HAL
open science

Méthodes et outils pour lier le web des données

François Scharffe, Jérôme Euzenat

To cite this version:

François Scharffe, Jérôme Euzenat. Méthodes et outils pour lier le web des données. Actes 17e conférence AFIA-AFRIF sur reconnaissance des formes et intelligence artificielle (RFIA), Jan 2010, Caen, France. pp.678-685. hal-00793284

HAL Id: hal-00793284

https://inria.hal.science/hal-00793284

Submitted on 22 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méthodes et outils pour lier le web des données

François Scharffe Jérôme Euzenat

INRIA & LIG
655 avenue de l'Europe
38330 Montbonnot Saint-Martin, France
prenom.nom@inria.fr

Résumé

Le web des données consiste à publier des données sur le web de telle sorte qu'elles puissent être interprétées et connectées entre elles. Il est donc vital d'établir les liens entre ces données à la fois pour le web des données et pour le web sémantique qu'il contribue à nourrir. Nous proposons un cadre général dans lequel s'inscrivent les différentes techniques utilisées pour établir ces liens et nous montrons comment elles s'y insèrent. Nous proposons ensuite une architecture permettant d'associer les différents systèmes de liage de données et de les faire collaborer avec les systèmes développés pour la mise en correspondance d'ontologies qui présente de nombreux points communs avec la découverte de liens.

Mots Clef

Web sémantique, alignement d'ontologies, Web des données.

Abstract

The Web of data consists of publishing data on the Web in such a way that they can be interpreted and connected together. It is thus critical to be able to establish links between these data, both for the Web of data and for the Semantic Web that this one contributes to feed. We propose a general framework and we show how the diverse techniques developed for establishing these links fit in the framework. We then propose an architecture allowing to associate various interlinking systems and to make them collaborate with systems developed for ontology matching that present many commonalities with link discovery techniques.

Keywords

Semantic Web, ontology alignment, Web of data.

1 Introduction

On désigne par "web des données" le réseau formé par la publication de jeux de données structurées décrites en RDF (*Resource Description Framework*) et reliées entre elles par des liens explicites. De grandes quantités de données structurées ont été publiées [1], notamment dans le cadre du projet *Linking Open Data* [2].

Le web des données requiert de lier entre elles les différentes sources de données publiées. Au vu de l'immense quantité de données publiées, il est nécessaire de fournir des méthodes de liage automatique de ces données. Plusieurs outils ont récemment été proposés pour résoudre ce problème, chacun ayant ses propres charactéristiques.

Les jeux de données sont exprimés en fonction d'une ou de plusieurs ontologies permettant de fixer le vocabulaire dans lequel les données sont exprimées. Dans de nombreux cas, des jeux de données décrivant des ressources d'un domaine similaire sont publiées en utilisant des ontologies différentes. Notre travail est motivé par l'hypothèse suivante : si un alignement entre ontologies spécifie de manière explicite les correspondances entre entités similaires provenant de deux ontologies, alors cet alignement spécifie aussi quelles ressources sont susceptibles d'être liées. Dès lors, la question se pose de savoir si l'établissement de liens doit être vu comme une extension de l'alignement d'ontologie et s'il convient d'en fusionner les outils.

Le but de ce travail est d'analyser les systèmes de liage existants et de déterminer (1) comment ils participent d'une même activité, (2) s'il est possible de définir un langage permettant de spécifier les techniques de liage à utiliser et (3) en quoi cette activité diffère-t-elle de, ou est-elle liée à, l'alignement d'ontologie.

Cet article est organisé comme suit : Après une introduction au web des données et à ses techniques (section 2), nous établissons (section 3) un cadre de travail prenant en compte les différent cas pouvant se présenter lors de l'alignement de deux jeux de données. Nous analysons ensuite (section 4) six outils d'alignement de jeux de données et les relions au cadre de travail défini. Nous proposons (section 5) une approche permettant aux outils d'alignement de données de profiter des alignements entre ontologies.

1. 4.2 Milliards de triplets RDF, reliés par 142 millions de liens (source Wikipedia, daté de Mai 2009)

2. `http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

2 Web des données et interrelations

Le web des données est basé sur les quatre principes suivants [3] :

1. Les ressources sont identifiées par des URI (*Uniform Resource Identifiers*, indentifiants uniformes de ressources).
2. Ces URI peuvent être déréférencées sur le web. C'est-à-dire que chaque URI sur le web des données au travers du protocole HTTP.
3. Toute URI déréférencée retourne des méta-données décrivant la ressource identifiée.
4. Tout jeu de données publié doit contenir des liens vers d'autres jeux de données.

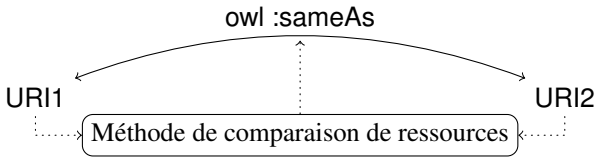
Deux méthodes de publication des données sont disponibles. D'un coté la publication de jeux de données autonomes : ces ensembles de ressources peuvent provenir de bases de données relationelles ou de documents XML, exportés et décrits en fonction d'une ontologie [4], ou bien construits par l'intermédiaire d'outils collaboratifs [18]. Une autre méthode consiste à inclure des données structurées en tant qu'annotations contenues dans les pages web [1]. Nous considérons principalement les méthodes et outils travaillant directement avec des jeux de données.
Une fois identifiés, les liens découverts entre les données sont publiés. Le vocabulaire VoiD [2] permet de décrire les liens entre deux jeux de données. Ci-dessous un example de graphe VoiD décrivant une correspondance entre deux ressources. Ce type de graphe de liens est appelé un *linkset*.

```
{pouvant
 <http://www.example.org/linkset/DBpedia-MB>
  a void:Linkset ;
  void:target <http://www.dpbedia.org>;
  void:target <http://www.musicbrainz.org>;
}
<http://www.example.org/linkset/DBpedia-MB>
{
 <http://www.dbpedia.org/Johann_S_Bach>
 owl:sameAs
 <http://www.musicbrainz.org/123a4536b> .
}
```

Une fois le linkset construit, deux approches sont possibles pour publier les équivalences entre ressources : il est possible d'associer à chaque entité un identifiant spécifique qui est ensuite relié aux différentes URI équivalentes représentant cette entité. C'est l'approche développée dans le projet OKKAM [7] qui propose l'utilisation d'*Entity Name Servers* (*Serveurs de noms d'entités*) ayant le rôle de répertoires de ressources. L'autre approche utilise, pour chaque URI des listes d'URI équivalentes. Il n'y a donc pas d'identifiant global ou de référence dans cette approche mais les liens équivalents peuvent être suivi en utilisant un service web tiers comme `http://sameas.org`, ou un protocole bilatteral [19].
La tâche de *liage de données* peut être accomplie manuellement ou avec l'aide d'outils de liage. Ces outils prennent en entrée deux jeux de données et produisent ultimement un "linkset". En complément, ces lieurs peuvent utiliser des *spécifications de liage*, c'est-à-dire des scripts spécifiant quoi et/ou comment lier. Au vu de la taille des jeux de données, l'espace de recherche peut atteindre plusieurs milliard de ressources (comme pour DBpedia). Il est donc nécessaire d'utiliser des heuristiques pour guider le processus de liage indiquant aux lieurs où chercher les ressources à lier dans les deux jeux de données. Ces spécifications de liage peuvent être spécifiques à une paire de jeu de données et peuvent être réutilisées pour mettre à jour des linksets.

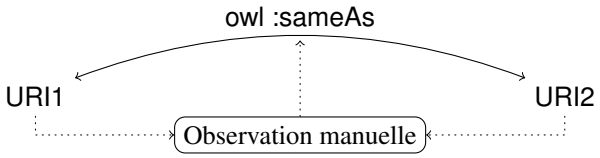
3 Une approche générale pour relier les données du web

Nous présentons dans cette section un cadre englobant les diverses approches utilisées pour lier les ressources du web des données. Ce cadre s'adapte aux différents cas de figure pouvant être rencontrés quand deux jeux de données sont reliés. Nous verrons dans la section suivante que chacun des outils analysés y trouve sa place.
Dans le cas général illustré ci-dessous deux jeux de données sont reliés par une méthode de comparaison des ressources. Ces jeux de données sont décrits chacun par une ontologie [3]. Le résultat de la méthode, automatique ou manuelle est un ensemble de relations `owl:sameAs` entre ces ressources.



3.1 Alignement manuel des ressources

Dans le cas illustré ci-dessous les ressources sont alignées manuellement.

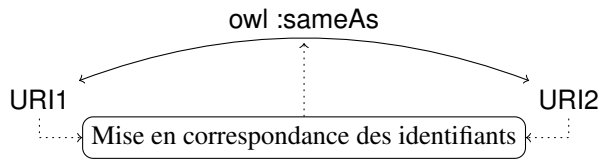


L'alignement manuel de ressources peut être effectué par des outils collaboratifs dans le cas de grands jeux de données.

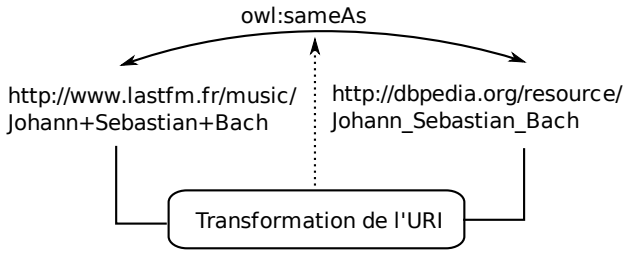
3.2 Mise en correspondance des identifiants

Dans le cas illustré ci-dessous les identifiants des ressources peuvent être mis en correspondance en utilisant une simple transformation.

3. En fait chacun peut être décrit par plusieurs ontologies qui peuvent différer d'un ensemble à l'autre.

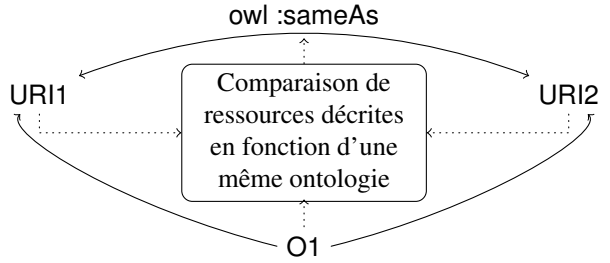


Dans ce cas, un ensemble de règles est défini pour identifier les ressources équivalentes à partir de leur identifiant. Par exemple, dans le jeu de données de LastFM, l'URI d'un artiste est construite sur le motif "Prénom+Nom". Les URI de personnes dans DBpedia sont construites sur le motif "Prénom_Nom". L'algorithme trouvant les artistes équivalents entre les deux jeux de données sera trivial à construire. Cet exemple est illustré ci-dessous pour le compositeur Jean-Sébastien Bach.



3.3 Alignement de données avec ontologie commune

Dans le cas illustré ci-dessous les deux jeux de données à aligner sont décrits avec la même ontologie. Le rôle du système d'alignement des données est d'inspecter les ressources de même type afin de détecter celles qui sont équivalentes. Pour cela le système va comparer les propriétés des ressources et construire une mesure de similarité. Les systèmes de cette catégorie sont paramétrés par les propriétés à comparer, le type d'algorithme de comparaison à utiliser pour chaque propriété, et la façon dont la mesure de similarité entre les ressources est construite en fonction de la similarité entre les propriétés.



Par exemple, Jamendo et Musicbrainz [4] sont tous deux décrits par rapport à une même ontologie de la musique [14]. L'artiste J-S Bach pourra être identifié dans les deux jeux de données en observant les propriétés nom et prénom de la class *MusicArtist*. Cet exemple est illustré Figure 1

4. Deux bases de données discographiques : Jamendo : `http://www.jamendo.com` et MusicBrainz : `http://www.musicbrainz.org`

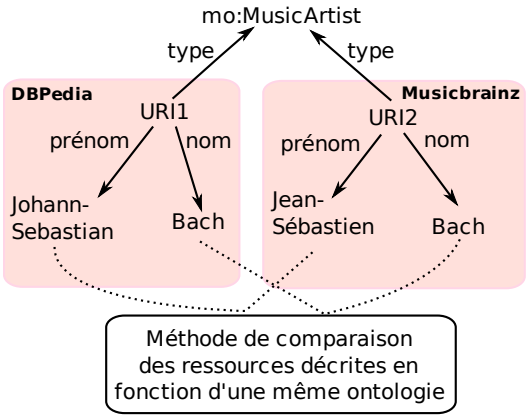
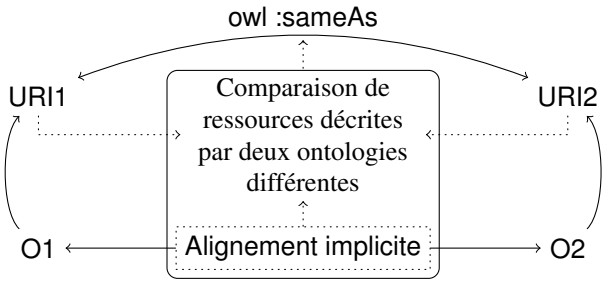


FIGURE 1 – Exemple de comparaison de ressources.

3.4 Ontologies différentes et alignement implicite

Dans ce cas illustré ci-dessous, les deux jeux de données à relier sont décrits par des ontologies hétérogènes. Un alignement entre les ontologies est utilisé pour indiquer au système d'alignement des données les correspondances entre les entités des ontologies. Le système fonctionne ensuite de façon similaire à un système à une seule ontologie. Dans ce cas, l'alignement est spécifié de manière implicite par l'utilisateur lorsque les types de ressources et les propriétés à examiner sont spécifiées.



Par exemple, OpenCyc [5] représente l'artiste Bach différemment de sa représentation dans MusicBrainz. Les propriétés nom et prénom correspondent à une propriété *EnglishID* dans laquelle sont concaténés le nom et prénom de l'artiste. La classe *MusicArtist* de l'ontologie de la musique est plus générale que la classe *Classical Music Performer* dans OpenCyc. Un alignement entre classes et propriétés doit être décrit pour pouvoir trouver l'équivalence entre ces deux ressources. Cet exemple est illustré par la figure 2.

3.5 Ontologies différentes et alignement explicite

Ce cas, illustré ci-dessous, est similaire au cas précédent. La seule différence est qu'un alignement est disponible entre les ontologies et utilisé par l'outil. L'utilisateur n'a pas dans ce cas à décrire d'alignement.

5. `http://sw.opencyc.org/`

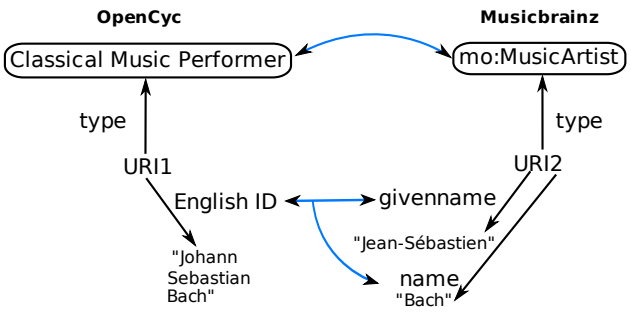
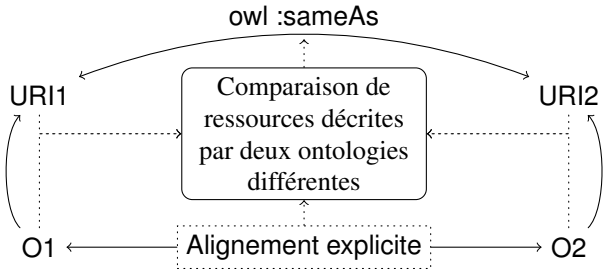


FIGURE 2 – Exemple de comparaison de ressources nécessitant un alignement entre ontologies.



Nous allons, dans la section suivante, analyser six systèmes d'alignement de données et les classer par rapport à ce cadre de travail.

4 Analyse des outils et langages de spécification d'alignement de jeux de données

Ce travail est le résultat d'une expérimentation menée conjointement avec la liste de diffusion *linking-open-data*. Nous avons proposé aux développeurs d'outils de liage de nous envoyer une spécification de liage afin que nous puissions les comparer et évaluer la possibilité de proposer une manière de les lier et de publier ces spécifications dans un langage commun. [6] Les systèmes étudiés ne sont donc que ceux ayant participé à cette initiative sachant qu'il en existe d'autres n'étant pas développés ici [16, 11].

Ces outils prennent en entrée deux jeux de données et une spécification indiquant comment lier les resources qu'ils contiennent. Certains jeux de données étant uniquement accessibles via un point d'entrée SPARQL, nous nous intéressons à la capacité des outils étudiés d'utiliser ce point d'accès en entrée. Une étape importante dans l'utilisation des outils de liage des données est la spécification des liens, c'est-à-dire une indication donnée à l'outil sur les resources qu'il doit examiner. En effet, vu la taille des jeux de données, l'espace de recherche peut atteindre plusieurs milliards de resources [5]. Des heuristiques sont donc nécessaires pour réduire l'espace de recherche. En sortie, un ensemble de liens est généré.

Ces outils se basent principalement sur des techniques d'alignement de chaînes de caractères, ainsi que sur des techniques plus élaborées utilisant des outils linguistiques et exploitant la structure de graphe des jeux de données.

6. Pour plus de détails : `http://melinda.inrialpes.fr`

4.1 Critères d'analyse

Pour chaque outil étudié nous posons un ensemble de questions énoncées ci-dessous. Nous décrivons plus bas chaque outil au regard de ces questions.

Degré d'automaticité

- L'outil est-il complètement automatique ? (boite noire)
- l'outil a-t'il besoin d'être paramétré par l'utilisateur ? Quel type de paramètres ? (alignement entre ontologies, techniques d'alignement des données)

Techniques d'alignement utilisées

- alignement de chaînes de caractères ?
- fonctions externes ? (conversions de valeurs, transformations de données)
- propagation de similarité ?
- autres techniques ?

Ontologies

- L'outil prend-il en compte les ontologies associées aux données ?
- L'outil permet-il d'aligner des jeux de données décrits en fonction d'ontologies différentes ?
- Dans le cas où elles sont différentes, l'outil aligne-t-il les ontologies ?

Sortie

- Qu'est-ce que l'outil produit en sortie ? (liens `owl:sameAs`, autre types de liens)
- L'outil propose-t-il de fusionner deux jeux de données ?

Ensembles de données Comment l'outil accède-t-il aux jeux de données ? (à travers un point d'accès SPARQL, à partir d'une URL, a partir d'une copie locale du jeu de données)

Domaine L'outil est-il spécifique à un certain domaine ?

Post-opératoire L'outil effectue-t-il des traitements post-opératoire ? (vérification de la consistence des jeux de données et des liens par rapport aux ontologies)

4.2 Outils

RKB-CRS Le système de résolution de co-référence (CRS) de la base de connaissances RKB [12] consiste en des listes d'équivalence entre URI. Ces listes sont construites en utilisant un programme Java sur mesure pour le domaine spécifique des conférences/universités. Un nouveau programme doit donc être réécrit pour chaque jeu de données. Chaque programme consiste à sélectionner les ressources à aligner, et à les comparer en appliquant des algorithmes de similarité de chaînes de caractères sur leurs attributs.

	Silk	**Knofuss**	**RKB CRS**	**RDF-AI**	**LD-Mapper**	**ODD**
Ontologies	multi	multi (not considered)	multi (not considered)	single	single	single
Automatisation	semi	semi	semi	semi	automatique	semi
Spec. utilisateur	specification liens methode d'alignement	fusion onto.	programme sur mesure	structure des donnèes methode d'alignement	non	spec. liens requète
Format entrée	Silk-LSL (XML)	OWL	Java	XML	prolog	LinQL
Techniques d'alignement	chaînes de car.	chaînes de car. apprentissage adaptatif	chaînes de car. alignement	chaînes de car. Wordnet	chaînes de car. propagation de similiarité	chaînes de car.
Alignment onto.	non	oui, en entrée	non	non	non	non
Sortie	linkset	format d'alignment jeu de données fusioné	`owl:sameAs`	format d'alignment linkset jeu de données fusioné	`owl:sameAs`	linkset
Accès aux données	SPARQL	copie locale	API	copie locale	copie locale	base ODBC
Domaine	multiple	multiple	publications	multiple	Music Ontology	multiple
Post-opératoire	non	résolution de consistence	non	non	non	non

TABLE 1 – Comparison des outils de liage de données.

LD-mapper LD-Mapper [15] est un outil d'intégration de jeux de données dans le domaine de la musique. Cet outil est basé sur un algorithme d'aggrégation de similarité prenant en compte la similarité des voisins d'une ressource dans le graphe la décrivant. Cet outil demande peu de configuration de la part de l'utilisateur mais ne fonctionne ainsi qu'avec des jeux de données décrits par la Music Ontology [7].

ODD-linker ODD-linker [10] est un outil de recherche d'équivalences implémenté à partir d'un outil de recherche d'équivalences entre enregistrements de bases de données relationelles. ODD-linker utilise des requêtes SQL pour identifier et comparer les resources à aligner. ODD-Linker consiste à traduire des spécifications de liens exprimés dans le langage LinQL en requètes SQL. ODD-linker est fait pour être utilisé avec des bases de données relationelles exportées en RDF.

RDF-AI RDF-AI [17] est une architecture pour l'alignement et la fusion de jeux de données. Une des particularités de l'architecture est de séparer l'alignement, en tant que spécification abstraite de liens existant entre les ressources des ensembles, de l'ensemble de liens concrets exprimés sous la forme d'un *linkset*. Cet outil génère donc un alignement qui peut être utilisé soit pour fusioner les deux jeux de données, soit pour générer un *linkset* contenant les triplets `owl:sameAs`. RDF-AI prend en entrée des fichiers XML spécifiant des paramètres pré-opératoire : réorganisation des noms, traduction ; spécifiant la structure du jeu de données, les techniques d'alignement à utiliser pour chaque type de ressources que l'on veut comparer ; des paramètres post-opératoire tel un seuil servant à générer le *linkset*, et des paramètres de fusion. Cet outil travaille avec une copie locale des jeux de données et est implémenté en Java.

Silk Silk [6] est paramétré par un langage de spécification de liens, le *Silk Link Specification Language*. L'utilisateur spécifie quelles entités doivent être reliées et indique la mesure de similarité à utiliser. Silk utilise plusieurs méthodes d'alignement de chaînes de caractères, des méthodes mesurant les similarité numériques, de dates, la distance entre concepts dans une taxonomie, et la similarité d'ensembles. Des transformations peuvent être spécifiées préparant le jeu de données avant le processus d'alignement afin d'en améliorer ses résultats. Silk prend en entrée deux jeux de données accessibles à travers un point d'accès SPARQL. Il fournit en sortie des triplets `owl:sameAs` ou d'autre prédicats spécifiés par l'utilisateur. Silk a été expérimenté sur plusieurs jeux de données indiqués sur la page de présentation du projet [8] et est implémenté en Python.

Knofuss L'architecture Knofuss [13] est destinée à fusionner des jeux de données. Une des particularités de Knofuss est sa capacité à aligner des jeux de données décrits en fonction d'ontologies hétérogènes. Le processus d'alignement des données est dirigée par une ontologie dédiée spécifiant les resources à comparer, ainsi que les techniques d'alignement adéquates à utiliser. Les resources sont sélectionnées en spécifiant des requêtes SPARQL adressées aux jeux de données. L'outil propose un ensemble d'algorithmes d'alignement de chaînes de caractères. Lorsque les deux jeux de données à aligner sont décrits par des ontologies différentes, il est possible de spécifier un alignement dans le format d'alignement [8], permettant d'utiliser un des nombreux systèmes d'alignement d'ontologies disponible. Knofuss étant à l'origine dédié à la fusion

7. `http://www.musicontology.com`

8. `http://www4.wiwiss.fu-berlin.de/bizer/silk/spec/`

Cas	Outils
Création de liens manuelle	
Mise en correspondance des identifiants	RKB-CRS
Ontologie commune	LD-Mapper, ODD-Linker
Ontologies différentes, alignement entre ontologies implicite	RDF-AI, Silk
Ontologies différentes, alignement entre ontologies explicite	Knofuss

TABLE 2 – Classification des outils analysés selon les cas

de données, il est possible de spécifier une stratégie de fusion. Une étape post-opératoire est effectuée par l'outil pour vérifier consistance du jeu de données résultant de l'opération de fusion. Cet outil utilise des copies locales des jeux de données et est implémenté en Java.

Chaque outil analysé correspond à une catégorie de systèmes présentée section 3 (voir table 2).

Il est clair de ce qui précède que les outils développés pour la recherche de liens, malgré leur diversité, relèvent du cadre proposé.

La figure 3 unifie tous les processus décrits ci-dessus dans une description unique. Il permet de clarifier les interactions entre liage de données et alignement d'ontologies. Il peut être utilisé comme base pour faire coopérer les outils. Ceci présenterait plusieurs avantages : en particulier il devient possible de distribuer, partager et améliorer les spécifications de liage. Il est aussi possible de les réutiliser ou les étendre au lieu de les recalculer à chaque fois qu'un jeu de données est modifié. Finalement, cela permettrait de composer les spécifications de liage. On considère ci-après la manière de mener cette intégration.

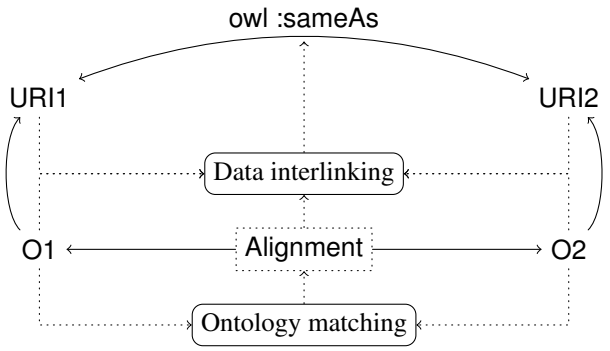


FIGURE 3 – Cadre général pour le liage de données impliquant l'alignement d'ontologies.

5 Liage et alignement

Il serait utile de tirer parti de cela pour que ces outils puissent interopérer. Cela aurait plusieurs avantages, en particulier la possibilité de partager, diffuser et améliorer les spécifications de liens ainsi que de pouvoir les réappliquer et/ou étendre les ensembles de liens déjà établis au lieu de les recalculer à chaque nouvelle publication des données. Cela permettrait aussi de composer ces spécifications afin de pouvoir passer d'un jeu de données directement à un autre sans passer par un ensemble intermédiaire. Pour cela, plusieurs pistes peuvent être proposées que l'on considère ci-après.

Bien que l'alignement d'ontologies et le liage de données peuvent être similaires dans un certain sens (tous deux relient des entités définies formellement), il y a d'importantes différences telles qu'illustrées dans le cadre précédent. En particulier, l'un agit au niveau du schéma alors que l'autre travaille au niveau des données. Ces différences sont réflètées dans le type de spécification de ces processus :

- Une assertion `sameAs` dit quelle `City` de wikipedia correspondent à quelle `P` (place) de geonames, e.g., Manchester sameAs Manchester.
- Une spécificaton de liage indique comment trouver des tels liens, par exemple pour lier une `City` à une `P`, évaluer comment l'étiquette (`label`) du premier est proche du nom (`name`) du second à l'aide d'une mesure appropriée (par exemple, `jaroSimilarity`), évaluer la proximité entre la `populationTotal` du premier `population` du second avec une autre mesure (par exemple, `numSimilarity`), calculer la moyenne des deux valeurs et si le résultat est supérieur à .9, alors engendrer le lien `sameAs`.
- Un alignement d'ontologies précise quels composants d'une ontologie correspond à quels composants de l'autre ontologie. Par exemple, `dbpedia:City` est une sorte de `geonames:P` et dans ce contexte, `label` est équivalent à `name` et `populationTotal` est équivalent à `population`.

Ceci résulte en deux spécification de processus – liage et alignement – et leurs résultats – linksets entre données et alignements entre ontologies. Cette situation est résumée dans la table 3.

	processus	résultat
instance	specification de liage	linkset
classe	matcher	alignment

TABLE 3 – Liage et processus d'alignement ainsi que leurs résultats.

En établissant ces différences, on obtient une partition naturelle entre les liens (ou linkset), les spécifications de liage et les alignments d'ontologies ainsi qu'entre les langages pour les spécifier :

Langage d'expression de liens Il permet d'exprimer

l'équivalence entre resources à l'aide de `owl:sameAs` (par exemple, RDF and VoiD) ;

Langage de spécification de lieurs Il permet de définir comment chercher les règles d'alignements entre instances (par exemple, Silk LSL) ;

Langage d'expression d'alignements Il permet de spécifier les règles d'équivalence de ressources (par exemple le format d'alignement ou EDOAL).

Ainsi, alignement d'ontologies et liage de données restent différents. Cependant, liage et données et alignement d'ontologies peuvent tirer bénéfice de collaborer. On propose a mécanisme par lequel les outils de liage peuvent s'appuyer sur des alignements pour contraindre leur espace de recherche. Le cadre de la section précédente, tel que présenté dans la figure 3 constitue un moyen naturel d'implémenter cette coopération.

5.1 Un langage d'alignement étendu

Dans un premier temps, on peut imaginer étendre le langage d'alignement que nous avons développé [8]. Il a l'avantage d'être déclaratif et permet d'autre part d'exprimer des transformations du types de celles qui peuvent être nécessaires pour engendrer les liens. Ainsi, dans le langage expressif EDOAL (*Expressive Declarative Ontology Alignment Language*, anciennement appelé *OMWG mapping language*) [9], il est possible d'exprimer que deux classes sont équivalentes mais que les instances de ces classes sont équivalentes modulo transformation à l'aide d'expressions régulières dans le cas de la figure 4.

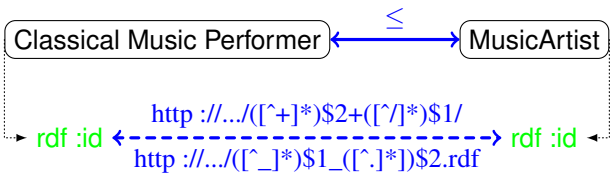


FIGURE 4 – Expression d'une équivalence de lien dans un langage d'alignement expressif.

Si celà permet d'exprimer déclarativement le résultat de l'alignement, cela ne permet pas de produire l'alignement. Or, la fonction de plusieurs outils considérés ci-dessus est de définir les algorithmes à adopter pour mettre en correspondance les URI d'une classe particulière. Il est imaginable d'intégrer ceci dans un langage d'alignment. Par exemple, en remplaçant les expressions régulières sur la flêche de la figure 4 par une spécification d'alignement de Silk.

5.2 Un langage de description de lieurs

L'autre approche est dictée par Silk qui dispose d'un langage relativement précis et lisible pour spécifier un algorithme de liage.

Nous présentons ci-dessous un exemple de spécification inspiré de l'outil Silk pour lier les données entre deux jeux de données décrivant des artistes musicaux.

```
<SourceDataset dataSource="OpenCyc" var="a">
 <RestrictTo>
  ?a rdf:type cyc:ClassicalMusicPerformer
 </RestrictTo>
 </SourceDataset>
 <TargetDataset dataSource="MusicBrainz" var="b">
  <RestrictTo>
   ?b rdf:type mo:MusicArtist
  </RestrictTo>
 </TargetDataset>
 <LinkCondition>
  <Compare metric="jaroSimilarity">
   <Param name="str1" path="?a/cyc:EnglishID" />
   <Param name="str2" path="?b/foaf:name" />
  </Compare>
 </LinkCondition>
```

On peut remarquer que les restrictions à `ClassicalMusicPerformer` et `MusicArtist` constituent en fait un alignement entre ces deux concepts. De même pour la comparaison entre les propriétés `EnglishID` et `name`. L'alignement pour ces correspondances est donné ci-dessous dans le langage EDOAL.

```
:cyc-mo a align:Alignment;
align:onto1 <http://opencyc.org/>
align:onto2 <http://www.musicontology.com>
   align:map [
  align:entity1 cyc:ClassicalMusicPerformer;
  align:entity2 mo:MusicArtist;
  align:relation align:subsumedBy.
];
   align:map [
  align:entity1 foaf:name;
  align:entity2 cyc:englishID;
  align:relation align:subsumes.
].
```

Ci-dessous notre proposition de simplification pour une spécification de liage reprenant la syntaxe de Silk et utilisant l'alignement.

```
<SourceDataset dataSource="OpenCyc" var="a">
<RestrictTo>
  ?a rdf:type cyc:ClassicalMusicPerformer
 </RestrictTo>
 </SourceDataset>
 <TargetDataset dataSource="MusicBrainz"/>
 <Alignement source="http://www.ex.org/cyc-mo"/>
 <LinkCondition>
  <Compare metric="jaroSimilarity">
   <Param name="str1" path="?a/cyc:EnglishID" />
  </Compare>
 </LinkCondition>
```

Il n'est alors plus nécéssaire d'exprimer dans la spécification de liens que des entités correspondant au jeu de données source, les cibles seront retrouvées en consultant l'alignement.

6 Conclusion

Dans le but d'offrir un cadre général dans lequel les candidats à la publication de leurs données puissent exprimer plus naturellement leurs techniques de liage, nous avons étudié les différentes techniques mises en œuvre actuellement. Il en ressort que :

– au delà des disparités entre les différentes techniques, il est possible de définir un cadre général dans lequel elles s'insèrent offrant plus ou moins de déclarativité dans leur facilité d'expression (qui va du programme Prolog à la composition de "matcheurs")
– bien que le parallèle avec l'alignement d'ontologie soit pertinent, il ne semble pas raisonnable de considérer intégrer tout dans un seul langage, en particulier car leurs destinations sont différentes.

Nous proposons donc d'utiliser une architecture fondée sur trois langages différents disposant chacun d'une tâche bien précise : exprimer les liens, les alignements entre ontologies et les techniques pour produire les liens.

Nous travaillons actuellement à l'implémentation du langage expressif de representation d'alignement entre ontologies EDOAL afin de permettre à celui-ci d'exprimer les techniques de liage de données les plus simples. Nous travaillons aussi à une extension de Silk prenant en compte les alignements entre ontologies. Cette extension devrait permettre une meilleure automatisation du liage de données, aidant ainsi à renforcer les interrelations des données sur le web.

Références

[1] Ben Adida and Mark Birbeck. RDFa primer. W3C working draft, W3C, June 2008. http ://www.w3.org/TR/2008/WD-xhtml-rdfa-primer-20080620/.

[2] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing linked datasets - on the design and usage of voiD, the 'vocabulary of interlinked datasets'. In *Linked Data on the Web Workshop (LDOW09), Workshop at 18th International World Wide Web Conference (WWW09)*, Madrid, Spain, 2009.

[3] Tim Berners-Lee. Linked-data design issues. W3C design issue document (on-line), `http://www.w3.org/DesignIssues/LinkedData.html`, 06 2009.

[4] Christian Bizer. D2R MAP - a database to RDF mapping language. In *Proc. 12th WWW conference poster session*, 2003.

[5] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia : a crystallization point for the web of data. *Journal of web semantics*, 7(3) :154–165, 2009.

[6] Christian Bizer, Julius Volz, Georgi Kobilarov, and Martin Gaedke. Silk - a link discovery framework for the web of data. In *18th International World Wide Web Conference*, April 2009.

[7] Paolo Bouquet, Heiko Stoermer, and Barbara Bazzanella. An Entity Naming System for the Semantic Web. In *Proceedings of the 5th European Semantic Web Conference (ESWC2008)*, LNCS, June 2008.

[8] Jérôme Euzenat. An API for ontology alignment. In Frank van Harmelen, Sheila McIlraith, and Dimitri Plexousakis, editors, *The Semantic Web - ISWC 2004 : Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298, pages 698–712. Springer, 2004.

[9] Jérôme Euzenat, François Scharffe, and Antoine Zimmermann. D2.2.10 : Expressive alignment language and implementation. Project deliverable 2.2.10, Knowledge Web NoE (FP6-507482), 2007.

[10] Oktie Hassanzadeh, Lipyeow Lim, Anastasios Kementsietsidis, and Min Wang. A declarative framework for semantic link discovery over relational data. In *WWW '09 : Proceedings of the 18th international conference on World wide web*, pages 1101–1102, New York, NY, USA, 2009. ACM.

[11] Aidan Hogan, Andreas Harth, and Stefan Decker. Performing object consolidation on the semantic web data graph. In *In Proceedings of 1st I3 : Identity, Identifiers, Identification Workshop*, 2007.

[12] Afraz Jaffri, Hugh Glaser, and Ian Millard. Managing URI synonymity to enable consistent reference on the semantic web. In *IRSW2008 - Identity and Reference on the Semantic Web 2008 at ESWC*, 2008.

[13] Andriy Nikolov, Victoria Uren, Enrico Motta, and Anne de Roeck. Handling instance coreferencing in the KnoFuss architecture. In *Proceedings of the workshop : Identity and Reference on the Semantic Web at 5th European Semantic Web Conference (ESWC 2008)*, 2008.

[14] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The music ontology. In *Proceedings of the International Conference on Music Information Retrieval*, 2007.

[15] Yves Raimond, Christopher Sutton, and Mark Sandler. Automtic interlinking of music datasets on the semantic web. In *Proceedings of the Linking Data On the Web workshop at WWW'2008*, 2008.

[16] Fatia Sais, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. *Journal of Data Semantics*, 12, 2008.

[17] François Scharffe, Yanbin Liu, and Chunguang Zhou. RDF-AI : an architecture for RDF datasets matching, fusion and interlink. In *Workshop on Identity and Reference in Knowledge Representation, IJCAI 2009*, 2009.

[18] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *WWW*, pages 585–594, 2006.

[19] Julius Volz, Christian Bizer, and Martin Gaedke. Web of data link maintenance protocol. Protocol specification, Frei Universität Berlin, 2009.