MeLinDa: an interlinking framework for the web of data

François Scharffe, Jérôme Euzenat

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

MeLinDa:
an interlinking framework for the web of data

François Scharffe — Jérôme Euzenat

N° 7691

July 2011

Knowledge and Data Representation and Management

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7691--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

MeLinDa:
an interlinking framework for the web of data

François Scharffe*, Jérôme Euzenat†

Theme : Knowledge and Data Representation and Management
Perception, Cognition, Interaction
Équipes-Projets exmo

Rapport de recherche n° 7691 — July 2011 — 21 pages

Abstract: The web of data consists of data published on the web in such a way that they can be interpreted and connected together. It is thus critical to establish links between these data, both for the web of data and for the semantic web that it contributes to feed. We consider here the various techniques developed for that purpose and analyze their commonalities and differences. We propose a general framework and show how the diverse techniques fit in the framework. From this framework we consider the relation between data interlinking and ontology matching. Although, they can be considered similar at a certain level (they both relate formal entities), they serve different purposes, but would find a mutual benefit at collaborating. We thus present a scheme under which it is possible for data linking tools to take advantage of ontology alignments.

Key-words: Semantic web, data interlinking, instance matching, ontology alignment, web of data

This work as been partially published as [Scharffe and Euzenat, 2010].

* LIRMM, Montpellier, France `Francois.Scharffe@lirmm.fr`; part of this work was achieved when this author was at INRIA Grenoble Rhône-Alpes.
† INRIA & LIG, Montbonnot, France `Jerome.Euzenat@inria.fr`

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

MeLinDa:
un cadre pour le liage des données du web

Résumé : Le web des données consiste à publier des données sur le web de telle sorte qu'elles puissent être interprétées et connecttées 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 considérons les diverses techniques proposées à cette fin et analysons leurs similarités et différences. Nous proposons un cadre général dans lequel s'inscrivent les différentes techniques utilisées et nous montrons comment elles s'y insèrent. Ce cadre permet de considérer les relations entre le liage de données et l'alignement d'ontologies: bien que ces activités puissent être considérées comme similaires (elles trouvent les relations entre entités), elles n'ont pas le même but mais bénéficieraient à collaborer. Nous proposons une architecture permettant aux outils de liage de tirer parti des alignements entre ontologies.

Mots-clés : Web sémantique, liage de données, alignement d'ontologies, web des données

1 Introduction

The web of data is the network resulting from publishing structured data sources in RDF and interlinking these data sources with explicit links. A large quantity of structured data is being published particularly through the Linking Open Data project[1]. Web data sets are expressed according to one or more vocabularies or ontologies, which range from simple database schema exposure to full-fledged ontologies.

The web of data requires to interlink the various published data sources. Given the large amount of published data, it is necessary to provide means to automatically link those data. Many tools were recently proposed in order to solve this problem, each having its own characteristics (see Section 4).

In many cases, data sets containing similar resources are published using different ontologies. Hence, data interlinking tools need to reconcile these ontologies before finding the links between entities. This could be done automatically, but more often this is done manually and built in the link specifications. This has two drawbacks: (a) this prevents to reuse the work made in ontology matching for reconciling ontologies, and (b) the information about reconciling the ontologies is mixed with the information about how to identify entities.

Hence, the goal of this work is to analyse existing interlinking tools and to determine (1) how they fit in the same framework, (2) if it is possible to define a language for specifying the linking techniques to be used, and (3) how is data interlinking related to ontology matching. This report contributions are as follows:

– A comprehensive survey of existing data interlinking tools,
– A characterization of task/problem categories for web data set interlinking,
– A proposal for improving data interlinking tools with ontology alignments.

For that purpose, after briefly introducing the challenges of data interlinking and ontology matching (Section 2), we provide a general framework for data interlinking in which all these tools can be included (Section 3). From this analysis, we review six data interlinking tools and the way they are built (Section 4). This framework clearly separates the data interlinking and ontology matching activities and we show how these can collaborate through three different languages for links, data linking specifications and ontology alignments (Section 5). We provide examples of an expressive alignment language (Section 6) and a linking specification language (Section 7). Finally, we show how these two languages can be adapted for cooperating (Section 8).

2 Web of data, data interlinking, and ontology alignment

We briefly introduce linked data and the data interlinking problem. We provide examples of this problem and why it would require specific linking tools. We then present why these tools could take advantage of ontology matching and alignments.

2.1 Linked data

The web of data is based on the following four principles [Berners-Lee, 2009; Heath and Bizer, 2011]:

1. Resources are identified by URIs.
2. URIs are dereferenceable.
3. When a URI is dereferenced, a description of the identified resource should be returned, ideally adapted through content negotiation.
4. Published web data sets must contain links to other web data sets.

[1] `http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`

As long as they follow these rules, *linked data* can be published in various ways (RDF data sets, SPARQL endpoints, XHTML+RDFa pages [Adida *et al.*, 2008], databases exposed through HTTP [Bizer, 2003; Sahoo *et al.*, 2009]). Web data sets can also be constructed collaboratively, through the use of specialized tools [Völkel *et al.*, 2006].

2.2 The data interlinking problem and linksets

A main problem on the web of data is to create links between entities of different data sets. Most often, this consists of identifying the same entity across different data sets and publishing a link between them as a `owl:sameAs` statement (shortened as `sameAs` hereafter). We call this task data interlinking and summarize it in Figure 1.

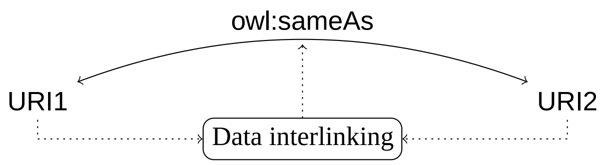

Figure 1: The data interlinking problem.

Once identified, the links discovered between two data sets must also be published in order to be reused. The VoiD vocabulary [Alexander *et al.*, 2009] allows for describing *linksets* as special data sets containing links between resources of two given data sets. A linkset is represented as an RDF named graph described using VoiD annotations, as shown in the RDF/N3 code below:

```
{
 <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/resource/Johann_Sebastian_Bach>
 owl:sameAs
 <http://www.musicbrainz.org/artist/24f1766e-9635-4d58-a4d4-9413f9f98a4c> .
}
```

Once linksets are constructed, two approaches are proposed to retrieve equivalences between resources: it is possible to assign to each real world entity a global identifier that will then be related to every URIs describing this entity. This is the approach taken in the OKKAM project [Bouquet *et al.*, 2008] that proposes the usage of Entity Name Servers taking the role of resource name repositories. The other approach uses equivalence lists maintained with interlinked resources across data sets. There is thus no global identifier in this approach but equivalence links can be followed using a third-party web service, e.g., `http://sameas.org`, or a bilatteral protocol [Volz *et al.*, 2009].

The *data interlinking* task can be achieved manually or with the help of data interlinking tools. These tools take as input two data sets and ultimately provide a linkset. In addition, they use what we call a *linking specification*, i.e., a “script” specifying how and/or what to link. Indeed, given data set sizes, the search space for resources interlinking can reach several billion resources[2]. It is thus necessary to use heuristics giving hints to the interlinking system about where to look for the corresponding resources in the two data sets. These linking specifications can be specific to a pair of

[2]4.2 billion RDF triples related by 142 million links: source Wikipedia, May 2009.

data sets and can be reused for regenerating linksets (we provide an example of such a specification in the Silk language in Section 7).

2.3 Interlinking data sets

Mining for similar resources in two web data sets raises many problems. Each data set having its own namespace, resources in different data sets are given different URIs. Also, although naming conventions exist [Sauermann and Cyganiak, 2008], there is no formal nor standard way of naming resources. For example, if we take the URI for the famous musician Johann Sebastian Bach in various web data sets we obtain different results though they all represent the same real world object (Table 1).

Dataset	URI
MusicBrainz	http://musicbrainz.org/artist/24f1766e-9635-4d58-a4d4-9413f9f98a4c
LastFM	http://www.last.fm/music/Johann+Sebastian+Bach
DBpedia	http://dbpedia.org/resource/Johann_Sebastian_Bach
OpenCyc	http://sw.opencyc.org/concept/Mx4rwJw6npwpEbGdrcN5Y29ycA

Table 1: Varying URIs across different data sets.

As this example demonstrates, URIs are different across data sets, both because of their namespaces and because of their fragments. Fragments are generated according to two strategies: an internal ID as for MusicBrainz and OpenCyc, or the concatenation of some of the resource properties, as for LastFM and DBpedia. When the first strategy is used, an interlinking system might not be able to find correspondences between two resources by looking at URIs only.

Fortunately, dereferencing URIs can be used for retrieving more information about entities: property values and related resources can be observed. But for the same real-world entity, the same property can take different values, making the interlinking process more difficult. This can be because of varying value approximations across data sets, because of different units of measure, because of mistakes in the data sets, or because of loose ontological specifications. For instance, the property `foaf:name` does not specify in what format should the name be given. “J.S. Bach”, “Bach, J.S.” or “Johann Sebastian Bach” are possible values for this property. Hence, data interlinking tools have to compare property values in order to decide if two entities are the same, and must be linked, or not. For that purpose, tools use similarity measures based on the type of values, e.g., string, numbers, dates, and aggregate the results of these measures. This activity is reminiscent of *record linkage* in database which has been given considerable attention [Fellegi and Sunter, 1969; Winkler, 2006; Elmagarmid *et al.*, 2007]. The tools studied in Section 4 reuse many of the record linkage techniques.

Another problem is caused by the usage of heterogeneous ontologies for describing data sets. In this case, a same resource is typed according to different classes and described with different predicates belonging to different ontologies. For example, a name in a data set can be attributed using the `foaf:name` data property from the FOAF ontology while it is attributed using the `vcard:N` object property from the VCard ontology in another data set.

Hence, for the interlinking techniques to work, it is necessary that the data sets use the same ontology or that data interlinking tools are aware of the correspondences between ontologies.

2.4 Ontology matching and alignment

Ontology matching allows for finding correspondences between ontology entities [Euzenat and Shvaiko, 2007]. The result of this process is called an *ontology alignment*. Once the ontologies matched, the alignment can be stored and retrieved when an application needs to use data described according to another ontology [Euzenat *et al.*, 2007a].

Matching ontologies requires to overcome the mismatches originating from the different conceptualizations of the domains described by ontologies [Visser *et al.*, 1997; Klein, 2001]. These mismatches may be of different nature: terminological mismatches concern differences of naming such as the usage of synonym terms for concept labeling; conceptual mismatches concern different conceptualizations of the domain such as structuring along different properties; structural mismatches concern heterogeneous structures, like different granularities in the class hierarchies. Ontology matching is similar to database schema matching [Rahm and Bernstein, 2001]. Specific works on ontology matching were proposed in the last ten years [Noy and Musen, 2000] that now reach maturity [Euzenat and Shvaiko, 2007]. It is not the purpose of this paper to describe any particular technique.

While different URI constructions and variations of property values can find automatic solutions, the problem of having heterogeneous ontologies is in most interlinking tools solved by manually specifying the correspondences (see Table 2, Section 4). This considerably complexifies the interlinking process. Ontology matching techniques can be used to facilitate the interlinking task and ontology alignments reused in linking specifications.

The goal of this paper is to investigate the relationships between data interlinking and ontology matching. In particular, we want to understand if these two activities should be merged into a single activity and share the same formats or if there are good reasons to keep them separated. In the second perspective, we also want to establish how they can benefit from one another. For that purpose, we analyzed available systems for data interlinking.

3 A framework for data interlinking

We provide, in this section, a general framework encompassing the various approaches used to interlink resources on the web of data.

In the most general case illustrated in Figure 1, two web data sets are interlinked using a method for comparing their resources. We do not specify at this stage if the method should be automatic or manual. Neither do we specify if the two data sets are described using a common ontology or if the ontologies describing their resources differ.

This is the goal of the following subsections to consider this. We first consider each case that may happen when interlinking data and describe them abstractly and through an example. In the end, we unify all this cases in a common framework.

3.1 Manual interlinking

In the first case, illustrated in Figure 2, resources are manually interlinked.

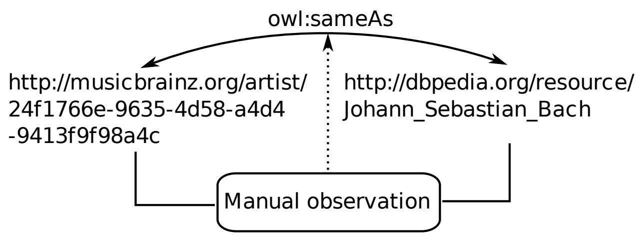

Figure 2: Example of manually linked resources.

Manually linking resources can be performed using collaborative tools in the case of large data sets.

3.2 URI correspondence.

In some cases, illustrated in Figure 3, resources can be trivially linked using a simple transformation of their URIs.

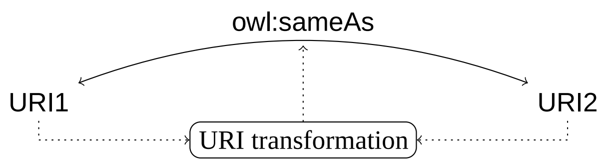

Figure 3: Data interlinking through URI transformation.

A set of rules can be defined to identify equivalent resources from their identifier. For example, in the data set LastFM[3], the URI representing an artist is built on the pattern "First_name+Last_name". Person URIs in DBpedia[4] are built around the pattern "FirstName_LastName". A trivial algorithm can be developed to find equivalent artists based on their URIs. This is illustrated in Figure 4 for the composer J.S. Bach.

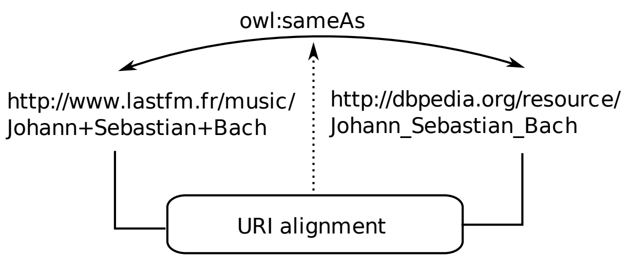

Figure 4: Example of resource linking using the correspondence between URIs.

3.3 Datasets sharing the same ontologies

Further than URIs, it may be necessary to consider the ontologies in order to identify entities. In a first case, illustrated in Figure 5, the two data sets to interlink are described by the same ontology. The role of the interlinking system is to analyze resources of the same type in order to detect the equivalent ones. To do this, the system compares resource properties with a similarity measure. Systems in this category take as input the properties to compare, the type of comparison algorithm to use for each property, and the method to aggregate the similarity measures of the various properties in order to construct a measure between two resources.

For example, Jamendo[5] and MusicBrainz[6], two data sets containing musicological data, are both described according to a common music ontology [Raimond *et al.*, 2007]. The artist J.S. Bach can

[3] `http://last.fm`
[4] `http://dbpedia.org`
[5] `http://www.jamendo.com`
[6] `http://www.musicbrainz.org`

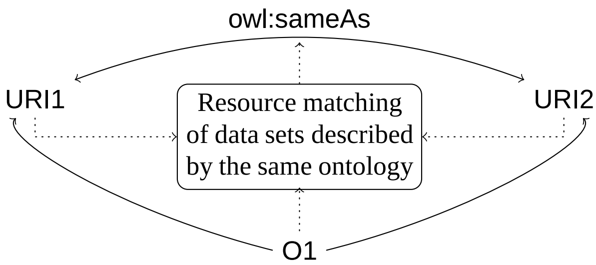

Figure 5: Interlinking two data sets described according to the same ontology.

be identified in both data sets by observing the first name and last name properties of the class *MusicArtist*. It is not possible in this case to identify the equivalence of resources based on their URIs. This example is illustrated in Figure 6.

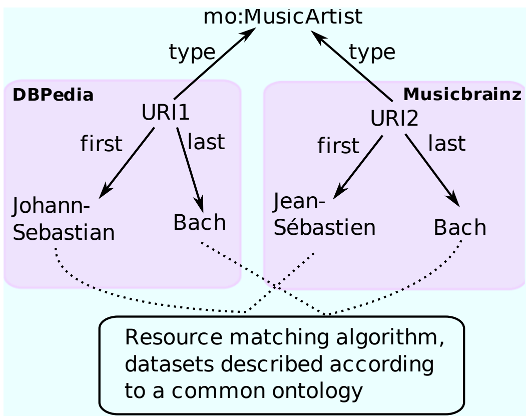

Figure 6: Example of linking resources described according to the same ontology.

3.4 Datasets described with heterogeneous ontologies

Datasets can be described by different ontologies. This case is illustrated in Figure 7. In order to know which types of entities have to be linked together, the system needs to know the correspondences between these types of entities. Then it can work similarly as if there were a single ontology.

We represent this case in Figure 7 by introducing the correspondences between ontology classes as an alignment. This alignment is presented as implicit because it does not exist as such, but it is mixed with the linking specification or the data interlinking system.

Consider two data sets, one described using FOAF, the other using VCard. The linking specification will indicate to the tool to compare entities of type `foaf:Person` and entities of type `vcard:VC`, and that when comparing resources of these types, the properties `foaf:givenname` should be compared to `vcard:fn`, as well as the property `foaf:familyname` compared to the property `vcard:ln`. This is an implicit alignment containing two correspondences.

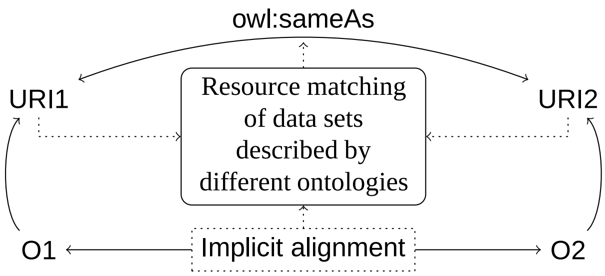

Figure 7: Two data sets interlinked using an implicit alignment.

For example, OpenCyc[7] represents the artist J.S. Bach using a different ontology than the one used to describe MusicBrainz. The properties "firstname" and "lastname" correspond to a property "EnglishID" in which both names are concatenated. The class *MusicArtist* in the Music Ontology corresponds to a class *Classical Music Composer* in OpenCyc. An alignment between classes and properties needs to be specified in order to find an equivalence between the two resources. This example is illustrated in Figure 8.

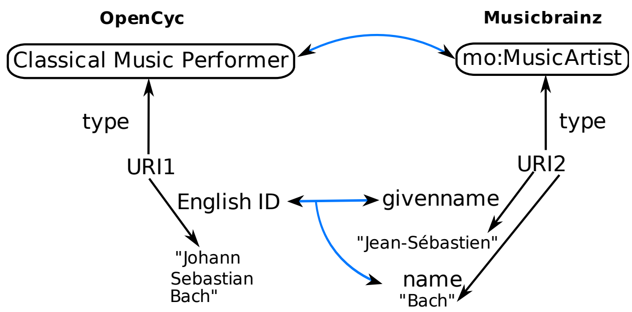

Figure 8: Example of two data sets described with heterogeneous ontologies.

3.5 Data interlinking with alignments

Another approach, illustrated in Figure 9, takes advantage of an already existing explicit alignment between the two ontologies used by the data sets.

An additional possibility, not found in existing systems, would be for the data linking system to first match the two ontologies before using the resulting alignment for supporting data interlinking. In such a system, ontology matching and data interlinking would be merged.

Figure 10 unifies all these processes in a single description. This framework leads to clarify interactions between data interlinking and ontology matching.

The next section discusses different systems and their position with respect to the proposed framework.

[7] `sw.opencyc.org`

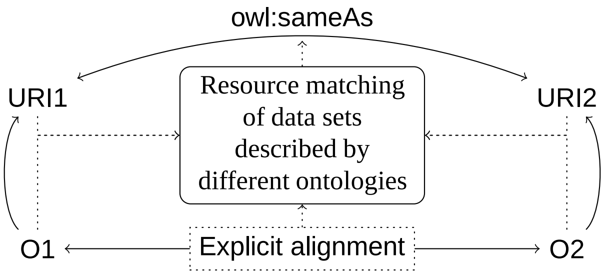

Figure 9: Two data sets matched using an explicit alignment.

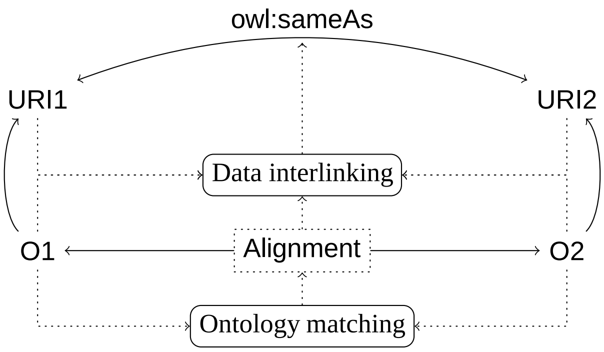

Figure 10: General framework for data interlinking involving ontology matching.

4 Data interlinking tool analysis

The work presented in this section is the result of the MeLinDa experiment conducted jointly with the linked open data mailing list. We asked interlinking tool developers to send us the linking specifications their tools take as input. We then compared these specifications and evaluated the possibility to publish them in a common language[8]. Six systems took part in the experiment. We are aware of at least two other systems not analyzed in this study [Saïs *et al.*, 2008; Hogan *et al.*, 2007].

We present below different criteria along which the tools can be compared, then we briefly describe the specifics of each tool and provide comparison of them along the criteria (Table 2).

4.1 Analysis criteria

For each analyzed tool, we tried to answer several questions reproduced below. We will then describe and categorize each tool according to these questions.

Degree of automation

- Is the tool completely automatic (a black box)?
- Does the tool need to be parametrized by the user? What kind of parameters (data matching techniques, ontology alignment)?

Used matching techniques

[8] `http://melinda.inrialpes.fr`. MeLinDa stands for Meta-Linking Data.

- String matching?
- External functions (values conversion, data transformations)?
- Similarity propagation?
- Other techniques?

Access

- How does the tool access data? (SPARQL endpoint, RDF Dump, URL)

Ontologies

- Does the tool take into account ontologies associated to the data sets?
- Does the tool allow to interlink data sets described according to different ontologies?
- If the ontologies differ, does the tool perform ontology matching?

Output

- What does the tool produce as output (`sameAs` links, VoiD linkset, other type of links)?
- Does the tool propose to merge the two input data sets?

Domain Is the tool specific for a given domain?

Post-processing Does the tool perform any post-processing operations (consistency checking and inconsistency resolution)?

4.2 Tools

We considered the 6 following tools. Table 2 summarizes the analysis.

4.2.1 RKB-CRS

The co-reference resolution system (CRS) of the RKB knowledge base [Jaffri *et al.*, 2008] is built around URI equivalence lists. These lists are built using a Java program working on the specific domain of universities and conferences. A new program needs to be written for each pair of data sets to integrate. Each program consists of the selection of the resources to compare, and their comparison using string similarity on the resource property values.

4.2.2 LD-Mapper

LD-Mapper [Raimond *et al.*, 2008] is a data set integration tool working on the music domain. This tool is based on a similarity aggregation algorithm taking into account the similarity of a resource's neighbors in the graph describing it. It requires little user configuration but only works with data sets described with the Music Ontology [Raimond *et al.*, 2007]. LD-Mapper is implemented in Prolog.

4.2.3 ODD-linker

ODD-Linker [Hassanzadeh *et al.*, 2009] is an interlinking tool implemented on top of a tool mining for equivalent records in relational databases. ODD-Linker uses SQL queries for identifying and comparing resources. The tool translates link specifications expressed in the LinQL dedicated language originally developed for duplicate records detection in relational databases. Its usage in the context of linked data is thus limited to relational databases exposed as linked data. LinQL is nonetheless an expressive formalism for link specifications. The language supports many string matching algorithms, hyponyms and synonyms, conditions on attribute values.

	RKB CRS	**LD-Mapper**	**ODD**	**RDF-AI**	**Silk**	**Knofuss**
Ontologies	multiple	multiple	single	single	single	multiple
Automation	semi-automatic	automatic	semi-automatic	semi-automatic	semi-automatic	semi-automatic
User input	program	none	link spec. query	data set structure alignment method	links spec. alignment method	fusion onto.
Input format	Java	prolog	LinQL	XML	Silk-LSL (XML)	OWL
Matching techniques	string	string, similarity propagation	string	string, Wordnet	string	string, adaptive learning
Onto. alignment	no	no	no	no	no	yes, in input
Output	owl:sameAs	owl:sameAs linkset	linkset	alignment format merged data set	alignment format linkset	alignment format merged data set
Data access	API	local copy	ODBC	local copy	SPARQL	local copy
Domain	publications	Music Ontology	independent	independent	independent	independent
Post-processing	no	no	no	no	no	inconsistency resolution

Table 2: Comparison of data linking tools.

4.2.4 RDF-AI

RDF-AI [Scharffe *et al.*, 2009] is an architecture for data set matching and fusion. It generates an alignment that can be later used either to generate a linkset, or to merge two data sets. The interlinking parameters of RDF-AI are given in a set of XML files corresponding to the different steps of the process. The data set structure and the resources to match are described in two files. This description corresponds to a small ontology containing only resources of interest and the properties to use in the matching process. A pre-processing file describes operations to perform on resources before matching. Translation of properties and name reordering are performed before looking for links. A matching configuration file describes which techniques should be used for which resources. A threshold for generating the linkset from the alignment can be specified. Additionally, when data sets need to be merged, a configuration file describes the fusion method to use. The prototype works with a local copy of the data sets and is implemented in Java.

4.2.5 Silk

Silk [Bizer *et al.*, 2009] is an interlinking tool parametrized by a link specification language: the Silk Link Specification Language (Silk LSL, see §7). The user specifies the type of resources to link and the comparison techniques to use. Datasets are referenced by giving the URI of the SPARQL endpoint from which they are accessible. A named graph can be specified in order to link only resources belonging to this graph. Resources to be linked are specified using their type, or the RDF path to access them. Silk uses many string comparison techniques, numerical and date similarity measures, concept distances in a taxonomy, and set similarities. A condition allows for specifying the matching algorithm used to match resources. Matching algorithms can be combined using a set of operators (MAX, MIN, AVG) and literals can be transformed before the comparison by specifying a transformation function, concatenating or splitting resources. Regular expressions can be be used to preprocess resources. Silk takes as input two web data sets accessible behind a SPARQL endpoint. When resources are matched with a confidence over a given threshold, the tool outputs `sameAs` links or any other RDF predicate specified by the user. The first version of Silk was implemented in Python; version 2 is a new implementation in Scala.

4.2.6 Knofuss

The Knossos architecture [Nikolov *et al.*, 2008] aims at providing support for data set fusion. A specificity of Knofuss is the possibility to use existing ontology alignments. The resource comparison process is driven by a dedicated ontology for each data set specifying resources to compare, as well as the comparison techniques to use. Each ontology gives, for each type of resource to be matched, an *application context* defined as a SPARQL query for this type of resource. An *object context model* is also defined to specify properties that will be used to match these resource types. Corresponding application contexts are given the same ID in the two ontologies and one application context indicates which similarity metric should be used for comparing them. When the two data sets are described using different ontologies, an ontology alignment can be specified. This alignment is given in the ontology alignment format [David *et al.*, 2011]. Knofuss allows for exporting links between data sets, but was originally designed to merge equivalent resources. It includes a consistency resolution module which ensures that the data sets resulting from the fusion of the two data sets is consistent with respect to the ontologies. The parameters of the fusion operation are also given in the input ontologies. Knofuss works with local copies of the data sets and is implemented in Java.

An analysis of these tools according to the criteria of Section 4.1 is summarized in Table 2. Obviously there is a lot of variation between these tools in spite of their common goal. Even if they are very diverse, each of these data interlinking tools fit in the proposed framework as shown on Table 3.

Case	Tool
Manual link specification (§3.1)	
URI correspondence (§3.2)	RKB-CRS
Common ontology (§3.3)	LD-Mapper, ODD-Linker
Different ontologies, implicit alignment (§3.4)	RDF-AI, Silk
Different ontologies, explicit alignement (§3.5)	Knofuss

Table 3: Classification of analyzed tools with regard to the framework.

The goal of the next section is to consider how using ontology alignments could lead to more automation for the interlinking task, as well as how linked data could provide evidence for obtaining better ontology alignments.

5 Matching/linking cooperation

Although ontology matching and data interlinking can be similar at a certain level (they both relate formal entities), there are important differences: one acts at the schema level and the other at the instance level. In fact, ontology matching can take advantage of linked data as an external source of information for ontology matching, and, conversely, data interlinking can benefit from ontology matching by using correspondences to focus the search for potential instance level links.

These differences are reflected in the types of specification involved in these processes:

- A link, e.g., a `sameAs` statement, tells which `City` in wikipedia correspond to which `P` (place) in geonames, e.g., Manchester sameAs Manchester.
- A linking specification tells how to find the former, e.g., for linking a `City` to a `P`, evaluate how the `label` of the first one is close to the `name` of the second one with some measure, e.g., `jaroSimilarity`, evaluate how the `populationTotal` of the first one is close to the `population` of the second one with another measure, e.g., `numSimilarity`, average the two values and if the result is above .9, then generate the `sameAs` statement.
- An ontology alignment tells which components from one ontology corresponds to which components in the other. For example, `dbpedia:City` is a kind of `geonames:P` and in this context, `label` is equivalent to `name` and `populationTotal` is equivalent to `population`.

This results in two process specifications – interlinking and matching – and their results – linksets between data and alignments between ontologies. The situation is summarized by Table 4.

	process	result
instance	linking specification	linkset
class	matcher	alignment

Table 4: Interlinking and matching processes and their results.

By clearly establishing these differences, we obtain a natural partitioning between data links, linking specifications and ontology alignments and the languages for expressing them:

The assertion expression language (e.g., RDF and VoiD) allows for representing equivalence between resources in data sets;

The linking specification language (e.g., Silk, LinQL) allows for defining how to search for equivalence between resources;

The alignment representation language (e.g., the Alignment format or EDOAL) allows for specifying equivalence rules between ontological entities.

It would be useful to take advantage of the framework of Section 3, to help tools interoperate. This would present many advantages, in particular the possibility to share, distribute and improve linking specifications, as well as reuse them or extend them instead of computing them again whenever a data set is modified. This would also allow to compose linking specifications such that it would be possible to go from one data set to another without going through an intermediary.

We propose a scheme under which it is possible for data linking tools to take ontology alignments as a way to constrain their solution space. Figure 10 provides a natural way to implement this collaboration.

We first present an expressive language for ontology alignments that can be exploited by data interlinking systems (Section 6). and briefly introduce the linking specification language Silk-LSL (Section 7). Then we show how they could fruitfully be combined for data interlinking (Section 8).

6 EDOAL: an expressive ontology alignment language

EDOAL (Expressive Declarative Ontology Alignment Language) is the new name of the OMWG mapping language for expressing ontology alignment [Euzenat *et al.*, 2007b] that has been available through the Alignment API since version 3.1. This language is an extension of the Alignment format [Euzenat, 2004] that can be generated by most matchers. Its main purpose is to offer more expressiveness in the way alignments are expressed. It presents the advantage to be declarative and also to specify transformations like those needed in order to construct links between resources.

A first advantage of the expressiveness of EDOAL is the possibility to express correspondences between non named entities. For instance, a simple assertion such as "a pianist is a musician who plays piano", can be expressed by (Figure 11):

```
:dbp-mo a align:Alignment;
    align:onto1 <http://dbpedia.org/ontology/>;
    align:onto2 <http://www.musicontology.com/>;
    align:map [ :map1 a align:Cell;
      align:entity1 dbp:Pianist;
      align:entity2 [ a edoal:Class;
        edoal:and mo:MusicArtist;
        edoal:and [ a edoal:PropertyValueConstraint;
          edoal:property mo:instrument;
          edoal:value mo:Piano.
        ].
      ];
      align:relation align:equivalent;
    ].
```

This can help restricting the search space of data interlinking tools far beyond what they currently do (named classes).

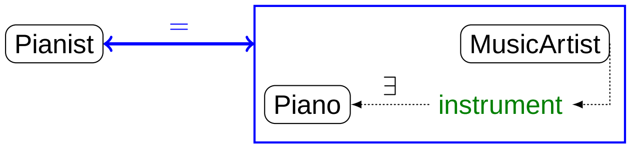

Figure 11: Correspondence between non named resources.

In addition, in EDOAL, it is possible to express that two classes are equivalent, and that their instances are equivalent modulo a transformation. This can be used for covering, without further information, the URI correspondence case of the framework (Section 3.2). For instance, Figure 12 shows an EDOAL correspondence using regular expression transformations for identifying musician instances between two data sets with different conventions.

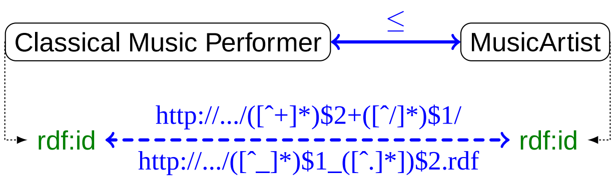

Figure 12: Expression of a resource equivalence represented in an expressive ontology alignment language.

Of course, this can only work when there exists such correspondences, i.e., an exact method for generating links. Most of the time, data interlinking systems still need to use heuristics to find links between entities. This can be provided by the simple Alignment format, but EDOAL can do more by indicating where to look for to establish the correspondence.

In particular, EDOAL allows for expressing contextual relations between elements. For instance, the typical example in Silk documentation is the linking of DBpedia cities and geoname P(laces) through comparing their names and populations. Expressing this with the simple alignment:

```
:dbp-geo a align:Alignment;
    align:onto1 <http://dbpedia.org/ontology/>;
    align:onto2 <http://www.geonames.org/ontology#>;
    align:map [ :map1 a align:Cell;
      align:entity1 dbpedia:City;
      align:entity2 gn:P;
      align:relation align:subsumedBy.
    ];
    align:map [ :map2 a align:Cell;
      align:entity1 dbpedia:populationTotal;
      align:entity2 gn:population;
      align:relation align:equivalent.
    ];
    align:map [ :map3 a align:Cell;
      align:entity1 rdfs:label;
      align:entity2 gn:name;
      align:relation align:equivalent.
    ].
```

does not express the expected meaning because, of course, `rdfs:label` is not equivalent to `gn:name`. One could consider expressing that `gn:name` is more specific than `rdfs:label`. This is correct but still not precise enough. The intended meaning is that, in the context of `dbpdia:City` and `gn:P`, these two properties are equivalent. This is what EDOAL can express through the schema of Figure 13 corresponding to the following alignment:

```
:dbp-geo a align:Alignment;
    align:onto1 <http://dbpedia.org/ontology/>;
    align:onto2 <http://www.geonames.org/ontology#>;
    align:map [ :map1 a align:Cell;
      align:entity1 dbpedia:City;
      align:entity2 gn:P;
      align:relation align:subsumedBy.
    ];
    align:map [ :map2 a align:Cell;
      align:entity1 [ a align:Property;
        edoal:and dbpedia:populationTotal.
        edoal:and [ a edoal:PropertyDomainRestriction;
          edoal:domain dbpedia:City. ];
```

```
      align:entity2 [ a align:Property;
        edoal:and gn:population;
        edoal:and [ a edoal:PropertyDomainRestriction;
          edoal:domain gn:P. ];
      align:relation align:equivalent.
    ];
    align:map [ :map2 a align:Cell;
      align:entity1 [ a align:Property;
        edoal:and rdfs:label.
        edoal:and [ a edoal:PropertyDomainRestriction;
          edoal:domain dbpedia:City. ];
      align:entity2 [ a align:Property;
        edoal:and gn:name;
        edoal:and [ a edoal:PropertyDomainRestriction;
          edoal:domain gn:P. ];
      align:relation align:equivalent.
    ].
```

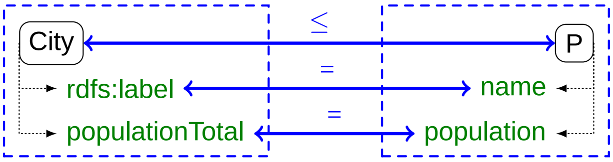

Figure 13: Contextual matching of of two classes and its properties.

Even if such an alignment would provide information to data interlinking tools, this is still not sufficient. Of course, it tells which properties should be equivalent and thus can be used for identifying entities. But it does not tell how to take them into account. So, this alignement would be sufficient to link entities if the values of `rdfs:label` were exactly the same as those of `gn:name` and the values of `populationTotal` were exactly the same as those of `population`, but not otherwise.

EDOAL provides more features for transforming this information, like we have seen in Figure 12. This could be helpful but the problem is deeper: data interlinking is a decision problem rather that just a transformation. It is the role of the data linking specification to tell when a particular `dbpedia:City` and a `gn:P` should be considered the same. This is why we propose to use data interlinking specifications together with alignments.

7 Silk-LSL: a linking specification language

Below is the Silk-LSL [Bizer *et al.*, 2009] specification to interlink cities in the two data sets DBpedia and Geonames:

```
<Silk>

  <Prefix id="rdfs" namespace=
          "http://www.w3.org/2000/01/rdf-schema#" />
  <Prefix id="dbpedia" namespace=
          "http://dbpedia.org/ontology/" />
  <Prefix id="gn" namespace=
          "http://www.geonames.org/ontology#" />

  <DataSource id="dbpedia">
    <EndpointURI>http://demo_sparql_server1/sparql
    </EndpointURI>
    <Graph>http://dbpedia.org</Graph>
  </DataSource>

  <DataSource id="geonames">
    <EndpointURI>http://demo_sparql_server2/sparql
```

```
    </EndpointURI>
    <Graph>http://sws.geonames.org/</Graph>
  </DataSource>

  <Interlink id="cities">
    <LinkType>owl:sameAs</LinkType>

    <SourceDataset dataSource="dbpedia" var="a">
      <RestrictTo>
         ?a rdf:type dbpedia:City
      </RestrictTo>
    </SourceDataset>

    <TargetDataset dataSource="geonames" var="b">
      <RestrictTo>
        ?b rdf:type gn:P
      </RestrictTo>
    </TargetDataset>

    <LinkCondition>
      <AVG>
        <Compare metric="jaroSimilarity">
          <Param name="str1" path="?a/rdfs:label" />
          <Param name="str2" path="?b/gn:name" />
        </Compare>
        <Compare metric="numSimilarity">
          <Param name="num1"
                 path="?a/dbpedia:populationTotal" />
          <Param name="num2" path="?b/gn:population" />
        </Compare>
      </AVG>
    </LinkCondition>

    <Thresholds accept="0.9" verify="0.7" />
    <Output acceptedLinks="accepted_links.n3"
      verifyLinks="verify_links.n3"
      mode="truncate" />
  </Interlink>

</Silk>
```

This specification fulfills two roles:

– It is an alignment: it specifies the classes in which entities to link can be found. Restrictions to `dbpedia:City` and `gn:P` are in fact an alignment between these two concepts. Similarly, the compared properties `populationTotal` and `population` and `rdfs:label` and `name`, respectively provide the correspondences between properties.

– It specifies how to link entities. Indeed, what Silk brings in addition is the specification of how to decide if two entities should be linked: when the average (`AVG`) of their respective distances (`Compare`) is over a threshold (`Threshold`, there are two thresholds, one for accepting automatically the equivalence and one for drawing the attention of a user).

It could be possible to refer to an external alignment between the two underlying ontologies instead of specifying it in the linking specification. This approach would present obvious reuse advantages when other data sets requiring the same alignment, i.e., using the same ontologies, need to be interlinked.

8 Data interlinking using ontology alignments

Apart from Knofuss, interlinking tools do not provide the possibility to use an ontology alignment. Knofuss still needs to specify queries on both data sets from which results equivalent resources will be identified.

Indeed, using an explicit alignment, provided that it is expressive enough, can serve two functions:

1. narrowing the search space through pointing to equivalent concepts, and
2. providing the properties that can be used for identifying concepts.

There are two ways to articulate ontology alignment and linking specifications:

– Transforming an expressive alignment into a linking specification: this requires that the alignment contains as much information as possible and that matchers be able to produce such descriptions. This has the advantage that from the alignment, the specification may be transformed into different linking specification languages.
– Enabling linking specifications to refer explicitly to alignments and eventually to matchers: this requires extending specification languages for that purpose.

We consider the latter option below. Given that the alignment is available, it is possible to simplify the Silk specification and refer to the alignment, by introducing three types of information: which alignments to use (`UseAlignment`), entities of which correspondences must be linked (`LinkCell`) and which matched properties can be compared for identifying entities (`CellParam`).

```
<UseAlignment rdf:resource="#dbp-geo" />

<Interlink id="cities">
  <LinkType>owl:sameAs</LinkType>

  <LinkCell rdf:resource="#map1" />

  <LinkCondition>
    <And combiner="AVG">
      <Compare metric="jaroSimilarity">
        <CellParam rdf:resource="#map2" />
      </Compare>
      <Compare metric="numSimilarity">
        <CellParam rdf:resource="#map3" />
      </Compare>
    </And>
  </LinkCondition>

  <Thresholds accept="0.9" verify="0.7" />
  <Output acceptedLinks="accepted_links.n3"
    verifyLinks="verify_links.n3"
    mode="truncate" />
</Interlink>
```

The specifics of the data interlinking task remain in this specification: how to compare values, how to aggregate their results and when to issue the link or not.

In fact, the symbiosis between the alignment and the linking specification can be rendered even more automatic, e.g., by defining default rules for comparing values of a given type, default rules for aggregating metrics, and default threshold rules. However, it is also useful that the linking specification designer can keep control on what the interlinking tool does and, even if a correspondence is not in an alignment, be able to define it.

This approach presents several advantages:

1. The link specification is simplified, reducing the manual input;
2. There is a clear separation between links, linking specification, and ontology alignments;
3. The same alignment can be reused for linking any two data sets described according to the same ontologies.

9 Conclusion

Interlinking data sets becomes an even more important problem as their number quickly increase. In order to scale, the interlinking task has to be as automated as possible.

We have studied various existing data interlinking tools and observed the following:

– Beyond the variations between these systems, it is possible to define a general framework covering the different levels of expressiveness (ranging from a Prolog program to compositions of linking specifications).

– Although there is a relevant similarity with ontology alignment, an ontology alignment language is not enough to express linking specifications, particularly because it is not its primarily goal to identify individual entities.

We have thus proposed an architecture based on three different languages having each its own precise purpose: expressing links, expressing linking specifications, and expressing ontology alignments.

This architecture can be used in order to organize a better collaboration between ontology matchers and data interlinking tools. This can be achieved with only minimal extensions to existing languages.

In particular, we have illustrated the ontology alignment part with EDOAL, an expressive ontology alignment language that offers the necessary concepts for being used in data interlinking. On the data interlinking side, we have focussed on the Silk-LSL language which seems to be at once declarative and powerful enough to express a wide range of constraints on data interlinking. Extending it with the capacity to benefit from ontology alignments would allow tools using it to benefit from the wide range of ontology alignment techniques and tools.

The domain of interlinking data on the web is quickly expanding. New needs and new techniques appear. It is thus important not to breed innovations with a narrow language. Developing standard tools to share link specifications will greatly improve those techniques. There is still a lot of work to do in order to achieve this goal.

Acknowledgements

We thank the data interlinking tool developers who provided us with linking specifications for their tools.

This work was conducted in the context of the Datalift project funded by the French ANR (Agence Nationale de la Recherche) under grant ANR-10-CORD-009.

References

[Adida *et al.*, 2008] Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. RDFa in XHTML: Syntax and processing. Recommendation, W3C, 2008. http://www.w3.org/TR/rdfa-syntax/.

[Alexander *et al.*, 2009] 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.

[Berners-Lee, 2009] Tim Berners-Lee. Linked-data design issues. W3C design issue document, June 2009. http://www.w3.org/DesignIssues/LinkedData.html.

[Bizer *et al.*, 2009] 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.

[Bizer, 2003] Christian Bizer. D2r map - a database to rdf mapping language. In *Proc. 12th WWW conference poster session*, 2003.

[Bouquet *et al.*, 2008] 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.

[David *et al.*, 2011] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The Alignment API 4.0. *Semantic web journal*, 2(1):3–10, 2011.

[Elmagarmid *et al.*, 2007] Ahmed Elmagarmid, Panagiotis Ipeirotis, and Vassilios Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, January 2007.

[Euzenat and Shvaiko, 2007] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[Euzenat *et al.*, 2007a] Jérôme Euzenat, Adrian Mocan, and François Scharffe. *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, chapter Ontology Alignments:an ontology management perspective. Springer, 2007.

[Euzenat *et al.*, 2007b] 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.

[Euzenat, 2004] 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.

[Fellegi and Sunter, 1969] Ivan Fellegi and Alan Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[Hassanzadeh *et al.*, 2009] 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.

[Heath and Bizer, 2011] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*, volume 1 of *Synthesis Lectures on the Semantic Web: Theory and Technology*. Morgan & Claypool, 1 edition, 2011.

[Hogan *et al.*, 2007] 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.

[Jaffri *et al.*, 2008] 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.

[Klein, 2001] Michael Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing*, 2001.

[Nikolov *et al.*, 2008] 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.

[Noy and Musen, 2000] Natalya Fridman Noy and Mark A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press / The MIT Press, 2000.

[Rahm and Bernstein, 2001] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.

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

[Raimond *et al.*, 2008] 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.

[Sahoo *et al.*, 2009] Satya Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau, Sören Auer, Juan Sequeda, and Ahmet Ezzat. A survey of current approaches for mapping relational databases to rdf. Report, W3C RDB2RDF incubator group, 2009.

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

[Sauermann and Cyganiak, 2008] Leo Sauermann and Richard Cyganiak. Cool URIs for the semantic web. W3C note, W3C, March 2008. http://www.w3.org/TR/2008/NOTE-cooluris-20080331/.

[Scharffe and Euzenat, 2010] François Scharffe and Jérôme Euzenat. Méthodes et outils pour lier le web des données. In *Actes 17e conférence AFIA-AFRIF sur reconnaissance des formes et intelligence artificielle (RFIA), Caen (FR)*, pages 678–685, 2010.

[Scharffe *et al.*, 2009] 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.

[Visser *et al.*, 1997] Pepjijn R. S. Visser, Dean M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *AAAI 1997 Spring Symposium on Ontological Engineering*, Stanford, USA, 1997.

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

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

[Winkler, 2006] William Winkler. Overview of record linkage and current research directions. Technical Report 2006-2, Statistical Research Division. U.S. Census Bureau, 2006.

Contents

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399