HAL
open science

Data Extraction, Transformation and Integration Guided by an Ontology

Chantal Reynaud, Nathalie Pernelle, Marie-Christine Rousset, Brigitte Safar, Fatiha Saïs

To cite this version:

Chantal Reynaud, Nathalie Pernelle, Marie-Christine Rousset, Brigitte Safar, Fatiha Saïs. Data Extraction, Transformation and Integration Guided by an Ontology. Ladjel Bellatreche. Data Warehousing Design and Advanced Engineering Applications: Methods for Complex Construction, Advances in Data Warehousing and Mining Book Series, IGI Global, 2009. inria-00432585

HAL Id: inria-00432585

https://inria.hal.science/inria-00432585

Submitted on 20 Nov 2009

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.

Chapter

Data Extraction, Transformation and Integration guided by an Ontology

Chantal Reynaud
Université Paris-Sud, CNRS (LRI) & INRIA (Saclay – Île-de-France), Orsay, France
Nathalie Pernelle
Université Paris-Sud, CNRS (LRI) & INRIA (Saclay – Île-de-France), Orsay, France
Marie-Christine Rousset
LIG – Laboratoire d'Informatique de Grenoble, St Martin d'Hères, France
Brigitte Safar
Université Paris-Sud, CNRS (LRI) & INRIA (Saclay – Île-de-France), Orsay, France
Fatiha Saïs
Université Paris-Sud, CNRS (LRI) & INRIA (Saclay – Île-de-France), Orsay, France

ABSTRACT
This paper deals with integration of XML heterogeneous information sources into a data warehouse with data defined in terms of a global abstract schema or ontology. We present an approach supporting the acquisition of data from a set of external sources available for an application of interest including data extraction, data transformation and data integration or reconciliation. The integration middleware that we propose extracts data from external XML sources which are relevant according to a RDFS+ ontology, transforms returned XML data into RDF facts conformed to the ontology and reconcile RDF data in order to resolve possible redundancies.

KEYWORDS
Data Integration, Semantic Integration, Data Warehouse, Ontologies, Automatic reasoning, Wrappers, Data extraction, Reference Reconciliation, Equation system, Iterative resolution

INTRODUCTION
A key factor for the success of the Semantic Web is to provide a unified, comprehensive and high-level access to voluminous and heterogeneous data. Such an access can be provided by an ontology in integrators supporting high-level queries and information interoperation. Our work takes place in the context of a data warehouse with data defined in terms of a global abstract schema or ontology. We advocate an information integration approach

supporting the acquisition of data from a set of external sources available for an application of interest. This problem is a central issue in several contexts, data warehousing, interoperate systems, multi-database systems, web information systems. Several steps are required for the acquisition of data from a variety of sources to a data warehouse based on an ontology (1) Data extraction: only data corresponding to descriptions in the ontology are relevant. (2) Data transformation: they must be defined in terms of the ontology and in the same format. (3) Data integration and reconciliation: the goal of this task is to resolve possible redundancies.

As a vast majority of sources rely on XML, an important goal is to facilitate the integration of heterogeneous XML data sources. Furthermore, most applications based on the Semantic Web technologies rely on RDF (McBride, 2004), OWL-DL (Mc Guinness & Van Harmelen, 2004) and SWRL (Horrocks et al., 2004). Solutions for data extraction, transformation and integration using these recent proposals must be favoured. Our work takes place in this setting. We propose an integration middleware which extracts data from external XML sources that are relevant according to a RDFS+ ontology (RDFS+ is based on RDFS (McBride, 2004)), transforms them into RDF facts conformed to the ontology, and reconciles redundant RDF data.

Our approach has been designed in the setting of the PICSEL3 project[i] whose aim was to build an information server integrating external sources with a mediator-based architecture and data originated from external sources in a data warehouse. Answers to users' queries should be delivered from the data warehouse. So data have to be passed from (XML) external sources to the (RDF) data warehouse and answers to queries collected from external sources have to be stored in the data warehouse. The proposed approach has to be totally integrated to the PICSEL mediator-based approach. It has to be simple and fast in order to deal with new sources and new content of integrated sources. Finally, it has to be generic, applicable to any XML information source relative to any application domain. In Figure 1 we present the software components designed in the setting of the project to integrate sources and data. This paper focuses on the description of the content of a source, the extraction and the integration of data (grey rectangles in Figure 1). The automatic generation of mappings is out of the scope of the paper.

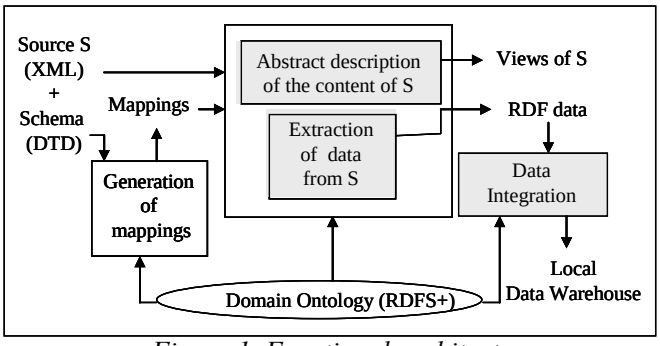

Figure 1. Functional architecture

The extraction and transformation steps rely on correspondences or mappings between local schemas of external sources and the ontology. In a previous work, we proposed techniques to automate the generation of these mappings (Reynaud & Safar, 2009). In this chapter, we present an approach which automates the construction of wrappers given a set of mappings. It starts from the description of the abstract content of an external source and performs data acquisition, i.e. data extraction and transformation in order to conform to a same global schema. The description of the abstract content of an external source is also usable to manage sources with data that remain locally stored, making that way our techniques quite integrated to the PICSEL mediator-based approach. The transformation phase is then followed by a reconciliation step whose aim is to handle several problems: possible mismatches between data referring to the same real world object (different conventions and vocabularies can be used to represent and describe data), possible errors in the data stored in the sources especially frequent when data are automatically extracted from the Web, possible inconsistencies between values representing the properties of the real world objects in different sources. This reconciliation step is essential because the

conformity to a same global schema does not indeed prevent variations between data descriptions. For this last step, we propose a knowledge-based and unsupervised approach, based on two methods, a logical one called L2R and a numerical one called N2R. The Logical method for Reference Reconciliation (L2R) is based on the translation in first order logic Horn rules of some of the schema semantics. In order to complement the partial results of L2R, we have designed a Numerical method for Reference Reconciliation (N2R). It exploits the L2R results and allows computing similarity scores for each pair of references.

The paper is organized as follows. In section 2, we present close related work and point out the novel features of the approach presented in this chapter. In section 3, we describe our approach. First, we define the data model used to represent the ontology and the data, the XML sources and the mappings automatically generated used as inputs in the data extraction and transformation process. We present the data extraction and transformation tasks and then the two reconciliation techniques (L2R and N2R) followed by a summary of the results that we have obtained. In section 4 we briefly describe future research directions. Finally, section 5 concludes the chapter.

BACKGROUND

Many modern applications such as data warehousing, global information systems and electronic commerce need to take existing data with a particular schema, and reuse it in a different form. For a long time data conversion has usually been done in an ad hoc manner by developing non reusable software. Later language-based and declarative approaches have provided tools for the specification and implementation of data and schema translations among heterogeneous data sources (Abiteboul et al.,1997; Cluet et al., 1998). Such rule-based approaches can deal with complex transformations due to the diversity in the data model and to schema matching. In the former case, the approach helps to customize general purpose translation tools. In the latter case, the idea is that the system automatically finds the matching between two schemas, based on a set of rules that specify how to perform the matching. All these works provide tools to design data conversion programs but they do not provide the ability to query external sources. More recently, the Clio system (Popa et al., 2002) has been proposed as a complement and an extension of the language-based approaches. Given value correspondences that describe how to populate a single attribute of a target schema, this system discovers the mapping query needed to transform source data to target data. It produces SQL queries and provides users with data samples to allow them to understand the mappings produced.

Our work can also be compared to data integration systems providing mechanisms for uniformly querying sources through a target schema but avoiding materializing it in advance. These works adopt either the Global-As-View (GAV) approach and describes the target schema in terms of the local schemas, either the Local-As-View (LAV) approach and describes every source schema in terms of the target one. Based on these two approaches, there is a hybrid approach, called Global-Local-As-View (GLAV) and performed in SWIM (Koffina et al., 2006), that allows to specify mappings between elements of the target schema and elements of the source ones, considered one by one. We adopted it also in our work. It simplifies the definition of the mappings and allows a higher automation of extraction and transformation tasks.

Compared with the approaches cited above, the present work shows several interesting features coming both from data conversion and data integration (mediator) work. Given a set of mappings, our approach is entirely automatic. Our solution has to be integrated in the PICSEL mediator-based approach. In PICSEL, queries are rewritten in terms of views which describe the content of the sources. Hence, a solution to data extraction and transformation that generates these views in an automatic way in the same time is a very interesting point. The specification of how to perform the matching between the sources and the data warehouse can then be automatically generated by producing XML queries from the mappings, the views and the ontology. It corresponds to the extraction and transformation steps performed on the source taken as a whole and not attribute per attribute as in the work aiming at converting a relational database in another one. The approach is directed by the ontology. Only data that can be defined in terms of the ontology are extracted. Furthermore XML queries are

capable to transform data in order to make them defined in terms of the ontology as well as in the same format. This is a way to handle the transformation task.

The problem of reference reconciliation was introduced by the geneticist Newcombe (1959) and was first formalized by (Fellegi & Sunter, 1969). Since then, several work and various approaches have been proposed. We distinguish these approaches according to the exploitation of the reference description, to how knowledge is acquired and which kind of result is obtained by the methods.

For the reference description we have three cases. The first one is the exploitation of the unstructured description of the text appearing in the attributes (Cohen, 2000; Bilke & Naumann, 2005). In these approaches, the similarity is computed by using only the textual values in the form of a single long string without distinguishing which value corresponds to which attribute. This kind of approaches is useful in order to have a fast similarity computation (Cohen, 2000), to obtain a set of reference pairs that are candidates for the reconciliation (Bilke & Naumann, 2005) or when the attribute-value associations may be incorrect. The second type of approaches consists in considering the reference description as structured in several attributes. A large number of methods have adopted this vision by proposing either probabilistic models (Fellegi & Sunter, 1969), which allow taking decisions of reconciliation after the estimation of the probabilistic model parameters, or by computing a similarity score for the reference pairs (Dey et al., 1998a) by using similarity measures (Cohen et al., 2003). The third one consists in considering, in addition to the reference description structured in a set of attributes, the relations that link the references together (Dong et al., 2005). These global approaches take into account a larger set of information. This allows to improve the results in terms of the number of false positive (Bhattacharya & Getoor, 2006) or in terms of the number of the false negative. Like those approaches, both the logical L2R and the numerical N2R methods are global, since they exploit the structured description composed of attributes and relations. The relations are used both in the propagation of reconciliation decisions by the logical rules (L2R) and in the propagation of similarity scores through the iterative computation of the similarity (N2R).

In order to improve their efficiency, some recent methods exploit knowledge which is either learnt by using supervised algorithms or explicitly specified by a domain expert. For instance, in (Dey et al., 1998b; Dong et al., 2005), knowledge about the impacts of the different attributes or relations are encoded in weights by an expert or learnt on labelled data. However, these methods are time consuming and dependent on the human experience for labelling the training data or to specify declaratively additional knowledge for the reference reconciliation. Both the L2R and N2R methods exploit the semantics on the schema or on the data, expressed by a set of constraints. They are unsupervised methods since no labelled data is needed by either L2R or N2R.

Most of the existing methods infer only reconciliation decisions. However, some methods infer non-reconciliation decisions for reducing the reconciliation space. This is the case for the so-called blocking methods introduced in (Newcombe, 1962) and used in recent approaches such as (Baxter et al., 2003).

THE PICSEL3 DATA EXTRACTION, TRANSFORMATION AND INTEGRATION APPROACH

In this section, we first define the data model used to represent the ontology and the data, the external XML sources and the mappings. In a second sub-section, we present the data extraction and transformation tasks and then the two reconciliation techniques (L2R and N2R) followed by a summary of the results that we have obtained by performing these methods on data sets related to the scientific publications.

Data Model, XML sources and mappings

We first describe the data model used to represent the ontology *O*. This model is called RDFS+ because it extends RDFS with some OWL-DL primitives and SWRL rules, both being used to state constraints that enrich the semantics of the classes and properties declared in RDFS. Then we describe the XML sources we are interested

in and the mappings that are automatically generated and then used as inputs of the data extraction and transformation process.

The RDFS+ data model

RDFS+ can be viewed as a fragment of the relational model (restricted to unary and binary relations) enriched with typing constraints, inclusion and exclusion between relations and functional dependencies.

The schema and its constraints

A RDFS schema consists of a set of classes (unary relations) organized in a taxonomy and a set of typed properties (binary relations). These properties can also be organized in a taxonomy of properties. Two kinds of properties can be distinguished in RDFS: the so-called relations, the domain and the range of which are classes and the so-called attributes, the domain of which is a class and the range of which is a set of basic values (e.g. Integer, Date, Literal). For example, in the RDFS schema presented in Figure 2, we have a relation *located* having as domain the class *CulturalPlace* and as range the class *Address*. We also have an attribute *name* having as domain the class *CulturalPlace* and as range the data type Literal.

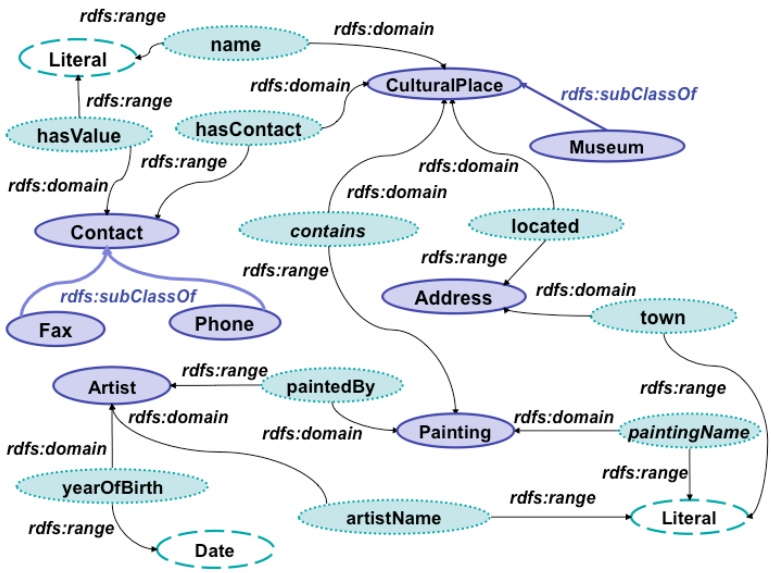

Figure 2. Example of a RDFS schema

We allow the declaration of constraints expressed in OWL-DL or in SWRL in order to enrich the RDFS schema. The constraints that we consider are of the following types:

- Constraints of disjunction between classes: DISJOINT(*C,D*) is used to declare that the two classes *C* and *D* are disjoint, for example : DISJOINT (*CulturalPlace*, *Artist*).
- Constraints of functionality of properties: PF(*P*) is used to declare that the property *P* (relation or attribute) is a functional property. For example, PF(*located*) and PF(*name*) express respectively that a cultural place is located in one and only one address and that a cultural place has only one name. These constraints can be generalized to a set $\{P_1,\ldots,P_n\}$ of relations or attributes to state a combined constraint of functionality that we will denote PF($P_1,\ldots,P_n$).
- Constraints of inverse functionality of properties: PFI(*P*) is used to declare that the property *P* (relation or attribute) is an inverse functional property. For example, PFI(*contains*) expresses that a painting cannot belong to several cultural places. These constraints can be generalized to a set $\{P_1,\ldots,P_n\}$ of relations or attributes to state a combined constraint of inverse functionality that we will denote PFI($P_1,\ldots,P_n$). For example, PFI(*located*, *name*) expresses that one address and one name cannot be associated to several cultural places (i.e. both are needed to identify a cultural place).

Data description and their constraints

A datum has a reference, which has the form of a URI (e.g. http://www.louvre.fr, NS-S1/painting243), and a description, which is a set of RDF facts involving its reference. A RDF fact can be:

- either a class-fact *C(i)*, where *C* is a class and *i* is a reference,
- or a relation-fact $R(i_1, i_2)$, where *R* is a relation and i_1 and i_2 are references,
- or an attribute-fact $A(i,v)$, where *A* is an attribute, *i* a reference and *v* a basic value (e.g. integer, string, date).

The data description that we consider is composed of the RDF facts coming from the data sources enriched by applying the RDFS entailment rules (Hayes, 2004). We consider that the descriptions of data coming from different sources conform to the same RDFS+ schema (possibly after schema reconciliation). In order to distinguish the data coming from different sources, we use the source identifier as the prefix of the reference of the data coming from that source. Example 1 provides examples of data coming from two RDF data sources S_1 and S_2, which conform to a same RDFS+ schema describing the cultural application previously mentioned.

Example 1: An example of RDF data
Source S1 : Museum(r607); name(r607, ″ Le Louvre ″); located(r607, d1e5); Address(d1e5); town(d1e5, ″Paris″); contains(r607, p112); paintingName(p112, ″La Joconde″*);*
***Source S2*:** Museum(r208); name(r208, ″musée du Louvre″); located(r208, l6f2); Address(l6f2); town(l6f2, ″ville de Paris″); contains(r208, p222) ; paintingName(p222, *″Iris ″);* contains(r208, p232); paintingName(p232, ″Joconde″*);*

We consider two kinds of axioms accounting for the Unique Name Assumption (UNA) and the Local Unique Name Assumption (denoted LUNA). The UNA states that two data of the same data source having distinct references refer to two different real world entities (and thus cannot be reconciled). Such an assumption is valid when a data source is clean. The LUNA is weaker than the UNA, and states that all the references related to a same reference by a relation refer to real world entities that are pairwise distinct.

The XML sources

The XML sources that we are interested in are valid documents, instances of a DTD that defines their structure. We consider DTDs without entities or notations. A DTD can be represented as an acyclic oriented graph with one node for each element definition. The links between two nodes are composition links. The attributes associated to the elements in a DTD are associated to element nodes in the graph representing to the DTD. Because the DTDs are acyclic, their associated graph may be represented as a forest of trees, whose roots correspond to entry points in the graph (nodes without predecessors). Nodes shared in the graph by several trees are duplicated in order to make these trees independent of each other. Figure 3 is an example of a DTD of a source to be integrated. It is represented by the tree T_1. A fragment of the XML document conformed to the DTD tree T_1 is presented in Figure 4.

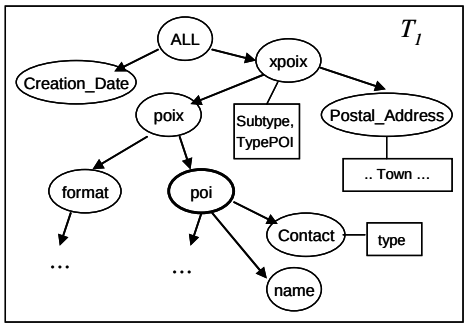

Figure 3. Example of a DTD tree

```
<xpoix id = 'PCUIDF07721' typePOI ='museum' >
  <poix version = '1' >
    <format >
          …
    </format >
    <poi>
      <name>  Le Louvre </name>
      <contact type = 'tel'> 01 60 20 11 06</contact>
      <contact type = 'fax'>01 60 20 44 02</contact>
    </poi>
  </poix>
  <Postal_Address>
    <town> Paris </town>
     …..
  </Postal_Address>
</xpoix>
```

Figure 4. Example of a XML document conformed to the DTD tree of the Figure 3

The mappings

Mappings are computed in a semi-automatic way. They are links between the ontology *O* and a DTD tree *D* (elements or attributes). The format of the mappings for the classes and the properties of *O* is described just below.

When c_1 is a concept of *O*, the format of the mappings may be:

- $c_1 \leftrightarrow$ //e
- $c_1 \leftrightarrow$ //e/@att
- $c_1 \leftrightarrow$ //e/[@att = 'val']/@att

When *R* is a relation between c_1 and c_2 of *O* such that $\exists$ $c_1 \leftrightarrow$ //a and $c_2 \leftrightarrow$ //b, the format of the mapping is: $r_1(c_1, c_2) \leftrightarrow r_1$(//a, //a/ …/b)

When *A* is an attribute of c_1 represented in the ontology *O* such that $\exists$ $c_1 \leftrightarrow$ //a and b being mapped to *A* in *T*, the format of the mapping is: A of $c_1 \leftrightarrow$ A(//a, //a/ …/b)

In this format, $\leftrightarrow$ indicates a mapping link between entities in *O* and entities in *T* defined by their path using XPath (Berglund et al., 2007) in the associated graph. *e* refers to an element in *T*, *@att* refers to the attribute *att*.

Note that we may have conditional mappings when the link with an attribute *att* depends on its value *val* ($C_1 \leftrightarrow$ //e/[@att = 'val']/@att).

Data Extraction and Transformation

Data extraction and transformation are completely automatic tasks usually performed by wrappers. It is a two-step process. First, an abstract description of the content of the external source is built. Second, data is extracted and presented in the format of the data warehouse.

Abstract description of a source

The content of an external source is described in terms of views in the language accepted by PICSEL (Rousset & Reynaud, 2003) by a set of rules. Each rule links a view $v_i(x)$ with a local name to domain relations $p(x)$ in the ontology. It indicates which kind of data can be found in the source. Our proposal is to build a limited number of views, one view per central concept in a source. A concept is said central if it is mapped to an element in *O* and if none of its predecessors is mapped.

The construction process of a view is incremental. At first, it is guided by the DTD tree *T* of the XML source in order to identify central concepts. A depth-first search is performed on the DTD tree *T* until an element e_D of *T* belonging to a mapping is found. This element will necessarily be associated to a class e_O in *O* representing a central concept. The search of additional central concepts will be pursued later starting from the brother node of e_D. Indeed, all the elements belonging to the sub-tree rooted in e_D and mapped with entities in *O* should be linked to e_O in *O*. Second the construction process of a view is guided by the ontology in order to complete the description of the central concepts. We introduce the properties of the classes corresponding in *O* to the central concepts (relations and attributes) if they are properties with mappings, the classes linked by the introduced relations (called subordinated concepts), their properties with mappings, and so on. Indeed, the same completion process is performed recursively on each subordinated concept. For example, *name*, *located* and *hasContact* are three properties of the class *CulturalPlace* with mappings. *located* and *hasContact* are two relations establishing respectively a link with the classes *Address* and *Contact*. The view under construction corresponding to S_1 will be:

$S_1(x,y,z,t) \rightarrow CulturalPlace(x) \wedge name(x,y) \wedge located(x,z) \wedge Address(z) \wedge hasContact(x,t) \wedge Contact(t)\ldots$

Furthermore, we take into account classes that have specializations in *O*. When specializations correspond to central concepts, we build one view per specialization. For example, *Museum* is a specialization of *CulturalPlace* which is a central concept. We build a new view for *Museum*:

$S_{12}\ (x,y,z,t) \rightarrow Museum(x) \wedge name(x,y) \wedge located(x,z) \wedge Address(z) \wedge hasContact(x,t) \wedge Contact(t)\ldots$

When subordinated concepts have specializations in *O*, our treatment depends on the cardinality of the relation establishing a link with the subordinated concept. If the cardinality is multiple (non functional property) as the cardinality of the relation *hasContact* in the example just before, we will introduce all the classes that are specializations in the same view. That way, the source S_1 providing instances of *Museum* as it is shown in Figure 4 will be described by a unique view grouping the class *Museum*, its properties and the classes *Address*, *Contact*, *Tel*, *Fax* linked by the relations *located* and *hasContact*:

$S_{12}\ (x,y,z,t, t_1, t_2) \rightarrow Museum(x) \wedge name(x,y) \wedge located(x,z) \wedge Address(z) \wedge hasContact(x,t) \wedge Contact(t) \wedge hasContact(x,t_1) \wedge Tel(t_1) \wedge hasContact(x, t_2) \wedge Fax(t_2)$.

On the opposite, if the relation is a functional property, we build one view per specialization, as it is done for central concepts with specializations.

Data extraction and transformation

For each view, we then generate a wrapper which will query the XML source in regard to its language and its vocabulary and transform returned instances into RDF facts conformed to the RDFS+ ontology. Wrappers are associated to queries expressed in XQuery (Boag et al., 2007). The *FLWO* part of a XQuery statement performs the extraction task while the *R* part performs the transformation task from XML to RDF using the terms of the ontology. The construction of wrappers follows the construction process of views. We build one query per view. Queries are built in an incremental way, performing at first the concept, followed by its properties. For each central concept named *conceptC* in *O*, we look for the instances of its corresponding element (or attribute) *mapC* in *D* (*FOR* part). For each instance we generate a unique identifier (*generate-Id*). The name of the concept in *O* is used as a tag in the *Return* part. Thus the initial form of the query is the following:

```
for $x in doc("source.xml")//mapC
let $idcpt := gi:generate-Id($x1)
return
<p3:conceptC  rdf:nodeID="{$idcpt}">
```

$x is associated to *mapC* and contains all the elements belonging to the tree rooted in *mapC* in the XML source. The objective of the query that we want to generate is to extract from *$x* all the elements which are properties in *O*. For this, we need mappings of these elements. The extraction of attributes in XQuery is made by indicating the path defined in the mapping and by using the primitive *Text()* to obtain the element without tags. The extraction of the relations needs a new identifier for the subordinated concept. A new XML fragment will be added to describe the subordinated concept and its properties. If the considered mappings are conditional, we introduce a Where part in the query in order to specify the condition. An example of a query leading to extract data from S_1 according to the view S_{12} (x,y,z,t) described above is given in Figure 5a and the extracted data in Figure 5b.

S_{12} (x,y,z,t) $\rightarrow$ *Museum*(x) $\wedge$ *name*(x,y) $\wedge$ *located*(x,z)$\wedge$ *Address*(z) $\wedge$ *Town*(z,t).

```
declare namespace gi = "java:pkg.GenerateId";
declare namespace rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#";
declare namespace p3 ="http://www.lri.fr/picsel3/tourismrdfs#";
for $x in doc("source.xml")//xpoix
  let $idcpt := gi:generate_Id()
  let $idrel := gi:generate_Id()
Where @x/@typePOI= "museum "
return
<p3: Museum rdf:nodeID="{$idcpt}" >
   <p3:name> {$x/poix/poi/name/text()} </p3:name>
   <p3:located rdf:nodeID="{$idrel}"/>
</p3:Museum>
<p3:Address rdf:nodeID="{$idrel}">
  <p3:town> {$x/Postal_Address/town/text()}</p3:town>
</p3: Address>
```

```
<p3: Museum rdf:nodeID="r6o7">
   <p3:name> Le Louvre </p3:name>
   <p3:Located rdf:nodeID="d1e5" />
</p3: Museum >

<p3: Address rdf:nodeID="d1e5" >
   <p3:town> Paris </p3:town>
</p3: Address >
```

Figure 5a-5b: A query (on the left side) and the extracted data (on the right side) from S_1

Data Integration

Let S_1 and S_2 be two data sources which conform to the same RDFS+ schema. Let I_1 and I_2 be the two reference sets that correspond respectively to the data of S_1 and S_2. The problem consists in deciding whether references are reconciled or not reconciled. Let *Reconcile* be a binary predicate. *Reconcile(X, Y)* means that the two references denoted by *X* and *Y* refer to the same world entity. The reference reconciliation problem considered in L2R consists in extracting from the set $I_1 \times I_2$ of reference pairs two subsets REC and NREC such that:

$$\begin{cases} \text{REC} = \{(i,\ i') \ / \ Reconcile\ (i,\ i')\} \\ \text{NREC} = \{(i,\ i'),\ \neg\ Reconcile(i,\ i')\} \end{cases}$$

The reference reconciliation problem considered in N2R consists in, given a similarity function $Sim_r : I_1 \times I_2 \rightarrow$ [0..1], and a threshold T_{rec} (a real value in [0..1] given by an expert, fixed experimentally or learned on a labeled data sample), computing the following set:

$$REC_{N2R} = \{(i,i') \in (I_1 \times I_2) \setminus (REC \cup NREC), tq.Sim_r(i,i') > T_{rec}\}$$

L2R: a Logical method for Reference Reconciliation

L2R (Saïs et al., 2007) is based on the inference of facts of reconciliation (*Reconcile(i,j)*) and of non-reconciliation ($\neg$*Reconcile(i',j')*) from a set of facts and a set of rules which transpose the semantics of the data sources and of the schema into logical dependencies between reference reconciliations. Facts of synonymy ($SynVals(v_1,v_2)$) and of no synonymy ($\neg\ SynVals(u_1,\ u_2)$) between basic values (strings, dates) are also inferred. For instance, the synonymy *SynVals("JoDS", "Journal of Data Semantics")* may be inferred. The L2R distinguishing features are that it is global and logic-based: every constraint declared on the data and on the

schema in RDFS+ is automatically translated into first-order logic Horn rules (rules for short) that express dependencies between reconciliations. The advantage of such a logical approach is that if the data are error-free and if the declared constraints are valid, then the reconciliations and non-reconciliations that are inferred are correct, thus guaranteeing a 100 % precision of the results.
We first describe the generation of the reconciliation rules. Then we present the generation of the facts and finally the reasoning, which is performed on the set of rules and facts.

Generation of the set of reconciliation rules

They are automatically generated from the constraints that are declared on the data sources and on their common schema.

- Translation of the constraints on the data sources

The UNA assumption, if it is stated on the sources S_1 and S_2, is translated automatically by four rules. For example, the following rule R1 expresses the fact that two distinct references coming from the same source cannot be reconciled.

$$R1: Src1(x) \wedge Src1(y) \wedge (x \neq y) \Rightarrow \neg\, Reconcile(x,y)$$

where $Src_i(x)$ means that the reference x is coming from a source S_i.
Analogous rules express that one reference coming from a source S_i can be reconciled with at most one reference coming from a source S_j. Similarly, two rules are generated for translating LUNA semantics.

- Translation of the schema constraints.

For each relation R declared as functional by the constraint PF(R), the following rule R6.1(R) is generated:
$R6.1(R)$: $Reconcile(x, y) \wedge R(x ,z) \wedge R(y, w) \Rightarrow Reconcile(z, w)$
For example, the following rule is generated concerning the relation *located* which relates references of cultural places to references of addresses and which is declared functional:
$R6.1(located)$: $Reconcile(x, y) \wedge located(x, z) \wedge located(y, w) \Rightarrow Reconcile(z, w)$
For each attribute A declared as functional by the axiom PF(A), a similar rule which concludes on *SynVals* is generated.

Likewise, analogous rules are generated for each relation R and each attribute A declared as inverse functional. Rules are also generated for translating combined constraints PF($P_1,\ldots ,P_n$) and PFI($P_1,\ldots ,P_n$) of (inverse) functionality. For example, the declaration PFI(*paintedBy, paintingName*) states a composed functional dependency which expresses that the artist who painted it jointly with its name functionally determines a painting.
For each pair of classes C and D involved in a DISJOINT(C,D) statement declared in the schema, or such that their disjunction is inferred by inheritance, a rule is generated to express the fact that their references cannot be reconciled. A transitivity rule allows inferring new reconciliation decisions by applying transitivity on the set of already inferred reconciliations.
See (Saïs et al., 2009) for a complete description of the generation process of reconciliation rules.

Reasoning method for reference reconciliation

In order to infer sure reconciliation and non-reconciliation decisions, we apply an automatic reasoning method based on the resolution principle (Robinson, 1965; Chang & Lee, 1997). This method applies to the clausal form of the set of rules R described above and a set of facts F describing the data, which is generated as follows.

- Generation of the set of facts.

The set of RDF facts corresponding to the description of the data in the two sources *S1* and *S2* is augmented with the generation of:
- new class-facts, relation-facts and attribute-facts derived from the domain and range constraints that are declared in RDFS for properties, and from the subsumption statements ;
- facts of the form $Src_1(i)$ and $Src_2(j)$;
- synonymy facts of the form $SynVals(v_1,v_2)$ for each pair (v_1,v_2) of basic values that are identical (up to some punctuation or case variations) ;

- non synonymy facts of the form $\neg SynVals(v_1,v_2)$ for each pair (v_1,v_2) of distinct basic values of a functional attribute for which it is known that each possible value has a single form. For instance, $\neg SynVals(\text{“France”}, \text{“Algeria”})$ can be added.

- Resolution-based algorithm for reference reconciliation.

The reasoning is applied to $R \cup F$: the set of rules (put in clausal form) and the set of facts generated as explained before. It aims at inferring all unit facts in the form of $Reconcile(i,j)$, $\neg Reconcile(i,j)$, $SynVals(v_1,v_2)$ and $\neg SynVals(v_1,v_2)$. Several resolution strategies have been proposed so that the number of computed resolutions to obtain the theorem proof is reduced (for more details about these strategies see (Chang & Lee, 1997)). We have chosen to use the unit resolution (Henschen & Wos, 1974). It is a resolution strategy where at least one of the two clauses involved in the resolution is a unit clause, i.e. reduced to a single literal. The unit resolution is complete for refutation in the case of Horn clauses without functions (Henschen & Wos, 1974). Furthermore, it is linear with respect to the size of clause set (Forbus & de Kleer, 1993). The unit resolution algorithm that we have implemented consists in computing the set of unit instantiated clauses contained in F or inferred by unit resolution on $R \cup F$. Its termination is guaranteed because there are no function symbols in $R \cup F$. Its completeness for deriving all the facts that are logically entailed has been stated in (Saïs et al., 2009).

N2R: a Numerical method for Reference Reconciliation

N2R has two main distinguishing characteristics. First, it is fully unsupervised: it does not require any training phase from manually labeled data to set up coefficients or parameters. Second, it is based on equations that model the influence between similarities. In the equations, each variable represents the (unknown) similarity between two references while the similarities between values of attributes are constants that are computed by using standard similarity measures on strings or on sets of strings. The functions modeling the influence between similarities are a combination of maximum and average functions in order to take into account the constraints of functionality and inverse functionality declared in the RFDS+ schema in an appropriate way.

Solving this equation system is done by an iterative method inspired from the Jacobi method (Golub & Loan, 1996), which is fast converging on linear equation systems. The point is that the equation system is not linear, due to the use of the max function for the numerical translation of the functionality and inverse functionality axioms declared in the RFDS+ schema. Therefore, we had to prove the convergence of the iterative method for solving the resulting non linear equation system.
N2R can be applied alone or in combination with L2R. In this case, the results of non-reconciliation inferred by L2R are exploited for reducing the reconciliation space, i.e., the size of the equation system to be solved by N2R. In addition, the results of reconciliations and of synonymies or non-synonymies inferred by L2R are used to set the values of the corresponding constants or variables in the equations.
We first use a simple example to illustrate how the equation system is built. Then, we describe how the similarity dependencies between references are modeled in an equation system and we provide the iterative method for solving it.

Example 2
Let us consider the data descriptions of the example 1 and the reference pairs <S1_r607,S2_r208>, <S_d1e5, S2_l6f2>, <S1_p112,S2_p222> and <S1_p112,S2_p232>.
The similarity score $Sim_r(ref, ref')$ between the references *ref* and *ref'* of each of those pairs is modeled by a variable :
x_1 models Sim_r *(S1_r607, S2_r208)*
x_2 models Sim_r *(S1_p112,S2_p222)*
x_3 models Sim_r *(S1_p112,S2_p232)*
x_4 models Sim_r *(S_d1e5, S2_l6f2)*

We obtain the following equations that model the dependencies between those variables:

$x_1=max(0.68, x_2, x_3, x_4/4)$
$x_2=max(0.1, x_1/2)$
$x_3=max(0.9, x_1/2)$
$x_4=max(0.42, x1)$

In this equation system, the first equation expresses that the variable x_1 strongly and equally depends on the variables x_2 and x_3, and also on 0.68, which is the similarity score between the two strings "*Le Louvre*" and "*musée du Louvre*" computed by the *Jaro-Winkler* function (Cohen et al., 2003). It also expresses that it weakly depends on x_4.
The reason of the strong dependencies is that *contains* is an inverse functional relation (a painting is contained in only one museum) relating S1_r607 and S2_r208 (the similarity of which is modeled by x_1) to S1_p112 for S1_r607 and S2_p222 for S2_r208, and *name* is a functional attribute (a museum has only one name) relating S1_r607 and S2_r208 respectively to the two strings "*Le Louvre"* and "*musee du Louvre*".
The weak dependency of x_4 onto x_1 is expressed by the term $x_4/4$ in the equation, where the ratio ¼ comes from that there are 4 properties (relations or attributes) involved in the data descriptions of S1_r607 and S2_r208. The dependency of x_4 onto x_1 is weaker than the previous ones because *located* is not an inverse functional relation.

The equations modeling the dependencies between similarities
For each pair of references, its similarity score is modeled by a variable x_i and the way it depends on other similarity scores is modeled by an equation: $x_i= f_i(X)$, where $i \in [1..n]$ and n is the number of reference pairs for which we apply N2R, and $X=(x_1, x_2, \ldots, x_n)$. Each equation $x_i= f_i(X)$ is of the form:
$f_i(X) =max(f_{i\text{-}df}(X), f_{i\text{-}ndf}(X))$.

The function $f_{i\text{-}df}(X)$ is the maximum of the similarity scores of the value pairs and the reference pairs of attributes and relations with which the *i*-th reference pair is functionally dependent. The maximum function allows propagating the similarity scores of the values and the references having a strong impact. The function $f_{i\text{-}ndf}(X)$ is defined by a weighted average of the similarity scores of the values pairs (and sets) and the reference pairs (and sets) of attributes and relations with which the *i*-th reference pair is not functionally dependent. Since we have neither expert knowledge nor training data, the weights are computed in function of the number of the common attributes and relations. See (Saïs et al., 2009) for the detailed definition of $f_{i\text{-}df}(X)$ and $f_{i\text{-}ndf}(X)$.

Iterative algorithm for reference pairs similarity computation
To compute the similarity scores, we have implemented an iterative resolution method. At each iteration, the method computes the variables values by using those computed in the precedent iteration.
Starting from an initial vector $X^0=(x_1^0, x_2^0,\ldots, x_n^0)$, the value of the vector X at the *k*-th iteration is obtained by the expression : $X^k=F(X^{k-1})$. At each iteration *k* we compute the value of each x_i^k:
$x_i^k =fi(x_1^{k-1}, x_2^{k-1},\ldots x_n^{k-1})$ until a fixpoint with precision ε is reached. The fixpoint is reached when: $\forall i, |x_i^k - x_i^{k-1}| \leq \varepsilon$. The more ε value is small the more the set of reconciliations may be large.
The complexity of this method is in (n^2) for each iteration, where *n* is the number of variables. We have proved its convergence for the resolution of our equation system.

The similarity computation is illustrated by the following equation system obtained from the data descriptions shown in Example 1. The constants correspond to the similarity scores of pairs of basic values computed by using the Jaro-Winkler measure. The constants involved in the value computation of the variables x_1, x_2, x_3 and x_4 are respectively:

- $b11= Sim_v$ ("Louvre", "musée du Louvre") = 0.68
- $b21= Sim_v$ ("La Joconde", "Iris") = 0.1
- $b31= Sim_v$ ("La Joconde", "Joconde") = 0.9
- $b41= Sim_v$ ("Paris", "Ville de Paris") = 0.42

The weights are computed in function of the number of common attributes and common relations of the reference pairs. The weights used in the value computation of the variables x_1, x_2, x_3 and x_4 are respectively: $\lambda 11 = ¼$, $\lambda 21 = ½$, $\lambda 31 = ½$ and $\lambda 41 = ½$.

We assume that fixpoint precision ε is equal to 0.005.

The equation system is the one given in Example 2. The different iterations of the resulting similarity computation are provided in Table 1.

Iterations	0	1	2	3	4
x_1=max(0.68, x_2,x_3,¼ *$_{x4}$)	0	0.68	0.9	0.9	0.9
x_2=max(0.1, ½ * x_1)	0	0.1	0.34	0.45	0.45
x_3=max(0.9, ½ * x_1)	0	0.9	0.9	0.9	0.9
x_4=max(0.42, x_1)	0	0.42	0.68	0.9	0.9

Table 1 –Example of iterative similarity computation

The solution of the equation system is $X=(0.9,0.45,0.9,0.9)$. This corresponds to the similarity scores of the four reference pairs. The fixpoint has been reached after four iterations. The error vector is then equal to 0. If we fix the reconciliation threshold T_{rec} at 0.80, then we obtain three reconciliation decisions: two cities, two museums and two paintings.

Experiments

L2R and N2R have been implemented and tested on the benchmark Cora[ii] (used by (Dong et al., 2005; Parag & Domingos, 2005)). It is a collection of 1295 citations of 112 different research papers in computer science. For this data set, the UNA is not stated and the RDF facts describe references, which belong to three different classes (*Article*, *Conference*, *Person*). We have designed a simple RDFS schema on the scientific publication domain, which we have enriched with disjunction constraints (e.g. DISJOINT(*Article*, *Conference*)), a set of functional property constraints (e.g. PF(*published*), PF(*confName*)) and a set of inverse functional property constraints (e.g. PFI(*title*, *year*, *type*), PFI(*confName*, *confYear*)). The recall and the precision can be easily obtained by computing the ratio of the reconciliations or non-reconciliations obtained by L2R and N2R among those that are provided in the benchmark.

L2R results

Since the set of reconciliations and the set of non-reconciliations are obtained by a logical resolution-based algorithm the precision is of 100% by construction. Then, the measure that it is meaningful to evaluate in our experiments is the recall. We focus on the results obtained for the *Article* and *Conference* classes, which contain respectively 1295 references and 1292 references.

	RDFS+	RDFS+ & DP
Recall (REC) Recall (NREC)	52.7 % 50.6 %	52.7 % 94.9 %
Recall	50.7 %	94.4 %
Precision	100 %	100 %

Table 2. L2R results on Cora data sets

As presented in the column named “RDFS+” of the Table 2, the recall is 50.7%. This can be refined in a recall of 52.7% computed on the REC subset and a recall of 50.6% computed on NREC subset.
For this data set, the RDFS+ schema can be easily enriched by the declaration that the property *confYear* is discriminant. When this property is exploited, the recall on NREC subset grows to 94.9%, as it is shown in the “RDFS+ & DP” column. This significant improvement is due to chaining of different rules of reconciliations: the non-reconciliations on references to conferences for which the values of the *confYear* are different entail in turn non-reconciliations of the associated articles by exploiting the constraint PF(*published*).
This recall is comparable to (while a little bit lower than) the recall on the same data set obtained by supervised methods like e.g., (Dong et al., 2005). The point is that L2R is not supervised and guarantees a 100% precision.

N2R Results
In the following we presents the results (see Figure 6) obtained by N2R after the application of L2R.

For $T_{rec}=1$, N2R do not obtain more results than L2R. The evolution of the recall and precision values in function of T_{rec} is interesting. Indeed, when the threshold is decreased to 0.85, the recall increases by 33% while the precision only falls by 6%. The best results are obtained when $T_{rec}=0.85$. The F-measure is then at its maximum value of 88%. Besides, when the recall value is almost of 100%, for $T_{rec}=0.5$, the precision value is still about 40%.

The exploitation of the non-reconciliation inferred by L2R allows an important reduction of the reconciliation space handled in N2R. For the Cora data set the size of the reconciliation space is about 37 millions of reference pairs. It has been reduced of 32.8 % thanks to the correct no reconciliations inferred by L2R. Moreover, the reconciliations inferred by L2R are not recomputed in N2R.

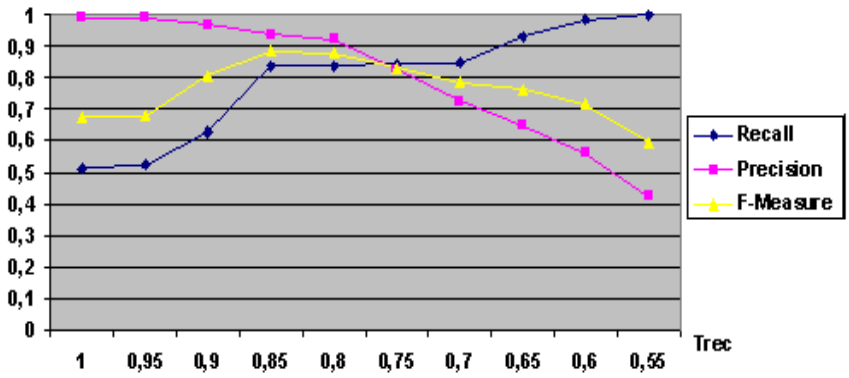

Figure 6. N2R results obtained on Cora data set

These experimentations show that good results can be obtained by an automatic and unsupervised method if it exploits knowledge declared in the schema. Furthermore, the method is able to obtain F-Measure which is better than some supervised methods such that (Parag & Domingos, 2005).

FUTURE RESEARCH DIRECTIONS

Work on integration has evolved in recent years to heterogeneous information sources available from the Web or through the intranets and the heterogeneity is becoming more and more important. In the future, we will have to deal with a still larger variety of data structures types (structured, semi-structured or unstructured sources), but also with sources containing both semi-structured and unstructured (or textual) parts. Current wrapper approaches could not be applied with the last kind of sources. Suitable data extraction and transformation techniques are required. In fact, the more heterogeneous documents are, the more complex data integration is. The key issue for

integrating systems that are more and more heterogeneous is to understand them. Semantic Web techniques will have an increasing role to play in the future in order to facilitate this understanding. Indeed, the concept of ontology which makes possible to add semantic information to the Web and the basic representation languages for the Semantic Web which allow reasoning on the content of sources are the foundations to obtain this understanding.

Reconciliation is an important information integration problem. It arises in other fields such as database area when data from various sources are combined. For example, mailing lists may contain several entries representing the same physical address, each entry containing different spellings. Identifying matching records is challenging because there are no unique identifiers across databases. Satisfactory solutions are not available yet. In all the applications where this problem arises, methods that are efficient while ensuring good results and being not vulnerable to changes of application domain are really required. Furthermore, since sources are more and more accessed from the Internet, additional problems appear and have to be studied: dealing with data freshness in order to store the freshest possible data, dealing with trust into sources which provide data, being capable to consider access rights when querying the most reliable sources.

Generally speaking, automatic methods will be of great importance in the future. Several directions or research can be taken. Unsupervised methods that guarantee a 100 % precision of the results if schema and data are error-free are one way to automate reconciliation. Indeed, they allow obtaining reconciliations and non reconciliations that are sure. Capitalization on experience so that methods become more efficient as they are applied is another interesting direction. For example saving the correct (no) synonymies inferred by L2R in a dictionary is an illustration of capitalization. It allows learning the syntactic variations of an application domain in an automatic and unsupervised way.

Finally, the demand for methods that ensure good results and which can be applied on new data again and again while remaining as efficient as ever will increase. Today there are a lot of difficulties to estimate in advance the precision of a system when it is applied to a new set of data. As a consequence two research objectives should be favored in a near future. A first one is to elaborate generic methods that guarantee sure results (a logical method of the kind of L2R for example). Such methods are very interesting but they can not be used in any case especially when the data is “dirty” or the global schema is an integrated schema resulting from an automatic matching process. Furthermore they must be complemented by others in order to obtain a better recall. A second objective is to propose methods, which reconcile data on the basis of similarity scores (not necessarily 100 %) designed together with mechanisms capable to reason on the uncertain reconciliation decisions. That means that uncertainty management will become a major challenge to be taken up. Uncertainty gathered in data warehouses while populating them will have to be exploited by reasoning on tracks of reconciliation decisions.

CONCLUSION

We have presented an information integration approach able to extract, transform and integrate data in a data warehouse guided by an ontology. Whatever the application domain is, the approach can be applied to XML sources that are valid documents and that have to be integrated in a RDF data warehouse with data described in terms of a RDFS ontology. Mappings between the external sources and the ontology are represented in a declarative way. Their definition is made apart from the extraction process. Extraction operates on any XML document given mappings represented in XPath in terms of the ontology. Data transformation consists in converting data in terms of the ontology and in the same format. Both tasks are performed through XML queries associated to views of the sources automatically built beforehand. Through data integration, we addressed the reference reconciliation problem and presented a combination of a logical and numerical approach. Both approaches exploit schema and data knowledge given in a declarative way by a set of constraints and are then generic. The relations between references are exploited either by L2R for propagating (non) reconciliation decisions through logical rules or by N2R for propagating similarity scores thanks to the resolution of the equation system. The two methods are unsupervised because no labeled data set is used. Furthermore, the

combined approach is able to capitalize its experience by saving inferred (non) synonymies. The results that are obtained by the logical method are sure. This distinguishes L2R from other existing works. The numerical method complements the results of the logical one. It exploits the schema and data knowledge and expresses the similarity computation in a non linear equation system. The experiments show promising results for recall, and most importantly its significant increasing when constraints are added.

REFERENCES

Abiteboul, S., Cluet, S., & Milo, T. (1997). Correspondence and translation for heterogeneous data. In *Proceedings of the International Conference on DataBase Theory*, (pp. 351-363).

Baxter, R., Christen, P., & Churches, T. (2003). A comparison of fast blocking methods for record linkage. In *Proceedings of ACM SIGKDD'03 Workshop on Data Cleaning Record Linkage and Object Consolidation*, Washington, DC, USA, (pp. 25-27).

Berglund, A., Boag, S., Chamberlin, D., Fernandez, M.F., Kay, M., Robie, J., & Simeon, J. (2007). XML Path Language (XPath) 2.0, from http://www.w3.org/TR/xpath20/.

Bhattacharya, I., & Getoor, L. (2006). Entity Resolution in Graphs. In L. B. Holder, D.J. Cook (Eds.), *Mining Graph Data*. Wiley.

Bilke, A., & Naumann, F. (2005). Schema Matching using Duplicates. In *Proceedings of the International Conference on Data Engineering*, (pp. 69-80).

Boag, S., Chamberlin, D., Fernandez, M. F., Florescu, D., Robie, J., & Simeon, J. (2007). XQuery 1.0 : An XML Query Language. W3C Recommendation, from http://www.w3.org/TR/xquery/.

Chang, C., & Lee, R. C. (1997). *Symbolic Logic and Mechanical Theorem Proving*. New-York: Academic Press.

Cluet, S., Delobel, C., Simeon J., & Smaga, K. (1998). Your mediators need data conversion ! In *Proceedings of SIGMOD'98*, Seattle, USA, (pp. 177-188).

Cohen, W. W. (2000). Data integration using similarity joins and a word-based information representation language. *ACM Transactions on Information Systems*. 18(3), 288-321.

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks, In *Proceedings of the Workshop on Information Integration on the Web*, (pp. 73-78).

Dey, D., Sarkar, S., & De, P. (1998). Entity Matching in Heterogeneous Databases: A Distance Based Decision Model. In *Proceedings of The Thirty-First Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington, DC, USA, (pp. 305-313).

Dey, D., Sarkar, S., & De, P. (1998). A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases. *Management Science*. 44(10), 1379-1395.

Dong, X., Halevy, A., & Madhavan, J. (2005). Reference reconciliation in complex information spaces, In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM Press, (pp. 85-96).

Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*. 64(328), 1183-1210.

Forbus, K. D., & De Kleer, J. (1993). *Building problem solvers*. USA:MIT Press.

Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*. 3rd Ed, Baltimore, MD, USA: Johns Hopkins University Press.

Hayes, P. (2004). RDF Semantics, from http://www.w3.org/TR/rdf-mt/.

Henschen, L. J., & Wos, L. (1974). Unit Refutations and Horn Sets. *Journal of the Association for Computing Machinery (ACM)*, 21(4), 590-605.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, from http://www.w3.org/Submission/SWRL.

Koffina, I., Serfiotis, G., & Christophides, V. (2006). Mediating RDF/S Queries to relational and XML Sources. *International Journal on Semantic Web & Information Systems,* 2(4), 78-91.

McBride, B. (2004). The resource Description Framework (RDF) and its Vocabulary Description Language RDFS. In S. Staab, R. Studer (Eds.), *Handbook on Ontologies*. (pp. 51-66). Springer.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL: Web Ontology Language Overview. *W3C recommendation*, from http://www.w3.org/TR/owl-features.

Newcombe, H. B., & Kennedy, J. M. (1962). Record linkage: making maximum use of the discriminating power of identifying information. *Communications of the ACM*, 5(11), 563-566.

Newcombe, H. B., Kennedy, J. M., Axford, S. J. & James, A. P. (1959). Automatic linkage of vital records. *Science,* 130, 954-959.

Parag, S., & Pedro, D. (2004). Multi-relational record linkage. In *Proceedings of the ACM SIGKDD Workshop on Multi-Relational Data Mining*, (pp. 31-48).

Popa, L., Velegrakis, Y., Miller, R. J., Hernadez, M. A., & Fagin, R. (2002). Translating Web data. In *Proceedings of the VLDB Conference*, (pp. 598-609).

Reynaud, C., & Safar, B. (2009). Construction automatique d'adaptateurs guidée par une ontologie pour l'intégration de sources et de données XML. *Technique et Science Informatiques*. Numéro spécial Web Sémantique, 28(2).

Robinson, A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. *Journal of Association for Computing Machinery,* 12(1), 23-41.

Rousset, M.-C., & Reynaud, C. (2003). Picsel and Xyleme: Two illustrative Information Integration Agents. In M. Klusch, S. Bergamaschi, P. Petta, P. Edwards (Eds.), *Intelligent Information Agents Research and development in Europe*: *An AgentLink Perspective*. Springer Verlag, LNCS State of the Art Surveys, 50-78.

Rousset, M.-C., Bidault, A., Froidevaux, C., Gagliardi, H., Goasdoué, F., Reynaud, C., & Safar, B. (2002). Construction de médiateurs pour intégrer des sources d'information multiples et hétérogènes : le projet PICSEL. *Revue I3*, 2(1), 9-59.

Saïs, F., Pernelle, N., & Rousset, M.-C. (2009). Combining a Logical and a Numerical Method for Data Reconciliation. *Journal of Data Semantics*, LNCS 5480, 12, 66-94.

Saïs, F., Pernelle, N., & Rousset, M.-C. (2007). L2R: A Logical Method for Reference Reconciliation, In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, (pp. 329-334).

ADDITIONAL READINGS

Amann, B., Beeri, C., Fundulaki, I., & Scholl, M. (2002). Querying XML sources using an ontology-based mediator. In R. Meersman and Z. Tari (Eds), In *Proceedings of Confederated International Conferences Doa, CoopIS and ODBASE,* London: Springer-Verlag, (pp. 429-448).

Batini, C. & Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems And applications)*, New-York: Springer Verlag.

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E. & Widom, J. (2009). Swoosh: a generic approach to entity resolution, *VLDB Journal.*.

Bilenko, M., Mooney, R. J., Cohen, W. W.; Ravikumar, P. & Fienberg, S. E. (2003). Adaptive Name Matching in Information Integration, *IEEE Intelligent Systems* 18(5), 16-23.

Calvanese, D., Miller, R., & Mylopoulos, J. (2005). Representing and Querying Data Transformations. In *Proceedings of International Conference on Data Engineering*, (pp. 81-92).

Christophides, V., Karvounarakis, G., Magkanaraki, A., Plexousakis, D., Vouton, V., Box, B. & Tannen, V. (2003). The ICS-FORTH Semantic Web Integration Middleware (SWIM). *IEEE Data Engineering Bulletine*, 26(4), 11-18.

Cohen, W. W. (1998). Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity, In *Proceedings of the SIGMOD Conference*, (pp. 201-212).

Doan, A., Lu, Y., Lee, Y. & Han, J. (2003). Profile-Based Object Matching for Information Integration, *IEEE Intelligent Systems,* 18(5), 54- 59.

Hammer, J., Garcia-Molina, H., Nestorov, S., Yerneni, R., Breunig, M. & Vassalov, V. (1997). Template-Based wrappers in the TSIMMIS System. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, (pp. 532-535).

Hernandez, M. A. & Stolfo, S. J. (1998). Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem, *Data Mining and Knowledge Discovery* 2(1), 9-37.

Herzog, T. N., Scheuren, F. J. & Winkler, W. E. (2007). *Data Quality and Record Linkage Techniques*, New-York: Springer Verlag.

Kalashnikov, D., Mehrotra, S. & Chen., Z. (2005). Exploiting relationships for domain-independent data cleaning. *SIAM Data Mining*.

Klein, M. (2002). Interpreting XML via an RDF Schema. In *Proceedings of the International Workshop on Database and Expert Systems Applications*, (pp. 889-893).

Kuhlins, S., & Tredwell, R. (2002). Toolkits for Generating Wrappers – A Survey of Software Toolkits for Automated Data Extraction from Websites. In M. Aksit, M. Mezini, R. Unland (Eds.), *International Conference NetObjectDays* (pp. 184-198). LNCS 2591, Springer.

Laender, H.F., Ribeiro-Neto, B. A., Da Silva, A., & Teixeira, J.S. (2002). A Brief Survey of Web Data Extraction Tools. *ACM SIGMOD Record*, 84-93.

Lenzerini, M. (2002). Data integration: a theoretical perspective, In *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems,* New York: ACM Press, (pp. 233-246).

Liu, L., Pu, C., & Han, W. (2000). XWRAP : An XML-enabled Wrapper Construction System for Web Information Sources. In *Proceedings of the International Conference on Data Engineering*, (pp. 611-621).

Miklos, Z., & Sobernig, S. (2005). Query Translation between RDF and XML: A Case Study in the Educational Domain. In *Proceedings of the Workshop of the Interoperability of Web-Based Educational Systems*.

Miller, R., Haas, A., & Hernandez, M., (2000). Schema Mapping as Query Discovery. *International Conference on VLDB* , 77-88.

Miller, R., Hernandez, M., Haas, A., Yan, L., Howard Ho, C., Fagin R., & Popa, L. (2001). The Clio Project: Managing Heterogeneity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, (pp. 78-83).

Milo, T., & Zohar, S. (1998). Using Schema matching to Simplify Heterogeneous Data Translation. In *Proceedings of the International Conference on VLDB*, (pp.122-133).

Monge, A. E. & Elkan, C. (1997), An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records, *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 23-29.

Saïs, F. & Thomopoulos, R. (2008). Reference Fusion and Flexible Querying, In *Proceedings of the Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, (pp. 1541-1549).

Velegrakis, Y., De Giacomo, G., Lenzerini, M., Nardo, M. & Riccardo, R. (2001). Data Integration in Data Warehousing. *International Journal of Cooperative Information Systems*, 10(3), 237-271.

Velegrakis, Y., Millar, R. & Mylopoulos, J. (2005). Representing and Querying Data Transformations. In *Proceedings of the International Conference on Data Engineering*, (pp. 81-92).

Verykios, V. S., Moustakides, G. V. & Elfeky, M. G. (2003). A Bayesian Decision Model for Cost Optimal Record Matching*, The VLDB Journal*, 12(1), 28-40.

KEY TERMS & DEFINITIONS

Data integration: In this chapter, data integration means data reconciliation.

Data warehouse: It contains data defined in terms of an ontology. These data come from different heterogeneous sources, are transformed according to the ontology of the data warehouse and then reconciled.

Mappings: correspondence relations between a global schema or ontology and the schemas of data sources.

Mediator-based approach: An approach integrating multiple data sources which can be syntactically or semantically heterogeneous while related to a same domain (e.g., tourism, culture). It provides a uniform interface for querying collections of pre-existing data sources that were created independently. Mediator systems are based on a single mediated schema in terms of which users' queries are issued and the information sources to integrate are described. The descriptions specify semantic relationships between the contents of the sources and the mediated schema. A user query that is formulated on a mediated schema is translated into a query against local schemas using views. Query plans are computed and executed through wrappers in order to get the answers to the user query. The goal is to give users the illusion that they interrogate a centralized and homogeneous system.

Ontology: A model of the objects of an application domain composed of concepts, attributes of concepts and relations between concepts.

Reference reconciliation: The reference reconciliation problem consists in deciding whether different identifiers refer to the same data, i.e., correspond to the same world entity.

Unsupervised approach: An approach where the program is not trained by some *data* that are labeled with the desired output and which are provided by human experts.

Wrapper: Systems that aim at accessing a source, extracting the relevant data and presenting such data in a specified format.

---

[i] A research project funded by France Telecom R&D (2005-2008)

[ii] Another version of Cora is available at http://www.cs.umass.edu/~mccallum/data/cora-refs.tar.gz