WORKSHOP
Interoperability through Semantic Data and Service Integration

Camogli (Genova), Italy
June 25th, 2009
Co-located with SEBD

Proceedings
WORKSHOP ORGANIZERS

CO-CHAIR
Sonia Bergamaschi, Università di Modena e Reggio Emilia, Italy
Tiziana Catarci, Sapienza Università di Roma, Italy
Carlo Batini, Università di Milano-Bicocca, Italy
Domenico Beneventano, Università di Modena e Reggio Emilia, Italy

ORGANIZING COMMITTEE
Carola Aiello Università di Roma
Serena Sorrentino Università di Modena e Reggio Emilia

PROGRAM COMMITTEE
Omar Boucelma, University of Aix-Marseille, France
Stuart Campbell, TIE Nederland B.V., Nederland
Dario Cerizza, CEFRIEL, Milano, Italy
Isabel Cruz, University of Illinois at Chicago, USA
Alfredo Cuzzocrea, University of Calabria, Italy
Piero De Sabbata, ENEA, SIC-UDA-PMI Bologna, Italy
Kevin Ginty, Centre for Internet Technologies, UK
Francesco Guerra, University of Modena and Reggio Emilia, Italy
Axel Hahn, University of Oldenburg, Germany
Maurizio Lenzerini, University of Rome La Sapienza, Italy
Andrea Maurino, University of Milano Bicocca, Italy
Michele Missikof, LEKS-IASI CNR, Roma, Italy
Gianluca Moro, University of Bologna, Italy
Aris M. Ouksel, University of Illinois at Chicago, USA
Fausto Rabitti, ISTI-CNR, Pisa, Italy
Claudio Sartori, University of Bologna, Italy
Maurizio Vincini, University of Modena and Reggio-Emilia, Italy
# Table of contents

Semantic Services for Business Documents Reconciliation - *E. Del Grosso, M. Missikov, F. Smith and F. Taglino*  
Searching for Data and Services - *F. Guerra, A. Maurino, M. Palmonari, G. Pasi and A. Sala*  
Combining Semantic and Multimedia Query Routing Techniques for Unified Data Retrieval in a PDMS - *C. Gennaro, F. Mandreoli, R. Martoglia, M. Mordacchini, W. Penzo and S. Sassatelli*  
The MOMIS - STASIS approach for Ontology-based Data Integration - *D. Beneventano, M. Orsini, L. Po and S. Sorrentino*  
A Framework For Privacy Preserving Face mAtching - *F. Volpi, M. Scannapieco and T. Catarci*  
Towards Automatic Support to Mapping discovery in eBusiness resources Exchange - *M. Missikoff, F. Smith and F. Taglino*  
Abbreviation Expansion In Lexical Annotation of Schema - *M. Gawinecki*  
Service-Finder: First Steps toward the realization of Web Service Discovery at Web Scale - *S. Brockmans, I. Celino, D. Cerizza, E. Della Valle, M. Erdmann, A. Funk, H. Lausen, W. Schoch, N. Steinmetz and A. Turati*  
New approach for user centric ontology design - *N. Dahlem, A. Hahn and M. Reinelt*  
Semantic Analysis for an Advanced ETL framework - *S. Bergamaschi F. Guerra, M. Orsini, C. Sartori and M. Vincini*  
Query Processing in a Mediator System for Data and Multimedia - *D. Beneventano, C. Gennaro, M. Mordacchini and R. C. Nana Mbinkeu*  
Semantic Access to Data from the Web - *R. Trillo, L. Po, S. Ilarri, S. Bergamaschi and E. Mena*  

Page numbers:
Semantic Services for Business Documents Reconciliation

Enrico Del Grosso\(^1\), Michele Missikoff\(^2\), Fabrizio Smith\(^2\), Francesco Taglino\(^2\)

\(^1\) TXT e-Solutions, Italy (http://www.txtgroup.com/it), enrico.delgrosso@txtgroup.com
\(^2\) IASI-CNR, Italy (http://leks-pub.iasi.cnr.it), michele.missikoff@iasi.cnr.it, fabrizio.smith@iasi.cnr.it, francesco.taglino@iasi.cnr.it

Abstract. This paper describes the innovative work that the COIN\(^1\) project is developing in the context of the information interoperability using semantic approaches. The starting point is represented by the semantic reconciliation suite developed in the European project Athena\(^2\). Then we aim at improving and enhancing the previous solution in order to ease the use of the platform itself with the support of automatic services\(^3\).

Keywords: semantics, service, ontology, interoperability

1 Introduction

In this paper we describe a service oriented architecture for supporting enterprise semantic interoperability. The goal of semantic interoperability is to allow the (seamless) cooperation of enterprise software applications, which were not initially developed for this purpose, by using ontology-based semantic techniques. In particular, we will focus on the exchange of business documents, which takes place

- between different enterprises, for instance, to automatically transfer a purchase order from a client to a supplier;
- within the same enterprise, for instance, to share documents between different departments which use different data organization.

A relevant work on this field has been done by the Athena project, where a semantic reconciliation suite for reconciliation of business documents was developed. Such a suite has been now acquired by the COIN project as part of the so called baseline services, which will be the starting point for the development of innovative services.

The objective of this paper is to describe the on going work in the COIN project aimed at improving the Athena semantic reconciliation suite in order to provide an automatic support to the usage of the platform itself. In fact, some of the steps required by the platform, the semantic annotation (declarative mapping) of business documents and the definition of transformation rules (operational mapping) are currently mainly manual activities, being therefore time consuming and error prone. For this reason, a set of innovative services will be developed in the COIN project

\(^1\) http://www.coin-ip.eu/
\(^3\) This work has been partially supported by the EU FP7 IST 216256 IP COIN European Project
with the aim of automatically support and optimize certain steps of the reconciliation process.

It is useful to note that **declarative mapping** represents a conceptual correspondence between a document schema and an ontology-based expression, while the **operational mapping** has the knowledge of how to actually transform ground resources (i.e., data) from a format/representation into another.

The rest of the paper is organized as follows: in Section 2, an overview of the approach followed by the Athena semantic reconciliation suite is given; section 3,4 and 5 present an overview of the three innovative services: (i) declarative mapping discovery; (ii) operational mapping generation; (iii) transformation rules fusion; section 6 reports about existing platforms for semantic reconciliation of documents; the Conclusion section will end the paper.

## 2  A brief recap of the Athena Semantic Reconciliation approach

The Athena semantic reconciliation approach is based on the use of a domain ontology as common reference for the harmonization of heterogeneous data. This harmonization is accomplished into two different phases: a preparation phase and a run time phase (Figure 1).

![Athena Semantic Reconciliation approach](image)

In the preparation phase, the schemas of documents to be shared by the software applications (say, App\textsubscript{A}, App\textsubscript{B}, App\textsubscript{C}) are mapped against the reference ontology. The mapping is a two steps activity, which starts from the semantic annotation and ends with the building of semantic transformation rules.
The semantic annotation allows the description of a document (or its parts), in terms of the reference ontology, by identifying conceptual correspondences between a schema and the reference ontology. Nevertheless, semantic annotations do not contain enough information for performing actual data reconciliation. For this reason, operational semantic transformation rules are built. In particular, starting from the previous annotation, for each document schema, a pair of transformation rules sets is built: a forward rule set and a backward rule set, which allow data transformation from the original format into the ontology representation and vice-versa, respectively.

The run-time phase concerns the actual exchange of data from an application to another. For instance, when the application AppC sends a document to the application, AppB, the reconciliation between the format of AppC and the format of AppB, is done by applying first the forward rules set of AppC, and after the backward rules set of AppB. The application of the first set of rules produces a representation of the document in terms of the reference ontology, while the application of the second set of rules to this intermediate representation, produces the desired result, that is the document expressed in terms of the format managed by the destination application.

Fig. 2. Semantic Interoperability Services Architecture

The objective within COIN is to enrich and extend the semantic reconciliation suite in three directions (Figure 2):

- **Automatic support to Semantic Annotation**: currently, the semantic annotation step is mainly performed manually. The aim is to develop a declarative mapping discovery service to help in the identification of semantic correspondences between a document schema and the reference ontology.
• **Automatic support to Semantic Transformation Rules generation:** reusing the knowledge representing the declarative mapping, the operational mapping generation service will support the building of transformation rules;

• **Semantic Transformation Rules fusion:** a transformation rules fusion service will allow the optimization of the run time phase. By avoiding the generation of the intermediate ontological representation of the exchanged document, the rules fusion will also avoid the application of useless transformations with respect to the aim of transferring a document between applications

3 **Declarative Mapping Discovery Service**

The objective of this service is to provide a support to the declarative mapping discovery between a structured business document schema (e.g., a purchase order) and a reference ontology. The service will represent a semi-automatic support to the definition of semantic annotation of business documents schemas. In fact, semantic annotation is a time expensive and error prone activity, especially when huge documents and huge ontologies are involved. In the Astar [1], the semantic annotation tool of the Athena reconciliation suite this activity is mainly manual.

As a semi-automatic support, a final validation by a human actor will be needed.

The **declarative mapping discovery service** will be articulated in three incremental steps, which are:

• **Terminology-based mapping.** This step involves only terms from the document schema and the reference ontology; no knowledge about the structure of the two resources is here used. The objective of this first mapping step is to identify, for each pair of terms, with a term from the schema and the other from the ontology, a similarity degree (term-sim). The terminological mapping discovery is based on the combination of a string matching technique (e.g., the Monge-Elkan measure [7]), which see words just as a sequence of letters, and a linguistic-based technique exploiting morphological properties of the input words (i.e., the Lin [4] similarity metrics) and the use of external linguistic resources (e.g., WordNet[^4]) to enrich the knowledge carried by each term.

• **Structure-based mapping.** The structural mapping works on paths, which are more complex fragments of the document schema and the reference ontology. Since the two resources are expressed in RDF/OWL, a path is a sequence of concatenated classes and properties. The objective is to discover correspondences between path of the document and of the RO according to a similarity metrics (path-sim). The previously defined terminology-based mapping will be used for supporting such a discovery.

• **Semantic mismatch patterns identification.** We briefly recall that in the design of the Athena reconciliation suite, two classes of semantic mismatch patterns (kinds of differences to reconcile) were identified: lossless and lossy

mismatches [1]. The former (e.g., different terminology, different structure, etc) can be addressed without loosing any information, while the latter (e.g., information coverage, etc) can cause loosing of information. The objective of this third mapping step is to discover, by using the knowledge acquired at the structural level, what is the mismatch interested by each path from the document schema, and to build semantic expressions by using a pre-defined set of abstract operators (i.e., unary and binary operators).

The mapping discovery has to be intended as a suggestion to the user, who should, in order to make the process more reliable, to validate the results of each step.

4 Operational mapping generation service

The objective of the operational mapping generation service is to provide a semi-automatic support to the definition of operational mapping (i.e., transformation rules), starting from the previously defined declarative mapping.

A declarative mapping is not able to fully represent how to actually transform data from a format to another. Nevertheless, the knowledge carried by the declarative mapping is extremely useful for generating actual transformation rules.

Respect to the current status of the semantic reconciliation suite, the objective of this service is:

- to really exploit the semantic annotation for supporting in a semi-automatic way the generation of language independent transformation rules;
- to allow the transformation of the previous result into actual executable rules specific for a certain transformation engine.

The service will work according to the following steps:

- For each declarative mapping, the identification of proper transformation rule template to be instantiated. This is identified by considering the correspondences between semantic mismatch patterns and predefined transformation rule templates (i.e., the kinds of rules pre-defined in accordance with the mismatch patterns).
- Automatic partial instantiation of the rule template by using the knowledge coming from the declarative mapping. At this step, generic operators used at declarative level are substituted with concrete operators (e.g., a binary operator substituted by a string concatenation operator).
- Rule validation and completion by a human user. Not all the knowledge needed for generating a transformation rule is contained in the annotation. For instance, concatenating two strings (e.g., firstname and lastname) needs the specification of a separator between them. After this step, the operational mapping according to an execution independent syntax (i.e., RIF based [8]) is built.
• Generation of the executable rules by transforming the execution independent rules into an executable syntax (e.g., Jena\textsuperscript{2}, ATL\textsuperscript{3}, Prolog\textsuperscript{4}).

5 Semantic transformation rule fusion service

Currently, in the Athena semantic reconciliation suite, the actual reconciliation phase, the run-time step, happens by applying first the forward set of rules of the sender, which allow the transformation from the original format to the ontology representation, and then the backward rules of the receiver, which allow the transformation from the ontology representation to the receiver’s format. The objective of this service is to provide an automatic fusion of the forward and backward rules in order to

• Have \textit{straight forward reconciliation} (no generation of the intermediate ontology-based representation of the original document is performed).

• \textit{Avoid useless transformations}. The rules fusion will optimize the rules for transforming data from the sender to the receiver. For instance, if the sender and the destination formats of a document represent the name of a person with one single field, say \textit{sName} and \textit{dName} respectively, and the ontology with two fields, say \textit{firstname} and \textit{lastname}, the information transfer will need a split rule from \textit{sName} to \textit{firstname} and \textit{lastname}, and a merge rule from \textit{firstname} and \textit{lastname} to \textit{dName}. The application of these two rules is actually useless, because the data organization of the sender and the receiver have the same structure even if they use different terms (i.e., \textit{sName} and \textit{dName}). In essence, the rule fusion should allow this situation to be overcome by producing one single rule focused on the only terminological mismatch.

The rule fusion must also preserve the semantics of the fused rules, in the sense that the application of the resulting rule after the fusion must have the same effect of the application of the two original sets of rules.

For what concerns the procedure of the fusion, logic programming based techniques will be investigated. Logic programming languages \cite{5}, and in particular Prolog, have been proposed as languages for representing and transforming XML documents. Logic programming languages enjoy some desirable properties: they have a declarative semantics based on first order logic that makes them suitable for reasoning tasks and, at the same time, they have an operational meaning that makes them suitable for computation.

\textsuperscript{5} http://jena.sourceforge.net/
\textsuperscript{6} http://www.eclipse.org/m2m/atl/
\textsuperscript{7} http://www.swi-prolog.org/
6 Related work

AMEF, the ARTEMIS Message Exchange Framework [2] for document reconciliation is the result of the ARTEMIS project. It allows the mediation of two OWL ontologies which represent the schemas of the documents to be reconciled. For this reason, the schemas of the documents to be reconciled are previously transformed into OWL by using a lift and normalization process. The semantic mediation is realized in two phases: (i) Message Ontology Mapping Process, where the two ontologies are mapped one to another, in order to build Mapping definitions (i.e., transformation rules), with the support of the OWLmt ontology mapping tool; (ii) Message Instance Mapping, where XML instances are first transformed into OWL instances, and then Mapping definitions are applied to transform messages from the original to the destination format.

The MAFRA (MApping FRAmework) [6] is a framework for mapping distributed ontologies. It is based on the definition of Semantic Bridges as instances of a Semantic Bridging ontology which represents the types of allowed bridges. Such bridges represent transformation rules. Also in this case a lift and normalization activity is performed to transform the original documents (schemas and data) into the ontology format. Afterwards, considering the transformed schemas, the Semantic Bridges between the two parties are created on the basis of the Semantic Bridge Ontology (SBO).

WSMT is a tool for defining mappings between ontologies. It is implemented as an Eclipse plug-in, as part of the Web Service Modeling Toolkit (WSMT) [3]. WSMT expresses the mappings in the Abstract Mapping Language proposed in [9, 10]. As an abstract language, it does not commit to any existing ontology representation language. Two main aspects characterize the tool. The first one is an automatic support for discovering correct mappings. It is accomplished by using a set of suggestion algorithms for both lexical and structural analysis of the concepts. As a result, for each pair of items (one from the source and the other from the target ontology) a so called eligibility factor is computed, and indicates the degree of similarity between the two items.

With respect to the above platforms, our approach assumes the existence of a common reference ontology against which the mapping of the schema of the documents to reconcile has to be performed.

7 Conclusions

In this paper we presented an overview of the ongoing work about semantic reconciliation of business documents between heterogeneous software applications, as part of the activities of the COIN European project. In particular, we started from the previous results produced in the ATHENA European project concerning the development of a semantic reconciliation suite. The suite is based on a domain ontology which represents the common reference that enables the documents

---

8 http://www.srdc.metu.edu.tr/webpage/projects/artemis/
9 http://www.w3.org/TR/owl-features/
exchange. The ATHENA approach is organized into a preparation and a run-time phase. In the preparation phase, the schemas of the documents to be reconciled are mapped against the reference ontology. This mapping starts with the definition of semantic annotation (i.e., a declarative mapping) and ends with the definition of semantic transformation rules (i.e., operational mapping). In the run-time phase, the rules are applied to the actual ground documents (the one which carry factual data) to allow the exchange of documents between software application that organize data in different manners.

Starting from the ATHENA suite, we are now addressing our efforts towards a set of services to be integrated in the suite and aiming at supporting a semi-automatic definition of the mapping, both declarative and operational, and optimizing the run time phase. The work is started in March of this year. A first implementation of the declarative mapping discovery mechanism will be available at the end of this summer, while the complete work is due by end of 2010.

8 References

Searching for Data and Services*

Francesco Guerra\(^2\) and Andrea Maurino\(^1\) and Matteo Palmonari\(^1\) and Gabriella Pasi\(^1\) and Antonio Sala\(^2\)

\(^1\) Università di Milano Bicocca
Viale Sarca 336, 20126 - Milano - Italy
\{maurino,palmonari,pasi\}@disco.unimib.it

\(^2\) Università di Modena and Reggio Emilia,
Via vignolese 905,41100 Modena - Italy
\{firstname.lastname\}@unimore.it

Abstract. The increasing availability of data and eServices on the Web allows users to search for relevant information and to perform operations through eServices. Current technologies do not support users in the execution of such activities as a unique task; thus users have first to find interesting information, and then, as a separate activity, to find and use eServices. In this paper we present a framework able to query an integrated view of heterogeneous data and to search for eServices related to retrieved data.

1 Introduction

The research on data integration and service discovering has involved from the beginning different (not always overlapping) communities. As a consequence, data and services are described with different models, and different techniques to retrieve data and services have been developed. Nevertheless, from a user perspective, the border between data and services is often not so definite, since data and services provide a complementary vision about the available resources: data provide detailed information about specific needs, while services execute processes involving data and returning as well an informative result.

Users need new techniques to manage data and services in a unified manner: both the richness of information available on the Web and the difficulties the user faces in gathering such information (as a service result, as a query on a form, as a query on a data source containing information extracted from web sites) make a tool for querying data and services at the same time, with the same language, really necessary.

Integration of data and services can be tackled from two different perspectives. In the most common perspective, access to data is guaranteed through Service Oriented Architectures (SOA), and Web services are exploited to provide information integration platforms \([6,8]\). This approach is also known as Data-as-a-Services (DaaS). In a second perspective, the goal is to provide a global view

\* Extended abstract of the paper[3]
of the data sources managed in a peer and on eServices available in the peer or even on the Web, in order to support the access to the two complementary kind of resources at a same time. This approach is also known as Service-as-a-Data (SaaD). This paper addresses the SaaD approach, which is very different from the first one, and, to the best of our knowledge, completely new in the literature.

To make the research statement more precise, we take on the following assumptions. We assume to have a mediator-based data integration system which provides a global virtual view of data sources and is able to query data based on SQL-like queries expressed in the global virtual view terminology. In the context of a semantic peer environment, we refer to this global virtual view as the Semantic Peer Data Ontology (SPDO). We also assume to have a set of semantically annotated service descriptions; these descriptions may refer to services developed within the peer, but they can be also descriptions retrieved on the Web about services provided by other peers and addressing many different domains. Ontologies used in the service descriptions can be developed outside the peer and are not known in advance in the integration process.

The semantic-based approach developed within the NeP4B project (Networked Peers for Business)\(^3\) consists in performing data and service querying according to the following principle: given a SQL-like query expressed in the terminology of the SPDO, retrieve all the services that can be considered “related” to the query on the data sources.

The approach is based on the integration of a mediator-based data integration system, namely the MOMIS system (Mediator envirOnment for Multiple Information Sources)\(^4\) [4, 2], and of a service retrieval engine based on Information Retrieval (IR) techniques performing semantic indexing of service descriptions and keyword-based semantic search. The approach is based on the construction of a Peer Virtual View (PPV), representing the service and data ontologies and the mappings between them, and on a query rewriting process returning data satisfying the query and a set of related services. In particular, in this paper we focus on the query process, that, transforms a SQL-like query on the data sources into a keyword-based query for service retrieval, thus allowing querying data and services in a unified way.

The outline of the paper is the following: Section 2 introduces a motivating scenario that will be adopted as running example. Section 3 describes how data and eServices are represented, Section 4 provides the description of the technique for retrieving data and related services. Finally, some conclusions and future work are sketched in Section 5.

---

\(^3\) [Link: http://www.dbgroup.unimo.it/nep4b]

\(^4\) Further publications about MOMIS are available at [Link: http://www.dbgroup.unimo.it/Momis]
2 Motivating scenario and Running Example

Let us introduce as an example three information systems providing information about Italian locations that may be integrated to create a larger information source available for touristic purposes:

- **BookAtMe** provides information about more than 30,000 hotels located in more than 8,000 destinations. For each hotel, information is provided about facilities, prices, policies, multimedia contents ... Some services are also available for checking the availability of a room and booking it.

- **Touring** provides information about Italian hotels, restaurants and cities. By means of three different forms, it is possible to find the available hotels, restaurants (described with some specific features) and main monuments for each city.

- **TicketOne** provides information about Italian cultural events. For each event, a description including place, price and details is provided. The information system offers services to check the ticket availability of a particular event and, also, to buy tickets.

A global view of the data sources provided by each of these information systems is created by means of a data integration system (see Figure 1). It is composed of four classes: accommodation, restaurant, event which are related to each other by means of the city class. Moreover, a set of WS descriptions has been made available starting from public repositories available on the Web; such services are related to different domains such as Economy, Communication, Education, Food, Medical, Travel and Weapon.

Now let us consider that a user ask the query Q1 below:

```
select A.Name, A.City, A.Country
from Accommodation as A
where A.City='Modena'
```

The problem we address in this paper is to retrieve, among the many services available possibly related to the many domains mentioned above, the ones that are possibly related to Q1, according to the semantics of the terms involved in the query.

3 Building the Global Data and Service View at Set-up Time

In our approach, data sources and services are grouped into semantic peers. Each semantic peer generates a Peer Virtual View (PVV), i.e. a unified representation of the data and the services held by the sources belonging to the peer. A PVV is made up of the following main components, shown in Figure 2:

- a **Semantic Peer Data Ontology (SPDO)** of the data, i.e. a common representation of all the data sources belonging to the peer;
Fig. 1. The tourist integrated schema

- a **Global Light Service Ontology (GLSO)** that provides, by means of a set of concepts and attributes, a global view of all the concepts and attributes viewed for the descriptions of the Web services available in the peer;
- a set of **mappings** which connects GLSO elements to SPDO elements.

Details about the building process are published in [3]. Here we observe that there is fundamental difference between the SPDO and the GLSO: the first ontology represents a global schema of data (data behave as instances of the concepts in the SPDO), the second one formally represents the vocabulary used for service semantic annotations (services are not instances of the concepts in the GLSO), i.e. services are related in different ways to the concepts in the GLSO.

The mappings are based on the identification of **clusters** of terms having the same meaning, according to the methodology described in [2]. Finally, Figure 2 represents two other data structures exploited in the query processing phase (see Section 4), which are built at set up time together with the GLSO: an Inverted Index File (IIF) and a Semantic Similarity Matrix (SSM) between terms of the GLSO.

### 4 Data and eService Retrieval

Let us introduce a query expressed in an SQL-like language:

```sql
select
  <select_attribute_list>
from
  <from_class_list>
where
  <condition>
```
The answer to this query is a data set from the data sources together with a set of services which are potentially useful, since they are related to the concepts appearing in the query and then to the retrieved data.

The query processing is thus divided into two steps, that are simultaneously executed:

- a data set from the data sources is obtained with a query processing on an integrated view
- a set of services related to the query is obtained by exploiting the mapping between SPDO and GLSOs and the concept of relevant service mapping

Data results are obtained by exploiting the MOMIS Query Manager (see [1] for a complete description) which rewrites the global query as an equivalent set of queries expressed on the local schemata (local queries); this query translation is carried out by considering the mapping between the SPDO and the local schemata. Since MOMIS follows a GA V approach, the query translation is thus performed by means of query unfolding. Results from the local sources are then merged exploiting the reconciliation techniques. As the query processing on an integrated view is already well described in literature, in the following we focus our attention on the queries for services.

Services are retrieved by the XIRE (eXtended Information Retrieval Engine) component, which is a service search engine based on the vector space model [5], and implemented with the open source libraries Lucene [7]; in particular, the approach takes as input a vector where each term has a relevance weight associated. In order to provide XIRE with this input, the query processing phases represented in Figure 3 need to be defined. The process starts extracting the
terms of the SPDO appearing in the SQL-like query; then, the query manager checks which elements of the GLSO are mapped on these terms of the SPDO; these keywords are expanded w.r.t. the SSM implemented at set-up time, and a query vector of weighted terms is provided to XIRE which will retrieve the related services as described in details in Section 4.1.

![Diagram](image)

**Fig. 3.** The query processing steps and their application to the reference example

### 4.1 eService Retrieval

**Keywords extraction.** Given a query in an SQL-like notation expressed in the SPDO terminology, the set of keywords $K^{SPDO}$ extracted consists of: all the classes given in the “FROM” clause, all the attributes and the values used in the “SELECT” and “WHERE” clauses and all their ranges defined by ontology classes. As an example, the set of keywords extracted from the query Q1 introduced in Section 2, consists of the set $K^{SPDO}\#1$ represented in Figure 3.

**Keywords rewriting and expansion.** The set of keywords $K^{SPDO}$ extracted from the query inserted by users are rewritten in a set of keywords $K^{GLSO}$ exploiting the mappings between the SPDO and the GLSO. Let $Sig^O$ be the signature of an ontology $O$, e.g. the set of all the concept, predicate and individual names in $O$; let us define a data to service ontology mapping function $\mu = Sig^{SPDO} \rightarrow P(Sig^{GLSO})$. The function, given a term $s \in SPDO$ returns a set of terms $T \subseteq Sig^{GLSO}$ iff every $t \in T$ is in the same cluster of $s$. Given a set of keyword $K^{SPDO} = \{k_0, ..., k_m\}$, each keyword $k_i$ with $0 \leq i \leq m$ is replaced by the set of keywords returned by $\mu (k_i)$. Assuming the PVV mappings (see Section 3), and assuming $\mu (Modena) = Modena$, and $\mu (City) = City$, the set
of keywords obtained in the reference example is the set $K^{GLSO}_{#1}$ represented in Figure 3.

Semantic similarity between GLSO terms defined in the SSM is exploited to expand the $K^{GLSO}$ set into a weighted terms vector $q = \langle (k_1, w_1), ..., (k_n, w_n) \rangle$, where for $1 \leq i \leq n$, $k_i \in Sig^{GLSO}$, and $w_i$ are weights that represent the relevance of every terms w.r.t. the specified keywords. The vector $q$ is obtained associating each keyword with a weight equal to 1, and adding a set of terms that are similar to the given keywords up to a given threshold weighted according to their similarity w.r.t. the given keywords.

More formally, let $simset_s(t) \subseteq Sig^{GLSO}$ be the set of terms of the GLSO such that their similarity w.r.t. $t$ is greater than a given threshold $s$. Given a set of keywords $K^{GLSO} = \{k_0, ..., k_m\}$, the vector $q$ is obtained as follows: all the keywords $k_i \in K^{GLSO}$, with $1 \leq i \leq m$, are inserted in $q$ and are associated with a weight $w_i = 1$; for every $k_i \in K^{GLSO}$ the set $simset_s(k_i)$ is inserted in $q$, and each elements $e \in simset_s(k_i)$ is associated with a weight $w_e = sim(k_i, e)$, where $sim$ is defined according to the SMM; duplicate terms in $q$ are discarded, keeping the terms associated with a greater weight.

For example, let us consider the set of keywords $K^{GLSO}_{#1}$ given in the reference example. Assume to set the similarity threshold $s = 0.3$; $simset_{0.3}(City) \subseteq \{Municipal_Unit, Capital_City\}$, $sim(City, Municipal_Unit) = 0.5$ (City is subclass of Municipal_Unit in the Travel ontology) and $sim(City, Capital_City) = 0.3$ (Capital_City is subclass of City); a piece of the resulting weighted term query vector $q_{#1}$, including Municipal_Unit, Capital_City and LuxuryHotel (added in an analogous way based on the ontology including Hotel), is represented in Figure 3.

**eServices retrieval.** Query evaluation is based on the vector space model[5]; by this model both documents (that is Web Service descriptions) and queries (extracted queries) are represented as a vector in a $n$-dimensional space (where $n$ is the total number of index terms extracted from the document collections). Each vector represents a document, and it will have weights different from zero for those keywords which are indexes for that description. The value of such weight is computed according to the weights of the six sections of the semantic description of the service in which the keyword appears. We assume that the implicit constraint specified in a user query, when selecting a query term (a single keyword) is that it must appear in at least one section of a service description in order to retrieve that service. Based on the above assumptions the weight which at query evaluation time is associated with a keyword and a service description is equal to the maximum of the weights of the service sections in which the keyword appears.

Relevance weights introduced in the previous section are used to modify the weights in the list resulting from keyword evaluation process described above. In particular, we consider a relevant weight, as a "fine-tuning" value of the keyword weights in the modified posting list. The modification is performed by

---

5 Remind that all the GLSO terms are URIs, although parts of the URI specification are omitted for sake of clarity
a product between the keyword weight and the relevant weight. The process returns an ordered list of services (results for Q1 include the service described in the reference example).

5 Conclusion and Future Work

In this paper we introduced a technique for publishing and retrieving a unified view of data and services. Such unified view may be exploited for improving the user knowledge of a set of sources and for retrieving a list of web services related to a data set. The approach is semi-automatic, and works jointly with the tools which are typically provided for searching for data and services separately.

Future work will be addressed on evaluating the effectiveness of the approach in the real cases provided within the NeP4B project, and against the OWLS-TC benchmark.

6 Acknowledgments

The work presented in this paper has been partially supported by the European IST project n. 27347 SEEMP - Single European Employment Market-Place and the Italian FIRB project RBNE05XYPW NeP4B - Networked Peers for Business.

References

Combining Semantic and Multimedia Query Routing Techniques for Unified Data Retrieval in a PDMS *

Claudio Gennaro¹, Federica Mandreoli²,⁴, Riccardo Martoglia², Matteo Mordacchini³, Wilma Penzo³,⁴, and Simona Sassatelli²

¹ ISTI - CNR, Pisa, Italy
² DII - University of Modena e Reggio Emilia, Italy
³ DEIS - University of Bologna, Italy
⁴ IEIIT - BO/CNR, Bologna, Italy

Abstract. The NeP4B project aims at the development of an advanced technological infrastructure for data sharing in a network of business partners. In this paper we leverage our distinct experiences on semantic and multimedia query routing, and propose an innovative mechanism for an effective and efficient unified data retrieval of both semantic and multimedia data in the context of the NeP4B project.

1 Introduction and Related Work

Information and communication technologies (ICTs) over the Web have become a strategic asset in the global economic context. The Web fosters the vision of an Internet-based global marketplace where automatic cooperation and competition are allowed and enhanced. This is the stimulating scenario of the ongoing Italian Council co-funded NeP4B (Networked Peers for Business) Project whose aim is to develop an advanced technological infrastructure for small and medium enterprises (SMEs) to allow them to search for partners, exchange data and negotiate without limitations and constraints.

According to the recent proposal of Peer Data Management Systems (PDMSs) [5], the project infrastructure is based on independent and interoperable semantic peers who behave as nodes of a virtual peer-to-peer (P2P) network for data and service sharing. In this context, a semantic peer can be a single SME, as well as a mediator representing groups of companies, and consists of a set of data sources (e.g. data repositories, catalogues) placed at the P2P network’s disposal through an OWL ontology. Data sources include multimedia objects, such as the descriptions/presentations of the products/services extracted from the companies’

* This work has been partially supported by the Italian National Research Council in the context of the NeP4B Project and of the research initiative “Ricerca a tema libero” and by the IST FP7 European Project S-Cube (Grant Agreement no. 215483)
Web sites. This information is represented by means of appropriate multimedia attributes in the peers’ ontologies (e.g. image in Peer1’s ontology of Figure 1) that are exploited in the searching process by using a SPARQL-like language properly extended to support similarity predicates. As an example, let us consider the query in Figure 1 which asks Peer1 for “Companies that sell Italian tiles similar to the

image "myimage.jpg"” that is used for query formulation and semantic mappings are used to reformulate the

query over its immediate neighbors, then over their immediate neighbors, and so on [5, 7]. For instance, in Figure 1 the concept origin translates into country when the query is forwarded to Peer2.

In such a distributed scenario, where query answers can come from any peer in the network which is connected through a semantic path of mappings [5], a key challenge is query routing, i.e. the capability of selecting a small subset of relevant peers to forward a query to. Flooding-based techniques are indeed not adequate for both efficiency and effectiveness reasons: not only they overload the network (forwarded messages and computational effort required to solve queries), but also overwhelm users with a large number of results, mostly irrelevant.

Query routing in P2P systems has attracted much research interest in the last few years, with the aim of effectively and efficiently querying both multimedia [1] and semantic data [9]. As far as we know, no proposal exists which operates on both these kinds of data in an integrated approach.

As part of the NeP4B project, we leverage our distinct experiences on semantic

[6, 8] and multimedia [3, 4] query routing and propose to combine the approaches...
we presented in past works in order to design an innovative mechanism for a unified data retrieval in such a context. Two main aspects characterize our scenario. The former one is due to the heterogeneity of the peers’ ontologies which may lead to semantic approximations during query reformulation. In this context, we pursue *effectiveness* by selecting, for each query, the peers which are semantically best suited for answering it, i.e. whose answers best fit the query conditions. The latter aspect is related to the execution of multimedia predicates, which is inherently costly (they typically require the application of complex functions to evaluate the similarity of multimedia features). In this setting, we also pursue *efficiency* by limiting the forwarding of a query towards the network’s zones where potentially matching instances are more likely to be found, while pruning the others. In this way, we give the user a higher chance of receiving first the answers which better satisfy the query conditions.

To this end, we introduce a query processing model which satisfies the interoperability demands highlighted in the NeP4B project. The proposed model does not compel to a fixed semantics but rather it is founded on a *fuzzy* settlement which proved to be sound for a formal characterization of the approximations originated in the NeP4B network for both semantic [6] and multimedia data [10]. The model leverages the query answering semantics (Sect. 2) to define a query routing approach which operates on both semantic and multimedia data in an integrated way (Sect. 3), and to show how routing strategies (Sect. 4) influence the order of the returned answers. This allows different query processing approaches to be implemented, based on the specific consortium needs and policies. The validity of the proposal is proved by the initial experiments we conducted on different settings (Sect. 5).

2 Query Answering Semantics

The main aim of this section is to propose a semantics for answering queries in the NeP4B network where two kinds of approximations may occur: the one given by the evaluation of multimedia predicates, and the other one due to the reformulation of queries along paths of mappings.

We denote with $\mathcal{P}$ the set of peers in the network. Each peer $p_i \in \mathcal{P}$ stores local data, modelled upon a local OWL ontology $O_i$. Peers are pairwise connected in a semantic network through semantic mappings between peers’ ontologies. For our query routing purposes, we abstract from the specific format that semantic mappings may have. For this reason, we consider a simplified scenario where each peer ontology $O_i$ is represented through a set of ontology classes $\{C_{i1}, \ldots, C_{im}\}$ and semantic mappings are assumed to be directional, pairwise and one-to-one. The approach we propose can be straightforwardly applied to more complex mappings relying on query expressions as proposed, for instance, in [5]. Mappings are extended with scores quantifying the strength of the relationship between the involved concepts. Their fuzzy interpretation is given in the following.

\footnote{Note that in OWL properties are specified through classes.}
Definition 1 (Semantic Mapping). A semantic mapping from a source schema $O_1$ to a target schema $O_j$, not necessarily distinct, is a fuzzy relation $M(O_1, O_j) \subseteq O_1 \times O_j$ where each instance $(C, C')$ has a membership grade denoted as $\mu(C, C') \in [0, 1]$. This fuzzy relation satisfies the following properties: 1) it is a 0-function, i.e., for each $C \in O_1$, it exists exactly one $C'$ in $O_j$ such that $\mu(C, C') \geq 0$; 2) it is reflexive, i.e., given $O_i = O_j$, for each $C \in O_i$, $\mu(C, C) = 1$.

For instance, Peer1 of Fig. 1 maintains two mapping relations: the mappings towards Peer2 ($M(O_1, O_2)$) and Peer3 ($M(O_1, O_3)$). For instance, $M(O_1, O_2)$ associates Peer1’s concept origin to Peer2's concept country with a score of 0.73, thus expressing that a semantic approximation is detected between the two concepts (for instance, the country might be only a part of an origin).

A query is posed on the ontology of the queried peer. Query conditions are expressed using predicates that can be combined in logical formulas through logical connectives, according to the syntax:

$$f ::= \langle \text{triple\_pattern} \rangle \langle \text{filter\_pattern} \rangle$$

$$\langle \text{triple\_pattern} \rangle ::= \text{triple} \mid \langle \text{triple\_pattern} \rangle \land \langle \text{triple\_pattern} \rangle$$

$$\langle \text{filter\_pattern} \rangle ::= \varphi \mid \langle \text{filter\_pattern} \rangle \land \langle \text{filter\_pattern} \rangle \mid \langle \text{filter\_pattern} \rangle \lor \langle \text{filter\_pattern} \rangle \mid ((\langle \text{filter\_pattern} \rangle))$$

where triple is an RDF triple and a filter $\varphi$ is a predicate where relational ($=$, $<$, $>$, $<=$, $!=$) and similarity ($\sim_i$) operators operate on RDF terms and values. In particular, note that $\sim_i$ refers to multimedia content and translates the \texttt{LIKE} operator where $t$ is the specified similarity threshold.

Each peer receiving a query first retrieves the answers from its own local data then it reformulates the query towards its own neighborhood.

The evaluation of a given query formula $f$ on a local data instance $i$ is given by a score $s(f, i)$ in $[0, 1]$ which says how much $i$ satisfies $f$. The value of $s(f, i)$ depends on the evaluation on $i$ of the filters $\varphi_1, \ldots, \varphi_n$ that compose the filter-pattern of $f$, according to a scoring function $sfun_\varphi$, that is: $s(f(\varphi_1, \ldots, \varphi_n), i) = sfun_\varphi(s(\varphi_1, i), \ldots, s(\varphi_n, i))$. Note that filters are predicates of two types: relational and similarity predicates. A relational predicate is a predicate which evaluates to either 1 (true) or to 0 (false). The evaluation of a similarity predicate $\varphi$ follows instead a non-Boolean semantics and returns a score $s(\varphi, i)$ in $[0, 1]$ which denotes the grade of approximation of the data instance $i$ with respect to $\varphi$. It is set to 0 when the similarity of $\varphi$ w.r.t. the predicate value is smaller than the specified threshold $t$, and to the grade of approximation measured, otherwise. The scoring function $sfun_\varphi$ combines the use of a t-norm ($\land_s$) for scoring conjunctions of filter evaluations, and the use of a t-conorm ($\lor_s$) for scoring disjunctions. A t-norm (resp., t-conorm) is a binary function on the unit interval that satisfies the boundary condition (i.e. $s_\land(s, 1) = s$, and resp., $s_\lor(s, 0) = s$), as well as the monotonicity, the commutativity, and the associativity properties. The use of t-norms and t-conorms generalizes the query evaluation model with respect to the use of specific

\footnote{Examples of t-norms are the \texttt{min} and the \texttt{algebraic product} operators, whereas examples of t-conorms are the \texttt{max} and the \texttt{algebraic sum} operators.}
functions. Therefore, for a given peer $p$, the query answers retrieved from the evaluation of $f$ on its own local data is given by $\text{Ans}(f, p) = \{(i, s(f, i)) \mid s(f, i) > 0\}$, i.e., it is the set of local data instances which satisfy $f$, possibly with a certain grade of approximation.

Due to the heterogeneity of schemas, any reformulation a peer $p_i$ performs on a given query formula $f$ towards one of its neighbors, say $p_j$, gives rise to a semantic approximation which depends on the strength of the relationship between each concept in $f$ and the corresponding concept in $O_j$. Such an approximation is quantified by a scoring function $s_{\text{fun}}$, which combines the $p_j$’s mapping scores on $f$’s concepts: $s(f, p_j) = s_{\text{fun}}(\mu(C_1, C'_1), \ldots, \mu(C_n, C'_n))$ where $C_1, \ldots, C_n$ are the concepts in $O_i$ involved in the query formula, and $C'_1, \ldots, C'_n$ are the corresponding concepts in $O_j$ according to $M(O_i, O_j)$. $s_{\text{fun}}$ is a t-norm as all the involved concepts are specified in the triple_pattern of $f$ and triples can only be combined through conjunctions.

Starting from the queried peer, the system can access data on any peer in the network which is connected through a semantic path of mappings. When the query is forward through a semantic path, it undergoes a multi-step reformulation which involves a chain of semantic approximations. The semantic approximation given by a semantic path $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_m$ (in the following denoted as $P_{p_1 \ldots p_m}$), where the submitted query formula $f_1$ undergoes a chain of reformulations $f_1 \rightarrow f_2 \rightarrow \ldots \rightarrow f_m$, can be obtained by composing the semantic approximation scores associated with all the reformulation steps: $s(f_1, P_{p_1 \ldots p_m}) = s_{\text{fun}}(s(f_1, p_2, s(f_2, p_3), \ldots, s(f_{m-1}, p_m))$, where $s_{\text{fun}}$ is a t-norm which composes the scores of query reformulations along a semantic path of mappings.

Summing up, given a query formula $f$ submitted to a peer $p$, the set of accessed peers $\mathcal{P}' = \{p_1, \ldots, p_m\}$, and the path $P_{p \ldots p_i}$ used to reformulate $f$ over each peer $p_i$ in $\mathcal{P}'$, the semantics of answering $f$ over $\{p\} \cup \mathcal{P}'$ is the union of the query answers collected in each accessed peer: $\text{Ans}(f, p) \cup \text{Ans}(f, P_{p \ldots p_1}) \cup \ldots \cup \text{Ans}(f, P_{p \ldots p_m})$ where each answer $\text{Ans}(f, P_{p \ldots p_i})$ contains the set of the results collected in the accessed peer $p_i$ together with the semantic approximation given by the path $P_{p \ldots p_i}$: $\text{Ans}(f, P_{p \ldots p_i}) = (\text{Ans}(f, p_i), s(f, P_{p \ldots p_i}))$. As observed before, as far as the starting queried peer is involved, no semantic approximation occurs as no reformulation is required (i.e. $s(f, p) = 1$).

3 Query Routing

In this section we define a query routing approach which operates on both semantic and multimedia data in an integrated way by first introducing the two approaches separately and then by meaningfully combining them.

3.1 Semantic Query Routing

Whenever a peer $p_i$ selects one of its neighbor, say $p_j$, for query forwarding, the query moves from $p_i$ to the subnetwork rooted at $p_j$ and it might follow any of

---

1 Note that $\mathcal{P}'$ not necessarily covers the whole network (i.e., $\mathcal{P}' \subseteq \mathcal{P}$).
the semantic paths originating at \( p_j \). Our main aim in this context is to introduce a ranking approach for query routing which promotes the \( p_i \)'s neighbors whose subnetworks are the most semantically related to the query.

In order to model the semantic approximation of \( p_j \)'s subnetwork w.r.t. \( p_i \)'s schema, the semantic approximations given by the paths in \( p_j \)'s subnetwork are aggregated into a measure reflecting the relevance of the subnetwork as a whole. To this end, the notion of semantic mapping is generalized as follows. Let \( p_\triangle j \) denote the set of peers in the subnetwork rooted at \( p_j \), \( O_\triangle j \) the set of schemas \( \{ O_{p_j k} | p_j k \in p_\triangle j \} \), and \( P_{p_i \ldots p_\triangle j} \) the set of paths from \( p_i \) to any peer in \( p_\triangle j \). The generalized mapping relates each concept \( C \) in \( O_i \) to a set of concepts \( C_\triangle \) in \( O_\triangle j \) taken from the mappings in \( P_{p_i \ldots p_\triangle j} \), according to an aggregated score which expresses the semantic similarity between \( C \) and \( C_\triangle \).

**Definition 2 (Generalized Semantic Mapping).** Let \( p_i \) and \( p_j \) be two peers, not necessarily distinct, and \( g \) an aggregation function. A generalized semantic mapping between \( p_i \) and \( p_j \) is a fuzzy relation \( M(O_i, O_\triangle j) \) where each instance \( (C, C_\triangle) \) is such that: 1) \( C_\triangle \) is the set of concepts \( \{ C_1, \ldots, C_h \} \) associated with \( C \) in \( P_{p_i \ldots p_\triangle j} \), and 2) \( \mu(C, C_\triangle) = g(\mu(C, C_1), \ldots, \mu(C, C_h)) \).

The aggregation function \( g \) is a continuous function on fuzzy sets which satisfies the monotonicity, the boundary condition, the symmetry and the idempotence properties. Several choices are possible for \( g \) satisfying the above properties, for instance functions such as the min, the max, any generalized mean (e.g. harmonic and arithmetic means), or any ordered weighted averaging (OWA) function (e.g. a weighted sum) [6].

Therefore, each peer \( p \) maintains a matrix named *Semantic Routing Index* (SRI), which contains the membership grades given by the generalized semantic mappings between itself and each of its neighbors \( \text{Nb}(p) \) and which is used as a routing index. A portion of Peer1’s SRI of the reference example is shown below:

<table>
<thead>
<tr>
<th>SRI_Peer1</th>
<th>Tile</th>
<th>origin</th>
<th>company</th>
<th>price</th>
<th>material</th>
<th>size</th>
<th>image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Peer2</td>
<td>0.85</td>
<td>0.70</td>
<td>0.83</td>
<td>0.95</td>
<td>0.83</td>
<td>0.92</td>
<td>1.0</td>
</tr>
<tr>
<td>Peer3</td>
<td>0.65</td>
<td>0.85</td>
<td>0.75</td>
<td>0.86</td>
<td>0.95</td>
<td>0.74</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Besides the first row, which represents the knowledge of Peer1’s local schema, it contains two entries, one for the upward subnetwork rooted at Peer2, and one for the downward one rooted at Peer3. For instance, from the stored scores, it follows that Peer3’s subnetwork better approximates the concept *origin* (score 0.85) than Peer2’s one (score 0.70). More details about the management of SRIs can be found in [6].

Thus, when a peer \( p \) receives a query formula \( f \), it exploits its SRI scores to determine a ranking \( R^p_{\text{sem}}(f) \) for its neighborhood, so as to identify the directions which best approximate \( f \). More precisely, for each neighbor \( p_i \) an overall score is computed by combining, by means of the scoring function \( sfun \), the scores the SRI row \( SRI[p_i] \) associates with the concepts \( C_1, \ldots, C_n \) in \( f \): \( R^p_{\text{sem}}(f)[p_i] = sfun(\mu(C_1, C_\triangle 1), \ldots, \mu(C_n, C_\triangle n)) \). Intuitively, the higher is the overall score, the
more likely the peer’s subnetwork will provide semantically relevant results to the query.

3.2 Multimedia Query Routing

The execution of multimedia predicates is inherently costly, both from a CPU and from an I/O point of view. Adopting a broadcast-based approach to solve multimedia predicates in the network could thus imply wasting precious resources. Instead, we introduce a routing mechanism for efficient similarity search in PDMS. The approach can be exploited to solve most multimedia predicates as it has been conceived for metric spaces. In a metric space, the similarity between two objects is evaluated according to their pairwise distance: the lower their distance, the higher their similarity. The key idea is to build a distributed index that provides a concise but yet sufficiently detailed description of the multimedia resources available in a given network area. This information is then used at query time to forward a query containing multimedia predicates only towards those directions with the highest number of potential matchings.

To this end, for a given peer \( p \), for each multimedia object \( X \) in \( p \)’s local dataset (e.g., an image), we extract one or more features. We represent each feature \( F_i \) of an object \( X \) as an element of a metric space and we denote it as \( X_{Fi} \). For each feature \( F_i \), each peer builds different feature indices, in order to allow also for multi-feature queries. Each index exploits \( m \) reference objects \( R_k^{Fi} \), with \( k = 1 \ldots m \), i.e. objects that are used to determine the position of other objects in the metric space. For ease of presentation, let us consider the case of a single feature. For simplicity, we also drop the symbol \( F_i \) from the following formulations, and when there is no possibility of confusion we use the same symbol \( X \) for indicating both the multimedia object and its associated metric feature. Formally, if we consider distances from a reference object \( R_k \) in the interval \( [a, b] \), we select \( h + 1 \) division points \( a = a_0 < a_1 < \ldots < a_h = b \) such that \( [a, b] \) is partitioned into \( h \) disjoint intervals \( [a_i, a_{i+1}) \), \( i = 0, 1, \ldots, h - 1 \). Given a peer \( p \), for each reference object \( R_k \) we build the histogram \( FeatureIdx(p)_{R_k} \), which measures the number of objects \( X \) for which \( d(X, R_k) \in [a_i, a_{i+1}) \) \( \forall i \). This index gives us a concise description of all the data owned by a peer and it represents how the peer’s objects are distributed with respect to the reference object \( R_k \).

Each peer \( p \) also maintains, for its neighborhood \( Nb(p) \), a set of Multimedia Routing Indices (MRIs), one for each reference object \( R_k \). Any MRI row \( MRI(p, p^i)_{R_k} \), represents the aggregated description of the resources available in the subnetwork \( p^i \) rooted at \( p_i \in Nb(p) \) and is built by summing up the index for \( R_k \) of the peers in the subnetwork.

\[
MRI(p, p^i)_{R_k} \equiv FeatureIdx(p_i)_{R_k} + \sum_{p_j \in Nb(p_i) \setminus p} MRI(p_i, p^j)_{R_k} \quad (1)
\]

As an example, consider the MRI of Peer1 represented in the following table, in which we give the number of the total objects in each subnetwork, for each distance interval of the reference object \( R_k \).
For our query routing purposes, we focus on similarity-based \textit{Range Queries} over metric objects, defined as follows: given the multimedia object $Q$ and the range $r$ specified as argument of any \texttt{LIKE} predicate, the query has to retrieve \{ $X \mid d(X, Q) \leq r$ \}. For a given range query, the values $d(Q, R_k) - r$ and $d(Q, R_k) + r$ are computed, for each $k = 1, \ldots, m$. The vector representation of each query index $\text{QueryIdx}(Q)$ is then built by setting to 1 all the entries that correspond to intervals that are covered (even partially) by the requested range. All the other entries are set to 0. This index has the same form of the histogram $\text{FeatureIdx}(p)$ but only contains values in \{0, 1\}.

When a peer $p$ receives a query formula $f$ containing a \texttt{LIKE} predicate, instead of flooding the network by forwarding $f$ to all its neighbors, $p$ matches the indices of $Q$ with the corresponding routing indices of its neighborhood. The matching phase outputs a score that suggests the degree of relevance of the query with respect to each of the possible forwarding directions. More precisely, $p$ determines a ranking $R_{\text{mm}}^p(f)$ for its neighborhood, by taking the minimum of the products (element by element) of the indices $\text{QueryIdx}(Q)$ and $\text{MRI}(p, p_i)$ for each neighbor $p_i \in \text{Nb}(p) = \{ p_1, \ldots, p_n \}$ and each reference object $R_k$, and then evaluating the following ratio:

$$R_{\text{mm}}^p(f)[p_i] = \frac{\min_k [\text{QueryIdx}(Q) \cdot \text{MRI}(p, p_i)]}{\sum_{j=1}^n \min_k [\text{QueryIdx}(Q) \cdot \text{MRI}(p, p_j)]}$$  \hspace{1cm} (2)$$

$R_{\text{mm}}^p(f)[p_i]$ gives an intuition of the percentage of potential matching objects underneath the subnetwork rooted at $p_i$ with respect to the total objects retrievable through all the peers in $\text{Nb}(p)$.

### 3.3 Combined Query Routing

Whenever a peer $p$ receives a query, both the semantic and the multimedia routing approaches associate each $p$’s neighbor a score quantifying the semantic relevance and the percentage of potential matching objects in its subnetwork, respectively. This allows $p$ to rank its own neighbors w.r.t. their ability to answer a given query effectively, i.e. minimizing the information loss due to its reformulation along semantic mappings, and efficiently, i.e. minimizing the network load due to the exploration of useless subnetworks.

Thus, since both the semantic and multimedia scores induce a total order, they can be combined by means of a proper aggregation function in order to obtain a global ranking. More precisely, given the two distinct rankings $R_{\text{sem}}^p(f)$ and $R_{\text{mm}}^p(f)$ computed for the query formula $f$ on peer $p$ we need an aggregation function $\odot$ which, when applied to $R_{\text{sem}}^p(f)$ and $R_{\text{mm}}^p(f)$, provide a $R_{\text{comb}}^p(f)$ reflecting

---

\[ \text{For ease of presentation, in the following we assume that each query formula } f \text{ contains at most one \texttt{LIKE} predicate.} \]
the overall goodness of the available subnetworks: 

\[ R_{\text{comb}}^p(f) = \alpha \cdot R_{\text{sem}}^p(f) \oplus \beta \cdot R_{\text{mm}}^p(f), \]

where \( \alpha \) and \( \beta \) can be set in order to give more relevance to either the semantic or multimedia aspect.

In [2] it is stated that optimal aggregation algorithms can work only with monotone aggregation function. Typical examples of these functions are the min and mean functions (or the sum, in the case we are not interested in having a combined grade in the interval \([0, 1]\)). As an example of how the aggregation process works, let us go back to the sample query in Figure 1 and suppose Peer1 obtains the scores in the following table.

<table>
<thead>
<tr>
<th>Peer1</th>
<th>Peer2</th>
<th>Peer3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRI</td>
<td>0.70</td>
<td>0.65</td>
</tr>
<tr>
<td>MRI</td>
<td>0.53</td>
<td>0.76</td>
</tr>
</tbody>
</table>

The rankings computed through SRI and MRI are Peer2-Peer3 and Peer3-Peer2, respectively. If we use the standard fuzzy conjunction \( \text{min} \), we compute the following final ranking: Peer3-Peer2. As a result, the most promising subnetwork will be the one rooted at neighbor Peer3.

The obtained ranking reflects the foreseen subnetworks ability in solving the received query both at schema (SRI-based information) and at multimedia (MRI-based information) levels and can thus be properly tailored in order to implement clever routing strategies. This is the subject of the following section.

4 Routing Strategies

Starting from the queried peer, the objective of any query processing mechanism is to answer requests by navigating the network until a stopping condition is reached. A query is posed on the schema of the queried peer and is represented as a tuple \( q = (id, f, \tau, nres) \) where: \( id \) is a unique identifier for the query; \( f \) is the query formula; \( nres \) is the stopping condition specifying the desired number of results as argument of the \( \text{LIMIT} \) clause; and \( \tau \) is an optional relevance threshold. Then, query answers can come from any peer in the PDMS that is connected through a semantic path of mappings.

In this context, the adoption of adequate query routing strategies is a fundamental issue. Indeed, Sec. 2 shows that any peer satisfying the query conditions may add new answers and different paths to the same peer may yield different answers. More precisely, at each reformulation step, a new peer \( p_{\text{Next}} \in \mathcal{P} \) is selected, among the unvisited ones, for query forwarding. The adopted routing strategy is responsible for choosing \( p_{\text{Next}} \), thus determining the set of visited peers \( \mathcal{P}' \) and inducing an order \( \psi \) in \( \mathcal{P}': [p_{\psi(1)}, \ldots, p_{\psi(m)}] \). Further, the visiting order determines the path \( P_{p \rightarrow p_{\psi}} \) which is exploited to reach each peer \( p_{\psi} \in \mathcal{P}' \) and, consequently, the set of returned local answers and the semantic approximation accumulated in reaching \( p_{\psi} \), i.e. \( \text{Ans}(f, P_{p \rightarrow p_{\psi}}) \).

Then, in order to increase the chance that users receive first the answers which better satisfy the query conditions, several routing policies can be adopted. More precisely, as we explained in Sec. 3.3, when a peer \( p \) receives a query \( q \) it exploits
its indices information to compute a ranking $R_{comb}^p(f)$ on its neighbors expressing
the goodness of their subnetworks w.r.t. the query formula $f$.

Afterwards, different query forwarding criteria relying on such ranked list are
possible, designed around different performance priorities. In [8] two main fami-
lies of navigation policies are introduced: The Depth First (DF) query execution
model, which pursues efficiency as its main objective, and the Global (G) model,
which is designed for effectiveness. Both approaches are devised in a distributed
manner through a protocol of message exchange, thus trying to minimize the
information spanning over the network.

In particular, the DF model is efficiency-oriented since its main goal is to limit
the query path. More precisely, with the aim of speeding up the retrieval of some
(but probably not the best) results, the DF model only performs a local choice
among the neighbors of the current peer, i.e. it exploits the only information
provided by $R_{comb}^p(f)$.

Differently from the DF one, in the G model each peer chooses the best peer
to forward the query to in a “global” way: It does not limit its choice among
the neighbors but it considers all the peers already “discovered” (i.e. for which a
navigation path leading to them has been found) during network exploration and
that have not been visited yet. More precisely, given the set $V$ of visited peers,
it exploits the information provided by $\bigcup_{v \in V} R_{comb}^p(f)$ in order to select, at each
forwarding step, the best unvisited peer. Obviously, going back to potential distant
peers has a cost in terms of efficiency, but always ensures the highest possible
effectiveness, since the most promising discovered peers are always selected.

According to our query answering semantics, $P'$ is thus defined as the ordered
set of visited peers $[p_{\psi(1)}, \ldots, p_{\psi(m)}]$ such that $|\text{Ans}(f, p) \cup \text{Ans}(f, p_{\psi(1)}) \cup \ldots \cup \text{Ans}(f, p_{\psi(m)})| \geq nres$ and $|\text{Ans}(f, p) \cup \text{Ans}(f, p_{\psi(1)}) \cup \ldots \cup \text{Ans}(f, p_{\psi(n-1)})| < nres$, where the ordering function $\psi$ is given by the adopted routing strategy.

As to the optional threshold $\tau$, when $\tau > 0$, those subnetworks whose relevance
(in terms of combined routing score) is smaller than $\tau$ are not considered for query
forwarding. The underlying reason is that a forwarding strategy which proceeds
by going deeper and deeper toward poorly relevant network areas (i.e. not very
semantically related to the query and containing few multimedia matching ob-
jects) can exhibit bad performances and, thus, it is better to start backtracking
towards other directions. The adoption of a threshold $\tau$ may thus positively in-
fluence the composition of $P'$, since “poorer” subnetworks are not considered for
query forwarding. On the other hand, a not-null threshold introduces a source of
incompleteness in the querying process, as the pruned subnetworks might contain
matching objects. Completeness can instead be guaranteed when $\tau = 0$, since
subnetworks with a 0 routing score can be safely pruned.

5 Experiments

In this section we present an initial set of experiments we performed in order to
evaluate our combined query routing approach. Notice that, since we are currently
in the initial phase of our testing, the considered scenarios are not particularly
complex; in the future we will enrich them with more complicated and larger ones.
For our experiments, we exploited our simulation environments for putting into action the SRI [6, 7] and MRI [3, 4] approaches. Through these environments we modeled scenarios corresponding to networks of semantic peers, each with its own schema, consisting of a small number of concepts, and a repository of multimedia objects. As to the multimedia contents, we use few hundreds of images taken from the Web and characterized by two MPEG-7 standard features: scalable color and edge histogram. We tested our techniques on different alternative network topologies, randomly generated with the BRITE tool (http://www.cs.bu.edu/brite/), whose mean size was in the order of few dozens of nodes. In order to evaluate the performance of our techniques we simulated the querying process by instantiating different queries on randomly selected peers and propagating them until their stopping condition on the number of retrieved results is reached: We evaluated the effectiveness improvement by measuring the semantic quality of the results (satisfaction) and, on the other hand, the efficiency improvement by measuring the number of hops performed by the queries. Satisfaction is a specifically introduced quantity which grows proportionally to the goodness of the results returned by each queried peer: Each contribution is computed by combining the semantic mapping scores of the traversed peers (see [6]). The search strategy employed is the depth first search (DF). In our experiments we compare our neighbor selection mechanism based on a combination of SRIs and MRIs (Comb) with the two mechanisms which only exploit the SRI (SRI) and MRI (MRI) values and with a baseline corresponding to a random strategy (Rand). The employed aggregation function is the mean. Notice that all the results we present are computed as a mean on several query executions.

Figure 2-a represents the trend of the obtained satisfaction when we gradually vary the stopping condition on the number of retrieved results. As we expected, the Rand and the MRI strategies show a similar poorly effective behavior since both select the subnetworks to explore without considering their semantic relevance. As we expected, they are thus outperformed by the SRI strategy which, on the contrary, is able to discriminate at each step the semantically best direction and, thus, increases the satisfaction in a substantial way. Nevertheless, the Comb routing reveals itself as the most effective one: it works by considering in an integrated way semantic and multimedia information and, consequently, tends to cover shorter paths which inherently have a lower approximation (and, thus, a higher satisfaction).

As to the efficiency evaluation, Figure 2-b represents the trend of the hops required for satisfying queries. Also this time, the Rand routing exhibits the worst behavior while the SRI one, which has no kind of knowledge on multimedia data, often comes closer to it. Though being poorly effective, the MRI strategy is instead the most efficient one, since, for each query, it selects the subnetworks with the higher number of (even not semantically good) multimedia matching objects. On the other hand, the lower efficiency of the Comb routing is motivated by the fact that it wastes more hops in searching semantically relevant objects.

Summing up, the Comb strategy represents the best alternative and proves to be able to increase the chance to retrieve first the answers which better satisfy the query conditions.
6 Concluding Remarks

We presented an innovative approach for processing queries effectively and efficiently in a distributed and heterogeneous environment, like the one outlined in the NeP4B project. As far as we know, this is the first research proposal specifically devised to enhance the processing of queries in a network of semantic peers which share both semantic and multimedia data.

References

The MOMIS-STASIS approach for Ontology-Based Data Integration

Domenico Beneventano  Mirko Orsini  Laura Po  Serena Sorrentino
Dipartimento di Ingegneria dell’Informazione, Università di Modena e Reggio Emilia, Italy
{firstname.lastname}@unimore.it

Abstract. Ontology based Data Integration involves the use of ontology(s) to effectively combine data and information from multiple heterogeneous sources [18]. Ontologies can be used in an integration task to describe the semantics of the information sources and to make the contents explicit. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts, i.e. for the definition of semantic mapping among concepts of the information sources. MOMIS is a Data Integration System which performs information extraction and integration from both structured and semi-structured data sources [7]. The goal of the STASIS project is to create a comprehensive application suite which allows enterprises to simplify the mapping process between data schemas based on semantics [1]. Moreover, in STASIS, a general framework to perform Ontology-driven Semantic Mapping has been proposed [4]. This paper describes the early effort to combine the MOMIS and the STASIS frameworks in order to obtain an effective approach for Ontology-Based Data Integration.

1 Introduction

Data integration is the problem of combining data residing at different autonomous sources, and providing the user with a unified view of these data. The problem of designing Data Integration Systems is important in current real world applications, and is characterized by a number of issues that are interesting from a theoretical point of view [12]. Integration System are usually characterized by a classical wrapper/mediator architecture [19] based on a Global Virtual Schema (Global Virtual View - GVV) and a set of data sources. The data sources contain the real data, while the GVV provides a reconciled, integrated, and virtual view of the underlying sources. Modeling the mappings among sources and the GVV is a crucial aspect. Two basic approaches for specifying the mapping in a Data Integration System have been proposed in the literature: Local-As-View (LAV), and Global-As-View (GAV), respectively [11, 17].

MOMIS (Mediator Environement for Multiple Information Sources) is a Data Integration System which performs information extraction and integration from both structured and semi-structured data sources by following the GAV approach [7, 6]. Information integration is performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus (defined by the framework) and descriptions of source schemas with a combination of clustering techniques and Description Logics. This integration
process gives rise to a virtual integrated view of the underlying sources for which mapping rules and integrity constraints are specified to handle heterogeneity.

Ontologies can be used in an integration task to describe the semantics of the information sources and to make the contents explicit [18]. With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts. In [18], three different approaches of how to employ the ontologies for the explicit description of the information source semantics are identified: single ontology approaches, multiple ontologies approaches and hybrid approaches. Single ontology approaches use one global ontology providing a shared vocabulary for the specification of the semantics: all data sources are related to one global ontology. In multiple ontology approaches, each information source is described by its own ontology and mappings between the ontologies are defined: these inter-ontology mappings identify semantically corresponding terms of different source ontologies, e.g. which terms are semantically equal or similar. In hybrid approaches similar to multiple ontology approaches the semantics of each source is described by its own ontology, but in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary which contains basic terms of a domain [18].

With respect to the above classification, the MOMIS Data Integration System uses a single ontology approach, where the lexical ontology WordNet [14] is used as a shared vocabulary for the specification of the semantics of data sources and for the identification and association of semantically corresponding information concepts. The main reason of this choice is that, by using a lexical ontology as WordNet (which it is characterized by a wide network of semantic relationships between concepts), the annotation of data sources elements can be performed in a semi-automatic way by using Word Sense Disambiguation techniques.

The STASIS IST project (www.stasis-project.net) is a Research and Development project sponsored under the EC 6th Framework programme. It aims to enable SMEs and enterprises to fully participate in the Economy, by offering semantic services and applications based on the open SEEM registry and repository network. The goal of the STASIS project is to create a comprehensive application suite which allows enterprises to simplify the mapping process between data schemas, by providing an easy to use GUI, allowing users to identify semantic elements in an easy way [2, 1].

Moreover, in the STASIS project, a general framework to perform Ontology-driven Semantic Mapping has been proposed, where the identification of mappings between concepts of different schemas is based on the schemas annotation with respect to ontologies [4].

In [5] this framework has been further elaborated and it has been applied to the context of products and services catalogues. In the STASIS project OWL is used as language to include in the framework generic external ontologies.

This paper describes an approach to combine the MOMIS and STASIS frameworks in order to obtain an effective approach for Ontology-Based Data Integration. The proposal is based on the extension of the MOMIS system by using the Ontology-driven Semantic Mapping framework developed in STASIS in order to address the following points:
1. enabling the MOMIS system to employ generic OWL ontologies, with respect to the limitation of using only the WordNet lexical ontology;
2. enabling the MOMIS system to exploit a multiple ontology approach with respect to the actual single ontology approach;
3. developing a new method to compute semantic mapping among source schemas in the MOMIS system.

The paper is organized as follows. Section 2, describes the proposed approach to use the Ontology-driven Semantic Mapping framework in the Global Schema generation process of MOMIS; in Section 3 future works are sketched out; finally, Section 4 is devoted to conclusions.

2 Ontology-Based Data Integration: the MOMIS-STASIS approach

This section describes our approach to use the Ontology-driven Semantic Mapping framework performed by STASIS for a different goal, i.e., during the Global Schema Generation process performed by the MOMIS system. Intuitively, with the Ontology-driven Semantic Mapping framework we may perform in the Data Integration System the annotation of data sources elements with respect to generic ontologies (expressed in OWL), by eliminating in this way the MOMIS limitation to use only the lexical ontology WordNet. Moreover, we introduce in the MOMIS system a multiple ontology approach with respect to the actual single ontology approach. In the following, we will refer to this new approach as the MOMIS-STASIS approach.

The MOMIS-STASIS approach is shown in Figure 1. It can be divided into two macro-steps: STASIS : Semantic Link Generation (shown in Figure 1-a) and MOMIS : Global Schema Generation (shown in Figure 1-b).

2.1 STASIS : Semantic Link Generation

As stated in [2, 1] the key aspect of the STASIS framework, which distinguishes it from most existing semantic mapping approaches, is to provide an easy to use GUI, allowing users to identify semantic elements in an easy way. Once this identification has been performed STASIS lets users map their semantic entities to those of their business partners where possible assisted by STASIS. This allows users to create mappings in a more natural way by considering the meaning of elements rather than their syntactical structure. Moreover, all mappings that have been created by STASIS, as well as all semantic entities, are managed in a distributed registry and repository network. This gives STASIS another significant advantage over traditional mapping creation tools as STASIS may reuse all mappings. This allows STASIS to make some intelligent mapping suggestions by reusing mapping information from earlier semantic links.

Besides the semantic links explicitly provided by the user, an Ontology-driven Semantic Mapping approach, for the STASIS framework, has been proposed [4]. The mappings between semantic entities being used in different schemas can be achieved based on annotations linking the semantic entities with some concepts being part of an ontology. In [5], this framework has been further elaborated and it has been applied to the
context of products and services catalogues. An overview of the process for Ontology-driven Semantic Mapping Discovery is given in Figure 1-a. It can be summed up into 3 steps (each step number is correspondingly represented in figure): (1) obtaining a neutral schema representation, (2) local source annotation, and (3) semantic mapping discovery.

**Step 1. Obtaining a neutral schema representation**

As sketched in Figure 1-a, the STASIS framework works on a neutral representation, which abstracts from the specific syntax and data model of a particular schema definition; therefore, all the structural and semi-structural local sources first need to be expressed in a neutral format. The neutral representation is obtained by describing the local schemas through a unified data model called Logical Data Model (LDM). For the purpose of this paper, we abstract from the specific features of LDM and we consider that this model contains common aspects of most semantic data models: it allows...
the representation of classes (or concepts) i.e. unary predicates over individuals, relationships (or object properties) i.e. binary predicates relating individuals, and attributes (or data-type properties) i.e. binary predicates relating individuals with values such as integers and strings; classes are organized in the familiar is-a hierarchy. Classes, relationships and attributes are called semantic entities.

Step 2. Local source annotation
The proposed mapping process identifies mappings between semantic entities through a “reasoning” with respect to aligned ontologies. For this purpose the semantic entities need to be annotated with respect to one or more ontologies. More formally, an annotation element is a 4-tuple \(< ID, SE, R, concept >\) where \(ID\) is a unique identifier of the given annotation element; \(SE\) is a semantic entity of the schema; \(concept\) is a concept of the ontology; \(R\) specifies the semantic relationship which may hold between \(SE\) and \(concept\). The following semantic relationships between semantic entities and the concepts of the ontology are used: equivalence (AR\_EQUIV); more general (AR\_SUP); less general (AR\_SUB); disjointness (AR\_DISJ).

Actually within the STASIS framework are implemented only simple automatic annotation techniques, e.g. the “name-based technique” where the annotation between a semantic entity and an ontology concept is discovered by comparing only the strings of their names. The main drawback of this automatic technique is due to the existence of synonyms (when different words are used to name the same entities, e.g. “Last Name” and ”Surname”) and homonyms (when the same words are used to name different entities, e.g. “peer” has a sense “equal” as well as another sense “member of nobility”) [9]. For these reason the designer has to manually refine the annotations in order to capture the semantics associated to each entities. In Section 3 a preliminary idea to overcome this limitation is described.

Step 3. Semantic mapping discovery
Based on the annotations made with respect to the ontologies and on the logic relationships identified between these aligned ontologies, reasoning can identify correspondences among the semantic entities and support the mapping process. Given two schemas \(S1\) and \(S2\), and assuming that OntologyA and OntologyB are the reference ontologies which have been used to annotate the content of \(S1\) and \(S2\) respectively, given a mapping between OntologyA and OntologyB which provides a correspondence between concepts and relationships in the two ontologies, a semantic mapping between the annotated schemas \(S1\) and \(S2\) is derived. The following semantic mappings between entities of two source schemas (called semantic link- SL) can be discovered: equivalence (EQUIV); more general (SUP); less general (SUB); disjointness (DISJ); this definition is based on the general framework proposed in [10].

More formally, an SL is a 4-tuple \(< ID, semantic\_entity1, R, semantic\_entity2 >\), where \(ID\) is a unique identifier of the given mapping element; \(semantic\_entity1\) is an entity of the first local schema; \(R\) specifies the semantic relationship which may hold between \(semantic\_entity1\) and \(semantic\_entity2\); \(semantic\_entity2\) is an entity of the second local schema.
An application example of the Ontology Driven Semantic Mapping approach is described in Section 2.3; other examples can be found in [5].

2.2 MOMIS: Global Schema Generation

The MOMIS Data Integration System, which performs information extraction and integration from both structured and semi-structured data sources, is presented in [7, 6].

Information integration is performed in a semi-automatic way, by exploiting the semantic links among source schemas and using a combination of clustering techniques and Description Logics. This integration process gives rise to a virtual integrated view of the underlying sources, where mapping rules and integrity constraints are specified to handle heterogeneity. Given a set of data sources related to a domain it is thus possible to synthesize, in a semi-automatic way, a Global Schema (called *Global Virtual View* - GVV) and the mappings among the local sources and the GVV. Mappings among source schemas and the GVV are defined with a Global-As-View (GAV) approach: each class of the GVV is characterized in terms of a view over the sources.

In the MOMIS System, semantic links among source schemas are mostly derived with lexicon techniques based on the lexical annotation with respect to WordNet; then, all these semantic links are collected in a Common Thesaurus.

In this paper we consider as semantic links among source schemas the semantic links defined with the STASIS framework; in other words, we consider as input of the GVV generation process the *Common Thesaurus SLs* generated by the STASIS framework. An overview of this GVV generation process is given in Figure 1-b.

In the GVV generation process classes describing the same or semantically related concepts in different sources are identified and clustered in the same global class. Exploiting the Common Thesaurus SLs and the local sources schemas, our approach generates a GVV consisting of a set of global classes, plus a Mapping Table (MT) for each global class, which contains the mappings to connect the global attributes of each global class with the local sources’ attributes. A MT is a table where the columns represent the local classes (LG) belonging to the global class G and whose rows represent the global attributes of G. An element $MT[GA][L]$ represents the set of local attributes of the local source L which are mapped onto the global attribute GA. The ontology designer may interactively refine and complete the proposed integration results; in particular, the mappings which has been automatically created by the system can be fine tuned in the Mapping Refinement step. Intuitively, the GVV is the intensional representation of the information provided by the Integration System, whereas the mapping assertions specify how such an intensional representation relates to the local sources managed by the Integration System.

A query posed by a user with respect to a global class is rewritten as an equivalent set of queries expressed on the local schemas (local queries); this query translation is carried out by considering the mappings between the GVV and the local schemas. Results from the local sources are then merged exploiting reconciliation techniques and proposed to the user.

For a complete description of the methodology to build and query the GVV see [7, 6].
2.3 Example

As a simple example let us consider two relational local sources L1 and L2, where each schema contains a relation describing purchase orders:

L1: PURCHASE_ORDER(ORDERID, BILLING_ADDRESS, DELIVERY_ADDRESS, DATE)
L2: ORDER(NUMBER, CUSTOMER_LOCATION, YEAR, MONTH, DAY)

In the following, we will described step by step the application of the MOMIS-STASIS approach on these two sources.

STASIS: Semantic Link Generation

Step 1. Obtaining a neutral schema representation
During this step the local sources L1 and L2 are translated in the neutral representation and are represented in LDM data model; for a complete and formal description of such representation see [4], where a similar example was discussed. As said before, for the purpose of this paper, we consider that the local schema L1 contains a class PURCHASE_ORDER with attributes ORDERID, BILLING_ADDRESS, DELIVERY_ADDRESS, DATE.

In this way L1.PURCHASE_ORDER, L1.PURCHASE_ORDER.BILLING_ADDRESS, L1.PURCHASE_ORDER.DELIVERY_ADDRESS etc. are semantic entities. In the same way the local schema L2 contains a class ORDER with attributes NUMBER, CUSTOMER_LOCATION, YEAR, MONTH, DAY.

Step 2. Local Source Annotation
For the sake of simplicity, we consider the annotation of schemas and the derivation of mappings with respect to a single common ontology ("Ontology-based schema mapping with a single common ontology" scenario considered in [4]). Let us give some examples of annotations of the above schemas with respect to the Purchase Order Ontology shown in Figure 2. In the examples the identifier ID is omitted and a concept C of the ontology is denoted by "O:C". In a simple annotation the concept O:C is a primitive concept or a primitive role of the ontology (e.g. the class O:ADDRESS or the property O:BILLING). In a complex annotation the concept O:C is obtained by using the OWL language constructs (e.g. "O:ADDRESS and BILLING-1.Purchase.Order" where BILLING-1 denotes the inverse of the property O:BILLING).

The following are examples of simple annotations:

(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV, O:ADDRESS)
and

(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV, O:BILLING).

These annotations are automatically discovered by applying the automatic "name-based" technique (see Section 2.1). However, as this technique does not consider the semantics associated to each entities, the following annotation

(L2.ORDER.CUSTOMER_LOCATION, AR_EQUIV, O:ADDRESS)
Fig. 2. The ontology of Purchase_order

is not discovered: the entities CUSTOMER_LOCATION and the concept ADDRESS have complete different names but, in this context, they have the same senses. In Section 3 a preliminary idea to overcome this problem is described.

An example of complex annotation is

(L1.PURCHASE_ORDER.DELIVERY_ADDRESS, AR_EQUIV, 
O:Address and Shipping-1.Purchase_Order)

which can be considered as a refinement by the designer of the above simple annotations to state that the address in the PURCHASE_ORDER table is the “address of the Shipping in a Purchase Order”.

Other examples of complex annotations are:

(L1.PURCHASE_ORDER.BILLING_ADDRESS, AR_EQUIV, 
O:Address and Billing-1.Purchase_Order)

where is explicitly declared by the designer to state that the address in the PURCHASE_ORDER table is the “address of the Billing in a Purchase_Order”.

(L2.ORDER.CUSTOMER_LOCATION, AR_EQUIV, 
O:Address and Shipping-1.Purchase_Order)

where is explicitly declared by the designer to state that the address in the ORDER table is the “address of the Shipping in a Purchase_Order”.

8
Moreover, the designer supplies also the annotations with respect to the ontology for the semantic entities L1.PURCHASE_ORDER.ORDERID, L1.PURCHASE_ORDER.DATE and L2.ORDER.NUMBER, L2.ORDER.YEAR, L2.ORDER.MONTH, L2.ORDER.DAY.

**Step 3. Semantic mapping discovery**

From the previous annotations, for example, the following semantic link is derived:

\[(L2.ORDER.CUSTOMER_LOCATION, \text{EQUIV}, \text{L1.PURCHASE_ORDER.DELIVERY_ADDRESS})\]

while no semantic link among CUSTOMER LOCATION and BILLING ADDRESS is generated.

**MOMIS: Global Schema Generation**

Given the set of semantic links described above and collected in the Common Thesaurus, the GVV is automatically generated and the classes describing the same or semantically related concepts in different sources are identified and clustered in the same global class. Moreover, the Mapping Table shown in Table 1 is automatically created by the MOMIS-STASIS approach.

<table>
<thead>
<tr>
<th>Global attributes</th>
<th>Local attributes</th>
<th>Local attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER</td>
<td>NUMBER</td>
<td>NUMBER</td>
</tr>
<tr>
<td>DATE</td>
<td>YEAR,MONTH,DAY</td>
<td>DATE</td>
</tr>
<tr>
<td>CUSTOMER_LOCATION</td>
<td>CUSTOMER_LOCATION</td>
<td>DELIVERY_ADDRESS</td>
</tr>
<tr>
<td>BILLING_ADDRESS</td>
<td>NULL</td>
<td>BILLING_ADDRESS</td>
</tr>
</tbody>
</table>

**Table 1.** Mapping Table example

The global class ORDER is mapped to the local class ORDER of the L1 source and to the local class PURCHASE_ORDER of the L2 source. The NUMBER, DATE and CUSTOMER_ADDRESS global attributes are mapped to both the sources, the BILLING_ADDRESS global attribute is mapped only to the L2 source.

3 **Future Work**

One of the main advantage of the proposed approach is an accurate annotation of the schemas that produces more reliable relationships among semantic entities. The relationships among semantic entities are then exploited in order to obtain a more effective integration process. On the other hand, this more accurate annotation has the disadvantage that is currently performed manually by the integration designer.

In this work, we describe only a preliminary idea to overcome the problem of manual annotation, which will be the main subject of our future research.

Several works about automatic annotation are proposed in literature but only a few of them are applied in the context of schemas/ontologies matching discovery. In [13,
where we introduced a mapping technique based on Lexical Knowledge Extraction:

first, an Automatic Lexical Annotation method is applied to annotate, with respect to
WordNet, schemas/ontologies elements then lexical relationships are extracted based
on such annotations.

Moreover, in [15] a way to apply our Automatic Lexical Annotation method to the
SCARLET matcher [3], is presented. SCARLET is a technique for discovering relation-
ships between two concepts by making use of online available ontologies. The matcher
can discover semantic relationships by reusing knowledge declared within a single on-
tology or by combining knowledge contained in several ontologies. By applying Au-
tomatic Lexical Annotation based on WordNet, the matcher validates the discovered
mapping by exploring the semantics of the terms involved in the matching process.

Starting from these works, we agree that the WordNet lexical ontology can be used
to improve the annotation phase of the Ontology Driven Semantic Mapping process.
The strength of an lexical ontology like WordNet is the presence of a wide network
of semantic relationships among the different words meanings, which represent a key
element for automatic annotation techniques.

Let us consider the example shows in Figure 3: during the local source annota-
tion step, the relationships between semantic entities and ontology concepts have to
be discovered. Our idea can be summed up in three main steps (each step number is
correspondingly represented in figure):

1. **Ontology and local source annotation with respect to WordNet**: both the ontology
and the local source, are annotated, with respect to WordNet, by using the Auto-
matic Lexical Annotation method described in [13]: e.g., as shown in Figure 3, the
semantic entity “Surname” is annotated with the first sense in WordNet (indicated
in Figure with “#1”) for the word “SURNAME” and the ontology concept “Last-
Name” is annotated with the first sense in WordNet for the word “LASTNAME”;

2. **WordNet semantic relationship discovery**: starting from the previous annotations,
a set of WordNet semantic relationships (synonym (equivalence), hypernym (more
general) etc.) is discovered among semantic entities and ontology concepts: e.g.,
as shown in Figure 3, a synonym relationship is discovered between the semantic
entity “Surname” and the ontology concept “LastName”.

3. **Local source annotation for Ontology Driven Semantic Mapping**: starting from the
set of WordNet semantic relationships previously discovered, a correspondent set of
annotation for Ontology-Driven Semantic Mapping can be discovered: e.g., starting
from the WordNet synonym relationship between “Surname” and the “LastName”,
the following annotation is established (the annotation unique identifier ID is omit-
ted):

   (surname, AR_EQUIV, O:LastName)

In this way, we can automatically annotate a set of local schema semantic entities with
respect to the considered ontology. However, these annotations can be incomplete (be-
cause WordNet does not contain many domain dependent words) and require a designer
refinement. Even if this preliminary idea needs to be further investigated, it represents a
fundamental start point to help the designer during the time consuming task of manual
annotation.
Another future work will be the investigation of automatic techniques to discover the relationships among semantic entities combining the exploration of multiple and heterogeneous online ontologies with the annotations provided by the WordNet lexical ontology. The use of online ontologies represents an effective way to improve the semantic mapping process. For example, in [16], automatic techniques to discover the relationships between two concepts automatically finding and exploring multiple and heterogeneous online ontologies, have been proposed.

Fig. 3. A preliminary idea to perform automatic annotation for Ontology-driven Semantic Mapping

4 Conclusions

In this paper, we have described the early effort to obtain an effective approach for Ontology-Based Data Integration combining the techniques provided by the MOMIS and the STASIS frameworks. In particular, with the Ontology-driven Semantic Mapping framework we have performed in the Data Integration System the annotation of data sources elements with respect to generic ontologies (expressed in OWL), by eliminating in this way the MOMIS limitation to use only the lexical ontology WordNet. Moreover, we have introduced in the MOMIS system a multiple ontology approach with respect to the actual single ontology approach. Even if this work needs to be further investigated (as described in Section 3), it represents a fundamental start point versus an automatic Ontology-Based Data Integration System.

5 Acknowledgment

This work supported by STASIS IST project (FP6-2005-IST-5-034980).
References


A Framework for Privacy Preserving Face Matching

Filippo Volpi\textsuperscript{1}, Monica Scannapieco\textsuperscript{2}, and Tiziana Catarci\textsuperscript{1}

\textsuperscript{1} Dipartimento di Informatica e Sistemistica, SAPIENZA Università di Roma
(volpi,catarci)@dis.uniroma1.it
\textsuperscript{2} Istat
scannapi@istat.it

Abstract. The paper proposes a framework for face matching that ensures privacy preservation. The framework mainly consists of a protocol involving three parties: two are the parties interested in finding common people in their image sets and the third party plays a role necessary for privacy preservation. We discuss the security aspects of the protocol and we provide extensive experiments to prove the effectiveness of the protocol.

KEYWORDS: privacy, face matching

1 Introduction

Modern information systems have the need of exchanging information while respecting privacy requirements. Different kind of information can be exchanged, including structured data, documents and images. This paper deals with facial images belonging to different organizations that want to find the “intersection” of the images stored by each organization. Due to privacy requirements, an organization cannot in general simply send the whole image set to the other one, in order to find common people. Therefore, specific methods need to be developed in order to address such requirements. In this paper, we propose a framework for face matching that ensures privacy preservation. The framework mainly consists of a protocol executed between the two parties interested in finding common people in their image set and a third party necessary for privacy preservation.

There are several real scenarios for the framework we propose, including analysis of images and videos for intelligence applications and federated identity management applications.

As we will see in the related work section, the area of face matching has a long history and there are plenty of methods that serve the purpose of matching facial images. However, to the best of our knowledge, this is the first contribution in the direction of providing a general way to match facial images in privacy-protected contexts. Moreover, there are also proposals dealing with the problem of privacy-preserving object intersection (e.g., [23]) but they are related to structural data.
More specifically, the contributions of the paper can be summarized as follows:

– Selection of an existing method for face matching, having all the requirements necessary for being integrated in a privacy preserving framework.
– Proposal of a protocol for privacy preserving face matching.
– Experimentation of the protocol in order to prove the effectiveness of the selected method when used in a privacy-preserving context (where several parameters need to be agreed among involved parties).

The rest of the paper is organized as follows. In Section 2 related work are described, followed by the description of the framework (Section 3). An extensive experimental activity is documented in Section 4, while Section 5 draws some conclusions.

2 Related Work

Face recognition technology is one of the few biometric methods that possesses the merits of both high accuracy and low intrusiveness. That makes it suitable for real applications including law enforcement and surveillance, information security, and access control [1]. However, there are several issues that need to be solved in order to actually perform a face recognition task. First, images are high-dimensional input and could be affected by noise. Second, face photos can be taken under uncontrolled conditions such as illumination, pose and expression changes. That implies the nonexistence of bijections between subjects and related photos, unlike often happens with textual records (under the hypothesis that no errors are present in textual records, otherwise a record linkage problem occurs also for these type of data). On the other side, real databases can contain only one image for subject; this is referred to as the one sample problem and represents a significant challenge for researchers.

In more than 30 years of research, a plethora of feature extraction methods for face recognition have been developed. They can be categorized in three groups: holistic, local and hybrid. The holistic matching methods use the whole face region as the raw input to a recognition system. One of the most widely used representations of the face region is eigenpictures, which are based on principal component analysis [3]. Among the other method, we cite \((PC)^2 A\) [16], noise model [17] and Discriminant Eigenfaces [18]. Local matching methods use the local facial features for recognition. Care must be taken when deciding how to incorporate global configurational information into local face model. Examples are Directional Corner Points (DCP) [19], 1D-HMM [20], Elastic Bunch Graph Matching (EBGM) [21] and Linear Binary Pattern (LBP) [22]. Hybrid methods, just as the human perception system, use both local features and the whole face region to recognize a face. One can argue that these methods could potentially offer the best of the two types of methods.

Traditional methods suffer from serious performance problems or even fail to work under the one sample problem condition [2]. If only one training sample
per person is used, the average recognition rate of Eigenface \[3\] falls to below 65%, a 30% drop from 95% when nine training samples per person are given. The same happens to all the eigenface-based techniques, including Probabilistic-based Eigenface \[4\], Linear Discriminative Analysis (LDA) based subspace algorithms \[5, 6\], Support Vector Machine (SVM) based method \[7\], Feature Line method \[8\], Evolution Pursuit \[9\] and Laplacian faces \[10\]. All of these approaches claimed to be superior to Eigenface. However, it may not be the case if only one training sample per person is available, due to the fact that most of them will either reduce to the basic Eigenface approach or simply fail to work in that case.

Supervisioned learning-based methods are excluded, because we require a completely automated technique for our protocol.

On the privacy-preserving side, the two parties must extract their feature independently. No data exchanges are allowed, but features must be comparable with the similarity functions. Methods based on learning mechanism generally can create privacy problems. Neural network and SVM are not good for the proposed task, because the classifiers obtained from the learning step on the two parties are different making the features uncomparable. The best eligible method is the Local Binary Pattern, because its feature extraction step is made image by image, so the features of one image are unrelated to the other images. So the two parties features are comparable, without revealing any information about the other party content. LBP method will be discussed in Section 3.1.

3 A Privacy-Preserving Face Matching Framework

The aim of this paper is to propose a protocol that while performing face matching, also preserves the privacy of matched images. The protocol considers two parties, named P and Q, each one owning a database of facial images. P wants to know which of the faces it stores are also present in the Q’s dataset, and similarly Q wants to know the images stored by P that are also stored by itself. The images are frontal and only one photo for subject is collected. The idea of the protocol is that P and Q extract the features; then, a third party, called W, performs the match in the feature space with a similarity function.

3.1 Background

In this subsection, we discuss the method chosen for feature extraction, that is Linear Binary Pattern (LBP). The original LBP operator, introduced by Ojala \[11\], is a powerful means of texture description. The operator labels the pixels of an image by thresholding the 3x3-neighborhood of each pixel with the center value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. Figure 2 shows the basic LBP operator.

There are some relevant extensions of LBP. First, the operator was extended to use neighborhoods of different sizes \[12\]. This extension permits any radius
and number of pixels in the neighborhood, using circular neighborhoods and bilinearly interpolating the pixel values. For neighborhoods we will use the notation (C,R) where C stands for sampling points on a circle of radius of R. See Figure 2 for an example of different neighborhoods in sampling and radius.

Another extension to the original operator uses the so called uniform patterns [12]. A Local Binary Pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110 and 10000011 are uniform patterns. Ojala et al. noticed that in their experiments with texture images, uniform patterns account for a bit less than 90% of all patterns when using the (8,1) neighborhood and for around 70% in the (16,2) neighborhood.

We use both extensions for the LBP operator that we have included in our protocol. In order to indicate our LBP operator we will use the following notation: \( \text{LBP}^{u2}_{C,R} \). The subscript indicates that we are using the operator in a (C,R) neighborhood. The superscript \( u2 \) stands for using only uniform patterns and labelling all remaining patterns with a single label.

A histogram of the labeled image \( f_l(x, y) \) can be defined as

\[
H_i = \sum_{x,y} I\{ f_l(x, y) = i \}, \quad i = 0, ..., n - 1,
\]

where \( n \) is the number of different labels produced by the LBP operator and

\[
I\{ A \} = \begin{cases} 
1, & \text{A is true} \\
0, & \text{A is false}.
\end{cases}
\]

This histogram contains information about the distribution of the local micropatterns, such as edges, spots and flat areas, over the whole image. For efficient face representation, one should retain also spatial information. For this
purpose, the image is divided into regions $R_0, R_1, ..., R_{m-1}$ (see Figure 3.a) and the spatially enhanced histogram is defined as

$$H_{i,j} = \sum_{x,y} I\{f(x, y) = i\} I\{(x, y) \in R_j\}, i = 0, ..., n - 1, j = 0, ..., m - 1.$$  

In this histogram, we effectively have a description of the face at three different levels of locality: (i) the labels for the histogram contain information about the patterns at a pixel-level, (ii) the labels are summed over a small region to produce information at a regional level and (iii) the regional histograms are concatenated to build a global description of the face.

From the pattern classification point of view, a usual problem in face recognition is having a plethora of classes and only a few, possibly only one, training sample(s) per class. For this reason, we do not need to rely on sophisticated classifiers, but we can consider a nearest-neighbor classifier. The Chi square statistic ($\chi^2$) is used for histograms:

$$\chi^2(S, M) = \sum_i (S_i - M_i)^2 \overline{S_i + M_i}.$$  

This measure can be extended to the spatially enhanced histogram by simply summing over $i$ and $j$.

![Fig. 3.](image)

**Fig. 3.** (a) An example of a facial image divided into 7x7 windows. (b) The weights set for weighted $\chi^2$ dissimilarity measure. Black squares indicate weight 0.0, dark grey 1.0, light grey 2.0 and white 4.0.

Once an image is divided into regions, it is common that some of the regions contain more useful information than others in terms of distinguishing between people. For example, eyes seem to be an important cue in human face recognition [13, 1]. To take advantage of this, a weight can be set for each region based on the importance of the information it contains (see Figure 3.b). For example, the weighted $\chi^2$ statistic becomes

$$\chi^2_w(S, M) = \sum_{i,j} w_j (S_{i,j} - M_{i,j})^2 \overline{S_{i,j} + M_{i,j}},$$  

in which \( w_j \) is the weight for the region \( j \).

### 3.2 Description of the Protocol

Our protocol for privacy-preserving face matching consists of three different steps, as shown in Figure 4.

![Diagram of the protocol](image)

**Fig. 4.** The phases of the protocol

In Step I the parties P and Q agree on some protocol parameters, namely: (i) sizes of the labeled images, (ii) \( C \) and \( R \) of the LBP operator, (iii) number of regions and eyes coordinates in the normalized image and (iv) kind of transposition method (see later Section 3.3). Then P and Q operate the normalization of the photos of each database. Two kind of normalization can be used. The first kind is an *eyes normalization*, in which eyes coordinates in the image are decided a priori and each photo goes through the same transformation. The reason to use this normalization is that pose and angles changes in a photo may affect the eyes distance and thus the matching, so a fixed distance needs to be imposed. The second kind of normalization is an *eyes and mouth normalization*, in which the same happens also to mouth coordinates. This latter normalization can take the same advantages of the eyes normalization, but it must be noticed that there is the possibility to cancel a discriminative measure. The result of the normalization step is a set of gray-scale images, with eyes (and mouth) at the same position. In this step there is also the option of masking the image that means to apply a mask that removes hairs and background.

Step II of the protocol requires to extract the features and to build the spatially enhanced histograms. It must be noticed that the mapping of all non-uniform micropattern to the same label is a form of dimensionality reduction. An option of this step is whether to use only the uniform micropatterns related information in the histogram or not. This will be discussed in the Section 4.

Step III of the protocol consists of a privacy preservation step and of the comparison of the spatially enhanced histograms by the third party W. Before
the transmission, data are encrypted in order to prevent W from disclosing original information. Usually cryptographic methods do not preserve distances in the feature space. This is because cryptographic methods are typically designed so that two “similar” objects have different cyphers. As explained in Section 3.3 we use a transposition cipher that actually preserves distances in a soft computational way. Thus, histograms are ciphered and then sent to W that performs the matching. In the privacy preservation step, we have the option whether to use the standard chi-square formula or the weighted one. That will be discussed in the next Section 3.3.

### 3.3 Privacy Guarantees of the Protocol

In order to make the protocol robust to privacy attacks, we consider a scenario where W is honest-but-curious. If W were honest, we can trust it and we are sure it won’t violate the protocol. Specifically W will return to P only the matching subset of the P’s data. The same thing will happen for Q. W will not have any look at the data, the matching takes place and the privacy is guaranteed.

Instead, we assume that W is honest-but-curious. Thus, W executes correctly all the steps of the protocol in which it is involved, but it also tries to collect, extract, store information. In this scenario, we start from the good news that the LBP operator is not at all invertible. The motivations are different:

- First, the thresholds of the singular windows that must be chosen when applying the LBP method are variable.
- Second, all the non-uniform micropatterns are mapped with the same label, loosing information.
- Third, the spacial information is contained only at the regional level.

Hence, W would have a quite hard task to piece together the original image from the spatially enhanced histogram. However, in order to have a proven privacy guarantee, a transposition cipher method is implemented. Transposition cipher methods are methods of encryption by which the positions held by units of the object to be ciphered are shifted according to a regular system, so that the cipher constitutes a permutation of the original objects. The two parties P and Q can perform different types of permutations, e.g. columnar transposition, disrupted transposition (where certain positions in the data are blanked out), etc. We have chosen to leave the type of permutation to be applied on the data by P and Q as a parameter to be agreed on between the parties in the initialization phase of the protocol. In the honest-but-curious scenario is strongly recommended not to use the $\chi^2$ weighted formula, because the region related weight can suggest information to W.

### 4 Experiments

#### 4.1 Experimental setup

Experiments are performed on a subset of the Color FERET database [14,15]. A subset $fa$ of 1201 images is assigned to party P and a subset $fb$ of 1201 is
assigned to Q. The two subsets contain corresponding images, that is the images are different for instance for the face expression, though referred to the same person. Experiments are divided into three phases.

In the first phase, the combination of parameters is explored, namely:

- Four different types of interpolation is tested during normalization (see Figure 5),
- four LBP operators (8,1), (8,2), (16,2) and (24,3),
- two types of normalization (eyes and eyes and mouth),
- whether masking the images or not,
- whether using the weighted formula of $\chi^2$ or not,
- whether using only the uniform related parts of histograms or not.

![Fig. 5. The four kind of considered interpolations.](image)

All these options result in $4 \times 4 \times 2 \times 2 \times 2 \times 2 = 256$ combinations. For each combination the distance matrix (1201x1201) has been computed.

The original \(f_a\) and \(f_b\) datasets, and hence the distance matrix, present some multiple image per subject. From the matrix 865 rows and 865 columns are randomly chosen, where 865 represent the numbers of different people both in rows and columns. In other words configurations with unique image per person on rows and columns are generated. In this way we have created datasets satisfying the under the one sample problem condition.

We have computed precision and recall metrics for all the 256 combinations. For each combination, 1100 configurations are randomly chosen, thus we performed a total of tests are $1100 \times 256 = 281600$.

Results of phase 1 are summarized in Figure 6.

Phases two and three of the experiments consist of local searches. In the phase two, we analyze the conjunction of parameter combinations, in other words we say that two subjects are considered matching if more than one combinations consider them as matching.

The phase three, instead, uses disjunction of parameter combinations to decide matching images. The total number of tests performed in phased two and three is 843200 and 925800 respectively.
<table>
<thead>
<tr>
<th>LBP</th>
<th>Interpolation</th>
<th>Normalization</th>
<th>&quot;eyes+mouth&quot;</th>
<th>Normalization</th>
<th>&quot;eyes&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Weighted Metric</td>
<td>Not weighted Metric</td>
<td>Weighted Metric</td>
<td>Not weighted Metric</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6. Precision and recall of the 256 configurations. The cells report the case in which non uniform pattern are considered (upper) and when they are not (lower).
For the sake of conciseness we report only summarizing results concerning these two phases in Figure 7.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Precision and recall</th>
<th>Configuration Parameters</th>
<th>Precision and recall</th>
<th>Configuration Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.8 % 76.2 %</td>
<td>(p,16,2,o,2,nm,u)</td>
<td>94.7 % 69.2 %</td>
<td>(np,16,2,o,2,nm,t)</td>
</tr>
<tr>
<td>2</td>
<td>98.6 % 75.4 %</td>
<td>(p,16,2,o,2,nm,t)</td>
<td>98.6 % 68.6 %</td>
<td>(np,16,2,o,4,nm,t)</td>
</tr>
<tr>
<td>3</td>
<td>95.5 % 77.5 %</td>
<td>(p,16,2,o,2,nm,t)</td>
<td>94.1 % 70.9 %</td>
<td>(np,16,2,o,4,nm,t)</td>
</tr>
</tbody>
</table>

Fig. 7. Precision and recall for the three experimental phases.

The parameter configuration that exhibits better performance includes the LBP(16,2) operator, the eyes normalization, the use of all the micropattern (uniform and non-uniform) and the use of not masked images. The best results are found in phase two. Under the honest W hypothesis, we reach a F-measure of 85.45 % (98.6% precision and 75.4% recall). Under the honest-but-curious W hypothesis instead, we reach a F-measure of 79.26 % (98.6% precision and 68.6% recall).

5 Conclusions

The paper proposes a framework for face matching in privacy preserving contexts. The work can be extended in several directions. First, we will perform the tests of different transposition cypher methods in order to realize if exists a method that better serves to the purpose of the framework. Second, we will extend experiments in order to consider also additional datasets exhibiting different characteristics. Third, more complex scenarios, like the one in which more than two parties are involved in the protocol will be also studied.

Acknowledgements

Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

References


Towards Automatic Support to Mapping discovery in eBusiness resources exchange

Michele Missikoff, Fabrizio Smith, Francesco Taglino
Istituto di Analisi dei Sistemi e dell'Informatica
Viale Manzoni 30, Roma, Italia
michele.missikoff@iasi.cnr.it, fabrizio.smith@iasi.cnr.it, francesco.taglino@iasi.cnr.it

Abstract. This paper addresses the problem of semantic annotation of business documents, by using a reference business ontology. This is essentially a mapping discovery task, where we have two structures represented as labeled directed graphs. Then, we need to find the correspondences that exist among the elements (and subgraphs) of the two graphs. We propose a layered approach, where we compute first a similarity matrix among elements and then correspondences between paths is derived, by applying a bipartite matching algorithm. Finally, the method proposes path expressions which are the final outcomes.

Keywords: ontology, mapping discovery, path expression, string matching, graph matching.

1 Introduction

In this paper we address the problem of semantic annotation of a business document by using a reference ontology. In particular, we consider the schema (e.g., XMLS, RDFS) of a business document (e.g., a Purchase Order) and we identify the concepts of a reference ontology (RO) that denote the meaning of the elements of such a schema (referred to as resource schema RS).

In essence, having represented the two structures in a graph-theoretic form, a semantic annotation is a mapping between elements (nodes, edges and paths) of the RS and elements of the RO. Mapping discovery is a hard task since, given a fragment of the reality, there are infinite ways of modeling it, by using different names, different relationships, different complex structures. We assume that the observed (and modeled) reality is faithfully represented by the RO and the RS is therefore a sub-model, in semantic terms, of the former. Then, the differences that appear on the terminological, syntactic, and structural ground will be reconciled on the semantic ground.

The proposed method proceeds with a layered approach, along the following steps:

- term matching
- structural analysis and path matching
- path expression building, with a rule-based approach

---

1 This work is partially supported by the EU FP7 IST 216256 IP COIN Project, www.coin-ip.eu
In the literature the problem of mapping discovery has been widely addressed, however the existing proposals have a limited scope, since they mainly address the mapping between individual elements (e.g., concepts and relationships) and only a few address complex sub-models as a whole. Furthermore, we go beyond the logic correspondence (e.g., subsumption), introducing a formalism to represent mappings based on an algebra of path composition and rule-based transformations. Our ultimate goal is to discover in a (semi)automatic way the set of operations that allow one structure to be transformed into another, without loss of semantics. This research has been carried out in the context of the semantic annotation and reconciliation platform of the European Project COIN.

The rest of the paper is organized as follows. Section 2 introduces the semantic annotations. Section 3 surveys related works. Section 4 describes our semantic annotations discovery approach in details. Section 5 concludes and presents future work.

2 Semantic Annotations

In this work we assume that both the resource schema and the reference ontology are represented as a directed labeled graph. The adopted graph format is conceived as follows: i) each concept or datatype is represented as a node; ii) for each role $S$ and for each pair of concepts $A$ and $B$ in the domain and range of $S$, respectively, there is a directed edge $S$ in the graph from $A$ to $B$; iii) for every attribute $P$ and for every concept $A$ and datatype $D$ in the domain and range of $P$, respectively, there is a directed edge $P$ from $A$ to $D$.

In the following, let us consider, as running example, the scenario in Figure 1 with fragments of a RS (left-hand side) and a RO (right-hand side), extracted from a purchase order of an eBusiness domain.

![Fig. 1. An example of fragments of a RS and a RO](image)

The semantic annotation (SA) process consists in establishing correspondences between elements of the RS and elements of the RO. To this, we propose three different levels of annotation, with a progression in richness and details. The first
Towards Automatic Support to Mapping discovery in eBusiness resources exchange

level (Terminological Annotation) is a keyword-based annotation where terms from the ontology are put in correspondence to terms from the resource with an m:n association. Let say in our example:

\{<RS:PO>,<RO:purchaseOrder>},\{<RS:purchaserInfo>,<RO:buyer>\},
\{<RS:name>, <RO:firstname,RO:surname,RO:organizationName>\}

In the second level (Path Annotation) paths in RO are mapped to paths in RS with an m:n association. There is a path between two nodes of the graph, A and B, if, starting at node A, it is possible to follow a sequence of adjacent edges to reach node B. We restrict our analysis to paths starting with a concept and ending with an attribute. The motivation is that this work is part of a wider effort concerning semantic reconciliation of business documents (e.g., an invoice). Therefore we focus on mappings that will support instance exchange and transformation. At this level, we simply aim at identifying couple of paths, one in RS and one in RO, that bears some affinity. Intuitively, a path annotation goes beyond concept matching, since it establishes a correspondence between two complex structures (subgraphs) in the resources we are considering. Path annotations make use of the terminological annotations related to the entities that occur in the paths.

In the third level (Semantic Annotation expression) we further enrich the picture by identifying the transformations that allow the rewriting of a fragment of RS into a corresponding fragment of RO, and vice-versa. Such transformations are achieved by using a few path operators (e.g. SPLIT, MERGE, MAP, CONVERT). MAP is a generic operator stating that two structures are in a correspondence. The other operators have a clear semantics. An example of semantic annotation expression is:

\{<RS:PO.orderHeader.header.purchaser.purchaserInfo.orgInfo.name> MAP
<RO:purchaseOrder.hasBuyer.buyer.organizationName>\}.

Now we further enrich the mapping expression by introducing a rule-based notation: we present it by means of an example (a more extensive elaboration is omitted for lack of space). Consider the previous SA expression with the associated terminological annotations. Given the above input, we may generate a mapping expression of the form:

RO:purchaseOrder(?x),RO:buyer(?y),RO:hasBuyer(?x,?y),
RO:organizationName(?y,?s):-
RS:PO(?x),RS:header(?u),RS:orderHeader(?x,?u),
RS:purchaserInfo(?y),RS:purchaser(?u,?y),RS:orgInfo(?v),
RS:organization(?y,?v),RS:name(?v,?s).

The expression is a logic rule, encoded in a Datalog like syntax, with the form head ← body, where the head is a conjunctive formula over the alphabet (concepts, roles, attributes) of RO, while the body is a conjunctive formula over the alphabet of RS. This kind of mapping rules (known in literature as GLAV or tgd [1,2]) are at the basis of the semantic reconciliation method developed in COIN. We will use this abstract rule language at a methodological level. When needed, it can be translated in a straight forward manner into, e.g., Jena\textsuperscript{2} rules to be executed on the Jena framework.

\textsuperscript{2} http://jena.sourceforge.net/
Since we represent both the RS and the RO as a directed labeled graph, our approach is neutral with respect to the particular data model of the resource schema. Adopting the terminology from Model Management [3], we can say that we take as input two models.

3 Related Works

In the recent period, the automation of the mapping discovery (i.e., finding correspondences or relationship) between entities of different schemas (or models) has attracted much attention in both the database (schema matching) and AI (ontology alignment) communities (see [4,5] for a survey). Schema matching and ontology alignment use a plethora of techniques to semi-automatically finding semantic matches. These techniques are based on different information sources that can be classified into: intensional knowledge (entities and associated textual information, schema structure and constraints), extensional knowledge (content and meaning of instances) and external knowledge (thesauri, ontologies, corpora of documents, user input). We can divide the methods/algorithms used in existing matching solutions in:

- rule-based methods, where several heuristics are used to exploit intensional knowledge, e.g. prompt [6], cupid [7];
- graph analysis, where ontologies are treated as graphs and the corresponding subgraphs are compared, e.g. similarity flooding [8], anchor-prompt [6];
- machine learning based on statistics of data content, e.g. GLUE [9];
- probabilistic approaches that combine results produced by other heuristics, e.g. OMEN [10].

Complex approaches, obtained by the combination of the above techniques, have been also proposed (e.g., OLA [11]), as well as frameworks that provide extensible libraries of matching algorithms and an infrastructure for the management of mappings (e.g., COMA [12]).

At the best of our knowledge, most of the previous matching approaches focus on finding a set of correspondences (typically 1:1) between elements of the input schemas, enriched eventually by some kinds of relationship (equivalence, subsumption, intersection, disjointness, part-of, merge/split). The construction of operational mapping rules (i.e., directly usable for integration and data transformation tasks) from such kind of correspondences is another challenging aspect. Clio [13], taking in input $n:m$ entity correspondences together with constraints coming from the input schemas (relational schema or XMLS), produces a set of logical mappings with formal semantics that can be serialized into different query languages (e.g., SQL, XSLT, XQuery). MapOnto [14] can be viewed as an extension of Clio when the target schema is an ontology.

4 Discovering Semantic Annotations

Given two models, let us say RS and RO, we build their graph representation (as explained in section 2). In doing this, we process subsumption (ISA) relation
materializing the inherited properties, i.e., adding edges to the graphs accordingly to the properties (roles or attributes) that concepts inherit from their subsumers.

In the first step of the matching process we consider only terminological knowledge. We start by processing the entity labels of the two graphs to build a term similarity matrix. The similarity matrix reports a similarity value (between 0 and 1) for every couple of elements \( \langle A, B \rangle \), where \( A \) belongs to RS and \( B \) belongs to RO and they are both either concepts, roles or attributes. After the terminological analysis the structure of the graphs is taken into account and paths belonging to RS and to RO are matched, starting from the similarity values computed in the previous step. This phase produces a first set of suggested path correspondences that are presented to the user for validation. Then, the actual annotation is achieved and the corresponding SA expressions are produced.

4.1 Terminological Analysis

In order to assign a similarity value to a pair of labels from the two graphs we combine the results of both a string similarity measure and a linguistic similarity measure. In the former, we consider labels as sequences of letters in an alphabet while in the latter, we consider labels as bags of words of a natural language (i.e. English) and we compute a similarity value based on their meaning.

**String Similarity.** We experimented several string similarity measures\(^3\) and finally we selected the Monge-Elkan distance [15], that was proposed as a field matching (i.e., record linkage) algorithm. The Monge-Elkan distance measures the similarity of two strings \( s \) and \( t \) evaluating recursively every substrings from \( s \) and \( t \); this approach is also capable to support the analysis of abbreviations or acronyms. To improve the accuracy of the algorithm, the input strings are normalized, i.e. characters are converted in lower case and special characters (digits, whitespaces, apostrophes, underscores) are discarded.

**Linguistic Similarity.** Our approach is based on the Lin Information-theoretic similarity [16] applied to the lexical database WordNet [17], that is particularly well suited for similarity measures since it organizes synsets (a synset is a particular meaning of a word) into hierarchies of ISA relations. Given two synsets of the WordNet taxonomy, the Lin measure can be used to state their similarity depending on the increase of the information content from the synsets to their common subsumers\(^4\). In order to use the Lin measure to compare two strings \( s \) and \( t \), we consider such strings as word entries of WordNet, and apply the Lin measure to all the pairs of synsets related to \( s \) and \( t \), respectively. We then define \( \text{sim}(s,t) \) as the higher computed similarity value, since we expect that words used in the RS and in the RO belong to the same domain sharing the same intended meaning.

We said that entity names are considered bags of words since they are, in general, compound words (purchaseOrder, contact_details) and unfortunately there is not a

---

\(^3\) available in the library SimPack (http://www.ifi.uzh.ch/ddis/simpack.html)

\(^4\) see WordNet Similarity Library (http://nlp.shef.ac.uk/result/software.html)
well-defined symbol that can be interpreted as a separator. For these reasons we
realized a label resolution algorithm to tokenize a string looking (from right to left)
for maximal substrings that have, after stemming, an entry in WordNet; also some
special characters (e.g., “ “ _,” -”) are taken into account. In this way, we can also
filter the “noise” in the labels, deleting substrings recognized as prepositions,
conjunctions, adverbs and adjectives that are not included in the WordNet taxonomy.

Finally we can define an algorithm to compute a linguistic similarity value given
two labels, represented as bags of strings:

\[
\text{Double: match}(\text{bag of strings } \text{term1}, \text{bag of strings } \text{term2})
\]

\[
\begin{align*}
\min L &= \min(\text{term1.length, term2.length}) \\
\max L &= \max(\text{term1.length, term2.length}) \\
\text{while } (\text{term1 and term2 notEmpty}) \\
\text{score} &= \max \text{ sim}(s,t), \text{forall } s \text{ in term1 and } t \text{ in term2} \\
\text{totalscore } &= \text{totalscore } + \text{ score} \\
\text{remove } s \text{ from term1 and } t \text{ from term2} \\
\text{denum} &= \max(\min L, \min L + \log(\max L - \min L + 1)) \\
\text{return } \text{totalscore}/\text{denum}
\end{align*}
\]

For example, given the labels “purchaserInfo” and “buyer”, we see that
\(\text{sim}(\text{“purchaser”,”buyer”})=1.0\). Therefore we calculate
\[
\text{match}(\{\text{“PURCHASER”,”INFO”}\},\{\text{“BUYER”}\})=1/(1+0.14)=0.87
\]
where the denominator is the number of words (after the label resolution) contained
in the shortest label increased by a logarithmic function.

4.2 Path Matching

Starting from the term similarity matrix, we identify label couplings, i.e. pairs of
terms in the two graphs having a similarity degree higher than a given threshold. For
every label coupling \((A,B)\) we compare all the pairs of attribute-ending paths starting
from \(A\) and \(B\) such that both paths are shorter than a pre-defined parameter. To derive
a similarity measure for a pair of paths \((p,q)\), we formulate a problem of maximum
weighted matching in a bipartite graph with precedence constraints where:

\(\text{─ the nodes } (p_1,...,p_m,q_1,...,q_n) \text{ of the bipartite graph represent the entities (node}
\text{ and edges) of } p \text{ and } q, \text{ and there is a weighted edge for every similarity value in}
\text{the similarity matrix;}\)

\(\text{─ a matching is valid only if for every pair of edges } (p_x,q_i) \text{ and } (p_y,q_j) \text{ included}
in the matching, } p_x \text{ precedes } p_y \text{ in } p \text{ and } q_i \text{ precedes } q_j \text{ in } q, \text{ or } p_x \text{ follows } p_y \text{ in}
\text{}} p \text{ and } q_i \text{ follows } q_j \text{ in } q.\)

Every pair of paths \((p,q)\) that satisfy the following inequation is considerate a
candidate path match:

\[
\frac{\text{match}(p,q)}{\min L} \geq \text{th}(\max L, \min L)
\]

where: i) \text{match}(p,q) \text{ is the value of the maximum weighted matching; ii) } \max L \text{ and}
\min L \text{ are the length of the longest and shortest path respectively; iii) } \text{th}(\max L, \min L)
\text{ is a thresholding function depending on the number of entities}
\text{involved in the match and on the difference between the length of the two paths.}
Coming back to our running example, figure 2 shows an example of path matching. The weights of the edges are the similarity values computed in the terminological analysis. Values marked with S are string similarity scores, while L stands for linguistic similarity score.

![Fig. 2. An example of path matching](image)

### 4.3 Path Selection and Composition

In the final step of the algorithm, every candidate path match that is contained in some other candidate path match is discarded. Given two candidate path matches \((p, q)\) and \((p', q')\), we say that \((p', q')\) is contained in \((p, q)\) if \(p'\) is contained in \(p\) and \(q'\) is contained in \(q\). The candidate path matches that are not discarded are proposed to the user for the validation in order to define the final path annotations. When a path annotation is validated, also the terminological annotations between the entities that have been matched during the path matching process are provided. Finally, path annotations can be composed by the user by using a set of operators to define semantic annotation expressions.

The output of this phase could be the following SA expression with the associated terminological annotations:

\[
\{<\text{RS:PO.orderHeader.header.purchaser.purchaserInfo.organization.orgInfo.contactPerson.personInfo.name}>\text{ SPLIT}\ <\text{RO:puchaseOrder.hasBuyer.buyer.hasContactDetails.contactDetails.firstname, RO:puchaseOrder.hasBuyer.buyer.hasContactDetails.contactDetails.surname} >\}
\]

### 5 Conclusions and Future Work

In this paper we presented a method for semantic annotations as general and expressive mappings between a resource schema and a reference ontology. The proposed method is articulated into three steps: i) terminological analysis based on linguistic similarity techniques, ii) path matching based on structural analysis, iii) path composition and semantic annotation expressions building. This research activity is part of the COIN project and aims at supporting semantic reconciliation in the field of enterprise interoperability.
We envision two main future directions. Firstly, to enhance our current prototype exploiting other sources of knowledge, both contained in the models (internal constraints, textual information) or external to them (acronym thesauri, previous matching results). Secondly, we are investigating the automation of the path composition considering the correspondences between patterns of the “semantic mismatches”, that may be detected between the structures, and predefined annotation templates.

References

Abbreviation Expansion In Lexical Annotation of Schema

Maciej Gawinecki

PhD student of the ICT Doctorate School
University of Modena and Reggio Emilia,
Via Vignolese 905, 41100 Modena, Italy
Maciej.Gawinecki@unimore.it

Abstract. Lexical annotation of schema elements can improve effectiveness of schema matching. However, it cannot be applied is those schema elements that contain abbreviations. In this work we address this problem by providing a new technique for abbreviation expansion in the context of schema of structured and semi-structured data.

1 Introduction

The aim of data integration systems is a creation of global schema successfully integrating schemata from different structural (relational) and semi-structural (XML, RDF, OWL) data sources [1]. This process requires understanding of the meaning standing behind the names of schema elements. In this situation lexical annotation helps explicate these meanings by labeling schema elements with concepts from shared models [2]. Shared models (industry standards, vocabularies, taxonomies, and ontologies) provide an agreement on the meaning and intended use of terms, making possible to match together different terms, but with the same or similar meaning. Unfortunately, the shared model may not include abbreviations, while the effectiveness of annotation process heavily suffers from presence of such words in the schema [2].

Current schema integration and annotation systems either do not consider the problem of abbreviation expansion at all or they use non-scalable solution of user-defined dictionary. In this paper we propose an algorithm for automated abbreviation expansion. Abbreviation expansion is an approach of finding a relevant expansion for a given abbreviation. Our contributions are as follows:

– We provide a method for expanding abbreviations using for complementary sources of long form candidates. Different sources of expansions are complementary to each other because they provide expansions for different types of abbreviations.
– We provide evaluation of effectiveness of each source separately and combinations of them to present pros and cons of each source.

Our method is implemented in MOMIS (Mediator envirOnment for Multiple Information Sources) system [1] where lexical annotation is done with respect
Fig. 1. Graph representation of two schemata with elements containing abbreviations and CNs: (a) relational database schema, (b) XML schema.

The paper is organized as follows. In Section 2 we define the problem of abbreviation expansion occurring in schema labels. Section 3 describes the proposed algorithm for expanding abbreviations in schema elements names. Section 4 presents the current state of the art in the field of abbreviation expansion. Finally in Section 4 we provide evaluation of the proposed solution and we conclude with proposals of future work in Section 5.

2 Problem definition

Element names represent an important source of assessing similarity between schema elements. This can be done semantically by comparison of their meanings.

Definition 1 Lexical annotation of a schema label is the explicit assignment of its meaning w.r.t. a thesaurus.

Definition 2 An abbreviation (short form) is a shortened form of a word or phrase (long form), that consists of one or more letters taken from the long form.

Figure 1 presents two schemata to be integrated, containing many labels with non-dictionary abbreviations, e.g. ‘PO’ (standing for “Purchase Order”), QTY
(“Quantity”). These labels cannot be directly annotated, because they do not have an entry in WN, while their corresponding expansions may be easily recognized by WN. Therefore it is necessary to identify and expand all abbreviations appearing in schema labels before performing lexical annotation.

**Definition 3** Abbreviation identification is the task of determining whether a given word has been used for abbreviation in the given context.

Very often legitimate English words are used for abbreviations in the schema context. For instance, ‘id’ is a dictionary word in WN standing, among many others, for “primitive instincts and energies underlying all psychic activity”, while in prevalent number of analyzed schemata it is a short form of ‘identifier’ (“a symbol that establishes the identity of the one bearing it”).

**Definition 4** Abbreviation expansion is the task of finding a relevant expansion for a given identified abbreviation.

There may be several possible long forms for a given short form. For instance, Abbreviations.com online dictionary provides 56 different expansions for abbreviation ‘PO’, including: “Post Office”, “Purchase Order”, “Parents Of” and others. Therefore, automatic abbreviation expansion can be split into two sub-problems: (a) searching for potential long forms (expansions) for a given short form; and (b) selecting the most appropriate long form from the set of potential long form candidates.

3 Proposed solution for automatic abbreviation identification and expansion

Dealing with abbreviations appearing in a schema label involves two operations: (1) identifying whether it is a short form or it contains short forms and then (2) providing relevant long forms for identified short forms. These operations should be performed for each schema label as it has been described on Figure 2. In the following subsections we describe how each operation is realized.

3.1 Abbreviation identification

We consider a word to be an abbreviation if it belongs to the list of standard abbreviations or it is not a dictionary word. The list of well-known standard abbreviations is employed here to reduce the number of false negatives caused by legitimate English words used for abbreviations.

Note, that non-dictionary labels can in fact consist of more then one word. We are tokenizing them using one the pre-existing approaches [4].

1 Please also note that, besides abbreviations a schema may contain other non-dictionary terms such as: multi-word terms (including compound nouns), mispellings, numbers – ‘3D’ and foreign language words that might affect the process of identifying abbreviations, but we are not dealing with them in this approach.
for each schema $S$ iterate over classes and their direct attributes using BFS order
for the label $l$ of each class/attribute in $S$:
if $l$ is a standard abbreviation then
\[ l_{fk} := \text{selectLongForm}(l) \]
else if $l$ is not a dictionary word then
  tokenize $l$ into words $(w_i)_i$
  for each word $w_i$ in $(w_i)_i$
    if $w_i$ is a standard abbreviation or not a dictionary word then
      \[ l_{fk} := \text{selectLongForm}(w_i) \]
    end if
  end for
end if
end for
end for

Fig. 2. Proposed solution.

3.2 Observations on abbreviation expansion
A schema can contain both standard and ad hoc abbreviations. However, only the standard abbreviations can be found in user-defined and online abbreviation dictionaries as they: either (a) denote important and repeating domain concepts or (b) are standard suffix/prefix words used to describe how a value of a given schema element is represented\(^2\). For instance 'Ind' (Indicator) defines a list of exactly two mutually exclusive Boolean values that expresses the only possible states of a schema element, like in ‘FragileInd’ on Figure 1. On the contrary, ad hoc abbreviations are mainly created to save space, from phrases that would not be abbreviated in a normal context [5, 6].

To observe how specifically ad hoc abbreviations can be handled automatically we analyzed short forms and their corresponding long forms in several open-source schemata. Based on our manual inspection, we found two sources relevant for finding possible long form candidates:

- context of short form occurrence, as it is common practice to prefix column an attribute name with short form of a class name, for instance ‘recentchanges’ table contains ‘rc_user’ and ‘rc_params’, while the ‘FIBranchID’ attribute is an element of ‘FinancialInstitutionType’ complex type.
- a complementary schema with which we would integrate with inspected schema; for instance when integrating two schemata from Figure 1, we found that the XML schema contains abbreviations (‘PO’, ‘uom’), which maybe expanded with long forms from relational database schema (‘Purchase Order’, ‘unit Of Measure’).

3.3 Proposed algorithm for abbreviation expansion
To handle well-known standard abbreviations the algorithm uses an online abbreviation dictionary (OD) and user-defined dictionary (UD), with abbreviations

typical for schema labels). The ad hoc abbreviations are expanded using context (C) and complementary schema (CS) as sources of long form candidates. The syntax of a short form itself does not provide any mean for distinguishing between ad hoc and standard abbreviations. Therefore, we are not able to choose in advance these sources which are more relevant for expansion of a certain short form. However, the context and complementary schema can be generally considered as the most relevant sources of long form candidates, because they closely reflect the intention of a schema designer.

For each identified abbreviation the algorithm inquires all four sources for long form candidates, scores candidates according to the relevance of the source, combines scores of repeating long forms and choose the top-scored one. The whole process is shown in Figure 3. Let us now describe how scoring technique is realized.

Technically, for each identified short form $sf$ the algorithm generates a list of long form candidates: $(lf_i; score(lf_i))$, obtained from all the sources. The list is sorted descendingly according to the $score(lf_i) \in [0, 1]$ of a long form candidate $lf_i$. The algorithm selects the top-scored long form candidate from the list. If the list is empty, then the original short form is preserved. The score of $lf_i$ is computed by combining scores from the single sources:

$$score(lf_i) = \alpha_{UD} \cdot score_{UD}(lf_i) + \alpha_{CS} \cdot score_{CS}(lf_i) + \alpha_{C} \cdot score_{C}(lf_i) + \alpha_{OD} \cdot score_{OD}(lf_i)$$

where $\alpha_{UD}+\alpha_{CS}+\alpha_{C}+\alpha_{OD} = 1$ are weights corresponding to different relevance of the sources.

For user-defined dictionary, context and complementary schema sources the score of $lf_i$ is 1, if the $lf_i$ is found in the given source or 0 – otherwise. For online dictionary we describe scoring procedure below.

The context of an occurrence of the short form $sf_i$ is defined as a name of the class to which the attribute containing a short form $sf_i$ belongs. The context is retrieved for possible long form candidates using the four abbreviation patterns

**Fig. 3.** Proposed procedure for selecting long form for the given short form.
proposed in [7]. The abbreviations patterns are, in practice, regular expression
created from characters of a short form.

The labels in the schema complementary to the schema in which $sf$ appears
are retrieved for matching long form candidates using the same abbreviation
patterns as in the context. Only the first matching candidate is considered.

Scoring of long form candidates from online dictionary is more complex, as
a dictionary may suggest more than one long form for a given short form. For
this purpose we propose disambiguation technique based on two factors: (a) the
number of domains a given long form shares with both schemata and (b) its
popularity in these domains. We assume information about the domain of a long
form and its popularity is given by the online dictionary.

Domain-based disambiguation is a popular approach for the problem of word-
sense disambiguation [8, 2]. The intuition behind this approach is that only mean-
ings of a word that belongs to the same domains that context describes, are
relevant. Please note that in our adaptation of this method we are selecting not
a meaning, but a long form from a set of long form candidates. Practically, we
may define score of long form candidate $score_{OD}(lf_i)$ as follows:

\[
score_{OD}(lf_i) = \frac{1}{P_{schema}} \sum_{d \in CD(lf_i, schemata)} p(lf_i, d)
\]

\[
P_{schema} = \sum_{d \in CD(lf_i, schemata)} \sum_{i} p(lf_i, d),
\]

\[
CD(lf_i, schemata) = D(lf_i) \cap D(schemata)
\]

where $D(schemata)$ is a list of prevalent WN Domains\(^3\) associated with schemata
to integrate (obtained using the algorithm from [2]). Computation of $CD(lf_i, schemata)$
—— the intersection of prevalent domains and domains associated with long form
$lf_i$ — involves the mapping between the categorization system of an online abbre-
viation dictionary and WN Domains classification. If there is no shared domain
for long form candidates, then score is computed as a general popularity of a
long form candidate.

There can be more than one online dictionary entry describing the same long
form $lf_i$, but in different domains. Therefore, the entry can be modeled as a
combination of a long form $lf_i$ and a domain $d_{i,k} \in D(lf_i)$ in which it appears
with the associated popularity. Formally, we define the $t$-th dictionary entry in
the following form: $<e_t, p(e_t)>$, where $e_t = <lf_i; d_{i,k}>$ and $d_{i,k} \in D(lf_i)$ is the
$k$-th domain in the set of domains $(D(lf_i))$, in which the long form $lf_i$ appears.
The popularity $p(e_t)$ is not explicitly reported by the considered dictionary but
can be easily estimated from the order of descending popularity in respect to
which entries are returned by the dictionary. Thus we are able to calculate $p(e_t)$
using the following induction: $p(e_{t+1}) = p(e_t)/\kappa$, $p(e_1) = 1$, where $\kappa > 1$ is an
experimentally defined factor\(^4\).

\(^3\) http://wndomains.itc.it/wordnetdomains.html
\(^4\) In experiments we successfully use $\kappa := 1.2$
Example 1. Let us assume the algorithm expands abbreviations in two schemata presented in Figure 1. The algorithm discovers the following abbreviations (given with number of their occurrences): ‘PO’ (5), ‘Info’ (2), ‘WHSE’ (1), ‘UOM’ (1), ‘QTY’ (1), ‘NO’ (1), ‘Nbr’ (1), ‘Ind’ (1). When expanding the first abbreviation in ‘PODeliveryInfo’ element the algorithm receives the following information from the particular sources:

- from the online dictionary ($\alpha_{OD} = 0.05$) – only two top-scored expansions: ‘Purchase Order’ (0.52, sharing commerce domain) and ‘Parents Of’ (0.25, sharing ‘sociology’ domain),
- from the context ($\alpha_{C} = 0.25$): ‘Purchase Order’ (1.0, extracted from the name of the containing element),
- from the complementary schema ($\alpha_{SC} = 0.10$): ‘Purchase Order’ (1.0).

The remaining source does not return any long form candidate. Next, the algorithm merges lists of proposed candidates into a single one: [Purchase Order (0.38), Parents Of (0.013)]. Particularly, the score for the top-scored expansion has been computed in the following way: $0.38 \approx 0.05 \cdot 0.52 + 0.25 \cdot 1.0 + 0.10 \cdot 1.0$.

4 Related work

The problem of abbreviation expansion has received much attention in different areas such as: machine translation, information extraction, information retrieval and software code maintenance.

Many techniques are based on the observation that in documents the short forms and their long forms usually occur together in patterns [9, 10]. As there are no such explicit patterns in data schemata, we adopted abbreviations patterns used for the similar problem of expanding abbreviated variable identifiers in program source codes [7]. Furthermore, in the literature selecting the most relevant long form from a single source is made with respect to different factors such as: inverted frequency of a long form in both domain-specific and general corpora [11], size of document scope in which both a short form and a matching long form appear [7] or syntactic similarity between a long form and a short form (e.g. whether a short form has been created by removing internal vowels from the given long form) [6]. Application of these approaches in our context is a subject of future research, as for now both context and complementary schema are returning only the first discovered match.

It has been observed that the presence of abbreviations in schema elements labels may affect the quality of elements name matching and requires additional techniques to deal with [12]. Surprisingly, current schema integration systems either do not consider the problem of abbreviation expansion at all or solve it in non-scalable way by inclusion of a simple user-defined abbreviation dictionary (e.g. Cupid [13], COMA/COMA++ [14]). Lack of scalability comes from the fact that: (a) the vocabulary evolves over the time and it is necessary to maintain the table of abbreviations and (b) the same abbreviations can have different
expansions depending on the domain. Moreover, this approach still requires an intervention of a schema/domain expert.

According to our knowledge, the work of Ratinov and Gudes [5] is the only one that attacks the problem of abbreviations in the context of data integration. Their technique focuses (a) on supervised learning of abbreviation patterns to deal with ad hoc abbreviations and (b) on using external corpus as a source of matching long forms. However, they do not report any details nor evaluation of possible disambiguation of candidates. We claim that ad hoc abbreviations can be handled using abbreviation patterns proposed in [7] and context and complementary schema as more relevant sources of long forms.

5 Evaluation

We implemented our method for abbreviation identification and expansion in the MOMIS system [1]. The implementation uses the following external information: (a) WordNet 3.0, (b) a list and a dictionary of standard abbreviations, and (c) Abbreviations.com online abbreviation dictionary. For domain-based disambiguation of long forms we created a mapping between the category hierarchy of this dictionary and WordNet Domains. We tested the performance of our method over the two relational schemata of the well known Almagam integration benchmark for bibliographic data [16]. Table 1 summarizes the test schemata features that are particularly suitable for the test. We acknowledge that one of the schemata contains four elements with unknown meaning and thus not considered in the experiments. We performed our evaluation with the following questions in mind

1. What is the effectiveness of abbreviation identification method?
2. What is the effectiveness of each source in providing correct long forms?
3. How effective are external sources (user-defined dictionary and online abbreviation dictionary) in comparison to internal sources (context and complementary schema) in dealing with different types of abbreviations (ad hoc and standard)?
4. How can we assign relevance weights to each source to limit their individual weaknesses and take advantage of their strengths when combining them together?

To answer to these questions we performed two groups of experiments. One group for evaluation of abbreviation identification method (question 1) and one for abbreviations expansion method (questions 2-4). We evaluated these methods separately, because errors produced during the first one (incorrectly tokenized labels and identified abbreviations) gives a different input for the second method and thus may impact its effectiveness.

[15] describes criteria and our survey of selection online abbreviation dictionary and procedure of creation of such a mapping. All mentioned information is more accessible at: http://www.ibspan.waw.pl/~gawinec/abbr/abbr.html.
### Table 1. Characteristics of test schemata.

<table>
<thead>
<tr>
<th></th>
<th>Number of Labels</th>
<th>Non-dictionary words</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema 1</td>
<td>117</td>
<td>66</td>
<td>24</td>
</tr>
<tr>
<td>Schema 2</td>
<td>51</td>
<td>28</td>
<td>28</td>
</tr>
</tbody>
</table>

5.1 Evaluating abbreviation identification

In this group of experiments we measured the correctness of abbreviation identification. We consider a label correctly identified if — in respect to manual identification: (a) the label as a whole has been correctly identified, and (b) the label has been correctly tokenized, and (c) all abbreviations in the label have been identified. During manual identification multi-word abbreviations (abbreviations shortening more than one word) were tokenized into abbreviations representing single words (e.g. ‘PID’ standing for “Publication Identifier” was tokenized into ‘P’ and ‘ID’), with exceptions to standard abbreviations (such as ‘ISBN’), that were left untokenized.

For automated tokenization we use two competing methods: simple — based on camel case and punctuation, and greedy — handling multi-word names without clearly defined word boundaries, e.g. ‘CATALOGGROUP’. The latter iteratively looks for the biggest prefixing/suffixing dictionary words and user-defined abbreviations [4]. Therefore, we evaluated identification method in three variants depending on tokenization method used: (1) ST: simple, (2) GT/WN: greedy with WN and (3) GT/Ispell: greedy with Ispell English words list\(^6\) as a dictionary.

The ST reaches nearly the same correctness (0.92) as GT/Ispell (0.93), because the schemata contain relatively few labels with unclearly undefined word boundaries (e.g. ‘bktitle’). On the contrary, the GT/WN remains far away from its competitors (0.70), because WN contains many short abbreviations (e.g. ‘auth’ is tokenized to: ‘au’, ‘th’). All three variants of the method have been affected by the usage of legitimate English words for abbreviations, that were not defined on a list of standard abbreviations.

5.2 Evaluating abbreviation expansion

In this group of experiments we measured the correctness of abbreviations expansion method. We performed 7 experiments: 4 measuring correctness of each single source of long forms, 2 two experiments for evaluation of external and internal sources and 1 evaluating correctness of all sources combined together. An identified abbreviation was considered to be correctly expanded with respect to the manual expansion. The input for each experiment was manually tokenized and identified. The results of experiments are shown on Figure 4.

It can be observed that internal sources of long forms (context and complementary schema) provide correct long forms complementary to external sources.

---

(user-defined dictionary and online abbreviation dictionary). User-defined dictionary provided correct expansions in 42% of abbreviations, but it still does not handle with ad hoc abbreviations such as ‘Auth’ (‘Author’), ‘coll’ (‘collection’) or ‘bk’ (‘book’), where internal sources behave better. Finally, online dictionary provided 19% of correct results, including well-known abbreviations such as ‘ISBN’ (“International Standard Serial Number”) and ‘ISSN’ (“International Standard Serial Number”). The relevant long forms were chosen among many others provided by the online dictionary, because they share the highest number of domains with both schemata, namely: telecommunication and publishing.

When combining external sources together we considered user-defined dictionary (UD) as a more relevant source then online abbreviation dictionary (OD), because it describes expansions more typical for schemata. When combining internal sources we gave more importance to context (C) source over complementary schema (CS), because context source reflects better user intention about a given schema. Finally, when combining external and internal sources, we found that: (1) complementary schema may provide less relevant long forms then user-defined dictionary, (2) online dictionary is considered the last chance source of long forms, when no other source provide relevant long forms. Following this guidelines we experimentally setup the final weights for all the sources: \( \alpha_{UD} := 0.40, \alpha_{C} := 0.30, \alpha_{CS} := 0.20, \alpha_{OD} := 0.10 \). For such weights the whole abbreviation expansion provided 83% correct expansions for manually tokenized and identified input.

6 Conclusion and Future Work

Abbreviations appearing in schema labels are serious obstacle for correct lexical annotation of schema elements. Surprisingly, current data integration systems either ignores the problem of abbreviations or handle it with a usage of non-scalable user-defined dictionary. To overcome this problem we presented a method for identifying and expanding abbreviations appearing in schema element names. We proposed a usage of four sources of expansions: context of abbreviation occurrence in schema, a schema complementary to the annotated one, an online abbreviation dictionary and user-defined dictionary of standard abbreviations. We have experimentally shown that these sources are complementary in recognizing particular types of abbreviations and their combination can provide high correctness for abbreviation identification and expansion.

For our future work we plan to (1) evaluate the method on larger dataset and to (2) estimate the impact of abbreviation identification on performance of abbreviation expansion. As a extension of our method we also consider usage of schema documentation (inline, external, structural and non-structural) as an alternative source of long form candidates.

7 Surprisingly, Abbreviations.com dictionary does not contain popular ‘Conference’ expansion for ‘Conf’ abbreviation, while its competitor – AcronymFinder.com – does.
7 Acknowledgements

The author would like to thank for valuable comments to Sonia Bergamaschi and Serena Sorrentino.

References

Table: Evaluation of abbreviation expansion for: (a) each long form sources separately, (b) internal sources (context and complementary schema) and external sources (user-defined dictionary and online abbreviation dictionary) and (c) all sources combined together. The letters in cells stands for: Y – expanded correctly; N – expanded incorrectly.

<table>
<thead>
<tr>
<th>Label short form</th>
<th>user-def. dict.</th>
<th>online dict.</th>
<th>compl. schema</th>
<th>context</th>
<th>external sources</th>
<th>internal sources</th>
<th>all sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct expansions in %</td>
<td>20</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>29</td>
<td>12</td>
<td>40</td>
</tr>
<tr>
<td>AID.A</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>annode.anmode</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>AuthID.Auth</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>bkfile.bk</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>collID.coll</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>CountryInfo.Conf</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>CountryPub.Pub</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>DIF.D</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>inst.inst</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ISSN.ISBN</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>ISSN.ISSN</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>JournalAnn.Ann</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>LD.L</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>loc.loc</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PID.P</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>RID.R</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>techreport.tech</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>TitleExt.Ext</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>vol.vol</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>AbstractInd.ind</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>AccessionNum.Num</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>AID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>articleID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>AuthID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>bookID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>collID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>CountryInfo.Info</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ContractNum.Num</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>DID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>LID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>num.num</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>NumRef.Num</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>NumRef.Ref</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>PID.P</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>RID.R</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>techID.ID</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Fig. 4. Evaluation of abbreviation expansion for: (a) each long form sources separately, (b) internal sources (context and complementary schema) and external sources (user-defined dictionary and online abbreviation dictionary) and (c) all sources combined together. The letters in cells stands for: Y – expanded correctly; N – expanded incorrectly.
Service-Finder:
First Steps toward the realization of Web Service Discovery at Web Scale

Saartje Brockmans¹, Irene Celino¹, Dario Cerizza¹, Emanuele Della Valle¹ ², Michael Erdmann⁴, Adam Funk⁵, Holger Lausen³, Wolfgang Schoch⁴, Nathalie Steinmetz³, and Andrea Turati¹

¹ CEFRIEL – ICT Institute Politecnico di Milano, Milano, Italy name.surname@cefriel.it
² Politecnico di Milano, Milano, Italy name.surname@polimi.it
³ seekda, Innsbruck, Austria name.surname@seekda.com
⁴ ontoprise, Karlsruhe, Germany surname@ontoprise.de
⁵ University of Sheffield, Sheffield, United Kingdom name-first-letter.surname@dcs.shef.ac.uk

Abstract. The Web is moving from a collection of static documents to one of Web Services. Search engines provide fast and easy access to existing Web pages, however up till now no comprehensive solution exists which provides a similar easy and scalable access to Web Services. The European research project Service-Finder is developed a first version of a portal for service discovery where service related information from heterogeneous sources is automatically integrated in a coherent semantic model to allow effective discovery and to collect user contribution in a Web 2.0 fashion.

Keywords: Web Services, Web Service Discovery, Semantic Web, Crawling, Automatic Annotation, Semantic Matching, Web 2.0

1 Motivation

The Web is moving from a collection of static documents to a collection of services. For realizing service interchange in a business to business setting, the service-oriented architecture together with the Web Services technology are widely seen as the most promising foundation. As a result, considerable attention has been given in both research and industry to Web Services and related technologies.

Within a Web of services in general, and in any service-oriented-architecture (SOA) in particular, the discovery of services is the essential building block for creating and utilizing dynamically created applications. However, current technologies only provide means to describe service interfaces on a syntactical level, providing only limited automation support. Several previous research projects (just to mention SWWS, DIP, ASG, InfraWebs or NeP4B) proved that descriptions on a semantic level (e.g., OWL-S [5], WSMO [4], MicroWSMO [3] and
SAWSDL [2]) can be used to enable precise discovery of services. However, they are not yet widely deployed. Moreover, existing solutions only cater for scenarios with a limited number of participants.

The Service-Finder project\(^6\) is addressing the problem of utilizing the Web Service technology for a wider audience by realizing a comprehensive framework for Discovery by making Web Services available to potential consumers similarly, to how current search engines do for content pages\(^7\).

An essential, but mostly unaddressed problem is the creation of such semantic descriptions of Web Services. The TAO project is the only ongoing project addressing the issue of semi-automatic creation of semantic service descriptions, but it is focused on the specific area of legacy applications and pre-supposes the existence of large documentation of the underlying software. Service-Finder aims to offer automatic creation of service descriptions for a different range of Services (all the publicly available services) and to enable service consumers, not just service providers, to enrich the semantic service descriptions following a typical contribution-based approach in a Web 2.0 fashion.

The paper is structured as follow. Section 2 provides an overview of the most relevant state of the art in Web Service discovery and Semantic Web Service discovery. Section 3 details the architecture of the Service-Finder portal describing the role of each internal component. Section 4 describes the demonstration portal currently available. Section 5 provides conclusions and an overview of the planned future works.

## 2 State of the Art

Existing solutions for Service Discovery include UDDI, a standard that allows programmatically publishing and retrieving a set of structured information belonging to a Web Service. Several companies have operated public UDDI repositories, however due to several shortcomings of the approach such as complicated registration, missing monitoring facilities, its success was limited and only a few repositories are still publicly available. At the same time a number of Portals dedicated to providing a repository of services have appeared. However, all of them rely on a manual registration and review process, which implies limited coverage as well as inherently outdated information. Alternatively, one can use the classical search engines; however, they do not provide effective means to identify Web Services. For now, there exists no standardized file suffix, such that a query like “filetype:wsdl” does not match all service descriptions (e.g. the wsdl description Microsoft Services will have the ending ".asmx?wsdl”). Moreover, a standard search engine does not make any pre-filtering based on availability and other service-related parameters; their retrieval model is optimized for finding content and not dynamic services.

---

\(^6\) [http://www.service-finder.eu](http://www.service-finder.eu)

\(^7\) The alpha release of the Service-Finder Portal is available at [http://demo.service-finder.eu](http://demo.service-finder.eu)
Current Web Service standards are limited to specify Service Requests as keywords and Service Offers as interface structures (WSDLs). Essentially Web Service technologies only allow describing Services at a syntactical level, this prevents the dynamic discovery and reconfiguration of Web Services to adapt automatically to changes, e.g. that a provider is going off-line or a cheaper provider entering the market. In this context, Semantic Web technologies can provide semantic descriptions that go beyond the syntactic descriptions of Web Services offered by current technologies, describing formally and with well-defined semantics the requester goals and the Web Service capabilities.

Several Description Frameworks (WSMO, OWL-S, SAWSDL) provide the semantic descriptions needed for dynamic location of Web Services that fulfill a given request. They allow, given a goal, the dynamic location of a Web Service. The focus so far has been to capture the functionality of the service in a sufficient level of detail, to ground the accurate matching of requester goals and Web Services. Little research has been conducted on the aspect of how to obtain those descriptions and on other aspects of a service description such as its non-functional properties. Moreover, the approaches developed so far are based on two assumptions: the discovery engine knows all the Web Service descriptions and, the Web Service descriptions are completely correct because an expert draws them up. However, in a Web scenario these assumptions are unreasonable. First, the services are located in different locations that could be unknown a priori, and sometimes they are not available. Secondly, most of the services are described only syntactically, so that their semantics have to be deduced from other available sources, such as service’s documentation, Web pages and so on. For this reason, we cannot assume the correctness of the description of a service. Due to these reasons, current approaches for discovering services are not suitable in a Web scenario.

3 The Architecture

Service-Finder overcomes current shortcomings of current Web Service portals and search engines by creating a public Web portal capable of:

- Employing automated methods to gather WSDLs and all related resources such as wikis, blogs or any webpage in which useful information are given;
- Leveraging semi-automatic means to create semantic service descriptions of a Web Service out of the information gathered on the Web;
- Describing and indexing the aggregated information in semantic models to allow matchmaking-based reasoning and to enable fast searches;
- Providing a Web 2.0 portal to support users in searching and browsing for Web Services, and facilitating community feedbacks to improve semantic annotations;
- Giving recommendations to users by tracking their behavior.

Figure 1 shows the general architecture of the Service-Finder Portal accessible at http://demo.service-finder.eu. The current components and flows
of data in Service-Finder can be summarized using the continuous arrows. The available services and related information are obtained by crawling the Web (Service Crawler - SC); then, the data gathered is enriched in an analysis step (Automatic Service Annotator - AA) accordingly to the ontologies (Service-Finder and categories ontologies) that models a coherent semantic model. The unified representation of each crawled and annotated service is stored and indexed (Conceptual Indexer and Matcher - CIM) to allow effective retrieval for Web Service search and user contributions on the Web 2.0 style (Service-Finder Portal Interface - SFP). In addition we gather data by analyzing the users' behavioral patterns when they search for services, and use these for Clustering (Cluster Engine - CE).

The seekda search engine\(^8\) is used as an external component delivering four functionalities improving the service-finder portal such as: provide relevant seed URLs to start crawling, supply a rank score for each service (based on various factors such as availability, number of views, online usage, number of inlinks, etc.), provide availability information (over time as graph), and other additional features such as: registration of new services and invocation of services.

Fig. 1. Overview of the Service-Finder components and dataflow

---

\(^8\) [http://www.seekda.com](http://www.seekda.com)
The dashed arrows in Figure 1 refer to the changes that we intend to implement for the beta release of the Service-Finder portal. While the components in the architecture remain the same, the dataflows are supplemented by some new ones: The CIM will forward the user feedback data to the AA, such that the AA can use the users’ re-categorizations of services as training data to improve the subsequent automatic category annotation. The CE will provide cluster data to the CIM, which will be used by the latter to provide recommendations to the SCP.

The following subsections describe briefly the contribute of each component.

3.1 Ontologies

The purposes of the ontologies is to provide a coherent model to collect and manage all the information related to the crawled Web Services.

The Service-Finder ontology focuses in modeling the information related to the core concepts of Service-Finder. The central entity is a service that is associated with several other entities. Several “pieces” of additional information can be gathered around a service. This additional information can stem from the user community as well as from the automatic annotator, respectively the Web. Information extracted by the annotator is obtained by analyzing several documents that can be related to a service. In addition to the service and the information and documents connected to it, we model the provider as the entity that operates a specific service, as well as the user that searches and eventually uses services.

The purpose of the Service Category Ontology is to provide a coarse grained classification to allow users to get an overview of available services. We provide a set of categories based on the experiences gathered during the initial services crawling and operation of the service search-engine at seekda.com. Thus we believe that the chosen categories cover the major domains in which currently publicly accessible services are available.

3.2 Service Crawler (SC)

The Service Crawler is responsible for gathering data from the Web. It pursues a focused crawl of the Web and only forwards relevant data to subsequent components for analyze, index and display purposes. It identifies both services and relevant service-related information on the Web. Around the located services it gathers as much related information as possible, i.e. information from external sources as from the provider’s service description, documents or Web pages pointing to that description or to the service, etc.

3.3 Automatic Annotator (AA)

The Automatic Annotator processes the data from the Service Crawler and produces semantic analysis for the Conceptual Indexer and Matcher. We adapted
natural language processing and information extraction techniques (using the GATE platform\(^9\) and bespoke software for Service-Finder) to obtain relevant information from the textual documents (HTML and PDF) relating to the web services and to compare them with the relevant pre-processed WSDL files. The Automatic Annotator processes each provider’s set of documents independently, extract information in RDF-XML, and forwards it to the Conceptual Indexer and Matcher, along with compressed archives of the plain text content of interesting documents for keyword indexing. We aim to evaluate the quality of the results by comparing them with a manually annotated sample of the output data and by examining the results of queries made to the Conceptual Indexer and Matcher.

3.4 Conceptual Indexer and Matcher (CIM)

One key principle to realize the goal of effective Web Service discovery is the actual matchmaking process, i.e. retrieving good services for a given user request, related to crawled services and the automatic annotations.

The Conceptual Indexing and Matchmaker is built over the OntoBroker reasoner that internally is based over F-Logic. Since the other components exchange RDF data, to bridge the gap between F-logic and RDF, a particular query language has been developed for the matchmaking component. Moreover, as the matchmaker combines syntactic and semantic retrieval techniques, it takes advantages of both the speed of syntactic search and the expressiveness of semantic relations.

3.5 Service-Finder Portal (SFP)

The main point of access for the final users to the results of the project is the Service-Finder Interface, a Web portal through which it is possible to search and browse for the services crawled, annotated and indexed by the other components. The current version of the Service-Finder Interface\(^10\), provides the basic functionalities to find services, navigate through them, add rating and simple annotations and test the services. The design and implementation of the Service-Finder Interface took into account the requirements of the users, the trends in Web 2.0 and interaction design fields and the research results in applying Semantic Web technologies to the development of portals and Web interfaces via the employment of the STAR:chart framework \(^11\). The logs of users interaction with the Service-Finder Interface are the basis for the work of the Cluster Engine.

3.6 Cluster Engine (CE)

Cluster Engine is responsible for providing recommendations to users meanwhile they are interacting with the Service-Finder portal. The literature presents two

\(^9\) [http://gate.ac.uk/](http://gate.ac.uk/)
\(^10\) Accessible on the Web at [http://demo.service-finder.eu](http://demo.service-finder.eu)
\(^11\) See also [http://swa.cefriel.it/STAR](http://swa.cefriel.it/STAR)
different approaches for making recommendations: content-based filtering and collaborative filtering. The first one aims at recommending items that are most similar to the ones that the user already appreciated, while collaborative filtering tries to derive similarities between users in order to recommend items that similar users tend to prefer. We chose to base the Cluster Engine over the collaborative filtering approach since it better fits the collaborative typical of Web 2.0 applications and it also foster the serendipity principle, which can help users in finding services. The Cluster Engine monitors users’ behaviour in interacting with the portal to derive users’ profiles and, then, compares all users’ profiles in order to estimate similarities between users. Exploiting such similarities, it provides users with personalized recommendations.

4 Service-Finder at Work

The Alpha release of the Service-Finder portal is freely accessible at http://demo.service-finder.eu where users can found a brief explanation of the portal.

By clicking on the search button, users access the search page. The portal provides three types of search:

- **Keyword Search** enables users to search services by keywords that are syntactically matched against the textual parts of the service model (service name, description, name of its operations) and the related resources (web pages, PDFs).
- **Category Search** supports users in browsing the taxonomy tree and searching for services. The category classification is performed by the Automatic Annotator and is refined thanks to the users contributions.
- **Tag Search** enables users to search for services using the tags that has been associated by other users.

At the time of writing this paper, Service-Finder crawled and indexed more than 25,000 services and about 200,000 related web pages. The results of searches are displayed to the users using standard pagination techniques for long lists.

When users select one specific service, all the details related to that service are shown organized in groups: generic service information; details about the operations with the possibility to invoke the remote service using a tool provided by seekda.com; non-functional aspects of services such as their availability; ratings and comments and, finally, the list of related documents.

Further to the service pages, the portal shows information related to the providers that aggregate services hosted by the same entity. Users can also contribute in suggesting new Web Services by providing the URL of their WSDL.

We envision that the achievements of the project are exploitable in different scenarios:

- Web 2.0 developers may use Service-Finder for building mash-ups of lightweight services.
ICT Companies engaged in Application Integration scenarios in large enterprises may actively use Service-Finder to look for "address validation", "fraud detection", "credit worthiness", etc. The Service-Finder technologies may be sold as an Appliance capable of providing services search capabilities at very low installation and maintenance costs, due to the automated crawling and annotation features.

5 Conclusion and Future Works

Service-Finder provided a first version of a portal able to bring Web Service discovery to a Web scale. The alpha version of the Service-Finder Portal is live at http://demo.service-finder.eu and it’s gathering feedbacks and contributions from Web users. Such feedbacks will work as inputs for the internal technical activities in order to provide an enhanced version of the portal.

Future plans on the project include improvements on the quality of information extraction (by using machine learning techniques to treat feedbacks coming from users’ annotations) and adding new features typical of Web 2.0 approaches (facet browsing, wiki, ...).

6 Acknowledgement

The work described in this paper is partially supported by the European project Service-Finder (FP7-215876). For further information, please visit http://www.service-finder.eu or contact the coordinator of the Service-Finder Project: emanuele.dellavalle@cefriel.it

References

New approach for user centric ontology design

Author1, Author2

Abstract. Ontologies are an important enabling technology for interoperability, but their acceptance and impact is limited by the fact that ontology design is complex and no tools targeted at non-experts are available. In order to bridge the ‘ontology gap’ we are developing a tool-supported methodology that enables non-expert users to create formalized knowledge. Our methodology and tool implement novel approaches for ontology design in a concise and user-friendly way.

1 Introduction

Ontologies are an important enabling technology for interoperability [1]. The acceptance and impact of ontologies is hindered by the fact that they are a technology only available to trained experts. This applies to their use in the context of interoperability, as well as the vision of the semantic web [2] as a whole. To turn this vision into reality a critical mass of formalized knowledge needs to be available. At the moment ontologies are not used by the general public, in contrast to the so-called Web 2.0 [3], which recently gained a lot of attention. To facilitate the acceptance of ontologies and make the semantic web real, the creation of ontologies needs to be simplified and ultimately should become as simple as creating web pages and writing weblogs. In our earlier work we presented criteria for the user-friendliness of ontology design methodologies [4], conducted a survey that assessed sixteen methodologies with respect to our criteria and pointed out weaknesses in existing methodologies [5]. In this paper we discuss approaches to make ontologies usable by non-experts and describe our approach to a tool supported methodology for reaching this goal. Our methodology is inspired by the lessons learned from our survey and the shortcomings of existing approaches. Then we describe our graphical editor prototype, which implements this methodology in a concise and user-friendly way. Eventually we give a conclusion and an outlook on our future work.

2 Related work

Existing tools, like Protégé or Altova SemanticWorks, are geared towards knowledge engineers or software engineers [6]. They are not usable without a considerable amount of training. Semantic wikis (see [7] for an overview, [8] for a comparison) aim at simplifying the usage of ontologies by utilizing the wiki-concept
many people are familiar with. On the downside semantic wikis are largely fo-
cused on the annotation of existing knowledge, not on the creation of ontologies
from scratch. Another approach is ontology editing using controlled vocabularies
in approaches like CLONE [9], ACE [10] and ROO [11]. An inherent drawback
of these approaches is their language-dependency and the fact that average users
are not used to this kind of systems. Therefore we developed a graphical editor
prototype utilizing generic metaphors that can be understood by a wide range of
people and the usage of drag&drop minimizing the need for textual interaction.
We envision three use cases for our approach:

1. Average users without a background in computer science, who want to con-
tribute knowledge on their subjects of interest to the semantic web.
2. Domain-experts who create formalized knowledge that is to be used in a
software system. Traditionally such ontologies would be created by ontol-
ogy engineers interviewing domain experts in an iterative process. Enabling
domain experts to create initial versions of those ontologies themselves po-
tentially leads to cost benefits when compared to the traditional way.
3. Ontology engineers using the methodology for rapid prototyping of ontolo-
gies.

3 Criteria for user-friendliness

In our previous work [4] we identified criteria for user-friendly ontology design
methodologies. This section will briefly present those criteria (labeled C1 to C13)
and give a short description of the survey we conducted in [5] and its results.

Besides the methodologies themselves the criteria also cover aspects of ontol-
ogy construction tools and the underlying ontology languages. Usage of adequate
terminology (C1), structure (C2), descriptiveness (C3) and transparency (C4)
are criteria for methodologies. Error avoidance (C5), robustness (C6), consis-
tency (C8) and conceptualization flexibility (C11) apply to methodologies as
well as to tools. Lookahead features (C7), the hiding of formality (C9) and the
comprehensible presentation of ontology assumptions (C12) and their effects are
criteria that apply to ontology construction tools. Tools support (C13) indicates
if a methodology is coherently implemented in a supporting tool. Finally expres-
siveness (C10) is an inherent property of the underlying ontology languages, but
is also reflected in the methodology itself, as it might restrict the expressiveness
to a subset of the underlying language.

4 Existing ontology design methodologies

In a survey [5] we assessed sixteen methodologies with respect to the aforemen-
tioned criteria. To provide a comprehensive overview we selected methodologies
from a broad range of backgrounds and with different foci. Three methodolo-
gies, namely CommonKADS [12], Cyc [13] and KBSI IDEF5 [14], have their
roots in the creation of Knowledge Based Systems (KBS). The DOGMA [15]
approach is inspired by database engineering. Five methodologies aim at the construction of ontologies from scratch: Grüninger and Fox [16], Uschold and King [17], METHONTOLOGY [18], Ontology Development 101 [19] (which is targeted at beginners in the field) and UPON (which uses UML) [20]. The DILIGENT [21] and HCOME [22] methodologies emphasize the collaborative evolution of ontologies. The methodologies focusing on reuse of existing knowledge are namely: SENSUS [23], which derives domain ontologies from large scale ontologies, KACTUS [24], which builds domain ontologies from existing ones and ONIONS [25], which builds ontologies from existing knowledge sources. DynamOnt [26] is a wiki-based solution to support evolution and reuse of knowledge. On-To-Knowledge [27] is inspired by Knowledge Management (KM) and focuses on the creation and maintenance of KM applications.

The survey has shown that available ontology development methodologies are not suitable for novice users. Table 1 gives an overview over the results. A plus signifies that this criterion is fulfilled; a minus that it is not. Empty cells mean that the criterion wasn’t applicable.

On the positive side one notes that all methodologies are consistent (C8) and most are well structured (C2). About 50% are described in sufficient detail (C3), but don’t make their steps transparent for the users (C4). Only three methodologies implement error-avoidance techniques (C5), while half of them provide some robustness against error through evaluation and revision loopbacks (C6). Five methodologies manage to hide some formality (C9) and five (partly different) methodologies provide adequate expressiveness (C10). Half of the methodologies are supported by tools (C13). User-adequate terminology (C1), lookahead features (C7), conceptualization flexibility (C11) and ontology assumptions (C12) are not considered in any methodology.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonKADS</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Cyc</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>DILIGENT</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>DOGMA</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>DynamOnt</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Grüninger &amp; Fox</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>HCOME</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>KACTUS</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>KBSI IDEF5</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>METHONTOLOGY</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>ONIONS</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>101</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>OTK</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>SENSUS</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>UPON</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Uschold &amp; King</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
</tbody>
</table>

**Table 1.** Survey result
5 Principles for a new methodology

We put our focus on the areas where existing methodologies and tools are lacking, namely: the usage of adequate terminology, conceptualization flexibility, the comprehensible presentation of ontology assumptions and hiding of formality. To provide non-expert users with the required guidance our methodology will be tool supported.

Our methodology and tool adheres to the following basic principles: hiding of formality, no distinction between A- and T-Box, conceptualization flexibility, flexible hierarchical views and making ontology assumptions explicit. In the following section we will briefly present the single principles.

Hiding of formality: According to [6] formality tends to increase the user’s perception of complexity. Consequently we try to hide all formality by using an intuitive graphical user interface. All steps of our methodology are supported in one view and we provide as much interaction in the form of drag&drop as possible. The only textual interaction is the entering of labels or description, but users won’t be confronted with any code. The graphical representation constitutes an intermediate language which is eventually translated in a formalized ontology.

No distinction between A- and T-Box: The distinction whether a concept should be regarded as a class (T-Box) or as an individual (A-Box) is application-specific (cf. [19]). There is no one way to model a domain of interest. Therefore we introduce so-called entities in our methodology, which in a later stage of our methodology are transformed into either classes or individuals. This makes our approach more flexible and enables user to focus on other aspect of the modeling process. Entities can be arranged in an inheritance hierarchy similar to classes, they can be connected using object properties and have datatype properties (with values) attached to them. Entities can be transformed into individuals if they are leaf nodes of the inheritance hierarchy and if all their datatype and object properties are defined in higher levels of the hierarchy. All other entities are transformed into classes.

Conceptualization flexibility is important for novice users as they are prone to committing errors and often learn a new tool by applying a trial&error strategy instead of reading available documentation. We provide flexible means to convert between entities, properties and relationships, thus reducing the amount of work a user has to invest when correcting errors. For example name might be specified as a datatype property, but later in the development process is decided that it should be represented as an entity with datatype properties for salutation, given name and last name. In our approach the user can simply drag the name datatype property out of its parent entity and thus transform it into an entity of its own linked to the original one via an object property. Our approach employs easy drag&drop or wizard-style conversions between all types of entities and properties. This flexibility is also reflected by the entities models, which treats classes and individuals equally and also allows for flexible conversion between them.
Flexible hierarchical views: Confusion between is-a and part-of relations is a common error by inexperienced users (cf. [19]). This stems from the fact, that hierarchical structures can be differentiated into logical and partitive (cf. [28]) and novice users are used to partitive hierarchical structures (e.g. a university campus contains buildings) not to logical ones (e.g. a full professor is a kind of faculty member). All tools known to the authors use the is-a relations as main structural relationship. Our tool will provide different views on the knowledge model besides the inheritance hierarchy and will allow users to define any given object property as the main relationship for displaying a hierarchy.

Making ontology assumptions explicit: In our evaluation it turned out, that novice users follow the closed-world and unique name assumptions, because it reflects their experiences with the information systems they are used to. The open-world assumption of ontologies and the possible non-uniqueness of names lead to confusion. For example most users in our evaluation automatically assumed that the individuals of a given class constituted a complete set and at first didn’t understand the need to explicitly specify this (for more examples see [29]). Our approach will provide users with feedback and ask guidance questions to make ensure that ontology assumptions are made explicit and the encoded meaning is the one the user intended to convey.

![Methodology life-cycle based on [18]](image)

Our methodology loosely follows the life-cycle described in [18]. It consists of specification (define scope/purpose), knowledge acquisition, conceptualization (organize knowledge), formalization, evaluation and maintenance, but our primary focus is the beginning of the life cycle (specification, knowledge acquisition, conceptualization and formalization phases). We employ the evolving prototype life-cycle model and there is no fixed order of phases; users can go back and forth between phases as necessary. For this the supporting tool implements all phases in one unified view (cf. figure 2). In the following section we describe the steps of the methodology (see figure 1) by giving a short presentation of our tool.

6 Tool

In this section we describe the prototype of the tool implementing our methodology. Tool-supporting and a tight coupling between methodology and tool is a key factor for the success with non-experts users, as instant feedback is of even greater importance for them, than for expert users. We illustrate the features of
our prototype using selected task of the creation of a simple purchase order ontology as our example. This ontology may be regarded as a small part in a bigger ontology covering commercial documents. As aforementioned we follow the life-cycle described in [18], figure 1 shows the steps of our methodology. The current status of our implementation covers the following phases: define scope/purpose, acquire and augment knowledge, organize knowledge and formalize knowledge. These phases are described here using our example.

Define scope/purpose: In the first phase scope and focus of the ontology are described in textual form. Users are also urged to provide (textual) competency questions the ontology should be able to answer. Guidance is provided by giving examples and an interactive question and answer process, e.g. ‘who is the intended audience of the ontology’, ‘what is its application area’, etc. For documentation purposes the results are added to the final ontology. In our tool this information is displayed in the form of post-it notes, whose visibility can be toggled by the user.

Acquire and augment knowledge: Our tool supports gathering knowledge manually (i.e. the user enters terms) and automatically. The automatic process uses a plug-in interface for various import formats, e.g. free texts (using text mining techniques), XML schemata, relational databases and mindmaps. The imported terms are displayed in a desktop-metaphor and each term is represented by an index card. Internally each term is represented as an entity. Users can then add textual definitions for the entities

Fig. 2. Screenshot of our tool: OntoClippy
either by hand or automatically, e.g. by reusing definitions from lexical ontologies like Wordnet [30]. These definitions are also stored for documentation purposes.

In our example we choose to enter knowledge manually. Figure 2 shows our editor containing the following entities: Order, Supplier, PurchaseOrder, ShippingAddress, Item, Qty, Schedule and Price.

The Wordnet definition for purchase order is ‘a commercial document used to request someone to supply something in return for payment and providing specifications and quantities’, gives ‘IBM received an order for a hundred computers as an example’ and shows that it is a derivationally related form of the term order.

Organize knowledge: The first step in organizing the knowledge is clustering related entities. Again this can be done manually or automatically exploiting similarity measures between the entities themselves and their definitions. The results of this step are visualized by the proximity of the index cards on the desktop. In a next step a hierarchy of entities is build. This is realized using drag&drop, dragging an entity on another one, makes it a sub-entity. The semantics of this depends on the main structural relationship, which is user-definable (cf. flexible hierarchical views). The default relationship is the is-a relation.

Utilizing the Wordnet definition from above PurchaseOrder is made a sub-entity of Order simply by drag&drop (see the first step in figure 3).

Additionally users are prompted with questions in order to ensure that the encoded meaning matches their intended meaning, e.g. asking the user if all sub-entities of an entity should be disjoint. Then properties are added to the entities. These are either created manually, created by converting existing entities, or selected from automatic proposals extracted from the entity definitions. This process is supported by using refactoring mechanism (as used in object-oriented software development), e.g. if all sub-entities share
common properties the user is asked if they should be moved to the parent entity.

In our example it is decided that Supplier should be included as a property of PurchaseOrder. This is realized by dragging the Supplier entity onto the PurchaseOrder entity, which transforms the former into a property of the latter (see second step on figure 3).

Then relationships are added to the ontology. Figure 4 shows a containment relationship, specifying that a PurchaseOrder contains Items.

Fig. 4. Adding relationships

Our tool supports the use of different relationships for creating hierarchies. Figure 5 shows the same information as figure 4, but uses the containment relationship as the main structural relationship.

Fig. 5. Using containment as primary structural relationship
Formalize knowledge: In a last step entities are marked as either classes or individuals. In our example figure 6) shows that CountryCode is specified as a class, while DE and IT are instances of this class.

![Diagram](image)

**Fig. 6.** Transforming entities into classes and individuals

This final form of the graphical representation is equivalent to a restricted subset of OWL and is then stored as valid OWL ontology for further usage.

7 Conclusions and Future Work

In this paper we presented our proposal for a tool supported methodology for ontology design by non-experts. It implements novel approaches to conceptualization flexibility and the comprehensible presentation of ontology assumptions. The described methodology and user-friendly graphical user interface is currently in early prototype stage and is still under active development.

The current implementation has been successfully used for the development of ontologies describing e-business messages from the automotive and furniture domains in various formats (EDI, XML, etc.). In our future work we will further develop the prototype and add ontology evaluation strategies to it and the underlying methodology. Additionally we will add more advanced algorithms for clustering, finding of similarities and automatically detecting classes and instances. Then we will test the prototype with users from various backgrounds and different levels of experience in ontology design. Based on their feedback an enhanced version of our tool supported methodology will be developed. Eventually we will methodically assess the efficiency and productivity by means of a comparative evaluation with other tools and approaches. Special regard will be given to the user-experience and the quality of the created ontologies.

8 Acknowledgements

Research work for this paper was partly funded by the EU IST FP6 project STASIS (http://www.stasis-project.net).
References

Semantic Analysis for an Advanced ETL framework

Sonia Bergamaschi\textsuperscript{1}, Francesco Guerra\textsuperscript{1}, Mirko Orsini\textsuperscript{1}, Claudio Sartori\textsuperscript{2}, and Maurizio Vincini\textsuperscript{1}

\textsuperscript{1} Università di Modena e Reggio Emilia\hspace{1cm} firstname.lastname@unimore.it
\textsuperscript{2} Università di Bologna\hspace{1cm} claudio.sartori@unibo.it

Abstract. In this paper we propose a system supporting the semi-automatic definition of inter-attribute mappings and transformation functions used as ETL tool in a data warehouse project. The tool supports both schema level analysis, exploited for the mapping definitions amongst the data sources and the data warehouse, and instance level operations, exploited for defining transformation functions that integrate in a common representation data coming from multiple sources.

Our proposal couples and extends the functionalities of two previously developed systems: the MOMIS integration system and the RELEVANT data analysis system.

1 Introduction

Enterprise Information Systems provide a technology platform that enables organizations to integrate and coordinate their business processes\textsuperscript{3}. The data warehouse represents definitely one of the most important components of information systems, since it enables business intelligence analysis on data coming from multiple sources. Traditional architectures of data warehouse systems rely on extraction, transformation and loading (ETL) tools for building and populating the data warehouse. Such tools support (a) the identification of relevant information at the source side, (b) the extraction of this information, (c) the customization and integration of the information coming from multiple sources into a common format, (d) the cleaning of the resulting data set on the basis of database and business rules, and (e) the propagation of the data to the data warehouse and/or data marts [11].

ETL processes are crucial for the data warehouse consistency, and are typically based on constraints and requirements expressed in natural language in the form of comments and documentations. Consequently, the aforementioned tasks are manually performed by the designer or the administrator of the data warehouse. In the context of traditional databases, this fact does not represent a real big issue: 1) the processes requiring the manual user intervention involve data source schemata that are generally

\textsuperscript{*} This work was partially supported by MIUR co-funded project NeP4B (http://www.dbgroup.unimo.it/nep4b).
\textsuperscript{3} from Wikipedia, http://en.wikipedia.org/wiki/Enterprise\_Information\_System
fixed. Thus, designers are asked to set up the ETL processes once per data source (during the start up phase); 2) all the data sources collected in the data warehouses typically belong to the same company and are known and managed by the same designers. Thus, technical documentations and personal knowledge are available and may be usefully exploited for building the data warehouse.

Nowadays, the actual business needs require enterprise information systems to have a great flexibility concerning the allowed business analysis and the treated data. Temporary alliances of enterprises, market analysis processes, the data availability on Internet push enterprises to quickly integrate unexpected data sources for their activities. Therefore, the reference scenario for data warehouse systems extremely changes, since data sources populating the data warehouse may not directly known and managed by the designers. We have classified four major critical activities:

- **Automating Extraction processes.** Designers may no longer rely on internal documents, comments and previous knowledge on the data source contents. Moreover, the manual exploration of the data source contents may be really time-consuming. Techniques for identifying the information held by the data sources and extracting the relevant data for populating the data warehouse are required.

- **Automating Transformation processes.** Data from different sources may not be homogeneous, i.e. different metrics may be used for expressing similar values, synonyms may be used for representing the same values (and vice-versa the same value in different sources may describe a different concept), and values may be expressed with different granularity levels. Therefore, the data transformation processes are crucial for making attribute values from different data sources comparable: techniques to automatically perform data transformations are needed.

- **Relaxing the Transformation processes.** Deeply homogenizing transformations risk to flatten data warehouse structure and contents, thus allowing less accurate data analysis on such data. A balance should be pursued between facility of introducing new sources and preservation of structural and contents homogeneity allowing accurate business intelligence processes.

- **Speeding-up and easying ETL process execution.** Fast and simple ETL execution is a crucial competition factor when the sources populating the data warehouse are dynamic. Moreover, ETL tools should be able to manage different kinds of data sources, ranging from DBMS, to flat file, XML documents and spreadsheets.

In this paper we propose a tool for automating extraction and transformation processes that couples and extends the MOMIS integration system and the RELEVANT system for the analysis of attribute values (see [4, 7] and related work section).

By means of a semantic analysis, the tool performs two tasks: 1) it works at the schema level, identifying the parts of the schemata of the data sources which are related to the data warehouse; 2) it works at the instance level, allowing the designer to define transformation rules for populating the data warehouse.

The paper is organized as follows: next section introduces some related work and an overview of both the previously developed MOMIS and RELEVANT systems. Section 3 describes the new tool along with a running example on a real application scenario; in section 4, some conclusion and future work are sketched out.
2 Related Work

Many approaches have been proposed for modeling ETL processes both at the logical and conceptual level. Several methods propose extensions of UML in order to describe ETL processes [9]. The advantage of those methods relies on the fact that they are based on a well-accepted, standard modeling language. Other approaches are based on 'ad hoc' techniques having the benefit of representing ETL processes without any restriction imposed by a previously defined, generic language [11].

Recently, techniques exploiting ontologies to manage the heterogeneity of the data in the mapping and transformation processes [10] have been proposed. As our approach, they follow an hybrid approach, where each schema is represented by its own ontology and a common shared vocabulary is provided, while the annotation process is manual and the vocabulary has been done by the designer.

Apart from research efforts, currently, there is a variety of ETL tools available in the market (see [8] for a survey). All major database vendors provide ETL solutions bundling them with their DBMS [1, 3, 2].

Our proposal relies on two previously developed systems: MOMIS and RELEVANT.

The Mediator EnvirOnment for Multiple Information Sources (MOMIS) [4] is a semiautomatic data integration system that follows a global-as-view approach to provide an intelligent access to heterogeneous, structured and semi-structured information sources. MOMIS is based on clustering techniques applied to a set of metadata collected in a Common Thesaurus in the form of relationships describing inter- and intra-schema knowledge about classes and attributes of the local source schemata. The relationships are: 1) extracted from descriptions of local schemata; 2) obtained from the relationships existing in the WordNet database between the meanings associated to the source elements; 3) inferred by means of Description Logics techniques.

RELEVANT [7] is based on the idea that analyzing an attribute domain, we may find values which may be clustered because strongly related. Providing a name to these clusters, we refer to a relevant value name which encompasses a set of values. More formally, given a class $C$ and one of its attributes $At$, a relevant value for it, $rv^{At}$ is a pair $rv^{At} = \langle rvn^{At}, values^{At}\rangle$. $rvn^{At}$ is the name of the relevant value set, while $values^{At}$ is the set of values referring to it. For computing the relevant values, the user may combine three different similarity measures (namely (1) syntactic, mapping all the words of the attribute values in an abstract space, and defining a syntactic similarity function in such space; (2) dominance, introducing a sort of generalization relationship between values; (3) lexical, which identifies semantically related values expressed with a different terminology) and select between two clustering algorithms (a hierarchical and an overlapping algorithm).

3 Making semantic ETL with MOMIS and RELEVANT

The data integration methodologies implemented in the MOMIS and RELEVANT systems are extended for supporting the automation of ETL processes. Section 3.1 de-

---

4 see http://www.dbgroup.unimo.it/Momis/ for more details and publications
scribes how the coupling of the two systems is exploited at the schema level for implementing the extraction process, and in section 3.2 we focus on supporting the semi-automatic definition of a new transformation function at the data level, for the data warehouse population.

3.1 Semantic extraction from data sources

Our approach is based on the idea that a new source cannot modify the data warehouse schema, thus guaranteeing to business intelligence applications and reporting services no changes on the schema they use. However, due to its complexity and the difficulty to manage and “understand” elements of unknown data sources, this task cannot be manually executed (or supported only by graphical interfaces).

The proposed tool takes as input the data warehouse and the data source schemata (more than one source may potentially be added as a new source for the data warehouse with a unique extraction process) and compute mapping between those schemata.

Figure 1 shows the functional architecture of the proposed tool. We identify three main phases: firstly, by exploiting the methodology implemented in MOMIS the descriptions of the sources are extracted and a Thesaurus of relationships between the schema elements is computed. Schema descriptions and the Thesaurus are then exploited by an extension of the RELEVANT technique for computing clusters of similar elements. Finally, by means of a cluster analysis tool, mappings between the data warehouse schema and the new data sources are defined. For each phase, a software component has been built with the following features:
1. The schema descriptions of data sources are extracted by means of wrappers. A wrapper-based technology allows the tool to deal with several kinds of data sources such as spreadsheets, XML documents, text files and databases.

2. The annotation according to a reference ontology/database (in our tool, the WordNet\(^5\) lexical database) allows the specification of a unique meaning to the schema description. In this way, it is possible to exploit the relationships among the ontology concepts to infer relationships between the annotated source schema elements. We support this process by means of automatic techniques (see [6] for more details).

3. The knowledge engine is the component in charge of creating a Thesaurus of relationships between the schema elements (in terms of classes and attributes). Three types of relationships are generated by the engine: relationship connecting synonym (SYN), broader terms/narrower terms (BT/NT) and generic relationships (RT). The process for extracting relationships is borrowed from MOMIS and is based on structural analysis (exploiting primary and foreign keys, attribute memberships), annotations and Description Logics techniques. Notice that the relationships are validated, i.e. only the relationships involving attributes with a compatible data-type are part of the Thesaurus.

4. **RELEVANT** is applied to the descriptions of sources extracted in the first step with the aim of computing clusters of related attributes. **RELEVANT** has been improved by adding new similarity measures to the ones previously implemented. In particular, we exploit three different kinds of measure: 1) syntactic similarity, which compares the alphabets used for describing the attribute values; 2) memberships, which represents the closeness of attributes belonging to the same table; 3) semantic similarity, which takes into account the Thesaurus relationships between classes and attributes. Each similarity measure is represented by an affinity matrix, where a similarity value is computed for each attribute with respect to all other attributes according to the selected semantics. The combination of the values of each affinity matrix is parametric, thus allowing a setup phase where the user assigns an importance degree to some specific similarities. The application of a clustering algorithm (with a user-select threshold) generates clusters of similar elements.

5. Mappings are automatically generated by analyzing the clustering result. The following cases are possible:

   (a) A cluster contains attributes from the data warehouse schema and the new data sources: for each data warehouse attribute a mapping to each attribute in the cluster is generated.

   (b) A cluster contains only attributes from the schema of new data sources: it is not exploited for the mapping generation. This cluster is due to the choice of a too selective clustering threshold.

   (c) A cluster contains only attributes from the data warehouse schema: it is not exploited for the mapping generation. This kind of cluster indicates that there are attributes in the data warehouse schema which are very close and may, perhaps, be fused into a unique table.

\(^{5}\) http://wordnet.princeton.edu/
Running example  The real scenario we refer to is an ongoing experiment within the CROSS lab project, funded by Italian Emilia Romagna region. It concerns the ETL process for the creation of a data warehouse in the field of beverage and food logistics software. A new system, called Bollicine Community business Intelligence (BCI), has been proposed to a consortium of companies for: 1) the analyzing and planning the enterprise market starting from its past data; 2) developing a performance benchmarking w.r.t. general indexes (KPI) obtained by aggregating data of all the members. To reach this goal it is necessary to load all the data about the consortium members in the BCI data warehouse. We experimented in this context our tool: preliminary qualitative results show that by loading their data in a declarative way, the enterprises considerably save human resources.

Let us consider, for example, the Sales fact table of the BCI data warehouse, constituted of 3 analysis dimensions:

\[
\text{SALES(CODE, ARTICLE\_CODE, BRANCH\_CODE, DATE\_MONTH, AMOUNT, QUANTITY, HECTOLITRES)}
\]

\[
\text{ARTICLE(ARTICLE\_CODE,DESCRIPTION,CATEGORY\_DESCRIPTION, SUBCATEGORY\_DESCRIPTION)}
\]

\[
\text{BRANCH(BRANCH\_CODE, DESCRIPTION)}
\]

\[
\text{TIME(DAY\_CODE,WEEK\_CODE,MONTH\_CODE,BIMESTRAL\_CODE,TRIMESTRAL\_CODE, QUADRIMESTRAL\_CODE,SEMESTRAL\_CODE,YEARLY\_CODE)}
\]

We exploited our tool for loading into the data warehouse the data of three new companies. The result is that correct mappings has been found, thus improving time and result quality of the extraction process. Let us focus on the attributes FAMILY\_DESCRIPTION, CATEGORY\_DESCRIPTION, CLASS\_LABEL of the new sources in figure 2. The designer may set the tool in order to take into account only the syntactic similarity measure. In this case the system returns two clusters, one made of the FAMILY\_DESCRIPTION(S1), CATEGORY\_DESCRIPTION(S2), CATEGORY\_DESCRIPTION(DW) attributes and the second one with only the CLASS\_LABEL attribute. Consequently the attribute DW CATEGORY\_DESCRIPTION attribute is mapped with the corresponding attributes in S1 and S2.

Since the attributes FAMILY\_DESCRIPTION(S1), CATEGORY\_DESCRIPTION(S2), CATEGORY\_DESCRIPTION(DW) are annotated with the same “description” concept in WordNet and CLASS\_LABEL is annotated with “label” that is a hyponym term of “description” in WordNet, the knowledge engine generates this set of SYN relationship in the Thesaurus.

\[
\text{ARTICLE\_CATEGORY\_DESCRIPTION(DW) SYN MERCHANDISE\_FAMILY\_DESCRIPTION(S1)}
\]

\[
\text{ARTICLE\_CATEGORY\_DESCRIPTION(DW) SYN ARTICLE\_CATEGORY\_DESCRIPTION(S2)}
\]

\[
\text{ARTICLE\_CATEGORY\_DESCRIPTION(DW) BT PRODUCT\_CLASS\_LABEL(S3)}
\]

\[
\text{ARTICLE\_CATEGORY\_DESCRIPTION(S2) SYN MERCHANDISE\_FAMILY\_DESCRIPTION(S1)}
\]

\[
\text{ARTICLE\_CATEGORY\_DESCRIPTION(S2) BT PRODUCT\_CLASS\_LABEL(S3)}
\]

\[
\text{MERCHANDISE\_FAMILY\_DESCRIPTION(S1) BT PRODUCT\_CLASS\_LABEL(S3)}
\]

These relationships may be exploited with the semantic similarity thus obtaining a unique cluster with all the attributes and consequently a set of mappings between the DW CATEGORY\_DESCRIPTION and the corresponding attributes in the new sources.
3.2 A transformation function based on RELEVANT values

Transformation functions are typically implemented for homogenizing attribute values from different sources, thus allowing users to compare values and to perform data analysis processes. Several functions are proposed by commercial ETL tools to transform numeric data-types on the basis of mathematical functions. Very few transformation functions have been developed for string data-types. Such functions are frequently based on concatenations of different attribute values, and on small syntactic changes, such as case modifications and stemming operations.

Our approach extends the Data Transformation System (DTS) we proposed in [5], for the declarative definition of transformation functions and the automatic query translation. We aim at providing a new kind of transformation function based on semantic analysis for string values. By means of this function, semantically related values of a chosen attribute in the new data source and the correspondent values into the data warehouse target are grouped, thus providing a semantic reconciliation of the attribute values. The transformation function is based on relevant values and on a temporary mapping table stored in a staging area for preserving the original values, thus allowing precise data analysis. The transformation function works according to the following steps:

- Attribute domains analysis. RELEVANT is used for evaluating if the domains of the new source and the data warehouse attribute are compatible, i.e. they describe
similar properties for the attributes. The evaluation is based on the idea that for similar domains, overlapped relevant values are computed. The overlapping degree shows the compatibility of the domains.

- If the domains are not compatible, the user may select to transform and load into the target attribute only the synthesized set of values represented by the relevant values. In this case a temporary mapping table is built for preserving the possibility of more accurate data analysis. The advantage of such solution is a reduced cardinality and consequently a more synthetic representation of the object with a reduced loss of semantics.

- If the domains are compatible, it is possible to completely replace the attribute domain with its relevant values. By means of this function, the values of such attribute, typically an analysis dimension, will consist of homogeneous values thus allowing OLAP analysis (i.e. drill down) on them.

**Running example** Let us consider the BCI table describing articles sold by companies:

```
ARTICLE(ARTICLE_CODE, DESCRIPTION, CATEGORY_DESCRIPTION, SUBCATEGORY_DESCRIPTION)
```

where `CATEGORY_DESCRIPTION` is a dimension for BI analysis. Three sources are involved in the data warehouse, and they use different values to describe similar article categories. The following fragment shows some of the attribute values:

Source: #1
Attribute: FAMILY_DESCRIPTION = {NOT DEFINED, WATER, BEER, WINE, SOFT DRINK, FOOD}

Source: #2
Attribute: CATEGORY_DESCRIPTION = {HARD LIQUOR, BOTTLE WINE, NOT DEFINED, JUICE DRINK, MINERAL WATER, BOTTLE BEER, SEVERAL ALCOHOLIC BEVERAGE}

Source: #3
Attribute: CLASS_LABEL = {NOT DEFINED, MINERAL WATER, BEER, WINE, ALCOHOLIC DRINK, FOOD, BOOZE, FOOD CATERING}

The **RELEVANT** application to these values defines a set of 8 clusters whose names are loaded into the data warehouse attribute instead of the 22 original values:

RV1: NOT DEFINED {NOT DEFINED}
RV2: WATER {MINERAL WATER, WATER}
RV3: FOOD {FOOD, FOOD, FOOD CATERING}
RV4: SEVERAL {SEVERAL}
RV5: BEER {BEER, BOTTLE BEER}
RV6: WINE {WINE, BOTTLE WINE}
RV7: ALCOHOLIC DRINK {ALCOHOLIC DRINK, ALCOHOLIC BEVERAGE, HARD LIQUOR, BOOZE}
RV8: SOFT DRINK {SOFT DRINK, JUICE DRINK}

The original values are stored in a temporary table and they may be exploited to update and revise the values of the data warehouse attribute.
4 Conclusion and future work

In this paper we proposed to couple and extend our previous research on data integration and data analysis for creating an ETL tool. In particular, we focused our work on the extraction phase, by implementing a technique that semi-automatically defines mappings between a data warehouse schema and a new data source, and on the transformation phase, by proposing a new function based on relevant value, particularly useful for supporting drill down operations. We experimented our approach on a real scenario, thus obtaining qualitative results on the effectiveness of the approach.

Future work will be addressed on identifying a benchmark and a set of measures in order to perform a complete technique evaluation.

References

Query Processing in a Mediator System for Data and Multimedia

Domenico Beneventano\textsuperscript{1} and Claudio Gennaro\textsuperscript{2} and Matteo Mordacchini\textsuperscript{2} and R. Carlos Nana Mbinkeu\textsuperscript{1}

\textsuperscript{1} DII - Università di Modena e Reggio Emilia
Via Vignolese 905, 41100 Modena- Italy
firstname.lastname@unimore.it
\textsuperscript{2} ISTI - CNR,
Via Moruzzi 1, Pisa - Italy
firstname.lastname@isti.cnr.it

Abstract. Managing data and multimedia sources with a unique tool is a challenging issue. In this paper, the capabilities of the MOMIS integration system and the MILOS multimedia content management system are coupled, thus providing a methodology and a tool for building and querying an integrated virtual view of data and multimedia sources.

1 Introduction

The research community has been developing techniques for integrating heterogeneous data sources for the last twenty years. One of the main motivations is that data often reside in different, heterogeneous and distributed data sources that users need to access in a single and unified way to gather a complete view of a specific domain. A common approach for integrating information sources is to build a mediated schema as a synthesis of them. By managing all the collected data in a common way, a global schema allows the user to pose a query according to a global perception of the handled information. A query over the mediated schema is translated into a set of sub-queries for the involved sources by means of automatic unfolding-rewriting operations taking into account the mediated and the sources schemata. Results from sub-queries are finally unified by data reconciliation techniques.

Integration systems relying on mediator-based architectures have been developed, and most of them are able to integrate at the same time heterogeneous data sources, i.e. sources that represent their contents with relational, object-oriented and semi-structured (XML and its derived standards) models.

Similarity search for content-based retrieval (where content can be any combination of text, image, audio/video, etc.) has gained importance in recent years, also because of the advantage of ranking the retrieved results according to their proximity to a query. The systems usually exploit data types such as sets, strings, vectors, or complex structures that can be exemplified by XML documents. Intuitively, the problem is to find

\* The work presented in this paper has been partially supported by the Italian FIRB project RBNE05XYPW NeP4B - Networked Peers for Business.
similar objects with respect to a query object according to a domain specific distance measure. However, the growing need to deal with large, possibly distributed, archives requires specialized indexing support to speedup retrieval. The common assumption is that the costs to build and to maintain an index structure are lower compared to the ones that are needed to execute a query.

In [5, 3] we proposed an approach that extends the action sphere of traditional mediator systems allowing them to manage “traditional” and “multimedia” data sources at the same time. The result is implemented in a tool for integrating traditional and multimedia data sources in a virtual global schema and transparently querying the global schema. We believe this is an interesting achievement for several reasons. Firstly, the application domain: there are several use cases where joining traditional and multimedia data is relevant (see Section 2 for a concrete scenario). Secondly, multimedia and traditional data sources are usually represented with different models. While there is a rich literature for transforming the differently modelled traditional data sources into a common model and it is possible to represent different multimedia sources with a uniform standard model such as MPEG-7, a standard for representing traditional and multimedia data does not exist. Finally, different languages and different interfaces for querying “traditional” and “multimedia” data sources have been developed. The former relies on expressive languages allowing expressing selection clauses, the latter typically implements similarity search techniques for retrieving multimedia documents similar to the one provided by the user.

In this paper we mainly focus on query processing in such a Mediator System for Data and Multimedia. First, we introduce the notion of DMS (Data and Multimedia Source) to represent and query data source and multimedia sources in a uniform way. Second, we discuss some cases where multimedia constraints can be expressed in a global query without requiring multimedia processing capabilities at the mediator level. This is an interesting result, since it upsets the usual paradigm on which mediator systems are based stating that the query processing power of a mediator is greater than the one of the integrated data sources [7]. Third, we face a new problem, that is how to optimize the mapping query associated to a global class to perform data fusion. In our framework, the mapping query is based on the full outer join operator which is considered a good candidate in today’s integrating information systems, to perform data fusion [2, 11]. The problem is that full outer join queries are very expensive, especially in a distributed environment as the one of mediator/integration systems. Database optimizers take full advantage of associativity and commutativity properties of join to implement efficient and powerful optimizations on join queries; however, only limited optimization is performed on full outer join [11]. This paper reports the description of work-in-progress about query optimization techniques for full outer join queries in mediator/integration systems. Specially, we deal with only conjunctive queries that consist of conditions of terms connected with AND’s.

This work is part of the NeP4B (Networked Peers for Business)³ project, where we aim to contribute innovative ICTs solutions for SMEs, by developing an advanced technological infrastructure to enable companies of any nature, size and geographic location

³ http://www.dbgroup.unimo.it/nep4b
to search for partners, exchange data, negotiate and collaborate without limitations and constraints. In the NeP4B project, we assume that data sources related to the same domain belong to the same semantic peer, i.e. a super peer exposing a semantic layer. Semantic peers are related by mappings, thus building a network.

In this paper we will focus on a single semantic peer and we will introduce a methodology and a system for building and querying a Semantic Peer Data Ontology (SPDO), i.e. a global schema including traditional and multimedia data sources related to a domain.

The paper is organized as follows: Section 2 describes an applicative scenario. Section 3 proposes an overview of the system; In Section 4 the notion of DMS (Data and Multimedia Source) is introduced to represent and query data source and multimedia sources in a uniform way; then the problem to query a DMS is discussed. Finally, the methodology to build the SPDO related to a set of DMSs is introduced. Section 5 introduces the problem of querying the SPDO and discusses how to optimize the mapping query associated to a global class. Finally, in Section 6, we sketch out some conclusion and future work.

2 An applicative scenario

Let us consider the tourist domain where we can find many portals and websites. The information about a tourist service, location or event is frequently widespread in different specific websites, due to the specialization of the information publisher. Thus, if a user wants to have a complete knowledge about a location, s/he has to navigate through several websites. This issue generates multiple queries with search engines, retrieving incomplete and overlapping information.

Fig. 1. The tourist integrated schema

An application, which provides a unified view of the data provided by different web sources, may provide the tourist promoters and travelers a more complete and easy to
find information. Moreover, since our approach provides only a unified representation of data which still reside on specialized websites, the possibility of having out-of-date information is the same as from navigating the single specialized websites one at a time.

In Figure 1 we provide a graphical representation of the schema obtained from the integration of three data sources. It is composed of 4 classes: hotel, restaurant, event which are related to each other by means of the city class. Special attributes of the City, Hotel and Event classes are Photo, and Poster which are Multimedia Objects. These Multimedia Objects will be annotated with the MPEG-7 standard. Each data source contains traditional data types (e.g. city name in the Event class) together with multimedia data type (e.g. poster in the same Event class). Current search engines for multimedia content perform retrieval of similar objects using advanced similarity search techniques. The NeP4B project aims at combining the exact search on traditional data with a similarity search on multimedia data, exploiting the SPDO obtained by integrating the different data sources (traditional as well as multimedia). As an example, consider a user who wants to attend an event which involves fireworks. Using advanced search techniques over multimedia documents, in this case “poster”, it will be possible to search for posters that contain “festival” and related images of fireworks.

3 The system at a glance

The developed system is based on the MOMIS and the MILOS systems, where MOMIS is exploited for building and querying the SPDO, MILOS for managing the interaction with the multimedia sources (see Figure 2).

MOMIS (Mediator enviRonment for Multiple Information Sources) is a framework to perform integration of structured and semi-structured data sources, with a query management environment able to process incoming queries on the integrated schema [4]. The MOMIS integration process gives rise to a SPDO, in the form of global classes and global attributes, which may be considered as a domain ontology for the data sources.

MILOS [1] is a Multimedia Content Management System tailored to support design and effective implementation of digital library applications. MILOS supports the storage and content based retrieval of any multimedia documents whose descriptions are provided by using arbitrary metadata models represented in XML.

4 A unified view of data and multimedia sources

The notion of DMS (Data and Multimedia Source) is introduced to represent and query data source and multimedia sources in a uniform way. Then the problem to query a DMS is discussed.

4.1 Data and Multimedia Sources

A DMS is represented with a local schema defined in ODLI3 and each class of a DMS schema, in general, includes a set of attributes declared using standard predefined ODLI3 types (such as string, double, integer, etc.). These attributes are referred to
as standard attributes and support selection predicates typical of structured and semi-structured data, such as $=, <, >, \ldots$. Along with the standard attributes, DMS includes another set of special attributes, declared by means of special predefined classes in ODL, which support similarity based searches. We refer to these attributes as multimedia attributes.

Referring to our previous example of the tourist domain (see Figure 1), we can suppose two DMSs, DMS1 with a local class resort and DMS2 with a local class hotel with the following ODL descriptions:

```java
interface resort() {
    // standard attributes
    attribute string Name;
    attribute string Telephone;
    attribute string Fax;
    attribute string Web-site;
    attribute integer Room_num;
    attribute integer Price_avg;
    attribute string City;
    attribute integer Stars;
    // multimedia attributes
    attribute Image Photo;
    attribute Text Description;
}

interface hotel() {
    // standard attributes
    attribute string denomination;
    attribute string tel;
    attribute string fax;
    attribute string www;
    attribute integer rooms;
    attribute integer mean_price;
    attribute string location;
    attribute boolean free_wifi;
    attribute integer stars;
    // multimedia attributes
    attribute Image img;
    attribute Text commentary;
}
```

### 4.2 Querying Data and Multimedia Sources

A DMS $M_i$ can be queried using an extension of standard SQL-like syntax SELECT clause. The WHERE clause consists of a conjunctive combination of predicate on the single standard attributes of $M_i$, as in the following:
SELECT $M_i.A_k, \ldots M_i.A_l, \ldots$
FROM $M_i$
WHERE $M_i.A_x \text{ op}_1 \text{ val}_1$
AND $M_i.A_y \text{ op}_2 \text{ val}_2$
...
ORDER BY $M_i.S_w(Q_1), M_i.S_z(Q_2), \ldots$
LIMIT $K$

Where for the selection predicate in the form $M_i.A_f \text{ op}_p \text{ val}_p$, the $\text{ val}_p$ is a constant or a list of constants, and $\text{ op}_p$ is a scalar comparison operator ($=, \neq, >, \geq, <, \leq$) or a set membership operator (IS IN, IS NOT IN). While $Q_1, Q_2, \ldots$ are constants.

The query is logically processed as follows. The selection predicates (if any) applied to their corresponding attributes select as usually a subset of the objects of $M_i$. The ORDER BY clause sorts the entire result set coming from the selection predicates (if any) in order of decreasing similarity with respect to the given queries objects $Q_1, Q_2, \ldots$. The LIMIT $K$ clause specifies as $K$ the maximum number of result objects desired. Note that, in contrast to what happen in traditional DBMS with standard attributes, the order in which the multimedia attributes appear in the LIMIT clause does not affect the order of the result set, as it is clarified later on.

We suppose that the following assumptions hold:

1. The way by which the returned objects are ordered in not known (black box);
2. The DMS does not return scores associated with the objects indicating the relevance of them with respect to the query;
3. If no ORDER BY clause is specified, DMS will return the records sorted in random order.

The rationale of the above assumptions is that our aim is to work in a general environment with heterogenous DMSs for which we do not have any knowledge of their scoring functions. The motivation is that the final scores themselves are often the result of the contributions of the scores of each attribute. A scoring function is therefore usually defined as an aggregation over partial heterogeneous scores (e.g., the relevance for text-based IR with keyword queries, or similarity degrees for color and texture of images in a multimedia database). Even in the simpler case of single multimedia attributes the knowledge of the scores become meaningless outside the context in which they are evaluated. As an example consider the $TF \times IDF$ scoring function used by normal text search engines. The score of a document depends upon the collection statistics and search engines could use different scoring algorithm.

However, the above assumptions of considering a local DMS as a black box that does not return any score associated to result elements, do not presume that local DMSs do not use internally scoring functions for combing different multimedia attributes. Typically modern multimedia systems use fuzzy logic to aggregate scores of different multimedia attributes that are graded in the interval [0,1]. Classical examples of these functions are the min and mean functions.

To overcome this problem, as will we show in the next sections our mediator system exploits the only the knowledge of the ranking ordering of the objects returned by the multimedia sources during the query in order to compute the final ranking of the merged objects.
4.3 Representing the SPDO

We build a conceptualization of a set of DMSs, composed of global classes and global attributes and mappings between the SPDO and the DMS schemata. We follow a GAV approach, thus this mapping is expressed by defining, for each global class $G$, a mapping query $q_G$ over the local schemata. This mapping query is defined in a semi-automatic way (i.e. the designer is supported by the system to define the mapping query) as follows:

1. A Mapping Table (MT) is specified for each global class $G$, whose columns represent the $n$ local classes $M_1, \ldots, M_n$ belonging to $G$ and whose rows represent the global attributes of $G$. An element $MT(g, l)$ (with $1 \leq g \leq m$ and $1 \leq l \leq n$) represents the local attribute of $M_l$ which is mapped onto a global attribute of $G$. The Mapping Table is automatically generated during the integration process, as described in [4].

2. Multimedia attributes can be mapped only onto Global multimedia attributes of the same type (for instance Image with Image, Text with Text, etc.).

3. Data Conversion Functions are manually specified and, for each not null element $MT(g, l)$, establish the operations that transform the values of the local attributes of $M_l$ into the values of the corresponding the global attribute. Multimedia attributes do not need conversion functions.

4. Join Conditions are defined between pairs of local classes belonging to $G$ and allow the system to identify instances of the same real-world object in different sources. Automatic object identification techniques (see for example [13]) or the designer knowledge may be exploited to define correct join conditions.

5. Resolution Functions [14, 6] are introduced for global attributes to solve data conflicts of local attribute values associated to the same real-world object. In [6] several resolution functions are described and classified; in our framework we consider and implement some of such resolution functions, in particular, the PREFERRED function, which takes the value of a preferred source and the RANDOM function, which takes a random value. If the designer knows that there are no data conflicts for a global attribute mapped onto more than one source (that is, the instances of the same real object in different local classes have the same value for this common attribute), s/he can define this attribute as an Homogeneous Attribute; of course, for homogeneous attributes resolution functions are not necessary. A global attribute mapped into only one local class is a particular case of an homogeneous attribute. For what concern the multimedia attributes, we introduce a new resolution function, called MOST_SIMILAR, which returns the multimedia objects most similar to the one expressed in the query (if any).

By using the introduced concepts, the mapping query is defined on the basis of the full outerjoin-merge operator introduced in [14]: for a global class $G$, the mapping query $q_G$ is obtained by performing the full outerjoin-merging of the local classes $L(G)$ belonging to $G$.

Let us consider for example the ODL description of the global class Hotel of Figure 1. This class is the result of the integration of the two local classes resort and hotel, which are defined above. The mapping table of such Global Classes takes into account
the multimedia attributes Commentary, Description (Text) and Photo, img (Image) introduced by multimedia sources.

<table>
<thead>
<tr>
<th></th>
<th>resort</th>
<th>hotel</th>
</tr>
</thead>
<tbody>
<tr>
<td>name (join)</td>
<td>Name denomination</td>
<td></td>
</tr>
<tr>
<td>telephone</td>
<td>Telephone tel</td>
<td></td>
</tr>
<tr>
<td>fax</td>
<td>Fax fax</td>
<td></td>
</tr>
<tr>
<td>www Web-site</td>
<td>www</td>
<td></td>
</tr>
<tr>
<td>room_num Room_num rooms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>price (RF)</td>
<td>Price_avg mean_price</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>City location</td>
<td></td>
</tr>
<tr>
<td>stars</td>
<td>Stars --</td>
<td></td>
</tr>
<tr>
<td>free_wifi</td>
<td>free_wifi</td>
<td></td>
</tr>
<tr>
<td>photo</td>
<td>Photo img</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>Description commentary</td>
<td></td>
</tr>
</tbody>
</table>

According to the mappings, we specify the name as join attribute intending that instances of the classes resort and hotel having the same Name (denomination) the same real object. Moreover, a resolution function may be defined for the global attribute price, i.e. the value of the global attribute price is the average of the values assumed by both the local sources.

5 Querying the SPDO

Given a global class $G$ with $m$ attributes of which $k$ multimedia attributes, denoted by $G.S_1, \ldots, G.S_k$ (as photo and description in the class Hotel) and $h$ standard attributes, denoted by $G.A_1, \ldots, G.A_h$ (as name and city in the class Hotel), a query on $G$ (global query) is a conjunctive query, expressed in a simple abstract SQL-like syntax as:

```
SELECT G.A_1, \ldots, G.S_j, \ldots
FROM G
WHERE G.A_x op_1 val_1
AND G.A_y op_2 val_2
...
ORDER BY G.S_w(Q_1), G.S_z(Q_2), \ldots
LIMIT K
```

To answer a global query on $G$, the query must be rewritten as an equivalent set of queries (local queries) expressed on the local classes $L(G)$ belonging to $G$. This query rewriting is performed by considering the mapping between the SPDO and the local schemata; in a GAV approach the query rewriting is performed by means of query unfolding, i.e., by expanding the global query on $G$ according to the definition of its mapping query $q_G$. 

108
As explained in the previous section, the ORDER BY clause is used to ideally reorder all the objects on the basis of their similarity expressed by the multimedia conditions. The final result will be a ranked list of the objects of each multimedia source. At a global level, the system needs to combine all this local results. We choose to fuse the results by using a Full Outerjoin-merge operation between the lists. Multimedia result lists are merged by means of both ranks and full join conditions in order to obtain a unique multimedia ranked list of results. The join conditions are defined by a set of join attributes, which are standard attributes of the local DMS.

The case of “standard” local data sources, as for instance a relational database, can be seen as a special case of multimedia source which has no multimedia attributes. As an example consider the example of the query on the city class give in the previous section.

```
SELECT Name
FROM city
WHERE Country = "Italy"
ORDER BY Photo("http://www.flickr.com/32e324e.jpg"), Location(41.89, 12.48)
LIMIT 100
```

Where 'http://www.flickr.com/32e324e.jpg' is the URL of the query image and (41.89, 12.48) are the coordinate of Rome.

5.1 Query unfolding

The process for executing the global query consists of the following steps:

1. **Computation of Local Query conditions:** Each atomic predicate $P_i$, and similarity predicate in the global query are rewritten into corresponding constraints supported by the local classes. For example, the constraints `stars = 3` is translated into a constrain `Stars = 3` considering the local class `resort` and is not translated into any constraint considering the local class `hotel`.

   The ORDER BY clause for global multimedia attributes such the following
   
   ORDER BY $G.S_w(Q_1), G.S_z(Q_2), \ldots$
   
   is translated in a corresponding ORDER BY on the local class $M_l$ restricting it only for its supported multimedia attributes (i.e., where $MT(w,l)$ is not null):

   ORDER BY $MT(w,l)(Q_1), MT(z,l)(Q_2), \ldots$

2. **Computation of Residual Conditions:** Conditions on not homogeneous standard attributes cannot be translated into local conditions: they are considered as residual and have to be solved at the global level. As an example, let us suppose that a numerical global attribute (as for instance `price` in our example) $GA$ is mapped onto $M_1$ and $M_2$, and an AVG function is defined as resolution function, the constraint ($GA = value$) cannot be pushed at the local sources, since the AVG function has to be calculated at a global level and the constraint may be globally true but locally false. It is easy to verify that the same happens defining for `price` the `PREFERRED` or the `RANDOM` resolution function. As mentioned in previous section for multimedia attribute we use the `MOST SIMILAR` resolution function, which
returns the multimedia objects most similar to the one expressed in the query (if any). For example, suppose we are searching for images similar to one specified in the query by means of “ORDER BY” clause. If we retrieve two or more multimedia objects with one or more corresponding images, MOST_SIMILAR function will simply select the image that is more similar to the query image. There are at least two possible implementations of the MOST_SIMILAR resolution function:

– **Mediator Bases Solution**: The mediator implements its similarity functions one for each multimedia attribute and hence it is able to decide which local multimedia attribute value is more similar to the query. This solution has the disadvantage that requires the evaluation of the similarity for each object returned by the sources, which can be, depending on the type of multimedia attribute, computational expensive. Moreover, the similarity implemented at level of Mediator can strongly differ from the ones implemented in the local sources.

– **Rank Based Solution**: Another approach is to simply exploit the rank of the objects in the returned list as indicator of similarity between the attributes values belonging to the objects. This solution is better explained in the next section and it is adopted in our implementation.

3. **Fusion of local answers**: for each local source involved in the global query, a local query is generated and executed on the local sources. The local answers are fused into the global answer on the basis of the mapping query $q_G$ defined for $G$, i.e. by using the Full Outerjoin-merge operation [14]. We assume that: (1) each $M_i$ contains a key, (2) all the join conditions are on key attributes, and (3) all the join attributes are mapped into the same set of global attribute, say $G.A_k$. Objects with the same value of the key attribute represent the same real world objects. Intuitively, this operation is substantially performed in two steps:

(a) **Computation of the full outer join** of local answers ($FOJ$). The result of this operation is ordered on the basis of the multimedia attributes specified in the query, this aspect is deeply examined in the next section.

(b) **Application of the Resolution Functions** : for each attribute $GA$ of the global query the related Resolution Function is applied to $FOJ$ thus obtaining a relation $R_{FOJ}$; in our example the result is the relation $R_{FOJ}(name, www, price)$. Notice that if the $www$ attribute does not have any Resolution Function applied, i.e. we hold all the values, all the $www$ values for a certain hotel X are retained: we will have only one record for X in $R_{FOJ}$, where the value of the $www$ attribute is the concatenation of the $www$ values (obviously adequately separated, e.g. by a comma).

4. **Application of the Residual Condition**: the result of the global query is obtained by applying the residual condition to the $R_{FOJ}$:

```
select name, www, price from R_{FOJ}
where price < 100
```

5.2 Query Fusion: Ranking

As discussed in the previous Section, DMSs have the capability of ranking the result of a query on the basis of the similarity with respect one or more multimedia attributes.
We have pointed out that in general, we are interested in considering the general case in which we do not know anything about how the Multimedia Sources rank the result. This approach allow us to open our work towards a real scenario of integration of heterogenous sources.

In case we do not specify any multimedia attribute in the global query, local queries will not exploit the ranking and then the multimedia records coming from the sources will be fused using the FOJ operator and presented in unspecified order.

If at least one multimedia attribute is specified in the global query, then, depending on the Mapping Table, one or more local $M_i$ will receive a multimedia query using the ORDER BY clause and hence will return ranked result lists. Let us neglect, for now, the LIMIT statement in the LIMIT clause (implicitly considering it as we had specified 'LIMIT infinity'). In this case, we will consider only the DMSs that receive a local query with multimedia attributes. Let $R_i = \langle r_{1,i}, \ldots, r_{n_i,i} \rangle$ be the set records satisfying the sub-query for the $M_i$ source, sorted in decreasing order of relevance from 1 to $n_i$. The Unfolding process will produce a (fused) set of $n_G$ global class records $R_G = \langle r_{1,G}, \ldots, r_{n_G,G} \rangle$ (global answer) sorted in decreasing order of relevance from 1 to $n_G$.

Several possible approach can be adopted to evaluate the order of the global answer, our solution is, at least in principle, to exploit an optimal rank aggregation method based on a distance measure to quantify the disagreements among different rankings. In this respect the overall ranking is the one that has minimum distance to the different rankings obtained from different sources. Several different distance measures are available in literature. However, as discussed in [12], the difficult of solving the problem of distance-based rank aggregation is related to the choice of the distance measure and its corresponding complexity that can be even NP-Hard in some cases. For instance, The rank aggregation obtained by optimizing the Kendall distance (called Kemeny optimal aggregation) for full lists is NP-hard even when the number of list four [9].

However, fortunately, our case falls into this category of the partial rank aggregation problems, in which we measures the distance between only the top-K lists rather than fully ranked lists. In fact: First, for definition of FOJ, a list return by a source, even if complete, can contain records for which certain values of the key attribute $G.A_k$ is not found in the other list, i.e., the DMSs partially overlap. Second, more in general if the number of records of source is very large the use of the full lists becomes unpracticable.

To work around this problem, as suggested in [10], a simple yet effective aggregation function for ordinal ranks is the median function. This function takes as the aggregated score of an object its median position in all the returned lists. Thus, given $n$ multimedia sources that answer to a query by returning $n$ different ranked lists $r_1, \ldots, r_n$, the aggregated score of an object $o$ will be $\text{median}(r_1(o), \ldots, r_n(o))$. The aggregated list will reflect the scores computed using the median function. The median function is demonstrated [10] to be near-optimal, even for top-k or partial lists. Then, it is possible to produce an aggregated list even if the different sources return only top $-k$ lists as their local answers to the query. For all these reasons, we take the median as the MOST_SIMILAR aggregation function for multimedia objects at the global level.
The objects coming from DMSs that do not receive a multimedia query or do not have multimedia attributes at all, will not influence the final rank of the global result list.

5.3 Full Outer Join Optimization Techniques

In this section we will show, by means of some examples, how to optimize the computation of the full outer join of local answers (step 3.a of the query unfolding process).

To show how the optimization method works, for the sake of simplicity and without loss of generality, we restrict ourselves to a global query without multimedia global attributes involved, i.e. without the ORDER BY clause. Let us consider the following global query:

```
select Name, www, price from Hotel
where price < 100 and stars = 3 and free_wifi= TRUE
```

For this global query, the following local queries are generated:

- LQ_resort = `select Name, Price_avg, Web-site from resort
  where Stars = 3`

- LQ_hotel = `select denomination, mean_price, www from hotel
  where free_wifi= TRUE`

Then the Computation of the full join of local answers is:

```
FOJ = LQ_resort join LQ_hotel on
     (LQ_resort.Name=LQ_hotel.Denomination)
```

In this example, we can optimize the FOJ expression with the following one by replacing the Full Outer Join with the join:

```
LQ_resort join LQ_hotel on
LQ_resort.Name=LQ_hotel.Denomination
```

Intuitively, since the predicate `stars = 3 (free_wifi = TRUE)` can be satisfied only in the class resort (hotel), an object of the result must be in LQ resort (LQ hotel) and then we can avoid the Right Outer Join on LQ resort (the Left Outer Join on LQ hotel).

As another example of simplification: If the predicate `stars = 3` is not present in the initial global query, we can optimize the FOJ expression replacing the Full Outer Join with the Left Outer Join.

This simple example shows as constraints for global attributes mapped into only a local class (star and free wi in the example) can be pushed at the local level and allow the simplification of the FOJ expression. It can be demonstrated that this kind of optimization, i.e. the constraint pushing at local level and the simplification of the FOJ expression, is valid in the case of more than two local classes but only if we consider constraint on homogeneous attributes.

On the other hand, the general case is more than two local classes with not-homogeneous attributes. In this case, as we will show in the following, the simplification of the FOJ expression is still valid but constraint cannot be pushed at the local level.
In order to consider an example of more than two local classes with non-homogeneous attributes, let us consider a new local class (called myhotel) added to the global class Hotel with the mapping table shown in Table 3.

Let us consider the following global query:

\[
\text{select Name, www, price from Hotel where stars = 3 and free_wifi= TRUE}
\]

for each local source involved in the global query, a local query is generated:

\[
LQ_{\text{resort}} = \text{select Name, Price_avg, Web-site from resort}
\]

\[
LQ_{\text{hotel}} = \text{select denomination, mean_price, www from hotel where free_wifi= TRUE}
\]

\[
LQ_{\text{myhotel}} = \text{select name, best_price, link from myhotel}
\]

Computation of the Full Join of local answers (FOJ):

\[
\begin{align*}
\text{FOJ}_1 &= LQ_{\text{resort}} \text{ FOJ } LQ_{\text{hotel}} \\
& \quad \text{on } LQ_{\text{resort}}.\text{Name}=LQ_{\text{hotel}}.\text{Denomination} \\
\text{MFJ}_1 &= RF(\text{FOJ}_1) \\
\text{FOJ}_2 &= \text{MFJ FOJ myhotel on MFJ.Name=LQ_{myhotel}.name} \\
\text{MFJ}_2 &= RF(\text{FOJ}_2)
\end{align*}
\]

The final computation of full join for our three classes is MFJ2. In this case we suppose the attribute stars non homogeneous therefore it is not translated at the local level. Intuitively, since the predicate stars = 3 can be satisfied only in the classes resort and myhotel the predicate free_wifi = TRUE can be satisfied only in the class hotel: an object of the result must be obtained combining tuples of LQ_resort and tuples of LQ_hotel or combining tuples of LQ_hotel and tuples of LQ_myhotel. This means that we can optimize the FOJ expression for these three classes in the following way:

\[
\begin{align*}
\text{FOJ}_1 &= LQ_{\text{resort}} \text{ left outer join } LQ_{\text{hotel}} \\
& \quad \text{on } LQ_{\text{resort}}.\text{Name}=LQ_{\text{hotel}}.\text{Denomination}
\end{align*}
\]
\[ MFJ'_1 = RF(FJ_1) \]
\[ FOJ'_2 = MFJ_1 \text{ right outer join myhotel} \]
\[ \text{on} \quad MFJ.Name = LQ_{myhotel}.name \]
\[ MFJ'_2 = RF(FJ_2) \]

The final computation of full join for ours three classes is \( MFJ'_2 \) optimized because the partial result of \( FJ'_1 \) is less than \( FJ_1 \) and \( FJ'_2 \) is less than \( FJ_2 \). Finally the application of residual predicate on \( MFJ_2 \) will be fast than \( MFJ_2 \) because the number of tuple in \( MFJ_2 \) involved in the selection operation is less than a number of tuple in \( MFJ_2 \).

Even if the non-homogeneous attribute \( \text{stars}=3+ \) has not been translated to the local level, we have seen how it is always possible to optimize the FOJ expression. So the fact that an attribute is not homogeneous does not affect the optimization of the FOJ expression. As in the first example, we can note that full outerjoin was replaced with a left(right) outerjoin or a join.

### 6 Conclusion and Future Work

We presented a methodology implemented in a tool that allows a user to create and query an integrated view of data and multimedia sources. The methodology joins and extends the capabilities of two previously developed tools: the MOMIS integration system and the MILOS multimedia content management system. In particular, concerning the query processing, we discussed some cases where multimedia constraints can be expressed in a global query without requiring multimedia processing capabilities at the global level. This is an interesting result, since it upsets the usual paradigm on which mediator systems are based, stating that the query processing power of a mediator is greater than the one of the integrated data sources [8]. We will evaluate the effect of this aspect in our future work.

Future work will also be devoted to experiment the tool in real scenarios. In particular, our tool will be exploited for integrating business catalogs related to the area of “tiles”. We think that such data may provide useful test cases because of the need of connecting data about the features of the tiles with their images.

### References

Semantic Access to Data from the Web

Raquel Trillo¹, Laura Po², Sergio Ilarri¹, Sonia Bergamaschi², and Eduardo Mena¹

¹ IIS Dept., Univ. of Zaragoza, María de Luna 1, Zaragoza, 50018, Spain.
{raqueltl,silarri,emena}@unizar.es
² II Dept., Univ. of Modena and Reggio Emilia, Via Vignolese 905, Modena, 41125, Italy.
{po.laura,bergamaschi.sonia}@unimore.it

Abstract. There is a great amount of information available on the web. So, users typically use different keyword-based web search engines to find the information they need. However, many words are polysemous and therefore the output of the search engine will include links to web pages referring to different meanings of the keywords. Besides, results with different meanings are mixed up, which makes the task of finding the relevant information difficult for the user, specially if the meanings behind the input keywords are not among the most popular in the web. In this paper, we propose a semantics-based approach to group the results returned to the user in clusters defined by the different meanings of the input keywords. Differently from other proposals, our method considers the knowledge provided by a pool of ontologies available on the Web in order to dynamically define the different categories (or clusters). Thus, it is independent of the sources providing the results that must be grouped.

1 Introduction

The big explosion of the World Wide Web in the last fifteen years has made a great and ever-growing amount of information available to users. In this context, search engines have become indispensable tools for users to find the information they need in such an enormous universe. However, traditional search engine techniques are becoming less and less useful because there is simply too much information to search.

Thus, for example, the use of polysemous words in traditional search engines leads to a decrease in the quality of their output [1,2]. For example, if a user inputs “mouse” as a keyword because he/she is interested in obtaining information about “the numerous small rodents typically resembling diminutive rats”, the search engine can return a very high number of hits. Particularly, in this case Google returns about 218,000,000 hits³. However, the first hit that refers to the animal is on the third page of the results (the 36th hit), while the previous hits refer to other meanings of the word “mouse”, such as “pointer

³ Data obtained on 24th April 2009.
device”, “non-profit organization that pioneers innovative school programs”, different companies, etc. Unfortunately, users usually check only one or two pages of the results returned by the search engine [3]. So, an approach is needed to tackle this problem to provide the user with the information that he/she needs.

The problem is that hits referring to different meanings of a user keyword are usually mixed in the output obtained from a search engine. Moreover, the top positions in the ranking are occupied by the meanings which are the most popular on the Web (in the previous example, “pointer device”), hiding the huge diversity and variety of information available to the users on the Web. So, presenting the results to the user classified in different categories, defined by the possible meanings of the keywords, would be very useful, providing interesting advantages, as it is claimed in several previous works (e.g., [4, 5]). Thus, it helps the user to get an overview of the results obtained, to access results that otherwise would be positioned far down in a traditional ranked list, to find similar documents, to discover hidden knowledge, to refine his/her query by selecting the appropriate clusters, to eliminate groups of documents from consideration, and even to disambiguate queries such as acronyms. Summing up, it improves the user’s experience by facilitating browsing and finding the relevant information.

In this paper we propose a new approach, based on semantic technologies, that is able to classify the hits in clusters according to the different meanings of the input keywords. Moreover, it discovers the possible meanings of the keywords to create the categories dynamically in run-time by considering heterogeneous sources available on the Web. As opposed to other clustering techniques proposed in the literature, our system considers knowledge provided by sources which are independent of the indexed data sources that must be classified. Thus, it relies on intensional knowledge provided by a pool of ontologies available on the Web instead of on extensional knowledge extracted from the sources to be grouped by means of statistical information retrieval techniques.

The structure of the rest of the paper is as follows. In Section 2 the main elements of our proposal are presented. Then, in Section 3 some possible improvements of the basic architecture are detailed. In Section 4 a methodology to evaluate our proposal is proposed, and in Section 5 some related works are presented. Finally, some conclusions and plans for future work appear in Section 6.

2 Architecture of the System

Along the last decades, different techniques to cluster documents have been proposed. However, traditional clustering algorithms cannot be applied to search result clustering [6, 4, 7] because, for example, it is not feasible to download and parse all the documents due to the need to provide quick results to the user. In the following, we first present the features that a clustering approach for web search should present. Then, we propose a basic architecture that, enhanced with the improvements described in Section 3, complies with these features. Finally, we explain the workflow of the system and illustrate it with an example.
2.1 Requirements of a Web Search Clustering Approach

Several works have identified the features that a suitable clustering approach in the context of web search should exhibit. Thus, in [6] the following six features are indicated:

- **Relevance**: the clustering approach should separate the pages relevant to the user’s query from the irrelevant ones.
- **Browsable summaries**: it should provide informative summaries of the clusters generated.
- **Overlap**: one web page could belong to more than one cluster, as web pages may have overlapping topics.
- **Snippet-tolerance**: methods that need the whole text of the document should be avoided, due to the excessive downloading time that they would require, in favor of approaches that only rely on the snippets. Moreover, in [6], the authors indicate that “Surprisingly, we found that clusters based on snippets are almost as good as clusters created using the full text [...].”
- **Speed**: clustering should be fast.
- **Incremental**: the clustering process should start as soon as some information is available, instead of waiting until all the information has been received.

Some authors emphasize or propose slightly different features; for example, [4] considers coherent clusters (which implies the need to generate overlapping clusters), efficiently browsable, and speed (algorithmic speed and snippet-tolerance). However, the previous six features are clearly a good representative.

2.2 Basics of the Proposed System

Considering the previous features, we have defined a basic architecture of a system, whose goal is to provide the user with data that satisfy his/her information needs, from the data obtained by a traditional search engine. We have also considered possible improvements to this basic approach, that we detail in Section 3, to totally fulfill the previous features. The proposed system performs two main steps (see Figure 1 for a general overview):

**Step 1: Discovering the semantics of user keywords**

In this step, the system needs to discover the intended meaning of the user keywords to only consider the hits returned by the traditional search engine that correspond with that semantics. So, it is required to find out the possible meanings (interpretations or senses) of each input keyword (ki) of the user, and then to select one interpretation for each keyword. In other words, a Word Sense Disambiguation (WSD [2]) algorithm is performed in two phases in run-time:

---

4 A snippet is a segment that has been snipped off the document returned as a hit. Typically, it is a set of contiguous text around the user keywords in the selected document.
1. In the Extraction of keyword senses phase, the system has to obtain a list of possible meanings for each keyword $k_i$ (called possible keyword senses and denoted as $\{s_{i1}, s_{i2}, ..., s_{in}\}$), so semantic descriptions are required. These descriptions are provided by different sources such as thesauri, dictionaries, ontologies, etc., and extracting these descriptions from them is needed.

At this point, we consider two possibilities to tackle this task:

- **Consulting a well-known general-purpose shared thesaurus such as WordNet.** The main advantage of this option is that it provides a reliable set of the possible meanings of a keyword, and allows to share with others the result of the process. Moreover, the fundamental peculiarity of a thesaurus is the presence of a wide network of relationships between words and meanings. The disadvantage is that it does not cover with the same detail different domains of knowledge. So, some terms or meanings may not be present; for example, the term “developer” with the meaning “somebody who design and implements software” is not stored in WordNet. Moreover, new possible interpretations of a word appear along time; for example, life as “the name of the well-known magazine” or “the name of a film”. These considerations lead to the need of expanding the thesaurus with more specific terms (this can be easily done using a MOMIS component, called WordNet Editor, which allows adding new terms and linking them within WordNet [8]). In contrast, other terms in the thesaurus may have many associated and related meanings; for example, the word “star” in WordNet has three very similar meanings “someone who is dazzlingly skilled in any field”, “an actor who plays a principal role” and “a performer who receives prominent billing”. Some
users could consider all these meanings as equivalent ones, whereas others could be interested into treating them as different ones.

Consulting the shared-knowledge stored in different pools of ontologies available on the Web and using synonym probability measurements to remove redundant interpretations. For this task we perform the process defined in [9, 10] for each keyword. In short, these works rely on the following idea: the more ontologies consulted (each one representing the point of view of their creators), the more chances to find the semantics that the user assigned to the entered keywords. Firstly, the ontological terms that match the keyword (and also their synonyms) are extracted from the whole ontologies by means of several techniques described in [9]. Then, the system treats the possible overlapping among senses (different terms representing the same meaning). For this task, the different terms extracted are considering as input of the following iterative algorithm. For each ontological term, its similarity degree with respect to the others terms is computed. If the similarity degree is lower than a threshold given as parameter (synonym threshold), the two terms are considered to represent different meanings of the keyword; otherwise, they are considered synonyms (they represent the same interpretation of the keyword) and they are merged (integrated) into a single sense following the techniques described in [9], in order to form a new multi-ontological term that will be compared with the rest of terms. Finally, the output of the process is a set of integrated senses, where each element corresponds to a possible meaning of the keyword. The main advantage of this approach is the use of an unrestricted pool of ontologies available on the web because it maximizes the possible interpretations for the keywords; for example, the meaning of “developer” related to “computer science” appears in some ontologies\(^5\). Moreover, it allows new interpretations of the words to be considered without extra effort. The system can also be set up to work with different synonym thresholds so the output can be dynamically changed. Notice that if the synonym threshold is risen, then the integration of terms decreases, so more fine-grained senses are provided as output, whereas if the synonym threshold decreases there is a higher degree of integration and less interpretations are provided for the same keyword. The main disadvantage of this approach is that it could introduce noise and irrelevant information. Besides, the complexity of the process could have a high cost in time, decreasing the average speed.

The trade-off between the two previous approaches is not totally clear. Therefore, we decide to begin the implementation of the proposed architecture considering only WordNet for simplicity reasons, and then to replace the keyword senses extraction module by the techniques described in [9, 10] and to compare the performance of the two approaches.

2. In the *Disambiguation of keyword senses* phase, the system selects the most probable intended meaning for each user keyword (called selected senses, denoted by \( s_{1x}, s_{2y}, \ldots, s_{nz} \)). Many features can be considered to discover the more suitable sense of a word in the context of a document written in natural language; however, the disambiguation process is more complex when a no so-rich context is available, as in this case where the input is a list of plain keywords, so we cannot take advantage of the syntax of a whole sentences or the collocation of the words. Moreover, user queries in keyword-based search engines normally has a length lower than five words. Nevertheless, even under these circumstances, in many cases for a human it is possible to select/discard meanings of the polysemous words. Thus, for example, the word star is expected to be used with the meaning “celestial body” when it appears with the word “astronomy”, and with the meaning “famous actor” when it appears with the word “Hollywood”. Therefore, we try to emulate this behavior by taking into account the context in which a keyword appears, i.e. the possible meanings of the rest of user keywords, to select its most probable intended meaning. In order to do that, we consider the use of the semantic relatedness measure based on information provided by traditional syntactic search engines and the disambiguation method defined in [1, 11]. However, the proposed architecture does not depend on a specific WSD method because not all WSD methods are valid or perform well in all possible contexts, as it is said in [12]. So, other approaches such as a *Probabilistic Word Sense Disambiguation (PWSD)* [12, 13] or using the profile of the user or other information available in the disambiguation process could be considered. In addition, it is also a possibility to ask for user intervention when the context is highly ambiguous (even for humans) as for example “life of stars”.

**Step 2: Semantics-guided data retrieval step**

The goal of this step is to provide the user with only the hits retrieved by a traditional search engine, such as Google or Yahoo!, in which he/she is interested and filter out the irrelevant results. In other words, the system must select the hits that have the same semantics as the intended meaning of the user keywords and discard the others. This process is performed in four phases in run-time:

1. The first phase requires performing a traditional search of hits on the Web, by taking as input the user keywords and using a traditional search engine such as *Google* or *Yahoo*! This search returns a set of relevant ranked hits, which represent the web pages where the keywords appear. The order of the hits, i.e. the ranking of the results provided by the search engines used, depends on the specific techniques used by the engine for that task and its numerous internal parameters. At first we only consider the first 100 hits returned for the search engine to be provided as input of the next phase. Notice that this process can be performed in parallel with the *Discovering the semantics of user keywords* step.
2. In the *Lexical annotation of hits* phase, each of the hits (composed of a title, a URL and a snippet), obtained in the previous phase, is automatically annotated lexically. In more detail, firstly, each returned hit goes through a cleansing process where stopwords are filtered out; then, a *lexical annotation* process is performed for each hit\(^6\). Each user keyword that appears in each filtered hit is marked with the most probable keyword sense by considering the context in which it appears (i.e., by considering its relevant neighbor words) and using WSD methods. As in the *Disambiguation of keywords senses* phase, here it is also possible to consider other WSD approaches such as PWSD\(^7\) (see Section 3 for more details). In this way, a more realistic approach would be considered, as it is usually very difficult even for a human to select only one meaning of the word when the context is limited. Besides, for each keyword we consider the possible meanings obtained in Step 1 and also a new *unknown meaning*. This allows the system to take into account the evolution of a natural language: new senses for a keyword appear when these senses start being used, and only after these senses become widespread they will be integrated in the resources where the semantic descriptions are provided. So, when a user keyword within the snippet of the hit cannot be annotated, it is assumed that it corresponds with the unknown sense.

It should be emphasized that only the information provided by the snippets of the hits is used in this step, without accessing to the full document by means of the URL provided. This is because an approach that needs to download the whole documents would be unsuitable for clustering web search results, due to the requirement to provide a quick answer to the user (see Section 2.1). Besides, to further reduce the processing time, we initially propose considering only the first 100 hits returned by the search engine\(^8\). Moreover, if needed, several hits could be annotated in parallel by using multiple computers/processors. Notice that this process can be performed in parallel with the *Disambiguation of Keyword Senses* phase, but it is needed that the *Extraction of keywords senses* had finished.

3. In the *Categorization of hits* phase, the hits, annotated in the previous phase, are grouped in clusters/categories by considering their lexical annotations. Firstly, the system defines the categories that are going to be considered. The potential categories are defined by the possible combinations of senses for the input keywords. For example, if the user introduces two keywords (k1

---

\(^6\) A *lexical annotation* is a piece of information added to a term that refers to a semantic knowledge resource such as dictionaries, thesauri, semantic networks or others which express, either implicitly or explicitly, a general ontology of the world or a specific domain.

\(^7\) *Probabilistic Word Sense Disambiguation* automatically annotates keywords and associates to any lexical annotation a probability value that indicates the reliability level of the annotation. Besides, it is based on a probabilistic combination of different WSD algorithms, so the process is not affected by the effectiveness of single WSD algorithms, in a particular context or application domain.

\(^8\) We plan to perform experiments to determine a good value for the number of hits to analyze, by studying the trade-off between the recall and the answer latency.
and $k_2$) and, in the previous step, two senses are discovered for $k_1$ ($S_{11}$ and $S_{12}$) and one sense for $k_2$ ($S_{21}$), then the following potential categories are considered: ($S_{11}$, $S_{21}$), ($S_{12}$, $S_{21}$), ($U_1$, $S_{21}$), ($S_{11}$, $U_2$), ($S_{12}$, $U_2$), where $U_1$ and $U_2$ represent the unknown meanings considered for the keywords $k_1$ and $k_2$ respectively. Secondly, each hit has to be assigned to the category defined by the meanings of the input keywords corresponding to the lexical annotation of that hit. The hits of each category are ordered following the ranking returned by the search engine. Therefore, popular hits within a category will appear first, as they reference well-known web pages with topics belonging to that category. Notice that, in this way, only considering the basic architecture, a hit is only assigned to one category (the most probable one). Besides, categories will be labeled with the definitions of the corresponding meanings and potential categories that are allocated no hits will be discarded. This process can be performed in parallel with the disambiguation of keyword senses, but once the Extraction of keyword senses phase has finished. Moreover, the different hits can also be classified in parallel.

4. Finally, the system considers the result of the Disambiguation of keyword senses phase, where a sense is obtained for each keyword, and the cluster that corresponds with these selected senses is provided to the user. We also advocate in this step the use of AJAX to improve the user’s experience (the clustering can be performed in background while the user uses the system).

It would be also possible to consider a different final output for the semantics-guided data retrieval, such as presenting all the hits obtained by traditional search engines grouped in categories considering the different meanings of the input keywords. For more details, see Section 3.

2.3 Workflow of the System and Running Example

In this section, we will illustrate the steps performed by our system with an example. Let us assume that the user is interested in the way that the jaguar species has evolved and he/she enters the input keywords “jaguar” and “evolution”, which we will denote by $k_1$ and $k_2$ respectively. The following steps take place (see Figure 2, which emphasizes the steps that can be performed in parallel):

1. The semantics of the user keywords are discovered (right part of Figure 2). For this, first the possible senses of each of the input keywords are obtained. For simplicity, let us assume that the system uses a well-known general purpose shared thesaurus (another alternative was discussed in Section 2.2). WordNet 3.0 gives just one possible meaning $S_{11}$ for the word “jaguar” (the animal) and two meanings $S_{21}$ and $S_{22}$ for “evolution” (“a process in which something passes by degrees to a different stage” and “the sequence of events involved in the evolutionary development of a species or taxonomic group of organisms”). Then, the disambiguation process selects $S_{11}$ and $S_{22}$ as the most probable meanings of $k_1$ and $k_2$. 
Data are retrieved based on the semantics (left part of Figure 2). First, the input keywords are entered in a traditional search engine and the first 100 hits (title, URL, and snippet) are cleaned and collected. Then, each hit is annotated by marking each keyword appearing in the hit with its most probable sense. The keyword \( k_1 \) can be annotated with two possible senses (\( S_{11} \) and \( U_1 \)) and the keyword \( k_2 \) with three possible senses (\( S_{21}, S_{22}, \) and \( U_2 \)), by considering the possible senses discovered for the keywords and the unknown senses. As an example, in the first hit returned by Google\(^9\), with title “Jaguar Evolution premium Class 5.25 inch Scissors” (a model of scissors), \( k_1 \) is annotated with \( U_1 \) and \( k_2 \) with \( U_2 \). In the second hit, with title “Jaguar Evolution Poster at AllPosters.com” (a poster showing the evolution of different models of the Jaguar car), \( k_1 \) is annotated with \( U_1 \) (the meaning of jaguar as a type of car is not available in WordNet) and \( k_2 \) is annotated with \( S_{21} \). As a final example, in the 20th hit, with title “Paleontology (Dinosaurs): Evolution of Jaguars, lions and tigers...”, \( k_1 \) is annotated with \( S_{11} \) and \( k_2 \) with \( S_{22} \); this is the first hit that is actually relevant for the user. These hits are then grouped in categories defined by

\(^9\) Data obtained on 12nd May 2009.
the possible combinations of senses. Two hits are grouped under the category characterized with $S_{11}$-$S_{22}$, three hits within the category $S_{11}$-$U_{2}$, 36 hits within $U_{1}$-$S_{21}$, 2 within $U_{1}$-$S_{22}$, and 57 within $U_{1}$-$U_{2}$. Finally, the cluster with the intended meaning (in this case, the cluster with the annotations $S_{11}$ and $S_{22}$) is presented to the user.

As illustrated in the previous example, our approach saves a lot of effort to the user in locating the relevant hits. Only the hits in the positions 20th and 63rd are relevant for the user and presented immediately to him/her within the relevant cluster.

3 Improvements of the Basic Architecture

In this section, we describe two improvements for the basic architecture presented before. First, we propose to use a Probabilistic Word Sense Disambiguation to consider the probabilities of the different senses of a keyword instead of just using the most probable sense. Second, we advocate considering synonyms of the user keywords to increase the coverage of the results returned to the user.

3.1 Probabilistic Word Sense Disambiguation

In the basic architecture described in the previous section, the system selects the most probable meaning for each user keyword and presents to the user the cluster with the hits that correspond to those meanings. However, trying to select the most probable sense could be risky. Indeed, in many cases, even the user himself/herself may find difficulties to assign a single meaning to his/her keywords. An alternative approach could make use of PWSD techniques to assign probabilities to the different senses of the keywords. Based on this idea, we suggest three possible ways to improve our proposal:

1. Show more interpretations to the user. This is a slight variant of the approach described in Section 2.2 where, instead of returning just one cluster to the user, which corresponds with the selected senses in the Disambiguation of Keyword senses phase, the system shows all the categories considered containing hits. Thus, this approach recognizes the possibility that the user needs to explore clusters other than the most probable one. For example, if the user keywords are $k_1$ and $k_2$, and $k_1$ has three possible meanings ($S_{11}$, $S_{12}$, $S_{13}$) and $k_2$ two possible meanings ($S_{21}$ and $S_{22}$), then the system retrieves twelve combinations/interpretations (taking into account the unknown senses). The order in which the different created clusters are shown depends on the probability of the selected senses that represent each category.

2. Multi-classification. In this case, the user keywords and categories are interpreted as in the previous case. However, we adopt a different approach to classify the hits. Thus, during the disambiguation of the keywords that
appear in the hits, we assign not just one meaning for each keyword but a list of possible meanings (it can be a ranked list if we are using a PWSD). So, depending on the meanings that are assigned to a keyword in a hit, the hit can be classified in different clusters. As a result, selecting different interpretations of the user keywords, the user can retrieve the same hit.

3. **Multi-classification with ranking.** In this case, the hits are clustered as in the previous case using a PWSD technique but, in addition, the hits in each cluster are ordered by considering their annotated probabilities and not only the ranking of the search engine from which they are retrieved, as in the previous case. During the *Lexical Annotation* phase the PWSD can retrieve different meanings for each user keyword in the snippet or title of a hit, each one associated with a probability value. So, an interpretation/combination of meanings in a hit (for example S11, S22) is associated to a probability computed as the joint probability of the meanings \( \text{Prob}(S11) \times \text{Prob}(S22) \). This probability can be combined with the rank retrieved by the traditional search engine considered, providing a new rank of hits. As a result, selecting an interpretation of the user keywords, the user can retrieve a ranked list of hits by considering their lexical annotations.

We consider that these techniques will be useful to facilitate the user in the task of finding the relevant web pages. An experimental evaluation is needed to compare these approaches.

### 3.2 Retrieval of Synonyms of User Keywords

Important improvements can be achieved by considering synonyms of the user keywords to retrieve more pages to be classified in the different categories [14]. Based on the meanings of the keywords and the semantic resources used, the system is able to identify words that are synonyms. Using these synonyms as input to the search engine, it is possible to retrieve new relevant hits. For example, the words “truck” and “lorry” are synonymous. If we enter “truck” in Google we obtain about 166,000,000 hits, but if the user enters “lorry” the number of hits is approximately 5,220,000. Therefore, if the system considers the synonyms of the user keywords and searches for both “truck” and “lorry” it will provide a better coverage of the relevant results.

It should be noted that if a word used to extract new hits is a polysemous word then the system needs to discard the hits that do not correspond with the intended meaning of the user. For example, if a user inputs “actor” he/she is also probably interested in “star” (as in “film star”). However, there will be some hits with the input keyword start that are irrelevant for the user, such as those hits containing information about the “celestial body”; only hits with the sense “someone who plays a main role in a film” will be interesting for the user. So the systems also performs a retrieval by using these synonyms and, after that, it must lexically annotate the snippets obtained in this enrichment and filter out those annotated hits with a no relevant meaning. Besides, the hits returned will
be grouped in the same clusters, so the number of clusters will not be affected and more relevant hits will be retrieved for each cluster.

It should be emphasized that this synonym expansion approach enhances the recall of the system especially in those cases where the user selects keywords that are not so common (instead of synonyms that are more popular).

4 Experimental Evaluation Methodology

At this point, we ask ourselves how this system can be evaluated. As indicated in [5], there is no general consensus about a good metric to evaluate the quality of the search result clusters; besides, comparing different clustering and search engines approaches is not easy because most of them have different goals (oriented to a specific topic vs. general coverage, allowing queries whose length is one word vs. several words, etc.). Despite this fact, three different methodologies to evaluate a clustering of search results are identified in [5]:

- **Anecdotal evidence** for the quality of the results. Following this methodology, we could evaluate the quality of the results by analyzing the way in which the users behave when using the system. Collecting significant anecdotal evidence is quite challenging.

- **User surveys** with different input queries. This approach suggests asking the users to evaluate the relevance of the results returned for different sample queries. This approach has also some shortcomings, such as the difficulty to find a statistically significant number of users or sample queries.

- **Mathematical measures** [5, 15]. These methods propose the use of metrics to evaluate the quality of the clusters. There are also some difficulties inherent to these methods, such as defining measures that consider the content of each cluster and not only the expressiveness of its labels. Some metrics that could be considered included in this category are: distance measurements between clusters (or ideal distributions), entropy, the Rand Index, the mutual information, purity, precision and recall, etc.

Based on these ideas and the experimental evaluation realized in similar works [16, 14], we plan to perform an evaluation of our system as follows. First, we will define several sets of user keywords, representing different queries. Then, we will enter each of these sets of keywords into a standard search engine and record the first results obtained (the first 100 hits for each user query). The third step involves performing a survey with real users. For each query, we will present the following information to the user: the list of words in the set, the possible meanings of these words (as identified by our knowledge bases), the list of categories created by the system, and the hits returned by the search engine (title and snippet). Then, the user will have to manually classify the hits in categories defined\(^{10}\). With all the groups manually performed by the users,

\(^{10}\) The user will be reminded that a hit can be classified in different categories simultaneously.
we will obtain a probabilistic grouping of the results: a set of groups with hits tagged with the probability that the hit belongs to that group (estimated as the percentage of users that classified the hit under that group). We will compare this “ideal” grouping with the one obtained by our approach, using mathematical measures mentioned above. The membership probabilities of the hits in the ideal distribution can be used either to remove from its clusters those hits with a probability lower than a predefined threshold or to compute the mathematical measures by considering also the difference between the probabilities in the ideal and the computed distribution. Besides these surveys, we also plan to present the user with the output from other clustering engines and use a set of questions to perform a qualitative comparison.

5 Related Work

The clustering of data is a problem that has been studied for a long time and applied in many different application scenarios, based on the so-called Cluster Hypothesis that relevant documents tend to be more similar to each other than to non-relevant documents [17, 5] (i.e., there is a high intra-cluster similarity and a low inter-cluster similarity [5]). In particular, several methods have been proposed to cluster collections of documents (hierarchical methods [18], K-means-based approaches [6], etc.). As opposed to classification methods [19, 20], clustering methods do not require predefined categories (the categories are discovered during the clustering process) and are therefore more adaptive to different types of queries [7, 5]. It may seem that our approach considers a set of predefined categories (the meanings of the user keywords); however, the potential categories used to allocate the hits retrieved are dynamically obtained depending on the user keywords and the knowledge bases queried, and besides they can be merged or refined by considering a synonym threshold given as parameter.

In the context of the web, document clustering can be either pre-computed over a complete collection of documents (e.g., [21]) or computed on-the-fly considering only the documents returned as a result of a search. The latter case (called ephemeral clustering in some works, such as [22]), which our proposal is based on, leads to better results because it focuses only on the documents that are considered relevant (hits) for the user’s query [17, 4, 23]. Besides, it adapts more naturally to the fact that the Web is constantly evolving.

are available about their implementation. From the previous works, Clusty is considered the state-of-the-art in many researches in this area [27, 28, 5, 29].

It is not our goal to provide an in-depth or exhaustive study of existing clustering approaches oriented to web-snippets; for that, we refer the interested readers to [5]. Nevertheless, it should be emphasized that our approach distinguishes itself from other proposals because of its ability to use, not the extensional knowledge provided by the data that have to be clustered by means of classical techniques from Information Retrieval, but the intensional knowledge provided by sources which are independent of the indexed data sources. Thus, we precisely consider the intended semantics of the keywords introduced by the user. Some other proposals that also use semantic techniques are [14, 16]. However, in [14] only WordNet is used and it is assumed the existence of a predefined set of categories. In [16], the use of WordNet is also proposed but they cluster different senses of a word. Nevertheless, this system is limited to queries with a single keyword and does not allow overlapping categories (i.e., hits being classified in more than one category). These limitations are avoided with our proposal.

6 Conclusions and Future Work

We have presented a semantics-based approach to group the results returned by a standard search engine in different categories defined by the possible senses of the input keywords. Our proposal satisfies desirable features identified in the literature [6] for this kind of systems: 1) relevance: the hits that are probably more relevant for the user’s query will be presented in a single cluster at the top of the ranked list of the groups identified; 2) browsable summaries: each cluster will be labeled with the selected meanings of the user keywords and with the snippet of the most representative hit; 3) overlap: a single hit can be classified in different categories by considering Probabilistic Word Sense Disambiguation (PWSD); 4) snippet tolerance: only the snippets of the hits are used to form the groups, without accessing to their whole corresponding document; 5) speed and incremental: we have proposed several techniques to provide results to the user, such as parallel processing and the use of AJAX-based techniques to perform some processing in the background, while the user is interacting with the system.

The next step will be devoted to the development, implementation, and testing of the system proposed. Besides we consider to adapt the approach to be used against heterogeneous structured or semi-structured data repositories such as data bases or UDDI repositories following the ideas of Bernstein et al. [30], that consider that a keyword search approach can be used against structured data to get a quick feel for what is available and set the stage for a more precise integration. The following topic must be also studied in more detail. Clustering the results provided by traditional search engines by considering the meanings of the keywords facilitates the search of information by the users but it is not enough. Thus, even when the semantics of the keywords in both the hits and the user keywords have been identified, the user could be looking for several different queries. For example, if a user inputs “fish” and “person” and indicates
her/his interest in the cluster whose meanings are “a creature that lives and can breathe in water” and “a human being”, she/he could be trying to find information for either mermaids (“a fabled marine creature, typically represented as having the upper part like that of a woman, and the lower like a fish”) or fishermen/fisherwomen (“people that earn their living fishing”) and these options are mixed yet. So, one more phase must be included to deal with these situations.

Acknowledgements

We would like to thank A. Illarramendi for her valuable comments. This work was partially supported by the CICYT project TIN2007-68091-C02-02, the MUR FIRB Network Peer for Business project (http://www.dbgroup.unimo.itnep4b), and the IST FP6 STREP project 2006 STASIS (http://www.dbgroup.unimo.itstasis).

References

1. Gracia, J., Trillo, R., Espinoza, M., Mena, E.: Querying the web: A multiontology disambiguation method. In: Sixth Int. Conf. on Web Engineering (ICWE’06), Palo Alto, California, USA, ACM (July 2006) 41–248
12. Po, L.: Automatic Lexical Annotation: an effective technique for dynamic data integration. PhD in Computer Science, Doctorate School of Information and Communication Technologies, University of Modena e Reggio Emilia, Italy (2009)
17. Hearst, M.A., Pedersen, J.O.: Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In: 19th Int. Conf. on Research and Development in Information Retrieval (SIGIR’96), Zurich, Switzerland, ACM (August 1996) 76–84