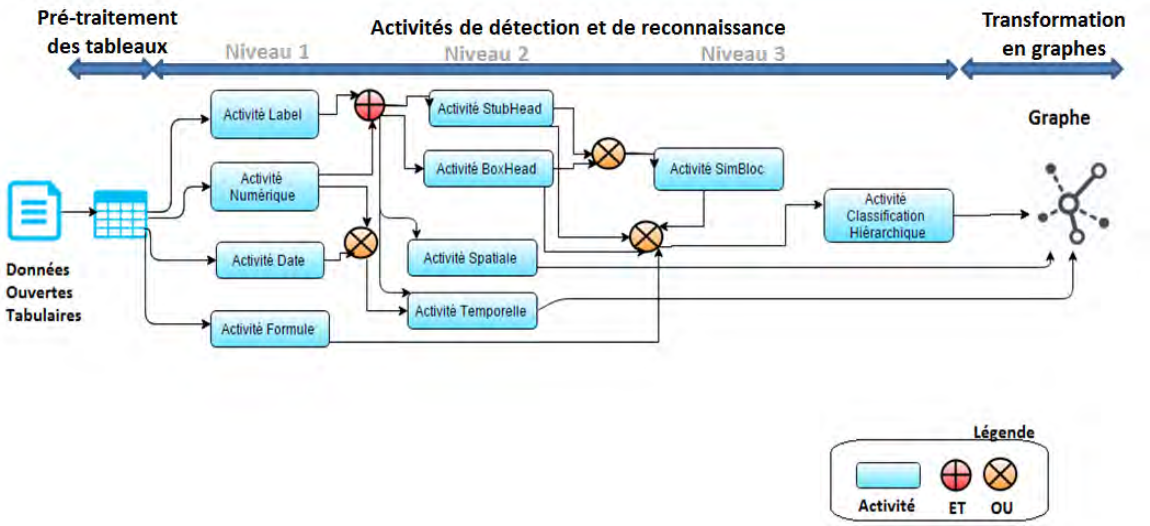


Figure II.5 — Un workflow pour la détection et la reconnaissance des tableaux

La Figure II.5 illustre :

– une première partie de pré-traitement des tableaux ; elle consiste à transformer les tableaux en matrices d'entiers.

– une deuxième partie qui comporte les activités de détection et de reconnaissance des composants du tableau. Les activités sont réparties sur trois niveaux en fonction d'une dépendance fonctionnelle entre eux. Chaque activité s'appuie sur le modèle du tableau, décrit dans la section précédente, pour détecter les composants ensuite produire leurs annotations (c'est-à-dire reconnaissance du contenu des composants).

– une troisième partie pour la transformation des résultats de détection et de reconnaissance en graphes.

3.2.1 Pré-traitement des tableaux

La première étape du workflow consiste à transformer les tableaux en matrices d'entiers. Les activités de détection et de reconnaissance vont se baser sur ces matrices d'entiers.

Les matrices d'entiers correspondent à un encodage des types des cellules des tableaux. Dans chaque source de données $S_{n,m}$, nous avons cinq types de cellule $\{Vide, Label, Numérique, Date, Formule\}$. Les cellules vides ne seront pas retenues pour la reconnaissance, mais elles sont utilisées dans les algorithmes de détection.

Nous définissons la fonction *ent* qui renvoie à chaque type de cellule un entier.

$$ent : \{Vide, Label, Numérique, Date, Formule\} \rightarrow \{-1, 0, 1, 2, 3\}$$

En utilisant cette fonction, nous avons transposé la source $S_{n,m}$ en une matrice d'entiers $E_{n,m}$ représentant le type des cellules. Les règles de transposition sont comme suit :

- si la cellule $s_{i,j}$ est de type Vide alors $e_{i,j} = ent(Vide) = -1$
- si la cellule $s_{i,j}$ est de type Label alors $e_{i,j} = ent(Label) = 0$
- si la cellule $s_{i,j}$ est de type Numérique alors $e_{i,j} = ent(Numérique) = 1$
- si la cellule $s_{i,j}$ est de type Date alors $e_{i,j} = ent(Date) = 2$
- si la cellule $s_{i,j}$ est de type Formule alors $e_{i,j} = ent(Formule) = 3$

3.2.2 Les activités de détection et de reconnaissance de niveau 1

Les activités de niveau 1 ont pour objectif de détecter et reconnaître les composants (cellules ou blocs de cellules) de même type. Comme le montre la Figure II.5, nous avons quatre activités dans le premier niveau : (1) activité dédiée aux composants labels, (2) activité dédiée aux composants numériques, (3) activité dédiée aux composants dates et (4) activité dédiée aux composants formules. Chaque activité détecte les délimitations des composants (cellules ou blocs de cellules) de même type puis les annote avec les valeurs des annotations intrinsèques ou topologiques correspondantes.

Les activités dédiées aux composants "labels" et "dates" vont chacune construire un composant *StructData*, lui définir sa position P et l'annoter intrinsèquement par la valeur correspondante de l'ensemble AI. Ces activités vont aussi construire un bloc de cellules *bs* pour l'ensemble des cellules adjacentes de même type, lui définir sa position P, l'annoter intrinsèquement par le type des données qui le composent. Ce bloc a momentanément un type par défaut *OtherBloc* puisque nous n'avons pas encore reconnu s'il fait partie des blocs *StubHead* ou *BoxHead*. Ces activités établissent aussi la relation d'appartenance entre le composant *StructData* et le bloc *bs* en ajoutant une annotation topologique dans le composant *StructData* qui prend comme valeur l'identifiant de *bs*.

L'activité dédiée aux composants "numériques" construit chaque composant *NumData*, lui définit sa position P et lui ajoute l'annotation intrinsèque *Numérique*. Cette activité construit aussi des blocs de cellules numériques *UnitNumBloc*, définit leur position P et les annote topologiquement par *Numérique*. Enfin, elle établit la relation d'appartenance entre les *NumData* et les *UnitNumBloc*, en ajoutant une annotation topologique dans les *NumData* qui porte la valeur de l'identifiant du bloc *UnitNumBloc*.

L'activité dédiée aux composants "formules" construit chaque composant *OtherData*, lui définit sa position P et lui ajoute l'annotation intrinsèque *Formule*. Cette activité construit des blocs de type *OtherBloc*, définit leur position P et les annote topologiquement par

Formule. Enfin, elle établit la relation d'appartenance entre les *OtherData* et les *OtherBloc*, en ajoutant un annotation topologique dans les *OtherData* de valeur l'identifiant du bloc auquel elle appartient.

Pour chacune des activités, le principe de détection des blocs de cellules selon leur type X (*Label*, *Numérique*, *Date*, *Label*) est illustré par l'algorithme II.1. Ce dernier prend en entrée la matrice $E_{n,m}$ et l'ensemble des blocs B construits par les autres activités de même niveau. En sortie, il met à jour B avec les nouveaux blocs de type X. La matrice $E_{n,m}$ est parcourue, si un entier $e_{i,j}$ pour le type X est trouvé alors nous cherchons le bloc b auquel appartient ce dernier (ligne 4). Ensuite, nous vérifions si le bloc b intersecte un bloc b' de B (ligne 5), si c'est le cas nous découpons le bloc b' en sous blocs rectangulaires et nous ajoutons ces derniers ainsi que le bloc b dans B (ligne 7), sinon nous ajoutons uniquement b dans B (ligne 10). Nous reprenons ainsi la détection des blocs de type X à partir de la ligne $b.FL + 1$ et la colonne $b.FC + 1$ (ligne 13-14).

Algorithme II.1 — Détection de bloc de type X

Input: $E_{n,m}$, ensemble de blocs B
Output: ensemble de blocs B

```
1:  i, j ← 1
2:  while i ≤ n et j ≤ m do
3:    if e_{i,j} = ent(X) then
4:      b ← PositionBloc(i,j,X)
5:      b' ← IntersectBloc(b, B)
6:      if b' ≠ ∅ then
7:        B ← B ∪ Decouper(b',b)
8:        B ← B\b'
9:      else
10:       B ← B ∪ b
11:     end if
12:   end if
13:   i ← B.FL + 1
14:   j ← B.FC + 1
15: end while
```

La fonction **PositionBloc** de détection des délimitations d'un bloc est illustrée par l'algorithme II.2. A partir des indices i et j nous lançons deux boucles. La première boucle (entre la ligne 3 et la ligne 13) cherche dans toutes les lignes qui succèdent i sur la colonne j, des cellules de même type X. Cette recherche est approximative puisque nous autorisons dans la ligne recherchée des sauts d'une cellule de type différent à X. La deuxième boucle (entre la ligne 16 et la ligne 26) cherche dans toutes les colonnes qui succèdent la colonne j sur la ligne i, des cellules de même type X. Nous utilisons la même approximation utilisée pour la première boucle. Notre algorithme heuristique va ainsi définir les délimitations du bloc en utilisant les indices renvoyés par les deux boucles, nous obtenons alors $P(b) = (i, k-1, j, l-1)$ (les lignes 27-30).

Exemple 3. *La Figure II.6 montre un enchaînement de détection de blocs par l'application des quatre*

Algorithme II.2 — **PositionBloc(i, j, X)**

Output: un bloc b

1: $k \leftarrow i+1$
2: $isX \leftarrow true$
3: **while** $k \leq n$ *et* isX **do**
4: **if** $e_{k,j} != ent(X)$ **then**
5: **if** $e_{k+1,j} != ent(X)$ **then**
6: $isX \leftarrow false$
7: **else**
8: $k \leftarrow k+2$
9: **end if**
10: **else**
11: $k \leftarrow k+1$
12: **end if**
13: **end while**
14: $l \leftarrow j+1$
15: $isX \leftarrow true$
16: **while** $l < m$ *et* isX **do**
17: **if** $e_{i,l} != ent(X)$ **then**
18: **if** $e_{i,l+1} != ent(X)$ **then**
19: $isX \leftarrow false$
20: **else**
21: $l \leftarrow l+2$
22: **end if**
23: **else**
24: $l \leftarrow l+1$
25: **end if**
26: **end while**
27: $b.DL \leftarrow i$
28: $b.FL \leftarrow k-1$
29: $b.DC \leftarrow j$
30: $b.FC \leftarrow l-1$

activités de détection de niveau 1. A l'étape (I), le tableau est dans son état initial. A l'étape (II), nous appliquons l'activité de détection numérique qui détecte un seul bloc numérique. A l'étape (III), nous appliquons l'activité de détection label, deux blocs de labels sont détectés dont l'un est inclus dans le bloc numérique ce qui engendre le découpage du bloc numérique en deux blocs numériques et un bloc de label. A l'étape (IV), nous appliquons l'activité de détection de date, deux blocs de date sont détectés mais n'intersectent aucuns blocs détectés par les autres activités. Dans la dernière étape , nous appliquons l'activité de détection de formule qui détecte deux blocs contenus dans deux blocs numériques ce qui implique le découpage de chaque bloc numérique en deux blocs numériques et un bloc de formule.

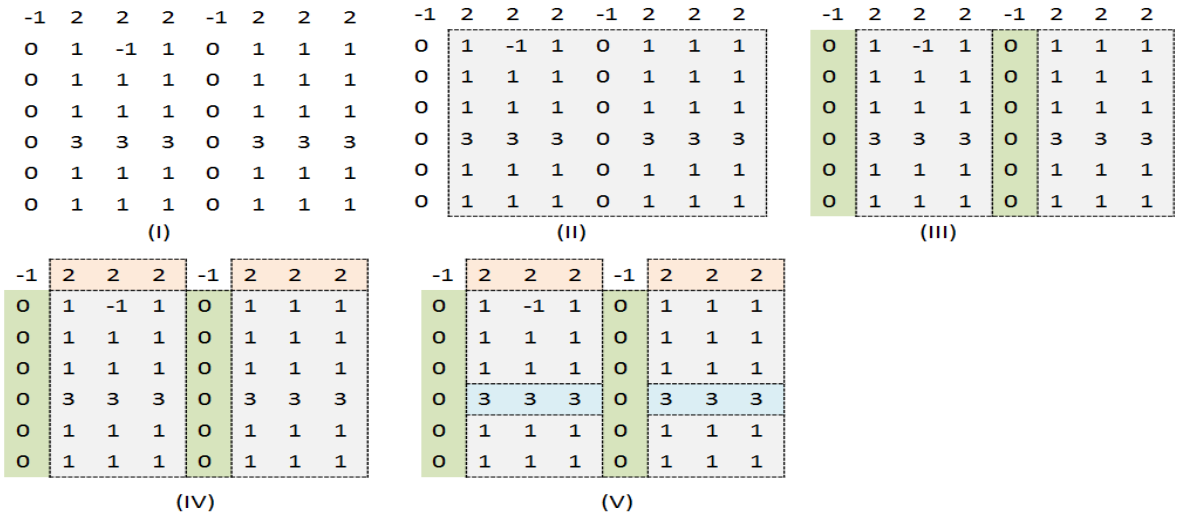


Figure II.6 — Exemple de détection des blocs d'un tableau : niveau 1

Après avoir défini les activités de premier niveau, nous décrivons dans la section suivante le fonctionnement des activités de niveau 2.

3.2.3 Les activités de détection et de reconnaissance de niveau 2

Les activités de niveau 2 ont pour objectif de détecter et de reconnaître les entêtes de lignes et de colonnes ainsi que les composants spatio-temporels. Comme le montre la Figure II.5, nous avons quatre activités dans le deuxième niveau : (1) activité dédiée aux entêtes de lignes, (2) activité dédiée aux entêtes de colonnes, (3) activité dédiée aux composants géographiques (ou spatiaux) et (4) activité dédiée aux composants temporels. Chaque activité détecte les composants en se basant sur les résultats des activités de niveau 1. Chaque activité précise également les annotations topologiques et/ou sémantiques de chaque composant détecté.

3.2.3.1 L'activité dédiée aux entêtes de lignes

L'activité de détection des entêtes de lignes permet d'identifier les entêtes de lignes *StubHead* de chaque *UnitNumBloc*. Ces entêtes sont situées à gauche du bloc numérique

unitaire et doivent se trouver dans l'un des *OtherBloc* ayant comme annotation intrinsèque *Label* que nous avons identifié au niveau 1 par l'activité "Détection Label". Ainsi l'exécution de l'activité de détection de *StubHead* dépend de l'exécution des activités de détection des labels et des numériques comme le montre la Figure II.5. L'algorithme II.3 résume le déroulement de l'activité de détection des StubHead. En effet, cet algorithme commence par chercher chaque bloc numérique b dans la liste des blocs B construit par les activités de niveau 1. Pour chaque bloc numérique b, nous parcourons B à la recherche d'un bloc de label b' dont la colonne de fin (b'.FC) est inférieure à la colonne de début de b (b.DC) (lignes 7). Si le bloc b' existe, nous cherchons s'il contient une colonne vide qui permettra de délimiter le stubhead. En effet, les délimitations de lignes (DL et FL) du stubhead sont celles du bloc numérique b indexé par ce dernier. Les délimitations de colonnes (DC et FC) du stubhead sont celles de l'indexe de la colVide et la dernière colonne de b' (lignes 8-16). Nous découpons par la suite le bloc b' selon les délimitations du stubhead et nous mettons à jour la liste des blocs B (lignes 17-18). La relation *estIndexéPar* et son inverse *indexe* sont créés entre le stubhead et le bloc b.

Algorithme II.3 — Détection de StubHead

Input: l'ensemble de blocs B

1: **for each** $b \in B$ **do**
2: **if** $b.IA = Numérique$ **then**
3: $stubNotDetect \leftarrow true$
4: $i \leftarrow 1$
5: **while** $B \neq \emptyset$ **and** $stubNotDetect$ **do**
6: $b' \leftarrow B[i]$
7: **if** $(b'.IA = Label\ et\ b'.FC < b.DC)$ **then**
8: $colVide \leftarrow$ **PremiereColonneVide(b')**
9: **if** $colVide > b'.DC$ **then**
10: $stubhead.DC \leftarrow colVide + 1$
11: **else**
12: $stubhead.DC \leftarrow b'.DC$
13: **end if**
14: $stubhead.FC \leftarrow b'.FC$
15: $stubhead.DL \leftarrow b.DL$
16: $stubhead.FL \leftarrow b.FL$
17: $B \leftarrow B \cup$ **Decouper(b', stubhead)**
18: $B \leftarrow B \backslash b'$
19: $stubNotDetect \leftarrow false$
20: **end if**
21: $i \leftarrow i + 1$
22: **end while**
23: **end if**
24: **end for**

3.2.3.2 L'activité dédiée aux entêtes de colonnes

L'activité de détection des entêtes de colonnes permet d'identifier les entêtes de colonnes (*BoxHead*) de chaque *UnitNumBloc*. Ces entêtes sont situées au-dessus du bloc numérique unitaire et doivent se trouver dans l'un des *OtherBloc* ayant comme annotation intrinsèque *Label* que nous avons identifié au niveau 1 par l'activité "Détection Label". L'algorithme II.4 de cette activité est très similaire à celui de l'activité de détection de *StubHead*, la seule différence est un raisonnement sur les lignes plutôt que sur les colonnes. La relation *estIndexéPar* et son inverse *indexe* sont créées entre le boxhead et le bloc numérique unitaire b.

Algorithme II.4 — Détection de BoxHead

Input: l'ensemble de blocs B

for each $b \in B$ **do**
  if $b.IA = Numérique$ **then**
    $boxNotDetect \leftarrow true$
    $i \leftarrow 1$
    while $B \neq \emptyset$ **and** $boxNotDetect$ **do**
      $b' \leftarrow B[i]$
      if $(b'.IA = Label\ et\ b'.FL < b.DL)$ **then**
        $ligneVide \leftarrow$ **PremiereLigneVide(b')**
        if $ligneVide > b'.DL$ **then**
          $boxhead.DL \leftarrow ligneVide + 1$
        else
          $boxhead.DL \leftarrow b'.DL$
        end if
        $boxhead.FL \leftarrow b'.FL$
        $boxhead.DC \leftarrow b.DC$
        $boxhead.FC \leftarrow b.FC$
        $B \leftarrow B \cup$ **Decouper(b', boxhead)**
        $B \leftarrow B \backslash b'$
        $boxNotDetect \leftarrow false$
      end if
      $i \leftarrow i + 1$
    end while
  end if
end for

3.2.3.3 L'activité dédiée aux composants géographiques

L'activité de détection géographique permet de rajouter des annotations sémantiques aux *StructData*, aux *NumData* et aux blocs qui les contiennent détectés par les activités de niveau 1. Nous utilisons des noms d'entités géographiques qui correspondent à une hiérarchie géographique dépendante du pays. Par exemple, si nous analysons les données ouvertes de la France nous utilisons le découpage géographique organisé en hiérarchie

comme suit : commune → canton → arrondissement → département → région. Les annotations seront {*Géographique.Commune*, *Géographique.Canton*, *Géographique.Arrondissement* , *Géographique.Département*, *Géographique.Région*}.

Nous avons récolté des listes d'instances de chaque niveau géographique de la base Geonames [11]. Nous comparons le contenu des données de nos sources avec ces listes pour identifier l'annotation sémantique des données. Si le processus de reconnaissance est positif, nous rajoutons une annotation sémantique géographique au *StructData* et au bloc qui la contient. Si nous identifiions des données géographiques dans les *NumData*, nous les transformons en *StructData*, nous attribuons l'annotation géographique correspondante et enfin nous découpons le bloc numérique qui contenait les *NumData* en un bloc *StructData* et un autre bloc de *NumData* dont on lui met à jour ces délimitations.

3.2.3.4 L'activité dédiée aux composants temporels

L'activité dédiée aux composants temporels permet de rajouter des annotations sémantiques aux *StructData* de type Date ou aux *NumData* instances de données temporelles. Les annotations temporelles sont les noms des niveaux de dimensions. Les annotations temporelles sont de la forme "Temporal. Nom du niveau de dimension". Les dimensions temporelles présentées par [Mansmann et Scholl, 2007] dans la Figure II.7 peuvent par exemple servir comme annotations. Nous avons proposé des expressions régulières pour détecter les données temporelles. Ces expressions sont appliquées sur les données numériques ou sur les données dates extraites respectivement par les activités de détection de numérique et de date du niveau 1.

Si nous détectons une colonne ou une ligne de données numériques respectant l'une des expressions régulières, les données *NumData* se transforment en données *StructData*. Elles gardent leurs annotations intrinsèques de type numérique et leurs annotations topologiques sont modifiées par l'identifiant du nouveau bloc de *StructData*. Les annotations sémantiques ("Temporal.Nom du niveau de dimension") sont remplacées par les valeurs de l'expression régulière correspondante. Ensuite, les délimitations de l'ancien bloc qui contenait ces données doivent être modifiées pour qu'il ne se chevauche pas avec le nouveau bloc de *StructData*. Dans le cas où nous détectons une colonne ou une ligne de données de type date, où ces dernières font déjà partie d'un bloc *StructData*, nous avons uniquement à ajouter l'annotation sémantique *Temporal.Date* pour ces données.

3.2.4 Les activités de détection et de reconnaissance de niveau 3

Les activités de niveau 3 ont pour objectif de détecter et de reconnaître le corps du tableau ainsi que les relations hiérarchiques entre les données structurelles. Comme le montre la Figure II.5, nous avons deux activités dans le troisième niveau : (1) activité dédiée aux blocs numériques similaires et (2) activité de classification hiérarchique. L'activité dédiée aux blocs numériques similaires permettra d'identifier le corps du tableau. L'activité de classification hiérarchique permettra la reconnaissance des relations hiérarchiques entre les

11. www.geonames.org

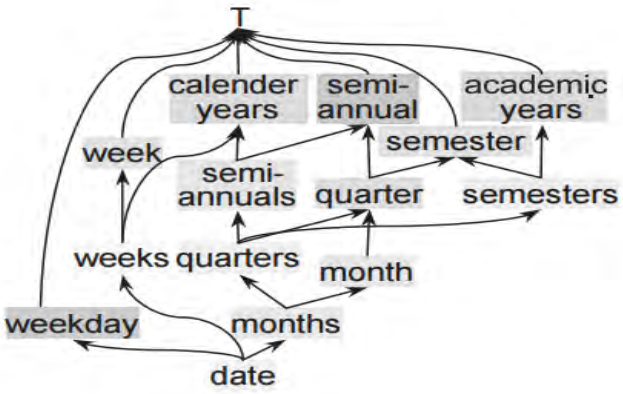


Figure II.7 — Exemple de hiérarchies de dimensions temporelles [Mansmann et Scholl, 2007]

données structurelles.

3.2.4.1 L'activité dédiée aux blocs numériques similaires

L'activité de détection des blocs numériques similaires *SimBloc*, Figure II.4, est la dernière activité de détection suite à laquelle le périmètre du tableau peut être tracé. En effet, ce périmètre comporte chaque *SimBloc* et ses entêtes de lignes et de colonnes. Si dans une source de données, nous avons détecté plus d'un *SimBloc* ceci signifie que la source contient plus d'un tableau.

L'algorithme II.5 résume le déroulement de cette activité pour les blocs similaires en lignes *SimBlocL*. Nous cherchons dans la liste des blocs B (construits par les activités de niveau 1 et 2) un *BoxHead b* pour lequel nous cherchons tous les blocs numériques unitaires qu'il indexe. Ces blocs unitaires formeront le simblocl (lignes 3-6). Nous définissons ensuite les délimitations du simblocl (lignes 8-11) et nous construisons un bloc stubhead composé par les différents *StubHead* qui indexent les blocs numériques unitaires (lignes 12-15). Les délimitations du bloc stubhead seront définies (lignes 16-19), les relations d'indexation entre le simblocl et ces entêtes seront ajoutées à R et enfin ces nouveaux blocs seront rajoutées à B. Nous ajoutons des annotations topologiques entre chaque StubHead et le nouveau bloc stubhead qui les contient. Nous ajoutons également des annotations topologiques entre les *UnitNumBloc* et le *SimBlocL* qui les contient.

L'algorithme II.6 résume le principe de détection des blocs numériques similaires en colonne *SimBlocC*. Cet algorithme est très similaire à l'algorithme II.5 mais avec un raisonnement sur les *StubHead*.

3.2.4.2 L'activité de classification hiérarchique

Les données structurelles *StructData* que nous avons extraites sont souvent aplaties dans les entêtes de lignes et de colonnes. Notre proposition consiste à appliquer une classification hiérarchique sur les données structurelles afin de découvrir les relations entre ces

Algorithme II.5 — Détection de SimBlocL

Input: l'ensemble de blocs B, l'ensemble des relations R

for each $b \in B$ **do**
  if $b = BoxHead$ **then**
    for each $r \in R$ **do**
      if $r(b, b') = indexe$ **then**
        $simblocl \leftarrow simblocl \cup b'$
      end if
    end for
    $simblocl.DC \leftarrow b.DC$
    $simblocl.FC \leftarrow b.FC$
    $simblocl.DL \leftarrow b.FL + 1$
    $simblocl.FL \leftarrow max_{b' \in simblocl}(b'.FL)$
    for each $b' \in simblocl$ **do**
      $s \leftarrow getStub(b', indexe, "StubHead")$
      $stubhead \leftarrow stubhead \cup s$
    end for
    $stubhead.DC \leftarrow min_{s \in stubhead}(s.DC)$
    $stubhead.FC \leftarrow max_{s \in stubhead}(s.FC)$
    $stubhead.DL \leftarrow min_{s \in stubhead}(s.DL)$
    $stubhead.FL \leftarrow max_{s \in stubhead}(s.FL)$
    $R \leftarrow R \cup indexe(stubhead, simblocl) \cup estIndexéPar(simblocl, stubhead)$
    $R \leftarrow R \cup indexe(b, simblocl) \cup estIndexéPar(simblocl, b)$
    $B \leftarrow B \cup simblocl \cup stubhead$
  end if
end for

données. Dans cette section, nous présentons deux stratégies de classification conceptuelle soumises toutes les deux à des contraintes particulières afin de transformer les données plates structurelles en hiérarchies de concepts. La première stratégie de classification hiérarchique est exacte, elle utilise les annotations produites par les activités de détections. La deuxième stratégie est approximative, elle combine des techniques de classification conceptuelle pour extraire les relations hiérarchiques entre les concepts structurels.

Avant d'expliquer les deux stratégies de classification hiérarchique, nous présentons tout d'abord les contraintes de classifications que nous imposons à ces stratégies.

Contraintes de classification

Un problème récurrent dans les systèmes décisionnels est la gestion des hiérarchies complexes et leur impact sur les problèmes d'additivité [Mazón *et al.*, 2010] [Hassan *et al.*, 2015]. Dans la littérature, ce problème n'a pas été considéré dans la phase de détection des schémas des sources. Toutefois, nous le trouvons bien étudié soit dans la phase d'intégration [Pedersen *et al.*, 1999] ou en temps réel dans la phase d'analyse [Hachicha, 2012]. Vu les différentes difficultés que posent ce problème, nous avons choisi de gérer les hiérarchies complexes au plus tôt dans notre démarche d'entreposage des données ouvertes afin d'évi-

Algorithme II.6 — Détection de SimBlocC

Input: l'ensemble de blocs B, l'ensemble des relations R

for each $b \in B$ **do**
  if $b = StubHead$ **then**
    for each $r \in R$ **do**
      if $r(b, b') = indexe$ **then**
        $simblocc \leftarrow simblocc \cup b'$
      end if
    end for
    $simblocc.DC \leftarrow min_{b' \in simblocc}(b'.DC)$
    $simblocc.FC \leftarrow max_{b' \in simblocc}(b'.FC)$
    $simblocc.DL \leftarrow b.DL$
    $simblocc.FL \leftarrow b.FL)$
    for each $b' \in simblocc$ **do**
      $b \leftarrow getBox(b', indexe, "BoxHead")$
      $boxhead \leftarrow boxhead \cup b$
    end for
    $boxhead.DC \leftarrow min_{b \in boxhead}(b.DC)$
    $boxhead.FC \leftarrow max_{b \in boxhead}(b.FC)$
    $boxhead.DL \leftarrow min_{b \in boxhead}(b.DL)$
    $boxhead.FL \leftarrow max_{b \in boxhead}(b.FL)$
    $R \leftarrow R \cup indexe(boxhead, simblocc) \cup estIndexéPar(simblocc, boxhead)$
    $R \leftarrow R \cup indexe(b, simblocl) \cup estIndexéPar(simblocl, b)$
    $B \leftarrow B \cup simblocc \cup boxhead$
  end if
end for

ter les problèmes d'additivité que nous pouvons rencontrer lors de la phase d'analyse. Les hiérarchies sont complexes quand elles sont non-strictes, non-couvrantes ou non-strictes et non-couvrantes. Nous rappelons qu'une hiérarchie est composée de paramètres.

- une hiérarchie est non-stricte [Malinowski et Zimányi, 2006] si un paramètre fils a plus d'un parent ; par exemple, un film A fait partie des deux catégories de films "science-fiction" et "tragédie".
- une hiérarchie est non-couvrante [Malinowski et Zimányi, 2006] si certains paramètres de la hiérarchie n'ont pas d'instances ; par exemple, dans la hiérarchie "magasin-ville-région-pays" un magasin peut être associé à une région sans être affecté à une ville ;
- une hiérarchie non-ontologique ou non-équilibrée est un cas particulier de hiérarchie non-couvrante qui comporte des instances manquantes pour les paramètres de niveau feuille ; par exemple dans la hiérarchie "magasin-ville-région-pays", une ville peut ne pas héberger de magasin.

Notre objectif est de ne pas produire des hiérarchies complexes lors des processus de classification conceptuelle. Nous avons donc défini trois contraintes que nous imposons au processus de classification. Ces contraintes sont comme suit :

– C1 : Pour chaque feuille fi de l'arbre(k), le chemin entre fi et la racine de l'arbre(k) est unique. Ce qui signifie que chaque noeud de l'arbre à l'exception de la racine a exactement un seul parent. Cette condition permet de garantir des hiérarchies strictes à l'échelle du schéma de la source de données.
– C2 : Si un noeud *n* dans l'arbre, à l'exception des racines, n'a pas de parent ou a un parent qui n'a pas de fils, nous dupliquons le noeud *n* dans le niveau manquant. Cette condition permet de garantir des hiérarchies couvrantes au niveau du schéma des sources.
– C3 : La hauteur de l'arbre doit être identique en partant de n'importe quelle feuille vers la racine de l'arbre. Cette condition permet de garantir des hiérarchies ontologiques au niveau du schéma de la source de données.

Classification conceptuelle hiérarchique exacte

Dans plusieurs sources de données ouvertes, l'organisation des données pourrait indiquer la classification conceptuelle des données structurelles *StructData*. Nous proposons dans cette section une première stratégie de classification conceptuelle exacte. Pour cette stratégie, nous avons en entrée les blocs de données structurelles qui contiennent les concepts à classifier et en sortie nous générons des relations hiérarchiques sous format d'arbres de concepts vérifiant les contraintes C1, C2 et C3.

Nous proposons les trois sous-stratégies suivantes :

– **Stratégie 1** une classification conceptuelle des concepts des entêtes de lignes (*StubHead*) qui se base sur la présence des blocs numériques similaires. Cette sous-stratégie est composée de deux étapes : la première étape permet d'affecter des niveaux aux concepts de l'entête de lignes et la deuxième étape permet de construire une hiérarchie de concepts en exploitant les niveaux de concepts.
  L'algorithme II.7 illustre le principe de la première étape d'affectation des niveaux. L'idée est de raisonner sur la disposition des blocs de données numériques *UnitNumBloc* contenus dans un *SimBlocL* pour classifier les concepts de l'entête de lignes *StubHead* qui indexent le *SimBlocL*. Nous parcourons chaque *UnitNumBloc* contenu dans le *SimBlocL* et nous affectons les *StructData* du *StubHead* qui l'indexent au niveau 1 (lignes 4-8). Nous cherchons ensuite le bloc numérique qui précède celui en cours et nous affectons aux concepts du *StubHead* (du *SimBlocL*), entre les deux *StubHead*, un niveau selon l'ordre dans lequel ces concepts apparaissent (lignes 9-16).
  L'algorithme II.8 illustre le principe de la construction d'une hiérarchie de concepts structurels de l'entête de lignes. L'idée consiste à relier par une relation de spécialisation chaque concept dans un *StructData* de niveau i aux concepts du *StructData* de niveau i-1. La Figure II.8 décrit à droite l'arbre résultant de l'application de l'algorithme II.7 et de l'algorithme II.8. Par exemple pour le premier bloc unitaire de numériques, les deux concepts "École maternelle" et "École élémentaire" sont au niveau 1 (feuille de l'arbre), le concept "Enseignement public" est au niveau 2 (parent de niveau 1) et "Premier degré" est au niveau 3 (racine de l'arbre et parent de niveau 2).
– **Stratégie 2** une classification conceptuelle des concepts des entêtes de lignes et de colonnes se basant sur la présence de cellules fusionnées dans les entêtes. Si un *BoxHead* contient une ou plusieurs cellules fusionnées, une relation de spécialisation entre

ces cellules et les cellules placées au-dessous d'eux se rajoute à l'ensemble R. Si un *StubHead* contient une ou plusieurs cellules fusionnées, une nouvelle relation de spécialisation entre ces cellules et les cellules placées sur leur droite se rajoute à l'ensemble R.

– **Stratégie 3** une classification conceptuelle des concepts des entêtes de lignes ou de colonnes qui se base sur la présence des blocs de formules. En effet, les données dans un bloc de formule ont été calculées à partir des données dans des blocs numériques. Ceci indique que les concepts qui indexent les blocs numériques appartiennent au même domaine. Donc il y a une relation de spécialisation entre les concepts structurels des entêtes et un nouveau concept représentant le domaine (pour l'instant nous lui attribuons une valeur inconnue). Dans la Figure II.9, nous montrons comment la disposition des blocs de formules permet de reconnaître une classification des concepts des entêtes de lignes et de colonnes. Dans l'ensemble R nous ajoutons une relation de spécialisation entre les concepts des entêtes et le nouveau concept (de valeur indéterminée) représentant un domaine.

Algorithme II.7 — Attribution des niveaux au StubHead

Input: un bloc similaire $SimBlocL$

1: $k \leftarrow size(SimBlocL)$
2: $i \leftarrow SimBlocL.FL$
3: **while** $i < SimBlocL.DL$ **and** $k > 0$ **do**
4: $\quad UnitNumBloc \leftarrow SimBlocL(k)$
5: $\quad$ **if** $i = UnitNumBloc.FL$ **then**
6: $\quad\quad$ **for each** $StructData \in StubHead(k)$ **do**
7: $\quad\quad\quad Niv(StructData) \leftarrow 1$
8: $\quad\quad$ **end for**
9: $\quad\quad i \leftarrow UnitNumBloc.DL$
10: $\quad\quad nextBloc \leftarrow SimBlocL(k-1)$
11: $\quad\quad cptConcept \leftarrow 2$
12: $\quad\quad$ **for** $j \leftarrow i, nextBloc.FL + 1$ **do**
13: $\quad\quad\quad Niv(StrucData) \leftarrow cptConcept$
14: $\quad\quad\quad cptConcept \leftarrow cptConcept + 1$
15: $\quad\quad\quad i \leftarrow i + 1$
16: $\quad\quad$ **end for**
17: $\quad\quad k \leftarrow k - 1$
18: $\quad$ **end if**
19: **end while**

Pour le choix de sous-stratégie à appliquer, nous pouvons combiner la sous-stratégie 3 avec soit la sous-stratégie 2 soit la sous-stratégie 1. Nous pouvons appliquer en même temps les deux sous-stratégies 1 et 2 dans le cas où nous avons des cellules fusionnées dans l'entête de colonnes et non pas dans l'entête de lignes. Si nous avons des cellules fusionnées dans l'entête de lignes et que nous avons la possibilité d'appliquer la sous-stratégie 1, il faudra appliquer uniquement la sous-stratégie 2.

Algorithme II.8 — Construction de l'arbre du StubHead

Input: *StubHead*

for $i \leftarrow StubHead.DL, StubHead.FL$ **do**
  $nivCrt \leftarrow Niv(StructData(i))$
  if $nivCrt > 1$ **then**
    while $j > i$ *et* $j < StubHead.FL$ *et* $Niv(StrutData(j))! = nivCrt$ **do**
      if $Niv(StructData(j)) = nivCrt - 1$ **then**
        $R \leftarrow R \cup specialisation(StructData(j), StructData(i))$
      end if
      $j \leftarrow j + 1$
    end while
  end if
end for

	2005-2006	2006-2007	2007-2008	2008-2009	2009-2010
Premier degré					
Enseignement public					
Écoles maternelles	26,1	25,9	25,8	25,8	25,5
Écoles élémentaires	22,5	22,6	22,7	22,6	22,6
Enseignement privé					
Écoles maternelles	27,6	27,3	26,9	26,7	27,1
Écoles élémentaires	22,8	22,9	23,1	23,0	22,9
Second degré					
Enseignement public (1)					
1er cycle (2)	24,0	24,0	23,9	24,1	24,1
2e cycle professionnel	20,1	19,7	19,4	19,0	19,0
2e cycle général et technologique	28,8	28,5	28,3	28,4	28,4
Enseignement privé					
1er cycle (2)	24,7	24,8	25,0	25,0	25,2
2e cycle professionnel	19,2	19,1	18,8	18,3	18,6
2e cycle général et technologique	25,0	24,9	24,9	24,8	25,1

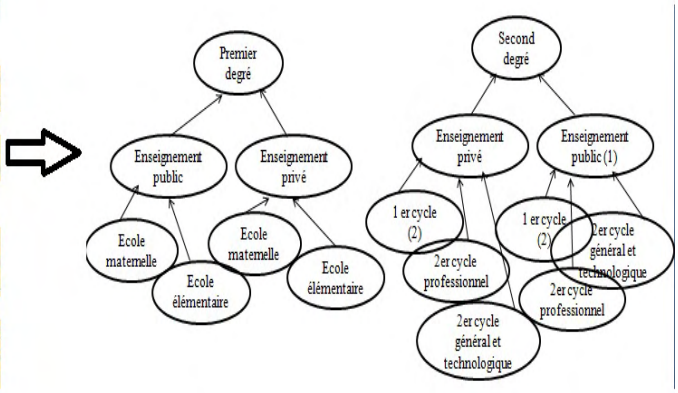


Figure II.8 — Un exemple de classification conceptuelle par la stratégie 1

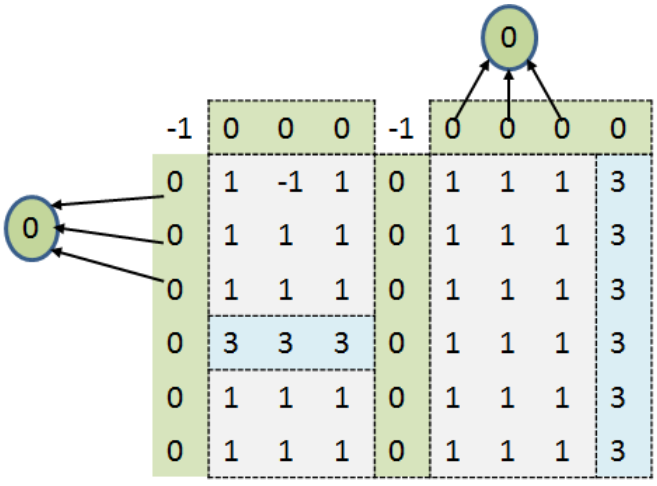


Figure II.9 — Un exemple de classification conceptuelle par la stratégie 3

	INTITULE	ILE-DE-FRANCE	CHAMPAGNE-ARDENNE	PICARDIE
A	Industries extractives, energie, eau, gestion des dechets et depollution	3 861	284	408
B	Industries extractives	215	25	30
C	Extraction de houille et de lignite	0	0	0
D	Extraction d'hydrocarbures	0 S		0
E	Extraction de minerais métalliques	0	0	0
F	Autres industries extractives	201 R		29
G	Services de soutien aux industries extractives	10	0	2
H	Production et distribution d'électricité, de gaz, de vapeur et d'air conditionné	915	32	55
I	Production et distribution d'eau ; assainissement, gestion des déchets et dépollution	2 732	227	324
J	Captage, traitement et distribution d'eau	182	10	38
K	Collecte et traitement des eaux usées	150	16 R	
L	Collecte, traitement et élimination des déchets ; récupération	2 143	201	258
M	Dépollution et autres services de gestion des déchets	256	0	0

Figure II.10 — Un exemple typique de tableau pour l'approche de classification approximative

3.2.4.2.1 Classification conceptuelle hiérarchique approximative L'hétérogénéité structurelle des données ouvertes nous a poussé à proposer une deuxième stratégie de classification approximative qui s'applique sur les entêtes de lignes et de colonnes lorsqu'aucune des sous-stratégies de la stratégie exacte n'est applicable. Dans notre approche approximative, nous croisons les résultats de classification de la technique des treillis de galois [Birkhoff, 1967] avec les résultats de l'approche RELEVANT [Bergamaschi *et al.*, 2007] sous les contraintes C1, C2 et C3 pour pouvoir transformer un ensemble de données structurelles en une hiérarchie à deux niveaux.

La classification conceptuelle avec les treillis de galois produit des contextes formels à plusieurs attributs. Cette technique ne prend pas en compte les aspects sémantiques des concepts et les hiérarchies produites ne sont pas strictes. Pour les aspects sémantiques, nous avons fait appel à l'approche RELEVANT qui considère la similarité sémantique pour regrouper un ensemble de concepts en groupes (ou clusters) représentés avec les attributs les plus pertinents selon la technique de regroupement choisie. L'approche RELEVANT propose deux techniques de regroupement : la première technique produit des groupes disjoints représentés par un concept composé de plusieurs termes, la deuxième technique produit des groupes non-disjoints représentés par un concept mono-terme. Nous avons choisi d'appliquer la première technique de regroupement puisqu'elle permet de générer des groupes disjoints vérifiant la contrainte C1. Dans ce qui suit, nous illustrons notre approche sur les concepts de l'entête de lignes du tableau de la Figure II.10. Ce tableau fait partie de la catégorie des tableaux sur lesquels nous ne pouvons pas appliquer les approches de classification exacte.

Les étapes de notre approche sont comme suit :

1. *Préparation des données* structurelles de l'entête de lignes ou de colonnes. La préparation consiste à découper le concept structurel en un sac de termes, éliminer les mots vides puis chercher les racines de chaque terme pour construire un sac de racines de termes. Pour l'ensemble des concepts structurels symbolisés alphabétiquement de A à M, la

préparation des données produit le sac de termes suivant = {(T1) industr ; (T2) extract ; (T3) énerg ; (T4) gestion ; (T5) dépollu ; (T6) houill ; (T7) lignit ; (T8) hydorcarbur ; (T9) métalliqu ; (T10) autr ; (T11) servic ; (T12) product ; (T13) distribu ; (T14) électr ; (T15) gaz ; (T16) condition ; (T17) déchet ; (T18) captag ; (T19) trait ; (T20) collect ; (T21) usé ; (T21) élimin }.

2. *Classification conceptuelle par les treillis de galois* sur les concepts structurels. L'objectif de cette étape est d'extraire un ensemble de concepts mono-termes commun à plusieurs concepts structurels. Pour réaliser cela, nous construisons un treillis de galois à partir des concepts structurels et leur sacs de termes puis nous extrayons de ce treillis les concepts mono-termes. Rappelons qu'un treillis de Galois représente un contexte formel $C = (O, A, I)$, tel que : O un ensemble fini d'objets, A un ensemble fini de termes et I une relation binaire entre O vers A. Dans notre cas, O est l'ensemble des concepts structurels, A est le sac des racines des termes obtenues à la première étape et I est la relation binaire qui indique si la racine des termes fait partie du concept structurel. A partir de la relation binaire I binaire, nous construisons tous les contextes formels du treillis qui se représentent comme un diagramme de Hasse. Les contextes sont des regroupements d'objets qui partagent un ou plusieurs attributs. Ce qui nous intéresse ce sont les contextes ayant un seul attribut (terme) et au moins deux objets (concepts structurels). En appliquant cette méthode sur notre exemple, nous obtenons la classification illustrée par la Figure II.11.

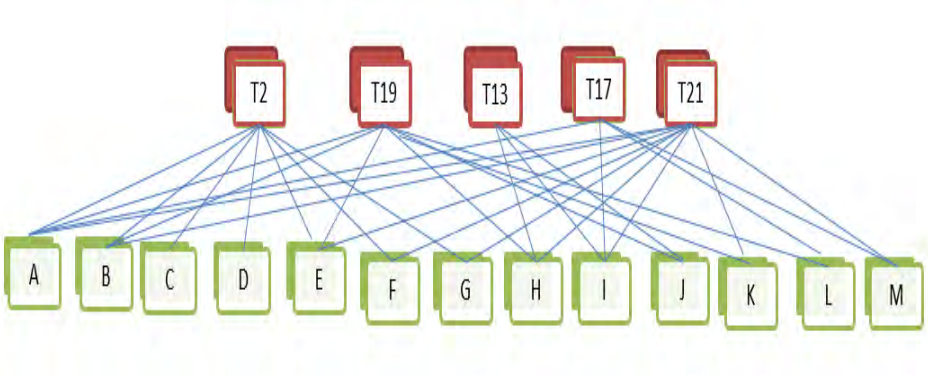


Figure II.11 — Exemple de classification conceptuelle par les treillis de galois

3. *Classification conceptuelle par l'approche RELEVANT* sur les concepts structurels. Nous avons appliqué l'approche RELEVANT avec le regroupement hiérarchique qui permet d'obtenir des groupes disjoints, cela veut dire que chaque concept structurel appartient à plus qu'un seul groupe. Chaque groupe est représenté par un ou plusieurs termes. Cette méthode appliquée sur notre exemple donne les résultats de la Figure II.12.
4. *Croisement des résultats de classification*, nous sélectionnons les mono-termes du treillis qui apparaissent dans les groupes disjoints et pertinents trouvés par RELEVANT. Pour notre exemple les termes pertinents sont $\{T2, T13, T17, T19\}$. Ensuite, nous relions chaque terme avec l'intersection des concepts structurels auxquelles il a été lié par chaque approche. Par exemple $T13$ sera lié à $\{H, I\} = \{H, I, M\} \cap \{H, I, J\}$ et $T17$ sera lié à $\{I, M\} = \{H, I, M\} \cap \{A, I, L, M\}$. Cette intersection peut produire des hiérarchies non-strictes, nous constatons que le concept I est indexé par le terme $T13$ et $T17$. Nous

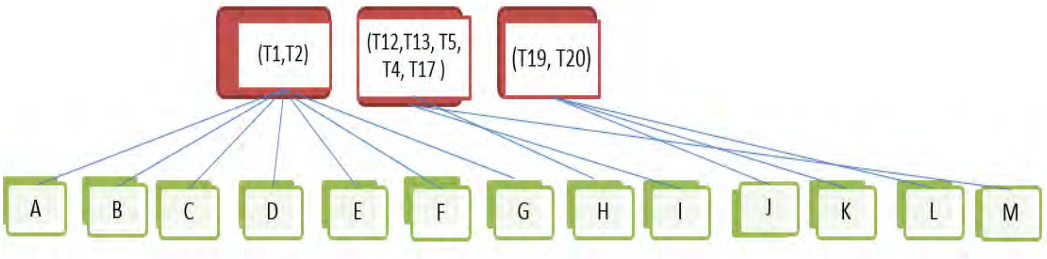


Figure II.12 — Exemple de classification conceptuelle par l'approche RELEVANT

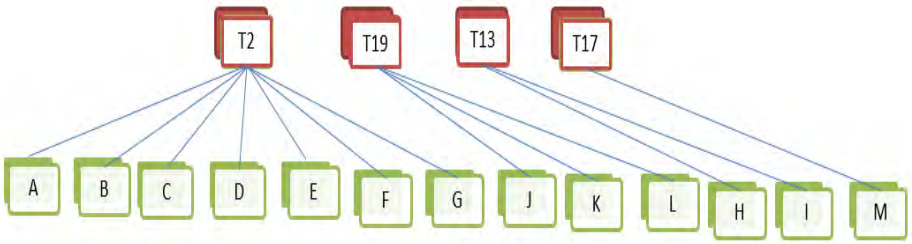


Figure II.13 — Résultat final de classification conceptuelle

utilisons la mesure de similarité de Jaccard afin de rattacher chaque concept au plus proche terme. Le résultat de cette étape appliquée à notre exemple est illustré dans la Figure II.13.

Après la description des différents algorithmes et techniques déployés pour la détection et la reconnaissance des tableaux dans les données ouvertes, nous décrivons dans la section suivante comment nous transformons ces tableaux annotés en graphes.

3.2.5 Transformation des tableaux annotés en graphes

Dans cette section, nous expliquons les étapes à suivre pour transformer les résultats de détection et de reconnaissance en graphes. Ces graphes représentent les schémas des tableaux que nous utilisons pour l'intégration des données tabulaires.

Cette section comportera trois sous-sections. Dans la première sous-section, nous expliquons les transformations des tableaux annotés en graphes de propriétés. Dans la deuxième sous-section, nous montrons comment les graphes de propriétés peuvent se transformer en graphes RDF. Dans la dernière sous-section, nous positionnons les résultats de notre contribution par rapport au nouveau projet de recommandation du W3C.

3.2.5.1 D'un tableau annoté vers un graphe de propriétés

Définition 1. *D'après [Rodriguez et Neubauer, 2010] un graphe de propriétés est une combinaison de graphes orientés, étiquetés, attribués et de multi-graphes. Les arcs sont orientés, les noeuds et les arcs sont étiquetés, des paires de propriétés sous forme clé/valeur sont associées aux noeuds et aux arcs et il peut y avoir plusieurs arcs entre deux noeuds. La Figure II.14 illustre un exemple de graphe de propriétés.*

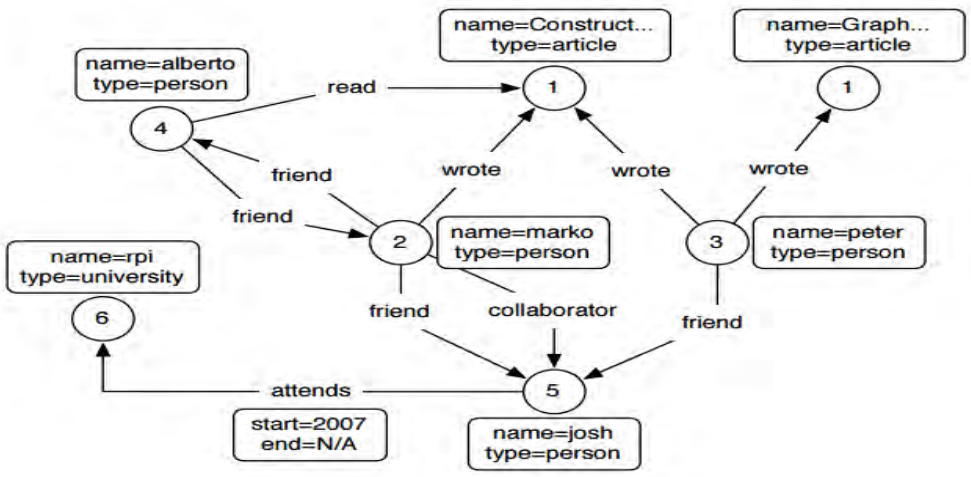


Figure II.14 — Un exemple de graphe de propriétés

Dans un premier temps, nous avons fait le choix de transformer les tableaux annotés vers un graphe selon le modèle de graphe de propriétés pour les raisons suivantes : (1) le graphe de propriétés est l'ancêtre de plusieurs modèles de graphe tels que le graphe orienté, graphe sémantique, graphe RDF, etc ; ceci garanti la généricité de notre approche et un passage facile vers d'autres modèles pour permettre la réutilisation de nos tableaux dans d'autres contextes, et (2) ce modèle nous permet de capitaliser toutes les annotations du tableau et la formalisation que nous avons définies pour les tableaux.

Les règles suivantes permettent la transformation d'un tableau annoté T vers un graphe de propriété $G_P = (V, E)$:

– Dans T, nous avons des composants simples comme les $StructData$ et les $NumData$ et des composants composites tels que les $StubHead$ et les $UnitNumBloc$. Chaque composant simple de C de délimitation DL, FL, DC, FC et ayant les annotations AI, AT et AS se transforme en un noeud identifié de façon unique dans l'ensemble V et ayant les propriétés $\{type, valeur, DL, FL, DC, FC, AI, AT, AS\}$. Le type est le nom du type du composant par exemple $StructData$, la valeur est le contenu de la cellule du composant. Pour les autres propriétés, s'il s'agit d'une chaîne de caractères ou d'un entier il sera repris tel qu'il est, sinon si c'est l'identifiant d'un composant il faut prendre l'identifiant du noeud correspondant à ce composant. Chaque composant composite de C se transforme en un noeud identifié de façon unique dans l'ensemble V et ayant les propriétés $\{type, DL, FL, DC, FC, AI, AT, AS\}$. Les même règles que nous avons ap-

pliquées pour les propriétés des composants simples s'appliquent sur les propriétés des composants composites.

– Dans T, nous avons un ensemble de relations R entre les composants du tableau. Ces relations se transforment en arcs entre les noeuds correspondant à ces composants. Nous avons essentiellement deux types de relations : (1) les relations de spécialisation entre les *StructData* et (2) les relations d'association "indexer" entre les composants composites. Pour les relations de spécialisation, nous créons un arc orienté non-étiqueté entre les noeuds de type *StructData*. Pour les relations d'association "indexer", nous créons des arcs orientés dont la source est le composant qui indexe et la cible est le composant indexé. Par déduction, nous construisons des arcs d'association "indexer" entre les *StructData* et les *NumData*.

La Figure II.15 illustre un extrait du graphe de propriétés du tableau de la Figure II.10.

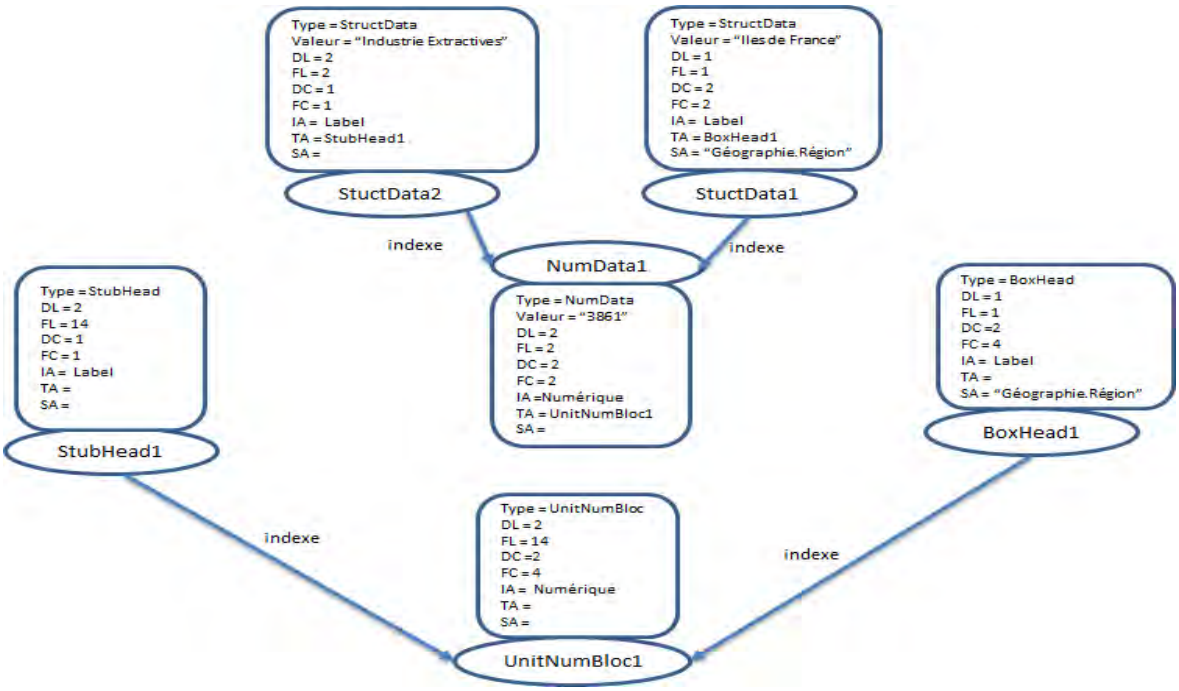


Figure II.15 — Un extrait d'un graphe de propriétés d'un tableau

3.2.5.2 D'un graphe de propriétés vers un graphe RDF

Dans un graphe RDF les données sont organisées en triplets de la forme (s, p, o) qui expriment qu'un sujet s est relié à l'objet o par un arc de propriété p. Les sujets, les propriétés et les objets identifient d'une façon unique, à travers des URI, des entités, des relations ou des concepts. Les objets sont les seuls à pouvoir prendre des valeurs constantes dites littéraux. Pour produire des données RDF, les étapes clés consistent à l'identification des données par des URI et à l'utilisation, dans la mesure du possible, de vocabulaires standardisés par le W3C.

Nous avons évoqué dans la section précédente qu'un graphe de propriétés peut se transformer en un graphe RDF. [Rodriguez et Neubauer, 2010] proposent d'enlever les proprié-

tés des noeuds du graphe de propriétés pour passer à un modèle de graphe étiqueté puis de transformer les labels en URI pour obtenir un graphe RDF. Cette proposition est intéressante mais elle a l'inconvénient de faire disparaître les propriétés. Pour palier à cet inconvénient, nous proposons un autre processus de transformation qui comporte les étapes suivantes :

1. Éclater les propriétés d'un noeud sous format de triplet RDF (s,p,o) où s est le noeud, p est le nom de la propriété et o est la valeur de la propriété. En appliquant cette étape sur le graphe de la Figure II.15, nous obtenons le graphe illustré par la Figure II.16.

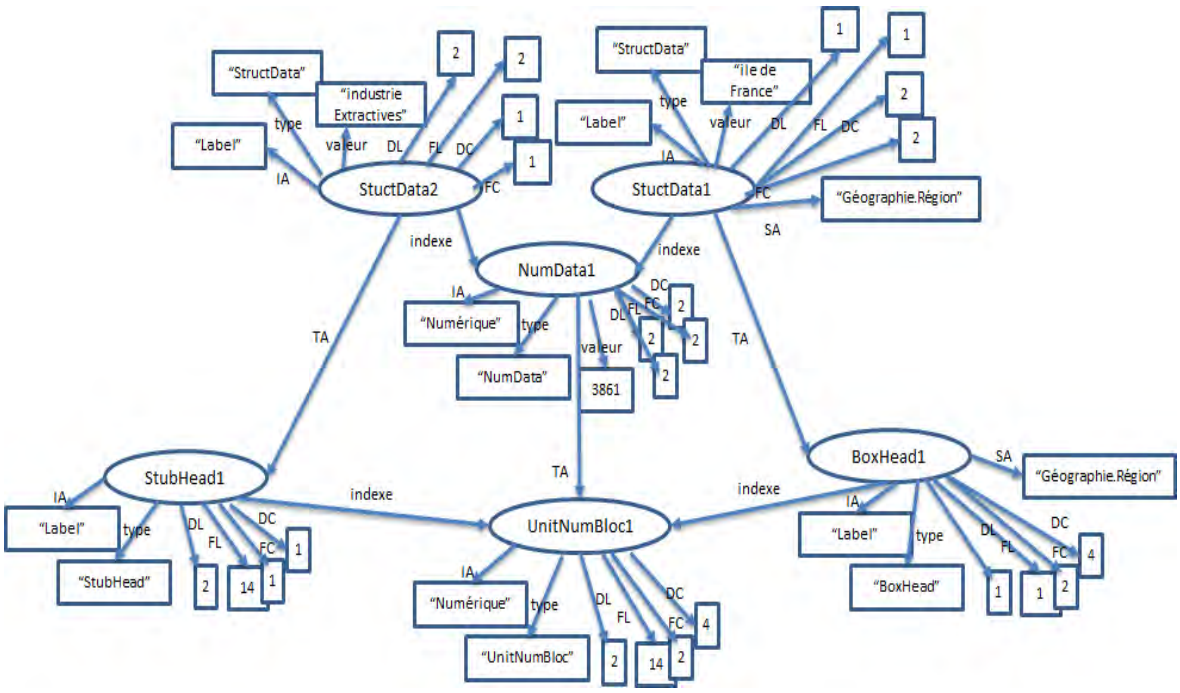


Figure II.16 — Un extrait de graphe de propriétés éclaté

2. Transformer chaque noeud de V qui a un identifiant unique en un URI.

3. Utiliser des vocabulaires standardisés, dans la mesure du possible, pour décrire les données. Nous avons choisi d'utiliser les vocabulaires RDFS [12] et SKOS [13]. Les transformations suivantes sont mises en place :
   - La propriété "type" est remplacée par "rdf :type" ;
   - La propriété "valeur" est remplacée par "rdfs :label" pour les données numériques ;
   - Les données structurelles de type "StructData" deviennent de type "skos :concept" ;
   - La propriété "valeur" est remplacée par "skos :label" pour les données structurelles ;
   - La propriété "is-a" pour la spécialisation entre les données structurelles est remplacée par la propriété "skos :broader".

   La Figure II.17 illustre un extrait du graphe RDF résultant après l'utilisation des parties de vocabulaires standards.

12. http ://www.w3.org/TR/rdf-schema/
13. http ://www.w3.org/TR/swbp-skos-core-spec

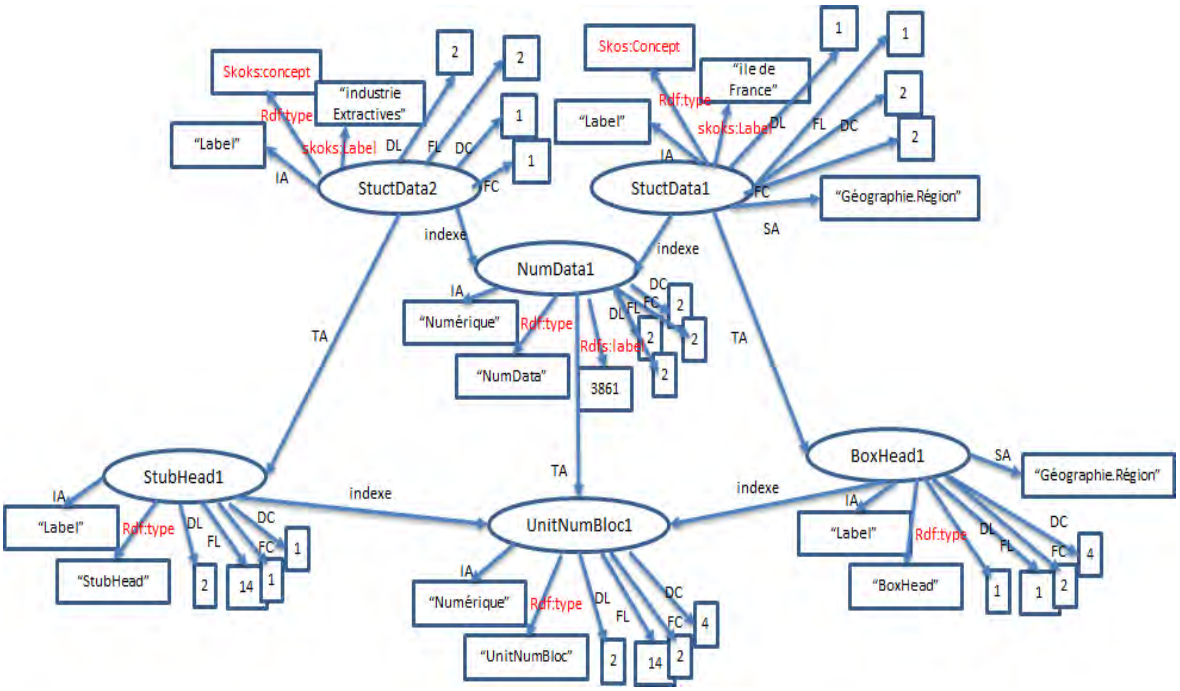


Figure II.17 — Un extrait de graphe RDF

3.2.5.3 Projet de recommandation du W3C

Le W3C consortium travaille depuis avril 2015 sur un projet de recommandation pour décrire et produire des données RDF à partir des données tabulaires. Ce projet comporte (i) un modèle pour les tableaux `http://www.w3.org/TR/2015/CR-tabular-data-model-20150716/#locating-metadata`, (ii) un vocabulaire de méta-données du tableau `http://www.w3.org/TR/2015/CR-tabular-metadata-20150716/` et son vocabulaire CSVW et (iii) un ensemble de procédures et de règles pour convertir des données tabulaires vers des données RDF, ces dernières sont synthétisées dans csv2rdf `http://www.w3.org/TR/2015/CR-csv2rdf-20150716/#bib-tabular-data-model`.

Nous rappelons qu'un projet de recommandation passe chronologiquement par les étapes suivantes : (1) différentes versions de brouillons "Working draft", (2) un appel à voter "last call", (3) un candidat de recommandation "candidate recommandation" , (4) proposition de recommandation "proposed recommandation" et (5) recommandation "recommandation". En juillet 2015, le modèle du tableau, les méta-données et csv2RDF sont candidats pour une recommandation.

Un modèle de tableau est composé de groupes de tableaux, de tableaux, de colonnes, de lignes, de cellules et de types de données. Ce modèle s'applique sur les tableaux relationnels où l'entête du tableau est située à la première ligne. Les méta-données du tableau sont illustrées dans la Figure II.18. La procédure de construction des données RDF du tableau exige qu'il y ait déjà un modèle de tableau annoté qui a été fourni.

Chronologiquement, nos propositions sont antérieures à ce projet. Toutefois, ce que nous avons proposé est tout à fait compatible mais aussi complémentaire aux propositions du

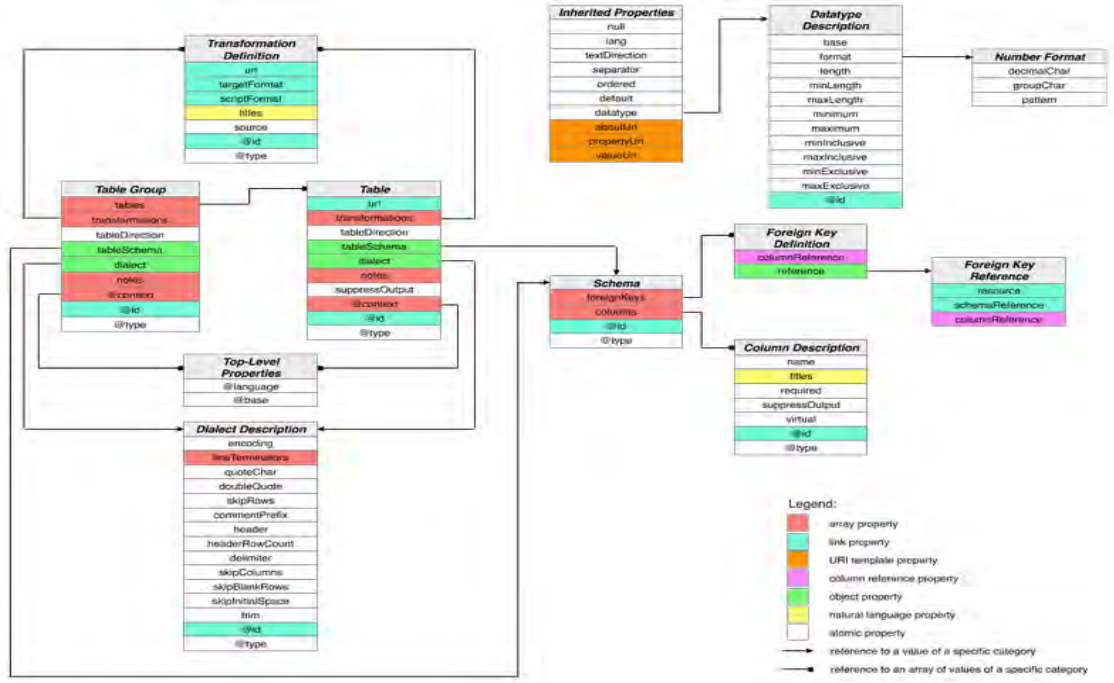


Figure II.18 — Les méta-données du tableau proposées par le W3C

W3C. La complémentarité de nos travaux réside dans la possibilité de traiter des tableaux relationnels et non-relationnels où les entêtes ne sont pas uniquement situées dans la première ligne. En outre, nous avons proposé des solutions automatiques pour le passage de données tabulaires vers des données tabulaires annotées puis vers des graphes RDF. Nos propositions sont aussi compatibles avec les propositions du W3C, il suffirait d'adapter les méta-données, vocabulaire et modèle de tableau dans nos algorithmes.

4 Conclusion

Dans ce chapitre, nous avons présenté une approche pour la détection et la reconnaissance des données ouvertes tabulaires. La détection permet de repérer l'emplacement du tableau et la reconnaissance permet d'analyser le contenu du tableau détecté. La finalité de notre approche est d'obtenir des schémas de tableaux nécessaires pour l'intégration des données provenant de différentes sources. Nous avons choisi de produire des schémas de tableaux sous forme de graphes (graphes de propriétés). Ces graphes ont la possibilité d'être étendus vers des formalismes plus spécifiques comme RDF. Dans les graphes, nous distinguons deux types de données : les données structurelles et les données numériques (les statistiques du tableau).

Notre approche repose sur un nouveau modèle de tableau et sur un workflow d'activités. Chaque activité réalise automatiquement la détection et la reconnaissance d'un composant du tableau. La détection s'appuie sur le modèle de tableau pour identifier l'emplacement du composant concerné. La reconnaissance s'appuie également sur le modèle de tableau pour annoter le composant détecté avec ses propriétés intrinsèques, topologiques et sémantiques.

L'utilisation du modèle de tableau pour la détection permet de pallier le problème d'hétérogénéité structurelle qui caractérise les données ouvertes tabulaires. En effet, l'hétérogénéité structurelle est engendrée par une organisation aléatoire des Open Data par les différents fournisseurs. De même l'utilisation du modèle de tableau pour la reconnaissance permet de décrire les tableaux sans avoir besoin de ressources externes. De ce fait, notre proposition s'applique génériquement sur n'importe quelle source de données ouvertes indépendamment de son domaine d'étude.

Parmi les activités proposées, nous avons mis l'accent sur la découverte automatique de relations hiérarchiques entre les données structurelles. Nous avons pris en considération le problème de hiérarchies complexes [Malinowski et Zimányi, 2006] connu dans les systèmes décisionnels. Cet aspect de notre proposition à un niveau avancé de la démarche ETL vise à simplifier la découverte des hiérarchies du schéma multidimensionnel par des non-experts.

Ces propositions ont été publiées dans le cadre des conférences nationales EDA'13 [Berro *et al.*, 2013] et INFORSID'14 [Berro *et al.*, 2014b] et internationale ADBIS'14 [Berro *et al.*, 2014a].

L'approche proposée dans ce chapitre permet de produire automatiquement ou semi-automatiquement des schémas de tableaux sous forme de graphes. Le chapitre suivant montre une nouvelle méthode pour l'intégration simultanée et automatique de plusieurs graphes de tableaux.

III Intégration holistique des graphes de données ouvertes tabulaires

L'intégration des données issues de multiples sources repose sur un ensemble de correspondances entre les modèles de données de ces sources. La recherche automatique des correspondances est un problème connu dans la littérature sous le nom de problème d'appariement. Notre problématique dans ce chapitre concerne la résolution automatique du problème d'appariement pour intégrer plusieurs graphes de données ouvertes tabulaires.

Nous allons présenter la difficulté du problème d'appariement, les approches proposées dans la littérature, leurs limites face à notre contexte et quelle solution nous proposons pour résoudre ce problème.

1 Introduction

Un panorama des domaines d'application est fourni par [Euzenat et Shvaiko, 2013] en fonction de la résolution du problème d'appariement tels que l'ingénierie d'ontologies, l'intégration d'information, la liaison des données (Linked Data), le partage d'information paire à paire, la composition de services, la communication de systèmes autonomes, l'interrogation du web, etc. Le problème d'appariement est connu aussi sous le nom de **Matching de modèles de données**. L'appariement des modèles de données consiste à déterminer les meilleures correspondances entre les éléments de ces modèles. La Figure III.1 illustre un exemple de correspondances résultantes de la résolution du problème d'appariement entre deux modèles de documents. Les modèles de données varient du moins expressif au plus expressif en termes, hiérarchies ad-hoc, thésaurus, XML, schémas de bases de données, ontologie [Euzenat et Shvaiko, 2013].

L'intégration des données tabulaires se situe dans l'un des domaines d'application cités ci-dessus en fonction de la nature du modèle de données. Par exemple, si le modèle est une ontologie alors l'intégration des données tabulaires fait partie du cadre applicatif d'ingénierie d'ontologies. Si les modèles sont des hiérarchies ad-hoc ou des schémas de bases de données alors l'intégration des données tabulaires fait partie du cadre applicatif d'intégration des données dans un système d'information.

Le contexte de nos propositions se situe au niveau de l'intégration des données tabulaires dans un système d'information. En particulier, nous intégrons les graphes de structures hiérarchiques des données structurelles extraites des tableaux. Ces graphes sont moins expressifs que les ontologies puisqu'ils contiennent uniquement des labels et des relations hiérarchiques entre ces labels. En plus, puisque les tableaux à intégrer proviennent de plusieurs fournisseurs, les données structurelles sont sémantiquement hétérogènes.

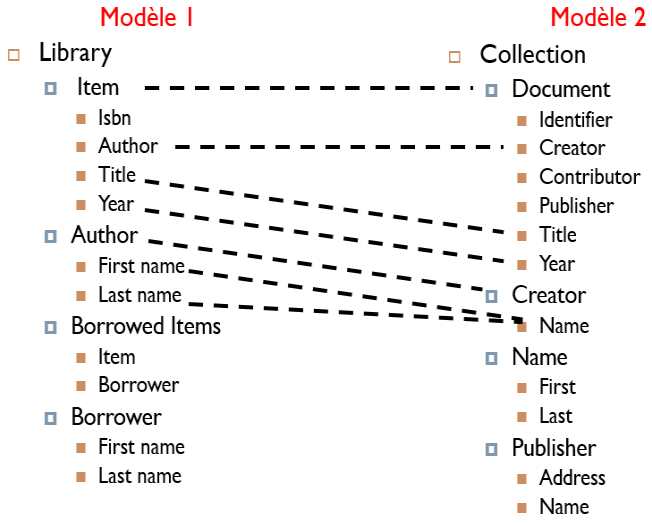


Figure III.1 — Exemple de correspondances résultantes de l'appariement du modèle 1 et modèle 2

Notre objectif est d'obtenir automatiquement une solution, unique et optimale, d'appariements holistiques (plusieurs graphes en même temps) pour des graphes de structures hiérarchiques. Nous souhaitons que la solution fournie soit également de structure hiérarchique afin de faciliter la définition du schéma multidimensionnel à partir de cette dernière. Nous ambitionnons aussi de faire face à l'hétérogénéité sémantique de sources ouvertes provenant de multiples fournisseurs.

Ce chapitre est organisé en deux parties. Dans la première partie, nous expliquons le problème d'appariement et ses spécificités. Ensuite, nous décrivons et discutons les travaux de la littérature pertinents par rapport à notre contexte. Dans la deuxième partie, nous abordons en détail notre proposition appliquée aux graphes de données ouvertes tabulaires.

2 État de l'art : Appariement des modèles de données

2.1 Le problème d'appariement

Le problème d'appariement consiste à déterminer l'ensemble des correspondances entre les éléments de modèles de données. Le rôle d'une approche d'appariement est de résoudre le problème d'appariement entre N modèles de données en entrée. Si $N = 2$, nous parlons d'approches d'appariement par paire et si $N \geq 2$ nous parlons d'approches d'appariement holistique.

Comme le montre la Figure III.2, une approche d'appariement peut dépendre de facteurs autres que les modèles de données à savoir : des correspondances de référence, des ressources externes (thésaurus, ontologies,..) et des paramètres de configuration tels que des seuils, des poids, etc.

L'ouverture de la boite noire d'une approche d'appariement révèle un workflow général [Rahm, 2011] de quatre étapes, voir Figure III.3 : (1) pré-traitement des modèles de données,

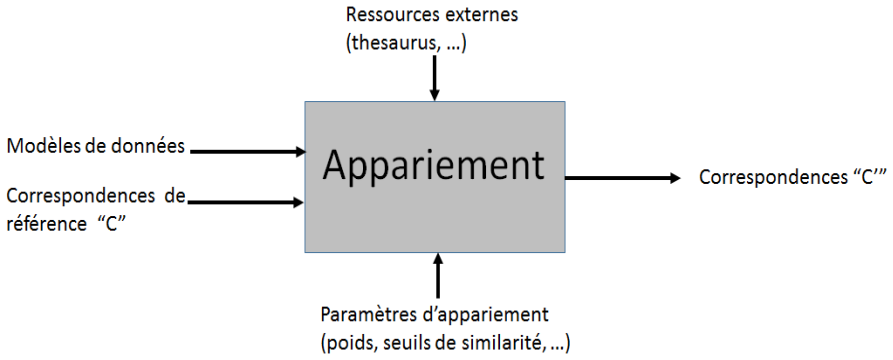


Figure III.2 — Les entrées/sorties d'une approche d'appariement [Shvaiko et Euzenat, 2005]

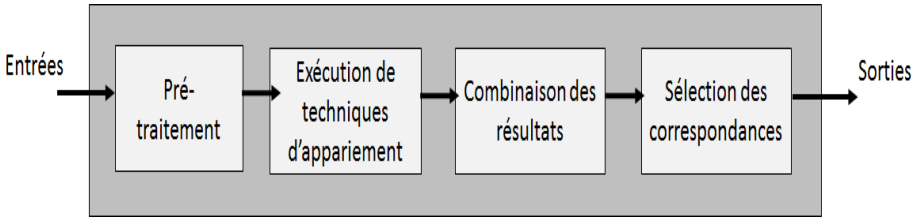


Figure III.3 — Un workflow général du processus d'appariement

(2) exécution de(s) technique(s) d'appariement, (3) combinaison des résultats, (4) sélection des correspondances. Ce workflow nous montre les grandes étapes d'une telle approche mais ces étapes ne sont pas figées puisque cela dépend des stratégies adoptées par chaque approche. Ces techniques et ces stratégies seront expliquées dans les sections qui suivent.

Une correspondance est le résultat de comparaison d'une paire de modèles. [Shvaiko et Euzenat, 2005] et [Euzenat *et al.*, 2004] ont définit une correspondance sous la forme de $C = \langle id, e, e', n, R \rangle$ tel que :

- id est l'identifiant de la correspondance.
- e et e' sont les éléments du premier et du deuxième modèle.
- n est la valeur de similarité entre e et e', elle appartient à l'intervalle $[0, 1]$.
- R est la relation entre e et e' qui peut être : équivalence $=$, plus général $\sqsupseteq$, disjointure $\perp$, intersection $\sqcap$.

Nous parlons de correspondances simples, de cardinalité 1 : 1 si les éléments e et e' apparaissent une seule fois dans l'ensemble des correspondances. Dans le cas contraire, par exemple si l'élément e apparaît dans deux correspondances différentes C et C' alors nous parlons de correspondances complexes de cardinalité $n : m$. Si nous comparons N schémas dans le cadre d'un appariement holistique, il faut chercher les regroupement de correspondances générés pour les différentes paires de schémas.

2.1.1 Les techniques d'appariement

Dans la littérature, les techniques d'appariement ont été classifiées dans trois livres [Euzenat et Shvaiko, 2007] [Rahm, 2011] [Euzenat et Shvaiko, 2013] et trois revues

[Rahm et Bernstein, 2001] [Shvaiko et Euzenat, 2005] [Bernstein *et al.*, 2011]. Nous avons retenu la plus récente classification proposée par [Euzenat et Shvaiko, 2013] pour présenter succinctement cette large variété de techniques.

Dans la Figure III.4, nous avons des rectangles qui correspondent aux techniques d'appariement et deux arborescences. L'arborescence supérieure est une classification des techniques par rapport à la granularité (niveau élémentaire ou niveau structurel) et l'interprétation (sémantique, syntaxique) des parties traitées dans les modèles. L'arborescence inférieure est une classification par rapport à l'origine (basée sur le contenu ou sur le contexte) et la nature des parties traitées dans les modèles (sémantique, syntaxique, terminologique, structurel, extensionnel). Nous illustrons les techniques à travers la classification supérieure, qui sera retenue pour la comparaison des approches d'appariement.

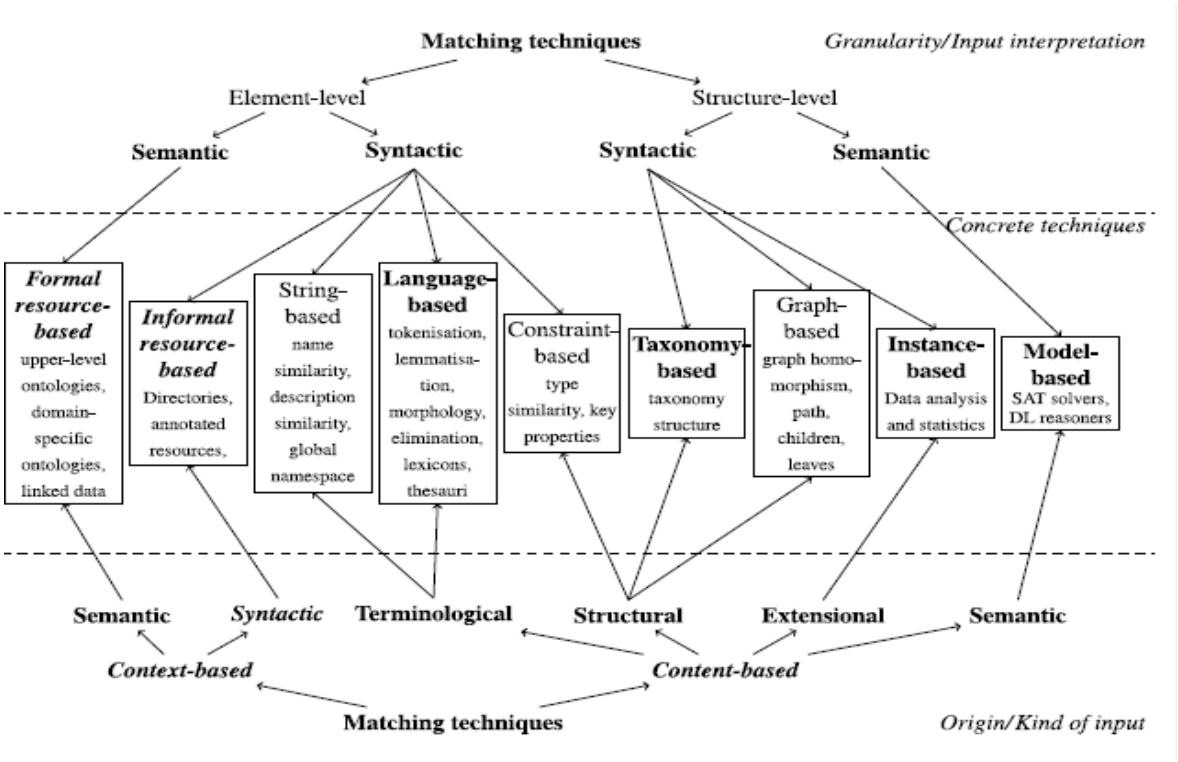


Figure III.4 — La classification des techniques d'appariement de [Euzenat et Shvaiko, 2013]

Les techniques élémentaires calculent les correspondances en analysant les éléments et/ou leurs instances en isolation c'est à dire en ignorant complètement les relations entre ces éléments [Euzenat et Shvaiko, 2013]. Ces techniques sont classifiées comme suit :

- Techniques élémentaires syntaxiques :
  - Les techniques basées sur le langage (Language-based) sont les premières techniques à appliquer sur les noms des éléments. Il s'agit de différentes opérations telles que le découpage en sacs de termes (tokenisation), la réduction d'un terme en sa racine (lemmatisation), l'identification de morphologie des termes (verbe, nom, etc..), l'élimination des termes vides. Les patrons utilisés par ces opérations dépendent du langage (anglais, français..).

- Les techniques basées sur les termes (String-based) se présentent sous la forme de fonctions, dites aussi mesure/distance de similarité, qui calculent une similarité, appartenant à l'intervalle $[0, 1]$, entre les noms des éléments. Ces techniques sont classifiées à leur tour en deux types d'après la récente revue de [Sun *et al.*, 2015] :
  - Les techniques basées sur les caractères (Character-based) s'appliquent entre deux termes. Elles dépendent uniquement de l'apparence et de l'enchaînement des séquences de caractères dans les termes. Les plus connus sont Préfixe, Suffixe, Edit-distance (ou Levenstein) [Levenshtein, 1966], Monge-Elkan [Monge et Elkan, 1996], Jaro-Winkler [Winkler, 1990], I-SUB [Stoilos *et al.*, 2005], N-gram.
  - Les techniques basées sur les jetons (Token-based) s'appliquent entre deux sacs de termes. Les plus connus sont Jaccard [Jaccard, 1912], Monge Elkan de niveau 2 et Soft TF-IDF.
- Les techniques basées sur les contraintes (Constraint-based) cherchent une égalité entre les contraintes des éléments par exemple le type d'élément, la cardinalité, les clés primaires, etc.
- Les techniques basées sur des ressources informelles (Informal resource-based) s'appuient sur le contexte dans lequel les éléments du modèle figurent. Par exemple, si les éléments sont prises de pages wikipédia alors les ressources informelles peuvent être les annotations de ces éléments ou l'organisation des pages.

- Techniques élémentaires sémantiques :
  - Les techniques basées sur des ressources formelles (Formal resource-based) consistent à utiliser des ressources externes formelles et sémantiques pour mesurer les distances de similarités entre les éléments des modèles en entrée et les éléments originaires d'une ou de plusieurs ressources externes. Ces ressources peuvent être des correspondances de référence, des thésaurus, des ontologies de domaine, des ontologies de haut-niveau, etc. A titre d'exemple, plusieurs distances de similarités dans la littérature se basent sur le thésaurus Wordnet [Miller, 1995] : Wup [Wu et Palmer., 1994], Lin [Lin, 1998], LCH [Leacock et Chodorow, 1998], LESK [Banerjee et Pedersen, 2002], etc [1].

Les techniques structurelles calculent les correspondances en analysant comment les éléments et/ou les instances apparaissent ensemble dans une même structure. Ces techniques sont classifiées comme suit :

- Techniques structurelles syntaxiques :
  - Les techniques basées sur les graphes (Graph-based) transforment les modèles de données en graphes étiquetés, à partir desquels le problème d'appariement peut se réduire à un problème connu tel que le problème d'homomorphisme ou le problème de couplage de poids maximal. Il est aussi possible d'utiliser les positions des éléments dans les graphes pour calculer des mesures de similarité structurelle entre ces éléments par exemple en analysant la position de leurs parents, leurs feuilles, leurs frères ou leurs ancêtres [Agreste *et al.*, 2014].
  - Les techniques basées sur les taxonomies (Taxonomy-based) représentent un cas particulier des techniques basées sur les graphes. Une taxonomie est composée de

1. https ://code.google.com/p/ws4j/

noeuds étiquetés et reliés par des relations de spécialisation. L'intuition de ces techniques est que si des éléments sont similaires alors leurs voisinages le sont également.
  - Les techniques basées sur les instances (instance-based) exploitent les instances des éléments du modèle (par exemple les instances d'une classe dans une ontologie) pour décider s'il y a une correspondance entre les éléments. Ils se représentent souvent sous la forme d'algorithmes d'analyse statistique des données.
- Techniques structurelles sémantiques :
  - Les techniques basées sur les modèles (Model-based) cherchent à vérifier s'il y a des interprétations sémantiques valides entre les éléments, par exemple sous la forme d'un problème de satisfiabilité SAT. Ces modèles permettent de déduire les quatre types de relation R entre les éléments.

2.1.2 Les stratégies d'appariement

A chaque étape du processus d'appariement, il peut y avoir différentes stratégies possibles.

Dans l'exécution des techniques d'appariement, nous avons pu cerner d'après [Euzenat et Shvaiko, 2013] et [Rahm, 2011] quatre stratégies d'exécution de techniques d'appariement comme suit :

- La stratégie séquentielle consiste à appliquer successivement différentes techniques d'appariement. Chaque technique dépend des modèles de données et des résultats de la technique qui l'a précédée. La Figure III.5 illustre le principe de cette stratégie.

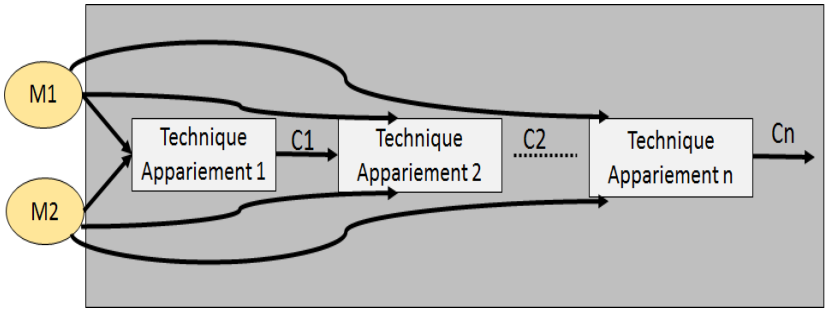


Figure III.5 — La stratégie de combinaison séquentielle

- La stratégie parallèle consiste à exécuter sur les modèles de données (après prétraitement) différentes techniques d'appariement, puis agréger les résultats de ces techniques pour obtenir un seul ensemble de correspondances. La Figure III.6 montre le principe de cette stratégie. [Euzenat et Shvaiko, 2013] distinguent deux sous-types de stratégie de composition parallèle : (1) la composition parallèle hétérogène où les modèles d'entrée sont fragmentés, chaque technique prend deux fragments de données de même type puis une agrégation se fait entre les résultats de toutes les techniques (2) la composition parallèle homogène où les modèles sont passés en entier à chaque technique puis une agrégation se fait entre ces différents résultats.
- La stratégie itérative consiste à appliquer la même technique d'appariement plusieurs fois jusqu'à un certain point fixe pour arrêter les itérations. C'est un cas particulier

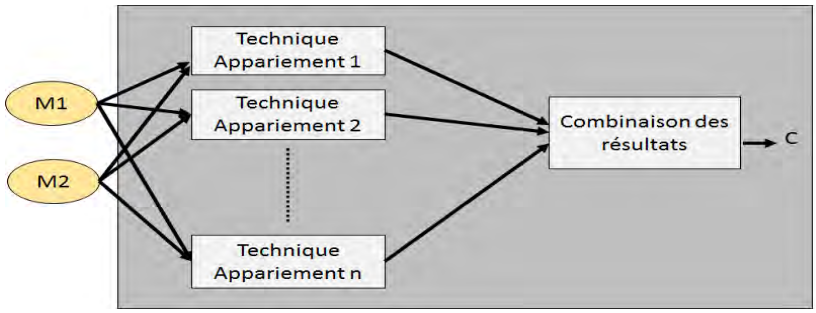


Figure III.6 — La stratégie de combinaison parallèle

de la stratégie parallèle puisque à chaque itération le calcul des nouvelles correspondances va dépendre du calcul précédant. La Figure III.7 illustre le principe de cette stratégie.

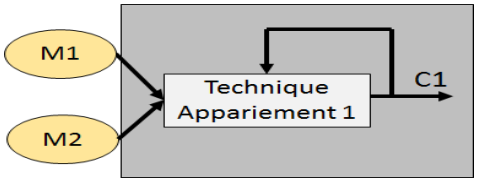


Figure III.7 — La stratégie de combinaison itérative

– La stratégie mixte est un mélange de toutes les autres stratégies dans n'importe quel ordre, tous les cas de figure sont possibles. Par exemple, nous pouvons appliquer une première technique, puis itérer sur une deuxième technique, puis appliquer en parallèle d'autres techniques, etc.

Dans la combinaison des résultats de techniques d'appariement, il y a trois stratégies :

– La stratégie de pondération dans laquelle des fonctions mathématiques sont utilisées pour combiner les résultats d'agrégation. Ces fonctions peuvent être min, max, produit pondéré, distance de Minkowski, somme pondérée, etc. La plupart de ces fonctions dépendent d'un poids qui doit être donné en entrée par l'utilisateur ou déduit par apprentissage.
– La stratégie de vote correspond à une mise en place d'un système de vote entre les résultats des différentes techniques d'appariement. Ces stratégies peuvent être un vote par majorité, un vote par majorité pondérée, etc.
– La stratégie d'argumentation consiste à faire une négociation entre deux ou plusieurs agents où chacun argumente les correspondances qu'il doit défendre. Cette stratégie peut être mise en place par des systèmes multi-agents.

Dans la sélection des correspondances, il y a trois stratégies possibles :

– La stratégie de sélection par seuil consiste à sélectionner les correspondances dont la valeur de similarité n est supérieure à un certain seuil. Il y a différents types de seuil : (1) le seuil strict (Hard threshold) correspond à une valeur donnée x, (2) le seuil delta (Delta threshold) est la différence entre la plus grande valeur de similarité et une valeur donnée x, (3) le seuil d'écart (Gap threshold) retient les correspondances dans l'ordre décroissant de leurs valeurs de similarité jusqu'à ce que la différence entre ces dernières devienne supérieure à une valeur donnée x, (4) le seuil proportionnel

(Porportional threshold) correspond au pourcentage de correspondances ayant la plus grande similarité, (5) le seuil de pourcentage correspond à une sélection de correspondances dont la valeur de similarité est au-dessus des x% valeurs de similarité des autres correspondances. Il peut aussi y avoir d'autres techniques statistiques pour l'apprentissage des seuils.

– La stratégie de sélection par "points forts ou faibles" consiste à appliquer une fonction qui s'appelle sigmoid avec un paramètre de pente pour découper les correspondances en zones de fortes et de faibles mesures de similarité. Les correspondances doivent être sélectionnées dans les zones supérieures.

– La stratégie de sélection par résolution du problème de mariage stable ou du problème de couplage de poids maximal. La résolution du problème de mariage stable permet d'extraire des correspondances de telle sorte qu'une entité figure dans au plus une seule correspondance. La résolution de ce problème se fait généralement par des algorithmes gloutons. Le problème de couplage de poids maximal consiste à chercher le meilleur ensemble de correspondances maximisant la somme de leurs valeurs de similarité. La résolution de ce problème peut se faire par des algorithmes gloutons, heuristiques ou par la programmation linéaire. Nous notons que le problème de mariage stable retourne un optimum local et le problème de couplage de poids maximal retourne un optimum global [Euzenat et Shvaiko, 2013]. Un optimum global est la meilleure solution dans l'espace de toutes les solutions possibles alors qu'un optimum local est une solution meilleure sur une partie de l'espace des solutions. L'optimum global est meilleur que l'optimum local et dans certains cas les deux peuvent coïncider. La Figure III.8 montre cette différence d'une façon générale et la Figure III.9 montre un exemple illustrant la différence entre la solution d'un mariage stable et la solution d'un couplage.

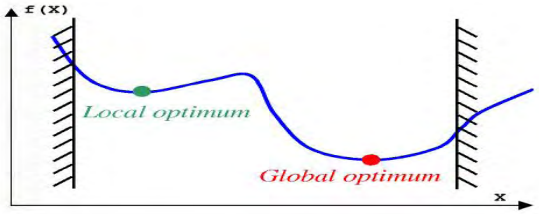


Figure III.8 — Solution optimale locale vs solution optimale globale

Nous soulignons que dans la phase de pré-traitement la plupart des approches transforment les modèles de données en entrée en un modèle de représentation interne de données [Agreste *et al.*, 2014]. Ce modèle de représentation interne peut être un tableau de termes, un arbre, une forêt, un graphe orienté acyclique, un graphe orienté / non-orienté étiqueté/non-étiqueté.

Après avoir donné un aperçu sur les diverses possibilités qui peuvent être déployées pour la résolution d'un problème d'appariement, nous étudions dans la suite les approches d'appariement les plus pertinentes par rapport à notre contexte.

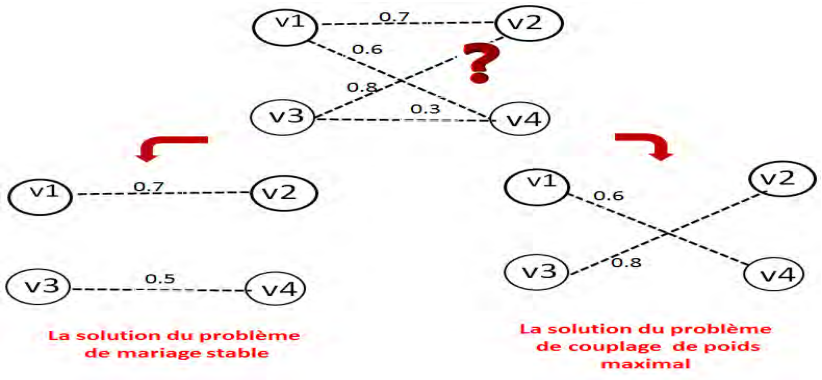


Figure III.9 — La solution du problème de mariage stable vs la solution du problème de couplage

2.2 Étude des approches d'appariement

Dans la littérature, une centaine d'approches connues ont été synthétisée par [Euzenat et Shvaiko, 2007] [Rahm, 2011] [Euzenat et Shvaiko, 2013] dont une majorité par paire et une minorité holistique. Dans un premier temps, nous décrivons et discutons les approches holistiques. Dans un second temps, nous décrivons et discutons les approches par paires. Enfin, nous présentons les approches d'intégration de données tabulaires qui ont abordé le problème d'appariement.

2.2.1 Les approches holistiques

Dans la littérature, les propositions pour résoudre le problème d'appariement holistique pour $N \geq 2$ modèles de données sont peu nombreuse par rapport aux approches par paire. Nous synthétisons la description de ces approches puis nous discutons leurs limites par rapport à notre contexte.

2.2.1.1 Étude des travaux

[He *et al.*, 2004][He *et al.*, 2005] ont proposé l'outil Wise-Integrator pour l'unification de différents formulaires web. Cet outil se base sur deux regroupements : le premier est un regroupement exact qui est suivi par un regroupement approximatif. Dans le regroupement exact, ils regroupent dans un même cluster tous les attributs de différents formulaires ayant exactement le même nom. Dans le regroupement approximatif, ils élargissent chaque cluster en lui rajoutant d'autres attributs dont la similarité est proche. Pour cela, ils utilisent des techniques d'appariement élémentaires approximatives comme edit-distance et cosine sur les types des attributs (extraits par la ressource externe Wordnet). Par la suite, ils filtrent les paires d'attributs dont la similarité est inférieure à un seuil donné. Enfin, ils déterminent l'attribut le plus représentatif d'un cluster (RAN) qui sera utilisé dans l'interface unifiée.

[Su *et al.*, 2006b] [Su *et al.*, 2006a] ont proposé une approche holistique, nommée PSM (Parallel Schema Matching), pour l'intégration de plusieurs formulaires web appartenant au même domaine. Dans cette approche, les auteurs n'utilisent pas de technique d'appariement

structurel puisque le modèle de données interne est une liste d'attributs pris des formulaires web (sans aucune structure). Dans la phase de pré-traitement, les auteurs construisent : (1) une liste de paires d'attributs synonymes qui sont des attributs rarement co-présents dans le même formulaire, seuls les attributs ayant un score (statistique) supérieur à un seuil donné sont gardés ; (2) les groupes d'attributs qui sont souvent co-présents dans le même formulaire, seuls les groupes ayant un score de groupement supérieur à un seuil donné sont gardés et (3) des schémas parallèles qui sont tous les paires de combinaisons de deux formulaires parmi les N formulaires en entrée. A partir de la liste des attributs synonymes et les schémas parallèles, ils calculent la mesure de similarité qui correspond à un score statistique (similaire au coefficient de Dice) évaluant la co-présence des attributs dans les différents schémas parallèles. La sélection des correspondances se fait à travers un algorithme glouton qui choisit les attributs similaires (les correspondances) ayant un score supérieur à un certain seuil. Comme les attributs synonymes appartiennent à des groupes d'attributs, cette approche trouve des correspondances complexes.

[He et Chang, 2006] ont proposé un framework, nommé DCM (Dual Correlation Mining), pour l'appariement holistique des interfaces web d'un même domaine dans l'objectif de construire une interface unifiée. Le modèle de données interne est la liste des attributs des formulaires web. Les auteurs appliquent quelques techniques d'appariement élémentaires telles que la normalisation des attributs, la reconnaissance de type ou le regroupement des attributs par la similarité de leurs termes. Ensuite, ils découvrent les groupes d'attributs positivement corrélés (qui co-existent dans la même interface) et parmi ces groupes ils retiennent ceux qui sont négativement corrélés (ceux qui ont des attributs similaires). Ces groupes sont des correspondances complexes candidates mais qui peuvent être conflictuelles. Les auteurs ordonnent les correspondances candidates et appliquent un algorithme glouton pour la sélection des correspondances complexes non-conflictuelles les mieux classées.

[Pei *et al.*, 2006] ont proposé une approche holistique basée sur les techniques de regroupement (Clustering) pour l'appariement des attributs de N formulaires web du même domaine. Cette approche permet de déterminer les correspondances simple qui se présentent sous la forme d'un groupe (cluster) d'attributs où chaque attribut provient d'un schéma (les attributs du formulaire) différent des schémas des autres attributs. Les auteurs ont proposé un algorithme de clustering des attributs qui est une sorte d'algorithme k-means incrémental avec k non fixé à l'avance. Ils ont aussi proposé une fonction de critères qui combine deux techniques élémentaires : la première est basée sur les types des attributs (constraint-based) et la deuxième est la distance cosinus entre les jetons (token-based). Sur ces groupes, ils appliquent d'autres algorithmes pour le raffinement des correspondances en distinguant les correspondances stables et non-stables.

[Saleem *et al.*, 2007] ont proposé l'outil PORSCHE (Performance Oriented Schema Matching) pour l'intégration de plusieurs schémas XML qui décrivent le domaine des livres. Ces schémas sont transformés en un modèle interne d'arbre. Dans la phase de pré-traitement, les auteurs calculent le contexte de chaque noeud qui correspond à son ordre et sa portée selon un parcours en profondeur dans un arbre. Ensuite, ils ont appliqué des techniques d'appariement élémentaires (language-based et formal resource-based, où ils ont proposé leur propre ressource) et ils fusionnent les labels similaires dans une même liste. Ceci est la par-

tie holistique de cette approche puisque tous les labels ont été comparés en même temps. La deuxième partie de découverte des correspondances se fait par un algorithme à base d'heuristiques qui compare les paires de schémas. Les auteurs ont supposé que le schéma qui a le plus de noeuds représente la version initiale du schéma intégré V_m. Ils comparent de façon incrémentale les noeuds de chaque schéma avec les noeuds du schéma V_m. Ils doivent vérifier si le label du noeud figure dans la liste des labels similaires et si le contexte du noeud assure une structure intégrée correcte (tree-mining) afin de pouvoir ajouter une correspondance simple entre les noeuds.

[Benharkat *et al.*, 2007] ont proposé la plateforme PLASMA (Platform for Large Schema Matching) pour l'appariement de schémas XML dans le domaine E-Business. Le modèle de données interne utilisé est un arbre. Dans la phase de pré-traitement, les auteurs décomposent les arbres des différents schémas en appliquant holistiquement l'algorithme de Dryade [Termier *et al.*, 2004] afin d'extraire les sous-arbres fréquents. Ensuite, ils appliquent des techniques d'appariement élémentaires (string-based, language-based et formal-resource-based) sur les sous-arbres fréquents pour découvrir les sous-arbres les plus pertinents. Ils ont aussi calculé des similarités structurelles avec une version avancée de l'algorithme EXSMAL [Chukmol *et al.*, 2005] entre toutes les paires des sous-arbres de façon indépendante. Enfin, ils ont combiné les similarités structurelles et élémentaires avec des fonctions de combinaisons en donnant des poids et ils ont sélectionné les correspondances au-dessus d'un certain seuil de similarité.

[Grütze *et al.*, 2012] ont proposé l'approche HCM (Holistic Concept Matching) pour la découverte de correspondances entre des concepts pris du LOD (Linked Open Data). Dans la phase de pré-traitement, ils construisent des WCF (Wikipedia concept Forest) pour chaque concept des différents modèles de données RDF (ils n'ont pas pris les relations entre les concepts, donc uniquement des listes de concepts). En effet, chaque concept devient une liste de termes par les techniques élémentaires de type language-based. Ils ont ensuite cherché ces termes dans wikipédia, retenu les top d pages et ils ont construit un arbre à d niveaux et dans chaque niveau ils ont placé les termes qui appartiennent à la page de niveau i. Dans un deuxième temps, ils ont affecté des termes pour chaque forêt WCF (ce sont les 10 termes qui ont les meilleurs scores TF-IDF), puis ils comparent les termes de chaque paire de WCF par la distance de Jaccard (token-based) qui doit être supérieure à un seuil donné et groupent les WCF de termes similaires dans un même groupe. Par la suite, ils calculent pour chaque groupe et entre toutes les paires de WCF une mesure de similarité structurelle qui consiste au recouvrement de deux forêts WCF. Enfin ils sélectionnent les correspondances consistantes au-dessus d'un certain seuil.

2.2.1.2 Synthèse et limites des travaux

Le Tableau III.1 synthétise les différentes caractéristiques des approches décrites ci-dessus. Notre premier constat concerne les travaux de [He *et al.*, 2004], [Su *et al.*, 2006b], [He et Chang, 2006] et [Pei *et al.*, 2006] qui ont tous proposé des approches holistiques plus ou moins différentes pour apparier les attributs de formulaires web. Le fait qu'ils étudient le même domaine sans structure, leur a permis de traiter holistiquement les attributs et les regrouper soit avec des analyses statistiques, des mesures de corrélation ou des algorithmes de

regroupement. Le regroupement a l'avantage de faciliter l'identification de correspondances complexes [Su *et al.*, 2006b] et [He et Chang, 2006], par contre dans notre contexte nous ne pouvons pas appliquer ces approches puisqu'elles n'impliquent pas des structures et elles exigent que les données appartiennent au même domaine. A première vue, l'approche de [Grütze *et al.*, 2012] nous fait penser aux travaux holistiques sur les formulaires. Cependant, ils ont trouvé un moyen pour structurer les termes des concepts en WCF. Donc cette approche devient similaire aux approches de [Benharkat *et al.*, 2007] et de [Saleem *et al.*, 2007]. Même si dans les détails ces approches utilisent des stratégies différentes, ils ont tous comparé les N schémas par paires avec des techniques structurelles et élémentaires, puis ils ont combiné les résultats et sélectionné les correspondances finales en utilisant un seuil. Aucune de ces approches ne considère le problème holistiquement, c'est à dire en comparant simultanément les résultats des différentes combinaisons des différentes paires étudiées. La seule approche que nous considérons véritablement holistique est [Su *et al.*, 2006b] avec leurs schémas parallèles, néanmoins cette approche comme nous l'avons déjà discutée est réductrice de notre contexte.

Tableau III.1 — Comparaison des approches d'appariement holistique

Approche	Modèle données	Modèle Représ. Interne	Type corresp.	Domai. étude	Dépen. Seuil	Ress. externe	Méthodes utilisées	Techniques d'appariement	
								Niv. élémentaire	Niv. structurel
[He *et al.*, 2004]	formulaire web	listes d'attributs	simple	oui	oui	oui	clustering	formal resource-based, character-based, token-based, language-based, constraint-based	-
[Su *et al.*, 2006b]	formulaire web HTML	liste d'attributs	complexe	oui	oui	non	analyse statistique	string-based	-
[He et Chang, 2006]	formulaire web HTML	liste d'attributs	complexe	oui	oui	non	correlation mining	string-based, langage-based	-
[Pei *et al.*, 2006]	formulaire web HTML	liste d'attributs	simple	oui	oui	non	clustering	constraint-based, token-based	-
[Saleem *et al.*, 2007]	XML	arbre	simple	oui	non	oui	incrémentale par paire	language-based, formal resource-based	tree-mining
[Benharkat *et al.*, 2007]	XML	arbre	simple	oui	oui	oui	décomposition, incrémentale par paire	string-based, formal resource-based, language-based, constraint-based	EXSMAL
[Grütze *et al.*, 2012]	concepts RDF	forêt (WCF)	simple	non	oui	oui	clustering	string-based, language-based	mesure de chevauchement de groupe

2.2.2 Les approches par paire

Les approches d'appariement par paire sont très nombreuses dans la littérature. Comme nous nous intéressons à l'optimisation combinatoire pour résoudre le problème d'appariement, nous avons sélectionné les travaux [Euzenat et Valtchev, 2004] [Yatskevich, 2008] [Niepert *et al.*, 2010] [Melnik *et al.*, 2002] qui ont réduit une ou plusieurs phases du processus d'appariement en un problème d'optimisation combinatoire. Nous avons aussi sélectionné des travaux de référence en appariement de schémas en particulier les deux approches [Aumueller *et al.*, 2005] et [Melnik *et al.*, 2002]. Enfin, nous avons choisi de discuter les travaux de thèse de [Duchateau, 2009] qui est le premier à proposer un banc d'essai orienté schémas [Duchateau et Bellahsene, 2014] pour l'évaluation des approches d'appariement de paire de schémas XML. Dans le dernier chapitre de ce manuscrit, nous présentons les résultats de notre approche sur ce banc d'essai.

2.2.2.1 Étude des travaux

[Melnik *et al.*, 2002] ont proposé l'approche Similarity Flooding (SF), implémentée dans l'outil RONDO, pour résoudre le problème d'appariement entre deux modèles de données qui peuvent être des schémas relationnels et XML. Cette approche transforme n'importe quel modèle de données en un graphe étiqueté orienté. Ces deux graphes sont fusionnés dans un graphe de paires connectées de la façon suivante : si dans le graphe G_A il y a une relation r1 du noeud A vers le neoud A1 et si dans le graphe G_B il y a une relation r1 du noeud B vers le noeud B1 alors ils créent une relation r1 du noeud (A,B) vers le noeud (A1,B1) dans le graphe de paires. Pour chaque paire d'éléments, ils calculent leurs similarités élémentaires par des mesures de préfixe et suffixe. Les similarités élémentaires est l'initialisation de l'algorithme de propagation de similarité. Cet algorithme propage itérativement, jusqu'à un certain point fixe, le poids des arcs pour augmenter la mesure de similarité des paires adjacentes à chaque paire d'éléments. Une fois ces mesures de similarité structurelles sont calculées, les auteurs appliquent le problème de mariage stable, sur les similarités supérieures à un certain seuil, pour sélectionner les correspondances finales. La solution retournée par l'approche SF correspond à un optimum local [Euzenat et Shvaiko, 2013].

[Euzenat et Valtchev, 2004] ont proposé l'approche OLA (OWL lite Alignment) pour l'alignement de deux ontologies en OWL. Dans cette approche, le problème est de trouver des mesures de similarité pour toutes les paires de propriétés et pour toutes les paires d'entités. Une fois ce problème résolu, ils appliquent le problème de couplage de poids maximal pour sélectionner les correspondances finales. La résolution du problème de calcul de mesures de similarités revient à résoudre un système d'équations. Chaque équation représente des dépendances entre les similarités d'une paire d'entités ou de propriétés. Ce système d'équations est initialisé par des similarités élémentaires et structurelles. Les similarités élémentaires sont calculées entre les labels des entités ou les labels des propriétés en utilisant des techniques d'appariement élémentaires. Les similarités structurelles sont calculées à partir des caractéristiques internes des éléments comme les domaines, les types de données et les cardinalités des propriétés. Après son initialisation, le système d'équation est soumis à un algorithme itératif. Cet algorithme change les valeurs des mesures de similarité tant que

les dépendances entre ces similarités ne stabilisent pas les valeurs. Cet algorithme s'arrête à un point fixe qui correspond à des changements infimes des mesures de similarités. La sélection des correspondances finales se fait par la résolution du problème de couplage de poids maximal dans un graphe bipartie dont les noeuds sont les propriétés ou les entités et les arcs sont les correspondances potentielles avec comme poids les valeurs des mesures de similarité. La solution qu'ils proposent est un optimum global [Euzenat et Shvaiko, 2013].

[Aumueller *et al.*, 2005] ont proposé l'outil générique COMA++(Combining Match Algorithms) pour l'appariement de deux modèles qui peuvent être des schémas relationnels, des fichiers XML/XSD, des graphes RDF ou des ontologies en OWL. COMA++ transforme les modèles de données en graphes orientés acycliques où les éléments sont les chemins. Avec cet outil, il est possible de : (i) combiner, évaluer et réutiliser les résultats d'appariement, (ii) fragmenter les modèles de données et appliquer les techniques d'appariement sur des fragments, (iii) appliquer des stratégies pré-définies ou définir sa stratégie ou modifier les paramètres (seuils, poids, etc). Dans cet outil, ils utilisent tout type de techniques d'appariement élémentaires, voir section 2.1.1, et des techniques structurelles qui consistent à mesurer des simialrités structurelles entre les chemins, les enfants et les feuilles.

[Duchateau *et al.*, 2007] [Duchateau, 2009] ont proposé les deux approches BMatch et YAM. BMatch est une approche pour l'appariement d'une paire de schémas XML. Ils ont combiné les techniques élémentaires trigramme et Levensthein pour calculer une mesure de similarité élémentaire. D'autre part, ils ont calculé une mesure structurelle avec la distance cosine entre les vecteurs des contextes des éléments. Un vecteur de contexte est la liste des éléments voisins et leurs distances dans l'arbre par rapport à l'élément en cours. Les auteurs combinent ces deux types de mesures et sélectionnent les correspondances de similarité supérieures à un seuil donné. La spécificité de l'approche BMatch est qu'elle utilise une structure b-tree pour indexer les éléments ayant des termes communs. La b-tree améliore la performance de la solution. YAM (Not Yet Another Matcher) est un générateur d'approche d'appariement sur mesure en fonction des schémas et en fonction des préférences des utilisateurs. YAM utilise la technique d'apprentissage supervisé sur un large corpus de schémas et différentes mesures de similarités. Il permet ainsi de produire sur mesure les fonctions d'agrégations, les seuils de similarité par un arbre de décision, etc. Il peut aussi réutiliser les correspondances dans un nouveau processus d'appariement.

[Giunchiglia *et al.*, 2004][Yatskevich, 2008] ont proposé l'approche S-Match/S-Match++ (Semantic Match) pour l'appariement de deux modèles de données de structures hiérarchiques. C'est une approche qui renvoie des relations de type équivalence ou spécialisation entre les correspondances. Les auteurs transforment les labels des éléments en des formules propositionnelles. Ces formules codifient la signification sémantique de chaque entité. Ils utilisent la ressource externe Wordnet pour trouver les relations entre les propositions. Dans cette approche, le problème d'appariement est réduit à la résolution du problème de staisfiabilité SAT qui est un problème NP-Complet. Les auteurs utilisent des solveurs pour résoudre ce problème.

[Niepert *et al.*, 2010][Huber *et al.*, 2011] ont proposé le système CODI (Combinatorial Optimisation for Data Integration) pour l'appariement de deux ontologies en format OWL. CODI implémente un framework probabiliste basé sur la logique de Markov proposé par

[Niepert *et al.*, 2010]. Ce framework transforme le problème d'appariement en résolution du problème de maximum-a-posteriori (MAP) qui se réduit au problème d'optimisation combinatoire Max-Sat connu pour être un problème NP-difficile. Pour deux ontologies, les auteurs appliquent la technique de Levensthein pour mesurer la similarité entre les paires d'entités des deux ontologies. Puis, ils éliminent les paires d'entités qui ont une similarité inférieure à un seuil donné. Ensuite ils proposent deux types de contraintes de la logique du premier ordre qui forment le réseau logique de Markov. Le premier type de contraintes dites "contraintes strictes" correspond aux assertions existantes dans l'ontologie. Le deuxième type de contraintes dites "contraintes souples" est composé de trois sous-types : cardinalité 1 : 1, cohérence et stabilité. Les auteurs ont utilisé le raisonneur Pellet pour générer ces contraintes. Ils utilisent aussi des poids pour les contraintes de stabilité qui sont donnés soit manuellement, soit obtenus par un processus d'apprentissage sur d'autres ontologies pour fixer ces poids. Comme les contraintes strictes sont prises comme des vérités, la recherche de correspondances correspond à satisfaire le plus de contraintes souples ce qui fait que leur problème est réduit à un problème Max-Sat. Pour résoudre ce problème, ils l'ont transformé en un programme linéaire en entiers mixtes avec l'approche TheBeast [Riedel, 2008].

2.2.2.2 Synthèse et limites des travaux

Nous avons synthétisé dans le tableau III.2, les caractéristiques des approches décrites ci-dessus. Analysons à présent les limites de ces approches par rapport à notre objectif qui est d'obtenir des correspondances formant une structure hiérarchique.

D'abord, nous allons comparer les approches [Aumueller *et al.*, 2005], [Melnik *et al.*, 2002] et [Euzenat et Valtchev, 2004]. [Aumueller *et al.*, 2005] se focalisent sur la diversification des techniques de calcul de similarité, en particulier ils mettent l'accent sur la similarité élémentaire et beaucoup moins sur les similarités structurelles. De l'autre côté, [Melnik *et al.*, 2002] utilisent une technique élémentaire très basique et se focalisent sur un algorithme itératif original (voisinage) pour calculer une similarité structurelle. Quant à [Euzenat et Valtchev, 2004], ils combinent les similarités structurelles et élémentaires en prenant en considération la dépendance qui peut exister entre les deux. D'après [Euzenat et Shvaiko, 2013], l'algorithme itératif utilisé dans [Euzenat et Valtchev, 2004] converge alors que l'algorithme itératif utilisé dans [Melnik *et al.*, 2002] peut ne pas converger. Le point commun entre les trois approches [Aumueller *et al.*, 2005], [Melnik *et al.*, 2002] et [Euzenat et Valtchev, 2004] est que le calcul des similarités est très décisif dans le processus d'appariement. Pour sélectionner les correspondances, [Aumueller *et al.*, 2005] déploie une stratégie classique qui consiste à combiner les mesures avec une fonction d'agrégation. Alors que [Melnik *et al.*, 2002] et [Euzenat et Valtchev, 2004] résolvent le problème d'une façon plus sophistiquée en le réduisant à la résolution d'un problème connu en optimisation combinatoire. De ce point de vue, la solution de [Euzenat et Valtchev, 2004] serait meilleure que l'approche de [Melnik *et al.*, 2002] : le premier trouve un optimum global alors que le second trouve un optimum local. Nous pouvons tirer comme conclusion sur ces approches qu'ils n'ont pas donné une importance à la structure du graphe intégré autant qu'ils l'ont donné pour les similarités des correspondances. Ce qui fait qu'elles ne sont pas adaptées à notre objectif.

L'approche Bmatch de [Duchateau *et al.*, 2007] est tout à fait un cas particulier de l'approche de [Aumueller *et al.*, 2005] avec l'originalité du b-tree qui permet d'améliorer la performance. YAM [Duchateau, 2009] rentre aussi dans le cadre d'approches qui se focalisent sur les mesures de similarités. YAM a l'originalité de pouvoir générer des approches d'appariement sur mesure avec des paramètres configurés en fonction du jeu de donnés. Néanmoins, YAM est très couteuse puisqu'elle fait de l'apprentissage supervisé. De telles approches d'apprentissage ne seront pas efficaces pour des cas isolés d'intégration de données ouvertes où l'utilisateur ne serait pas en mesure de fournir les jeux de données pour l'apprentissage.

L'approche de [Niepert *et al.*, 2010][Huber *et al.*, 2011] est la plus pertinente par rapport à notre contexte. Dans cette approche, les contraintes de cohérence et l'une des contraintes de stabilité ne s'appliquent pas sur les graphes de structures hiérarchiques puisqu'il n'y pas les assertions concernées par ces contraintes. La deuxième contrainte de stabilité pourrait être appliquée sur les structures hiérarchiques, par contre elle est trop générique et n'évite pas la génération de structures hiérarchiques simples. Par ailleurs, la taille du problème générée par les instances augmente la difficulté de résolution de cette approche étant donné qu'elle est réduite à un problème NP-difficile (Max-Sat). Nous pensons que ceci est parmi les raisons pour lesquelles les auteurs ont réduit la taille du problème en entrée en utilisant un seuil de similarité. Par contre dans le cas d'hétérogénéité forte entre les ontologies, notamment les données ouvertes tabulaires, le seuil de similarité ne peut être que très faible et dans ce cas la résolution de leur problème serait incertaine. Nous notons aussi que les auteurs n'ont pas proposé de modèle de programme linéaire (comme il a été généré sur les instances par une autre approche) ce qui fait qu'il est difficile de voir comment cette approche pourrait évoluer vers une approche holistique pour traiter N ontologies par exemple. Enfin, nous avons remarqué que la contrainte de cardinalité telle qu'elle est proposée en logique génère beaucoup de contraintes dans le programme linéaire, puisqu'elle compare les correspondances deux à deux. Nous pensons qu'il est possible de proposer la même contrainte d'une façon plus optimisée.

Tableau III.2 — Comparaison des approches d'appariement par paire

Approche	Modèle donn.	Modèle Représen. Interne	Ress. externe	Techniques d'appariement		Réduction à un prob. d'optimisation combinatoire		
				Niv. élémentaire	Niv. structurel	Problème	Complexité	Dans l'étape de
[Melnik *et al.*, 2002]	relationnel, XML	graphe étiqueté orienté	non	string-based	propagation itérative d'une similarité structurelle	Mariage stable	polynomial	sélection
[Euzenat et Valtchev, 2004]	OWL	graphe étiqueté	non	string-based, language-based	internal-structure based	problème de coulage de poids maximal	polynomial	sélection
[Aumueller *et al.*, 2005]	relationnel, XML, XSD, RDF,OWL	graphe orienté acyclique	oui	string-based, constraint-based, resource-based, language-based	similarité entre enfants, feuilles, chemins	-	-	-
[Duchateau *et al.*, 2007]	XML	arbre	non	string-based	Btree	-	-	-
[Giunchiglia *et al.*, 2004] [Yatskevich, 2008]	XML	graphe	oui	string-based, language-based, external resource-based	-	SAT	NP-complet	appariement élémentaire
[Niepert *et al.*, 2010] [Huber *et al.*, 2011]	OWL	-	non	character-based	markov logique framework	Max-SAT	NP-difficile	appariement structurel et sélection

2.2.3 Les approches d'appariement pour l'intégration des données tabulaires

Dans cette section, nous reprenons la description de quelques approches d'intégration de données tabulaires. Nous rappelons que la partie détection et annotation de ces approches a été décrite dans le chapitre précédent, nous nous intéressons à présent à la manière dont ces approches ont résolu le problème d'appariement.

2.2.3.1 Étude des travaux

[Tijerino *et al.*, 2005] ont proposé le système TANGO (Table Analysis for Generating Ontologies) pour la construction d'une ontologie commune à plusieurs tableaux. Après les deux premières phases d'annotation et de construction d'ontologies, ils procèdent à la découverte de correspondances et à la fusion itérative des mini-ontologies deux à deux. Pour la découverte de correspondances, ils combinent : (i) des techniques élémentaires pour mesurer la similarité des labels (basée sur les termes ou par utilisation de la ressource externe Wordnet, les contraintes internes des éléments [Biskup et Embley, 2003]), (ii) la similarité entre les instances [Embley *et al.*, 2001] en utilisant la technique d'apprentissage et (iii) des techniques structurelles [Xu et Embley, 2003] en calculant une similarité structurelle entre les contextes d'adjacences de chaque élément (dans arcs sortants/entrants et les voisins).

[Coletta *et al.*, 2012][Castanier *et al.*, 2013] ont utilisé l'approche YAM++ [Ngo *et al.*, 2011][Ngo et Bellahsene, 2012], développée dans la même équipe de recherche, pour résoudre le problème d'appariement entre deux paires de tableaux annotés dans l'environnement WebSmatch. YAM++ combine 14 techniques d'appariement pour résoudre le problème d'appariement entre deux schémas/instances. Ces techniques calculent : (i) les similarités élémentaires entre les labels ou en utilisant la ressource externe Wordnet ou d'autres thésaurus et (ii) les similarités structurelles en utilisant l'algorithme de Similaritry Flooding [Melnik *et al.*, 2002].

[Scharffe *et al.*, 2012] ont utilisé l'outil Silk [Volz *et al.*, 2009] pour la découverte de liens entre les tableaux annotés et transformés en RDF. Silk est un outil dédié à la découverte de liens entre les instances des sources en RDF. En effet, l'utilisateur doit manuellement établir un fichier de spécification pour exprimer les correspondances entre les éléments de deux schémas. Ensuite, Silk raisonne sur ce fichier de spécification pour générer les liens entre le instances des éléments des schémas.

2.2.3.2 Synthèse et limites des travaux

Nous remarquons que ces trois approches intègrent les données tabulaires annotées en comparant les tableaux deux à deux. De plus, ils ne se focalisent pas sur une structure intégrée hiérarchique. Par rapport à ces approches, notre proposition est plus générique. En effet, notre proposition est la première approche de type holistique généralisant l'intégration de plusieurs tableaux en même temps. Notre solution holistique permettra d'éviter l'intervention humaine et de produire une solution unique. Ceci assure une automatisation du processus d'intégration de plusieurs sources. Notre approche possède également la spécificité de produire une vue qui aiderait dans la définition d'un schéma multidimensionnel

pour différents tableaux.

3 Contribution à l'intégration holistique des graphes hiérarchiques

Dans cette section, nous décrivons notre approche holistique pour l'intégration de graphes de données ouvertes. Cette approche doit nous permettre de construire une vue unifiée de plusieurs hiérarchies. Elle doit aussi résoudre le problème d'appariement avec le minimum d'intervention des utilisateurs pour être la plus automatique possible. De plus, notre approche doit faire face à l'hétérogénéité sémantique de sources de multiples domaines. Notre dernier objectif est de garantir une solution optimale globale pour le problème d'appariement.

Afin de réaliser ces objectifs, nous proposons un programme linéaire, nommé LP4HM (Linear Program For Holistic Matching) qui contient des contraintes relatives à la structure et des contraintes relatives au problème d'appariement. Il retourne une solution optimale globale au problème puisqu'il se réduit au problème de couplage de poids maximal. Il a été aussi modélisé de façon à pouvoir traiter holistiquement toutes les structures hiérarchiques en entrée. Nous avons proposé de combiner plusieurs techniques élémentaires et d'utiliser le thésaurus Wordnet comme ressource linguistique externe afin d'obtenir une bonne qualité entre les correspondances.

Nous décrivons dans la section suivante une vue globale de notre approche puis nous détaillons les différentes parties de cette approche.

3.1 Description globale de l'approche

Notre approche prend en entrée $N \geq 2$ graphes de structure hiérarchique et renvoie en sortie un graphe intégré G_{int}. Ce dernier doit se présenter sous la forme de structures hiérarchiques strictes où chaque élément doit avoir au plus un seul parent. De telles structures dans le graphe intégré, faciliteront à des utilisateurs non-experts l'identification du schéma multidimensionnel de l'entrepôt.

L'approche se déroule en trois phases, comme le montre la Figure III.10 :

- Dans la phase de préparation des données, nous construisons des matrices de directions, nous préparons $N(N-1)/2$ paires de graphes et nous combinons dans des matrices de similarités les résultats de plusieurs techniques d'appariement élémentaires. Par rapport à la Figure III.3, cette phase englobe les étapes de pré-traitement, d'exécution de techniques d'appariement de type élémentaire et de combinaison des résultats.
- Dans la deuxième phase, nous construisons et nous résolvons un programme linéaire (LP4HM). Ce programme linéaire contient des variables de décision, une fonction objectif et des contraintes linéaires. Ces dernières seront construites en utilisant les matrices de direction, les paires de graphes et les matrices de similarité. Pour la résolution de notre modèle, nous faisons appel à un solveur qui est un logiciel basé sur

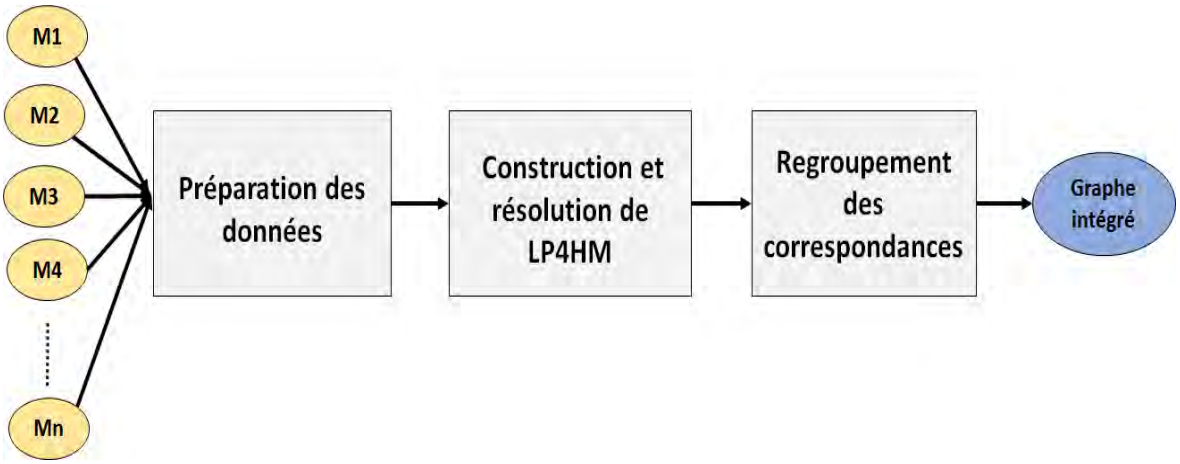


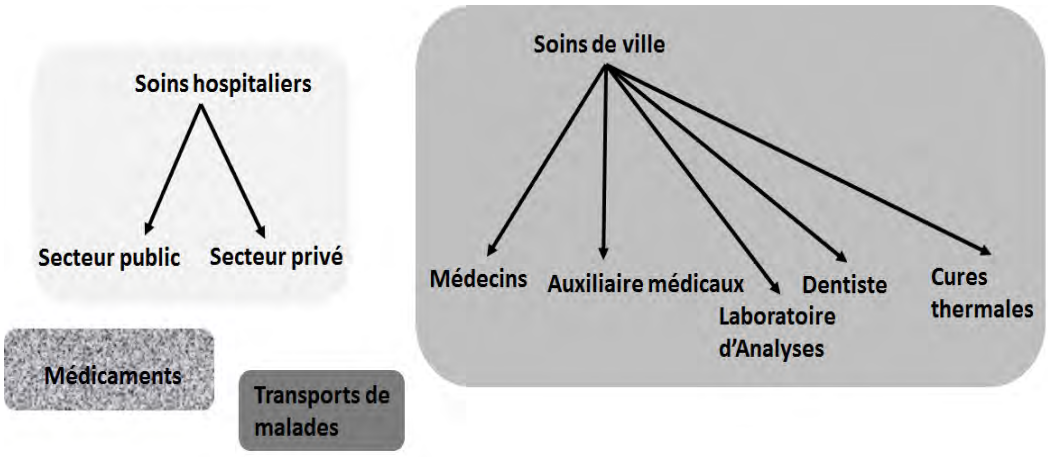
Figure III.10 — Un aperçu global des étapes de notre approche

des techniques mathématiques et un moteur d'optimisation [2]. La deuxième phase fusionne l'étape d'appariement structurel sans calcul de mesure de similarité structurelle et l'étape de sélection des correspondances. Nous pouvons remarquer que notre approche comporte moins d'étapes que celles proposées dans le processus général de la Figure III.3.

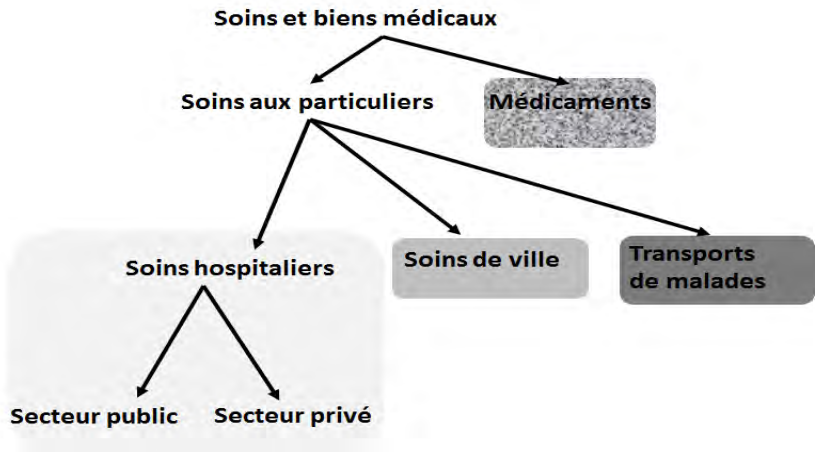
– Dans la dernière phase, nous regroupons les correspondances et nous construisons un unique graphe intégré qui correspond à une vue unifiée des différents modèles de données fournis en entrée.

Étude de cas Nous proposons au lecteur de suivre notre démarche à travers un scénario qui concerne l'intégration de deux graphes de données de soins en France. La Figure III.11 montre ces deux graphes qui correspondent aux données structurelles hiérarchisées des tableaux disponibles sur le portail data.gouv.fr [3] [4]. Le graphe III.11(a) contient des statistiques sur la consommation des soins par type de soin et par année. Le graphe III.11(b) contient des statistiques sur les dépenses de soins par année et par type de soin. Ce qui nous intéresse est l'obtention d'une vue unifiée des données structurelles sur le type de soins. A l'aide des mêmes codes couleurs entre les éléments des deux schémas, nous remarquons qu'il y a plusieurs correspondances et des labels en communs. Il s'agit d'un exemple de faible hétérogénéité puisque tous les labels sont soit identiques, soit proches sémantiquement. Généralement, dans ces cas un seuil de similarité très élevé suffit pour donner des résultats pertinents. Nous illustrons à travers cet exemple comment il est possible de trouver des résultats cohérents en imposant des contraintes de structures sans utiliser un seuil de similarité puis l'impact du seuil de similarité sur les résultats.

2. http ://www.cat-logistique.com/vocabulaire.htm
3. https ://www.data.gouv.fr/fr/datasets/comptes-nationaux-de-la-sante-2010-consommation-de-soins-et-de-biens-medicaux-en-volume-bas-30378565/
4. https ://www.data.gouv.fr/fr/datasets/comptes-nationaux-de-la-sante-2010-depenses-de-sante-par-type-de-financeur-2006-30378525/



(a) graphe 1



(b) graphe 2

Figure III.11 — Les graphes de données ouvertes en entrée

3.2 Préparation des données

La première étape de préparation des données a pour objectif d'élaborer des matrices de direction et des matrices de similarité ; ces matrices seront utilisées dans le programme linéaire. L'étape de préparation comprend deux sous-étapes : (1) une sous-étape de construction des matrices de direction et (2) une sous-étape de construction des matrices de similarité.

Cette étape de préparation prend en entrée $N \geq 2$ modèles de données. Dans notre proposition, ces modèles de données peuvent être soit des graphes de propriétés, soit des graphes RDF issus de données ouvertes tabulaires (voir chapitre 2). Ces graphes contiennent des données structurelles annotées et organisées en hiérarchies strictes ainsi que des données numériques. Pour notre approche d'intégration, nous ne prenons en compte que les données structurelles, leurs propriétés sémantiques et les relations de spécialisation entre ces données. Ces données structurelles forment en effet le schéma des données ouvertes tabulaires.

Les données structurelles de notre graphe de propriétés éclatée ou de notre graphe RDF

correspondent à un graphe orienté acyclique $G_i = (V_i, E_i)$ où $1 \leq i \leq N$, tel que :

- $V_i = \{v_{i_k},\ \forall k \in [1, n_i]\}$ représente l'ensemble des noeuds du graphe G_i.
- $E_i = \{e_{i_{k,l}} = (v_{i_k}, v_{i_l}),\ \forall k, l \in [1, n_i]\}$ représente l'ensemble des arcs du graphe G_i.

Avec k correspond à l'ordre de chaque noeud après un parcours en profondeur du graphe et $n_i = |V_i|$ correspond au nombre de noeuds du graphe G_i. Les N graphes en entrée seront ordonnés selon un ordre décroissant de n_i afin de réduire la taille du modèle LP4HM.

La Figure III.12 montre les notations associées aux graphes G_1 et G_2 de notre scénario.

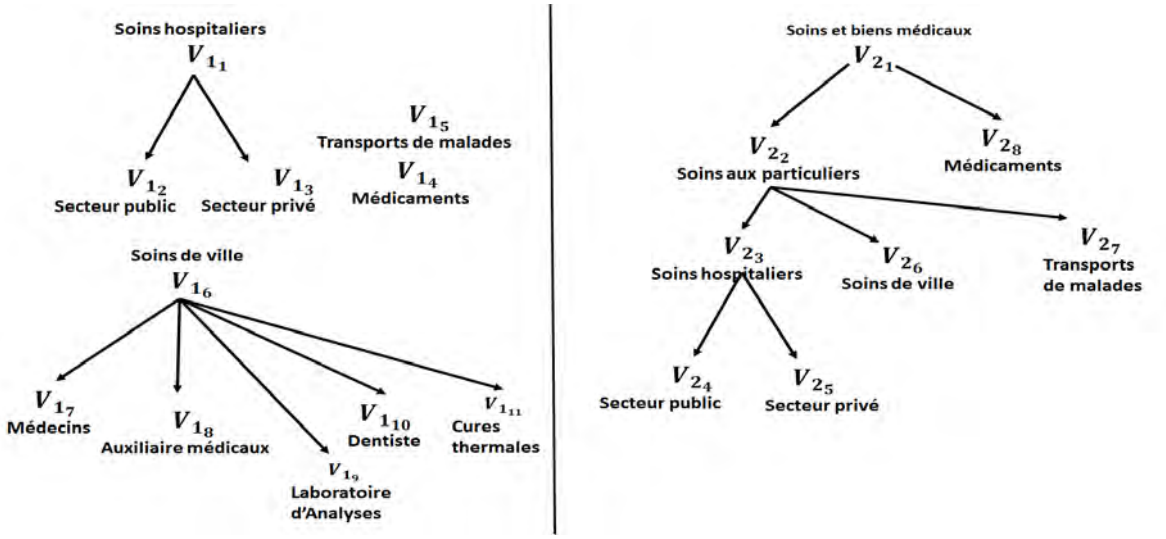


Figure III.12 — Les notations des graphes à intégrer

Remarque 1. *D'une façon générale, notre approche peut accepter n'importe quel modèle hiérarchique de données tel que des sources XML ou des taxonomies. Ces sources peuvent se transformer en graphe orienté acyclique en éclatant leurs propriétés.*

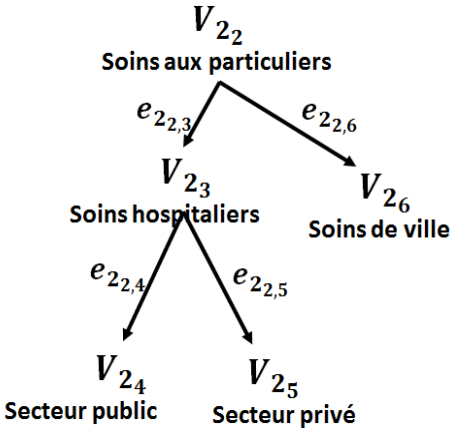
3.2.1 Préparation des matrices de direction

Les matrices de direction encodent la direction des arcs dans les graphes sachant que les arcs sont de même type. Une matrice de direction, notée Dir_i, de taille $n_i \times n_i$ pour chaque graphe G_i $\forall i \in [1, N]$ est définie comme suit :

$$Dir_i = \{dir_{i_{k,l}},\ \forall k \times l \in [1, n_i] \times [1, n_i]\}$$

$$dir_{i_{k,l}} = \begin{cases} 1 & \text{si } e_{i_{k,l}} \in E_i \\ -1 & \text{si } e_{i_{l,k}} \in E_i \\ 0 & \text{sinon} \end{cases}$$

Dans la Figure III.13, nous avons un extrait du graphe G_2 (de notre étude de cas) et un extrait de la matrice de direction Dir_2 qui lui correspond. Prenons le cas des deux noeuds V_{2_2} et V_{2_3} : l'arc $e_{2_{2,3}} \in E_2$ alors $dir_{2_{2,3}} = 1$ et $dir_{2_{3,2}} = -1$.



	V_{2_2}	V_{2_3}	V_{2_4}	V_{2_5}	V_{2_6}
V_{2_2}	0	1	0	0	1
V_{2_3}	-1	0	1	1	0
V_{2_4}	0	-1	0	0	0
V_{2_5}	0	-1	0	0	0
V_{2_6}	-1	0	0	0	0

$\subset Dir_2$

Figure III.13 — Un exemple de matrice de direction

3.2.2 Préparation des matrices de similarité

Dans cette étape nous construisons $N(N-1)/2$ matrices de similarité des N graphes. Ces matrices de similarité seront utilisées par le programme linéaire.

Une matrice de similarité $Sim_{i,j}$ de taille $n_i \times n_j$ contient des mesures de similarité sim_{i_k,j_l} calculées entre toutes les combinaisons des paires de noeuds v_{i_k} et v_{j_l} appartenant respectivement à G_i $\forall i \in [1, N-1]$ et G_j $\forall j \in [i+1, N]$:

$$Sim_{i,j} = \{sim_{i_k,j_l}, \forall k \in [1, n_i], \forall l \in [1, n_j]\}$$

Afin de calculer les mesures de similarité sim_{i_k,j_l}, nous avons opté pour une stratégie optimiste qui consiste à prendre le maximum de plusieurs techniques d'appariement élémentaires. Nous utilisons trois techniques d'appariement élémentaires différentes : (1) des techniques élémentaires basées sur le langage, (2) des techniques élémentaires basées sur le caractère, (3) des techniques élémentaires basées sur les groupes de termes et (4) des techniques élémentaires basées sur des ressources formelles externes.

La combinaison de techniques différentes a pour objectif de pallier les problèmes d'hétérogénéité sémantique entre les données structurelles provenant de plusieurs sources. En effet, l'usage de toutes ces techniques permet une meilleure estimation de la similarité.

Dans un premier temps, nous appliquons les techniques d'appariement élémentaires basées sur le langage. Ces techniques nous permettent de décomposer les labels des noeuds de chaque graphe en des sacs de termes. Ces sacs de termes seront utilisés par les trois autres types de techniques. Les techniques élémentaires basées sur le langage se déroulent comme suit : nous découpons chaque label de chaque noeud de chaque graphe en sacs de termes en utilisant des expressions régulières, puis nous éliminons les termes vides enfin nous cherchons les racines de chaque terme en utilisant la libraire Snowball [5]. Notons par c_{i_k} le label du noeud v_{i_k} dans un graphe G_i. Après cette première itération préparatoire, c_{i_k} sera

5. http ://snowball.tartarus.org/

considéré comme à un sac de racines de termes. Il est noté comme suit $c_{i_k} = \bigcup_{n=1}^{|c_{i_k}|} c_{i_{k_n}}$ avec $c_{i_{k_n}}$ est une racine de terme contenue dans le label c_{i_k}.

Dans un deuxième temps, nous appliquons les trois autres techniques élémentaires en utilisant les sacs de racines de termes. Chaque technique se calcule indépendamment des autres techniques. Le choix des techniques utilisées s'est basé principalement sur une récente étude comparative proposée par [Sun *et al.*, 2015].

Dans l'étude de [Sun *et al.*, 2015], les techniques élémentaires basées sur le caractère **Edit-Distance**, **Monge-Elkan**, **Jaro-Winkler**, **ISUB** et **Trigramme** donnent de bons résultats, nous les avons ainsi choisies pour calculer la similarité entre deux termes. Afin de calculer les similarités entre les labels des noeuds (sacs de racines de termes), nous avons utilisé la méthode de **Mongue-Elkan généralisée** [Jimenez *et al.*, 2009] sur chacune des techniques élémentaires basées sur le caractère. Cette méthode est exprimée comme suit :

$$MongueElkanGen[TechElem]_{i_k,j_l}(c_{i_k}, c_{j_l}) = (1/|c_{i_k}| \sum_{n=1}^{|c_{i_k}|} (max\{TechElem(c_{i_{k_n}}, c_{j_{l_m}})\}_{m=1}^{|c_{j_l}|})^2)^{1/2}$$

Avec TechElem= {EditDistance, Monge-Elkan, Jaro-Winkler, ISUB, Trigramme}

Parmi les techniques élémentaires basées sur les groupes de termes, nous avons sélectionné les mesures **Jaccard** et **SoftTFIDF** qui obtiennent également de bons résultats.

Parmi les techniques élémentaires basées sur les ressources formelles externes, nous avons retenu la technique **Lin** étudiée par [Sun *et al.*, 2015]. Nous avons également retenu la technique **Wup** [Wu et Palmer., 1994] qui est jugée comme une technique "bien élaborée" dans [Euzenat et Shvaiko, 2013].

En fonction de ces différentes techniques, notre mesure de similarité est exprimée comme suit :

$$\begin{aligned} sim_{i_k,j_l} = max(&MongueElkanGen[EditDistance]_{i_k,j_l}, MongueElkanGen[MongeElkan]_{i_k,j_l} \\ &MongueElkanGen[JaroWinkler]_{i_k,j_l}, MongueElkanGen[ISUB]_{i_k,j_l}, \\ &MongueElkanGen[3-gram]_{i_k,j_l}, Jaccard_{i_k,j_l}, SoftTFIDF_{i_k,j_l}, \\ &Lin_{i_k,j_l}, Wup_{i_k,j_l}) \end{aligned}$$

Actuellement, différentes librairies implémentent ces techniques, celles qui ont été utilisées pour calculer les mesures de similarité de chaque technique sont : OntoSim [6], SimMetric [7], SecondString [8] et WS4J [9].

Calcul d'un seuil de similarité prédéfini

Lors du calcul des matrices de similarités, notre approche calcule un seuil de similarité. En effet, pour chaque matrice entre paire de graphes, nous calculons le maximum de chaque

6. http ://ontosim.gforge.inria.fr/
7. http ://sourceforge.net/projects/simmetrics/
8. http ://secondstring.sourceforge.net/
9. https ://code.google.com/p/ws4j/

ligne, puis nous prenons la médiane des maximums des lignes qui représente un seuil local pour chaque matrice. Sur les $N(N-1)/2$ matrices de similarité, le seuil de similarité global est la médiane de tous les seuils locaux.

3.3 Le programme linéaire LP4HM

3.3.1 Préliminaires

3.3.1.1 La programmation linéaire

Un programme linéaire [Balinski, 1965] [Burke et Kendall, 2005] contient des variables de décision, des contraintes et une fonction objectif : (1) les variables de décision prennent des valeurs numériques, (2) les contraintes sont utilisées pour limiter les valeurs possibles dans une région de faisabilité et elles doivent être linéaires en fonction des variables de décision, (3) la fonction objectif définit quelles sont les affectations optimales possibles pour les variables de décision permettant de maximiser ou de minimiser la valeur de la fonction objectif. La fonction objectif doit être aussi linéaire en fonction des variables de décision.

Dans la littérature, [Plastria, 2002] ont proposé des fondements théoriques qui expliquent comment il est possible de passer d'implications logiques, de la forme *Si* **A** *Alors* **B**, en contraintes linéaires. En supposant que A et B sont des variables de décision en 0-1 [Plastria, 2002] ont proposé le théorème suivant :

Théorème 1. *Soit x_i une variable 0-1 $\forall i$ appartenant à un ensemble fini I et soit une variable $1 \leq y \leq 1$ alors*

$$\textbf{\textit{Si}} \;\; x_i = 0 \;\; \forall i \in I \;\; \textbf{\textit{Alors}} \;\; y = 0$$

est exactement exprimé par l'inégalité

$$y \leq \sum_{i \in I} x_i$$

Les contraintes linéaires de LP4HM sont basées sur ce théorème.

3.3.1.2 La relation entre le problème d'appariement et le problème de couplage de poids maximal

En optimisation combinatoire, un des problèmes connus [Schrijver, 2003] est le problème de couplage (ou appariement) de poids maximal (Maximum-weighted graph matching problem). Pour un graphe $G = (V, E)$, [Schrijver, 2003] définit ce problème comme suit : "chercher un couplage (= un ensemble disjoint d'arcs) M dans G de poids $w(M)$ le plus grand possible" ce qui est équivalent formellement à :

$$max\{w(M) | M \; matching \; dans \; G\}$$

La programmation linéaire est parmi les techniques qui ont été employées pour résoudre ce problème [Schrijver, 2003]. Les graphes peuvent être bipartis ou non-bipartis. Un graphe est dit biparti s'il existe une partition de son ensemble de sommets V en deux sous-ensembles

U et U', tel que chaque arête de E ait une extrémité dans U et l'autre extrémité dans U', si cette condition n'est pas vérifiée alors le graphe est dit non-biparti.

Dans un graphe biparti, le polytope de couplage (c-à-d l'espace géométrique des solutions possibles) est défini par les contraintes suivantes :

$$x(e) \geq 0 \quad pour\ e \in E$$

$$\sum_{v \in e} x(e) \leq 1 \quad pour\ v \in V$$

Ces contraintes expriment que pour chaque noeud $v \in V$, il faut qu'il y ait au plus un seul arc $e \in E$, parmi les arcs qui touchent v, qui s'affecte au couplage M. La solution du problème de couplage de poids maximal dans un graphe biparti peut être trouvée en temps polynomial [Schrijver, 2003]. Dans la Figure III.14, nous illustrons un exemple de graphe biparti G et la solution du problème de couplage de poids maximal dans ce graphe.

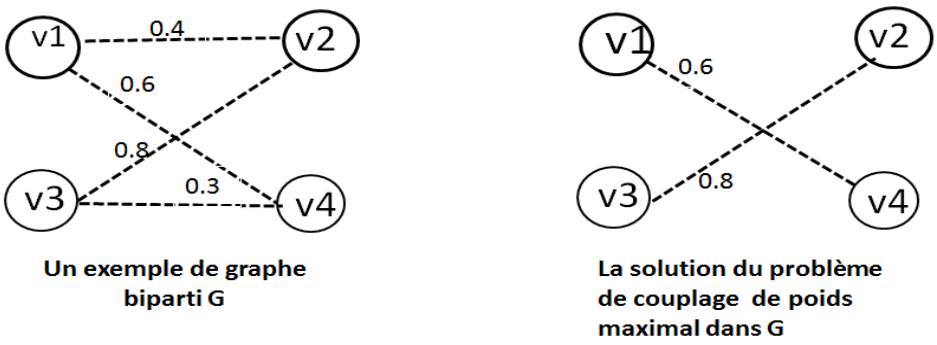


Figure III.14 — Un exemple de graphe biparti G et la solution du problème de couplage de poids maximal dans G

Dans un graphe non-biparti, [Edmonds, 1965] a proposé d'enlever les cycles impairs dans le polytope des solutions possibles avec la contrainte suivante :

$$\sum_{e \in U} x(e) \leq \lfloor 1/2|U| \rfloor \quad pour\ chaque\ cycle\ de\ taille\ impair\ U \in V$$

La résolution du problème de couplage de poids maximal dans un graphe non-biparti est également possible en temps polynomial [Schrijver, 2003].

Après cette introduction au problème de couplage de poids maximal dans un graphe biparti et non-biparti, examinons à présent la relation entre le problème d'appariement de schémas et ce dernier.

Nous avons constaté que le problème d'appariement par paire peut se réduire au problème de couplage de poids maximal dans un graphe biparti. Nous illustrons dans la Figure III.15 cette réduction. Supposons dans un premier temps que nous avons deux graphes $G_A = (V_A, E_A)$ et $G_B = (V_B, E_B)$, chacun a une structure entre ces noeuds. Dans un deuxième temps, nous avons calculé les similarités entre chaque paire d'éléments de G_A et G_B par exemple $sim(a, b) = 0.4$. Dans un troisième temps, supposons que nous enlevons les arcs de structure de chaque graphe et que nous cherchons des correspondances simples (où chaque

élément d'un graphe doit correspondre à au plus un autre élément de l'autre graphe), ceci correspond exactement à la résolution du problème de couplage de poids maximal dans un graphe biparti G où $V = V_A \cup V_B$ (les noeuds de A et de B qui forment une partition) et $E = \{e = (v_A, v_B)\ et\ w(e) = sim(v_A, v_B)\}$ (les arcs entres des éléments de A et des éléments de B dont le poids correspond à la similarité entre les éléments). Cette réduction entre le problème d'appariement par paire et un problème d'optimisation combinatoire polynomial, motive notre proposition d'une modélisation équivalente à ce problème. Nous proposons également d'étendre cette modélisation par des contraintes relatives à la structure dans les graphes initiaux et des contraintes relatives au problème d'appariement entre schémas.

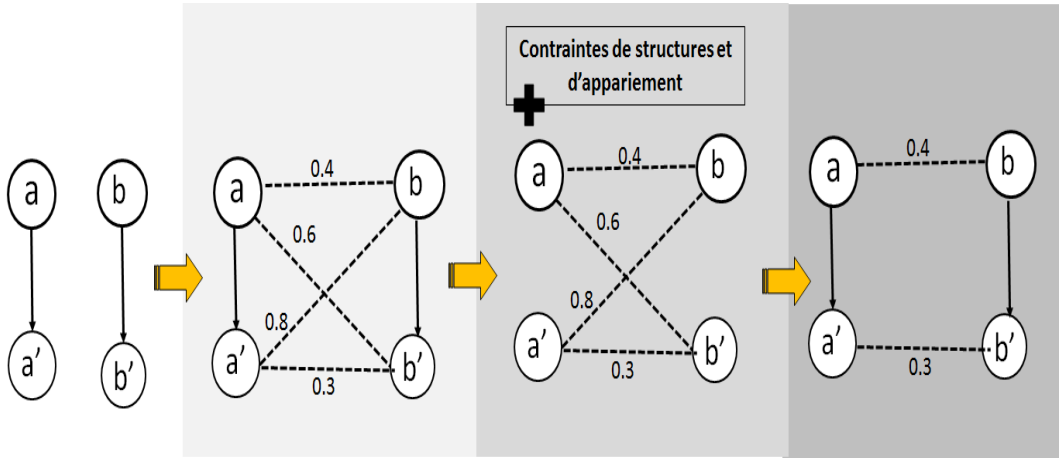


Figure III.15 — La relation entre le problème d'appariement par paire et le problème de couplage de poids maximal dans un graphe biparti

Nous avons aussi constaté que le problème d'appariement holistique peut se réduire au problème de couplage de poids maximal dans un graphe non-biparti. Nous illustrons dans la Figure III.16 l'explication de cette réduction. Dans un premier temps, supposons que nous avons trois graphes $G_A = (V_A, E_A)$, $G_B = (V_B, E_B)$ et $G_C = (V_C, E_C)$, chacun de ces graphes a sa propre structure. Dans un deuxième temps, nous calculons les similarités entre les éléments pour toutes les paires de noeuds de chaque paire de graphes (G_A, G_B), (G_A, G_C) et (G_B, G_C). Si nous enlevons les structures de chaque graphe, nous obtenons un nouveau graphe non-biparti $G = (V, E)$ tel que $V = V_A \cup V_B \cup V_C$ et $E = \{e = (v_i, v_i')\ tel\ que\ i \neq i'\ et\ i, i' \in \{A, B, C\}\ et\ w(e) = sim(v_i, v_i')\}$. La résolution du problème d'appariement holistique, avec des correspondances simples, entre les graphes G_A, G_B et G_C sans structure se réduit à une relaxation de la résolution du problème de couplage de poids maximal dans le graphe non-biparti G. Il s'agit bien d'une relaxation puisque dans ce dernier, nous souhaitons conserver les correspondances qui forment des cycles. Ceci nous amène à relâcher la contrainte d'Edmonds. LP4HM appliqué sur N graphes en même temps est une adaptation du problème de couplage de poids maximal dans un graphe non-biparti, étendu par des contraintes sur les structures et sur les spécificités du problème d'appariement.

Si nous résumons, le modèle LP4HM part du principe de l'adaptation et de l'extension d'un problème connu en optimisation combinatoire pour résoudre d'une façon générique et holistique le problème d'appariement en considérant toutes les spécificités du problème

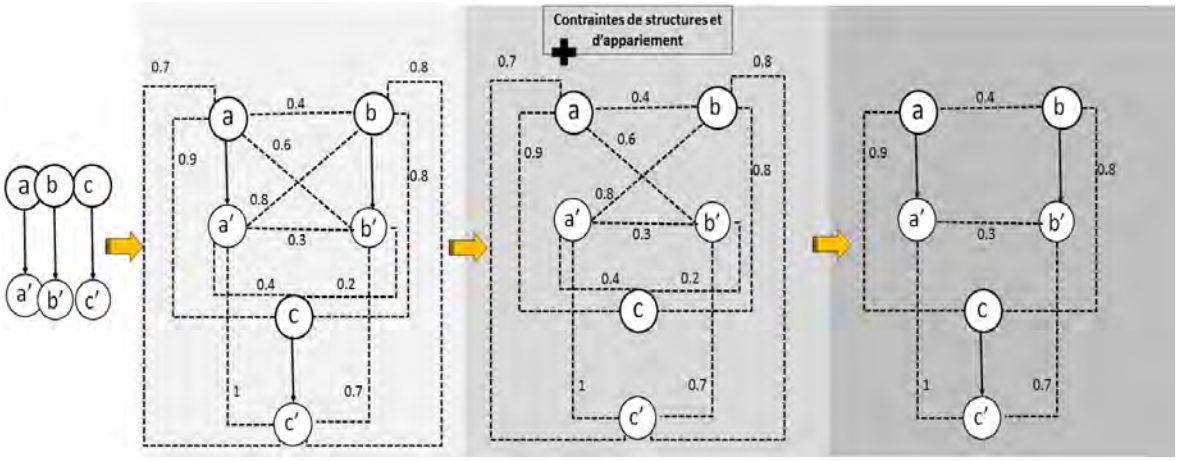


Figure III.16 — La relation entre le problème d'appariement holistique et le problème de couplage de poids maximal dans un graphe non-biparti

d'appariement.

Dans les sections suivantes, nous détaillons le programme linéaire LP4HM.

3.3.2 Variables de décision et fonction objectif

3.3.2.1 Variables de décision

LP4HM possède un seul type de variables de décision. Ces variables expriment la possibilité ou pas d'avoir une correspondance entre deux noeuds appartenant à une paire de graphes parmi les $N(N-1)/2$ paires de graphes.

Pour chaque paire de graphe G_i et G_j tel que $i \in [1, N-1]$ *et* $j \in [i+1, N]$, nous notons x_{i_k,j_l} une variable de décision binaire qui est égale à 1 s'il y a une correspondance entre le noeud v_{i_k} du graphe G_i et le noeud v_{j_l} du graphe G_j et 0 sinon.

Exemple 4. *Prenons le noeud v_{1_1} du graphe G_1 et les noeuds du graphe G_2, il existe 8 variables de décision entre le noeud v_{1_1} et les noeuds v_{2_l} de G_2, ces variables de décision sont $x_{1_1,2_l}$ tel que $l \in [1, 8]$. A chaque variable de décision $x_{1_1,2_l}$, nous associons la mesure de similarité qui a été calculée entre les deux labels des noeuds v_{1_1} et v_{2_l}. La Figure III.17 illustre ces variables de décision.*

3.3.2.2 Fonction objectif

LP4HM a pour objectif de trouver le meilleur ensemble de correspondances possible entre les différentes combinaisons de paires de graphes. Ceci correspond à maximiser la somme des similarités qui sont attachées aux variables de décisions.

Pour les $N(N-1)/2$ combinaisons de paire de graphes G_i et G_j, $\forall i \in [1, N-1]$ et $\forall j \in [i+1, N]$, la fonction objectif s'exprime comme suit :

$$max \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} sim_{i_k,j_l} \; x_{i_k,j_l}$$

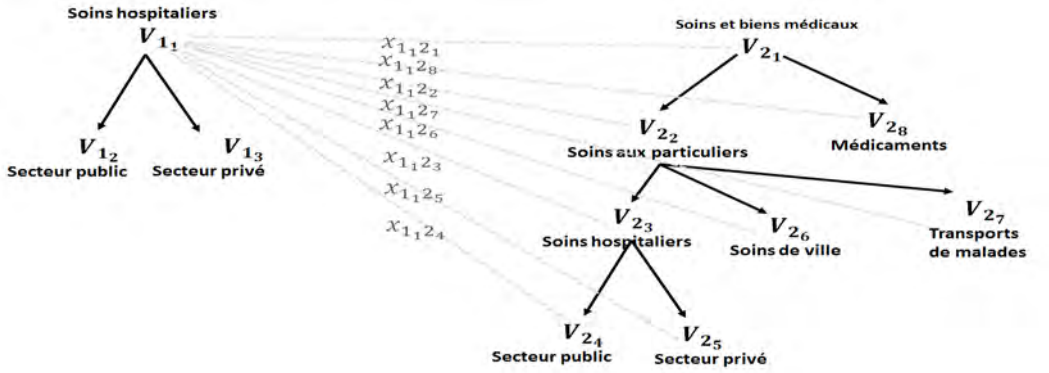


Figure III.17 — Exemples de variables de décision

3.3.3 Contraintes linéaires

LP4HM possède quatre contraintes linéaires qui expriment d'une part la structure entre les éléments des graphes, d'autre part des spécificités du problème d'appariement. Nous expliquons dans ce qui suit ces contraintes et nous illustrons par des exemples comment la combinaison de ces contraintes nous permet de résoudre efficacement le problème d'appariement.

3.3.3.1 La cardinalité des correspondances

Nous cherchons à travers cette contrainte à imposer à LP4HM de retourner des correspondances simples c'est à dire de cardinalité 1 : 1. Il faut alors que chaque noeud v_{i_k} d'un graphe G_i $\forall i \in [1, N-1]$ corresponde à au plus un seul noeud v_{j_l} d'un graphe G_j $\forall j \in [i+1, N]$. Dans la Figure III.18, nous schématisons le principe de cette contrainte. Dans le tableau de cette figure, nous avons en entête de lignes les noeuds du graphe G_i, en entête de colonnes les noeuds du graphe G_j. Le contenu du tableau correspond aux variables de décision entre les graphes G_i et G_j. Pour avoir au plus une seule correspondance pour chaque noeud il faut que la somme de chaque ligne et la somme de chaque colonne soit inférieure ou égale à 1, cela veut dire que nous autorisons au plus à une seule variable de décision binaire à être affectée à la valeur de 1.

D'une façon générale, pour chaque combinaison de graphe G_j $\forall i \in [1, N-1]$ et de graphe G_j $\forall j \in [i+1, N]$ la contrainte de correspondance simple s'exprime comme suit :

$$\sum_{l=1}^{n_j} x_{i_k, j_l} \leq 1, \quad \forall k \in [1, n_i]$$

Remarquons que notre contrainte est similaire à la contrainte $\sum_{v \in e} x(e) \leq 1 \quad pour\ v \in V$ dans le problème de couplage de poids maximal, toutefois la façon avec laquelle elle a été exprimée dépend des variables de décision que nous avons définies pour le problème d'appariement holistique.

Exemple 5. *Nous avons appliqué LP4HM sur notre scénario, avec uniquement la contrainte de*

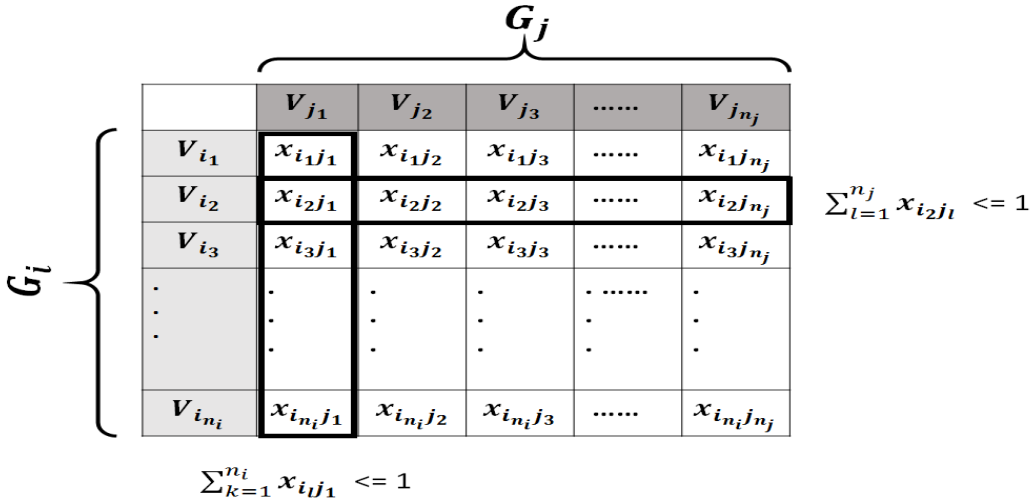


Figure III.18 — Le principe de la contrainte sur la cardinalité des correspondances

cardinalité des correspondances. Le résultat d'intégration des deux graphes avec les correspondances obtenues est montré dans la Figure III.19. LP4HM a retourné 8 correspondances dont quelques unes ont des similarités égales à 1 et d'autres ont des similarités inférieures à 1. Par exemple entre soins de ville du graphe 1 et soins de ville du graphe 2 la similarité est égale à 1, par contre pour les deux correspondances (soins et biens médicaux du graphe 2, cures thermales de graphe 1) et (soins aux particuliers du graphe 2, médecins du graphes 1) les similarités sont respectivement égales à 0.6 et à 0.7. LP4HM a en effet retourné des correspondances de cardinalité 1 : 1 *et il a maximisé la somme globale des correspondances. Comme nous n'avons pas indiqué la structure entre les éléments, LP4HM génère des arcs en double sens dans le graphe intégré, ce qui ne correspond pas à ce que nous estimons avoir dans le graphe intégré résultant de ces modèles. En effet, nous cherchons à construire des graphes intégrés avec des arcs simples afin que ce graphe ressemble le plus à une hiérarchie stricte, ce qui permettra de faciliter à des non-experts la conception d'un schéma multidimensionnel à partir du graphe intégré. Nous expliquons dans la contrainte suivante pourquoi ces arcs sont générés et comment notre contrainte permettra d'éviter ces cas.*

3.3.3.2 La direction des arcs

L'objectif de cette contrainte est d'interdire les arcs conflictuels dans le graphe intégré c'est à dire deux arcs dans un sens inverse reliant deux noeuds. La motivation de cette contrainte est d'obtenir des graphes intégrés de structure similaire à une hiérarchie. Nous avons schématisé dans la Figure III.20 un exemple de génération des arcs conflictuels de la Figure III.19. En effet, sim(soins de ville, soins de ville) + sim(soins aux particuliers, médecins) = 1.7 est la meilleure somme de similarité pour les correspondances de cardinalité 1 : 1 que le programme linéaire peut trouver sans aucune information sur la structure. Toutefois, en examinant la structure, cette solution correspond à un appariement entre le parent1 et le fils2 et un appariement entre le fils1 et le parent2, ce qui contredit une structure hiérarchique.

Afin d'interdire ce cas de figure, nous proposons la contrainte sur la direction des arcs dont le principe est illustré par la Figure III.21. En utilisant les matrices de direction de

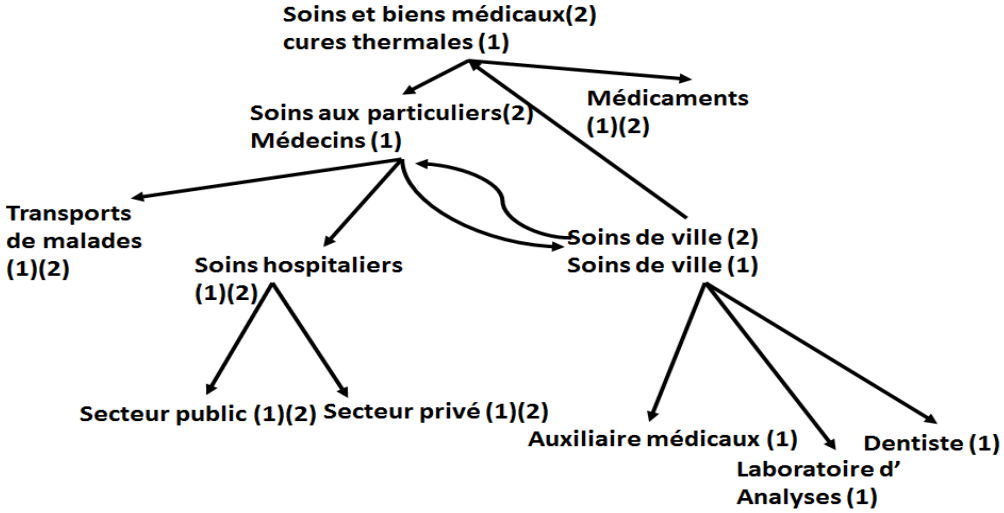


Figure III.19 — Résultat d'intégration en utilisant LP4HM avec la contrainte de cardinalité

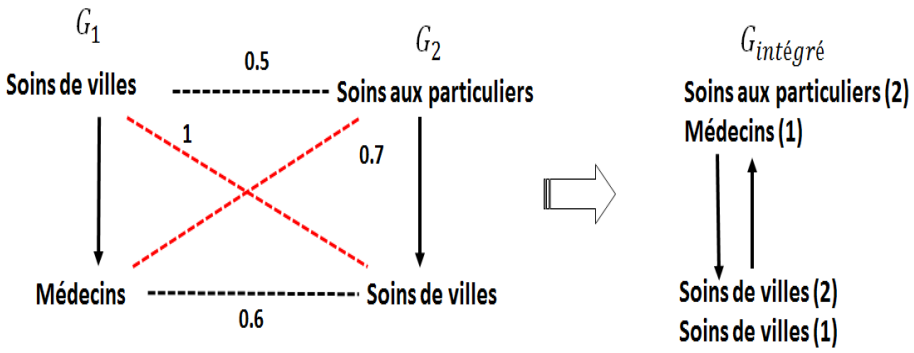


Figure III.20 — Explication de la génération d'arcs conflictuelles

chaque graphe, nous imposons au programme linéaire que le produit des directions des arcs des noeuds impliqués dans les variables de décision soit égal à 1. Dans le cas contraire, il ne peut pas affecter la valeur 1 aux variables de décisions correspondantes. Comme nous pouvons le voir sur la Figure III.21, le programme linéaire ne peut pas affecter les variables de la meilleure somme de similarité puisqu'elles ne vérifient pas cette contrainte.

D'une façon générale, pour chaque paire de graphe $G_i \; \forall i \in [1, N-1]$ et $G_j \; \forall j \in [i+1, N]$ et $\forall k, k' \in [1, n_i] \; \forall l, l' \in [1, n_j]$, la contrainte de direction des arcs s'exprime comme suit :

$$x_{i_k,j_l} + x_{i_{k'},j_{l'}} + (dir_{i_{k,k'}} dir_{j_{l,l'}}) \leq 0$$

Exemple 6. *Nous appliquons LP4HM sur notre scénario, en utilisant la contrainte de cardinalité des correspondances et la contrainte sur la direction des arcs. Nous obtenons le graphe de la Figure III.22. Le programme linéaire respecte les contraintes que nous lui avons imposées mais il propose une solution qui contredit notre deuxième exigence. Cette dernière consiste à avoir des hiérarchies strictes dans le graphe intégré. Dans une hiérarchie stricte, chaque noeud doit avoir au plus un seul*

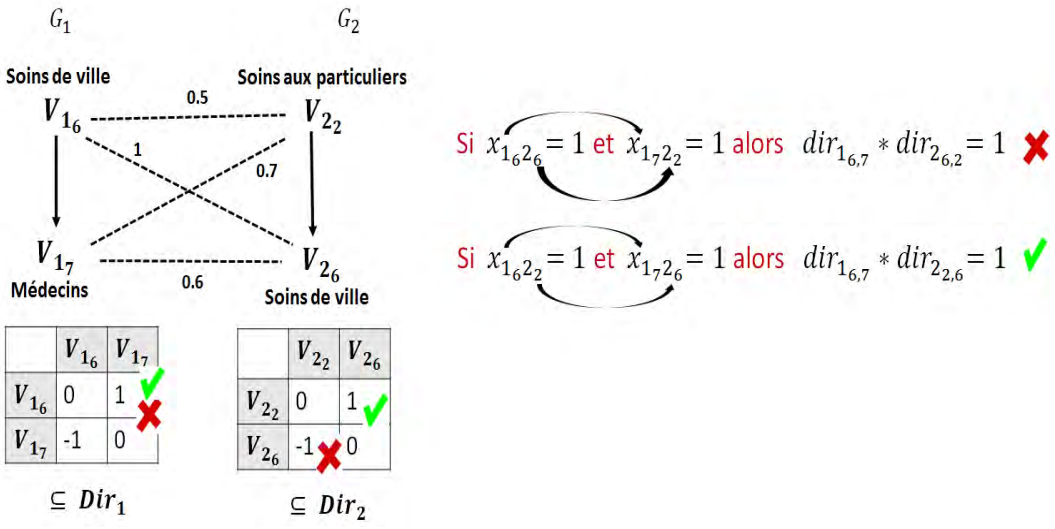


Figure III.21 — Le principe de la contrainte sur la direction des arcs

parent. Ceci n'est pas le cas ici, par exemple les noeuds secteurs privés et secteurs publics ont actuellement deux parents. Nous examinons dans la section suivante les contraintes permettant d'avoir des hiérarchies strictes [Malinowski et Zimányi, 2006].

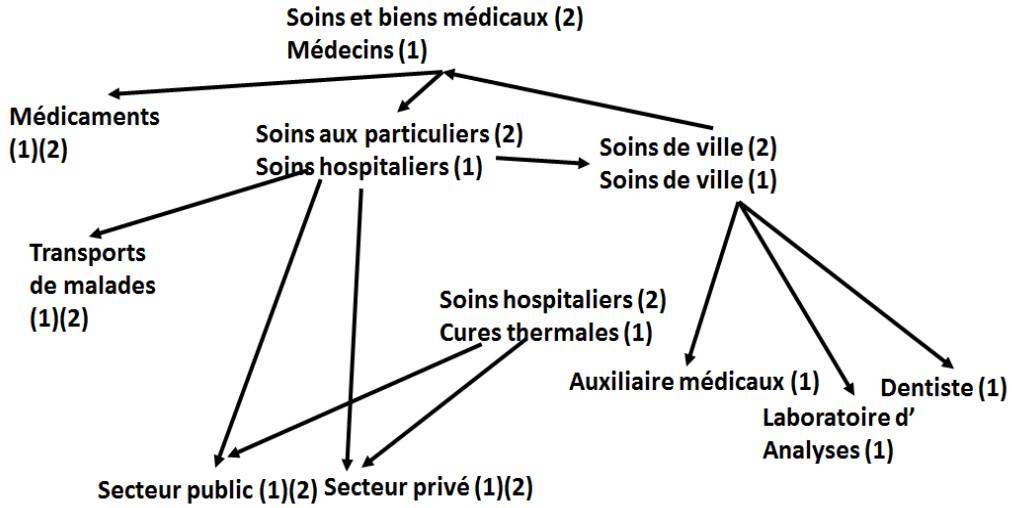


Figure III.22 — Résultat d'intégration en utilisant LP4HM avec la contrainte de cardinalité et la contrainte des directions des arcs

3.3.3.3 Les relations hiérarchiques

L'objectif de cette contrainte est d'obtenir des structures hiérarchiques intégrées strictes dans le graphe intégré. Ceci est important puisque nous envisageons de définir un entrepôt de données ouvertes conformément à un schéma multidimensionnel à partir du graphe intégré dans la troisième étape de notre démarche ETL. L'idée de cette contrainte est comme suit : "s'il y a une correspondance entre deux noeuds fils alors leurs parents doivent aussi correspondre, mais s'il y a une correspondance entre les parents alors les fils peuvent ne pas

correspondre". La Figure III.23 illustre comment cette contrainte est mise en place. Si le programme linéaire veut affecter à une variable de décision la valeur de 1, nous lui demandons que la variable de décision entre les prédécesseurs des noeuds de l'autre variable soit aussi égale à 1. Si ceci n'est pas possible il affecte les deux à 0. Par contre le programme linéaire peut affecter à la variable de décision des parents la valeur de 1 et à la variable de décision des enfants la valeur de 0, si par exemple ceci ne se répercute pas sur les grands-parents.

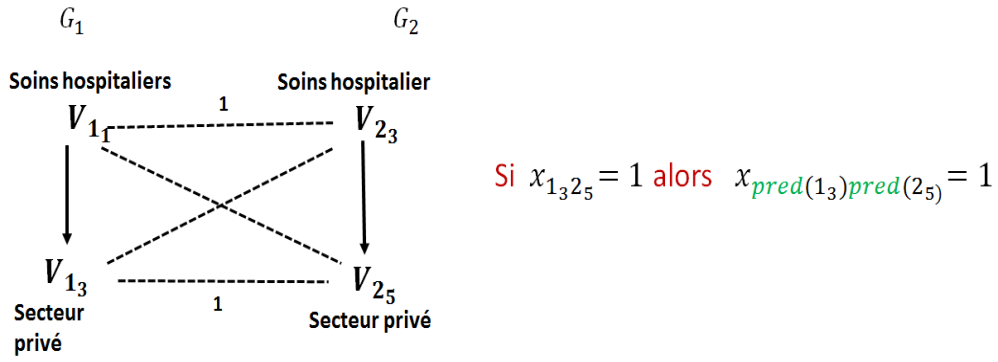


Figure III.23 — Principe de la contrainte des hiérarchies strictes

D'une façon générale, pour chaque paire de graphe G_i $\forall i \in [1, N-1]$ et G_j $\forall j \in [i+1, N]$ et $\forall k \in [1, n_i], l \in [1, n_j]$, la contrainte des structures hiérarchiques strictes est exprimée comme suit :

$$x_{i_k, j_l} \leq x_{i_{pred(k)}, j_{pred(l)}}$$

Exemple 7. *Sur notre scénario, nous appliquons LP4HM avec la contrainte sur les correspondances, la contrainte sur la direction des arcs et la contrainte sur les hiérarchies strictes. Le résultat d'intégration des deux graphes en fonction de leurs correspondances est illustré dans la Figure III.24. Nous remarquons que la solution obtenue comporte des hiérarchies strictes, des arcs non-conflictuels et des correspondances de cardinalité* 1 : 1 *et ceci sans aucune restriction sur les seuils de similarité.*

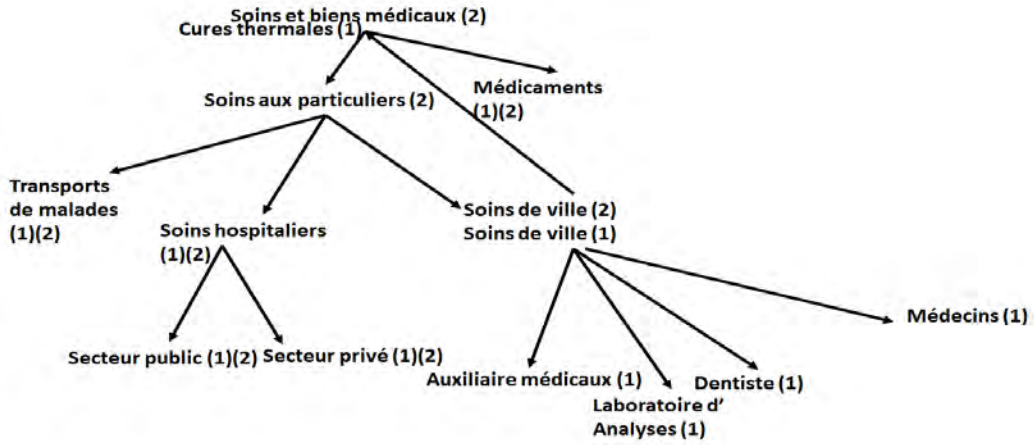


Figure III.24 — Résultat d'intégration en utilisant LP4HM avec la contrainte de cardinalité, la contrainte des direction des arcs et la contrainte des hiérarchies strictes

Nous attirons l'attention du lecteur que d'une façon générale les hiérarchies non-strictes

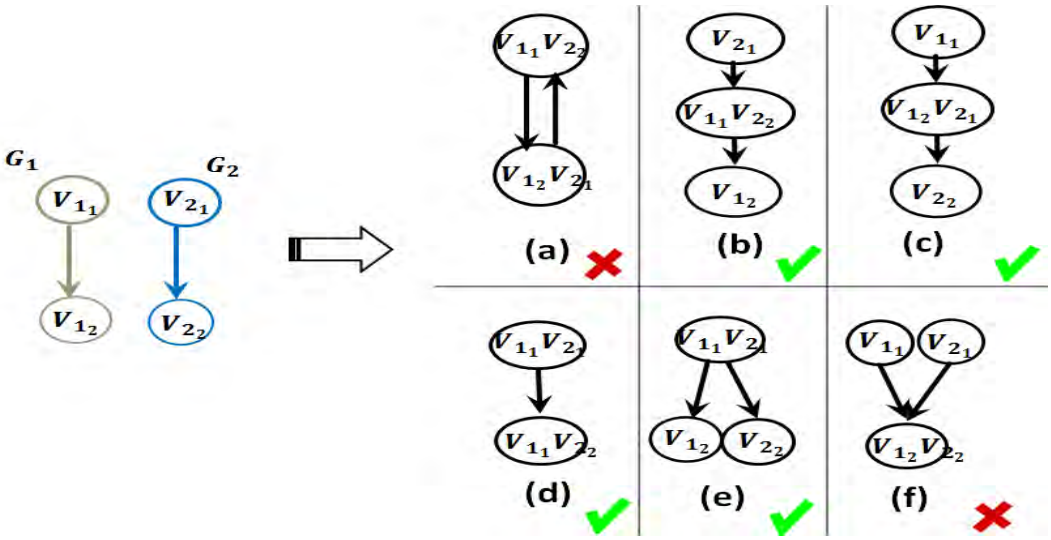


Figure III.25 — Résumé sur les cas d'intégration possibles en présence des contraintes structurelles

ne naissent pas par application de la contrainte sur la direction des arcs. En effet, elles peuvent apparaître sans aucun lien avec la contrainte des directions. Pour cela, l'originalité de notre modèle réside dans le fait que les deux contraintes structurelles collaborent pour obtenir une structure hiérarchique du graphe intégré. Nous résumons dans la Figure III.25 les différents cas d'intégration valides et non valides. Les cas non valides sont traités par les contraintes structurelles.

3.3.3.4 Le seuil de similarité

L'objectif de cette contrainte est d'améliorer la qualité des correspondances. En indiquant au programme linéaire un seuil de similarité, il restreint la recherche de la solution optimale aux variables de décision attachées à une similarité supérieure à ce seuil. Ce seuil peut être introduit par l'utilisateur ou proposé par le système (dans notre cas un seuil est pré-calculé lors de la phase de préparation des données).

D'une façon générale, pour chaque paire de graphe G_i $\forall i \in [1, N-1]$ et G_j $\forall j \in [i+1, N]$ $\forall k \in [1, n_i]$ $\forall l \in [1, n_j]$, la contrainte sur le seuil de similarité s'exprime comme suit :

$$sim_{i_k,j_l}\ x_{i_k,j_l} \geq seuil\ x_{i_k,j_l}$$

Exemple 8. *Nous appliquons sur notre scénario, les quatre contraintes de LP4HM avec un seuil* $= 0.9$. *La dernière contrainte avec le seuil de similarité génère la contrainte suivante* $0.6x_{1_1,2_1} \geq 0.9x_{1_1,2_1}$ *: ce n'est pas valide par conséquence la variable de décision* $x_{1_1,2_1}$ *est affectée à 0. Ceci permet l'élimination de cette correspondance dans la solution optimale de LP4HM. Le résultat d'intégration que nous obtenons est illustré dans la Figure III.26. Il s'agit d'un résultat satisfaisant à la fois en termes d'intégration hiérarchique des structures.*

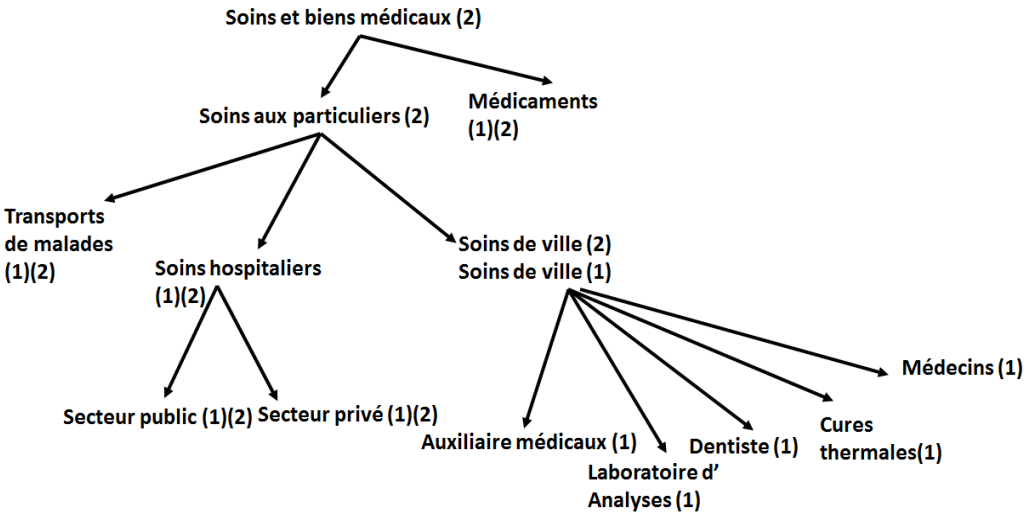


Figure III.26 — Résultat d'intégration en appliquant LP4HM avec ces quatre contraintes

3.3.4 Le modèle résultant

Le modèle complet du programme linéaire LP4HM se présente comme suit :

$$
\begin{cases}
\max \quad \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} sim_{i_k,j_l}\ x_{i_k,j_l} \\
s.t. \quad \sum_{l=1}^{n_j} x_{i_k,j_l} \leq 1,\ \forall k \in [1,n_i] \\
\qquad\qquad \forall i \in [1,N-1]\ \forall j \in [i+1,N] \\
x_{i_k,j_l} + x_{i_{k'},j_{l'}} - (dir_{i_{k,k'}} dir_{j_{l,l'}}) \leq 1 \\
\qquad\qquad \forall i \in [1,N-1]\ \forall j \in [i+1,N] \\
\qquad\qquad \forall k,k' \in [1,n_i],\ \forall l,l' \in [1,n_j] \\
x_{i_k,j_l} \leq x_{i_{pred(k)},j_{pred(l)}} \\
\qquad\qquad \forall i \in [1,N-1]\ \forall j \in [i+1,N] \\
\qquad\qquad \forall k \in [1,n_i],\ \forall l \in [1,n_j] \\
sim_{i_k,j_l}\ x_{i_k,j_l} \geq seuil\ x_{i_k,j_l} \\
\qquad\qquad \forall i \in [1,N-1]\ \forall j \in [i+1,N] \\
\qquad\qquad \forall k \in [1,n_i],\ \forall l \in [1,n_j] \\
x_{i_k,j_l} \in \{0,1\}\ \ \forall i \in [1,N-1]\ \forall j \in [i+1,N] \\
\qquad\qquad \forall k \in [1,n_i],\ \forall l \in [1,n_j]
\end{cases}
$$

Ce modèle comporte :
- $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} n_i n_j$ variables de décision.
- $\sum_{i=1}^{N-1} n_i\ (N-i)$ contraintes sur la cardinalité des correspondances.
- Au plus $\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |E_i||E_j|$ contraintes sur la direction des arcs.

– Au plus $\sum_{i=1}^{N-1} n_i - 1$ contraintes de structures hiérarchiques.
– $\sum_{i=1}^{N-1} n_i - 1$ contraintes sur le seuil de similarité.

Ce modèle génère un nombre important de variables de décisions et de contraintes linéaires. Néanmoins, il se résout très rapidement puisque le temps de résolution est polynomial [Almohamad et Duffuaa, 1993] en fonction de la taille des graphes.

3.3.5 La relaxation du programme linéaire

Au delà de l'application de notre programme linéaire pour l'intégration des données ouvertes, nous avons poussé notre réflexion pour qu'il puisse résoudre d'autres verrous de la littérature. En particulier, le problème de recherche de correspondances complexes de cardinalité $n : m$. La flexibilité de notre modèle nous a permis de répondre simplement à cette question. En effet, il suffit de relaxer les variables de décision binaires en variables de décision fractionnaires dans l'intervalle [0,1] pour obtenir des correspondances complexes.

Exemple 9. *Dans la Figure III.27 nous cherchons à identifier les correspondances entre deux graphes représentant les personnes. Comme la sim(name, first name) = sim(name, last name) et que leurs parents sont identiques, nous sommes dans un cas où name doit correspondre à {first name, last name}. Il s'agit donc d'une correspondance complexe. Toutefois, LP4HM retourne uniquement l'une des correspondances soit (name, first name) soit (name, last name). Dans ce cas nous perdons la deuxième correspondance qui est pourtant pertinente. En relaxant les variables de décision, les variables de décision prendront les valeurs 0.5 et 0.5 pour les deux correspondances. Il est alors possible à LP4HM relaxé de capter des correspondances complexes pertinentes.*

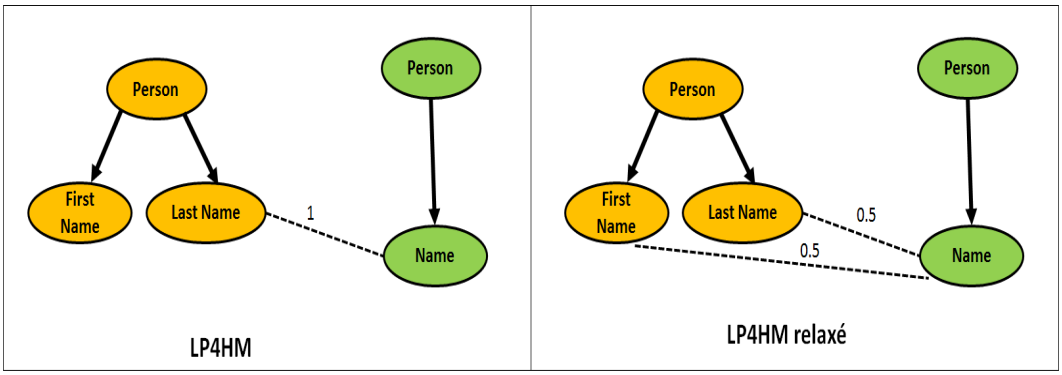


Figure III.27 — Un exemple de résolution de correspondance complexe par LP4HM relaxé

3.4 Regroupement des correspondances et construction du graphe intégré

La dernière partie de notre approche d'intégration des données consiste à regrouper les correspondances retournées par le programme linéaire LP4HM pour construire un graphe intégré.

3.4.1 Regroupement des correspondances

Les correspondances retournées par LP4HM, dont les valeurs de variables de décision sont égales à 1, sont stockées dans un tableau unique Mt. Ce tableau est de la forme suivante $\langle NoeudSrc, IdGraphSrc, NoeudTrg, IdGraphTrg, Sim(NoeudSrc, NoeudTrg)\rangle$.
Si $x_{i_k,j_l} = 1$ alors nous rajoutons un nouvel enregistrement dans Mt tel que :

- $NoeudSrc \leftarrow v_{i_k}$
- $IdGraphSrc \leftarrow i$,
- $NoeudTrg \leftarrow v_{j_l}$
- $IdGraphTrg \leftarrow j$
- $Sim(NoeudSrc, NoeudTrg) \leftarrow sim_{i_k,j_l}$.

Une fois ce tableau est construit, nous appliquons l'algorithme III.1 pour identifier les groupes de correspondances. Cela consiste à parcourir le tableau Mt tant qu'il n'est pas vide, prendre le premier enregistrement dans Mt et chercher toutes les correspondances qui ont un couple (noeud source, identifiant graphe source) ou un couple (noeud destination, identifiant graphe destination) identique au couple (noeud source, identifiant graphe source) de l'enregistrement en cours, voir les lignes 4-10 dans l'algorithme III.2. Ensuite il faut appliquer récursivement la fonction *GetInterCorresp* sur les autres couples (noeud, identifiant graphe) des correspondances qui avaient un couple commun avec l'enregistrement en cours.

Algorithme III.1 — ClusterMatchedVertices

```
1: Begin
2: groupCorresp ← ∅
3: InterCorresp ← ∅
4: while Mt ≠ ∅ do
5:    InterCorresp ← Mt.get(1)
6:    GetInterCorresp(Mt.get(1).NoeudSrc, Mt.get(1).IdGraphSrc, Mt, InterCorresp)
7:    groupCorresp ← groupCorresp ∪ InterCorresp
8: end while
9: End
```

3.4.2 Construction du graphe intégré

Dans le chapitre précédent, nous avons évoqué la possibilité de transformer les données ouvertes annotées en graphes de propriété ou en graphes RDF. Dans le deux premières phases de notre approche d'intégration, il s'agit des même données structurelles que nous avons comparées pour extraire les correspondances. Mais au niveau de la construction du graphe intégré il y a quelques différences dans la matérialisation des données même si le principe est identique. Nous expliquons dans ce qui suit comment nous procédons dans cette dernière phase d'intégration.

Algorithme III.2 — GetInterCorresp(NoeudSrcInput,IdGraphSrcInput, Mt, InterCorresp)

```
1: Begin
2: InterCorrespTmp ← ∅
3: for each m ∈ Mt do
4:   if m.NoeudSrc = NoeudSrcInput and m.IdGraphSrc = IdGraphSrcInput then
5:     InterCorrespTmp ← InterCorrespTmp ∪ m
6:     Mt ← Mt \ {m}
7:   end if
8:   if m.NoeudTrg = NoeudSrcInput and m.IdGraphTrg = IdGraphSrcInput then
9:     InterCorrespTmp ← InterCorrespTmp ∪ m
10:    Mt ← Mt \ m
11:  end if
12: end for
13: InterCorresp ← InterCorresp ∪ InterCorrespTmp
14: for each i ∈ InterCorrespTmp do
15:   node ← getReflexiveNode(InterCorrespTmp(i), NoeudSrcInput)
16:   idGraph ← getReflexiveId(InterCorrespTmp(i), IdGraphSrcInput)
17:   GetSimilarVertices(node, idGraph, Mt, InterCorresp)
18: end for
19: End
```

3.4.2.1 Un graphe de propriété intégré

Pour chaque groupe de correspondances, nous créons un nouveau noeud de type noeud imbriqué. Ce dernier est un noeud mais qui a la structure d'un graphe composé par différents noeuds. Nous attribuons au noeud imbriqué la *valeur* du label commun à la majorité des noeuds ou bien s'il n'y pas de majorité n'importe quel label de la liste des noeuds du groupe. Nous ajoutons les identifiants des noeuds du groupe de correspondances comme des sous noeuds du noeud imbriqué. Par la suite, nous examinons chaque correspondance dans le groupe de correspondance et nous ajoutons un arc non-orienté entre le noeud source du graphe source et le noeud destination du graphe destination. Enfin, les arcs qui reliaient les noeuds des correspondances du groupe actuel avec d'autres noeuds de correspondances d'autres groupes seront modifiés par un arc entre les noeuds imbriqués représentatifs de chaque groupe. D'autre part, s'il existe un arc entre un groupe de correspondances et un noeud qui n'a pas été impliqué dans une correspondance, cet arc sera remplacé par un arc entre le noeud imbriqué et le noeud non apparié.

3.4.2.2 Un graphe RDF intégré

Pour chaque groupe de correspondances, nous créons un nouveau noeud de type *skoks* : *Collection* qui représente un regroupement de concept, nous lui attribuons comme *skoks* : *label* le label commun à la majorité des noeuds ou bien s'il n'y pas de majorité n'importe quel label de la liste des noeuds. Ensuite, nous rajoutons des propriétés *skoks* : *member* entre ce nouveau noeud et les identifiants des noeuds qui figurent dans le groupe.

Par ailleurs, nous matérialisons la relation de correspondance entre les différents noeuds structurels (qui sont des *skoks* : *concept* dans le graphe RDF). En effet, si la $Sim(NoeudSrc, NoeudTrg) = 1$ alors nous ajoutons la propriété *skoks* : *exactMatch* entre le noeud source et le noeud destination sinon nous ajoutons la propriété *skos* : *closeMatch* entre ces deux noeuds.

Le graphe RDF intégré représente notre solution pour générer des données ouvertes liées à partir des données tabulaires.

4 Conclusion

Dans ce chapitre, nous avons présenté une méthode d'intégration holistique de données ouvertes tabulaires. La méthode repose sur une présentation en graphes des données tabulaires (obtenus à l'issue de la phase 1 décrite au chapitre précédent). L'intégration holistique nous permet de prendre en compte simultanément $N \geq 2$ graphes. L'intérêt de cette méthode est de garantir une solution unique correspondant à l'optimum global. En effet, l'intégration par paire de graphes à pour inconvénient de trouver une solution localement optimale dans l'espace des solutions formé par l'ensemble des graphes. De plus, suivant l'ordre avec lequel l'appariement par paire de graphes est effectué, la solution optimale (locale) est le plus souvent différente. Notre solution est plus facilement exploitable pour un utilisateur car il dispose d'un graphe intégré unique toujours identique quelque soit l'ordre d'intégration des graphes.

Notre méthode combine plusieurs mesures de similarité. Nous avons adopté des mesures syntaxiques et sémantiques pour assurer une meilleure mise en correspondance des noeuds des graphes. Nous exploitons la complémentarité des similarités syntaxiques et sémantiques en maximisant les scores obtenus.

Notre méthode d'intégration adapte et étend le problème de couplage de graphes à poids maximal, connu en optimisation combinatoire. Nous avons modélisé notre méthode sous la forme d'un programme linéaire, nommé LP4HM, afin de garantir une résolution du problème en temps polynomial [Almohamad et Duffuaa, 1993] [Schrijver, 2003]. Ce choix est confirmé par nos expérimentations décrites dans le chapitre 5. Un autre intérêt de la programmation linéaire réside dans la possibilité de définir un ensemble de contraintes. Ces contraintes nous permettent de modéliser des appariements cohérents.

Cette approche a également la possibilité de se passer de la configuration de seuil de similarité. Fixer un seuil dans les outils d'appariement est une tâche particulièrement difficile pour les utilisateurs. Elle nécessite parfois une phase d'apprentissage. Notre méthode rend possible l'appariement de plusieurs graphes sans apprentissage et sans l'utilisation du seuil. L'intérêt de ce choix est confirmé par les résultats d'expérimentations présentées au chapitre 5.

Enfin, notre méthode peut être étendue pour prendre en compte des appariements complexes. L'appariement complexe permet de faire correspondre plusieurs noeuds d'un graphe avec plusieurs noeuds d'un autre graphe ; on parle d'appariement $n : m$. Pour permettre cela, la variable de décision du programme linéaire est relaxée en prenant ses valeurs dans

l'intervalle $[0, 1]$ au lieu d'être contrainte aux valeurs binaires 0 et 1.

Ces propositions ont été publiées dans le cadre de la conférence nationale EDA'15 [Berro *et al.*, 2015c] et les conférences internationales RCIS'15 [Berro *et al.*, 2015b] et DEXA'15 [Berro *et al.*, 2015d].

La méthode d'intégration définie dans ce chapitre permet d'obtenir automatiquement un graphe intégré. Le chapitre suivant montre comment un utilisateur peut définir progressivement une base de données multidimensionnelle à partir du graphe intégré.

IV Conception de schémas multidimensionnels

L'analyse OLAP repose sur un schéma multidimensionnel basé sur la dichotomie fait/dimension [Kimball, 1996] [Ravat *et al.*, 2001] [Ravat *et al.*, 2008], [Abello *et al.*, 2015]. Ce schéma couramment conçu à partir de sources relationnelles [Romero et Abelló, 2009] doit être élaboré, dans notre contexte, à partir du graphe intégré de données ouvertes tabulaires. Dans ce chapitre nous répondons à deux questions. Comment faut-il simplement exploiter ce graphe pour concevoir un schéma multidimensionnel ? Est-il possible de ne pas matérialiser les données dans un entrepôt de données multidimensionnelles tout en permettant leurs analyse ?

1 Introduction

Les opérations OLAP constituent une solution pour l'analyse des données statistiques du web [Kämpgen *et al.*, 2012]. Ces opérations reposent sur un schéma multidimensionnel qui est traditionnellement extrait de sources structurées (relationnelles) à travers les processus ETL et implanté dans un entrepôt de données. Or, les données statistiques disponibles sur le web sont des sources non-structurées, en l'occurrence les données ouvertes tabulaires, ou semi-structurées telles que des données exprimées en RDF ou XML [Ravat *et al.*, 2010]. Ces sources exigent une adaptation des démarches de conception de schémas multidimensionnels [Romero et Abelló, 2009].

Dans la littérature, certaines approches [Romero et Abelló, 2007] [Danger et Berlanga, 2009] ont proposé de concevoir des schémas multidimensionnels à partir d'ontologies. Ces approches sont conditionnées par l'expressivité des ontologies, notamment au niveau des cardinalités entre les relations. D'autres approches de la littérature partent d'un schéma multidimensionnel en QB [Kämpgen *et al.*, 2012] ou génèrent des données RDF annotées avec un vocabulaire multidimensionnel [Etcheverry *et al.*, 2014]. Leur objectif est d'interroger directement les données du web sans passer par la matérialisation des données. Bien que la matérialisation est un principe fondamental favorisant la performance des requêtes OLAP [Laborie *et al.*, 2015], elle se trouve remise en cause dans le cadre d'OLAP exploratoire ou les processus ETQ (Extract-Transform-Query) [Abello *et al.*, 2015]. Nous constatons que les approches de la littérature divergent sur la matérialisation des données. Dans le cadre de notre étude, nous nous adressons à des utilisateurs qui ne sont pas forcément des experts en décisionnel. Nous partirons dans l'hypothèse où ces utilisateurs peuvent avoir des avis différents [Ravat et Teste, 2008] sur la matérialisation. Pour cela, nous proposons qu'un processus de conception puisse supporter les deux approches.

Après avoir extrait et intégré des données ouvertes tabulaires, nous souhaitons que l'utilisateur puisse appliquer des opérations OLAP sur ces données intégrées. Il convient alors de modéliser ces données selon la dichotomie fait/dimension. Nos données ouvertes tabulaires ont été transformées dans un graphe intégré. Ce graphe est formé de données numériques liées à des données conceptuelles organisées en hiérarchies. Nous proposons un processus progressif de conception mutlidimensionnelle. Notre processus supporte la matérialisation et la non-matérialisation des données d'une façon complètement transparente à l'utilisateur. Ce processus se compose de deux vues :

- Une vue utilisateur dans laquelle le graphe intégré se transforme progressivement en un schéma multidimensionnel de niveau conceptuel selon le formalisme graphique du modèle conceptuel proposé par [Ravat *et al.*, 2007b]. L'utilisateur identifie progressivement les composants multidimensionnels. En effet, il identifie d'abord les dimensions et leurs composants. Puis, il identifie les faits et leurs composants.
- Une vue système dans laquelle le système se charge d'appliquer les étapes nécessaires pour matérialiser ou non-matérialiser les données en fonction du choix de l'utilisateur.
  - Pour matérialiser les données, le système génère progressivement les scripts sql permettant de matérialiser un entrepôt de données dans une base de données ROLAP. Suite à ceci, les données peuvent être interrogées par des opérations OLAP [Ravat *et al.*, 2002].
  - Pour non-matérialiser les données, le système produit progressivement des annotations multidimensionnelles dans un graphe RDF équivalent au graphe intégré visualisé par le concepteur. La production des annotations repose sur les correspondances entre le modèle conceptuel multidimensionnel [Ravat *et al.*, 2007b] et le vocabulaire multidimensionnel QB4OLAP [Etcheverry *et al.*, 2014]. Il serait possible d'interroger directement ce graphe avec des opérations OLAP-SPARQL [Etcheverry *et al.*, 2014]. L'usage des graphes montre ici son intérêt puisque notre démarche ETL sans matérialisation peut devenir une démarche ETQ [Abello *et al.*, 2015].

Ce chapitre est divisé en deux parties. La première partie est un état de l'art sur les travaux qui utilisent des sources non-relationnelles (telles que des ontologies, des schémas XML, des données liées, etc..) pour la conception d'un schéma multidimensionnel et l'alimentation d'un entrepôt de données. La deuxième partie est dédiée à la description de notre proposition pour la conception et l'annotation multidimensionnelle à partir du graphe intégré des données ouvertes tabulaires.

2 État de l'art

Dans cette section, nous mettons l'accent sur les travaux de la littérature qui portent sur la conception d'un schéma multidimensionnel [Ghozzi *et al.*, 2005] et sur l'alimentation d'un entrepôt de données [Annoni *et al.*, 2006b]. Les travaux ciblés utilisent des sources de données conceptuelles [Khouri, 2013] structurées ou non-structurées telles que les ontologies ou les schémas XML.

2.1 La conception d'un schéma multidimensionnel

Dans la littérature, plusieurs démarches ont été proposées [Romero et Abelló, 2009] pour élaborer un schéma multidimensionnel. Un tel schéma est conforme à un modèle multidimensionnel de niveau d'abstraction conceptuel, logique ou physique [Teste, 2009] :

- **Le niveau conceptuel** fournit une représentation qui se base sur la dichotomie fait/dimension [Abello *et al.*, 2015]. Ce modèle est facile à interpréter par les utilisateurs et indépendant de toutes contraintes d'implantation [Rizzi *et al.*, 2006]. Trois catégories [Rizzi *et al.*, 2006] de modèles conceptuels ont été proposées dans la littérature :
  - les modèles basés sur le paradigme entité/association [Sapia *et al.*, 1999], [Hahn *et al.*, 2000],[Malinowski et Zimányi, 2006], [Malinowski et Zimanyi, 2008].
  - les modèles basés sur le paradigme objet [Buzydlowski *et al.*, 1998], [Trujillo et Palomar, 1998], [Ravat *et al.*, 1999], [Ravat et Teste, 2000], [Pedersen *et al.*, 2001], [Abello, 2002], [Trujillo *et al.*, 2003], [Annoni *et al.*, 2006b], [Abello *et al.*, 2006].
  - les modèles spécifiques [Golfarelli *et al.*, 1998], [Cabibbo et Torlone, 2000], [Ravat *et al.*, 2001], [Schneider, 2003], [Ghozzi *et al.*, 2005], [Tournier, 2007], [Schneider, 2008].
- **Le niveau logique** fournit un modèle logique issu de la transformation du modèle conceptuel [Rizzi *et al.*, 2006] selon les caractéristiques d'une technologie cible (relationnel, objet...). Quatre types de modèles logiques sont utilisés :
  - le modèle relationnel **R-OLAP** transforme chaque fait et chaque dimension du modèle conceptuel en des tables relationnelles [Kimball, 1996]
  - le modèle multidimensionnel **M-OLAP** implante le modèle conceptuel sous forme multidimensionnelle telle que des cubes de données, des matrices ou des vecteurs à n dimensions [Agrawal *et al.*, 1997]
  - le modèle hybride **H-OLAP** implante les données détaillées dans des tables relationnelles et les données agrégées dans des matrices multidimensionnelles.
  - le modèle NoSQL implantent les données dans quatre approches différentes : clé-valeur, graphe, orientée-documents et orientée-colonnes [Dehdouh *et al.*, 2014] [Chevalier *et al.*, 2015b] [Chevalier *et al.*, 2015a].
- **Le niveau physique** fournit une implémentation du modèle logique dans un système de gestion de base de données ou un système OLAP particulier (Oracle, Sql Server...).

Chaque démarche de conception suit l'un des paradigmes suivants [Annoni *et al.*, 2006a] [Romero et Abelló, 2009] :

- **Descendant**, dit aussi dirigé par les besoins, dont le principe est de partir des besoins explicites ou implicites des utilisateurs pour construire le schéma multidimensionnel puis faire correspondre ce dernier avec les schémas des sources pour alimenter l'entrepôt de données.
- **Ascendant**, dit aussi dirigé par les données, dont le principe est d'appliquer un processus de rétro-conception sur les sources de données pour obtenir le schéma multidimensionnel.
- **Hybride** qui consiste à impliquer les besoins des utilisateurs et l'analyse des sources de données pour la conception du schéma multidimensionnel. Le paradigme hybride

peut être séquentiel ou intercalé. Dans le paradigme hybride séquentiel, on applique séparément une conception à partir des besoins des utilisateurs et une autre conception à partir des sources des données puis on réconcilie les deux résultats de conception. Dans le paradigme hybride intercalé, les deux stratégies sont appliquées itérativement profitant ainsi des retours du concepteur tout au long de la conception [Annoni *et al.*, 2006a].

La plupart des travaux [Golfarelli *et al.*, 1998], [Hüsemann *et al.*, 2000], [Moody et Kortink, 2000], [Bonifati *et al.*, 2001], [Phipps et Davis, 2002], [Giorgini *et al.*, 2005], [Prat *et al.*, 2006], [Atigui, 2013], [Abdelhédi, 2014] traitent des sources de données classiques sous forme de schéma de bases de données relationnelles et optent pour un paradigme hybride. Parmi ces travaux, [Hüsemann *et al.*, 2000], [Moody et Kortink, 2000], [Phipps et Davis, 2002] se basent sur un modèle conceptuel Entité/Association, certains [Prat *et al.*, 2006], [Annoni *et al.*, 2006a] exploitent un modèle conceptuel UML, et d'autres [Golfarelli *et al.*, 1998], [Bonifati *et al.*, 2001], [Giorgini *et al.*, 2005], [Ravat *et al.*, 2006a], [Abdelhédi, 2014] se basent sur un modèle conceptuel spécifique.

D'un autre côté, très peu de travaux [Vrdoljak *et al.*, 2003], [Romero et Abelló, 2007], [Danger et Berlanga, 2009], [Nebot *et al.*, 2009] ont exploité la conception d'un schéma multidimensionnel à partir de sources ontologiques ou XML. Nous nous intéressons particulièrement à ces travaux que nous décrivons dans la section suivante.

2.1.1 Étude des travaux

[Vrdoljak *et al.*, 2003] ont proposé une approche semi-automatique pour la conception d'un schéma multidimensionnel à partir d'un schéma XML. La démarche comprend quatre étapes : (1) le pré-traitement des schémas XML, (2) la création et la transformation de schéma XML en un graphe où les dépendances fonctionnelles sont explicitement représentées, (3) l'identification du noeud de fait dans le graphe et (4) la construction d'un schéma multidimensionnel pour chaque fait qui englobe la construction d'un graphe de dépendance dont la racine est le fait et l'identification des relations de cardinalité $1:n$ ou $n:m$ pour définir les dimensions et les hiérarchies des dimensions.

[Romero et Abelló, 2007] ont proposé une approche semi-automatique pour la conception d'un schéma multidimensionnel à partir d'une ontologie OWL. L'approche est fondée sur différentes contraintes y compris des contraintes pour éviter les problèmes d'additivité (cf. §2.1.2). L'approche se définit en trois étapes, d'abord la détection des faits puis la détection des plus bas niveaux des dimensions enfin la détection des hiérarchies des dimensions. Pour la détection des faits, les auteurs proposent un algorithme pour identifier à chaque concept de l'ontologie les potentielles dimensions et mesures, puis l'utilisateur intervient pour désigner, parmi ces concepts, celui qui représente le fait. L'algorithme de découverte des dimensions est fondé sur l'idée de recherche d'une composition de propriétés r entre deux concepts A et B qui vérifient qu'une instance du concept A (potentiel fait) doit être reliée via r à une seule instance du concept B (potentielle dimension). Pour chaque concept, les auteurs examinent les types des attributs et ils désignent un seul attribut numérique comme une potentielle mesure. Pour la détection des plus bas niveaux des dimension (ou

bases des dimensions), les auteurs proposent un algorithme heuristique qui suit différentes règles. En effet, l'ensemble des bases des dimensions doit être minimal, les bases de cet ensemble doivent être orthogonales et les niveaux des dimensions intermédiaires entre le fait et chaque base doivent pouvoir jouer le rôle de base de dimension. La dernière étape de cette approche consiste à détecter les hiérarchies des dimensions. L'idée proposée est de chercher pour chaque base de dimension un graphe orienté formé par des propriétés de cardinalité $1 : n$, les concepts qui se trouvent dans ce graphe peuvent être soit des attributs d'un niveau de dimension, soit un niveau de dimension.

[Danger et Berlanga, 2009] ont proposé une approche pour l'analyse des instances dans une ontologie. Les auteurs proposent des algorithmes pour l'identification des dimensions et des hiérarchies des dimensions. L'algorithme d'identification de hiérarchie sélectionne les relations qui maximisent le gain d'informations. Les auteurs utilisent un modèle multi-dimensionnel spécifique qui applique les opérations de sélection, regroupement, agrégation sur les instances de l'ontologie.

[Nebot *et al.*, 2009] ont proposé un framework pour la conception d'un schéma conceptuel d'entrepôt semi-structuré de données XML et RDF fournies par le web sémantique. A partir d'ontologies de domaines, le concepteur définit manuellement les concepts multidimensionnels de faits, dimensions, mesures et hiérarchies des dimensions. Ainsi, une ontologie applicative est générée à partir des composants multidimensionnels identifiés par le concepteur en utilisant le langage OntoPath [Jiménez-Ruiz *et al.*, 2007]. Ce dernier est un langage permettant de rechercher un fragment d'ontologie. Les auteurs proposent aussi de vérifier certaines contraintes multidimensionnelles dans le schéma de l'ontologie applicative.

[Mansmann *et al.*, 2014] ont proposé une approche pour la découverte et l'enrichissement de modèles multidimensionnels des données semi-structurées issues de tweets. Nous nous intéressons notamment à la partie découverte. Les tweets de format JSON sont transformés en fichiers de format XML. Les auteurs appliquent un processus de rétro-conception pour obtenir un schéma relationnel des tweets. A partir de ce dernier, ils conçoivent semi-automatiquement le schéma multidimensionnel en deux temps : (1) les dimensions et les faits sont définis manuellement par les auteurs en interprétant le schéma relationnel des tweets. En particulier, les attributs numériques sont les mesures et les attributs descriptifs sont les dimensions ou les niveaux des dimensions. Le fait est l'événement du tweet et (2) les cardinalités entre les dimensions et les faits ont été déduites automatiquement en se basant sur les travaux de [Mansmann, 2008]. Ils ont opté pour le modèle conceptuel x-DFM [Mansmann, 2008] dans lequel les dimensions sont structurées en graphe. Les noeuds du graphe sont les niveaux des dimensions et les arcs du graphe sont les liens roll-up.

Parmi ces travaux, nous constatons que seules les propositions de [Romero et Abelló, 2007] et de [Nebot *et al.*, 2009] anticipent le problème d'additivité

lors de la phase de conception d'un schéma multidimensionnel à partir de sources non-traditionnelles. Nous présentons plus en détail ce problème dans la section suivante.

2.1.2 Le problème d'additivité

Assurer l'additivité (le terme anglophone est *summarizability*) désigne une application correcte de l'opérateur de forage vers le haut (roll-up) [Hassan *et al.*, 2012] [Hassan *et al.*, 2013] d'un niveau inférieur à un niveau supérieur dans une hiérarchie. Il s'agit d'un problème incontournable pour assurer une fiabilité des résultats d'analyse OLAP. [Rafanelli et Shoshani, 1990] est le premier travail à détecter ce problème pour les bases de données statistiques. Pour assurer l'additivité, il propose trois contraintes sur les cardinalités des relations entre les niveaux d'une hiérarchie :

1. La cardinalité maximale associée à la relation entre un niveau inférieur et un niveau supérieur dans une hiérarchie ne peut être que 1.
2. La cardinalité associée à une relation de niveau supérieur au niveau inférieur dans une hiérarchie doit être de type 1 : n
3. La cardinalité minimale associée à la relation entre un niveau inférieur et un niveau supérieur dans une hiérarchie ne peut pas être de valeur zéro.

[Lenz et Shoshani, 1997] sont les premiers à pointer le problème d'additivité pour les bases de données multidimensionnelles. Ils définissent trois conditions nécessaires qui valident l'additivité entre fait-dimension et entre les hiérarchies d'une dimension :

- La disjonction entre l'ensemble des instances appartenant à deux catégories différentes. Par exemple, dans une hiérarchie, les instances des niveaux feuilles ou intermédiaires doivent avoir au plus un seul parent de leur niveau supérieur. Cette condition est équivalente à la contrainte (1) de [Rafanelli et Shoshani, 1990] et sa vérification assure des hiérarchies strictes dans une dimension [Pedersen *et al.*, 1999].
- La complétude des instances manquantes et des instances non-affectées. Les instances manquantes peuvent figurer entre fait et dimension ou dans les hiérarchies d'une dimension. L'absence d'instances dans une hiérarchie correspond aux hiérarchies non-ontologiques (des instances manquantes au niveau feuille de la hiérarchie) et aux hiérarchies non-couvrantes (des instances manquantes dans des niveaux intermédiaires) [Pedersen *et al.*, 1999]. Les instances non-affectées correspondent à une cardinalité de 0 entre niveaux hiérarchiques ceci est équivalent à la contrainte (3) de [Rafanelli et Shoshani, 1990].
- La compatibilité du type de la fonction d'agrégation par rapport au type de mesure et à la nature de dimension analysée. Les mesures se classifient en stock, flux et valeurs unitaires. Quant aux dimensions, elles sont soit temporelles soit non-temporelles. Pour assurer l'additivité, deux règles doivent être prises en compte : (1) il ne faut pas appliquer la fonction d'agrégation somme sur n'importe quel type de mesure pour les dimensions temporelles et (2) il ne faut pas appliquer la fonction d'agrégation somme sur les mesures de type valeurs unitaires pour les dimensions non-temporelles.

En plus des travaux de [Danger et Berlanga, 2009] et [Nebot *et al.*, 2009] qui anticipent le problème d'additivité lors de la conception d'un schéma multidimensionnel, nous identifions trois autres catégories de travaux qui s'intéressent à ce problème. Certains travaux

tels que [Prat *et al.*, 2012] détectent le problème d'additivité dans les schémas ontologiques d'entrepôt de données par le biais de règles décrites en OWL-DL, mais ces auteurs ne proposent pas de solutions à ce problème. D'autres travaux tels que [Pedersen *et al.*, 1999] et [Mansmann et Scholl, 2007] détectent le problème et le traitent sur les instances du schéma conceptuel. [Pedersen *et al.*, 1999] a proposé trois algorithmes MakeCover, MakeOnto et MakeStrict pour corriger ces problèmes. Quant à [Mansmann et Scholl, 2007], ils ont proposé une version plus sémantique des algorithmes MakeCover et MakeOnto de [Pedersen *et al.*, 1999] en rajoutant une instance "Autre" dans les niveaux intermédiaires des hiérarchies non-couvrantes et en rajoutant des instances feuilles "Autre" pour les hiérarchies non-ontologiques. Une dernière catégorie de travaux tels que [Horner et Song, 2005] et [Hachicha, 2012] détectent et traitent le problème d'additivité en temps réels à la phase d'analyse.

2.1.3 Synthèse et positionnement

Les approches qui exploitent des sources ontologiques pour la conception d'un schéma multidimensionnel peuvent atteindre un niveau d'automatisation plus important [Abello *et al.*, 2015] que les approches traditionnelles qui traitent les schémas relationnels. Toutefois, ceci dépend fortement de l'expressivité de ces sources, notamment l'applicabilité des algorithmes de [Romero et Abelló, 2007] ou de [Vrdoljak *et al.*, 2003] dépend de la présence de cardinalités ou des dépendances fonctionnelles entre les concepts. Ces mêmes limites ont été aussi soulignées par les auteurs [Nebot *et al.*, 2009]. Ces derniers définissent manuellement le schéma multidimensionnel à partir d'une ontologie de domaine afin de chercher automatiquement des instances conformes à ce schéma dans les données RDF et XML.

Notre proposition consiste à concevoir un schéma multidimensionnel à partir des données non-relationnelles. Elle prend en entrée un graphe intégré qui présente des données tabulaires issues de plusieurs sources. Ce graphe peut contenir d'une part des données structurelles potentiellement des composants du schéma multidimensionnel, d'autre part des données structurelles potentiellement instances des composants multidimensionnels. L'organisation des hiérarchies de données structurelles autour des données numériques dans notre graphe intégré rappelle l'organisation des dimensions autour d'un fait. Les algorithmes proposées par [Romero et Abelló, 2007] et [Vrdoljak *et al.*, 2003] sont non exploitables dans notre graphe vu le manque d'expressivité des cardinalités. Par contre les algorithmes de parcours de graphes sont tout à fait applicables. Par rapport à l'approche de [Nebot *et al.*, 2009], nous n'avons pas d'ontologie de domaine sur laquelle nous pouvons nous appuyer. Le concepteur utilisera le graphe pour identifier ou définir manuellement les composants multidimensionnels tout en étant assisté par le système. L'originalité de notre proposition est d'exploiter la définition manuelle du schéma multidimensionnel pour générer automatiquement les annotations multidimensionnelles dans un graphe RDF équivalent au graphe manipulé par le concepteur.

Par ailleurs, dans notre démarche d'entreposage, nous avons anticipé la génération de hiérarchies strictes dans ses deux premières phases. Concernant, les hiérarchies non-couvrantes et non-ontologiques, nous avons opté pour la proposition

de [Mansmann et Scholl, 2007]. Pour assurer l'additivité des fonctions d'agrégation, nous proposons de définir plusieurs fonctions d'agrégation sur plusieurs niveaux [Hassan *et al.*, 2015] [Hassan, 2014]. Nous proposons aussi d'implémenter les axiomes proposées par [Prat *et al.*, 2012] dans le graphe RDF intégré annoté pour vérifier l'additivité en fonction du type de la dimension. Nous pouvons constater que nous ne proposons pas une nouvelle méthode pour détecter et traiter le problème d'additivité mais nous combinons des méthodes existantes pour automatiser le plus possible la résolution de ce problème.

2.2 L'alimentation d'un entrepôt de données

Une importante partie des travaux de la littérature a été focalisée sur les processus ETL qui permettent d'intégrer les sources de données et d'alimenter un entrepôt de données. Dans l'étude proposée par [Khouri, 2013], ces travaux se classifient en trois catégories :

- Des travaux, dédiés à l'intégration des données, qui visionnent un ED comme un système d'intégration de données [Khouri, 2013]. Ces travaux modélisent les transformations permettant de réconcilier un schéma global avec les schémas des sources de données [Müller *et al.*, 1999], [Calvanese *et al.*, 2001], [Skoutas et Simitsis, 2007], [Serment *et al.*, 2008], [Romero *et al.*, 2011], [Bergamaschi *et al.*, 2011].
- Des travaux, dédiés à la modélisation des processus ETL, qui formalisent les activités ETL au niveau conceptuel [Vassiliadis *et al.*, 2002] [Trujillo et Luján-Mora, 2003], [Khouri, 2013], [Atigui, 2013], logique [Vassiliadis *et al.*, 2002], [Vassiliadis *et al.*, 2005] ou physique [Luján-Mora et Trujillo, 2006], [Tziovara *et al.*, 2007].
- Des travaux, dédiés à l'alimentation d'un entrepôt de données, qui décrivent les processus permettant de produire les données de table de faits et de dimensions [Labio *et al.*, 2000], [Kämpgen et Harth, 2011], [Nebot et Berlanga, 2012] et de [Inoue *et al.*, 2013].

2.2.1 Étude des travaux

[Labio *et al.*, 2000] ont proposé un framework décrivant les activités abouties ou non lors de l'alimentation d'un entrepôt de données par des sources relationnelles. Ils définissent un algorithme permettant l'identification et le filtrage des tuples pertinents à recharger lors d'un échec d'alimentation.

[Kämpgen et Harth, 2011] ont proposé une approche pour l'entreposage des données RDF-QB dans un entrepôt de données relationnels. Les sources de données RDF-QB représentent des cubes de données dans l'univers du web sémantique. Les auteurs proposent un système ETL pour placer ces données dans un entrepôt relationnel ROLAP afin de bénéficier de la stabilité et la performance des outils OLAP disponibles. Les auteurs ont établi des règles de mapping entre leur modèle multidimensionnel conceptuel et les termes du vocabulaire QB décrivant les dimensions, faits et mesures. Ils interrogent avec SPARQL les sources de données, proposées par l'utilisateur, pour identifier les composants du modèle multidimensionnel et le construire. Ensuite, ils alimentent ce modèle multidimensionnel par les instances qui sont également extraites par les requêtes SPARQL.

[Nebot et Berlanga, 2012] ont proposé une approche semi-automatique pour la généra-

tion d'un entrepôt de données à partir de données RDF/OWL. En pré-requis, un utilisateur doit définir manuellement un schéma multidimensionnel de l'entrepôt en choisissant le fait, les mesures, et les dimensions parmi les concepts d'une ontologie. L'approche proposée utilise le schéma multidimensionnel et l'instance de l'ontologie en RDF pour identifier et extraire dans un premier temps les instances de la table de fait. Dans un deuxième temps, les auteurs proposent d'extraire les hiérarchies des dimensions en utilisant les données de la table de fait et les instances de l'ontologie. L'extraction de la table de fait est proposée comme un processus ETL composé de quatre étapes : (1) l'extraction, à partir d'une ontologie, de l'ensemble des triplets (s,p,o) tel que s est le concept de fait, o est un concept agrégé entre mesure et dimension et p est une chaîne de propriétés entre s et o, (2) la génération des instances des triplets à partir de l'instance de l'ontologie et par sélection des instances valides, (3) la projection des instances des triplets sur les dimensions et mesures pour construire la table de fait, (4) la transformation du fait en appliquant des calculs et agrégations sur les mesures. Pour la reconstruction des hiérarchies des dimensions, les auteurs proposent d'extraire les taxonomies dont la racine est le concept de la dimension et de reconstruire une hiérarchie sur les noeuds denses de la taxonomie reliés par des propriétés transitives.

[Inoue *et al.*, 2013] ont proposé un framework ETL pour l'entreposage relationnel des données ouvertes liées (LOD) non-décrites par un vocabulaire multidimensionnel. L'approche est découpée en trois phases : (1) la transformation des triplets RDF en tables de propriétés relationnelles. L'idée est de regrouper les triplets RDF en partitions qui ont le même type pour la propriété rdf :type. Puis les propriétés d'une partition sont extraites pour construire une table relationnelle dont les attributs sont les propriétés, et les tuples sont les sujets des triplets et leurs valeurs. Les auteurs estiment la cardinalité des relations entre tables de propriétés en fonction du nombre d'instances dans une table, (2) l'utilisateur doit sélectionner une table de fait parmi les tables de propriétés et choisir un attribut numérique qui représente la mesure. Le système utilise la table de fait pour déduire les dimensions. En effet, il propose de hiérarchiser les attributs spatio-temporels selon les hiérarchies des ressources externes Geonames ou Time Ontology. Pour les autres attributs, il exploite l'auto-référencement dans les tables de propriétés pour induire des hiérarchies, (3) l'utilisateur doit choisir parmi les potentielles dimensions. Le système génère le schéma logique R-OLAP et alimente l'entrepôt de données relationnelles en projetant les attributs validés des tables de propriétés sur les tables de dimension et de fait.

Lors de l'alimentation d'un entrepôt de données, la plupart des approches se heurtent aux problèmes de qualité des données. Nous rappelons brièvement les types de problèmes de qualité dans la section suivante.

2.2.2 Le problème de qualité des données

La qualité des données est un problème particulièrement important pour les entrepôt de données [English, 1999] et [Shin, 2003] d'autant plus que l'objectif est d'aider à la prise de décision. La philosophie "publier d'abord puis raffiner" des données du web accentue d'avantage la présence des problèmes de qualité dans les données provenant du web [Zaveri *et al.*, 2014]. Il existe quatre catégories de problèmes de qualité de données

[Berti-Equille et Moussouni, 2005], [Equille, 2012] :

- les problèmes des données dupliquées ou ambiguës qui peuvent être résolus par l'élimination des duplicatas [Ananthakrishna *et al.*, 2002], la désambiguïsation des noms et la résolution des entités [Benjelloun *et al.*, 2009].
- les problèmes des données inconsistantes et conflictuelles provenant de plusieurs sources qui peuvent être résolus par les techniques de fusion de données récemment proposées dans [Li *et al.*, 2012].
- les problèmes des données manquantes et incomplètes qui peuvent être résolus par les techniques d'imputation de données [Zaamoune *et al.*, 2013] [Tsikriktsis, 2005].
- les problèmes des données obsolètes par manque de fraîcheur qui peuvent être résolus par mises à jour et rafraîchissement des données.

2.2.3 Synthèse et positionnement

Les approches que nous avons présentées dans la section 2.2.1 exploitent les données instances pour alimenter un entrepôt de données. [Labio *et al.*, 2000], [Kämpgen et Harth, 2011] et [Inoue *et al.*, 2013] se focalisent sur un entrepôt relationnel classique tandis que [Nebot et Berlanga, 2012] génèrent un entrepôt sémantique en RDF. [Nebot et Berlanga, 2012] et [Labio *et al.*, 2000] utilisent en pré-acquis un schéma multidimensionnel alors que [Kämpgen et Harth, 2011] et [Inoue *et al.*, 2013] déduisent le schéma en même temps que l'alimentation. Nous utilisons un graphe de propriétés pour définir graphiquement le schéma et les instances d'un entrepôt relationnel ; nous annotons en même temps un graphe RDF par un vocabulaire multidimensionnel qui est le point de départ de l'approche de [Kämpgen et Harth, 2011]. Notre proposition est assez similaire à la proposition de [Inoue *et al.*, 2013] avec en plus un traitement du problème des hiérarchies complexes.

Concernant le problème de qualité des données, les approches de [Labio *et al.*, 2000] et [Kämpgen et Harth, 2011] ne semblent pas s'en occuper. L'approche de [Nebot et Berlanga, 2012] fixe des règles pour éviter les données inconsistantes et accepte des données manquantes. L'approche de [Inoue *et al.*, 2013] ne traite pas les données inconsistantes et accepte aussi les données manquantes. Quant à notre approche, nous traitons les données inconsistantes par fusion. Nous utilisons la technique de vote en acceptant les données manquantes afin d'éviter tous biais qui peut être introduit lors de l'imputation des données.

3 Un processus progressif de conception multidimensionnelle

Dans cette section, nous abordons la dernière partie de notre démarche d'entreposage des données ouvertes tabulaires. Nous rappelons que nous avons transformé, grâce aux deux premières étapes automatiques de notre démarche, différents tableaux statistiques hétérogènes et sans schémas en un graphe intégré comportant des hiérarchies strictes.

Afin de pouvoir interroger les données intégrées par des opérations OLAP, nous proposons un processus progressif pour la conception de schéma multidimensionnel à partir du

graphe intégré. Ce schéma multidimensionnel peut être matérialisé ou non-matérialisé en fonction du choix de l'utilisateur. Par conséquence, notre processus comporte deux vues : (1) une vue utilisateur et (2) une vue système.

- Dans la vue utilisateur, nous avons fait le choix d'une représentation conceptuelle des données afin de rendre transparent à l'utilisateur la matérialisation ou la non-matérialisation. En d'autres termes, l'utilisateur définit une seule fois le schéma multidimensionnel qu'il soit matérialisé ou non. L'utilisateur va définir de façon progressive les composants multidimensionnels en partant d'un graphe intégré. Progressivement ce graphe intégré se transforme en un schéma multidimensionnel selon le formalisme graphique du modèle conceptuel proposé par [Ravat *et al.*, 2001]. Notre démarche de modélisation conceptuelle est hybride étant donné que les besoins de l'utilisateur seront pris en compte implicitement lors de la manipulation des sources de données (le graphe intégré).
- Dans la vue système
  - Si l'utilisateur a choisi de matérialiser les données alors le système se charge de transformer le schéma multidimensionnel selon l'approche R-OLAP classique des entrepôts de données.
  - Si l'utilisateur a choisi de non-matérialiser les données alors le système utilise des mécanismes issus du web sémantique pour produire un graphe RDF annoté. Plus précisément, le système utilise le vocabulaire QB4OLAP [Etcheverry *et al.*, 2014] pour annoter [Cabanac *et al.*, 2007] le graphe RDF intégré équivalent au graphe intégré manipulé par l'utilisateur.

Cette section comporte trois sous-sections. Dans la première sous-section, nous présentons les préliminaires sur lesquels s'appuie notre processus. Dans la deuxième sous-section, nous présentons une description globale de notre processus. Dans la troisième sous-section, nous détaillons les différentes étapes de notre processus.

3.1 Préliminaires

L'objectif de cette section est de présenter les préliminaires sur lesquels s'appuie notre processus de conception ainsi que les correspondances qui existent entre ces préliminaires.

Notre processus utilise :

- le modèle conceptuel proposé par [Ravat *et al.*, 2001] dans la vue utilisateur. En effet, le formalisme graphique de ce modèle est utilisé pour transformer le graphe intégré en un schéma multidimensionnel.
- le vocabulaire QB4OLAP proposé par [Etcheverry *et al.*, 2014] dans la vue système. En effet, les annotations de ce vocabulaire sont utilisées pour augmenter le graphe RDF intégré avec des descriptions multidimensionnelles.

3.1.1 Un modèle conceptuel de données multidimensionnelles

Depuis les années 2000, une expertise a été développée dans notre équipe autour de problèmes variés sur les bases de données multidimensionnelles. [Ravat *et al.*, 2001] [Teste, 2009] a proposé un modèle conceptuel générique et son formalisme graphique

pour représenter une base de données multidimensionnelles sous la forme d'une constellation [Kimball, 1996]. Autour de ce modèle ont gravité plusieurs propositions notamment l'algèbre OLAP [Ravat *et al.*, 2008] et le prototype Graphic-OLAP avec un langage assertionnel [Annoni, 2003] [Ravat *et al.*, 2002] et un langage graphique [Tournier, 2004] [Ravat *et al.*, 2007b] pour la manipulation, l'interrogation et l'analyse visuelle de données multidimensionnelles. La transformation des graphes intégrés issus de données ouvertes tabulaires dans le modèle conceptuel proposé par notre équipe permet de profiter du socle existant pour boucler une chaîne multidimensionnelle complète de l'acquisition des sources à l'interrogation et à la visualisation des données.

Le modèle conceptuel présentant une constellation $\mathscr{C}$ comporte les concepts suivants :
- $\mathscr{F}$ l'ensemble des faits,
- $\mathscr{M}$ l'ensemble des mesures,
- $\mathscr{D}$ l'ensemble des dimensions,
- $\mathscr{H}$ l'ensemble des hiérarchies,
- $\mathscr{A}$ l'ensemble des attributs des dimensions, $\mathscr{A} = \mathscr{P} \cup \mathscr{W}$,
- $\mathscr{P}$ l'ensemble des paramètres,
- $\mathscr{W}$ l'ensemble des attributs faibles.

Définition 2. *Une constellation $\mathscr{C}$ est définie par $(\mathscr{F}; \mathscr{D}; Star)$ tel que :*
- *$\mathscr{F} = \{F_1, \ldots, F_n\}$ est un ensemble fini de faits,*
- *$\mathscr{D} = \{D_1, \ldots, D_m\}$ est un ensemble fini de dimensions,*
- *$Star : \mathscr{F} \rightarrow 2^{\mathscr{D}}$ est une fonction qui associe à chaque fait un ensemble de dimensions.*

Définition 3. *Tout fait $F_i \in \mathscr{F}$ est défini par $(NF_i\,;\, M_i)$ tel que :*
- *NF_i est le nom du fait,*
- *$M_i = \{m_1, \ldots, m_{xi}\} \subseteq \mathscr{M}$ est l'ensemble des mesures associées au fait.*

Définition 4. *Toute dimension $D_i \in \mathscr{D}$ est définie par $(ND_i,\, A_i,\, H_i)$ tel que :*
- *ND_i est le nom de la dimension,*
- *$A_i = P_i \cup W_i \cup \{id_i, All_i\}$ est l'ensemble des attributs de la dimension. Cet ensemble contient des paramètres $P_i \subseteq \mathscr{P}$ et des attributs faibles $W_i \subseteq \mathscr{W}$,*
- *$H_i = \{h_1, \ldots, h_{pi}\} \subseteq \mathscr{H}$ est un ensemble de hiérarchies.*

Définition 5. *Toute hiérarchie $h_j \in H_i$ est définie par $(Nh_j\,;\, P_{h_j}\,;\, \prec h_j\,;\, Weak_{h_j})$ tel que :*
- *Nh_j est le nom de la hiérarchie,*
- *$P_{h_j} = \{p_1, \ldots, p_y\} \subseteq \mathscr{P}$ est l'ensemble des paramètres de la hiérarchie,*
- *$\prec h_j$ est une relation d'ordre sur P_{h_j} telle que :*
  - *l'ordonnancement des paramètres suit un ordre total $\forall p_{k1}, p_{k2} \in P_{h_j}\ k1 \neq k2, p_{k1} \prec h_j p_{k2} \vee p_{k2} \prec h_j p_{k1}$*
  - *il existe un paramètre racine $\forall p_{k1} \in P_{h_j}\ Id_i \prec h_j p_{k1}$*
  - *il existe un paramètre extrémité $\forall p_{k1} \in P_{h_j}\ p_{k1} \prec h_j All_i$*
- *$Weak_{h_j} : Param_{h_j} \rightarrow 2^{\mathscr{W}_{h_j}}$ est une fonction qui associe un ou plusieurs attributs faibles aux paramètres.*

Le formalisme graphique [Golfarelli *et al.*, 1998] [Ravat *et al.*, 2007b] pour représenter ces différents concepts est synthétisé dans les Figures IV.1 et IV.2.

Exemple 10. *La Figure IV.3 illustre le schéma multidimensionnel de l'étude de cas sur les statis-*

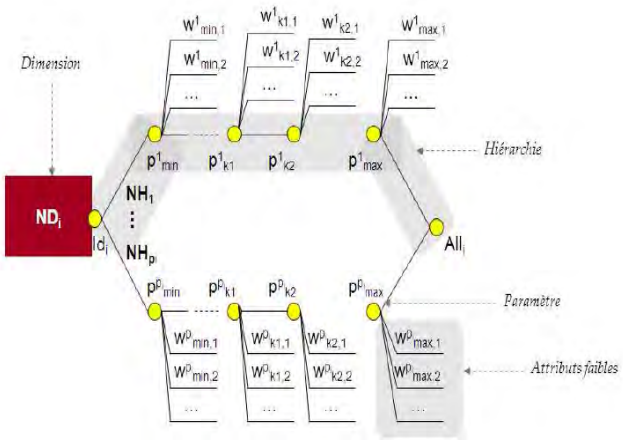


Figure IV.1 — Le formalisme graphique d'une dimension et ses composants

Figure IV.2 — Le formalisme graphique d'un fait et ses composants

tiques des données médicales (cf. chapitre 3). Ce schéma suit le modèle conceptuel décrit ci-dessus ; il contient un fait et deux dimensions.

La constellation est ({Statistiques Médicales}, {Temps, Soin}, {Star(Statistiques Médicales)= {Temps, Soin})
Le fait est (Statistiques Médicales, {SUM(Dépenses), SUM(VolumeSoinConsommé)})
Les dimensions sont (Temps, {Année, All}, {H_Temps}) et (Soins, {Sous-famille de soins, Famille de soins, Sous-catégorie de soins, Catégorie de soin, All}, {H_Soins}).

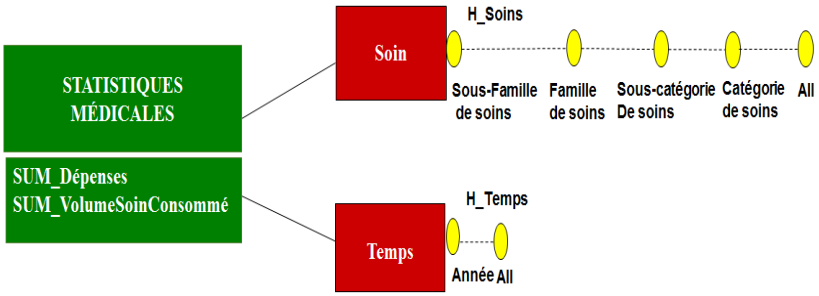


Figure IV.3 — Un exemple de schéma multidimensionnel pour des statistiques médicales

3.1.2 Un vocabulaire d'annotation multidimensionnelle

La représentation des données multidimensionnelles en RDF a fait l'objet de travaux dans la littérature notamment le vocabulaire standard RDF data cube (RDF QB) [Cyganiak et Reynolds., 2012] proposé par le W3C consortium, l'approche de [Kämpgen *et al.*, 2012] et le vocabulaire QB4OLAP [Etcheverry *et al.*, 2014]. Parmi ces approches, le vocabulaire QB4OLAP, qui étend le vocabulaire RDF QB, est le plus complet puisqu'il couvre les concepts et les instances de modèle de données multidimensionnelles en particulier le modèle conceptuel que nous avons décrit dans la section précédente. Le schéma du vocabulaire QB4OLAP est illustré par la Figure IV.4. Nous soulignons que dans ce dernier, les instances des dimensions doivent être des concepts SKOS [1] ce qui est le cas pour les données structurelles du graphe intégré (cf. chapitre 3).

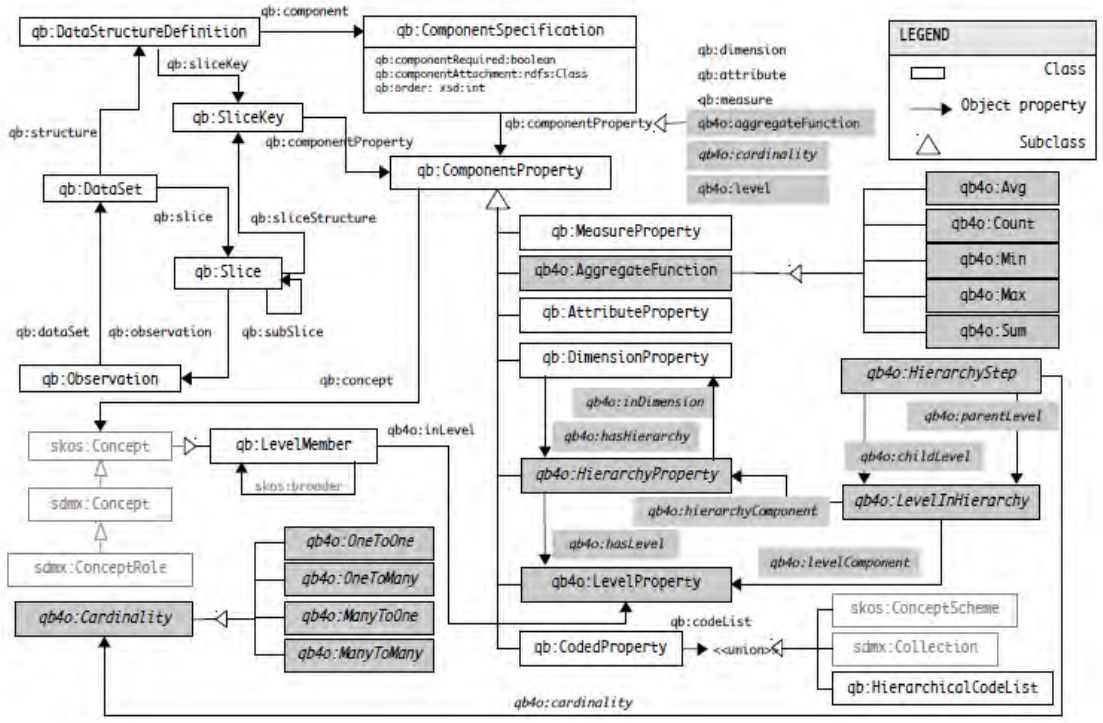


Figure IV.4 — Le vocabulaire QB4OLAP

Les correspondances entre les composants du modèle conceptuel et le vocabulaire QB4OLAP sont résumées dans le tableau IV.1

Exemple 11. *La représentation du schéma multidimensionnel (en simplifiant les URI par les labels des composants) de la Figure IV.3 en QB4OLAP est la suivante :*

– *Les dimensions :*
Temps a qb : dimensionProperty
Soin a qb : dimensionProperty

1. http ://www.w3.org/2004/02/skos/

Tableau IV.1 — Les correspondances entre les composants du modèle conceptuel et le vocabulaire QB4OLAP

Une constellation $\mathscr{C}$	un $qb : dataStructureDefinition$
Un fait de $\mathscr{F}$	un $qb : dataset$
Une mesure de $\mathscr{M}$	une $qb : MeasureProperty$
Une dimension de $\mathscr{D}$	une $qb : DimensionProperty$
Une hiérarchie de $\mathscr{H}$	une $qb4o : HierarchyProperty$
Un paramètre de $\mathscr{P}$	un $qb4o : levelProperty$
Un attribut faible de $\mathscr{W}$	un $qb : AttributeProperty$

– *Les hiérarchies :*
$H\_Temps \ \ a \ \ qb4o : hierarchyProperty \ \ \ qb4o : hasAllLevel"true"$
$H\_Soin \ \ a \ \ qb4o : hierarchyProperty \ \ \ qb4o : hasAllLevel"true"$

– *Les paramètres :*
$SousFamilleSoin \ \ a \ \ qb4o : LevelProperty$
$FamilleSoin \ \ a \ \ qb4o : LevelProperty$
$SousCategorieSoin \ \ a \ \ qb4o : LevelProperty$
$CategorieSoin \ \ a \ \ qb4o : LevelProperty$
$Annee \ \ a \ \ qb4o : LevelProperty$

– *Les relations entre dimensions, hiérarchies et paramètres :*
$Temps \ \ qb4o : hasHierarchy \ \ H\_Temps$
$Soin \ \ qb4o : hasHierarchy \ \ H\_Soin$
$H\_Temps \ \ qb4o : inDimension \ \ Temps$
$H\_Soin \ \ qb4o : inDimension \ \ Soin$
$H\_Temps \ \ qb4o : hasLevel \ \ Annee$
$H\_Soin \ \ qb4o : hasLevel \ \ SousFamilleSoin; FamilleSoin; SousCategorieSoin; CategorieSoin$

– *Le fait et les mesures :*
$StatistiquesMedicales \ \ a \ \ qb : dataset \ Depenses \ \ a \ \ qb : MeasureProperty$
$VolumeSoinConsomm \ \ a \ \ qb : MeasureProperty$

– *La constellation :*
$ConstelStatMed \ a \ qb : DataStructureDefinition ; qb : component \ Temps ; qb : component \ Soins ; qb : component \ [VolumeSoinConsomme \ qb4o : aggregateFunction \ qb4o : sum] ; qb : component \ [Dpenses \ qb4o : aggregateFunction \ qb4o : sum]$

3.2 Description globale du processus de conception

Notre processus de conception, illustrée par la Figure IV.5, a pour objectif de transformer le graphe intégré des données ouvertes tabulaires en des données multidimensionnelles qui peuvent être interrogées par les opérations OLAP. Avant de commencer le processus de conception, l'utilisateur doit choisir s'il veut un résultat matérialisé ou non-matérialisé des

données multidimensionnelles. Indépendamment de son choix, l'utilisateur opère, à un niveau conceptuel, la même démarche de conception. Quant au système, il utilise le choix de l'utilisateur pour effectuer deux traitements différents afin d'aboutir à des données multidimensionnelles matérialisées ou non-matérialisées. Pour cela, notre processus dispose de deux vues : une vue utilisateur et une vue système, comme le montre la Figure IV.5.

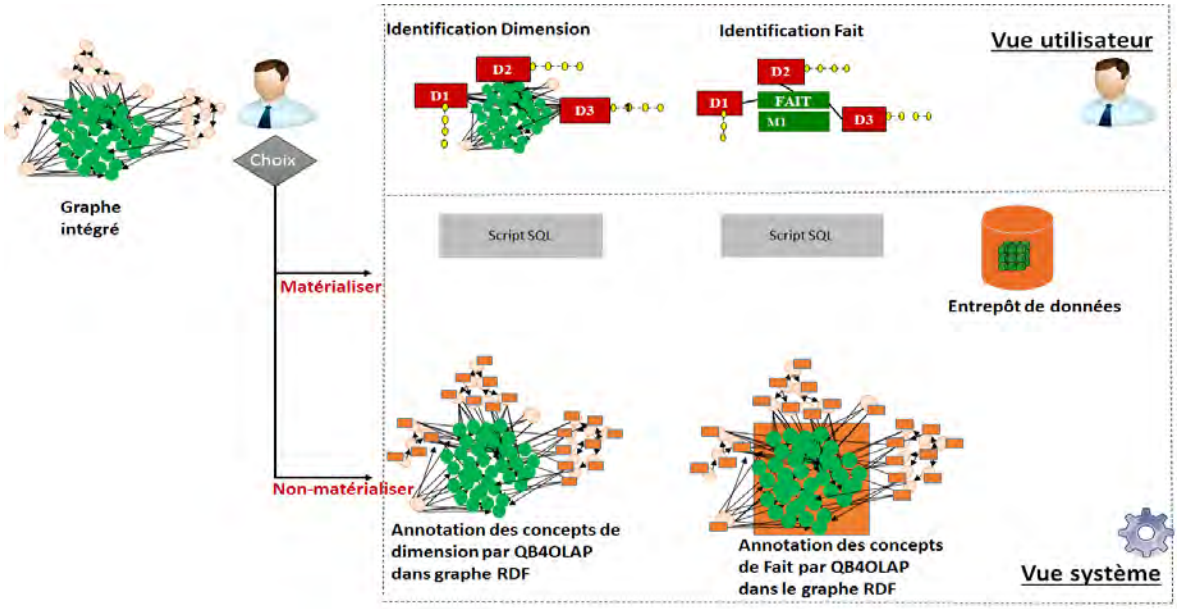


Figure IV.5 — Description globale du processus de conception multidimensionnelle

Dans la vue utilisateur, l'utilisateur a deux étapes à effectuer : (1) identification des dimensions et leurs composants et (2) identification des faits et leurs composants. A ce niveau conceptuel, nous nous appuyons sur le modèle conceptuel proposé par [Ravat *et al.*, 2001] pour transformer progressivement le graphe intégré en un schéma multidimensionnel conforme aux formalismes graphiques du modèle de [Ravat *et al.*, 2001].

Dans la vue système, le système suit les étapes effectuées par l'utilisateur et effectue le traitement approprié pour matérialiser ou non-matérialiser les résultats :

- Si l'utilisateur a choisi de matérialiser les résultats, le système suit la démarche classique R-OLAP pour générer les scripts sql de matérialisation du schéma et de ses instances dans un entrepôt de données R-OLAP.
- Si l'utilisateur a choisi de non-matérialiser les résultats, le système augmente le graphe RDF intégré (équivalent au graphe intégré manipulé visuellement par l'utilisateur) par les annotations multidimensionnelles du vocabulaire QB4OLAP [Etcheverry *et al.*, 2014].

Contrairement aux approches de la littérature, notre proposition commence par la conception des dimensions puis des faits. Ce choix est motivé par deux raisons : (1) faciliter aux concepteurs le repérage visuel des composants multidimensionnels, en effet en simplifiant le visuel des données numériques (par exemple en emboitant ces noeuds) il est plus simple de repérer des arbres que de repérer des mesures ; une fois que les concepts structurels sont simplifiés et remplacés par les formalismes graphiques des dimensions, les noms des mesures sont plus facilement repérables dans le graphe, (2) appliquer l'approche de

[Hassan, 2014] pour la définition des fonctions d'agrégation et les règles de [Prat *et al.*, 2012] pour la vérification de l'additivité du type des fonctions d'agrégation exigent que les dimensions et les hiérarchies des dimensions soient définies.

Dans les sections suivantes, nous détaillons notre processus de conception.

3.3 Description détaillée du processus de conception

3.3.1 Identification des dimensions

Vue utilisateur

Dans le graphe intégré, les données structurelles organisées en arborescence constituent les composants potentiels des dimensions. Certains concepts peuvent représenter les noms de dimensions ou d'attributs (paramètres ou attributs faibles) ; d'autres concepts peuvent constituer les instances d'un attribut. Le concepteur doit identifier dans l'ordre : (1) le nom de la dimension, (2) les attributs de la dimension (nous nous focalisons particulièrement sur les paramètres) et (3) les hiérarchies.

Afin d'assister l'utilisateur dans la tâche d'identification des dimensions, nous pré-calculons automatiquement les potentielles dimensions dans le graphe intégré comme suit :

1. Nous cherchons dans le graphe tous les arbres dont les racines sont des données structurelles et dont les feuilles sont des données structurelles reliées à des données numériques.
2. Nous vérifions si chaque arbre est ontologique et complet sinon nous le complétons par des noeuds "Autre".
3. Nous comptons la hauteur h de chaque arbre. Pour chaque arbre, nous générons une dimension avec h paramètres dont les instances sont les noeuds de chaque niveau dans l'arbre.

A droite de la Figure IV.6, nous avons un exemple de dimension pré-calculée par le système ; à gauche dans la même figure nous avons une dimension proposée par le concepteur. Nous pouvons constater que la compréhension de la sémantique des données structurelles nécessite des compétences et qu'il nous apparaît difficile à cette phase d'automatiser la tâche. Notamment, certains concepts peuvent être des noms de composants de la dimension, ce qui change la structure de la dimension pré-calculée automatiquement. Nous avons alors choisi d'utiliser les résultats de pré-calcul pour assister le concepteur dans la tâche d'identification de la dimension. Cette tâche comporte trois étapes : (1) identification du nom de la dimension, (2) identification des paramètres, (3) identification des hiérarchies. Le concepteur doit itérer ces étapes pour chaque dimension.

Vue système

Le système réagit progressivement avec les trois sous-étapes effectuées par l'utilisateur : (1) identification du nom de la dimension, (2) identification des attributs de la dimension et (3) identification des hiérarchies. Si le choix est de matérialiser alors le système crée progressivement les scripts de création du schéma et d'alimentation de la table de dimension. Si le choix est de non-matérialiser alors le système enrichit progressivement le graphe RDF intégré par les annotations correspondantes à chaque sous-étape.

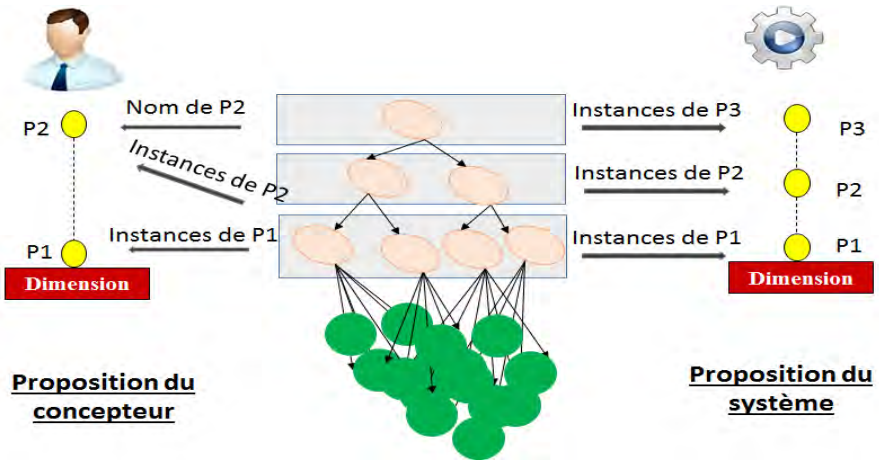


Figure IV.6 — La différence entre une dimension pré-calculée par le système et une dimension identifiée par le concepteur

3.3.1.1 Identification du nom de la dimension

Vue utilisateur

Le concepteur commence par identifier le nom de la dimension. Pour cela, soit il choisit le nom d'un noeud structurel dans le graphe, soit il rentre manuellement le nom de la dimension.

- Si le nom de la dimension est le nom d'un noeud structurel ce noeud est transformé dans le formalise graphique d'une dimension.
- Si le nom de la dimension est rentré manuellement alors un nouveau noeud est créé selon le formalisme graphique d'une dimension.

Vue système

- Si l'utilisateur a choisi le nom de la dimension à partir d'un noeud structurel dans le graphe alors :
  - Si le choix est de matérialiser alors le système prépare une requête de création de table avec le nom de la dimension.
  - Si le choix est de non-matérialiser alors le système ajoute l'annotation *a qb* : *dimensionProperty* pour le noeud structurel dans le graphe RDF intégré équivalent au graphe intégré manipulé par l'utilisateur.
- Si l'utilisateur a rentré manuellement le nom de la dimension alors :
  - Pour matérialiser, le système prépare une requête de création de table avec le nom de la dimension rentré manuellement.
  - Pour non-matérialiser, le système crée un nouveau noeud structurel qui porte le nom de la dimension et qui est une instance de *qb* : *dimensionProperty*.

Exemple 12. *Dans notre exemple, le concepteur identifie manuellement le nom de la dimension "Soin". Un nouveau noeud "Soin" est ajouté selon le formalisme de dimension dans le graphe intégré, comme le montre la Figure IV.7.*

Afin de matérialiser, le système génère ce script sql du schéma de la table Soin :
CREATE TABLE Soin (

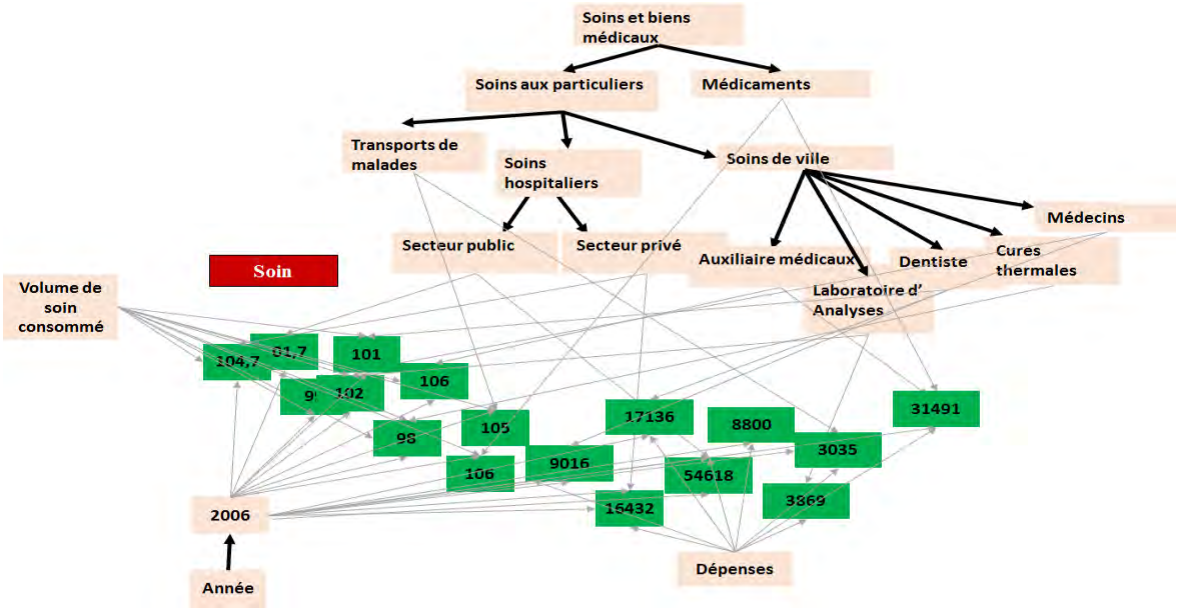


Figure IV.7 — Ajout d'un noeud de dimension par **MDGen**

id_dim_soin DECIMAL,
CONSTRAINT pk_soin PRIMARY KEY (id_dim_soin))

Afin de non-matérialiser, le système génère l'annotation suivante dans le graphe RDF pour le noeud Soin :
"Soin a qb : dimensionProperty"

3.3.1.2 Identification des paramètres de la dimension

Vue utilisateur

La deuxième sous-étape est l'identification des paramètres de la dimension. Le concepteur peut sélectionner le nom du paramètre d'un noeud structurel ou le rentrer manuellement puis il sélectionne dans le graphe les noeuds instances de ce paramètre.

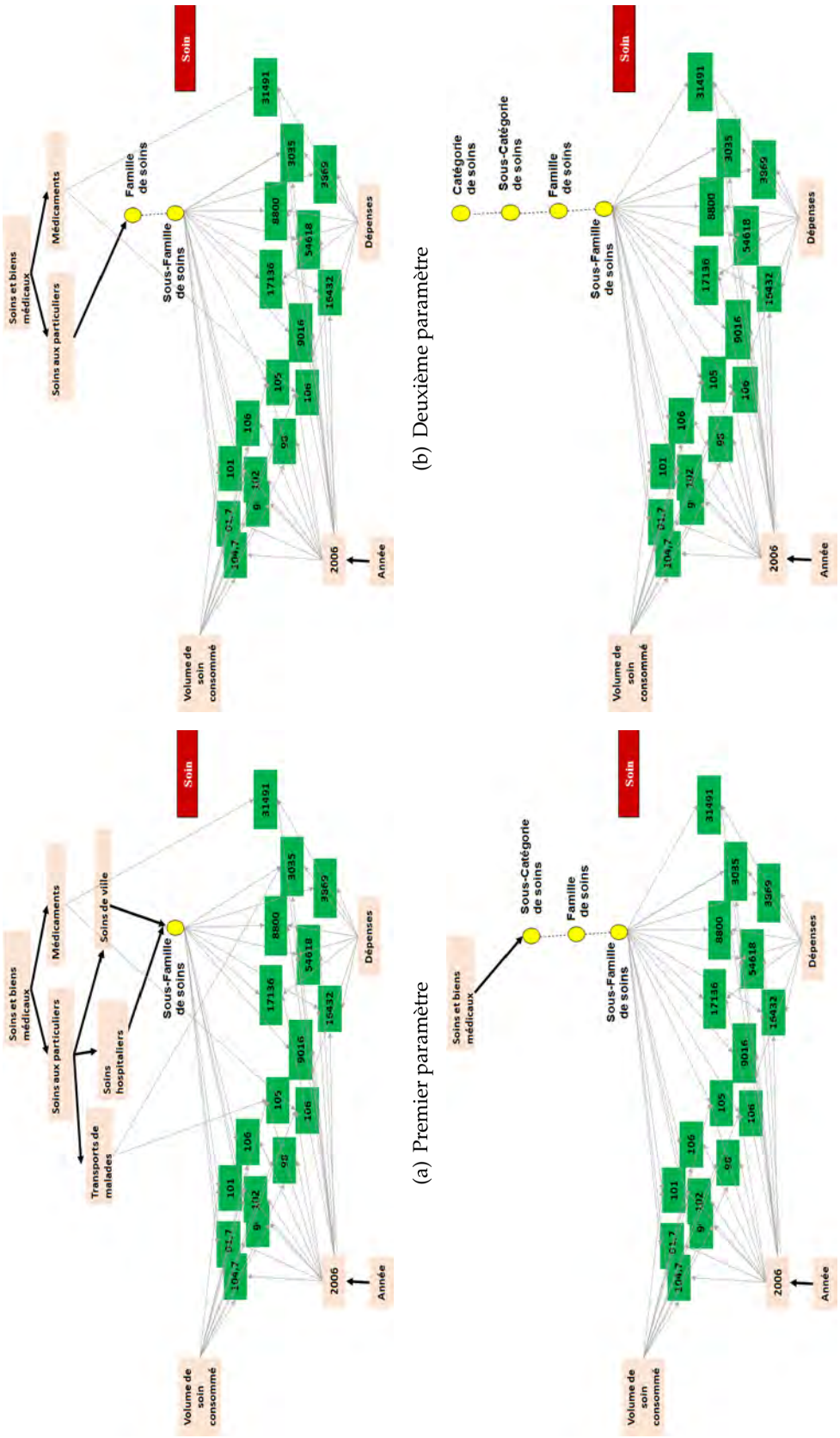
- Si le nom du paramètre est sélectionné à partir du nom d'un noeud structurel alors le concepteur est assisté par une proposition automatique. En effet, nous proposons à l'utilisateur les instances potentielles de ce paramètre qui sont les noeuds qui le succèdent. Le noeud du paramètre est transformé selon le formalise graphique d'un paramètre. Les noeuds des instances sont éliminés et les arcs sortants des instances sont reportés vers le noeud du paramètre.
- Si le nom du paramètre est rentré manuellement alors le concepteur doit sélectionner les noeuds des instances dans le graphe. Un nouveau noeud est créé selon le formalisme graphique d'un paramètre. Les noeuds des instances sont éliminés et les arcs sortants des instances sont reportés vers le nouveau noeud du paramètre.

Vue système

- Si le nom du paramètre est sélectionné à partir du nom d'un noeud structurel alors :

  - Pour matérialiser, le système ajoute le nom du paramètre dans la requête de création de la table de dimension. Il prépare également la liste des instances indexées par le nom du paramètre.
  - Pour non-matérialiser, le système ajoute dans le graphe RDF l'annotation *aqb4o : LevelProperty* pour le noeud structurel du paramètre, les annotations *aqb : levelMember* pour les noeuds structurels instances du paramètre et la relation *qb4o : inLevel* entre les instances et le paramètre.
- Si le nom du paramètre est rentré manuellement
  - Pour matérialiser, le système ajoute le nom du paramètre dans la requête de création de la table de dimension.
  - Pour non-matérialiser, le système crée un nouveau noeud structurel de label le nom du paramètre et instance de *qb : LevelProperty*. Il ajoute les annotations *a qb : levelMember* pour les noeuds instances du paramètre. Il ajoute également la relation *qb4o : inLevel* entre les instances et le paramètre.

Exemple 13. *Nous illustrons dans les Figures IV.8(a), IV.8(b), IV.8(c) et IV.8(d), l'évolution [Ravat* et al.*, 2006b] du graphe intégré à chaque fois que le concepteur identifie un nouveau paramètre pour la dimension "Soin". Tant que l'utilisateur n'a pas encore défini les hiérarchies ces paramètres ne sont pas encore reliés à la dimension.*



(a) Premier paramètre

(b) Deuxième paramètre

(c) Troisième paramètre

(d) Quatrième paramètre

Figure IV.8 — Exemple d'identification des paramètres

Afin de matérialiser, le système utilise les liens de précédences, qui ont été simplifiés après l'identification des paramètres, pour la construction des instances de la table de dimension qui sera produite après identification des hiérarchies.

Afin de non-matérialiser, le système :

- *génère les annotations suivantes pour les paramètres :*
  SousFamilleSoin a qb4o : LevelProperty
  FamilleSoin a qb4o : LevelProperty
  SousCategorieSoin a qb4o : LevelProperty
  CategorieSoin a qb4o : LevelProperty

- *génère les annotations suivantes pour les instances des paramètres :*
  secteurPublic a qb : LevelMember
  secteurPrive a qb : LevelMember
  AuxiliairesMedicaux a qb : LevelMember
  LaboratoireDanalyse a qb : LevelMember
  Dentiste a qb : LevelMember
  CureThermales a qb : LevelMember
  Medecins a qb : LevelMember
  SoinsHospitaliers a qb : LevelMember
  SoinsVille a qb : LevelMember
  TransportsMalades a qb : LevelMember
  SoinsAuxParticuliers a qb : LevelMember
  Medicaments a qb : LevelMember
  SoinsEtBiensMedicaux a qb : LevelMember

- *génère les annotations suivantes pour les relations entre instances et paramètres :*
  secteurPublic qb4o : inLevel SousFamilleSoin
  secteurPrive qb4o : inLevel SousFamilleSoin
  AuxiliairesMedicaux qb4o : inLevel SousFamilleSoin
  LaboratoireDanalyse qb4o : inLevel SousFamilleSoin
  Dentiste qb4o : inLevel SousFamilleSoin
  CureThermales qb4o : inLevel SousFamilleSoin
  Medecins qb4o : inLevel SousFamilleSoin
  SoinsHospitaliers qb4o : inLevel FamilleSoin
  SoinsVille qb4o : inLevel FamilleSoin TransportsMalades qb4o : inLevel SousCatgorieSoin SoinsAuxParticuliers qb4o : inLevel SousCatgorieSoin Medicaments qb4o : inLevel SousCatgorieSoin SoinsEtBiensMedicaux qb4o : inLevel CatgorieSoin

3.3.1.3 Identification des hiérarchies de la dimension

Vue utilisateur

Le concepteur doit donner un nom à chaque hiérarchie, choisir les paramètres et leurs ordres. Le système assiste le concepteur dans le choix de l'ordre des paramètres en lui indi-

quant que le niveau le plus bas doit être relié aux données numériques. De plus, au cours de l'étape d'identification des paramètres, le système a relié les paramètres qui ont un lien de précédence entre leurs instances ce qui indique aussi quel ordre doit être établi. Dans le graphe intégré un label de la hiérarchie est rajouté. Le lien entre les paramètres de la hiérarchie et la dimension est également établi.

Vue système

- Pour matérialiser, le système met à jour la requête de création de la table de dimension avec les paramètres qui ont été sélectionnés dans les hiérarchies. A ce stade, le système résout le problème des hiérarchies non-ontologies et non-couvrantes pour les instances de la dimension. Afin d'optimiser le calcul, nous pouvons vérifier si la dimension courante a été détectée et corrigé.
- Pour non-matérialiser, le système génère les annotations pour les hiérarchies, les relations entre hiérarchies-paramètres et les relations entre hiérarchies-dimensions.

Exemple 14. *Après identification de la hiérarchie H_Soin, nous pouvons constater qu'une nouvelle dimension et ses composants apparaissent dans le graphe intégré d'après la Figure IV.9.*

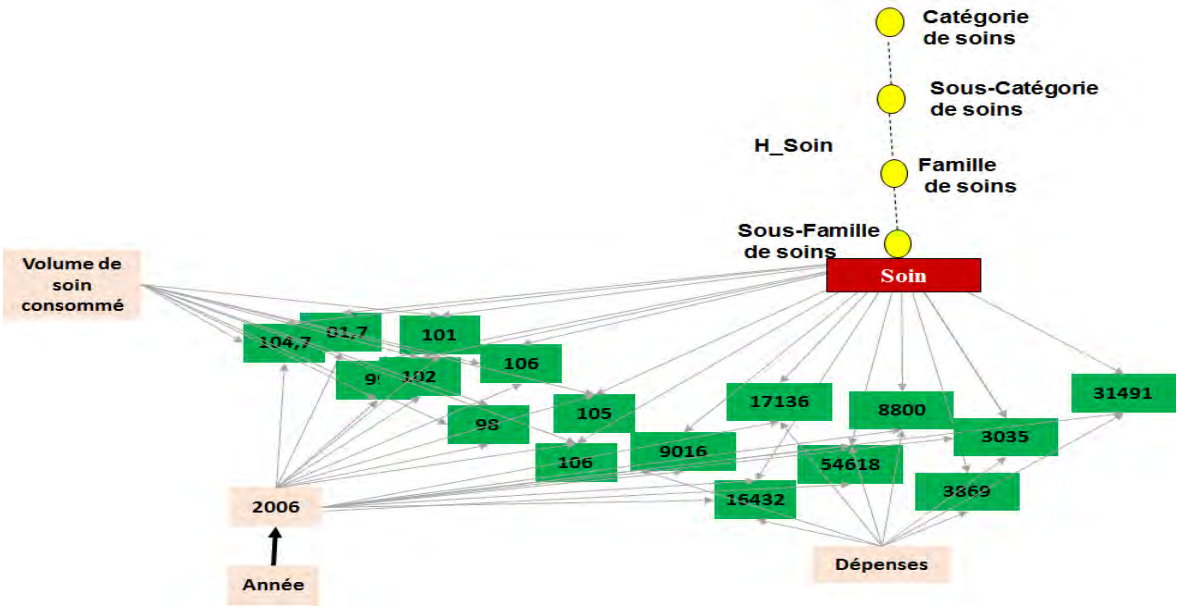


Figure IV.9 — Exemple d'identification d'une hiérarchie

Afin de matérialiser, le système génère la requête de création du schéma de la table de dimension comme suit :
CREATE TABLE Soin (
id_dim_soin DECIMAL,
Sous_Famille_Soin VARCHAR(50),
Famille_Soin VARCHAR(50),
Sous_Categorie_Soin VARCHAR(50),
Categorie_Soin VARCHAR(50),
CONSTRAINT pk_soin PRIMARY KEY (id_dim_soin))

Le système produit aussi une requête d'alimentation (insertion) de la table de dimension dont les résultats sont illustrés dans la Figure IV.10

Id	Sous_Famille_Soin	Famille_Soin	Sous_Catégorie_Soin	Catégorie_Soin
1	Secteur public	Soins hospitaliers	Soins aux particuliers	Soins et biens médicaux
2	Secteur privé	Soins hospitaliers	Soins aux particuliers	Soins et biens médicaux
3	Auxiliaires médicaux	Soins de ville	Soins aux particuliers	Soins et biens médicaux
4	Laboratoire d'analyse	Soins de ville	Soins aux particuliers	Soins et biens médicaux
5	Auxiliaires médicaux	Soins de ville	Soins aux particuliers	Soins et biens médicaux
6	Dentiste	Soins de ville	Soins aux particuliers	Soins et biens médicaux
7	Cures thermales	Soins de ville	Soins aux particuliers	Soins et biens médicaux
8	Médecins	Soins de ville	Soins aux particuliers	Soins et biens médicaux
9	Autre_1	Transports de malades	Soins aux particuliers	Soins et biens médicaux
10	Autre_2	Autre	Médicaments	Soins et biens médicaux

Figure IV.10 — Les instances de la table de dimension

Afin de non-matérialiser, le système génère les annotations suivantes :
$H\_Soin\ \ a\ \ qb4o : hierarchyProperty\ \ qb4o : hasAllLevel"true"$
$H\_Soin\ \ qb4o : inDimension\ \ Soin$
$Soin\ \ qb4o : hasHierarchy\ \ H\_Soin$
$H\_Soin\ \ qb4o : hasLevel\ \ SousFamilleSoin\ \ ; FamilleSoin\ \ ; SousCatgorieSoin\ \ ; CatgorieSoin$

3.4 Identification des faits

Vue utilisateur

Après avoir défini toutes les dimensions, le concepteur procède à la définition du Fait et ses composants. D'abord, il doit préciser le nom du Fait et sélectionner les dimensions liées. Le système vérifie les données numériques qui sont reliées au niveau le plus bas d'au moins une dimension (nous rappelons la possibilité d'avoir des données numériques manquantes issues du croisement des différentes sources d'où l'absence de données numériques reliées à des niveaux de base de certaines dimensions) et propose les données structurelles, non-utilisées dans l'identification des dimensions, comme nom de mesure. Pour chaque dimension et pour chaque mesure, le concepteur peut définir pour chaque paramètre de chaque hiérarchie différentes fonctions d'agrégation selon la proposition de [Hassan, 2014]. Au niveau du graphe, les noeuds des données numériques sont supprimés et le nouveau noeud de Fait avec ses mesures est créé. Le noeud Fait est également relié aux dimensions selon le formalisme graphique du modèle conceptuel de [Teste, 2009].

Vue système

– Pour matérialiser, le système prépare la requête de création du schéma de la table de

Fait et la requête pour son alimentation. A ce stade, le système gère automatiquement le problème des données numériques redondantes pour les mêmes instances de dimensions en utilisant la technique de vote. Par ailleurs, les données numériques manquantes de certaines instances de dimensions seront des valeurs nulles dans la table de Fait.

– Pour non-matérialiser, le système génère un nouveau noeud structurel avec le label du Fait et lui attribue l'annotation *a qb : dataset*. Il rajoute l'annotation *a qb : MeasureProperty* pour les données structurelles identifiées comme mesures. Chaque fonction d'agrégation peut avoir les annotations *qb4o : Avg*, *qb4o : Min*, *qb4o : Max*, *qb4o : Count*, *qb4o : Count* ou *qb4o : sum*. Une fonction d'agrégation doit se définir comme une propriété entre un composant de la dimension (paramètre, attribut ou hiérarchie) et entre la mesure. Enfin, un nouveau noeud pour la constellation est créé avec l'annotation *a qb : dataStructureDefintion*. Le lien entre le Fait et la constellation est établi par l'annotation *Nom_Fait qb : structure Nom_Constellation*. Pour chaque dimension reliée au Fait, le système rajoute l'annotation *Nom_Constellation qb : Component Nom_Dimension*. Comme nous considérons que les concepteurs ne sont pas forcément des experts dans la conception multidimensionnelle, le système doit au préalable implémenter les règles de vérification d'additivité (sous forme d'assertions OWL-DL) proposées par [Prat *et al.*, 2012] .

Exemple 15. *La Figure IV.11 montre une itération intermédiaire dans la création de Fait pour l'exemple de motivation.*

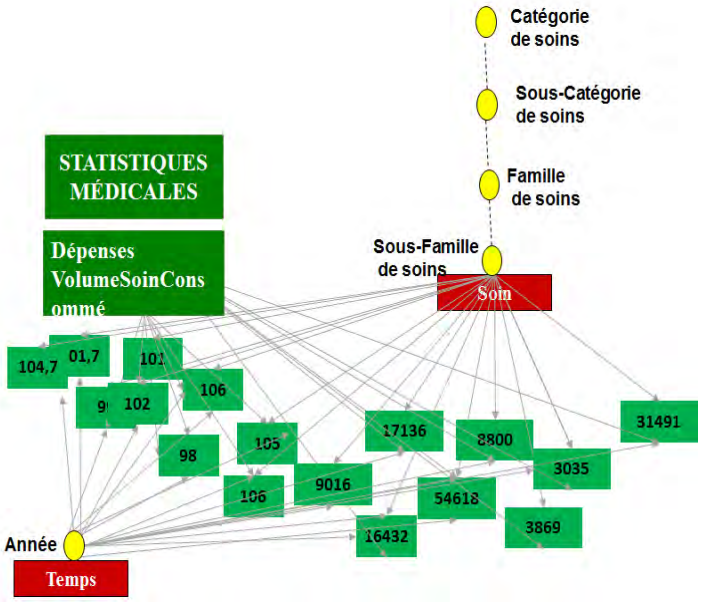


Figure IV.11 — Exemple d'itération dans la création de Fait

Afin de matérialiser, le système génère la requête de création du schéma de la table de fait comme suit :
CREATE TABLE StatistiquesMedicales (
id_fact_stats DECIMAL,
id_dim_soin DECIMAL,

id_dim_temps DECIMAL,
Dépenses DECIMAL,
VolumeSoinConsommé DECIMAL,
CONSTRAINT pk_statMed PRIMARY KEY (id_fact_stats),
CONSTRAINT fk_soin FOREIGN KEY (id_dim_soin) REFERENCES Soin(id_dim_soin),
CONSTRAINT fk_temps FOREIGN KEY (id_dim_temps) REFERENCES Temps(id_dim_temps),
)

Le système produit aussi une requête d'alimentation de la table de fait dont le résultat est illustré dans la Figure IV.12.

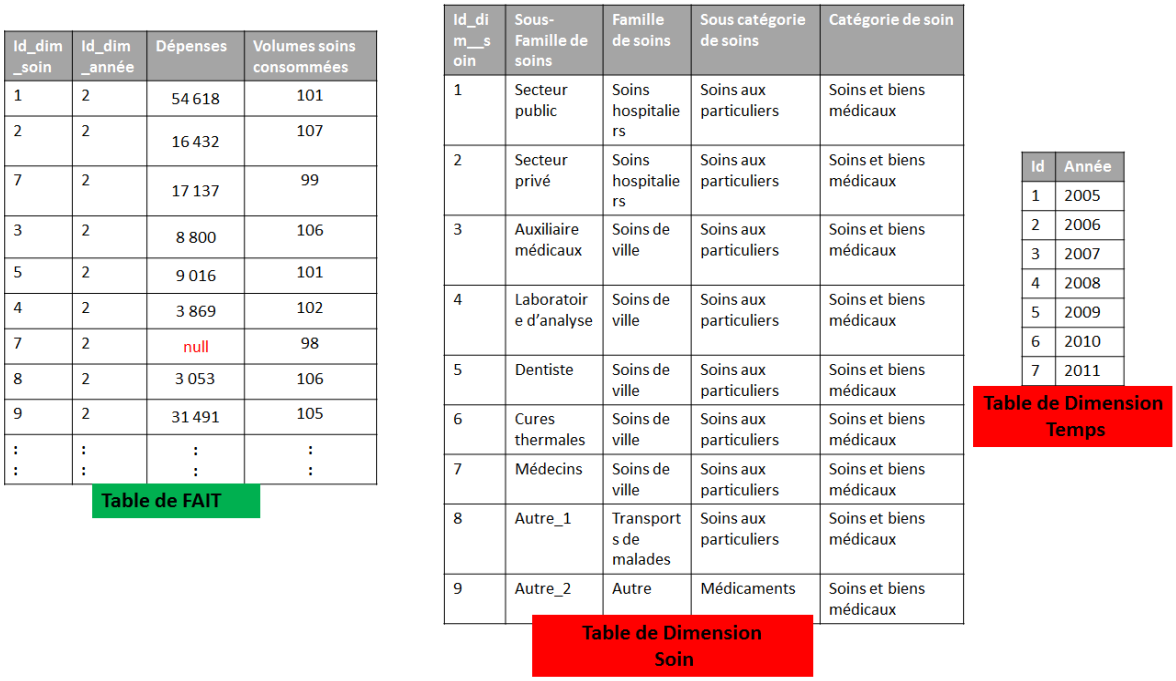


Figure IV.12 — Exemple de table de Fait avec les tables de dimensions

Afin de non-matérialiser, le système produit les annotations suivantes dans le graphe RDF.

StatistiquesMedicales a qb : dataset Depenses a qb : MeasureProperty
VolumeSoinConsomme a qb : MeasureProperty
ConstelStatMed a qb : DataStructureDefinition
qb : component Temps
qb : component Soins
StatistiquesMedicales qb : structure ConstelStatMed

4 Conclusion

Nous avons présenté dans ce chapitre la dernière étape de notre démarche d'entreposage d'Open Data. Nous rappelons que les deux premières étapes de notre démarche ont permis de transformer les données tabulaires en graphes puis les intégrer. L'objectif de cette

étape est de permettre l'exploitation analytique de ces données intégrées. L'exploitation analytique dans les systèmes décisionnels est réalisée par des opérations OLAP. Ces opérations nécessitent l'organisation des données selon un schéma multidimensionnel. Pour cela, cette dernière étape est consacrée à la transformation des données ouvertes intégrées en données multidimensionnelles (c'est-à-dire décrites par un schéma multidimensionnel). Dans la littérature, il y a deux approches pour la matérialisation des données multidimensionnelles. Certaines approches conçoivent un schéma multidimensionnel puis matérialisent les données dans un entrepôt. D'autres approches décrivent les données par des vocabulaires multidimensionnels et les interrogent directement sans matérialisation. Étant donné que notre approche s'adresse à une large audience (contexte self-service BI), nous supportons les deux approches. Pour cela, nous avons proposé un processus progressif composé de deux vues :

- Une vue utilisateur dans laquelle l'utilisateur définit progressivement, à un niveau conceptuel, les composants multidimensionnels (dimensions et faits) à partir du graphe intégré. Ce graphe se transforme progressivement en un schéma multidimensionnel selon le formalisme graphique du modèle conceptuel proposé par [Ravat *et al.*, 2007b].
- Une vue système dans laquelle le système s'occupe de la matérialisation ou de la non-matérialisation des données d'une façon complètement transparente à l'utilisateur. Le système demanderait juste au début du processus le choix de l'utilisateur concernant la matérialisation.
  - Pour matérialiser les données, le système opère pour générer un entrepôt de données selon la démarche R-OLAP classique. Pour cela, le système génère progressivement les scripts sql pour le schéma de création et d'alimentation de l'entrepôt de données. Il est possible alors d'interroger l'entrepôt de données avec des opérations OLAP.
  - Pour non-matérialiser les données, le système produit automatiquement des annotations multidimensionnelles dans un graphe RDF équivalent au graphe intégré visualisé par le concepteur. Les annotations mutlidimensionnelles utilisées appartiennent au vocabulaire QB4OLAP [Etcheverry *et al.*, 2014]. Nous avons choisi ce vocabulaire multidimensionnel puisqu'il est plus complet que d'autres vocabulaires, en instance QB [2]. Il est possible d'interroger directement le graphe RDF annoté avec des opérations OLAP-SPARQL [Etcheverry *et al.*, 2014]. L'usage des graphes génériques nous a permis d'aboutir à une démarche ETQ [Abello *et al.*, 2015].

Ces propositions ont été publiées dans le cadre de la conférence nationale EDA'13 [Berro *et al.*, 2013] et la conférence internationale ICEIS'15 [Berro *et al.*, 2015a].

Dans le chapitre suivant, nous décrivons le prototype implémenté pour la validation de notre démarche. Puis, nous présentons les résultats des expérimentations effectuées sur nos propositions.

2. http ://www.w3.org/TR/vocab-data-cube/

V Prototype et évaluations

NOUS abordons dans ce dernier chapitre deux volets concernant la validation de notre démarche d'entreposage. Dans le premier volet, nous décrivons les différents modules du prototype qui ont été développés pour notre démarche ETL. Dans le deuxième volet, nous illustrons les résultats d'évaluations de nos algorithmes de détection des données tabulaires ainsi que les résultats d'évaluation de la qualité et de la performance du programme linéaire d'intégration de données.

1 Introduction

L'ouverture des données tabulaires statistiques par des organismes publics constitue de nouvelles sources pour alimenter les systèmes de prise de décision. Néanmoins, ces données posent plusieurs problèmes : hétérogénéité, manque de structure, qualité, etc. Elles nécessitent alors des solutions pour les croiser et les analyser de manière simple et rapide.

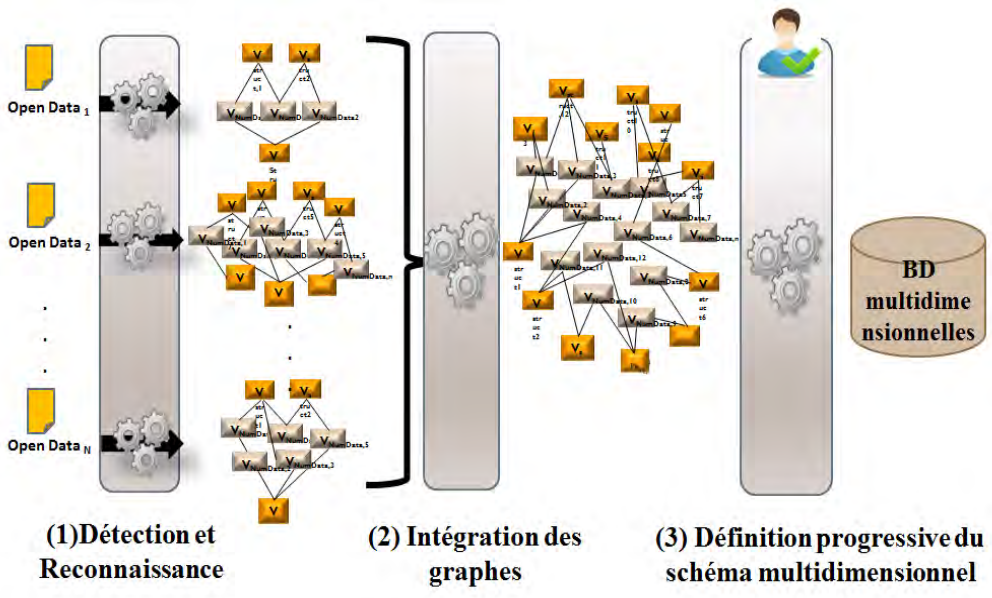


Figure V.1 — Une démarche d'entreposage des données ouvertes tabulaires statistiques

Nous répondons à ces besoins en proposant une démarche ETL basée sur les graphes. Cette démarche est constituée de trois phases, Extraction- Transformation- Loading ; elles sont illustrées dans la Figure V.1. La première phase consiste à la détection et la reconnaissance des données tabulaires. La deuxième phase consiste à l'intégration des graphes de

données ouvertes et la troisième phase consiste à la définition de schémas multidimensionnels à partir du graphe intégré. L'utilisation des graphes dans cette démarche permet une plus grande généricité de nos travaux.

Les phases de cette démarche ont été détaillées dans les chapitres 2, 3 et 4 de ce manuscrit. Le but de ce chapitre est d'illustrer le prototype développé pour valider notre démarche et évaluer celle-ci.

Ce chapitre s'articule en deux sections : prototype et évaluations. Dans la section prototype, nous décrivons les trois modules qui ont été implémentés pour les trois étapes de notre démarche. Nous nous appuyons sur un scénario d'étude pour présenter ces modules. Dans la section évaluations, nous présentons d'abord les résultats du module de détection des données tabulaires puis nous détaillons les résultats d'évaluation de la qualité de l'appariement par paire sur des bancs d'essais de référence. Nous finissons avec les résultats d'évaluation de la performance de l'appariement holistique pour les graphes de notre scénario d'étude.

2 Prototype

Dans cette section, nous décrivons le prototype que nous avons développé pour valider notre approche ETL. Nous présentons un scénario d'étude, dans la sous-section 2.1, pour illustrer le fonctionnement de notre prototype. Ensuite, nous présentons l'architecture fonctionnelle de ce prototype dans la sous-section 2.2. Enfin, nous détaillons les trois modules de notre prototype dans les sous-sections 2.3, 2.4 et 2.5.

2.1 Un scénario d'étude

Nous présentons dans cette section un exemple d'utilisation de données ouvertes.

Department for Environment Food & Rural Affairs

UNITED KINGDOM CEREAL YIELDS
1885 onwards

Tonnes per hectare

Year [a]	Wheat	Barley	Oats	Rye	Mixed Corn	Triticale	Oilseed Rape
1998	7,6	5,3	6,0	4,3	3,4	5,7	2,9
1999	8,1	5,6	5,9	5,6	4,7	6,2	3,2
2000	8,0	5,8	5,9	6,1	5,5	6,1	2,9
2001	7,1	5,3	5,5	4,9	3,9	4,7	2,6
2002	8,0	5,6	6,0	5,8	4,7	4,7	3,4
2003	7,8	5,9	6,2	5,8	4,3	4,1	3,3
2004	7,8	5,8	5,8	5,7	4,3	4,1	2,9
2005	8,0	5,9	5,8	6,7	4,4	4,2	3,2
2006	8,0	5,9	6,0	6,1	4,5	4,3	3,3
2007	7,2	5,7	5,5	5,7	3,9	3,9	3,1
2008	8,3	6,0	5,8	6,1	4,4	4,4	3,3
2009	7,9	5,8	5,8	6,6	4,1	4,1	3,4
2010	7,7	5,7	5,5	6,3	4,1	4,0	3,5
2011	7,7	5,7	5,6	5,4	3,9	4,1	3,9
2012	6,7	5,5	5,1	5,2	4,2	3,5	3,4
2013	7,4	5,8	5,5	5,6	4,2	3,9	3,0

Figure V.2 — Le rendement de la production de céréales par année et par type de céréale au royaume uni.

Supposons qu'une entreprise agricole souhaite lancer une nouvelle production de céréales en Anglettre. Ce projet nécessite une étude de marché afin de prendre une décision sur le type de produit céréalier qui permettra la plus grosse marge. Cette entreprise ne peut pas acquérir les statistiques de ses concurrents mais elle dispose de données ouvertes gouvernementales disponibles gratuitement sur le web et qui peuvent l'aider à analyser l'état du marché.

L'entreprise a procédé à la récolte de données ouvertes sur le web à partir desquelles elle souhaite construire un entrepôt de données. La finalité est d'analyser l'état du marché et de recenser les besoins de l'entreprise par croisement de plusieurs sources de données. Le portail World Bank fournit plus que 1,000 indicateurs statistiques par catégorie, par année et par ville. Ces indicateurs sont disponibles sur le lien `http://data.worldbank.org/indicator`. Dans la catégorie agriculture et développement durable, il est possible de récupérer les données du Royaume Uni relatives au rendement de céréales et à la production agricole. Ces données sont disponibles sous format tabulaire sur le lien `http://data.worldbank.org/country/united-kingdom`.

Les données fournies par le portail World Bank sont agrégées. L'entreprise doit collecter d'avantage d'informations afin d'approfondir son étude et ses analyses. Sur le lien `https://www.gov.uk/government/statistical-data-sets/structure-of-the-agricultural-industry-in-england-and-the-uk-at-june` sont fournies plusieurs données statistiques sur la production de céréale en Royaume uni par type de céréale, par taille de ferme, année, etc. Ces données sont plus détaillées que ceux du World Bank. En multipliant les recherches sur les différents fournisseurs de données ouvertes, il est possible de collecter jusqu'à une cinquantaine de sources qui peuvent contenir un ou plusieurs tableaux.

Nous illustrons dans les Figures V.2, V.3 et V.4 des extraits de tableaux de ce scénario d'étude.

WHEAT

Areas	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
UNITED KINGDOM	1 847	2 086	1 635	1 996	1 836	1 990	1 867	1 833	1 816	2 080
England	1 746	1 957	1 541	1 876	1 726	1 865	1 748	1 709	1 691	1 935
North East	64	76	54	70	66	71	68	66	66	77
North West and Merseyside	23	28	17	27	28	35	31	31	30	36
Yorkshire & The Humber	234	258	199	247	233	251	236	229	226	262
East Midlands	362	401	319	387	352	385	348	346	344	391
West Midlands	150	167	127	163	150	167	157	154	153	178
Eastern	490	545	452	529	482	513	487	475	471	537
South East and London	243	277	207	260	240	251	241	235	229	258
South West	181	204	166	194	174	191	180	174	172	196
Wales	13	15	11	15	14	15	15	16	13	20
Scotland	85	109	80	97	87	101	96	100	103	114
Northern Ireland	3	5	4	7	7	9	8	9	9	12

Yields	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
UNITED KINGDOM	8,0	8,0	7,1	8,0	7,8	7,8	8,0	8,0	7,2	8,3
England	8,1	8,0	7,1	8,0	7,8	7,7	7,9	8,0	7,2	8,3
North East	7,4	7,7	6,8	8,1	8,1	7,7	8,3	8,5	7,7	8,1
North West and Merseyside	6,7	7,1	5,2	6,7	7,1	6,9	5,3	6,2	5,2	6,0
Yorkshire & The Humber	8,2	8,3	7,6	8,5	8,2	7,6	8,1	8,2	7,3	8,4
East Midlands	8,4	8,2	7,0	8,2	7,5	7,7	8,1	8,3	7,1	8,8
West Midlands	7,6	7,5	6,4	7,5	7,7	7,6	7,5	7,3	7,0	7,8

Figure V.3 — La surface et le rendement de la production du blé par région et par année

Notre approche vise à faciliter l'intégration et l'entreposage de ces données à travers le

OATS

Areas	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
UNITED KINGDOM	92	109	112	126	121	108	90	121	129	135
England	63	80	85	98	93	80	66	93	102	107
North East	4	4	5	5	5	5	4	6	6	6
North West and Merseyside	3	3	3	4	4	4	3	4	5	5
Yorkshire & The Humber	4	5	5	6	5	5	4	6	7	8
East Midlands	7	9	10	12	11	10	8	11	12	13
West Midlands	13	15	16	18	17	15	12	17	19	20
Eastern	4	6	6	7	7	7	6	9	10	10
South East and London	15	20	19	24	22	18	14	19	21	22
South West	14	18	20	22	20	18	14	21	21	23
Wales	3	4	3	4	4	4	3	4	4	4
Scotland	23	22	22	22	22	22	20	23	21	22
Northern Ireland	3	3	2	2	2	2	2	2	2	2

Yields	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
UNITED KINGDOM	5,9	5,9	5,5	6,0	6,2	5,8	5,8	6,0	5,5	5,8
England	6,2	6,2	5,6	6,3	6,3	6,0	6,0	6,0	5,4	5,9
North East	6,4	6,3	6,1	6,4	6,9	6,0	6,5	6,3	6,2	6,5
North West and Merseyside	6,7	6,8	5,0	6,1	5,9	4,7	4,8	5,0	5,0	4,8
Yorkshire & The Humber	6,8	4,8	6,3	6,7	6,5	5,9	6,6	6,1	5,4	5,9
East Midlands	6,4	6,4	5,3	6,7	6,2	6,5	6,5	6,0	5,4	5,9
West Midlands	6,1	6,4	5,5	6,5	6,5	5,8	5,5	6,3	5,3	5,6

Figure V.4 — La surface et le rendement de la production de l'avoine par région et par année

processus ETL automatisant autant que possible l'intégration de ces tableaux multiples.

2.2 L'architecture fonctionnelle du protoytpe

L'architecture fonctionnelle de notre prototype est illustrée dans la Figure V.5. Ce prototype est composé de trois modules correspondant aux trois étapes d'Extraction-Transformation-Chargement. Le premier module permet de détecter et de reconnaître des données tabulaires dans des sources en format XLS ou CSV. Ce module retourne des graphes codés dans le format Graphml. Le deuxième module permet d'intégrer holistiquement (plusieurs graphes en même temps) les graphes en format Graphml issus du premier module. Ce deuxième module fournit un graphe intégré en format graphml. Le troisième module permet de concevoir un schéma multidimensionnel à partir du graphe intégré issu du deuxième module. Ce dernier module fournit en sortie un script de création et d'alimentation d'une base de données multidimensionnelles.

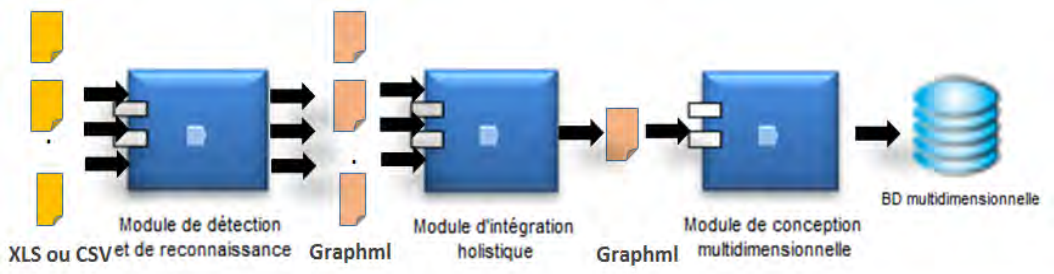


Figure V.5 — Architecture fonctionnelle de notre prototype

2.3 Le module de détection et de reconnaissance de données tabulaires

Le premier module assure la détection et la reconnaissance des tableaux statistiques. Ce module appelé ODET (Open Data Extraction Tool) a été implémenté dans un environnement JAVA. ODET prend en entrée des fichiers tabulaires en format excel ou csv, et génère en sortie un graphe de propriétés sérialisé sous la forme d'un fichier GraphML [1] (graphe encodé en XML). Nous utilisons la librairie blueprints [2] qui permet la gestion des fichiers GraphML.

La Figure V.6 montre l'outil ODET dans son état initial, l'utilisateur doit ouvrir un fichier tabulaire en cliquant sur le menu File → Open, puis sélectionner une ou plusieurs sources de données à traiter.

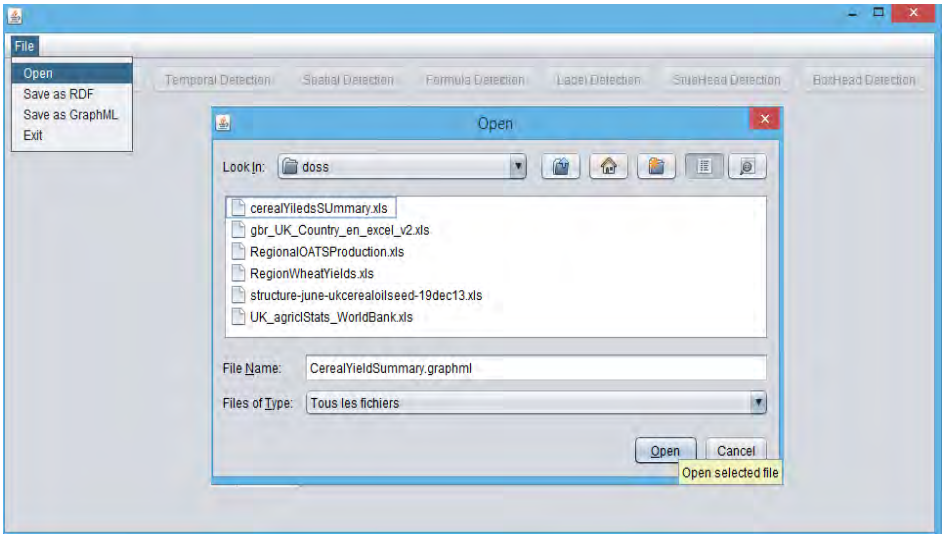
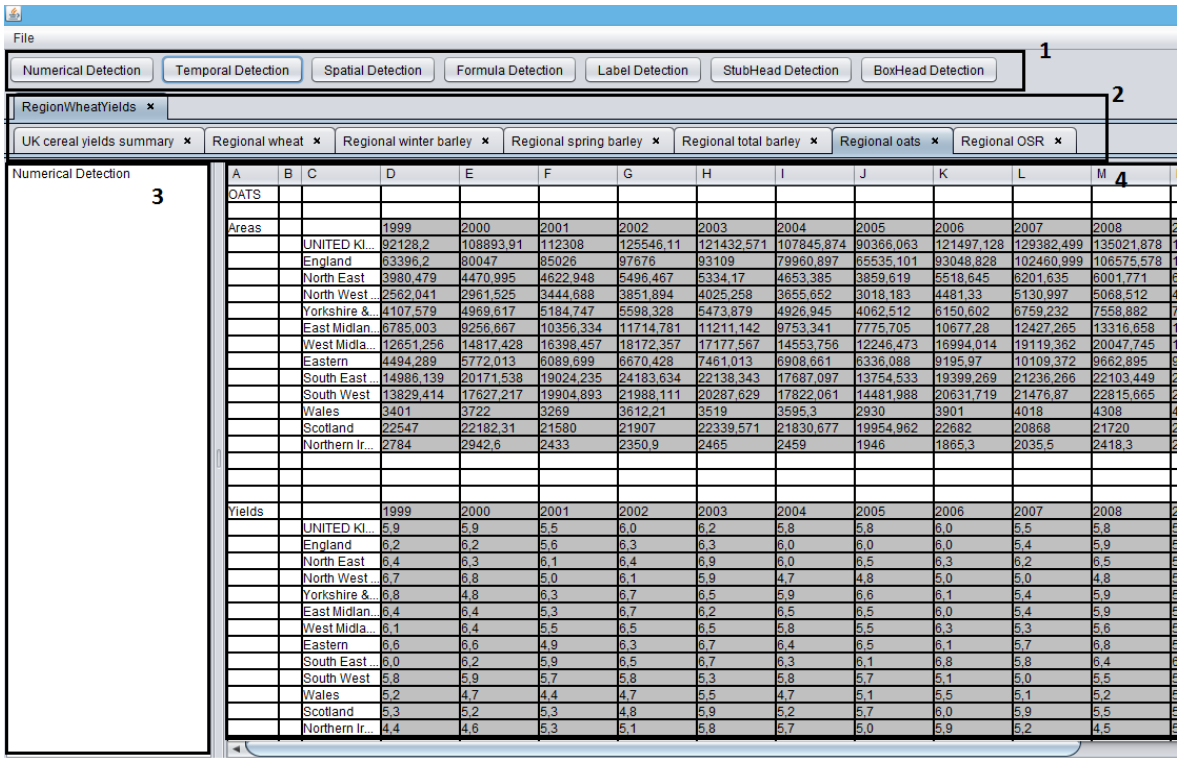


Figure V.6 — Menu fichier de l'outil ODET

Les actions de détection et de reconnaissance automatiques des composants du tableau s'affichent dans la barre d'outils (le cadre 1). Un nouvel onglet ainsi que des sous-onglets s'ouvrent pour la source de données dans le cadre 2, comme le montre la Figure V.7. Dans le cadre 3, l'historique des actions de détection s'affiche afin de permettre de faire des corrections sur les détections. Dans le cadre 4, nous affichons le contenu de la source de données sous format matriciel.

Concernant les actions de détection automatique (Détection Numérique, Label...), nous avons attribué un ordre dans l'exécution de ces activités conformément aux trois niveaux d'activité que nous avons évoqué dans le chapitre 2. Par exemple, pour les activités de détection de StubHead et de BoxHead, un message d'erreur s'affiche si l'utilisateur n'a pas exécuté les activités qui dépendent de ces dernières. De plus, nous avons attribué à chaque activité un code couleur spécifique : lorsque l'utilisateur clique sur une activité alors le bloc de cellules détecté sera coloré par le code couleur correspondant et le nom de l'activité exécutée s'affiche dans le cadre 3. Dans la Figure V.7, les blocs colorés en gris résultent de l'exécution de l'activité de détection numérique. Comme les activités appartiennent à trois niveaux différents et qu'il faut afficher une seule couleur à la fois sur les blocs, nous avons

1. http ://graphml.graphdrawing.org/
2. https ://github.com/tinkerpop/blueprints/wiki



A	B	C	D	E	F	G	H	I	J	K	L	M
OATS												
Areas			1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
		UNITED KI...	92128,2	108893,91	112308	125546,11	121432,571	107845,874	90366,063	121497,128	129382,499	135021,878
		England	63396,2	80047	85026	97676	93109	79960,897	65535,101	93048,828	102460,999	106575,578
		North East	3980,479	4470,995	4622,948	5496,467	5334,17	4653,385	3859,619	5518,645	6201,635	6001,771
		North West	2562,041	2961,525	3444,688	3851,894	4025,258	3655,652	3018,183	4481,33	5130,997	5068,512
		Yorkshire &	4107,579	4969,617	5184,747	5598,328	5473,879	4926,945	4062,512	6150,602	6759,232	7558,882
		East Midlan	6785,003	9256,667	10356,334	11714,781	11211,142	9753,341	7775,705	10677,28	12427,265	13316,658
		West Midla	12651,256	14817,428	16398,457	18172,357	17177,567	14553,756	12246,473	16994,014	19119,362	20047,745
		Eastern	4494,289	5772,013	6089,699	6670,428	7461,013	6908,661	6336,088	9195,97	10109,372	9662,895
		South East	14986,139	20171,538	19024,235	24183,634	22138,343	17687,097	13754,533	19399,269	21236,266	22103,449
		South West	13829,414	17627,217	19904,893	21988,111	20287,629	17822,061	14481,988	20631,719	21476,87	22815,665
		Wales	3401	3722	3269	3612,21	3519	3595,3	2930	3901	4018	4308
		Scotland	22547	22182,31	21580	21907	22339,571	21830,677	19954,962	22682	20868	21720
		Northern Ir	2784	2942,6	2433	2350,9	2465	2459	1946	1865,3	2035,5	2418,3
Yields			1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
		UNITED KI...	5,9	5,9	5,5	6,0	6,2	5,8	5,8	6,0	5,5	5,8
		England	6,2	6,2	5,6	6,3	6,3	6,0	6,0	6,0	5,4	5,9
		North East	6,4	6,3	6,1	6,4	6,9	6,0	6,5	6,3	6,2	6,5
		North West	6,7	6,8	5,0	6,1	5,9	4,7	4,8	5,0	5,0	4,8
		Yorkshire &	6,8	4,8	6,3	6,7	6,5	5,9	6,6	6,1	5,4	5,9
		East Midlan	6,4	6,4	5,3	6,7	6,2	6,5	6,5	6,0	5,4	5,9
		West Midla	6,1	6,4	5,5	6,5	6,5	5,8	5,5	6,3	5,3	5,6
		Eastern	6,6	6,6	4,9	6,3	6,7	6,4	6,5	6,1	5,7	6,8
		South East	6,0	6,2	5,9	6,5	6,7	6,3	6,1	6,8	5,8	6,4
		South West	5,8	5,9	5,7	5,8	5,3	5,8	5,7	5,1	5,0	5,5
		Wales	5,2	4,7	4,4	4,7	5,5	4,7	5,1	5,5	5,1	5,2
		Scotland	5,3	5,2	5,3	4,8	5,9	5,2	5,7	6,0	5,9	5,5
		Northern Ir	4,4	4,6	5,3	5,1	5,8	5,7	5,0	5,9	5,2	4,5

Figure V.7 — Activités d'ODET

défini un ordre entre les couleurs des activités. L'ordre des couleurs des activités de détection sémantique (détection spatiale et temporelle) est supérieur à l'ordre des couleurs des activités topologique (détection StubHead et BoxHead) qui est lui même supérieur à l'ordre des couleurs des activités intrinsèques (détection label, formule, numérique).

Dans la Figure V.8, nous illustrons les résultats de détections automatiques selon l'historique des exécutions affichées dans le cadre 3. Le bloc jaune correspond à un BoxHead de données numériques qui ont une sémantique temporelle. La couleur jaune de la détection temporelle est la dernière couleur affectée à ce bloc. Les labels détectés sont colorés en bleu par exemple la cellule OATS. Les blocs roses et jaunes ont été bleus puis lorsque nous avons exécuté l'activité détection StubHead, ils ont pris la couleur rose, ensuite lorsque nous exécutons l'activité détection spatiale, nous obtenons le bloc jaune des données spatiales.

Puisque les activités de détection et de reconnaissance peuvent induire des erreurs, l'outil ODET donne la possibilité à l'utilisateur de corriger les erreurs du système mais aussi de rajouter des blocs qu'il estime pertinents. L'utilisateur peut annuler les actions dans l'historique des activités exécutées comme le montre la Figure V.9. Il peut aussi changer le cadrage et le type des blocs avec des actions manuelles. La Figure V.10 montre comment un utilisateur peut changer par quelques clics manuels le type de données dans le bloc numérique en formules (bloc coloré en orange).

En ce qui concerne l'activité de la classification hiérarchique des concepts, nous l'avons rattachée, dans la version actuelle du prototype, au bouton de sauvegarde des activités "save as GraphML" (comme le montre la Figure V.6). Dans les fichiers GraphML, nous sauvegardons toutes les informations sur les données qui ont été détectées. Dans la Figure V.11, nous

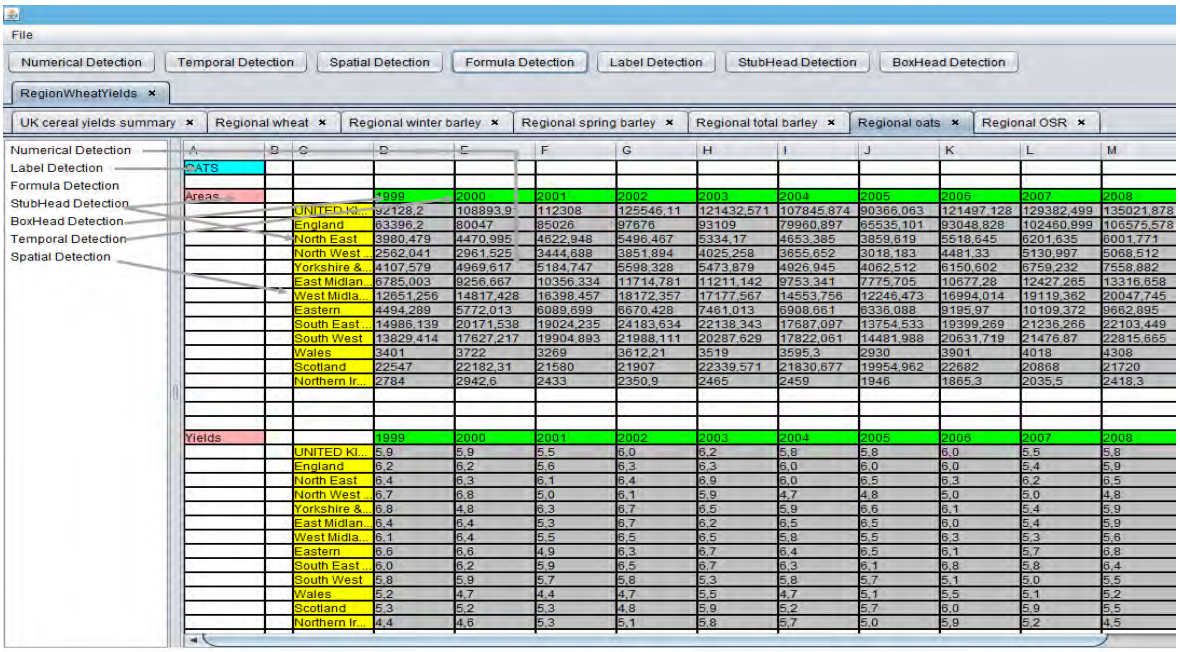


Figure V.8 — Exemples des détections automatiques d'ODET

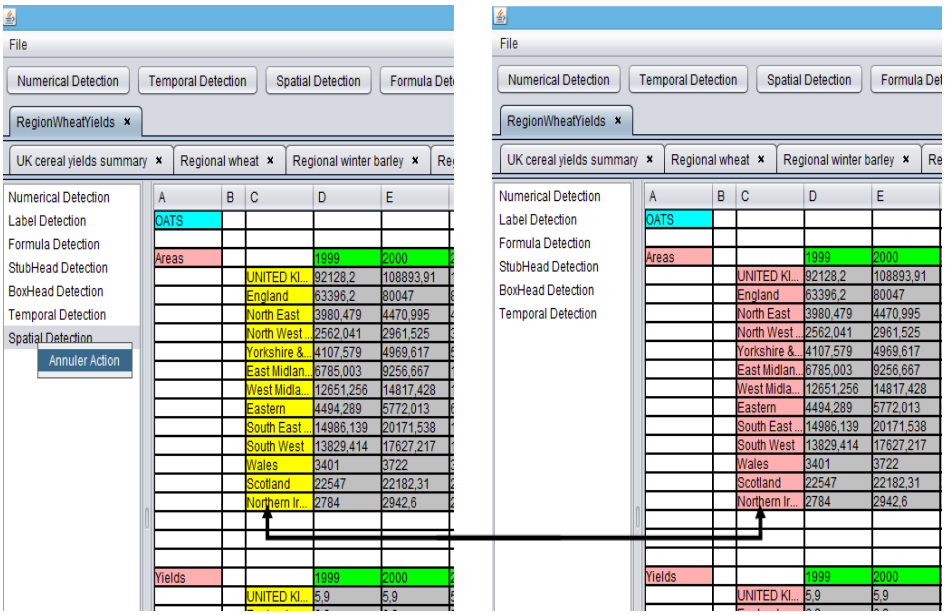


Figure V.9 — Annulation d'activité

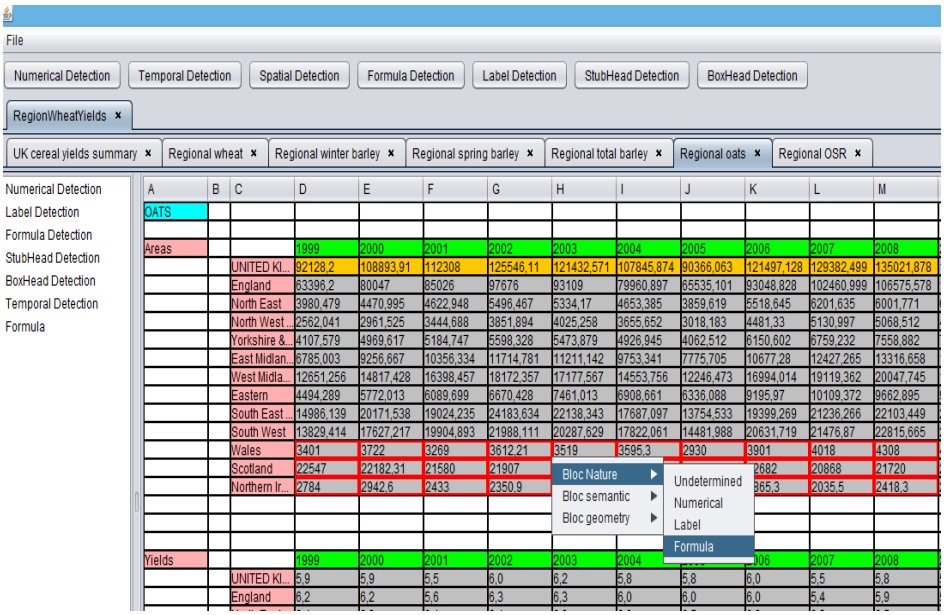


Figure V.10 — Changement du type intrinsèque d'un bloc

avons un extrait d'un graphe de propriétés sérialisé en GraphML. Nous trouvons dans ce dernier les propriétés (DisplayedValue, InherentAnnotation, TopologicAnnotation, SemanticAnnotation, Visualized, idSrcOrigin, nbcCol, nbrLine). La distinction entre les données NumData et les données StructData a été intégrée au niveau des identifiants des noeuds.

```
      <node id="numData:6:3">
           <data key="DisplayedValue"> 92128,2</data>
           <data key="InherentAnnotation">Formula</data>
           <data key="SemanticAnnotation">Undetermined</data>
           <data key="TopologicAnnotation">NumericalBlock:6</data>
           <data key="Visualized">false</data>
           <data key="idSrcOrigin">459888594</data>
           <data key="nbrCol">4</data>
           <data key="nbrLine">4</data>
       </node>

<node id="structData:11:11">
           <data key="DisplayedValue">areas</data>
           <data key="InherentAnnotation">Label</data>
           <data key="SemanticAnnotation">Undetermined</data>
           <data key="TopologicAnnotation">stubHead:3</data>
           <data key="Visualized">true</data>
           <data key="idSrcOrigin">459888594</data>
           <data key="nbrCol">1</data>
           <data key="nbrLine">3</data>
       </node>

     <edge id="structToNumData:3:0:stubhead:3" source="structData:0:1" target="structData:1:2" label=""></edge>
     <edge id="structToNumData:3:0:stubhead:3" source="structData:0:1" target="structData:1:3" label=""></edge>
     <edge id="structToNumData:3:0:stubhead:3" source="structData:0:1" target="structData:1:4" label=""></edge>
     <edge id="structToNumData:3:0:stubhead:3" source="structData:0:1" target="structData:1:5" label=""></edge>
```

Figure V.11 — Un extrait du fichier GraphML

Nous avons illustré dans cette section quelques écrans des fonctionnalités du module ODET. Ce module assure la première partie de notre démarche ETL. Pour ce prototype, il est prévu d'avoir deux sorties, une première sortie en graphe de propriétés et une deuxième sortie en graphe RDF. Le graphe RDF permet d'étendre notre démarche vers un contexte

web sémantique.

2.4 Le module d'intégration holistique de graphes de données tabulaires

Dans cette section, nous présentons le deuxième module destiné à l'intégration holistique des graphes. Ce module a été implémenté pour intégrer les graphes de propriétés (GraphML) des données ouvertes tabulaires conformément à la deuxième phase de notre démarche. Ce prototype prend en entrée plusieurs graphes de propriétés (GraphML) et génère en sortie un graphe de propriétés intégré en format GraphML. Ce prototype accepte d'autres formats de fichiers tels que XML ou XSD. Nous avons rajouté des couches pour transformer les formats XML ou XSD en format GraphML. Cette extension nous a permis d'évaluer notre approche d'intégration sur des bancs d'essais dont les résultats sont illustrés dans la deuxième partie de ce chapitre.

Le module d'intégration a été implémenté dans un environnement Java. Nous avons utilisé l'API Jung [3] pour la manipulation et la visualisation des graphes, l'API cplex-java de l'environnement CPLEX 12.6.1 en version académique pour l'exécution du programme linéaire, l'API blueprints pour la manipulation de GraphML. Pour le calcul de similarité nous avons utilisé les librairies suivantes : OntoSim [4], SimMetric [5], SecondString [6] et WS4J [7].

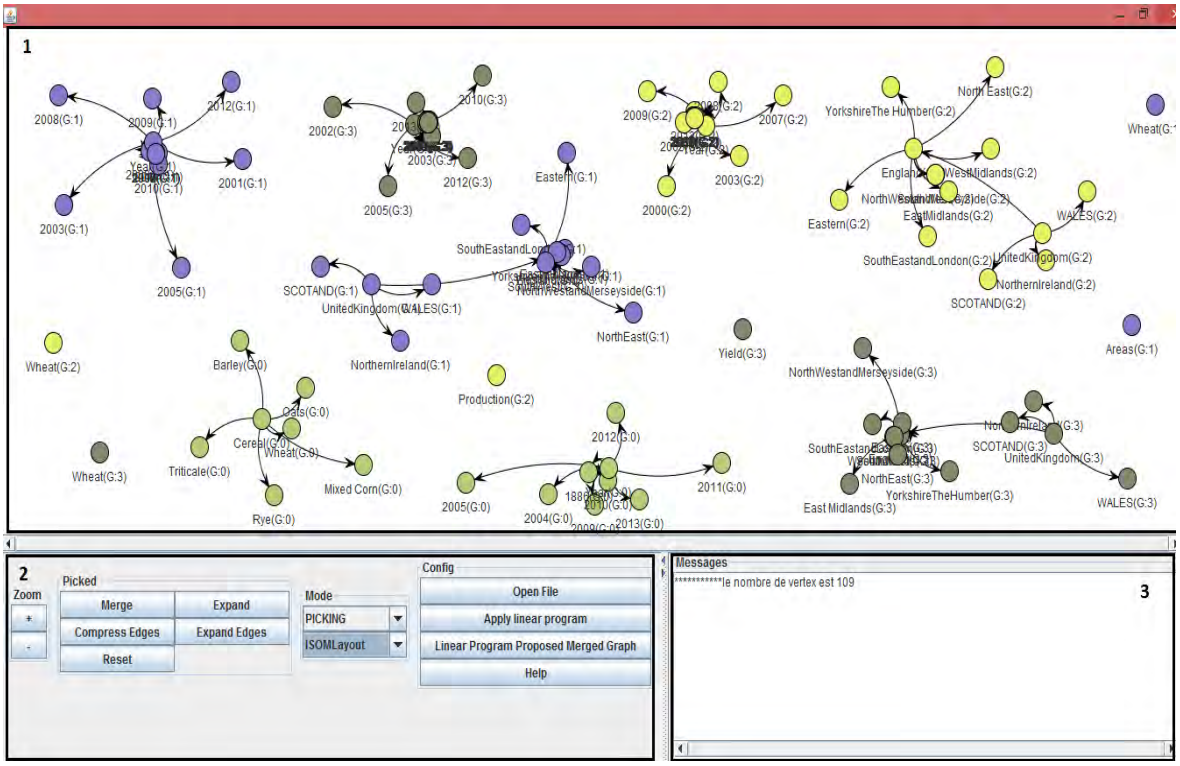


Figure V.12 — La visualisation des graphes avant l'intégration

3. http ://jung.sourceforge.net/
4. http ://ontosim.gforge.inria.fr/
5. http ://sourceforge.net/projects/simmetrics/
6. http ://secondstring.sourceforge.net/
7. https ://code.google.com/p/ws4j/

Supposons qu'un utilisateur sélectionne trois graphes de propriétés issus de données tabulaires de notre scénario d'étude. Après l'ouverture de ces graphes, nous avons le résultat visuel de la Figure V.12. Comme le montre cette dernière, le prototype comporte trois parties qui correspondent aux cadres 1, 2 et 3. Dans le cadre 1, nous visualisons les graphes numérotés. Dans le cadre 2, nous avons :

- l'onglet "picked" qui permet de fusionner ou de séparer des noeuds ou des arcs que l'utilisateur sélectionne.
- l'onglet "Mode" configure le mode de modification ou de non-modification du graphe. L'utilisateur dispose aussi d'un menu déroulant pour changer la mise en page (ou layout) du graphe. Le TreeLayout permet de disposer les noeuds sous format d'arbre, il est adapté pour l'affichage des données structurelles hiérarchiques. Dans la Figure V.12, nous avons choisi le ISOMLayout qui dispose les noeuds connectés dans des zones denses.
- l'onglet "Config" contient l'action d'ouverture des fichiers, l'action de lancement du programme linéaire et l'action d'obtention d'un graphe intégré visuel. Après validation des correspondances sur le graphe intégré visuel en utilisant les boutons merge et expand, l'utilisateur peut générer automatiquement un graphe intégré en GraphML.

Dans le cadre 3, nous affichons des messages sur le temps de calcul écoulé par LP4HM, les correspondances (numéro du graphe source, numéro graphe destination, les labels des noeuds et la mesure de similarité entre eux), le nombre de correspondances trouvées et éventuellement des messages d'erreurs.

Dans la Figure V.13, nous illustrons le résultat visuel du graphe intégré après l'application du programme linéaire.

2.5 Le module de conception d'un schéma multidimensionnel

Le dernier module concerne la conception progressive d'un schéma multidimensionnel à partir du graphe de propriétés intégré.

Ce module est implémenté dans un environnement Java. Il génère un script sql pour le schéma et pour l'alimentation de la base de données ; nous avons testé les scripts générés avec une base de données Postgresql. La Figure V.14 montre un exemple de graphe intégré de notre scénario d'étude. Nous différencions ici les données numériques en vert et les données structurelles en jaune. Dans la deuxième partie de l'interface, nous avons l'arborescence du schéma multidimensionnel.

L'utilisateur doit commencer par identifier les dimensions avec le bouton droit "Add dimension", voir Figure V.15. L'utilisateur spécifie le nom de la dimension, celui des paramètres et sélectionne les noeuds du graphe correspondants aux instances de ces paramètres. Ces étapes sont illustrées dans la Figure V.16 pour la dimension "Time". Lorsque l'utilisateur valide la création de cette dimension, le script de création du schéma de la table et celui de l'alimentation de cette table sont générés. Dans le graphe, les noeuds disparaissent et laissent la place pour un nouveau noeud correspondant à la dimension Time. L'arborescence se met à jour avec la dimension et ses paramètres, voir la Figure V.17.

Une fois que toutes les dimensions sont définies, l'utilisateur peut créer un fait "add

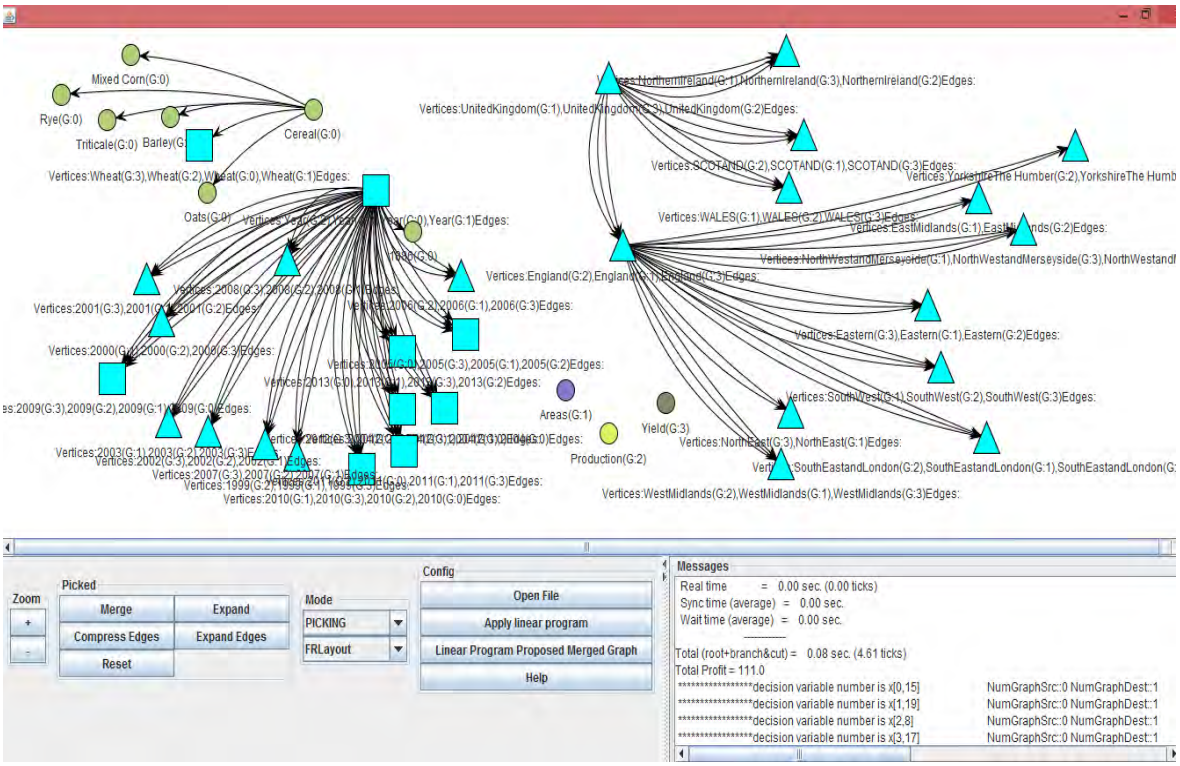


Figure V.13 — La visualisation des graphes après l'intégration

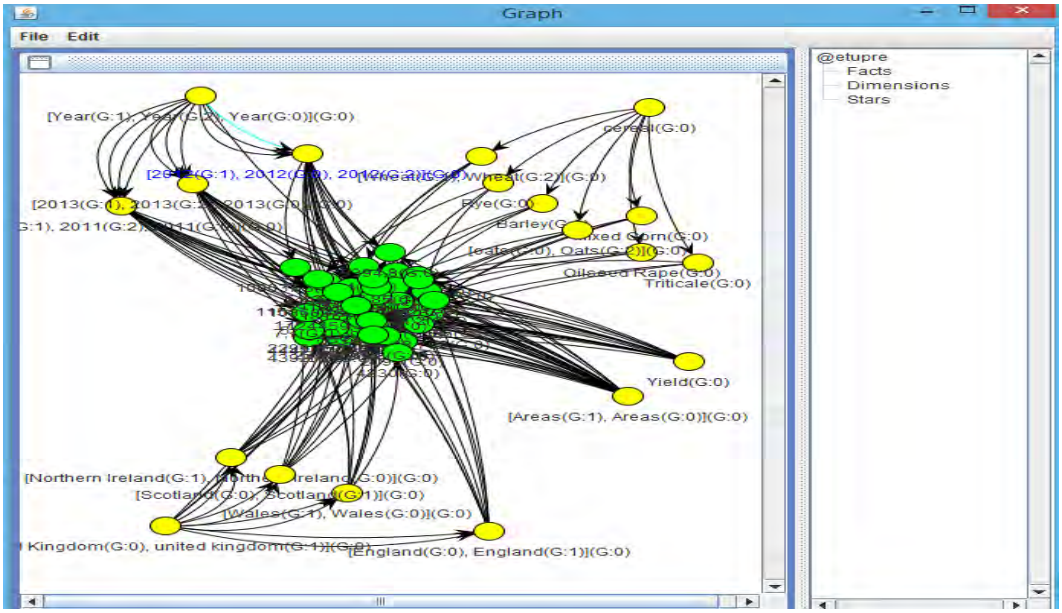


Figure V.14 — Exemple d'un graphe intégré avec les données numériques

fact", comme le montre la Figure V.18. Il doit sélectionner les dimensions et les mesures (dans notre exemple yield et area). La fonction d'agrégation attribuée est avg pour les deux mesures. Nous pouvons remarquer que le graphe de propriétés intégré a été simplifié et transformé progressivement.

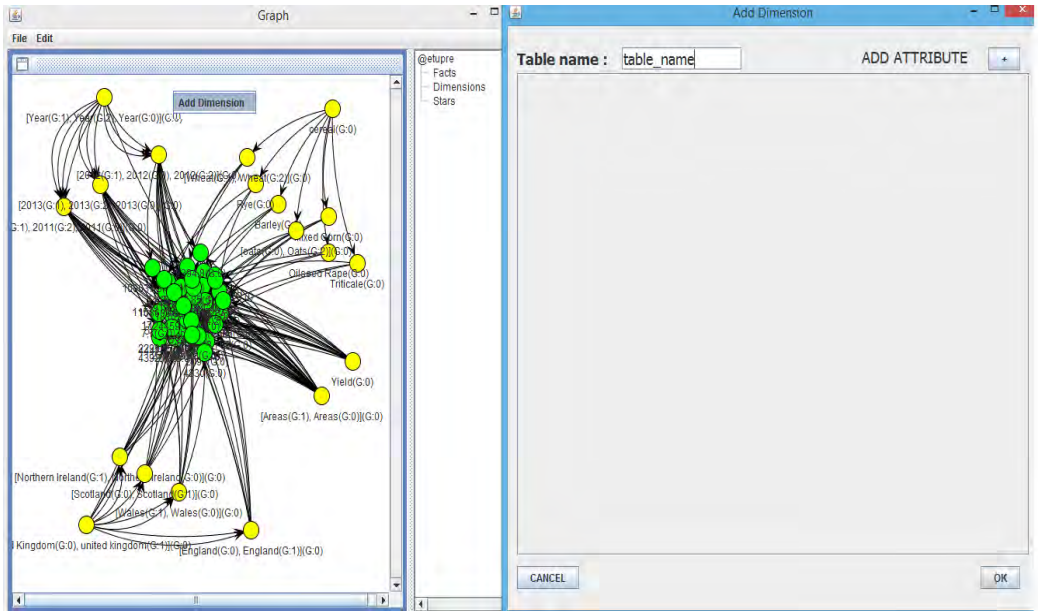


Figure V.15 — Exemple de l'interface d'ajout de dimension

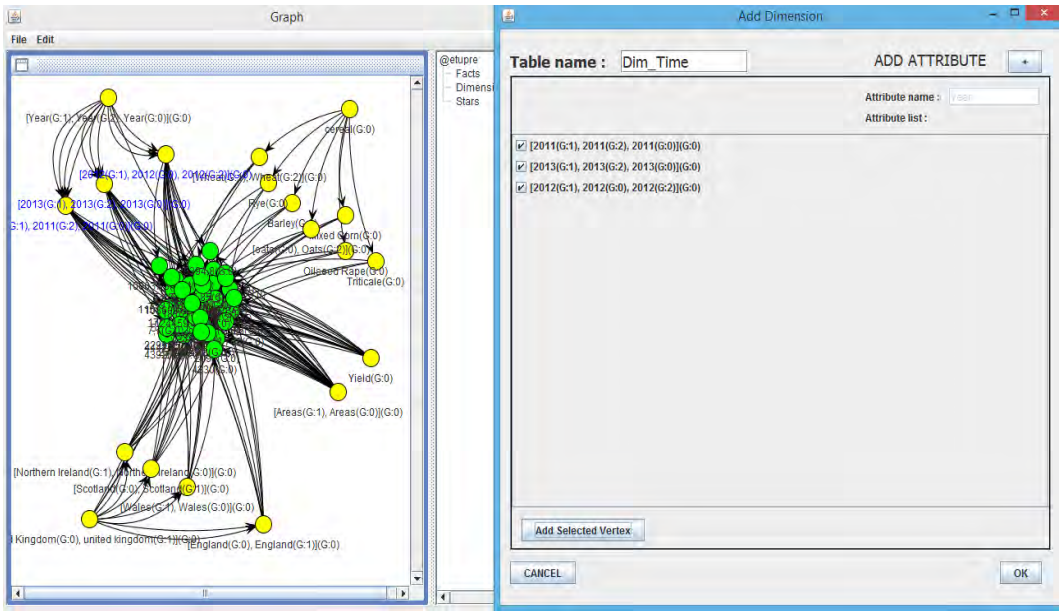


Figure V.16 — Exemple d'ajout de la dimension Time

3 Évaluations

3.1 Évaluation de la détection des données tabulaires

Dans cette section, nous illustrons les résultats d'évaluation de la détection des données tabulaires réalisée par l'outil ODET. L'évaluation a été effectuée sur 100 sources de données tabulaires en format Excel que nous avons sélectionnées sur le portail `data.gouv.fr`. Nous avons sélectionné les dix premières sources en format excel de chaque domaine. Ces

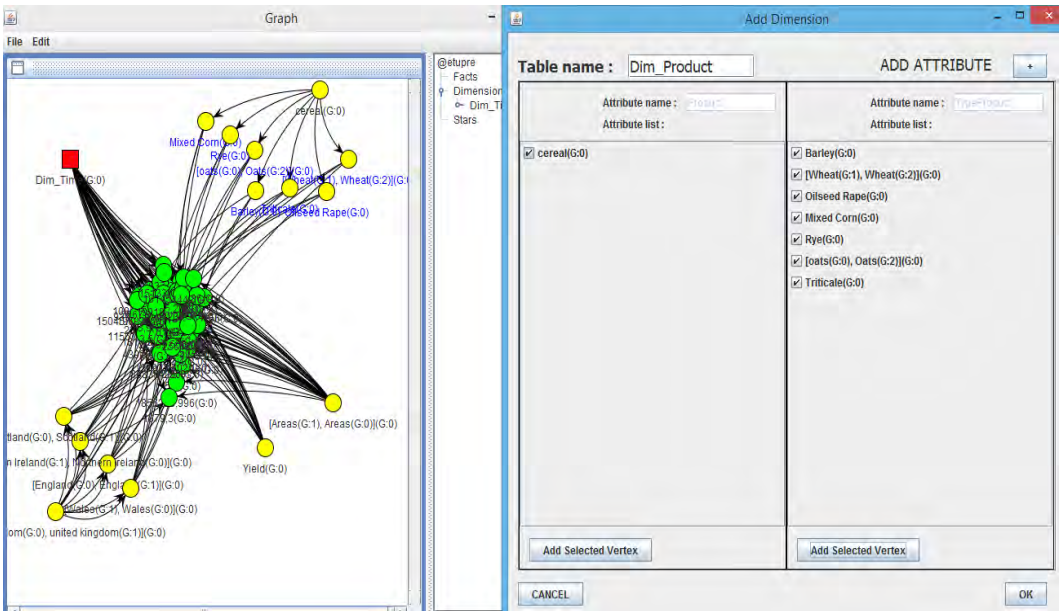


Figure V.17 — Exemple d'ajout de la dimension Product

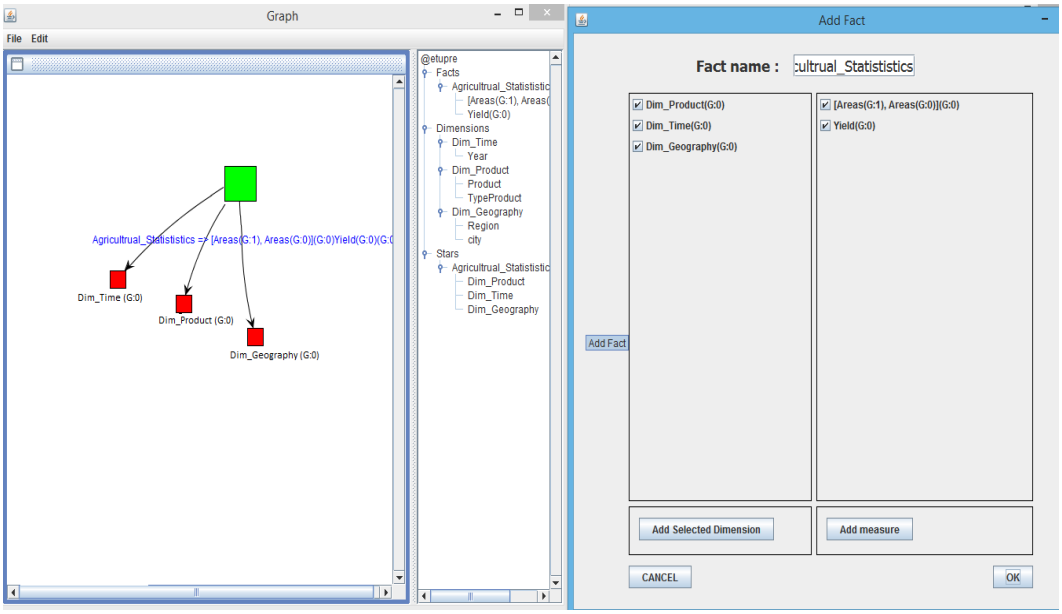


Figure V.18 — Exemple d'ajout du Fait

sources sont en langue française et comportent des données géographiques de la France. Elles couvrent neuf domaines d'étude :

- Agriculture
- Culture
- Travail et économie
- Recherche et éducation
- Hébergement
- International et Europe
- Société

Tableau V.1 — L'efficacité des fonctions de détection automatique

	Détection Numérique	Détection Label	Détection Formule	Détection Temporelle	Détection Spatiale
Efficacité	98,88%	92,22%	57,89%	95,23%	53,84%
Proportion du type de données	100%	100%	21,11%	46,66%	43,33%

– Transport
– Santé

L'objectif de cette évaluation est de valider l'efficacité des fonctions de détection automatique des blocs pertinents du point de vue utilisateur. Nous rappelons qu'un bloc de type i peut être : bloc de label (StubHead et BoxHead), bloc numérique, bloc spatial, bloc temporel et bloc de formule. Nous avons manuellement identifié pour chaque source l'emplacement et le type de chaque bloc. Ensuite, nous avons lancé les fonctionnalités automatiques de l'outil ODET et nous avons vérifié si les blocs détectés automatiquement correspondent ou pas aux blocs que nous avons identifiés manuellement. Pour chaque source et pour chaque type, nous avons calculé le nombre de blocs détectés automatiquement qui figurent dans la liste des blocs manuellement identifiés. Puis, nous avons fait la somme, par type de bloc i, de tous les blocs systèmes pertinents ad_i et tous les blocs md_i provenant de l'expert (nous). L'efficacité de détection correspond ainsi au rapport entre ces deux derniers nombres calculés pour les 100 sources, $Efficacite(Type_i) = \frac{ad_i}{md_i}$.

Le tableau V.1 illustre les résultats de cette évaluation ainsi que la proportion des sources ayant des blocs de type i par rapport au nombre total de sources étudiées, en l'occurrence 100.

Dans le Tableau V.1, nous constatons que toutes les sources contiennent des données numériques et labels, et 50% des sources contiennent des données temporelles et des données spatiales. Par contre les formules sont très rares.

Concernant les résultats d'efficacité des fonctions de détection automatique implémentées dans l'outil ODET, nous remarquons que les algorithmes de détection des données numériques et des labels (y compris les StubHead et les BoxHead) sont très efficaces puisque plus de 90% de blocs de ces types ont été correctement identifiés. Notre proposition de détection des données temporelles est également très performante, nous avons plus que 90% de détection de données temporelles. Par contre, notre proposition de détection des données spatiales doit être améliorée puisque 47% de ces données n'ont pas été correctement identifiées. En particulier, la détermination des numéros des départements parmi les données numériques reste problématique. Nous constatons aussi qu'il faudrait améliorer nos algorithmes de détection des formules.

3.2 Évaluation de l'appariement par paire sur des bancs d'essai comparatifs

Dans cette section, nous présentons les résultats de comparaison des alignements produits par notre approche et des alignements produits par d'autres approches référencées dans la littérature. Nous avons choisi deux bancs d'essai : le premier est orienté utilisateur et le deuxième est orienté schéma.

3.2.1 Les mesures d'évaluation

Nous présentons dans cette partie, les deux types de mesures que nous utilisons dans notre étude comparative :

- les mesures de la qualité des correspondances ;
- les mesures de l'effort engagé par les utilisateurs.

3.2.1.1 La qualité des correspondances

La qualité des correspondances retournées par un système d'appariement est communément évaluée par les mesures de **précision** et de **rappel**. Pour une paire de schémas (S1,S2) dont le nombre d'éléments est N1 et N2 respectivement, le produit cartésien $N1 \times N2$ représente l'ensemble de toutes les correspondances possibles. Parmi cet ensemble, un expert va donner un sous-ensemble de correspondances, noté E, qu'il estime le plus pertinent. Quant au système d'appariement, il trouve un sous-ensemble de correspondances, noté S, qui doit dans l'idéal se confondre avec E. Comme le montre la Figure V.19, l'intersection de $S \cap E$ représente les vrais positifs alors que $E - S$ correspond aux faux négatifs puisque le système ne les a pas trouvé alors qu'il devrait les trouver. $S - E$ représente les faux positifs puisque le système les a retourné alors que l'expert ne les a pas jugés pertinents. Enfin le reste des correspondances sont les vrais négatifs puisqu'elles ne figurent ni dans les correspondances du système, ni dans les correspondances de l'expert.

La **Précision** P est le rapport entre les vrais positifs $S \cap E$ et les correspondances trouvées par le système S . La précision mesure le degré d'exactitude des correspondances trouvées par le système [Euzenat et Shvaiko, 2013].

$$Precision = P = \frac{|E \cap S|}{|E|}$$

Le **Rappel** R est le rapport entre les vrais positifs $S \cap E$ et les correspondances proposées par l'expert E. Le rappel mesure le degré de complétude des correspondances trouvées par le système [Euzenat et Shvaiko, 2013].

$$Rappel = R = \frac{|E \cap S|}{|S|}$$

L'évaluation de la qualité en se basant uniquement sur le rappel et la précision est insuffisante car ces deux mesures sont inversement proportionnelles. En effet, le rappel peut être maximisé au détriment d'une faible précision en retournant tous les vrais positifs. Alors que la précision peut être maximisée au détriment d'un faible rappel en ne retournant que quelques correspondances de l'ensemble des vrais positifs. Il est alors indispensable d'examiner une mesure combinée de la précision et du rappel. Cette mesure combinée est connue sous le nom de **F-Mesure**, définie généralement par la formule suivante :

$$FMeasure_{\alpha} = \frac{P \times R}{(1-\alpha) \times P + \alpha R}$$

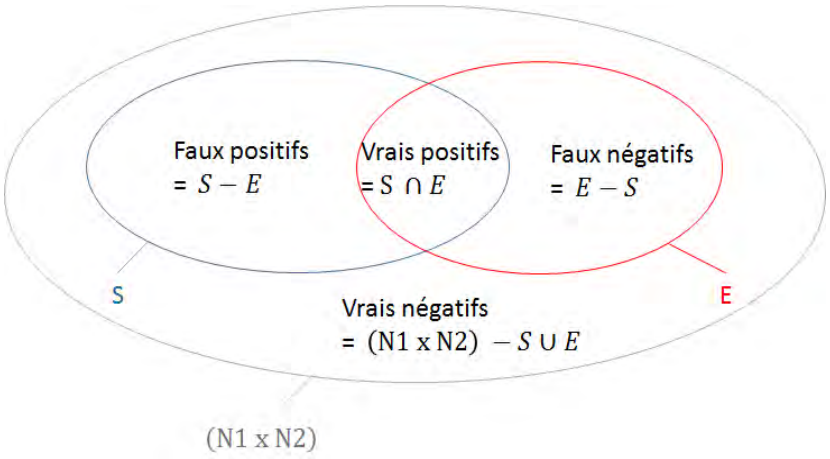


Figure V.19 — La relation entre les correspondances du système et les correspondances d'expert [Euzenat et Shvaiko, 2013]

Usuellement, la valeur de poids α est égale à 0.5, représentant ainsi une moyenne harmonique entre la précision et le rappel. La formule de la F-Mesure devient comme suit :

$$FMeasure_{0.5} = \frac{2 \times P \times R}{P + R}$$

Plus α s'approche de 1, plus la F-Mesure est influencée par la précision. En contrepartie, plus α s'approche de 0, plus la F-Mesure est influencée par le rappel.

3.2.1.2 L'effort engagé par l'utilisateur

L'effort engagé par l'utilisateur (post-match effort) est l'estimation maximale de la quantité de travail qu'un utilisateur doit engager pour valider les correspondances proposées par le système et les compléter [Duchateau *et al.*, 2011]. Deux mesures ont été proposées dans la littérature pour évaluer l'effort engagé par l'utilisateur : Accuracy et HSR.

– La mesure **Accuracy** ou Overall, proposée par [Melnik *et al.*, 2002], évalue l'effort que l'utilisateur engage pour corriger le nombre de faux négatifs par rapport au nombre de correspondances pertinentes E proposées par l'expert. Cette mesure assume que l'effort engagé pour la validation des correspondances systèmes est le même que l'effort engagé pour chercher les correspondances manquantes. La mesure d'accuracy prend ses valeurs dans l'intervalle $[-1, 1]$. Elle est définie par la formule suivante :

$$Accuracy = 1 - (\frac{|E - S| + |S - E|}{E}) = R \times (2 - \frac{1}{P})$$

La mesure d'accuracy est corrélée négativement par rapport aux valeurs de la précision. En effet, si la valeur de la précision est inférieure à 0.5 alors la valeur de l'accuracy est négative [Melnik *et al.*, 2002] [Euzenat et Shvaiko, 2013]. Ceci s'interprète en considérant que l'effort engagé pour la suppression des correspondances non-pertinentes et l'ajout des correspondances manquantes est beaucoup plus important que celui de rechercher les correspondances manuellement.

– La mesure **HSR** (Human Spared Resources), proposée par [Duchateau et Bellahsene, 2014], évalue l'effort humain épargné en utilisant les résultats d'un système d'appariement. Cette mesure prend en considération qu'il y a un schéma majeur S_L et un schéma mineur S_l, où le nombre d'éléments du premier est plus grand que le nombre d'éléments du second. La mesure HSR, dont les valeurs sont comprises dans l'intervalle $[0, 1]$, est exprimée comme suit :

$$HSR = 1 - \frac{NUI}{|S_L| \times |S_l|}$$

NUI représente le nombre d'interactions que l'utilisateur doit effectuer pour valider les correspondances S trouvées par le système et pour découvrir manuellement les correspondances manquantes faisant partie de l'ensemble des faux négatifs. La formule NUI est calculée comme suit :

$$NUI = \begin{cases} |S_l| + |S_L| & \text{si } |S| = 0 \\ \frac{|S_l|-|S\cap E|}{|E|-|S\cap E|} + \sum_{i=1}^{|S_l|-|S\cap E|}(|S_L| - |S \cap E| - \frac{i\times(|E|-|S\cap E|)}{|S_l|-|S\cap E|} - \frac{|R|-|S\cap E|}{|S_l|-|S\cap E|}) & \text{sinon} \end{cases}$$

Nous pensons qu'il est intéressant d'examiner l'estimation de l'effort engagé de ces deux mesures même si l'accuracy affiche plusieurs limites par rapport au HSR. En effet, dans l'étude comparative entre HSR et accuracy [Duchateau *et al.*, 2011], dont les résultats sont présentés dans la Figure V.20, les auteurs affirment que l'accuracy est une mesure plus optimiste que le HSR pour les valeurs de précisions importantes alors que toutes les valeurs de l'accuracy sont inférieures à zéro dés que les valeurs de la précision sont inférieures à 50%. La mesure HSR est plus équilibrée que l'accuracy puisqu'elle se base sur l'estimation du nombre d'interactions de l'utilisateur et pas uniquement sur la précision et le rappel. Un autre inconvénient de l'accuracy est qu'elle ne considère pas la taille des schémas alors que l'effort qu'un utilisateur engage pour chercher les correspondances manquantes pour de grands schémas est beaucoup plus important que l'effort engagé pour de petits schémas [Duchateau *et al.*, 2011].

3.2.2 Résultats d'évaluation sur un banc d'essais orienté utilisateurs

Dans cette section, nous examinons les résultats d'évaluation de notre approche d'appariement par rapport à d'autres approches de la littérature sur le banc d'essais orienté utilisateurs proposé par [Melnik *et al.*, 2002]. Nous rappelons que notre approche est constituée de deux versions du programme linéaire LP4HM : (1) la version primaire avec des variables de décision binaires que nous notons **LP4HM** et (2) la version relaxée avec des variables fractionnaires que nous notons **LP4HM(Relaxé)**. Ces deux versions seront évaluées sans la contrainte du seuil de similarité.

Les approches que nous avons sélectionnées de la littérature sont :

– **COMA++** [Aumueller *et al.*, 2005] implémentée dans l'outil COMA3.0, `http://dbs.uni-leipzig.de/Research/coma.html`. Nous avons utilisé le résultat d'appariement combiné des différentes stratégies proposées par cet outil à l'exception de la stratégie de réutilisation et de fragmentation.

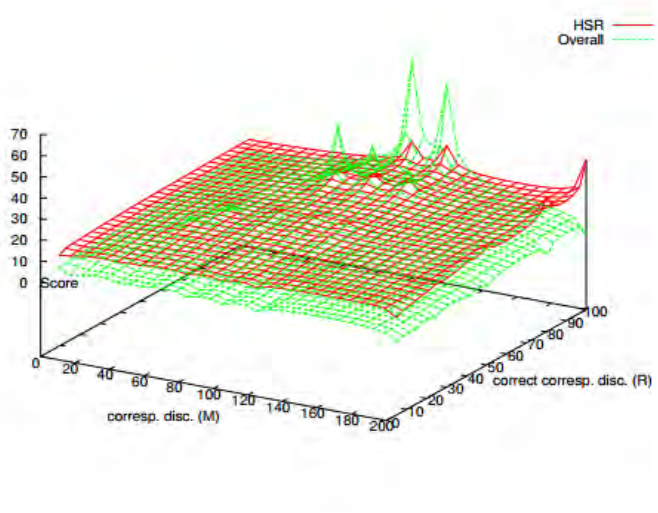


Figure V.20 — Comparaison entre HSR et Overall [Duchateau *et al.*, 2011]

- **BMatch** [Duchateau *et al.*, 2007] implémentée dans une API disponible sur le lien `http://liris.cnrs.fr/~fduchate/research/tools/bmatch/`. Nous avons utilisé cet API avec le paramétrage par défaut.
- **Similarity Flooding** (SF) [Melnik *et al.*, 2002] implémentée dans l'outil RONDO disponible sur le lien `http://infolab.stanford.edu/~melnik/mm/rondo/`. Nous avons utilisé cette approche avec son paramétrage par défaut. Originairement, l'approche SF a été évaluée par ce banc d'essais [Melnik *et al.*, 2002]. Nous avons refait l'expérimentation sans modifier les paramètres par défaut de l'outil afin d'examiner les mesures de rappel, précision et HSR qui ne sont pas disponibles dans l'évaluation faite par les auteurs [8].

3.2.2.1 Description du banc d'essais

Nous avons choisi le banc d'essais (ou benchmark) proposé par [Melnik *et al.*, 2002] et disponible sur le lien `http://infolab.stanford.edu/~melnik/mm/sfa/`. Il s'agit de l'unique [9] banc d'essais orienté utilisateurs proposé pour l'appariement de paires de schémas.

Ce banc d'essais est composé de 9 tâches : les tâches 1,2,3 concernent l'appariement de schémas XML, les tâches 4,5,6 concernent l'appariement de schémas XML avec des instances et les tâches 7,8, 9 concernent l'appariement de schémas de bases de données relationnelles. Sept utilisateurs de l'université de Stanford ont participé à la génération des correspondances. Aucune méta-donnée ou explication sur les contextes des schémas n'a été fournie et les utilisateurs étaient libres de proposer des correspondances complexes ($n : m$) ou simples ($1 : 1$). Les correspondances proposées par les utilisateurs forment l'ensemble E.

Nous avons étudié les paires de schémas de chaque tâche selon les trois caractéristiques

8. Les mesures d'accuracy que nous avons trouvées sont légèrement différentes des mesures publiées dans [Melnik *et al.*, 2002], nous pensons que cela provient du paramétrage par défaut.
9. http ://www.ontologymatching.org/evaluation.html

Tableau V.2 — Caractéristiques des schémas des différentes tâches

	Hétérogénéité			Structure		Écart		
	faible	moyenne	forte	plate	imbriquée	faible	moyen	fort
Tâche 1	×			×		×		
Tâche 2		×			×	×		
Tâche 3			×		×	×		
Tâche 4	×				×	×		
Tâche 5			×		×	×		
Tâche 6			×		×	×		
Tâche 7			×	×			×	
Tâche 8			×	×			×	
Tâche 9			×	×				×

suivantes :

- l'hétérogénéité des labels des éléments répartie sur trois niveaux : "faible" ($[0, 0.3[$), "moyenne" ($[0.3, 0.6[$) et "forte" ($[0.6, 1]$). Ces intervalles représentent l'estimation de l'hétérogénéité calculée en fonction de l'écart entre la médiane des maxima des distances terminologiques et la médiane des maxima des distances linguistiques que nous utilisons.
- la structure des schémas qui est définie en fonction de la profondeur des schémas. Elle peut être "plate" (profondeur <3) ou "imbriquée" (profondeur >=3).
- l'écart entre deux schémas qui est calculé en divisant le nombre d'éléments du schéma mineur sur le nombre d'éléments du schéma majeur. Ce rapport est réparti sur trois variantes : "faible" ($[0.6, 1]$), "moyen" ($[0.3, 0.6[$) et "fort" ($[0, 0.3[$).

Le tableau V.2 synthétise les caractéristiques des schémas de chaque tâche.

Afin de pouvoir appliquer le banc d'essais sur notre approche, nous avons transformé chaque schéma en un graphe de propriétés comme le montre la Figure V.21 pour les schémas de la tâche 3. Nous avons utilisé les noms des éléments comme labels de noeuds et nous avons transformé l'organisation entre les éléments en hiérarchie.

3.2.2.2 Résultats globaux

Tableau V.3 — Les résultats des mesures de qualité par approche pour la moyenne des utilisateurs et des tâches

	Précision (%)	Rappel (%)	F-Measure (%)	Accuracy (%)	HSR (%)
LP4HM	67	58	62	30	**81**
LP4HM(Relaxé)	58	**66**	60	23	**81**
COMA++	72	50	58	32	76
BMatch	22	47	28	0	69
Similarity Flooding	**81**	55	**65**	**43**	80

Le tableau V.3 montre les résultats des mesures de qualité pour la moyenne des utilisateurs et des tâches pour les différentes approches. Nous avons d'une part les deux versions de notre approche qui n'utilisent pas de seuil de similarité, d'autre part nous

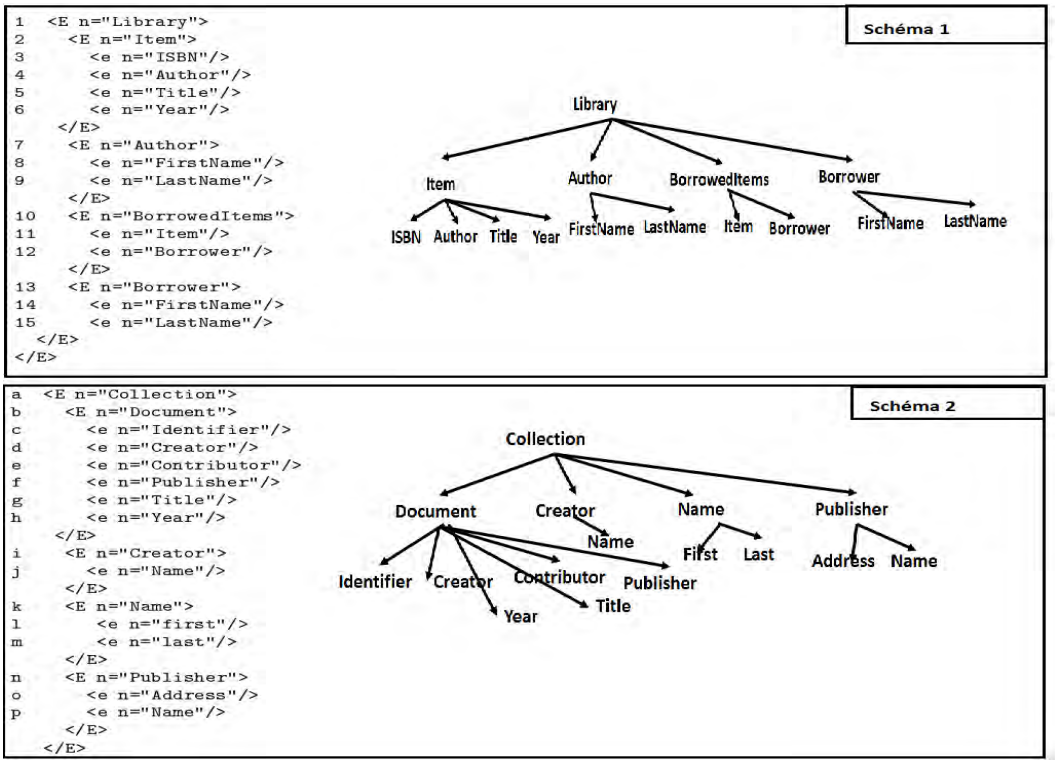


Figure V.21 — Transformation des schémas XML en graphes

avons les approches COMA++, BMatch et SF qui utilisent des seuils de similarités. Ne pas utiliser un seuil de similarité pour notre approche est un choix intentionné. En effet, la configuration d'un seuil de similarité est un problème pour les outils d'appariement [Shvaiko et Euzenat, 2013] qui rend les outils difficilement utilisables par les utilisateurs en particulier non-experts. Nous voulons montrer qu'il est envisageable de se passer du seuil à travers notre approche. Par ailleurs, ce banc d'essai est orienté utilisateurs donc le seuil de similarité peut dépendre aussi de chaque utilisateur. Pouvoir fournir des résultats convenables à tous les utilisateurs sans se soucier de la configuration du seuil de similarité nous semble constituer un avantage de notre approche.

Nous montrerons un exemple concret de la difficulté que rencontre les outils d'appariement pour le choix du seuil de similarité à travers les résultats de l'approche SF. D'après le tableau V.3, SF dépasse les autres approches dans les résultats de précision et d'accuracy toutefois ces résultats ont été sélectionnés et recommandés par les auteurs avec un seuil de similarité maximisé égal à 1. En effet, les auteurs affirment dans [Melnik *et al.*, 2002] que s'ils n'utilisent pas de seuil de similarité ils obtiennent un rappel égal à 100%, une précision égale à 4% et une accuracy égale à -2144%. La F-Mesure dans ce cas est égale à 7%. L'écart entre les résultats de cette approche sans et avec un seuil de similarité est très important ce qui illustre la problématique de l'utilisation et du choix d'un seuil. Les résultats de SF, sans seuil de similarité, sont très mauvais par rapport à notre approche. En fait, même si le système renvoi toutes les correspondances que l'utilisateur a estimé pertinentes (100% de rappel), ces résultats ne présentent que 4% des correspondances retournées par le système. Sachant que l'utilisateur doit valider/invalider ces résultats, il doit fournir beaucoup d'effort pour

invalider les résultats non-pertinents et repérer les résultats pertinents (ce qui est reflété par l'énorme valeur négative de l'accuracy). Les deux versions de notre approche s'avèrent meilleures que l'approche de SF sans utilisation de seuil de similarité.

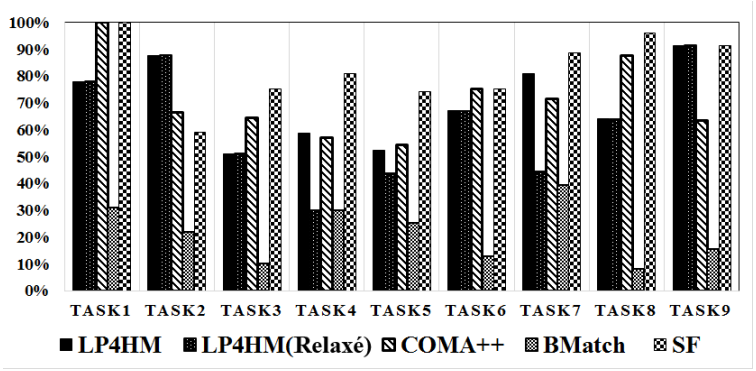
Les résultats de notre approche sont également compétitifs par rapport aux résultats des autres approches utilisant des seuils de similarité. En effet, les résultats de rappel pour les deux versions de notre approche sont meilleurs que les résultats de rappel pour les autres approches. La précision dépasse les 50% pour les deux versions LP4HM et LP4HM(Relaxé). Nous considérons que ce sont des valeurs très correctes par rapport à la précision de COMA++ et SF qui utilisent des seuils de similarité élevés et à la précision de l'approche BMatch qui utilise un seuil de similarité faible. Comme la précision et le rappel sont insuffisants pour qualifier la qualité d'une approche, nous examinons les résultats de F-Mesure ; les deux versions de notre approche occupent les 2ème et 3ème place en F-Measure juste après l'approche SF avec un écart très faible. Ces résultats sont assez significatifs puisqu'ils avoisinent les meilleurs résultats de l'approche SF configurée.

Examinons à présent les mesures de qualité de l'effort épargné par les utilisateurs. Pour la mesure d'accuracy notre approche est en troisième position précédée par les approches SF et COMA++ ; pour la mesure HSR notre approche est en première position suivie de très près par SF. Nous avons indiqué dans la section précédente que la mesure HSR semble plus significative que la mesure accuracy puisqu'elle ne pénalise pas les faibles mesures de précision et prend en considération le nombre d'éléments des schémas appariés [Duchateau et Bellahsene, 2014]. Les résultats de l'approche BMatch appuient cette intuition. En effet, pour BMatch la précision est inférieure à 50% d'où une accuracy inférieure à 0% (nous avons arrondi cette valeur à 0%) alors que la valeur du HSR est de 69%, ce qui est tout de même important, et montre qu'il y a un effort épargné pour les 47% de correspondances pertinentes retournées par l'approche.

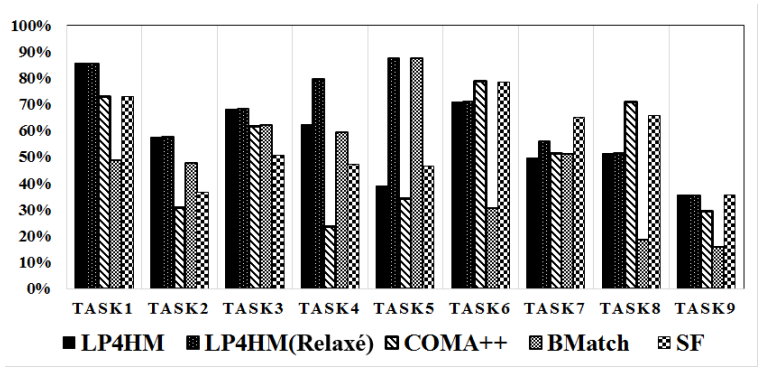
Nous concluons cette section par un comparatif des deux versions de notre approche. On observe que LP4HM(Relaxé) est meilleur en rappel et moins bon en précision par rapport à LP4HM. On observe aussi que l'écart que LP4HM(Relaxé) a gagné en rappel a été perdu en précision. En effet, comme le nombre de correspondances proposées par les utilisateurs ne varient pas, la différence entre le rappel de LP4HM(Relaxé) et le rappel de LP4HM correspond au nombre de correspondances de cardinalités $n : m$ que l'approche LP4HM(Relaxé) a réussi de capturer par relaxation des variables de décision. Par contre la précision a diminué puisque la relaxation des variables de décision LP4HM(Relaxé) retourne beaucoup plus de correspondances inutiles de LP4HM. Donc le nombre total de correspondances retournées par la version LP4HM(Relaxé) est beaucoup plus important que le nombre de correspondances retournées par la version LP4HM ce qui explique la baisse de la valeur de précision.

3.2.2.3 Résultats détaillés

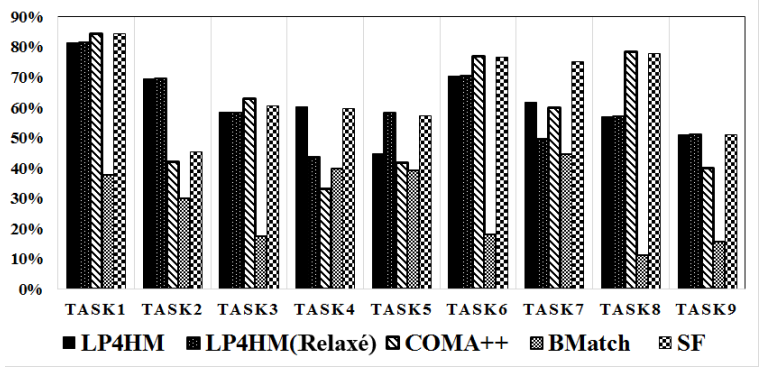
Concernant les résultats détaillés, nous allons examiner les mesures de qualité des correspondances par tâche pour la moyenne des utilisateurs puis par utilisateur pour la moyenne des tâches. Nous regardons ensuite de la même façon le détail des mesures de l'effort engagé par les utilisateurs.



(a) La précision



(b) Le rappel



(c) La F-Mesure

Figure V.22 — Les résultats de précision, rappel et F-Mesure par tâche pour la moyenne des utilisateurs

Les figures V.22(a), V.22(b) et V.22(c) montrent respectivement la précision, le rappel et la F-Measure des différentes approches par tâche pour la moyenne des utilisateurs. Globalement pour les deux versions de notre approche et BMatch, les valeurs de rappel sont plus importantes que les valeurs de précision. Alors que pour les approches COMA++ et SF, les valeurs de précision sont plus importantes que les valeurs de rappel. Au niveau F-Mesure, nous remarquons que les écarts qui existaient entre rappel et précision se redressent par la moyenne harmonique. Pour la plupart des tâches (à l'exception des tâches 2 et 8) les résultats de notre approche sont quasiment les mêmes que l'approche SF. Pour COMA++, les résultats de F-Mesure sont variables et pour BMatch ces résultats sont plus mauvais que les résultats des autres approches.

En général pour une hétérogénéité faible et une structure plate, un seuil de similarité élevé améliore les résultats de la précision mais pas forcément les résultats de rappel. Prenons l'exemple de la tâche 1, d'hétérogénéité faible, de structure plate et de faible écart. SF et COMA++ qui utilisent des seuils élevés, atteignent 100% de précision et 73% de rappel alors que LP4HM(Relaxé) atteint 78% de précision et 85% de rappel. Pour cette tâche, un seuil élevé maximise la précision alors que notre solution optimale, sans seuil, maximise le rappel et non pas la précision. Toutefois cette dernière dépasse les 70%, ce qui reste satisfaisant.

Pour les tâches 2, 3, 4, 5, 6, de structure imbriquée et de faible écart, nous remarquons que les résultats de précision, rappel et F-Mesure sont globalement plus faibles que les résultats de la tâche 1. Nous expliquons ceci par la difficulté que pose la structure aux outils d'appariement. Les résultats de ces tâches peuvent être analysés sur trois temps. D'abord, pour les tâches 2 et 6 les deux versions de notre approche affichent des précisions plus importantes que les précisions des tâches 3, 4 et 5. En fait, LP4HM(Relaxé) trouve les mêmes résultats que LP4HM, cela veut dire que la solution optimale ne contient pas de correspondances complexes. Le nombre réduit de correspondances retournées explique pourquoi les résultats de précision sont importants. Or, cinq utilisateurs ont proposé des correspondances complexes que notre approche n'a pas capturées, abaissant ainsi les résultats de rappel. Nous notons aussi que la tâche 2 nécessite des mesures linguistiques, l'utilisation du dictionnaire générique Wordnet a nettement contribué à obtenir des résultats plus importants que ceux des autres approches.

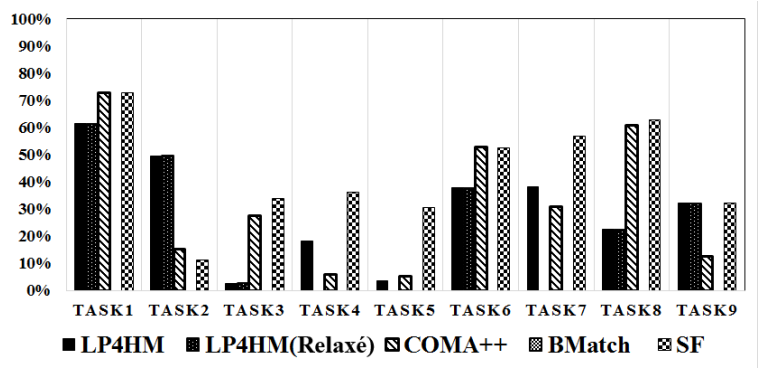
Les tâches 3 et 5 sont caractérisées par un nombre important de correspondances complexes proposées par tous les utilisateurs ainsi qu'une forte hétérogénéité. Les résultats de rappel, notamment pour la version relaxée, dépassent les 80% donc il y a eu une prise en compte des correspondances complexes. Alors que la précision ne dépasse pas les 50% ce qui est corrélé négativement avec les résultats d'accuracy.

Pour la tâche 4, nous remarquons qu'il y a un écart important entre le rappel et la précision notamment pour la version relaxée de notre approche. En comparant le rappel de SF 47%, de LP4HM 62% et de LP4HM(Relaxé) 80%, nous remarquons qu'avec un seuil égal à 1, SF ne capture que la moitié des correspondances pertinentes alors que LP4HM sans seuil capture 62% de correspondances de cardinalité 1 : 1. LP4HM(Relaxé) capture encore d'avantage de correspondances complexes que LP4HM. L'ordre des résultats de la précision pour ces trois approches est inversé, c'est à dire que le premier en rappel est le dernier en

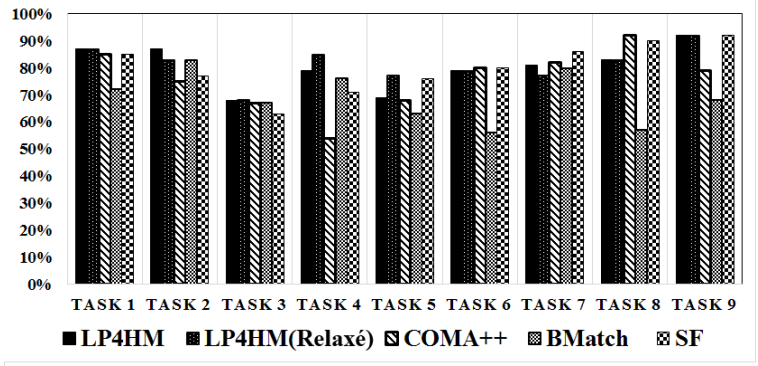
précision : SF 81%, LP4HM 59% et LP4HM(Relaxé) 30%. Ceci est dû au fait que pour une faible hétérogénéité, il est préférable d'utiliser un seuil de similarité élevé afin de réduire l'espace de recherche des correspondances ; c'est le cas de l'approche SF.

Les tâches 7 et 8 ont une forte hétérogénéité, une structure plate et un écart moyen. Les résultats de SF en précision et en rappel sont meilleurs que les résultats de notre approche. En effet, SF a utilisé les contraintes de clés primaires et de types de données qui existent dans le schéma relationnel.

Pour la tâche 9 de forte hétérogénéité, de structure plate et d'écart important, nous remarquons que les résultats de précision de notre approche et de l'approche SF sont de l'ordre de 90% alors que leurs résultats de rappel ne dépassent pas les 40%. En effet, le nombre de correspondances proposées par le système est minoré par le nombre d'éléments du plus petit schéma (qui est la moitié du nombre d'éléments du plus grand schéma). En outre, il est majoré par le nombre de correspondances complexes que le système peut retourner. En sachant que l'hétérogénéité est forte, il est difficile pour les utilisateurs d'identifier facilement les correspondances complexes.



(a) L'accuracy

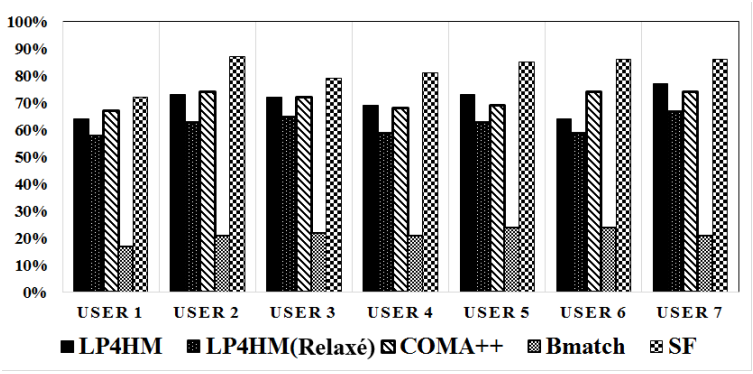


(b) Le HSR

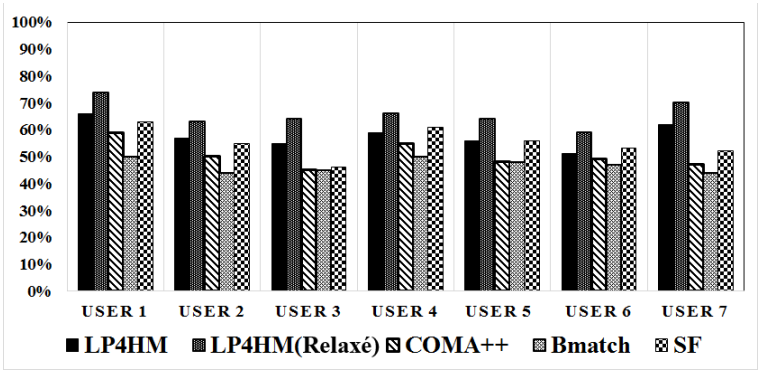
Figure V.23 — Les résultats d'accuracy et HSR par tâche pour la moyenne des utilisateurs

Nous passons aux résultats des mesures de l'effort épargné par les utilisateurs. Pour la moyenne des utilisateurs par tâche, nous remarquons que les résultats de HSR sont plus importants que les résultats d'accuracy, illustrés respectivement dans les Figures V.23(b) et V.23(a). Pour la moyenne des utilisateurs, qui ont été d'avis complètement différents sur les

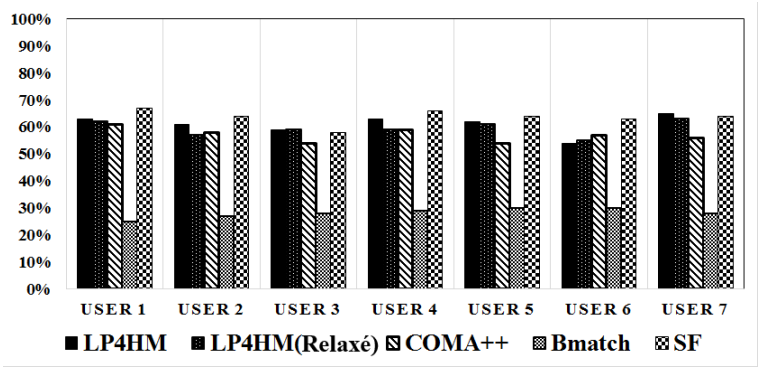
correspondances proposées par tâche, il y a un gain important en utilisant notre algorithme d'appariement.



(a) La précision



(b) Le rappel

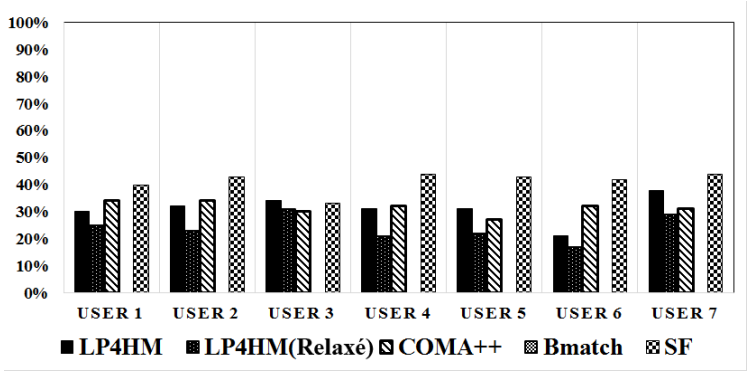


(c) Le F-Mesure

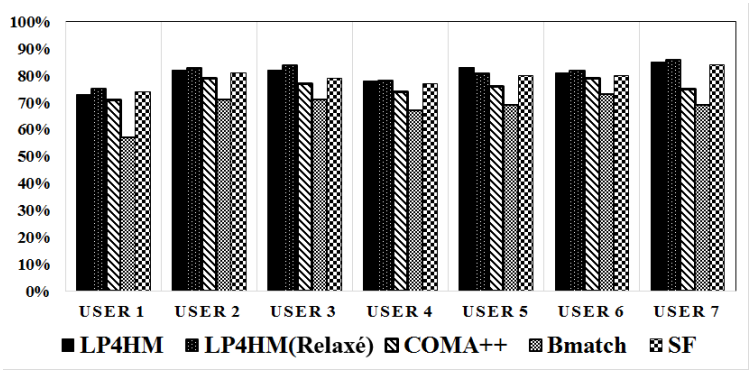
Figure V.24 — Les résultats de précision, rappel et F-Mesure par utilisateur pour la moyenne des tâches

En ce qui concerne les résultats de précision, rappel et F-Mesure pour la moyenne des tâches par utilisateur, illustrés respectivement dans les Figures V.24(a), V.24(b), V.24(c), nous constatons que pour tous les utilisateurs notre approche est meilleure que les autres approches en rappel. SF nous dépasse au niveau précision néanmoins avec un écart faible. Pour la F-Mesure notre approche est assez compétitive par rapport à l'approche SF. Concernant la mesure accuracy, Figure V.25(a), nous remarquons au niveau accuracy pour chaque utilisateur qu'il y a un gain par l'approche SF plus important que les autres. Pour le HSR,

Figure V.25(b), notre approche est légèrement meilleure que SF pour tous les utilisateurs.



(a) L'accuracy



(b) Le HSR

Figure V.25 — Les résultats d'accuracy et HSR par utilisateur pour la moyenne des tâches

3.2.2.4 Bilan

Le bilan que nous pouvons tirer sur les résultats d'évaluation de notre approche sur ce banc d'essais orienté utilisateurs est le suivant :

- Les deux variantes LP4HM et LP4HM(Relaxé) montrent expérimentalement sur différents utilisateurs et différentes tâches une bonne qualité d'appariement aussi bien sur des schémas de forte ou de faible hétérogénéité, de structure imbriquée ou plate. Les résultats encourageants pour une forte hétérogénéité et une structure imbriquée répondent à un problème majeur pour lequel notre algorithme d'appariement a été dédié initialement, à savoir d'intégrer des données ouvertes hiérarchiques (imbriquées) et provenant de différentes sources (hétérogènes).
- Nos résultats sont les meilleurs pour le rappel sans utilisation d'un seuil de similarité, ceci confirme l'efficacité de la recherche de solution optimale par la technique de programmation linéaire. Nous avons montré ainsi qu'il est possible de se passer du problème de configuration de seuil de similarité, ce qui donne un accès démocratisé aux outils d'appariements.
- Nos résultats de rappel forment un indicateur pertinent pour le cadre d'appariement holistique. En effet, si la précision était meilleure que le rappel alors les utilisateurs

devraient identifier plus de correspondances manquantes pour N schémas simultanément. Alors que si le rappel est plus élevé que la précision l'élimination des correspondances non-pertinentes est beaucoup plus facile.

3.2.3 Résultats d'évaluations sur un banc d'essais orienté schémas

Récemment, un nouveau banc d'essais orienté schémas a été proposé par [Duchateau et Bellahsene, 2014] pour évaluer les outils d'appariement par paires sur des schémas XML. Les résultats discutés dans cette section concernent les approches suivantes :

- Notre approche en stratégie A consiste à ne pas utiliser la contrainte de seuil de similarité. Nous avons expérimenté les deux versions :
  - LP4HM_A : variables de décision binaires + sans seuil de similarité ;
  - LP4HM_A(Relaxé) : variables de décision fractionnaires + sans seuil de similarité.
- Notre approche en stratégie B consiste à utiliser toutes les contraintes du modèle linéaire en utilisant un seuil de similarité pré-calculé lors de l'étape de préparation des données. Ce seuil est la médiane des maximums de chaque ligne dans une matrice de similarité. Nous avons expérimenté deux versions :
  - LP4HM_B : variables de décision binaires ;
  - LP4HM_B(Relaxé) : variables de décision fractionnaires.
- COMA++ [Aumueller *et al.*, 2005] a été expérimentée avec les trois stratégies (AllContext, FilteredContext et NoContext) et les meilleurs résultats ont été maintenus.
- Similariry Flooding (SF) [Melnik *et al.*, 2002] a été expérimentée avec un seuil de similarité égal à 1.
- YAM [Duchateau *et al.*, 2009] a été expérimenté en effectuant 200 exécutions [Duchateau et Bellahsene, 2014] par jeu de données pour pouvoir faire de l'apprentissage sur les seuils de similarité et les paramètres de configuration de l'outil d'appariement.

Ces approches sont comparées selon les mesures de qualité des correspondances et les mesures de l'effort épargné par les utilisateurs.

3.2.3.1 Description du banc d'essai

Le banc d'essais, proposé par [Duchateau et Bellahsene, 2014], est composé de dix jeux de données. Chaque jeu de données comporte deux schémas XML, un ensemble de correspondances (dont la majorité représente des correspondances simples) proposées manuellement par un expert ainsi que des schémas auxiliaires pour les outils qui font de l'apprentissage tel que YAM. Notre approche, COMA++ et SF utilisent uniquement les deux schémas proposés pour l'évaluation sans apprentissage alors que les résultats de YAM résultent d'un apprentissage sur les schémas auxiliaires et les schémas du jeu de données.

Nous reportons dans le tableau V.4 les caractéristiques des jeux de données qui ont été synthétisées par [Duchateau et Bellahsene, 2014]. Chaque banc d'essais représente un domaine donné comme suit :

- BETTING et FINANCE représentent des jeux de données extraits de sites web [Marie et Gal, 2008].

- BIOLOGY est représentatif du domaine de biologie avec deux collections UniProt [10] et GeneCards [11] décrivant des protéines.
- CURRENCY et SMS sont des schémas de services web ouverts [12] qui sont notamment utilisés pour la composition de services web.
- PERSON contient deux schémas décrivant les informations d'une personne que nous pouvons trouver dans des formulaires web.
- TRAVEL représente des schémas extraits de formulaires web de réservation de billets d'avions [Computer Science department, 2003].
- UNIV-COURS représente des schémas extraits de la collection Thalia [Hammer *et al.*, 2005] sur les cours offerts par des universités.
- UNIV-DEPT représente les départements des universités. Il a été largement utilisé dans la littérature [Doan *et al.*, 2003]

Tableau V.4 — Caractéristiques des schémas

	Hétérogénéité			Structure		Taille			Spec. Dom.
	faible	moyenne	forte	plate	imbriquée	petit	moyen	large	
PERSON	×				×	×			
TRAVEL		×		×		×			
UNIV-DEPT			×	×		×			
BETTING		×		×			×		
CURRENCY		×			×		×		
FINANCE		×		×			×		×
SMS		×			×		×		
UNV-COURS		×		×			×		
BIOLOGY		×			×			×	×
ORDER	×				×			×	

3.2.3.2 Résultats globaux

Le Tableau V.5 et la Figure V.26 présentent les résultats globaux de la précision, rappel, F-Mesure, accuracy et HSR pour les deux versions de notre approche dans les deux stratégies A et B ainsi que pour les approches COMA++, SF et YAM.

Tableau V.5 — Les résultats globaux de LP4HM, COMA++, SF and YAM

	Précision (%)	Rappel (%)	F-Mesure (%)	Accuracy (%)	HSR (%)
LP4HM_A	39	50	42	14	53
LP4HM_A(Relaxé)	24	68	34	0	59
LP4HM_B	59	50	52	27	52
LP4HM_B(Relaxé)	50	50	47	25	53
COMA++	66	36	43	28	34
SF	50	41	44	15	39
YAM	61	55	56	27	47

D'une part, nous pouvons remarquer que les résultats de rappel sont plus élevés que les résultats de précision pour notre approche en stratégie A, notamment pour

10. http ://www.uniprot.org/docs/uniprot.xsd
11. http ://www.geneontology.org
12. htpp ://free-web-services.com/

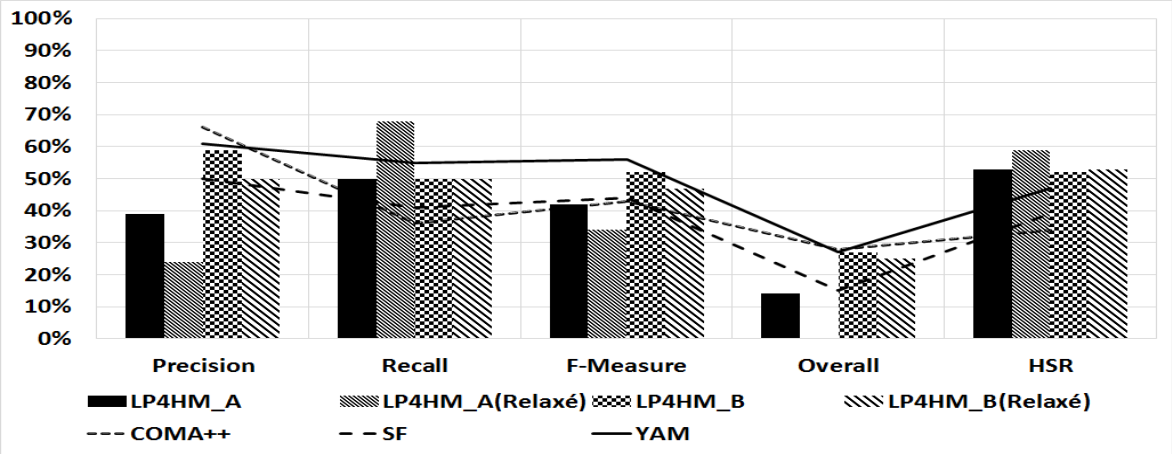


Figure V.26 — La représentation graphique des résultats globaux de LP4HM, COMA++, SF et YAM

LP4HM_A(Relaxé). D'autre part, nous observons que les résultats de précision sont plus élevés ou égaux aux résultats de rappel pour la stratégie B. Ceci concerne aussi les approches SF, COMA++ et YAM qui utilisent eux aussi des seuils de similarité. Nous en concluons ainsi que l'utilisation d'un seuil de similarité aide à améliorer les valeurs de la mesure de précision. Toutefois les valeurs de rappel s'avèrent sensibles face à la configuration du seuil. Par exemple, l'approche de YAM qui génère la meilleure configuration en fonction du jeu de données, réalise 55% de rappel ce qui est la meilleure valeur de rappel par rapport à toutes les approches qui utilisent un seuil de similarité. Notre approche LP4HM_A(Relaxé) indépendante du seuil réalise un rappel de 68%, mieux que le rappel de YAM. Ainsi, il est difficile de maximiser le rappel et la précision à la fois en visant uniquement sur les paramètres d'un outil d'appariement. La recherche de solution optimale, par la technique de programmation linéaire, semble tout à fait prometteuse pour pallier le problème de configuration de seuil de similarité.

Pour la F-mesure, les résultats de notre approche en stratégie B sont plus équilibrés que les résultats de la stratégie A. Ils sont aussi plus proches du résultat de l'approche YAM. Nous constatons que notre approche a battu l'approche SF qui était légèrement meilleure pour le banc d'essais orienté utilisateurs.

En ce qui concerne les mesures d'évaluation de l'effort utilisateur, nous observons que pour l'accuracy la stratégie B est meilleure que la stratégie A. Ceci rejoint l'affirmation sur la corrélation entre la précision et l'accuracy [Melnik *et al.*, 2002]. LP4HM_B et YAM ont la même valeur d'accuracy. Nous signalons aussi que SF, la meilleure approche dans l'autre banc d'essais orienté utilisateurs, est en dernière position en accuracy pour ce banc d'essais orienté schémas.

Par ailleurs, tous les résultats de HSR de notre approche sont meilleurs que les résultats des approches SF, COMA++ et YAM. Ceci montre clairement l'efficacité de notre approche à épargner les efforts des utilisateurs.

Ces résultats globaux montrent l'efficacité de notre méthode pour les deux bancs d'essais proposés dans la littérature. Nous constatons que notre approche, sur les deux bancs d'es-

sais, réussit à avoir les meilleurs résultats en rappel et en accuracy. Contrairement à SF dont les résultats de qualité sont moins bons pour ce banc d'essais, notre approche a maintenu sa compétitivité par rapport à l'approche YAM. En plus de sa compétitivité, nous rappelons que notre méthode est plus générique que les autres approches puisqu'elle s'applique sur $N \geq 2$ schémas en même temps.

3.2.3.3 Résultats détaillés

Nous examinons dans cette section les résultats détaillés en fonction de la taille des jeux de données.

Les jeux de données peu volumineux ($<$ 10 éléments)

Nous avons trois jeux de données peu volumineux : PERSON, TRAVEL, et UNIVERSITY DEPARTMENT (UNIV-DEPT). Le jeu de données PERSON est caractérisé par une faible hétérogénéité et une structure imbriquée. Le jeu de données TRAVEL est caractérisé par une hétérogénéité moyenne et une structure plate. Le troisième jeu de données UNIV-DEPT a une hétérogénéité forte et une structure plate.

Pour les jeux de données de faible ou de moyenne hétérogénéité (PERSON et TRAVEL), nous remarquons que les résultats de notre approche en stratégie B sont plus élevés que les résultats de notre approche en stratégie A. Alors que, pour le jeu de données de forte hétérogénéité (UNIV-DEPT), nous remarquons que les résultats de notre approche en stratégie A sont plus importants que les résultats de notre approche en stratégie B. Nous retenons de cette comparaison qu'en situation de forte hétérogénéité, il est préférable d'utiliser notre approche sans seuil de similarité. Ces résultats sont très positifs et prometteurs concernant l'intégration holistique de données ouvertes avec une forte hétérogénéité puisque l'utilisateur serait épargné du choix d'un seuil convenable à N schémas en même temps dans ce contexte difficile.

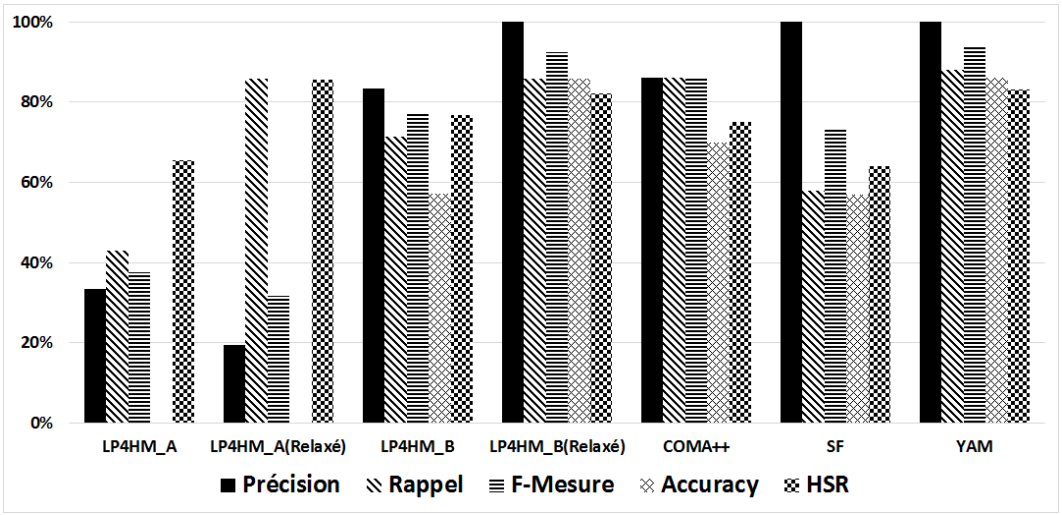


Figure V.27 — Les résultats du jeu de données PERSON

Pour le jeu de données PERSON, Figure V.27, YAM et LP4HM_B(Relaxé) ont les meilleurs résultats. Les résultats de LP4HM_B(Relaxé) sont obtenus après un seule exé-

cution tandis que les résultats de YAM sont obtenus après une moyenne de 200 exécutions. Ainsi pour la même qualité, l'utilisateur gagne du temps avec notre approche. LP4HM_B(Relaxé) et LP4HM_A (Relaxé) ont les mêmes résultats de rappel, par contre la précision de LP4HM_B(Relaxé) dépasse celle de LP4HM_A(Relaxé).

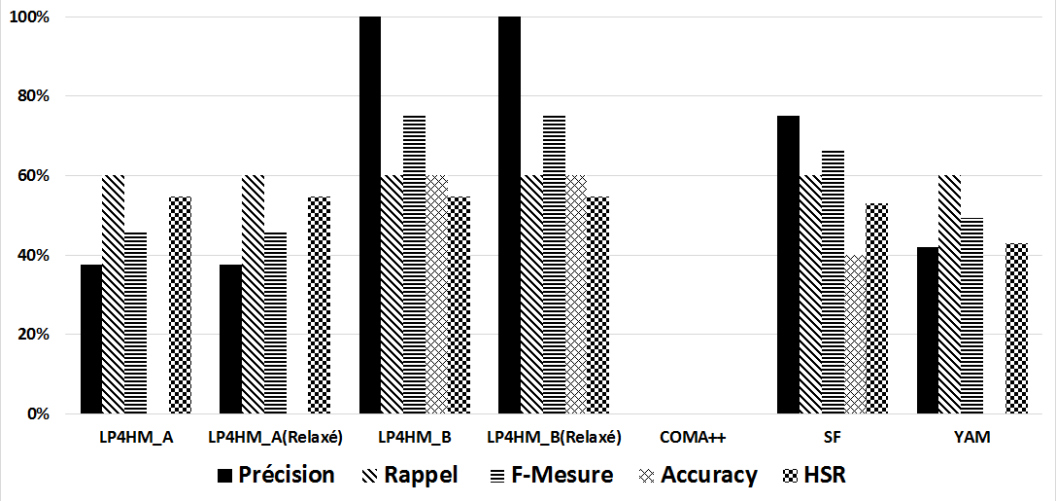


Figure V.28 — Les résultats du jeu de données TRAVEL

Pour le jeu de données TRAVEL, Figure V.28, nous observons que les deux versions de la stratégie B donnent les meilleurs résultats. Toutefois, nous signalons que les résultats de rappel et de HSR sont les mêmes pour les stratégies A ou B. Le seuil pré-calculé est pertinent puisqu'il ne touche pas au rappel. Aucun résultat de COMA++ n'intersecte les correspondances des experts ce qui explique ses résultats à zéro. L'approche SF et YAM sont à rappel égal mais SF est légèrement meilleure que YAM vu que sa précision est plus élevée.

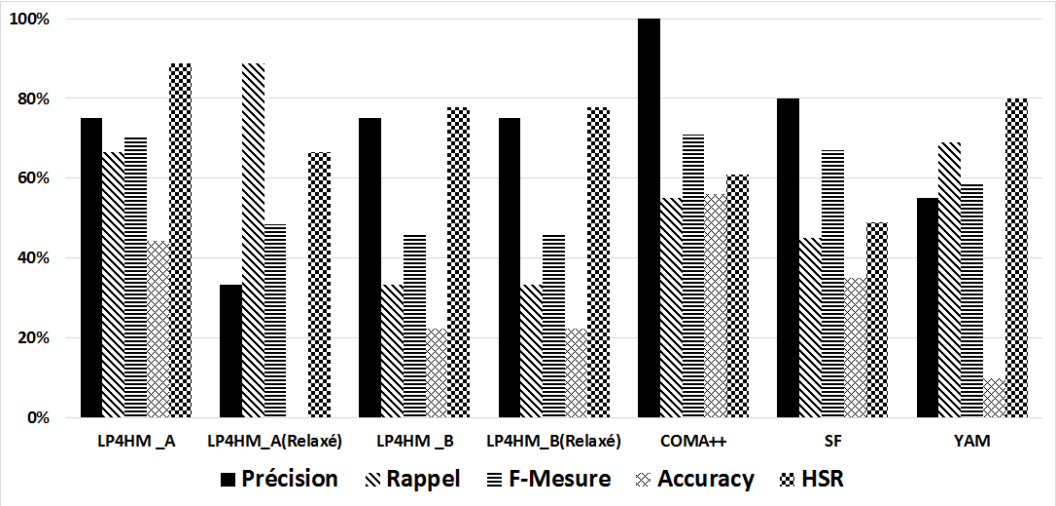


Figure V.29 — Les résultats du jeu de données UNIV-DEPT

Pour le jeu de données UNIV-DEPT, Figure V.29, nous observons que LP4HM_A(Relaxé) atteint les 90% de rappel contre 70% de rappel pour l'approche YAM. La non utilisation de seuil de similarité pour des jeux de données fortement hétérogènes a permis de capturer

les correspondances pertinentes qui ont une faible similarité. Pour une structure plate, le calcul de la solution du problème est plus influencé par l'optimisation de la similarité de la solution que par les contraintes structurelles.

Les jeux de données moyennement volumineux (entre 10 et 100 éléments)

Nous avons cinq jeux de données moyennement volumineux :

– BETTING, FINANCE et UNIV-COURS sont des jeux de données de moyenne hétérogénéité et de structure plate. En outre, FINANCE contient des labels spécifiques à un domaine.
– CURRENCY et SMS sont deux jeux de données de moyenne hétérogénéité et de structure imbriquée.

Concernant les jeux de données BETTING, FINANCE et UNIV-COURS, LP4HM_A(Relaxé) a les meilleurs scores de rappel par contre la stratégie LP4HM_B obtient les résultats les plus équilibrés. Lorsque la structure est plate, l'enjeu concerne plutôt l'optimisation de la similarité des correspondances et le choix de seuil de similarité. Nous constatons clairement que l'approche YAM réussit mieux sur les jeux de données BETTING, FINANCE et UNIV-COURS que les jeux de données CURRENCY et SMS. L'utilisation de notre approche avec un seuil donne des résultats plus équilibrés.

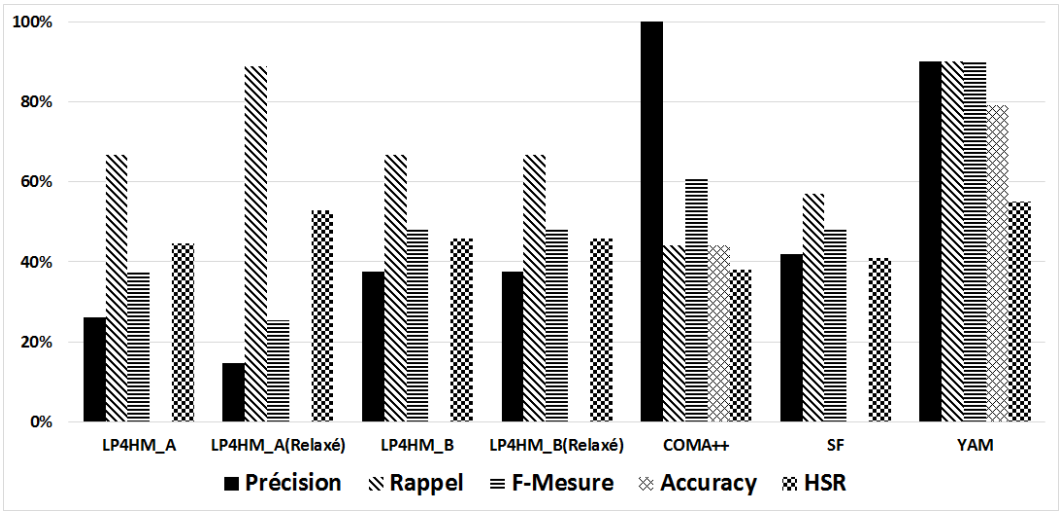


Figure V.30 — Les résultats du jeu de données BETTING

Pour le jeu de données BETTING, Figure V.30, nous remarquons que les résultats de rappel et de HSR de LP4HM_A(Relaxé) sont les mêmes que ceux de l'approche YAM. Toutefois, notre approche est moins précise que YAM. Les résultats de la stratégie B sont les mêmes que les résultats de l'approche SF et COMA++ (sauf pour la précision). La corrélation entre la précision et le rappel est clairement illustrée par les résultats de LP4HM_A(Relaxé) et COMA++. En fait, LP4HM_A(Relaxé) atteint 89% de rappel et la pire précision alors que COMA++ atteint 100% de précision et le plus mauvais rappel.

Pour le jeu de données FINANCE, Figure V.31, notre approche en stratégie A et B est est largement meilleure que les trois autres approches COMA++, SF et YAM, à l'exception de la précision de COMA++. Toutes les versions de notre approche ont la même valeur de rappel. La précision de la stratégie B est légèrement meilleure que la précision de la stratégie

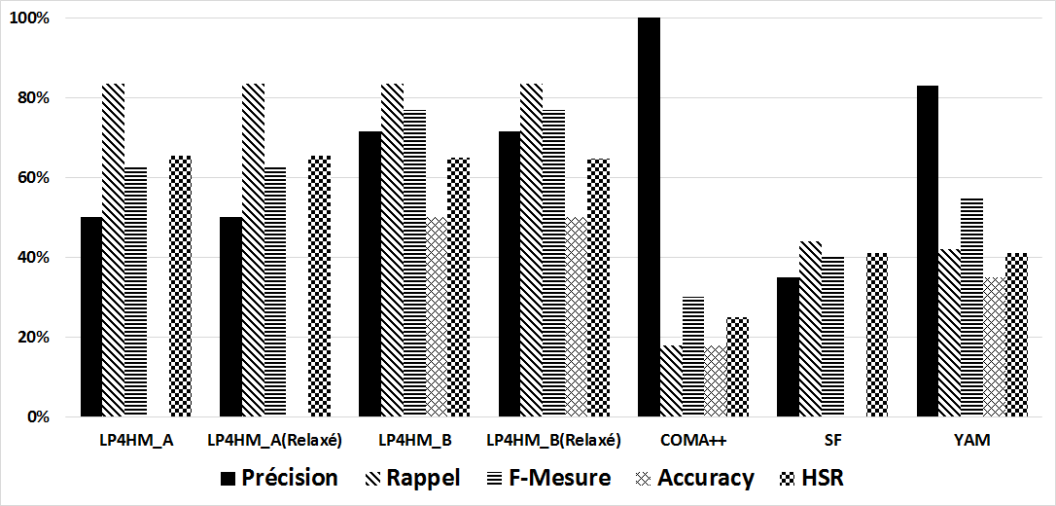


Figure V.31 — Les résultats du jeu de données FINANCE

A. En outre, nous rappelons que ce jeu de données utilise un vocabulaire de domaine qui n'a pas nécessité dans notre cas l'utilisation d'une ressource externe additionnelle à celle du dictionnaire Wordnet [13]. L'utilisation d'un dictionnaire générique pour le calcul de similarité a montré son efficacité sur certains domaines standards.

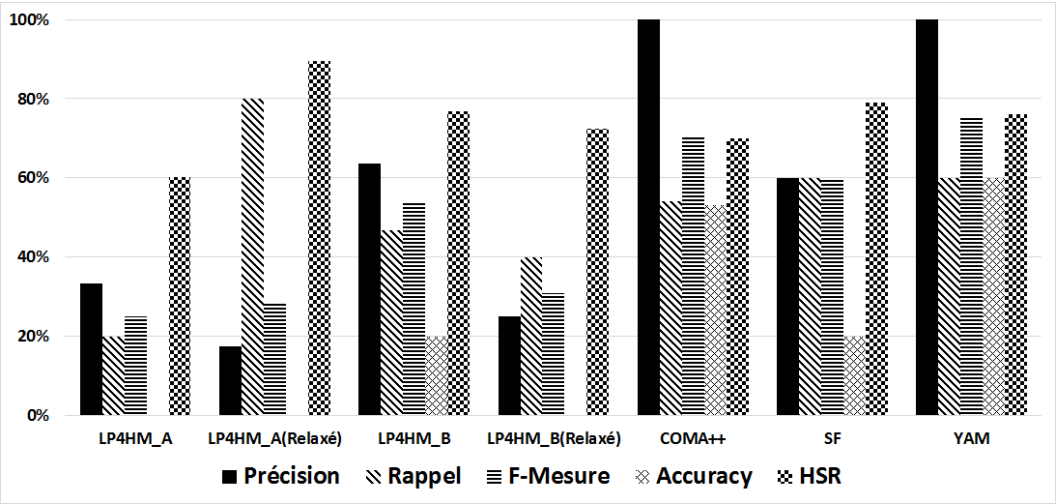


Figure V.32 — Les résultats du jeu de données UNIV-COURS

Pour le jeu de données UNIV-COURS, Figure V.32, la version relaxée de la stratégie A est meilleure que les autres versions de notre approche. Les utilisateurs gagnent au moins 90% pour le HSR ; LP4HM_A(Relaxé) trouve 80% de correspondances pertinentes. Même si la précision est très faible, il est toujours plus facile d'invalider les correspondances non-pertinentes que de rechercher celles qui le sont.

Concernant les deux jeux de données CURRENCY et SMS qui possèdent une structure imbriquée, la remarque générale est que les contraintes structurelles sont très efficaces. Ceci est reflété par des résultats meilleurs que ceux des autres approches. Globalement, la stra-

13. https ://wordnet.princeton.edu/

tégie A est plus efficace que la stratégie B. Il faudrait améliorer les seuils de similarité pour ces jeux de données afin d'améliorer les résultats de précision et de rappel.

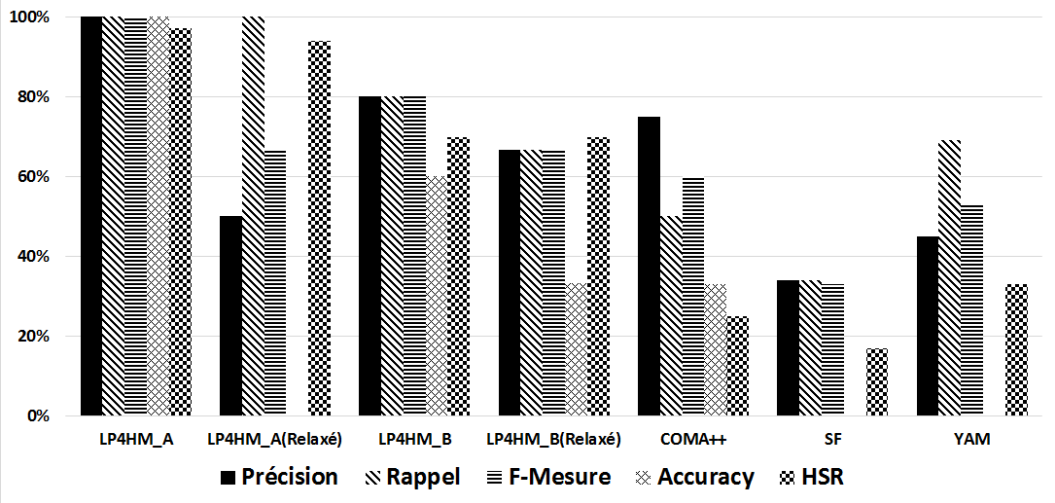


Figure V.33 — Les résultats du jeu de données CURRENCY

Pour le jeu de données CUREENCY, Figure V.33, nous observons que les résultats de notre approche dans les différentes versions et stratégies sont meilleurs que les résultats de COMA++, SF et YAM. La stratégie A est plus efficace que la stratégie B pour ce jeu de données. Étant donné que l'hétérogénéité est moyenne mais la structure est imbriquée, les contraintes structurelles montrent une efficacité pour trouver les bonnes correspondances. L'écart entre notre approche et les autres est important, nous arrivons à mieux gérer la structure par rapport aux autres.

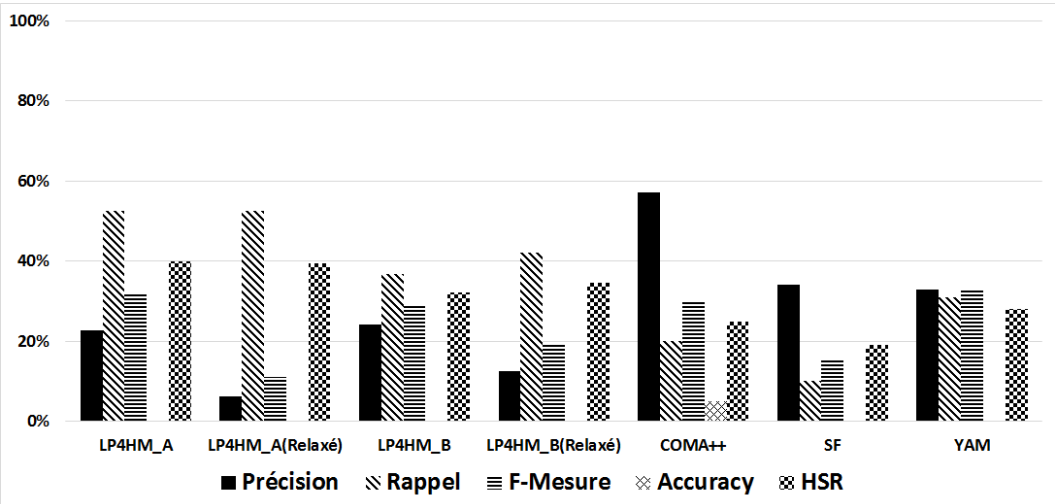


Figure V.34 — Les résultats du jeu de données SMS

Pour le jeu de données SMS, Figure V.34, nous remarquons que la stratégie A est aussi plus efficace que la stratégie B au niveau rappel et HSR. Nous avons perdu certaines correspondances pertinentes par l'utilisation du seuil de similarité dans la stratégie B. Si nous écartons les valeurs de la précision, notre approche est meilleure que COMA++, SF and

YAM

Les jeux de données de large volume (> 100 éléments)

Nous avons deux jeux de données de large volume : ORDER et BIOLOGY. Ces derniers ont des structures imbriquées. BIOLOGY utilise un vocabulaire moyennement hétérogène spécifique au domaine de la biologie et spécifiquement sur les protéines. ORDER a une faible hétérogénéité.

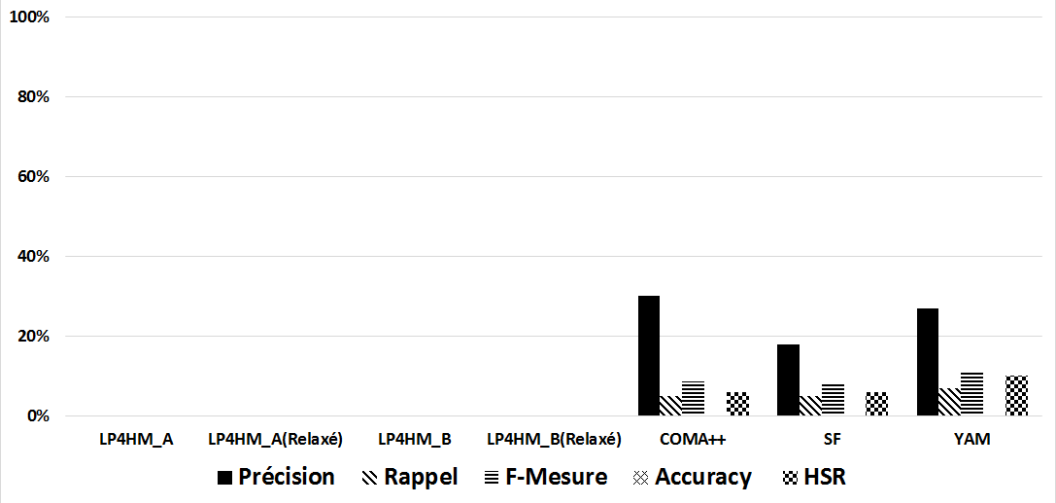


Figure V.35 — Les résultats du jeu de données BIOLOGY

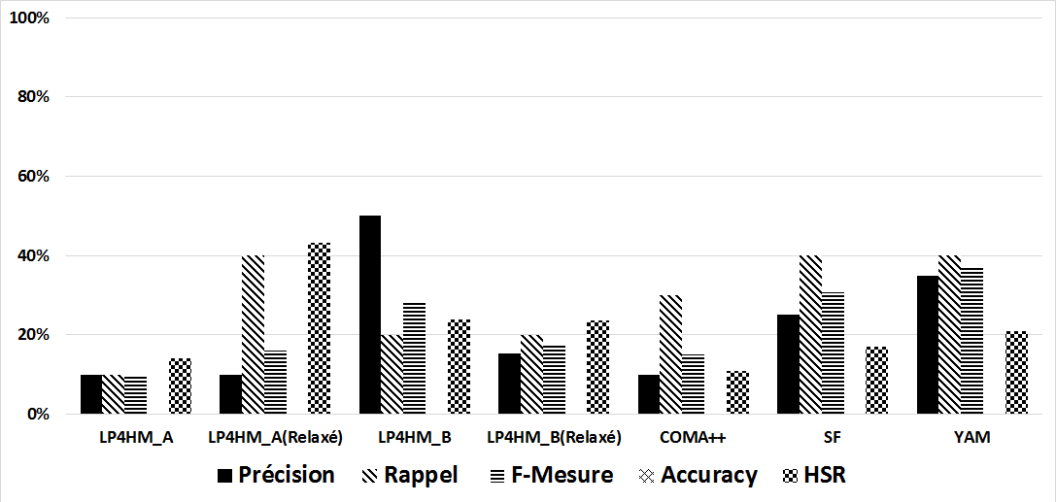


Figure V.36 — Les résultats du jeu de données ORDER

Globalement, il n'y a aucune approche qui réussit ces jeux de données d'après les Figures V.35 et V.36. Pour le jeu de données BIOLOGY, les différentes versions de notre approche ont des valeurs à zéro. Ces résultats sont décevants même si les autres approches n'ont réussi à capturer que 5% des correspondances pertinentes. Cela mérité d'être dit que les correspondances des experts sur les jeux de données BIOLOGY et ORDER ne sont pas exhaustifs, il y a une partie infiniment petite d'éléments qui a été examinée. Donc dans le grand espace de solutions possibles s'il y a juste un ensemble infime de solutions, le pourcentage

de capturer ces solutions devient faible. En plus, pour notre approche les correspondances retournées faisaient partie des parties non-examinées par l'expert. Pour le jeu de données ORDER, nous avons le même constat sur les mauvais résultats de qualité pour toutes les approches, les meilleurs rappel sont à 40%. Néanmoins, ces résultats sont meilleurs que pour le jeu de données BIOLOGY.

3.2.3.4 Bilan

Nous synthétisons les résultats de notre approche pour ce banc d'essais orienté schémas comme suit :

- Pour les jeux de données de petite taille, nous différencions deux cas : (1) pour une faible ou moyenne hétérogénéité, notre approche doit être utilisée de préférence avec un seuil de similarité pour améliorer la précision et par conséquent l'accuracy et la F-Mesure ; (2) pour une forte hétérogénéité, nous recommandons notre approche en stratégie A sans seuil de similarité pour capturer les correspondances pertinentes de faible distance de similarité.
- Pour les jeux de données de taille moyenne et de structure imbriquée, les contraintes structurelles montrent leurs efficacités à trouver des résultats pertinents. Nos résultats sont nettement meilleurs que les approches COMA++, SF et YAM. La stratégie A sans seuil de similarité suffirait pour répondre à des jeux de données de la même catégorie. Ces résultats valident la capacité de notre approche à résoudre efficacement la tâche d'intégration de données ouvertes de taille moyenne ayant une structure hiérarchique imbriquée.
- Pour les jeux de données de grande taille, les résultats de ce banc d'essais ne sont pas suffisants pour tirer des conclusions sur l'efficacité de notre approche ni des autres approches puisque les correspondances proposées par l'expert manquent énormément d'exhaustivité ce qui biaise les résultats.

Les résultats de ce banc d'essais nous permettent de compléter l'analyse du comportement de notre approche face à des schémas de différentes caractéristiques. Les problèmes d'hétérogénéité et de structure des schémas sont accentués dans ce banc d'essais par rapport à celui que nous avons présenté dans la section précédente. Nous avons comparé notre approche sans et avec seuil de similarité, ce qui nous permet d'identifier dans quel cas il faudrait choisir sans ou avec le seuil en fonction des caractéristiques du jeu de données.

3.3 Évaluation de la performance de l'appariement holistique sur des données ouvertes tabulaires

Nous avons évalué la performance de l'appariement holistique en étudiant le temps de résolution de LP4HM en fonction de la taille (nombre de noeuds) de plusieurs graphes de données ouvertes tabulaires. En tenant en compte du fait que l'appariement ne se fait que sur les données structurelles, nous constatons que la taille de chaque graphe en entrée n'est pas très importante. En effet, les données structurelles représentent à peu près 12% de la totalité des données dans un fichier tabulaire. Pour cela, nous jugeons que la taille totale des graphes pris dans cette expérimentation est significative pour évaluer l'appariement

holistique. Les sources de données ouvertes tabulaires qui ont été utilisées sont celles du scénario d'étude présenté dans la section 2.1.

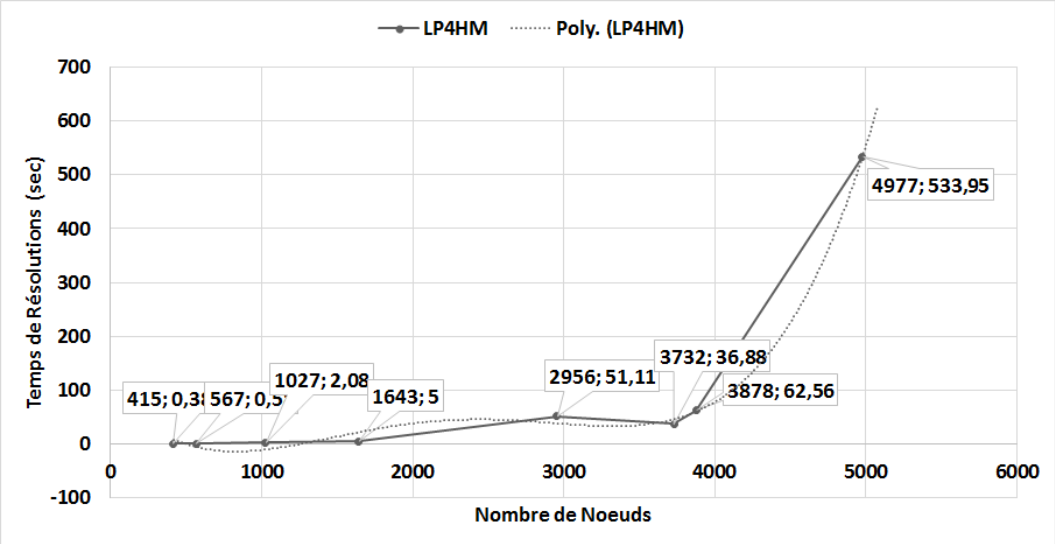


Figure V.37 — Le temps de résolution en fonction du nombre de noeuds des graphes

La Figure V.37 illustre le temps de résolution en fonction de la taille des graphes. La courbe bleue continue relie les valeurs mesurées et la courbe rouge discontinue est une courbe de tendance (trendline). Nous pouvons observer que le temps de résolution a une tendance polynomiale de l'ordre de O(n^4). Nous pouvons constater que pour 415 noeuds dans 9 graphes, notre modèle trouve une solution optimale en 0,38 sec. De plus, si la taille des graphes augmente jusqu'à 4977 noeuds dans 46 graphes alors LP4HM trouve une solution dans 533,95 sec $\rightsquigarrow$ 9,23 min, ce qui est un temps de résolution raisonnable. D'autant plus si nous comparons cette automatisation par rapport au temps qui serait nécessaire à un concepteur pour effectuer cette tâche manuellement. Il faut aussi noter que notre approche fournit une solution unique globalement optimale (sur les 46 graphes) ce qui là encore facilite le travail d'intégration en évitant la comparaison de plusieurs solutions localement optimales comme par exemple COMA++ ou SF. Avec les approches par paires, il faudrait itérer l'appariement du premier et deuxième schémas puis leur résultat avec le troisième schéma, ainsi de suite jusqu'à atteindre le 46ème schéma. C'est très long comme processus. De plus, l'ordre avec lequel l'appariement par paires est fait n'aboutit pas au même résultat.

4 Conclusion

Ce dernier chapitre a été consacré au prototypage et aux évaluations expérimentales de notre démarche. A chaque phase de cette démarche, un module du prototype a été implémenté pour valider les algorithmes proposés. Par ailleurs, nous avons évalué les deux premières phases de notre démarche. Nous avons engagé une évaluation comparative sur deux bancs d'essais orientés utilisateurs et schémas de notre algorithme d'intégration afin de positionner la qualité de notre algorithme par rapport aux travaux référencés dans la littérature. Les résultats d'évaluation montrent une bonne qualité des correspondances gé-

nérées par notre approche. Les meilleurs résultats sont dans le contexte des jeux de données de forte hétérogénéité et de structure imbriquée. Pour le banc d'essais orienté utilisateurs, notre approche est celle qui épargne le plus d'effort à tous les utilisateurs. L'appariement holistique de plusieurs graphes de taille importante se fait en quelques minutes (en temps polynomial selon la taille des graphes). Ainsi, LP4HM résout efficacement et rapidement l'appariement holistique de plusieurs graphes hiérarchiques.

Plusieurs extensions sont prévues pour le prototype et pour l'évaluation. Pour le prototype, nous envisageons de mettre en place les fonctionnalités nécessaires pour la gestion des graphes RDF. Une mise en ligne publique de ce prototype nous permettra aussi de faire une évaluation de la qualité globale de notre démarche. Elle nous permettra aussi de mieux mesurer les difficultés que peuvent rencontrer les utilisateurs de profils variés avec notre démarche.

VI Conclusion et perspectives

1 Conclusion générale

Les travaux présentés dans ce manuscrit concernent l'intégration des données ouvertes (Open Data) tabulaires dans les systèmes d'information décisionnels (SID). Dans de tels systèmes, l'intégration des données est assurée par des processus d'Extraction-Transformation-Chargement dits processus ETL. Ces processus sont remis en question vu leur lenteur, leur complexité et leur manque d'automatisme. Ils répondent mal aux caractéristiques des données ouvertes mais aussi au besoin du Self-Service BI. Les données ouvertes tabulaires sont caractérisées par une hétérogénéité sémantique et structurelle, une absence de schémas et des problèmes de qualité. Le Self-Service BI [Abello *et al.*, 2013] concerne la possibilité de donner la main à des utilisateurs non-experts pour intégrer et analyser eux mêmes les données.

Afin de répondre à ces problématiques, nous avons proposé une nouvelle démarche ETL, basée sur les graphes, pour automatiser le plus possible l'entreposage des Open Data tabulaires. Notre démarche est composée de trois étapes :

- La première étape permet la découverte et l'extraction automatique d'un schéma d'Open Data tabulaires, sous forme d'un graphe. Nous avons proposé un workflow d'activités, où chaque activité détecte l'emplacement du composant du tableau puis l'annote en s'appuyant sur un modèle de tableau. Le workflow aboutit à une distinction entre les données structurelles et les données numériques. Nous avons également proposé différentes stratégies de classification hiérarchique pour préparer à l'issue de cette étape une structure hiérarchique des tableaux servant à la définition du schéma multidimensionnel.
- La deuxième étape permet l'intégration automatique de plusieurs graphes de tableaux. Nous avons proposé d'exploiter l'optimisation combinatoire, notamment la technique de programmation linéaire, pour résoudre le problème d'appariement holistique. Ceci nous a permis de produire une solution unique et optimale comportant les correspondances de plusieurs graphes. A l'issue de cette résolution, nous avons pu intégrer automatiquement et en temps polynomial plusieurs graphes de tableaux.
- La troisième étape permet la définition d'un schéma multidimensionnel à partir d'un graphe intégré. Nous avons proposé un processus progressif dans lequel l'utilisateur assure la définition des composants multidimensionnels à partir du graphe intégré. Nous avons aussi exploré la possibilité d'enrichir le graphe intégré avec des anno-

tations multidimensionnelles. Cette dernière étape a supporté l'hypothèse de la divergence des utilisateurs à propos la matérialisation des données à travers les deux solutions proposées.

En guise de validation, nous avons présenté un prototype de cette approche composé de trois modules qui correspondent à chaque étape. Nous avons également expérimenté notre approche. En particulier, nous avons évalué notre proposition d'intégration sur deux bancs d'essais référencés dans la littérature. Nos résultats sont très satisfaisants sur les deux bancs d'essais, notamment en ce qui concerne l'appariement des schémas de forte hétérogénéité et de structures imbriquées. Les résultats que nous avons obtenus sans l'utilisation d'un seuil de similarité sont également satisfaisants. Ainsi notre approche est aussi simple à utiliser pour les non-experts.

2 Perspectives

Nous envisageons de poursuivre ce travail par les perspectives suivantes :

De l'ETL vers l'ETQ. Nous planifions à très court terme de mettre en place les fonctionnalités de manipulation et d'interrogation des graphes RDF afin de transformer notre approche ETL en une approche ETQ [Abello *et al.*, 2015] pour un contexte combiné Self-Service BI et web sémantique. Cette approche ETQ serait sous la forme d'une application web ce qui nous permettra de mesurer l'efficacité et les difficultés de telles approches.

Appariement d'ontologies. Nous envisageons d'étendre le programme linéaire pour l'appariement des ontologies de n'importe quelles structures. Nous rappelons qu'actuellement, notre solution s'applique sur les ontologies de type taxonomique. Nous mettrons en place de nouvelles variables et de nouvelles contraintes pour exprimer les contraintes logiques entre les propriétés (les relations dans une ontologie). Nous allons nous inspirer des contraintes d'incohérences proposées par [Meilicke, 2011]. Ces contraintes ont été proposées dans l'objectif de réparer les appariements erronés à la fin du processus d'appariement. Nous envisageons la transformation de ces contraintes en contraintes linéaires compatibles avec notre modèle.

Production de données ouvertes liées. Nous envisageons de spécialiser l'étape de détection et de reconnaissance des tableaux dans la production de données ouvertes liées. Nous intégrerons dans le module ODET la possibilité de choisir le modèle de tableau proposé par le W3C. Étant donné qu'il y a une compatibilité entre le modèle que nous avons proposé et ce dernier, il suffit juste de modifier les annotations utilisées en fonction du modèle. Nous relâcherons également la restriction d'automatiser cette étape au profit d'un gain au niveau sémantique et expressivité du contenu des tableaux. Nous pourrions intégrer des ressources ontologiques apportant plus de précision pour les activités de détection spatio-temporelle et celles de détection de formules. Nous pourrions aussi profiter des avancées dans le domaine du traitement des langages naturels tel que la reconnaissance d'entités nommées pour améliorer nos algorithmes d'annotation des labels qui entourent le tableau (titres, notes...). Nous permettrons aux utilisateurs de choisir parmi les vocabulaires des données liées (LOV) et de créer des ancres vers le nuage des données ouvertes liées (LOD). L'usage d'une telle plateforme serait dédiée à des utilisateurs connaissant le web sémantique.

Un banc d'essais pour l'intégration holistique. Nous nous sommes heurtés lors de l'expérimentation de notre approche au manque de banc d'essais de référence pour la comparaison des approches holistiques. Ainsi, nous envisageons de proposer deux bancs d'essais répondant à ce besoin. Le premier banc d'essais sera réalisé par différents utilisateurs de l'université Toulousaine qui devront retourner les résultats d'appariement de plusieurs schémas XML en même temps. Le deuxième banc d'essais sera réalisé par des experts en ontologies sur la problématique d'appariement holistique de plusieurs ontologies.

Une intégration à large échelle. Avec l'émergence des plateformes de calculs parallèles pour la gestion des données massives (Big Data), il est tout à fait envisageable de les exploiter pour gérer l'intégration à large échelle. Même si notre approche montre un temps de calcul polynomial et très rapide pour des schémas de l'ordre de 5000 noeuds. Nous estimons que pour des millions de noeuds malgré le fait que notre algorithme soit polynomial, le temps de calcul peut devenir très important. L'exploitation du calcul parallèle et de nouvelles stratégies de partitionnement permettrait de répondre à ce problème.

Bibliographie

[Abdelhédi, 2014] ABDELHÉDI, F. (2014). *Conception assistée d'entrepôts de données et de documents XML pour l'analyse OLAP*. Thése de doctorat, Université Toulouse 1 Capitole.

[Abello, 2002] ABELLO, A. (2002). *Yam2 : a multidimensional conceptual model*. Thése de doctorat, Université Polytechnique de Catalogne, Espagne.

[Abello *et al.*, 2013] ABELLO, A., DARMONT, J., ETCHEVERRY, L., GOLFARELLI, M., MAZON, J., NAUMANN, F., PEDERSEN, T., RIZZI, S., TRUJILLO, J., VASSILIADIS, P. et VOSSEN, G. (2013). Fusion cubes : Towards Self-Service business intelligence. *International Journal of Data Warehousing and Mining*, 9(2):66–88.

[Abello *et al.*, 2015] ABELLO, A., ROMERO, O., BACH PEDERSEN, T., BERLANGA, R., NEBOT, V., ARAMBURU, M. et SIMITSIS, A. (2015). Using semantic web technologies for exploratory olap : A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 27(2):571–588.

[Abello *et al.*, 2006] ABELLO, A., SAMOS, J. et SALTOR, F. (2006). Yam2 : a multidimensional conceptual model extending {UML}. *Information Systems*, 31(6):541 – 567.

[Agrawal *et al.*, 1997] AGRAWAL, R., GUPTA, A. et SARAWAGI, S. (1997). Modeling multidimensional databases. *In Proceedings of the Thirteenth International Conference on Data Engineering*, ICDE '97, pages 232–243.

[Agreste *et al.*, 2014] AGRESTE, S., MEO, P. D., FERRARA, E. et URSINO, D. (2014). {XML} matchers : Approaches and challenges. *Knowledge-Based Systems*, 66:190 – 209.

[Almohamad et Duffuaa, 1993] ALMOHAMAD, H. A. et DUFFUAA, S. (1993). A linear programming approach for the weighted graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell*, pages 522–525.

[Ananthakrishna *et al.*, 2002] ANANTHAKRISHNA, R., CHAUDHURI, S. et GANTI, V. (2002). Eliminating fuzzy duplicates in data warehouses. *In Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 586–597. VLDB Endowment.

[Annoni, 2003] ANNONI, E. (2003). Conception et développement d'un langage assertionnel pour les bases de données multidimensionnelles.

[Annoni *et al.*, 2006a] ANNONI, E., RAVAT, F. et TESTE, O. (2006a). Traitements à l'origine des systèmes d'information décisionnels. catégorisation et formalisation des traitements à l'origine des SID. *Ingénierie des Systèmes d'Information*, 11(6):115–143.

[Annoni *et al.*, 2006b] ANNONI, E., RAVAT, F., TESTE, O. et ZURFLUH, G. (2006b). Towards multidimensional requirement design. *In Data Warehousing and Knowledge Discovery, 8th*

International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings., pages 75–84.

[Atigui, 2013] ATIGUI, F. (2013). *Approche dirigée par les modèles pour l'implantation et la réduction d'entrepôts de données*. Thése de doctorat, Université Toulouse 1 Capitole.

[Atigui *et al.*, 2012] ATIGUI, F., RAVAT, F., TESTE, O. et ZURFLUH, G. (2012). Using OCL for automatically producing multidimensional models and ETL processes. *In Data Warehousing and Knowledge Discovery - 14th International Conference, DaWaK 2012, Vienna, Austria, September 3-6, 2012. Proceedings*, pages 42–53.

[Aumueller *et al.*, 2005] AUMUELLER, D., DO, H.-H., MASSMANN, S. et RAHM, E. (2005). Schema and ontology matching with coma++. *In SIGMOD '05*, pages 906–908.

[Azirani *et al.*, 2015] AZIRANI, E. A., GOASDOUÉ, F., MANOLESCU, I. et ROATIS, A. (2015). Efficient OLAP operations for RDF analytics. *In 31st IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2015, Seoul, South Korea, April 13-17, 2015*, pages 71–76.

[Balinski, 1965] BALINSKI, M. (1965). Integer programming : Methods, uses, computation. *In Management Science A 12*, pages 253–313. Springer Berlin Heidelberg.

[Banerjee et Pedersen, 2002] BANERJEE, S. et PEDERSEN, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. *In Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, UK. Springer-Verlag.

[Benharkat *et al.*, 2007] BENHARKAT, A., RIFAIEH, R., SELLAMI, S., BOUKHEBOUZE, M. et AMGHAR, Y. (2007). PLASMA : A platform for schema matching and management. *IBIS*, 5:9–20.

[Benjelloun *et al.*, 2009] BENJELLOUN, O., GARCIA-MOLINA, H., MENESTRINA, D., SU, Q., WHANG, S. et WIDOM, J. (2009). Swoosh : a generic approach to entity resolution. *The VLDB Journal*, 18(1):255–276.

[Bergamaschi *et al.*, 2011] BERGAMASCHI, S., GUERRA, F., ORSINI, M., SARTORI, C. et VINCINI, M. (2011). A semantic approach to {ETL} technologies. *Data and Knowledge Engineering*, 70(8):717 – 731.

[Bergamaschi *et al.*, 2007] BERGAMASCHI, S., SARTORI, C., GUERRA, F. et ORSINI, M. (2007). Extracting relevant attribute values for improved search. *IEEE Internet Computing*, 11(5): 26–35.

[Bernstein *et al.*, 2011] BERNSTEIN, P. A., MADHAVAN, J. et RAHM, E. (2011). Generic schema matching, ten years later. *PVLDB*, 4(11):695–701.

[Berro *et al.*, 2013] BERRO, A., MEGDICHE, I. et TESTE, O. (2013). Vers l'intégration multidimensionnelle d'open data dans les entrepôts de données. *In Actes des 9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA 2013, Blois, France, Juin 13-14, 2013*, pages 95–104.

[Berro *et al.*, 2014a] BERRO, A., MEGDICHE, I. et TESTE, O. (2014a). A content-driven ETL processes for open data. *In New Trends in Database and Information Systems II, Selected papers of the 18th East European Conference on Advances in Databases and Information Systems, ADBIS'14*, volume 312, pages 29–40. Springer International Publishing.

[Berro *et al.*, 2014b] BERRO, A., MEGDICHE, I. et TESTE, O. (2014b). Transformer les open data brutes en graphes enrichis en vue d'une intégration dans les systèmes OLAP. *In Actes du XXXIIème Congrès INFORSID, Lyon, France, 20-23 Mai 2014.*, pages 95–111.

[Berro *et al.*, 2015a] BERRO, A., MEGDICHE, I. et TESTE, O. (2015a). Graph-based ETL processes for warehousing statistical open data. *In ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015*, pages 271–278.

[Berro *et al.*, 2015b] BERRO, A., MEGDICHE, I. et TESTE, O. (2015b). Holistic statistical open data integration based on integer linear programming. *In 9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, Athens, Greece, May 13-15, 2015*, pages 468–479.

[Berro *et al.*, 2015c] BERRO, A., MEGDICHE, I. et TESTE, O. (2015c). Intégration holistique des graphes basée sur la programmation linéaire pour l'entreposage des open data. *In Actes des 11es journées francophones sur les Entrepôts de Donnés et l'Analyse en Ligne, EDA 2015, Bruxelles, Belgique, 2-3 avril 2015*, pages 113–128.

[Berro *et al.*, 2015d] BERRO, A., MEGDICHE, I. et TESTE, O. (2015d). A linear program for holistic matching : Assessment on schema matching benchmark. *In Database and Expert Systems Applications - 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part II*, pages 383–398.

[Berti-Equille et Moussouni, 2005] BERTI-EQUILLE, L. et MOUSSOUNI, F. (2005). Quality-aware integration and warehousing of genomic data. *In Proceedings of the 2005 International Conference on Information Quality (MIT IQ Conference), Sponsored by Lockheed Martin, MIT, Cambridge, MA, USA, November 10-12, 2006*.

[Birkhoff, 1967] BIRKHOFF, G. (1967). Lattice theory. *In Colloquium Publications*, volume 25. Amer. Math. Soc., 3. édition.

[Biskup et Embley, 2003] BISKUP, J. et EMBLEY, D. W. (2003). Extracting information from heterogeneous information sources using ontologically specified target views. *Inf. Syst.*, 28(3):169–212.

[Bizer *et al.*, 2009] BIZER, C., HEATH, T. et BERNERS-LEE, T. (2009). Linked data the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1 –22.

[Bonifati *et al.*, 2001] BONIFATI, A., CATTANEO, F., CERI, S., FUGGETTA, A. et PARABOSCHI, S. (2001). Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.*, 10(4):452–483.

[Buche *et al.*, 2013] BUCHE, P., DIBIE-BARTHÉLEMY, J., IBANESCU, L. et SOLER, L. (2013). Fuzzy web data tables integration guided by an ontological and terminological resource. *IEEE Trans. Knowl. Data Eng.*, 25(4):805–819.

[Burke et Kendall, 2005] BURKE, E. K. et KENDALL, G., éditeurs (2005). *Search methodologies : introductory tutorials in optimization and decision support techniques*. Springer, New York.

[Buzydlowski *et al.*, 1998] BUZYDLOWSKI, J. W., SONG, I.-Y. et HASSELL, L. (1998). A framework for object-oriented on-line analytic processing. *In Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*, DOLAP '98, pages 10–15.

[Cabanac *et al.*, 2007] CABANAC, G., CHEVALIER, M., RAVAT, F. et TESTE, O. (2007). An annotation management system for multidimensional databases. *In Data Warehousing and Knowledge Discovery, 9th International Conference, DaWaK 2007, Regensburg, Germany, September 3-7, 2007, Proceedings*, pages 89–98.

[Cabibbo et Torlone, 2000] CABIBBO, L. et TORLONE, R. (2000). The design and development of a logical system for OLAP. *In Data Warehousing and Knowledge Discovery, Second International Conference, DaWaK 2000, London, UK, September 4-6, 2000, Proceedings*, pages 1–10.

[Cafarella *et al.*, 2008a] CAFARELLA, M. J., HALEVY, A., WANG, D. Z., WU, E. et ZHANG, Y. (2008a). Webtables : Exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549.

[Cafarella *et al.*, 2008b] CAFARELLA, M. J., HALEVY, A. Y., ZHANG, Y., WANG, D. Z. et WU, E. (2008b). Uncovering the relational web. *In 11th International Workshop on the Web and Databases, WebDB 2008, Vancouver, BC, Canada, June 13, 2008*.

[Calvanese *et al.*, 2001] CALVANESE, D., DE GIACOMO, G., LENZERINI, M., NARDI, D. et ROSATI, R. (2001). Data integration in data warehousing. *Int. J. Cooperative Inf. Syst.*, 10(3):237–271.

[Castanier *et al.*, 2013] CASTANIER, E., COLETTA, R., VALDURIEZ, P. et FRISCH, C. (2013). Websmatch : a tool for open data. *In Proceedings of the 2nd International Workshop on Open Data, WOD 2013, Paris, France, June 3, 2013*, pages 10 :1–10 :2.

[Chaudhuri et Dayal, 1997] CHAUDHURI, S. et DAYAL, U. (1997). An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26(1):65–74.

[Chaudhuri *et al.*, 2011] CHAUDHURI, S., DAYAL, U. et NARASAYYA, V. (2011). An overview of business intelligence technology. *Commun. ACM*, 54(8):88–98.

[Chen *et al.*, 2000] CHEN, H.-H., TSAI, S.-C. et TSAI, J.-H. (2000). Mining tables from large scale html texts. *In Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 166–172, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Chevalier *et al.*, 2015a] CHEVALIER, M., MALKI, M. E., KOPLIKU, A., TESTE, O. et TOURNIER, R. (2015a). Benchmark for OLAP on nosql technologies comparing nosql multidimensional data warehousing solutions. *In 9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, Athens, Greece, May 13-15, 2015*, pages 480–485.

[Chevalier *et al.*, 2015b] CHEVALIER, M., MALKI, M. E., KOPLIKU, A., TESTE, O. et TOURNIER, R. (2015b). Implementing multidimensional data warehouses into nosql. *In ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015*, pages 172–183.

[Chukmol *et al.*, 2005] CHUKMOL, U., RIFAIEH, R. et BENHARKAT, N. (2005). Exsmal : Edi/xml semi-automatic schema matching algorithm. *In E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on*, pages 422–425.

[Colazzo *et al.*, 2014] COLAZZO, D., GOASDOUÉ, F., MANOLESCU, I. et ROATIS, A. (2014). Analyse de données RDF. lentilles pour graphes sémantiques. *Ingénierie des Systèmes d'Information*, 19(4):87–117.

[Coletta *et al.*, 2012] COLETTA, R., CASTANIER, E., VALDURIEZ, P., FRISCH, C., NGO, D. et BELLAHSENE, Z. (2012). Public data integration with websmatch. *In International Workshop on Open Data, WOD '12, Nantes, France, May 25, 2012*, pages 5–12.

[Computer Science department, 2003] COMPUTER SCIENCE DEPARTMENT, u. o. I. a. U.-C. (2003). The uiuc web integration repository.

[Crescenzi *et al.*, 2001] CRESCENZI, V., MECCA, G. et MERIALDO, P. (2001). Roadrunner : Towards automatic data extraction from large web sites. *In Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 109–118, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Cyganiak et Reynolds., 2012] CYGANIAK, R. et REYNOLDS., D. (2012). The rdf data cube vocabulary (w3c working draft). `http://www.w3.org/TR/vocab-data-cube/`.

[Danger et Berlanga, 2009] DANGER, R. et BERLANGA, R. (2009). A semantic web approach for ontological instances analysis. *In* FILIPE, J., SHISHKOV, B., HELFERT, M. et MACIASZEK, L., éditeurs : *Software and Data Technologies*, volume 22 de *Communications in Computer and Information Science*, pages 269–282. Springer Berlin Heidelberg.

[Dehdouh *et al.*, 2014] DEHDOUH, K., BOUSSAID, O. et BENTAYEB, F. (2014). Columnar nosql star schema benchmark. *In Model and Data Engineering - 4th International Conference, MEDI 2014, Larnaca, Cyprus, September 24-26, 2014. Proceedings*, pages 281–288.

[Doan *et al.*, 2003] DOAN, A., MADHAVAN, J., DOMINGOS, P. et HALEVY, A. (2003). Ontology matching : A machine learning approach. *In Handbook on Ontologies in Information Systems*, pages 397–416. Springer.

[Duchateau, 2009] DUCHATEAU, F. (2009). *Towards a Generic Approach for Schema Matcher Selection : Leveraging User Pre- and Post-match Effort for Improving Quality and Time Performance*. Thèse de doctorat, Phd thesis, University of Montpellier 2.

[Duchateau et Bellahsene, 2014] DUCHATEAU, F. et BELLAHSENE, Z. (2014). Designing a benchmark for the assessment of schema matching tools. *Open Journal of Databases (OJDB)*, 1(1):3–25.

[Duchateau *et al.*, 2011] DUCHATEAU, F., BELLAHSENE, Z. et COLETTA, R. (2011). Matching and alignment : What is the cost of user post-match effort ? *In* MEERSMAN, R., DILLON, T., HERRERO, P., KUMAR, A., REICHERT, M., QING, L., OOI, B.-C., DAMIANI, E., SCHMIDT, D., WHITE, J., HAUSWIRTH, M., HITZLER, P. et MOHANIA, M., éditeurs : *On the Move to Meaningful Internet Systems : OTM 2011*, volume 7044 de *Lecture Notes in Computer Science*, pages 421–428. Springer Berlin Heidelberg.

[Duchateau *et al.*, 2007] DUCHATEAU, F., BELLAHSENE, Z. et ROCHE, M. (2007). Bmatch : a semantically context-based tool enhanced by an indexing structure to accelerate schema matching. *In 23èmes Journées Bases de Données Avancées, BDA 2007, Marseille, 23-26 Octobre 2007, Actes (Informal Proceedings)*.

[Duchateau *et al.*, 2009] DUCHATEAU, F., COLETTA, R. et MILLER, R. J. (2009). Yam : a schema matcher factory. *In CIKM*, pages 2079–2080.

[Eberius *et al.*, 2012] EBERIUS, J., THIELE, M., BRAUNSCHWEIG, K. et LEHNER, W. (2012). Drillbeyond : Enabling business analysts to explore the web of open data. *Proc. VLDB Endow.*, 5(12):1978–1981.

[Edmonds, 1965] EDMONDS, J. (1965). Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130.

[Embley *et al.*, 2006] EMBLEY, D., HURST, M., LOPRESTI, D. et NAGY, G. (2006). Table-processing paradigms : a research survey. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(2-3):66–86.

[Embley *et al.*, 2001] EMBLEY, D. W., JACKMAN, D. et XU, L. (2001). Multifaceted exploitation of metadata for attribute match discovery in information integration. *In In Proceedings of the International Workshop on Information Integration on the Web (WIIW'01*, pages 110–117.

[Embley *et al.*, 2002] EMBLEY, D. W., TAO, C. et LIDDLE, S. W. (2002). Automatically extracting ontologically specified data from HTML tables of unknown structure. *In Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Proceedings*, pages 322–337.

[English, 1999] ENGLISH, L. P. (1999). *Improving Data Warehouse and Business Information Quality : Methods for Reducing Costs and Increasing Profits*. John Wiley & Sons, Inc., New York, NY, USA.

[Equille, 2012] EQUILLE, L. B. (2012). *La qualité et la gouvernance des données :au service de la performance des entreprises*. Hermes Lavoisier.

[Etcheverry *et al.*, 2014] ETCHEVERRY, L., VAISMAN, A. et ZIMÀNYI, E. (2014). Modeling and querying data warehouses on the semantic web using QB4OLAP. *In Proceedings of the 16th International Conference on Data Warehousing and Knowledge Discovery, DaWaK'14*, Lecture Notes in Computer Science. Springer-Verlag.

[Euzenat *et al.*, 2004] EUZENAT, J., LE BACH, T., BARRASA, J., BOUQUET, P., DE BO, J., DIENG-KUNTZ, R., EHRIG, M., HAUSWIRTH, M., JARRAR, M., LARA, R., MAYNARD, D., NAPOLI, A., STAMOU, G., STUCKENSCHMIDT, H., SHVAIKO, P., TESSARIS, S., VAN ACKER, S. et ZAIHRAYEU, I. (2004). State of the art on ontology alignment. Rapport technique 2.2.3, Knowledge web.

[Euzenat et Shvaiko, 2007] EUZENAT, J. et SHVAIKO, P. (2007). *Ontology matching*. Springer.

[Euzenat et Shvaiko, 2013] EUZENAT, J. et SHVAIKO, P. (2013). *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2nd édition.

[Euzenat et Valtchev, 2004] EUZENAT, J. et VALTCHEV, P. (2004). Similarity-based ontology alignment in owl-lite. *In Proc. 16th european conference on artificial intelligence (ECAI), Valencia (ES)*, pages 333–337. IOS press.

[Ferrara *et al.*, 2011] FERRARA, A., NIKOLOV, A. et SCHARFFE, F. (2011). Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, 7(3):46–76.

[Ghozzi *et al.*, 2005] GHOZZI, F., RAVAT, F., TESTE, O. et ZURFLUH, G. (2005). Méthode de conception d'une base multidimensionnelle contrainte. *In Actes de la 1ère journée francophone sur les Entrepôts de Données et l'Analyse en ligne, EDA 2005, Lyon, France, Juin 10, 2005*, pages 51–70.

[Giorgini *et al.*, 2005] GIORGINI, P., RIZZI, S. et GARZETTI, M. (2005). Goal-oriented requirement analysis for data warehouse design. *In Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*, DOLAP '05, pages 47–56, New York, NY, USA. ACM.

[Giunchiglia *et al.*, 2004] GIUNCHIGLIA, F., SHVAIKO, P., YATSKEVICH, M., GIUNCHIGLIA, F., SHVAIKO, P. et YATSKEVICH, M. (2004). S-match : an algorithm and an implementation of semantic matching. *In In Proceedings of ESWS*, pages 61–75.

[Golfarelli *et al.*, 1998] GOLFARELLI, M., MAIO, D. et RIZZI, S. (1998). The dimensional fact model : A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7:215–247.

[Governatori *et al.*, 2014] GOVERNATORI, G., LAM, H., ROTOLO, A., VILLATA, S., ATEMEZING, G. A. et GANDON, F. L. (2014). LIVE : a tool for checking licenses compatibility between vocabularies and data. *In Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 77–80.

[Grütze *et al.*, 2012] GRÜTZE, T., BÖHM, C. et NAUMANN, F. (2012). Holistic and scalable ontology alignment for linked open data. *In WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*.

[Hachicha, 2012] HACHICHA, M. (2012). *Modélisation de hiérarchies complexes dans les entrepôts de données XML et traitement des problèmes d'additivié dans l'analyse en ligne XOLAP*. Thèse de doctorat, Université Lumière Lyon 2.

[Hahn *et al.*, 2000] HAHN, K., SAPIA, C. et BLASCHKA, M. (2000). Automatically generating olap schemata from conceptual graphical models. *In* MISSAOUI, R. et SONG, I.-Y., éditeurs : *DOLAP*, pages 9–16. ACM.

[Hammer *et al.*, 2005] HAMMER, J., STONEBRAKER, M. et TOPSAKAL, O. (2005). Thalia : Test harness for the assessment of legacy information integration approaches. *In Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 485–486.

[Hassan, 2014] HASSAN, A. (2014). *Modélisation des Bases de Données Multidimensionnelles : Analyse par Fonctions d'Agrégation Multiple*. Thése de doctorat, Université de Toulouse I Capitole.

[Hassan *et al.*, 2012] HASSAN, A., RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2012). Differentiated multiple aggregations in multidimensional databases. *In Data Warehousing and Knowledge Discovery - 14th International Conference, DaWaK 2012, Vienna, Austria, September 3-6, 2012. Proceedings*, pages 93–104.

[Hassan *et al.*, 2013] HASSAN, A., RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2013). Agrégations multiples différentiées dans les bases de données multidimensionnelles. *Ingénierie des Systèmes d'Information*, 18(2):75–102.

[Hassan *et al.*, 2015] HASSAN, A., RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2015). Differentiated multiple aggregations in multidimensional databases. *T. Large-Scale Data- and Knowledge-Centered Systems*, 21:20–47.

[He et Chang, 2006] HE, B. et CHANG, K. C. (2006). Automatic complex schema matching across web query interfaces : A correlation mining approach. *ACM Trans. Database Syst.*, 31(1):346–395.

[He *et al.*, 2004] HE, H., MENG, W., YU, C. et WU, Z. (2004). Automatic integration of web search interfaces with wise-integrator. *The VLDB Journal*, 13(3):256–273.

[He *et al.*, 2005] HE, H., MENG, W., YU, C. et WU, Z. (2005). WISE-Integrator : a system for extracting and integrating complex web search interfaces of the deep web. *In VLDB '05 : Proceedings of the 31st international conference on Very large data bases*, pages 1314–1317. VLDB Endowment.

[Hignette, 2007] HIGNETTE, G. (2007). *Annotation sémantique foue de tableaux guidée par une ontologie*. Thèse de doctorat, Institut des Sciences et Industries du Vivant et de l'Environnement (Agro Paris Tech).

[Horner et Song, 2005] HORNER, J. et SONG, I.-Y. (2005). A taxonomy of inaccurate summaries and their management in olap systems. *In* DELCAMBRE, L. M. L., KOP, C., MAYR, H. C., MYLOPOULOS, J. et PASTOR, O., éditeurs : *ER*, volume 3716, pages 433–448. Springer.

[Hu *et al.*, 2002] HU, J., KASHI, R., LOPRESTI, D. et WILFONG, G. (2002). Evaluating the performance of table processing algorithms. *International Journal on Document Analysis and Recognition*, 4(3):140–153.

[Huber *et al.*, 2011] HUBER, J., SZTYLER, T., NOSSNER, J. et MEILICKE, C. (2011). Codi : Combinatorial optimization for data integration : results for oaei 2011. *In OM*, volume 814 de *CEUR Workshop Proceedings*. CEUR-WS.org.

[Hurst et Douglas, 1997] HURST, M. et DOUGLAS, S. (1997). Layout and language : Preliminary investigations in recognizing the structure of tables. *In 4th International Conference Document Analysis and Recognition (ICDAR '97), 2-Volume Set, August 18-20, 1997, Ulm, Germany, Proceedings*, pages 1043–1047.

[Hüsemann *et al.*, 2000] HÜSEMANN, B., LECHTENBÖRGER, J. et VOSSEN, G. (2000). Conceptual data warehouse modeling. *In Proceedings of the Second Intl. Workshop on Design and Management of Data Warehouses, DMDW 2000, Stockholm, Sweden, June 5-6, 2000*, page 6.

[Inmon, 1992] INMON, W. H. (1992). *Building the Data Warehouse*. John Wiley & Sons, Inc., New York, NY, USA.

[Inoue *et al.*, 2013] INOUE, H., AMAGASA, T. et KITAGAWA, H. (2013). An etl framework for online analytical processing of linked open data. *In* WANG, J., XIONG, H., ISHIKAWA, Y., XU, J. et ZHOU, J., éditeurs : *Web-Age Information Management*, volume 7923 de *Lecture Notes in Computer Science*, pages 111–117. Springer Berlin Heidelberg.

[Jaccard, 1912] JACCARD, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

[Jentzsch *et al.*, 2010] JENTZSCH, A., ISELE, R. et BIZER, C. (2010). Silk - generating RDF links while publishing or consuming linked data. *In Proceedings of the ISWC 2010 Posters & Demonstrations Track : Collected Abstracts, Shanghai, China, November 9, 2010*.

[Jimenez *et al.*, 2009] JIMENEZ, S., BECERRA, C., GELBUKH, A. F. et GONZALEZ, F. A. (2009). Generalized mongue-elkan method for approximate text string comparison. *In Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009, Mexico City, Mexico, March 1-7, 2009. Proceedings*, pages 559–570.

[Jiménez-Ruiz *et al.*, 2007] JIMÉNEZ-RUIZ, E., BERLANGA, R., NEBOT, V. et SANZ, I. (2007). Ontopath : A language for retrieving ontology fragments. *In* MEERSMAN, R. et TARI, Z., éditeurs : *On the Move to Meaningful Internet Systems 2007 : CoopIS, DOA, ODBASE, GADA,*

and IS, volume 4803 de *Lecture Notes in Computer Science*, pages 897–914. Springer Berlin Heidelberg.

[Kämpgen et Harth, 2011] KÄMPGEN, B. et HARTH, A. (2011). Transforming statistical linked data for use in olap systems. *In Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 33–40, New York, NY, USA. ACM.

[Kämpgen *et al.*, 2012] KÄMPGEN, B., O'RIAIN, S. et HARTH, A. (2012). Interacting with statistical linked data via OLAP operations. *In The Semantic Web : ESWC 2012 Satellite Events - ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012. Revised Selected Papers*, pages 87–101.

[Khouri, 2013] KHOURI, S. (2013). *Cycle de Vie Sémantique de Conception de Systèmes de Stockage et Manipulation de Données*. Thése de doctorat, ISAE-ENSMA et ESI.

[Khouri *et al.*, 2013] KHOURI, S., SARAJ, L. E., BELLATRECHE, L., ESPINASSE, B., BERKANI, N., RODIER, S. et LIBOUREL, T. (2013). Cidhouse : Contextual semantic data warehouses. *In Database and Expert Systems Applications - 24th International Conference, DEXA 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings, Part II*, pages 458–465.

[Kifer *et al.*, 1995] KIFER, M., LAUSEN, G. et WU, J. (1995). Logical foundations of object-oriented and frame-based languages. *JOURNAL OF THE ACM*, 42:741–843.

[Kimball, 1996] KIMBALL, R. (1996). *The Data Warehouse Toolkit : Practical Techniques for Building Dimensional Data Warehouses*. John Wiley & Sons, Inc., New York, NY, USA.

[Kimball et Ross, 2002] KIMBALL, R. et ROSS, M. (2002). *The Data Warehouse Toolkit : The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 2nd édition.

[Labio *et al.*, 2000] LABIO, W., WIENER, J. L., GARCIA-MOLINA, H. et GORELIK, V. (2000). Efficient resumption of interrupted warehouse loads. *In SIGMOD Conference*, pages 46–57. ACM.

[Laborie *et al.*, 2015] LABORIE, S., RAVAT, F., SONG, J. et TESTE, O. (2015). Combining business intelligence with semantic web : Overview and challenges. *In Actes du XXXIIIème Congrès INFORSID, Biarritz, France, May 26-29, 2015*, pages 99–114.

[Laurentini et Viada, 1992] LAURENTINI, A. et VIADA, P. (1992). Identifying and understanding tabular material in compound documents. *In Pattern Recognition, 1992. Vol.II. Conference B : Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pages 405–409.

[Leacock et Chodorow, 1998] LEACOCK, C. et CHODOROW, M. (1998). Combining local context and wordnet similarity for word sense identification. *In* FELLFAUM, C., éditeur : *MIT Press*, pages 265–283, Cambridge, Massachusetts.

[Lenz et Shoshani, 1997] LENZ, H.-J. et SHOSHANI, A. (1997). Summarizability in olap and statistical data bases. *In SSDBM*, pages 132–143. IEEE Computer Society.

[Levenshtein, 1966] LEVENSHTEIN, V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

[Li *et al.*, 2012] LI, X., DONG, X. L., LYONS, K., MENG, W. et SRIVASTAVA, D. (2012). Truth finding on the deep web : Is the problem solved ? *PVLDB*, 6(2):97–108.

[Lin, 1998] LIN, D. (1998). An information-theoretic definition of similarity. *In In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann.

[Liu *et al.*, 2007] LIU, Y., BAI, K., MITRA, P. et GILES, C. L. (2007). Tableseer : Automatic table metadata extraction and searching in digital libraries. *In Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '07, pages 91–100. ACM.

[Liu *et al.*, 2006] LIU, Y., MITRA, P., GILES, C. et BAI, K. (2006). Automatic extraction of table metadata from digital documents. *In Digital Libraries, 2006. JCDL '06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pages 339–340.

[Lopresti et Nagy, 2000] LOPRESTI, D. et NAGY, G. (2000). A tabular survey of automated table processing. *In* CHHABRA, A. et DORI, D., éditeurs : *Graphics Recognition Recent Advances*, volume 1941 de *Lecture Notes in Computer Science*, pages 93–120. Springer Berlin Heidelberg.

[Luján-Mora et Trujillo, 2006] LUJÁN-MORA, S. et TRUJILLO, J. (2006). Physical modeling of data warehouses using UML component and deployment diagrams : Design and implementation issues. *J. Database Manag.*, 17(2):12–42.

[Malinowski et Zimányi, 2006] MALINOWSKI, E. et ZIMÁNYI, E. (2006). Hierarchies in a multidimensional model : From conceptual modeling to logical representation. *Data Knowl. Eng.*, 59(2):348–377.

[Malinowski et Zimanyi, 2008] MALINOWSKI, E. et ZIMANYI, E. (2008). A conceptual model for temporal data warehouses and its transformation to the {ER} and the object-relational models. *Data and Knowledge Engineering*, 64(1):101 – 133.

[Mansmann, 2008] MANSMANN, S. (2008). *Extending the OLAP technology to handle non-conventional and complex data*. Thèse de doctorat, Phd thesis, University of Konstanz, Germany.

[Mansmann *et al.*, 2014] MANSMANN, S., REHMAN, N. U., WEILER, A. et SCHOLL, M. H. (2014). Discovering OLAP dimensions in semi-structured data. *Information Systems*, 44: 120–133.

[Mansmann et Scholl, 2007] MANSMANN, S. et SCHOLL, M. H. (2007). Empowering the OLAP technology to support complex dimension hierarchies. *IJDWM*, 3(4):31–50.

[Marie et Gal, 2008] MARIE, A. et GAL, A. (2008). Boosting schema matchers. *In* MEERSMAN, R. et TARI, Z., éditeurs : *On the Move to Meaningful Internet Systems : OTM 2008*, volume 5331 de *Lecture Notes in Computer Science*, pages 283–300. Springer Berlin Heidelberg.

[Mazón *et al.*, 2012] MAZÓN, J., ZUBCOFF, J. J., GARRIGÓS, I., ESPINOSA, R. et RODR ÍGUEZ, R. (2012). Open business intelligence : On the importance of data quality awareness in User-friendly data mining. *In Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT 2012, pages 144–147, New York, NY, USA. ACM.

[Mazón *et al.*, 2010] MAZÓN, J.-N., LECHTENBORGER, J. et TRUJILLO, J. (2010). A survey on summarizability issues in multidimensional modeling. *In* TENIENTE, E. et ABRAHÃ£O, S., éditeurs : *JISBD*, pages 327–327. IBERGARCETA Pub. S.L.

[Meilicke, 2011] MEILICKE, C. (2011). *Alignment Incoherence in Ontology Matching*. Phd thesis, University Mannheim, Deutshland.

[Melnik *et al.*, 2002] MELNIK, S., GARCIA-MOLINA, H. et RAHM, E. (2002). Similarity flooding : A versatile graph matching algorithm and its application to schema matching. *In Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 117–. IEEE Computer Society.

[Miller, 1995] MILLER, G. (1995). Wordnet : A lexical database for english. *Communication ACM*, 38(11):39–41.

[Monge et Elkan, 1996] MONGE, A. et ELKAN, C. (1996). The field matching problem : Algorithms and applications. *In In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270.

[Moody et Kortink, 2000] MOODY, D. L. et KORTINK, M. A. R. (2000). From enterprise models to dimensional models : a methodology for data warehouse and data mart design. *In DMDW*, volume 28 de *CEUR Workshop Proceedings*, page 5. CEUR-WS.org.

[Müller *et al.*, 1999] MÜLLER, R., STÖHR, T. et RAHM, E. (1999). An integrative and uniform model for metadata management in data warehousing environments. *In Proceedings of the Intl. Workshop on Design and Management of Data Warehouses, DMDW'99, Heidelberg, Germany, June 14-15, 1999*, page 12.

[Nebot et Berlanga, 2012] NEBOT, V. et BERLANGA, R. (2012). Building data warehouses with semantic web data. *Decis. Support Syst.*, 52(4):853–868.

[Nebot *et al.*, 2009] NEBOT, V., LLAVORI, R. B., PÉREZ-MARTÍNEZ, J. M., ARAMBURU, M. J. et PEDERSEN, T. B. (2009). Multidimensional integrated ontologies : A framework for designing semantic data warehouses. *J. Data Semantics*, 13:1–36.

[Ng *et al.*, 1999] NG, H. T., LIM, C. Y. et KOO, J. L. T. (1999). Learning to recognize tables in free text. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 443–450, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ngo et Bellahsene, 2012] NGO, D. et BELLAHSENE, Z. (2012). YAM++ : A multi-strategy based approach for ontology matching task. *In Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, pages 421–425.

[Ngo *et al.*, 2011] NGO, D., BELLAHSENE, Z. et COLETTA, R. (2011). A generic approach for combining linguistic and context profile metrics in ontology matching. *In On the Move to Meaningful Internet Systems : OTM 2011 - Confederated International Conferences : CoopIS, DOA-SVI, and ODBASE 2011, Hersonissos, Crete, Greece, October 17-21, 2011, Proceedings, Part II*, pages 800–807.

[Niepert *et al.*, 2010] NIEPERT, M., MEILICKE, C. et STUCKENSCHMIDT, H. (2010). A Probabilistic-Logical Framework for Ontology Matching. *In Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1413–1418. AAAI Press.

[Pedersen *et al.*, 1999] PEDERSEN, T. B., JENSEN, C. S. et DYRESON, C. E. (1999). Extending practical pre-aggregation in on-line analytical processing. *In* ATKINSON, M. P., ORLOWSKA, M. E., VALDURIEZ, P., ZDONIK, S. B. et BRODIE, M. L., éditeurs : *VLDB*, pages 663–674. Morgan Kaufmann.

[Pedersen *et al.*, 2001] PEDERSEN, T. B., JENSEN, C. S. et DYRESON, C. E. (2001). A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26:383–423.

[Pei *et al.*, 2006] PEI, J., HONG, J. et BELL, D. (2006). A robust approach to schema matching overweb query interfaces. *In Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 46–46.

[Phipps et Davis, 2002] PHIPPS, C. et DAVIS, K. C. (2002). Automating data warehouse conceptual schema design and evaluation. *In* LAKSHMANAN, L. V. S., éditeur : *DMDW*, volume 58 de *CEUR Workshop Proceedings*, pages 23–32. CEUR WS.org.

[Pivk *et al.*, 2004] PIVK, A., CIMIANO, P. et SURE, Y. (2004). From tables to frames. *In* MCILRAITH, S., PLEXOUSAKIS, D. et van HARMELEN, F., éditeurs : *The Semantic Web ISWC 2004*, volume 3298 de *Lecture Notes in Computer Science*, pages 166–181. Springer Berlin Heidelberg.

[Plastria, 2002] PLASTRIA, F. (2002). Formulating logical implications in combinatorial optimisation. *European Journal of Operational Research*, 140(2):338 – 353.

[Prat *et al.*, 2006] PRAT, N., AKOKA, J. et COMYN-WATTIAU, I. (2006). A uml-based data warehouse design method. *Decision Support Systems*, 42(3):1449 – 1473.

[Prat *et al.*, 2012] PRAT, N., MEGDICHE, I. et AKOKA, J. (2012). Multidimensional models meet the semantic web : defining and reasoning on OWL-DL ontologies for OLAP. *In DOLAP 2012, ACM 15th International Workshop on Data Warehousing and OLAP, Maui, HI, USA, November 2, 2012, Proceedings*, pages 17–24.

[Quinlan, 1993] QUINLAN, J. R. (1993). *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Rafanelli et Shoshani, 1990] RAFANELLI, M. et SHOSHANI, A. (1990). Storm : A statistical object representation model. *In Proceedings of the Fifth International Conference on Statistical and Scientific Database Management*, SSDBM V, pages 14–29, New York, NY, USA. Springer-Verlag New York, Inc.

[Rahm, 2011] RAHM, E. (2011). Towards large-scale schema and ontology matching. *In* BELLAHSENE, Z., BONIFATI, A. et RAHM, E., éditeurs : *Schema Matching and Mapping*, Data-Centric Systems and Applications, pages 3–27. Springer Berlin Heidelberg.

[Rahm et Bernstein, 2001] RAHM, E. et BERNSTEIN, P. A. (2001). A survey of approaches to automatic schema matching. *VLDB JOURNAL*, 10.

[Ravat et Teste, 2000] RAVAT, F. et TESTE, O. (2000). A temporal object-oriented data warehouse model. *In Database and Expert Systems Applications, 11th International Conference, DEXA 2000, London, UK, September 4-8, 2000, Proceedings*, pages 583–592.

[Ravat et Teste, 2008] RAVAT, F. et TESTE, O. (2008). Personalization and olap databases. *Annals of Information Systems, New Trends in Data Warehousing and Data Analysis*, 3:71–92.

[Ravat *et al.*, 2007a] RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2007a). A conceptual model for multidimensional analysis of documents. *In Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November 5-9, 2007, Proceedings*, pages 550–565.

[Ravat *et al.*, 2007b] RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2007b). Graphical querying of multidimensional databases. *In Advances in Databases and Information Systems, 11th East European Conference, ADBIS 2007, Varna, Bulgaria, September 29-October 3, 2007, Proceedings*, pages 298–313.

[Ravat *et al.*, 2008] RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2008). Algebraic and graphic languages for OLAP manipulations. *International Journal of Data Warehousing and Mining*, 4(1):17–46.

[Ravat *et al.*, 2010] RAVAT, F., TESTE, O., TOURNIER, R. et ZURFLUH, G. (2010). Finding an application-appropriate model for XML data warehouses. *Inf. Syst.*, 35(6):662–687.

[Ravat *et al.*, 1999] RAVAT, F., TESTE, O. et ZURFLUH, G. (1999). Towards data warehouse design. *In Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management, Kansas City, Missouri, USA, November 2-6, 1999*, pages 359–366.

[Ravat *et al.*, 2001] RAVAT, F., TESTE, O. et ZURFLUH, G. (2001). Modélisation multidimensionnelle des systèmes décisionnels. *In Extraction et gestion des connaissances (EGC'2001), Actes des premières journées Extraction et Gestion des Connaissances, Nantes, France, 17-19 janvier 2001*, pages 201–212.

[Ravat *et al.*, 2002] RAVAT, F., TESTE, O. et ZURFLUH, G. (2002). Langage pour bases multidimensionnelles : OLAP-SQL. *Ingénierie des Systèmes d'Information*, 7(3):11–38.

[Ravat *et al.*, 2006a] RAVAT, F., TESTE, O. et ZURFLUH, G. (2006a). Constraint-Based Multi-Dimensional Databases. *In* ZONGMIN, M., éditeur : *Database Modeling for Industrial Data Management*, pages 323–368. Idea Group.

[Ravat *et al.*, 2006b] RAVAT, F., TESTE, O. et ZURFLUH, G. (2006b). A multiversion-based multidimensional model. *In Data Warehousing and Knowledge Discovery, 8th International Conference, DaWaK 2006, Krakow, Poland, September 4-8, 2006, Proceedings.*, pages 65–74.

[Riedel, 2008] RIEDEL, S. (2008). Improving the accuracy and efficiency of map inference for markov logic. *In* MCALLESTER, D. A. et MYLLYMAKI, P., éditeurs : *UAI*, pages 468–475. AUAI Press.

[Rizzi *et al.*, 2006] RIZZI, S., ABELLÓ, A., LECHTENBÖRGER, J. et TRUJILLO, J. (2006). Research in data warehouse modeling and design : Dead or alive ? *In Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP*, DOLAP '06, pages 3–10.

[Rodriguez et Neubauer, 2010] RODRIGUEZ, M. A. et NEUBAUER, P. (2010). Constructions from dots and lines. *CoRR*, abs/1006.2361.

[Romero et Abelló, 2007] ROMERO, O. et ABELLÓ, A. (2007). Automating multidimensional design from ontologies. *In Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP*, DOLAP '07, pages 1–8, New York, NY, USA.

[Romero et Abelló, 2009] ROMERO, O. et ABELLÓ, A. (2009). A survey of multidimensional modeling methodologies. *IJDWM*, 5(2):1–23.

[Romero *et al.*, 2011] ROMERO, O., SIMITSIS, A. et ABELLO, A. (2011). Gem : Requirement-driven generation of etl and multidimensional conceptual designs. *In* CUZZOCREA, A. et DAYAL, U., éditeurs : *DaWaK*, volume 6862 de *Lecture Notes in Computer Science*, pages 80–95. Springer.

[Rumelhart *et al.*, 1988] RUMELHART, D. E., HINTON, G. E. et WILLIAMS, R. J. (1988). Learning internal representations by error propagation. *In* ANDERSON, J. A. et ROSENFELD, E., éditeurs : *Neurocomputing : Foundations of Research*, pages 673–695. MIT Press.

[Saad *et al.*, 2013] SAAD, R., TROJAHN, C. et TESTE, O. (2013). OLAP manipulations on RDF data following a constellation model. *In First International Workshop on Semantic Statistics, collocated with the 12th International Semantic Web Conference, Sydney*, page (on line). DataLift.

[Saïs *et al.*, 2005] SAÏS, F., GAGLIARDI, H., HAEMMERLÉ, O. et PERNELLE, N. (2005). Enrichissement sémantique de documents XML représentant des tableaux. *In Extraction et gestion des connaissances (EGC'2005), Actes des cinquièmes journées Extraction et Gestion des Connaissances, Paris, France, 18-21 janvier 2005, 2 Volumes*, pages 407–418.

[Saleem *et al.*, 2007] SALEEM, K., BELLAHSENE, Z. et HUNT, E. (2007). Performance oriented schema matching. *In DEXA*, volume 4653, pages 844–853.

[Sapia *et al.*, 1999] SAPIA, C., BLASCHKA, M., HÖFLING, G. et DINTER, B. (1999). Extending the e/r model for the multidimensional paradigm. *In Proceedings of the Workshops on Data Warehousing and Data Mining : Advances in Database Technologies*, ER '98, pages 105–116, London, UK, UK. Springer-Verlag.

[Scharffe *et al.*, 2012] SCHARFFE, F., ATEMEZING, G., TRONCY, R., GANDON, F., VILLATA, S., BUCHER, B., HAMDI, F., BIHANIC, L., KÉPÉKLIAN, G., COTTON, F., EUZENAT, J., FAN, Z., VANDENBUSSCHE, P.-Y. et VATANT, B. (2012). Enabling linked data publication with the Datalift platform. *In Proc. AAAI workshop on semantic cities*, page No pagination., Toronto, Canada.

[Schneider, 2003] SCHNEIDER, M. (2003). Well-formed data warehouse structures. *In Design and Management of Data Warehouses 2003, Proceedings of the 5th Intl. Workshop DMDW'2003, Berlin, Germany, September 8, 2003*.

[Schneider, 2008] SCHNEIDER, M. (2008). A general model for the design of data warehouses. *International Journal of Production Economics*, 112(1):309 – 325.

[Schneider *et al.*, 2011] SCHNEIDER, M., VOSSEN, G. et ZIMÁNYI, E. (2011). Data Warehousing : from Occasional OLAP to Real-time Business Intelligence (Dagstuhl Seminar 11361). *Dagstuhl Reports*, 1(9):1–25.

[Schrijver, 2003] SCHRIJVER, A. (2003). *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

[Serment *et al.*, 2008] SERMENT, J., ESPINASSE, B. et TRANVOUEZ, E. (2008). Systèmes d'aide à la décision environnementale. une infrastructure d'intégration orientée agent. *Journal of Decision Systems*, 17(2):269–300.

[Shin, 2003] SHIN, B. (2003). An exploratory investigation of system success factors in data warehousing. *J. AIS*, 4:0–.

[Shvaiko et Euzenat, 2005] SHVAIKO, P. et EUZENAT, J. (2005). A survey of schema-based matching approaches. *In Journal on Data Semantics IV*, pages 146–171. Springer Berlin Heidelberg.

[Shvaiko et Euzenat, 2013] SHVAIKO, P. et EUZENAT, J. (2013). Ontology matching : State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176.

[Skoutas et Simitsis, 2007] SKOUTAS, D. et SIMITSIS, A. (2007). Ontology-based conceptual design of etl processes for both structured and semi-structured data. *Int. J. Semantic Web Inf. Syst.*, 3(4):1–24.

[Stoilos *et al.*, 2005] STOILOS, G., STAMOU, G. et KOLLIAS, S. (2005). A string metric for ontology alignment. *In Proceedings of the 4th International Conference on The Semantic Web*, ISWC'05, pages 624–637, Berlin, Heidelberg. Springer-Verlag.

[Su *et al.*, 2006a] SU, W., WANG, J. et LOCHOVSKY, F. H. (2006a). Holistic query interface matching using parallel schema matching. *In* LIU, L., REUTER, A., WHANG, K. Y. et ZHANG, J., éditeurs : *ICDE*, page 122.

[Su *et al.*, 2006b] SU, W., WANG, J. et LOCHOVSKY, F. H. (2006b). Holistic schema matching for web query interfaces. *In Advances in Database Technology - EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26-31, 2006, Proceedings*, pages 77–94.

[Sun *et al.*, 2015] SUN, Y., MA, L. et SHUANG, W. (2015). A comparative evaluation of string similarity metrics for ontology alignement. *Journal of Information & Computational Science*, 12(3):957 – 964.

[Tenier *et al.*, 2006] TENIER, S., TOUSSAINT, Y., NAPOLI, A. et POLANCO, X. (2006). Instantiation of relations for semantic annotation. *In 2006 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2006), 18-22 December 2006, Hong Kong, China*, pages 463–472.

[Termier *et al.*, 2004] TERMIER, A., ROUSSET, M.-C. et SEBAG, M. (2004). Dryade : a new approach for discovering closed frequent trees in heterogeneous tree databases. *In Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pages 543–546.

[Teste, 2000] TESTE, O. (2000). *Modélisation et Manipulation d'Entrepôts de Données Complexes et Historisées*. Thése de doctorat, Université Paul Sabatier.

[Teste, 2009] TESTE, O. (2009). *Modélisation et manipulation des systèmes OLAP : de l'intégration des documents à l'usager*. Habilitation à diriger des recherches en informatique, Université Paul Sabatier Toulouse III.

[Tijerino *et al.*, 2005] TIJERINO, Y., EMBLEY, D., LONSDALE, D., DING, Y. et NAGY, G. (2005). Towards ontology generation from tables. *World Wide Web*, 8(3):261–285.

[Tournier, 2004] TOURNIER, R. (2004). Vers un langage de manipulation graphique des bases multidimensionnelles.

[Tournier, 2007] TOURNIER, R. (2007). *Analyse en Ligne (OLAP) des documents*. Thése de doctorat, Université Paul Sabatier Toulouse III.

[Trujillo et Luján-Mora, 2003] TRUJILLO, J. et LUJÁN-MORA, S. (2003). A UML based approach for modeling ETL processes in data warehouses. *In Conceptual Modeling - ER 2003, 22nd International Conference on Conceptual Modeling, Chicago, IL, USA, October 13-16, 2003, Proceedings*, pages 307–320.

[Trujillo *et al.*, 2003] TRUJILLO, J., LUJAN-MORA, S. et SONG, I.-Y. (2003). Applying uml for designing multidimensional databases and olap applications. *In Advanced Topics in Database Research, Vol. 2*, pages 13–36.

[Trujillo et Palomar, 1998] TRUJILLO, J. et PALOMAR, M. (1998). An object oriented approach to multidimensional database conceptual modeling (oomd). *In Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*, DOLAP '98, pages 16–21.

[Tschinkel *et al.*, 2014] TSCHINKEL, G., VEAS, E. E., MUTLU, B. et SABOL, V. (2014). Using semantics for interactive visual analysis of linked open data. *In Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 133–136.

[Tsikriktsis, 2005] TSIKRIKTSIS, N. (2005). A review of techniques for treating missing data in om survey research. *Journal of Operations Management*, 24(1):53 – 62.

[Tziovara *et al.*, 2007] TZIOVARA, V., VASSILIADIS, P. et SIMITSIS, A. (2007). Deciding the physical implementation of ETL workflows. *In DOLAP 2007, ACM 10th International Workshop on Data Warehousing and OLAP, Lisbon, Portugal, November 9, 2007, Proceedings*, pages 49–56.

[Van Assem *et al.*, 2010] VAN ASSEM, M., RIJGERSBERG, H., WIGHAM, M. et TOP, J. (2010). Converting and annotating quantitative data tables. *In* PATEL-SCHNEIDER, P., PAN, Y., HITZLER, P., MIKA, P., ZHANG, L., PAN, J., HORROCKS, I. et GLIMM, B., éditeurs : *The Semantic Web ISWC 2010*, volume 6496 de *Lecture Notes in Computer Science*, pages 16–31. Springer Berlin Heidelberg.

[Vassiliadis, 2009] VASSILIADIS, P. (2009). A survey of extract-transform-load technology. *IJDWM*, 5(3):1–27.

[Vassiliadis et Simitsis, 2009] VASSILIADIS, P. et SIMITSIS, A. (2009). Extraction, transformation, and loading. *In Encyclopedia of Database Systems*, pages 1095–1101.

[Vassiliadis *et al.*, 2005] VASSILIADIS, P., SIMITSIS, A., GEORGANTAS, P., TERROVITIS, M. et SKIADOPOULOS, S. (2005). A generic and customizable framework for the design of etl scenarios. *Inf. Syst.*, 30(7):492–525.

[Vassiliadis *et al.*, 2002] VASSILIADIS, P., SIMITSIS, A. et SKIADOPOULOS, S. (2002). Modeling ETL activities as graphs. *In Design and Management of Data Warehouses 2002, Proceedings of the 4th Intl. Workshop DMDW'2002, Toronto, Canada, May 27, 2002*, pages 52–61.

[Volz *et al.*, 2009] VOLZ, J., BIZER, C., GAEDKE, M. et KOBILAROV, G. (2009). Discovering and maintaining links on the web of data. *In Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 650–665.

[Vrdoljak *et al.*, 2003] VRDOLJAK, B., BANEK, M. et RIZZI, S. (2003). Designing web warehouses from xml schemas. *In* KAMBAYASHI, Y., MOHANIA, M. et W., W., éditeurs : *Data Warehousing and Knowledge Discovery*, volume 2737 de *Lecture Notes in Computer Science*, pages 89–98. Springer Berlin Heidelberg.

[Wang, 1996] WANG, X. (1996). *Tabular Abstraction, Editing, and Formatting*. Thèse de doctorat, Phd thesis, University of Waterloo, Waterloo, Ontario, Canada.

[Winkler, 1990] WINKLER, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *In Proceedings of the Section on Survey Research*, pages 354–359.

[Wu et Palmer., 1994] WU, Z. et PALMER., M. (1994). Verb semantics and lexical selection. *In In 32nd. Annual Meeting of the Association for Computational Linguistics, New Mexico State University, Las Cruces, New Mexico.*, pages 133–138.

[Xu et Embley, 2003] XU, L. et EMBLEY, D. (2003). Using domain ontolologies to discover direct and indirect matches for schema elements. *In In Proceedings ofthe workshop on semantics integration WSI'03 Sanibel Island , Florida*, pages 105–110.

[Yatskevich, 2008] YATSKEVICH, M. (2008). *Semantic Matching algorithms and implementation*. Thèse de doctorat, Phd thesis, University of Trento.

[Zaamoune *et al.*, 2013] ZAAMOUNE, M., BIMONTE, S., PINET, F. et BEAUNE, P. (2013). Intégration des données champs continus incomplets dans l'olap : de la modélisation conceptuelle à l'implémentation. *In Actes des 9èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA 2013, Blois, France, Juin 13-14, 2013*, pages 33–42.

[Zanibbi *et al.*, 2004] ZANIBBI, R., BLOSTEIN, D. et CORDY, R. (2004). A survey of table recognition : Models, observations, transformations, and inferences. *Int. J. Doc. Anal. Recognit.*, 7(1):1–16.

[Zaveri *et al.*, 2014] ZAVERI, A., MAURINO, A. et BERTI-EQUILLE, L. (2014). Web data quality : Current state and new challenges. *Int. J. Semantic Web Inf. Syst.*, 10(2):1–6.

Résumé :

Les statistiques présentes dans les Open Data ou données ouvertes constituent des informations utiles pour alimenter un système décisionnel. Leur intégration et leur entreposage au sein du système décisionnel se fait à travers des processus ETL. Il faut automatiser ces processus afin de faciliter leur accessibilité à des non-experts. Ces processus doivent pallier aux problèmes de manque de schémas, d'hétérogénéité structurelle et sémantique qui caractérisent les données ouvertes. Afin de répondre à ces problématiques, nous proposons une nouvelle démarche ETL basée sur les graphes. Pour l'extraction du graphe d'un tableau, nous proposons des activités de détection et d'annotation automatiques. Pour la transformation, nous proposons un programme linéaire pour résoudre le problème d'appariement holistique de données structurelles provenant de plusieurs graphes. Ce modèle fournit une solution optimale et unique. Pour le chargement, nous proposons un processus progressif pour la définition du schéma multidimensionnel et l'augmentation du graphe intégré. Enfin, nous présentons un prototype et les résultats d'expérimentations.

Mots clés: Données ouvertes, ETL, Graphes, Détection tableaux, Intégration holistique, Entrepôt de données.

Abstract :

Statistical Open Data present useful information to feed up a decision-making system. Their integration and storage within these systems is achieved through ETL processes. It is necessary to automate these processes in order to facilitate their accessibility to non-experts. These processes have also need to face out the problems of lack of schemes and structural and sematic heterogeneity, which characterize the Open Data. To meet these issues, we propose a new ETL approach based on graphs. For the extraction, we propose automatic activities performing detection and annotations based on a model of a table. For the transformation, we propose a linear program fulfilling holistic integration of several graphs. This model supplies an optimal and a unique solution. For the loading, we propose a progressive process for the definition of the multidimensional schema and the augmentation of the integrated graph. Finally, we present a prototype and the experimental evaluations.

Keywords: Open Data, ETL, Graphs, Table detection, Holistic integration, Data warehouses.