Elena Simperl · Barry Norton
Dunja Mladenic · Emanuele Della Valle
Irini Fundulaki · Alexandre Passant
Raphaël Troncy (Eds.)

LNCS 7540

The Semantic Web: ESWC 2012 Satellite Events

ESWC 2012 Satellite Events
Heraklion, Crete, Greece, May 27–31, 2012
Revised Selected Papers

Springer

Lecture Notes in Computer Science 7540

More information about this series at http://www.springer.com/series/7409

Elena Simperl · Barry Norton
Dunja Mladenic · Emanuele Della Valle
Irini Fundulaki · Alexandre Passant
Raphaël Troncy (Eds.)

The Semantic Web: ESWC 2012 Satellite Events

ESWC 2012 Satellite Events
Heraklion, Crete, Greece, May 27–31, 2012
Revised Selected Papers

Springer

Editors

Elena Simperl
University of Southampton
Southampton
UK

Barry Norton
British Museum
London
UK

Dunja Mladenic
Jožef Stefan Institute
Ljubljana
Slovenia

Emanuele Della Valle
DEIB - Politecnico di Milano
Milano
Italy

Irini Fundulaki
Foundation for Research and Technology -
Hellas (FORTH)
Heraklion
Greece

Alexandre Passant
MDG Web Limited
Dublin
Ireland

Raphaël Troncy
Multimedia Communications Department
EURECOM, Campus SophiaTech
Biot
France

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-662-46640-7 ISBN 978-3-662-46641-4 (eBook)
DOI 10.1007/978-3-662-46641-4

Library of Congress Control Number: 2015935043

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web and HCI

Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media
(www.springer.com)

Preface

The Extended Semantic Web Conference (ESWC) is a major venue for discussing the latest scientific results and technology innovations around semantic technologies. This volume contains the post-proceedings of the ESWC 2012 conference, which took place in Heraklion (Crete, Greece), during May 27–31, 2012.

Besides the presentations given in the research tracks[1], the conference featured a rich and broad tutorial and workshop program consisting of 12 workshops and 6 tutorials.

Further, 13 posters and 23 demonstrations were presented at the poster and demonstration sessions, respectively. This volume contains 36 papers describing the posters and demonstrations. While demonstration papers are up to five pages long, poster papers are shorter with two pages maximum.

Further, the volume also contains the 26 best workshop papers, which were selected by the Workshop Chairs and the General Chair on the basis of their reviews and nominations by the Workshop Organizers.

We hope you enjoy the volume!

June 2012

Elena Simperl
Barry Norton
Dunja Mladenic
Emanuele Della Valle
Irini Fundulaki
Alexandre Passant
Raphaël Troncy

[1] For those interested in the lectures of the research tracks, recordings can be found at http://videolectures.net/eswc2012_heraklion/.

Organization

Organizing Committee

General Chair

Elena Simperl	Karlsruhe Institute of Technology, Germany

Program Chairs

Philipp Cimiano	Bielefeld University, Germany
Axel Polleres	Siemens AG Österreich, Vienna, Austria

Poster and Demo Chairs

Barry Norton	Queen Mary University of London, UK
Dunja Mladenic	Jožef Stefan Institute, Slovenia

Workshop Chairs

Alexandre Passant	DERI, National University of Ireland, Ireland
Raphaël Troncy	EURECOM, France

Tutorials Chairs

Emanuele Della Valle	University of Aberdeen, UK
Irini Fundulaki	FORTH-ICS, Greece

PhD Symposium Chairs

Oscar Corcho	UPM, Spain
Valentina Presutti	Institute of Cognitive Sciences and Technologies, Italy

Semantic Technologies Coordinator

Olaf Hartig	Humboldt University of Berlin, Germany

Sponsorship Chair

Frank Dengler	Karlsruhe Institute of Technology, Germany

Publicity Chair

Paul Groth	Vrije Universiteit Amsterdam, The Netherlands

Panel Chair

John Davies	British Telecom, UK

Proceedings Chair

Antonis Bikakis	University College London, UK

Treasurer

Alexander Wahler	STI International, Austria

Local Organization and Conference Administration

STI International, Austria

Program Committee

Linked Open Data Track

Sören Auer	Chemnitz University of Technology, Germany
Juan Sequeda	University of Texas at Austin, USA

Machine Learning Track

Claudia d'Amato	University of Bari, Italy
Volker Tresp	Siemens AG, Germany

Mobile and Sensor Web Track

Alasdair J.G. Gray	University of Manchester, UK
Kerry Taylor	CSIRO ICT Centre, Australia

Natural Language Processing & Information Retrieval Track

Paul Buitelaar	DERI, National University of Ireland, Ireland
Johanna Völker	University of Mannheim, Germany

Ontologies Track

Chiara Ghidini	Fondazione Bruno Kessler, Italy
Dimitris Plexousakis	University of Crete and FORTH-ICS, Greece

Reasoning Track

Giovambattista Ianni	Universitá della Calabria, Italy
Markus Kroetzsch	University of Oxford, UK

Semantic Data Management Track

Claudio Gutierrez	University of Chile, Chile
Andreas Harth	Karlsruhe Institute of Technology, Germany

Services, Processes, and Cloud Computing Track

Matthias Klusch	DFKI, Germany
Carlos Pedrinaci	KMi, The Open University, UK

Social Web and Web Science Track

Fabien Gandon	Inria, France
Matthew Rowe	KMi, The Open University, UK

In-use & Industrial Track

Philippe Cudré-Mauroux	eXascale Infolab, University of Fribourg, Switzerland
Yves Raimond	BBC, UK

Digital Libraries and Cultural Heritage Track

Antoine Isaac	Europeana/Vrije Universiteit Amsterdam, The Netherlands
Vivien Petras	Humboldt University of Berlin, Germany

EGovernment Track

Asunción Gómez-Pérez	Universidad Politécnica de Madrid, Spain
Vassilios Peristeras	European Commission, Belgium

Demonstration and Poster Program Committee

Ravish Bhagdev	University of Sheffield, UK
Barry Bishop	Ontotext AD, UK
Eva Blomqvist	STLab (ISTC-CNR) & Linköping University, Sweden
Luka Bradesko	Jožef Stefan Institute, Slovenia
Ajay Chakravarthy	IT Innovation Center, University of Southampton, UK
Emanuele Della Valle	DEIB - Politecnico di Milano, Italy
Stefan Dietze	L3S Research Center, Germany
Federico Michele Facca	Create-Net, Italy
Carolina Fortuna	Jožef Stefan Institute, Slovenia
Stefania Galizia	INNOVA S.p.A, Italy
Saeed Hassanpour	Stanford University, USA
Anja Jentzsch	Freie Universität Berlin, Germany
Reto Krummenacher	STI Innsbruck, Austria
Jens Lehmann	Universität Leipzig, Germany

Vanessa Lopez	IBM, Dublin, Ireland
Marko Luther	DoCoMo Euro-Labs, Munich, Germany
Maria Maleshkova	KMi, The Open University, UK
Diana Maynard	University of Sheffield, UK
Pablo Mendes	Freie Universität Berlin, Germany
Alexandra Moraru	Jožef Stefan Institute, Slovenia
Andrea Giovanni Nuzzolese	STLab (ISTC-CNR), Italy
Martin O'Connor	Stanford University, USA
Tope Omitola	University of Southampton, UK
Valentina Presutti	STLab (ISTC-CNR), Italy
Yves Raimond	BBC, UK
Dumitru Roman	SINTEF, Norway
Matthew Rowe	KMi, The Open University, UK
Francois Scharffe	LIRMM, University of Montpellier, France
Juan F. Sequeda	University of Texas at Austin, USA
Monika Solanki	Birmingham City University, UK
Tadej Štajner	Jožef Stefan Institute, Slovenia
Thomas Steiner	Google, Germany
Ioan Toma	STI Innsbruck, Austria
Ruben Verborgh	Ghent University, Belgium
Tomas Vitvar	STI Innsbruck, Austria
Johanna Voelker	University of Mannheim, Germany

Workshops

SBPM - Semantic Business Process Management

Nenad Stojanovic	FZI, University of Karlsruhe, Germany
Opher Etzion	IBM Research Laboratory, Haifa, Israel
Ljiljana Stojanovic	FZI, University of Karlsruhe, Germany

KNOW – Knowledge Discovery and Data Mining Meets Linked Open Data

Johanna Völker	University of Mannheim, Germany
Heiko Paulheim	Technical Universität Darmstadt, Germany
Jens Lehmann	Universität Leipzig, Germany
Mathias Niepert	University of Mannheim, Germany

LAPIS - Linked APIs for the Semantic Web: An Essential Problem in Need of a Fresh Look

Craig Knoblock	University of Southern California, USA
Barry Norton	Queen Mary University of London, UK
Ruben Verborgh	Ghent University, Belgium
Sebastian Speiser	Karlsruhe Institute of Technology, Germany
Maria Maleshkova	KMi, The Open University, UK

RED - REsource Discovery

Maria-Esther Vidal	Universidad Simón Bolívar, Venezuela
Zoé Lacroix	Arizona State University, USA
Edna Ruckhaus	Universidad Simón Bolívar, Venezuela

SIMI - Semantic Interoperality in Medical Informatics

Theodora Varvarigou	National Technical University of Athens, Greece
Michael Schroeder	Dresden University of Technology, Germany
George Tsatsaronis	Technical University of Berlin, Germany
Vassiliki Andronikou	National Technical University of Athens, Greece

FEOSW - Finance and Economics on the Semantic Web

Juan Miguel Gomez Berbis	University Carlos III of Madrid, Spain
Angel García Crespo	University Carlos III of Madrid, Spain
Alejandro Rodríguez González	University Carlos III of Madrid, Spain
Brahmananda Sapkota	University of Twente, The Netherlands

IWEST - Evaluation of Semantic Technologies

Raúl García Castro	Universidad Politécnica de Madrid, Spain
Lyndon Nixon	STI International, Austria
Stuart Wrigley	University of Sheffield, UK

DOWNSCALE - Downscaling the Semantic Web

Christophe Gueret	Vrije Universiteit Amsterdam, The Netherlands
Stefan Schlobach	Vrije Universiteit Amsterdam, The Netherlands
Florent Pigou	Association OLPC, France

CVM - Common Value Management

Dieter Fensel	STI Innsbruck, Austria
Holger Kett	Fraunhofer IAO, Stuttgart, Germany
Marko Grobelnik	Jožef Stefan Institute, Slovenia

Interacting with Linked Data

Philipp Cimiano	Bielefeld University, Germany
Christina Unger	Bielefeld University, Germay
Vanessa Lopez	IBM, Dublin, Ireland
Enrico Motta	KMi, The Open University, UK
Paul Buitelaar	DERI, National University of Ireland, Ireland
Richard Cyganiak	DERI, National University of Ireland, Ireland

SWPM - Semantic Web in Provenance Management

Khalid Belhajjame	University of Manchester, UK
Jose Manuel Gomez-Perez	iSOCO, Spain

Paolo Missier	Newcastle University, UK
Satya S. Sahoo	Case Western Reserve University, USA
Jun Zhao	University of Oxford, UK

SEPUBLICA – Scholarly Communication in the Semantic Web

Benjamin Good	Scripps Research Institute, USA
Frank van Harmelen	Vrije Universiteit Amsterdam, The Netherlands
Alexander Garcia Castro	Florida State University, USA
Christoph Lange	University of Birmingham, UK

Steering Committee

Chair

John Domingue

Members

Grigoris Antoniou
Lora Aroyo
Sean Bechhofer
Fabio Ciravegna
Marko Grobelink
Eero Hyvönen
Paolo Traverso

Sponsoring Institutions

Contents

Best Workshop Papers

Demonstration Session

Poster Session

Best Workshop Papers

Finding Concept Coverings in Aligning Ontologies of Linked Data

Rahul Parundekar(✉), Craig A. Knoblock, and José Luis Ambite

Information Sciences Institute and Department of Computer Science,
University of Southern California, 4676 Admiralty Way, Suite 1001,
Marina del Rey, Sunnyvale, CA 90292, USA
{parundek,knoblock,ambite}@usc.edu

Abstract. Despite the recent growth in the size of the Linked Data Cloud, the absence of links between the vocabularies of the sources has resulted in heterogenous schemas. Our previous work tried to find conceptual mapping between two sources and was successful in finding alignments, such as equivalence and subset relations, using the instances that are linked as equal. By using existential concepts and their intersections to define specialized classes (*restriction classes*), we were able to find alignments where previously existing concepts in one source did not have corresponding equivalent concepts in the other source. Upon inspection, we found that though we were able to find a good number of alignments, we were unable to completely cover one source with the other. In many cases we observed that even though a larger class could be defined completely by the multiple smaller classes that it subsumed, we were unable to find these alignments because our definition of *restriction classes* did not contain the disjunction operator to define a union of concepts. In this paper we propose a method that discovers alignments such as these, where a (larger) concept of the first source is aligned to the union of the subsumed (smaller) concepts from the other source. We apply this new algorithm to the Geospatial, Biological Classification, and Genetics domains and show that this approach is able to discover numerous concept coverings, where (in most cases) the subsumed classes are disjoint. The resulting alignments are useful for determining the mappings between ontologies, refining existing ontologies, and finding inconsistencies that may indicate that some instances have been erroneously aligned.

1 Introduction

The Web of Linked Data has seen huge growth in the past few years. As of September 2011, the Linked Open Data Cloud has grown to a size of 31.6 billion triples. This includes a wide range of data sources belonging to the government (42 %), geographic (19.4 %), life sciences (9.6 %) and other domains[1]. A common way that the instances in these sources are linked to others is the use of the *owl:sameAs* property. Though the size of Linked Data Cloud seems to be

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/state/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 3–15, 2015.
DOI: 10.1007/978-3-662-46641-4_1

increasing drastically (10 % over the 28.5 billion triples in 2010), inspection of the sources at the ontology level reveals that only a few of them (15 out of the 190 sources) have some mapping of the vocabularies. For the success of the Semantic Web, it is important that these heterogenous schemas be linked. As described in our previous papers on Linking and Building Ontologies of Linked Data [8] and Aligning Ontologies of Geospatial Linked Data [7], an extensional technique can be used to generate alignments between the ontologies behind these sources. In these previous papers, we introduced the concept of *restriction classes*, which is the set of instances that satisfy a *conjunction* of value restrictions on properties (*property-value pairs*).

Though our algorithm was able to identify a good number of alignments, it was unable to completely cover one source with the classes in the other source. Upon closer look, we found that most of these alignments that we missed did not have a corresponding *restriction class* in the other source, and instead subsumed multiple *restriction classes*. While reviewing these subset relations, we discovered that in many cases the union of the smaller classes completely covered the larger class. In this paper, we describe how we extend our previous work to discover such concept coverings by introducing more expressive set of class descriptions (unions of value restrictions)[2]. In most of these coverings, the smaller classes are also found to be disjoint. In addition, further analysis of the alignments of these coverings provides a powerful tool to discover incorrect links in the Web of Linked Data, which can potentially be used to point out and rectify the inconsistencies in the instance alignments.

This paper is organized as follows. First, we describe the Linked Open Data sources that we try to align in the paper. Second, we briefly review our alignment algorithm from [8] along with the limitations of the results that were generated. Third, we describe our approach to finding alignments between unions of restrictions classes. Fourth, we describe how outliers in these alignments help to identify inconsistencies and erroneous links. Fifth, we describe the experimental results on union alignments over additional domains. Finally, we compare against related work, and discuss our contributions and future work.

2 Sources Used for Alignments

Linked Data, by definition, links the instances of multiple sources. Often, sources conform to different, but related, ontologies that can also be meaningfully linked [8]. In this section we describe some of these sources from different domains that we try to align, instances in which are linked using an equivalence property like *owl: sameAs*.

Linking *GeoNames* with places in *DBpedia*: *DBpedia* (dbpedia.org) is a knowledge base that covers multiple domains including around 526,000 places and other geographical features from the Geospatial domain. We try to align

[2] This work is an extended version of our workshop paper [6]. We have extended the method to find coverings in the Biological Classification and Genetics domains.

the concepts in *DBpedia* with *GeoNames* (geonames.org), which is a geographic source with about 7.8 million things. It uses a flat-file like ontology, where all instances belong to a single concept of *Feature*. This makes the ontology rudimentary, with the type data (e.g. mountains, lakes, etc.) about these geographical features instead in the *Feature Class & Feature Code* properties.

Linking *LinkedGeoData* with places in *DBpedia:* We also try to find alignments between the ontologies behind *LinkedGeoData* (linkedgeodata.org) and *DBpedia*. *LinkedGeoData* is derived from the *Open Street Map* initiative with around 101,000 instances linked to *DBpedia* using the *owl:sameAs* property.

Linking species from *Geospecies* with *DBpedia:* The *Geospecies* (geospecies.org) knowledge base contains species belonging to plant, animal, and other kingdoms linked to species in *DBpedia* using the *skos:closeMatch* property. Since the instances in the taxonomies in both these sources are the same, the sources are ideal for finding the alignment between the vocabularies.

Linking genes from *GeneID* with *MGI:* The Bio2RDF (bio2rdf.org) project contains inter-linked life sciences data extracted from multiple data-sets that cover genes, chemicals, enzymes, etc. We consider two sources from the Genetics domain from Bio2RDF, *GeneID* (extracted from the National Center for Biotechnology Information database) and *MGI* (extracted from the Mouse Genome Informatics project), where the genes are marked equivalent.

Although we provide results of the above four mentioned alignments in Section 4, in the rest of this paper we explain our methodology by using the alignment of *GeoNames* with *DBpedia* as an example.

3 Aligning Ontologies on the Web of Linked Data

First, we briefly describe our previous work on finding subset and equivalent alignments between *restriction classes* from two ontologies. Then, we describe how to use the subset alignments to finding more expressive *union alignments*. Finally, we discuss how outliers in these union alignments often identify incorrect links in the Web of Linked Data.

3.1 Our Previous Work on Aligning Ontologies of Linked Data

In [8] we introduced the concept of *restriction classes* to align extensional concepts in two sources. A *restriction class* is a concept that is derived extensionally and defined by a conjunction of value restrictions for properties (called *property-value pairs*) in a source. Such a definition helps overcome the problem of aligning rudimentary ontologies with more sophisticated ones. For example, *GeoNames* only has a single concept (*Feature*) to which all of its instances belong, while *DBpedia* has a rich ontology. However, *Feature* has several properties that can be used to define more meaningful classes. For example, the set of instances in *GeoNames* with the value *PPL* in the property *featureCode*, nicely aligns with the instances of *City* in *DBpedia*.

Our algorithm explored the space of *restriction classes* from two ontologies and was able to find equivalent and subset alignments between these *restriction classes*. Fig. 1 illustrates the instance sets considered to score an alignment hypothesis. We first find the instances belonging to the *restriction class* r_1 from the first source and r_2 from the second source. We then compute the *image* of r_1 (denoted by $I(r_1)$), which is the set of instances from the second source linked to instances in r_1 (dashed lines in the figure). By comparing r_2 with the intersection of $I(r_1)$ and r_2 (shaded region), we can determine the relationship between r_1 and r_2. We defined two metrics P and R, as the ratio of $|I(r_1) \cap r_2|$ to $|I(r_1)|$ and $|r_2|$ respectively, to quantify set-containment relations. For example, two classes are equivalent if $P = R = 1$. In order to allow a certain margin of error induced by the data-set, we used the relaxed versions *P’* and *R’* as part of our scoring mechanism. In this case, two classes were considered equivalent if $P' > 0.9$ and $R' > 0.9$ For example, consider the alignment between *restriction classes* (*lgd:gnis%3AST_alpha*=NJ) from *LinkedGeoData* and (*dbpedia:Place#type*=http://dbpedia.org/resource/City_(New_Jersey)) from *DBpedia*. Based on the extension sets, our algorithm finds $|I(r_1)| = 39$, $|r_2| = 40$, $|I(r_1) \cap r_2| = 39$, $R' = 0.97$ and $P' = 1.0$. Based on our error margins, we assert the alignment as equivalent in an extensional sense. The exploration of the space of alignments and the scoring procedure is described in detail in [8].

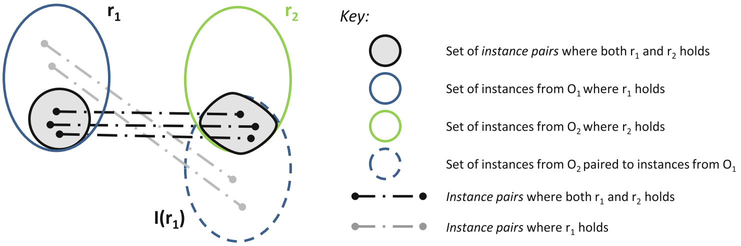

r1
r2
I(r1)
Key:
Set of *instance pairs* where both r1 and r2 holds
Set of instances from O1 where r1 holds
Set of instances from O2 where r2 holds
Set of instances from O2 paired to instances from O1
Instance pairs where both r1 and r2 holds
Instance pairs where r1 holds

Fig. 1. Comparing the linked instances from two ontologies.

Though the approach produced a large number of equivalent alignments, we were not able to find a complete coverage because some *restriction classes* did not have a corresponding equivalent *restriction class* and instead subsumed multiple smaller *restriction classes*. For example, in the *GeoNames* and *DBpedia* alignment, we found that {*rdf:type=dbpedia:EducationalInstitution*} from *DBpedia* subsumed {*geonames:featureCode=S.SCH*}, {*geonames:featureCode=S.SCHC*} and {*geonames:featureCode=S.UNIV*} (i.e. Schools, Colleges and Universities from *GeoNames*). We discovered that taken together, the union of these three *restriction classes* completely define *rdf:type=dbpedia:EducationalInstitution*. To find such previously undetected alignments we decided to extend the expressivity of our *restriction classes*by introducing a disjunction operator to detect concept coverings completely.

3.2 Identifying Union Alignments

In our current work, we use the subset and equivalent alignments generated by the previous work to try and align a larger class from one ontology with a union of smaller subsumed *restriction classes* in the other ontology. Since the problem of finding alignments with conjunctions and disjunction of *property-value pairs* of *restriction classes* is combinatorial in nature, we focus only on subset and equivalence relations that map to an *restriction classes* with a single *property-value pair*. This helps us find the simplest definitions of concepts and also makes the problem tractable. Alignments generated by our previous work that satisfy the single *property-value pair* constraint are first grouped according to the subsuming *restriction classes* and then according to the property of the smaller classes. Since *restriction classes* are constructed by forming a set of instances that have one of the properties restricted to a single value, aggregating *restriction classes* from the group according to their properties builds a more intuitive definition of the union. We can now define the disjunction operator that constructs the union concept from the smaller *restriction classes* in these sub-groups. The disjunction operator is defined for *restriction classes*, such that *(i)* the concept formed by the disjunction of the classes represents the union of their set of instances, *(ii)* each of the classes that are aggregated contain only a single *property-value pair* and *(iii)* the property for all those *property-value pairs* is the same. We then try to detect the alignment between the larger common *restriction class* and the union by using an extensional approach similar to our previous paper. We call such an alignment a hypothesis *union alignment*.

We define U_S as the set of instances that is the union of individual smaller *restriction classes* Union(r_2); U_L as the image of the larger class by itself, Img(r_1)); and U_A as the overlap between these sets, union($Img(r_1) \cap r_2$)). We check whether the larger *restriction class* is equivalent to the union concept by using scoring functions analogous to P' & R' from our previous paper. The new scoring mechanism defines P'_U as $\frac{|U_A|}{|U_S|}$ and R'_U as $\frac{|U_A|}{|U_L|}$ with relaxed scoring assumptions as in P' & R'. To accommodate errors in the data-set, we consider it a complete coverage when the score is greater than a relaxed score of 0.9. That is, the hypothesis *union alignment* is considered equivalent if $P'_U > 0.9$ & $R'_U > 0.9$. Since by construction, each of the subset already satisfies $P' > 0.9$, then we are assured that P'_U is always going to be greater than 0.9. Thus, a *union alignment* is equivalent if $R'_U > 0.9$.

Figure 2 provides an example of the approach. Our previous algorithm finds that {*geonames:featureCode = S.SCH*}, {*geonames:featureCode = S.SCHC*}, {*geonames:featureCode = S.UNIV*} are subsets of {*rdf:type=dbpedia:EducationalInstitution*}. As can be seen in the Venn diagram in Fig. 2, U_L is *Img*({*rdf: type = dbpedia:EducationalInstitution*}), U_S is {*geonames:featureCode = S.SCH*} ∪ {*geonames:featureCode = S.SCHC*} ∪ {*geonames:featureCode = S.UNIV*}, and U_A is the intersection of the two. With the educational institutions example, R'_U for the alignment of *dbpedia:EducationalInstitution* to the union of *S.SCH, S.SCHC & S.UNIV* is 0.98. We can thus confirm the hypothesis and consider this *union alignment* equivalent. Section 4 shows additional examples of *union alignments*.

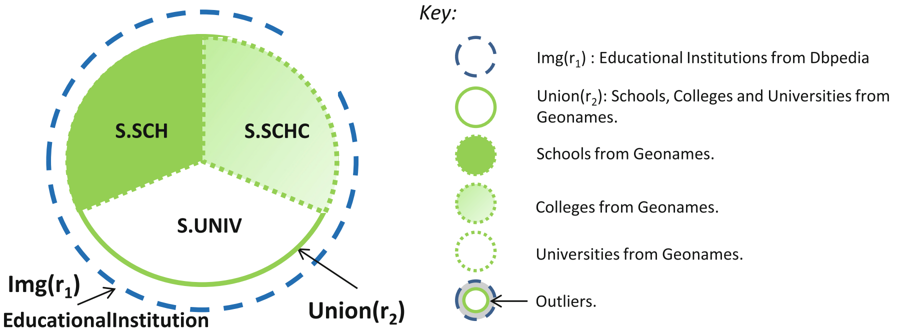

S.SCH
S.SCHC
S.UNIV
Img(r1)
EducationalInstitution
Union(r2)
Key:
Img(r1) : Educational Institutions from Dbpedia
Union(r2): Schools, Colleges and Universities from Geonames.
Schools from Geonames.
Colleges from Geonames.
Universities from Geonames.
Outliers.

Fig. 2. Spatial covering of Educational Institutions from *DBpedia*

3.3 Using Outliers in Union Mappings to Identify Linked Data Errors

The computation of *union alignments* allows for a margin of error in the subset computation. It turns out that the outliers, the instances that do not satisfy the *restriction classes* in the alignments, are often due to incorrect links. Thus, our algorithm also provides a novel method to curate the Web of Linked Data.

Consider the outlier found in the $\{\textit{dbpedia:country} = \textit{Spain}\} \equiv \{\textit{geonames:countryCode} = \textit{ES}\}$ alignment. Of the 3918 instances of *dbpedia:country=Spain*, 3917 have a link to a *geonames:countryCode=ES*. The one instance not having country code ES has an assertion of country code IT (Italy) in *GeoNames*. The algorithm would flag this situation as a possible linking error, since there is overwhelming support for the ES being the country code of Spain. A more interesting case occurs in the alignment of $\{\textit{rdf:type} = \textit{dbpedia:EducationalInstitution}\}$ to $\{\textit{geonames:featureCode} \in \{\textit{S.SCH, S.SCHC, S.UNIV}\}$. For $\{\textit{rdf:type} = \textit{dbpedia:EducationalInstitution}\}$, 396 instances out of the 404 Educational Institutions were accounted for as having their *geonames:featureCode* as one of *S.SCH, S.SCHC or S.UNIV*. From the 8 outliers, 1 does not have a *geonames:featureCode* property asserted. The other 7 have their feature codes as either S.BLDG (3 buildings), S.EST (1 establishment), S.HSP (1 hospital), S.LIBR (1 library) or S.MUS (1 museum). This case requires more sophisticated curation and the outliers may indicate a case for multiple inheritance. For example, the hospital instance in geonames may be a medical college that could be classified as a university.

Our union alignment algorithm is able to detect similar other outliers and provides a powerful tool to quickly focus on links that require human curation, or that could be automatically flagged as problematic, and provides evidence for the error.

4 Experimental Results

The results of union alignment algorithm over the four pairs of sources we consider appear in Table 1. In total, the 7069 union alignments explained (covered) 77966 subset alignments, for a compression ratio of 90 %.

Table 1. Union alignments found in the 4 source pairs

Source1	Source2	Union alignments 12 (Subset Alignments 21)	Union alignments 21 (Subset Alignments 21)	Total union alignments
GeoNames	*DBpedia*	434 (2197)	318 (7942)	752
LinkedGeoData	*DBpedia*	2746 (12572)	3097 (48345)	5843
Geospecies	*DBpedia*	191 (1226)	255 (2569)	446
GeneID	*MGI*	6 (29)	22 (3086)	28

The resulting alignments were intuitive. Some interesting examples appear in Tables 2, 3 and 4. In the tables, for each *union alignment*, column 2 describes the large *restriction class* from *ontology*$_1$ and column 3 describes the union of the (smaller) classes on *ontology*$_2$ with the corresponding property and value set. The score of the union is noted in column 4 ($R'_U = \frac{|U_A|}{|U_L|}$) followed by $|U_A|$ and $|U_L|$ in columns 5 and 6. Column 7 describes the outliers, i.e. values of v_2 that form *restriction classes* that are not direct subsets of the larger *restriction class*. Each of these outliers also has a fraction with the number of instances that belong to the intersection upon the the number of instances of the smaller *restriction class* (or $\frac{|Img(r_1) \cap r_2|}{|r_2|}$). It can be seen that the fraction is less than our relaxed subset score. If the value of this fraction was greater than the relaxed subset score (i.e. 0.9), the set would have been included in column 3 instead. The last column mentions how many of the total U_L instances were we able to explain using U_A and the outliers. For example, the *union alignment* #1 of Table 2 is the Educational Institution example described before. It shows how educational institutions from *DBpedia* can be explained by schools, colleges and universities in *GeoNames*. Column 4, 5 and 6 explain the alignment score R'_U (0.98), the size U_A (396) and the size of U_L (404). Outliers (S.BLDG, S.EST, S.LIBR, S.MUS, S.HSP) along with their P' fractions appear in column 7. We were able to explain 403 of the total 404 instances (see column 8).

We find other interesting alignments, a representative few of which are shown in the tables. In some cases, the *union alignments* found were intuitive because of an underlying hierarchical nature of the concepts involved, especially in case of alignments of administrative divisions in geospatial sources and alignments in the biological classification taxonomy. For example, #3 highlights alignments that reflect the containment properties of administrative divisions. Other interesting types of alignment were also found. For example #7 tries to map two non-similar concepts. It explains the license plate codes found in the state (bundesland) of Saarland[3]. Due to lack of space, we explain the other *union alignments* alongside in the tables. The complete set of alignments discovered by our algorithm are available on our group page.[4]

[3] http://www.europlates.com/publish/euro-plate-info/german-city-codes.

[4] http://www.isi.edu/integration/data/UnionAlignments.

Table 2. Example alignments from the *GeoNames-DBpedia LinkedGeoData-DBpedia*.

#	$\{r_1\}$	$p_2 \in \{v_2\}$	$R'_U = \frac{\lvert U_A\rvert}{\lvert U_L\rvert}$	$\lvert U_A\rvert$	$\lvert U_L\rvert$	***Outliers***	***# Explained Instances***
***DBpedia* (larger) - *GeoNames* (smaller)**							
1	{*rdf:type = dbpedia:EducationalInstitution*}	*geonames:featureCode* ∈ {S.SCH, S.SCHC, S.UNIV}	0.9801	396	404	S.BLDG (3/122), S.EST (1/13), S.LIBR (1/7), S.HSP (1/31), S.MUS (1/43)	403
	As described in Section 4, Schools, Colleges and Universities in *GeoNames*make Educational Institutions in *DBpedia*						
2	{*dbpedia:country = dbpedia:Spain*}	*geonames:countryCode = ES*	0.9997	3917	3918	IT (1/7635)	3918
	The concepts for the country Spain are equal in both sources. The only outlier has it's country as Italy, an erroneous assertion.						
3	*dbpedia:region = dbpedia:Basse-Normandie*	*geonames:parentADM2* ∈ {geonames:2989247, geonames:2996268, geonames:3029094}	1.0	754	754		754
	We confirm the hierarchical nature of administrative divisions with alignments between administrative units at two different levels.						
4	{*rdf:type = dbpedia:Airport*}	*geonames:featureCode* ∈ {S.AIRB, S.AIRP}	0.9924	1981	1996	S.AIRF (9/22), S.FRMT (1/5), S.SCH (1/404), S.STNB (2/5) S.STNM (1/36), T.HLL (1/61)	1996
	In alignmening airports, an airfield should have been an an airport. However, there was not enough instance support.						
***GeoNames* (larger) - *DBpedia* (smaller)**							
5	{*geonames:countryCode = NL*}	*dbpedia:country* ∈ {dbpedia:The_Netherlands, dbpedia:Flag_of_the _Netherlands.svg, dbpedia:Netherlands}	0.9802	1939	1978	dbpedia:Kingdom_of _the_Netherlands	1940
	The Alignment for Netherlands should have been as straightforward as #2. However we have possible alias names, such as *The Netherlands* and *Kingdom of Netherlands*, as well a possible linkage error to *Flag of the Netherlands.svg*						
6	{*geonames:countryCode = JO*}	*dbpedia:country* ∈ {dbpedia:Jordan, dbpedia:Flag_of_Jordan.svg}	0.95	19	20		20
	The error pattern in #5 seems to repeat systematically, as can be seen from this alignment for the coutry of Jordan.						
***DBpedia* (larger) - *LinkedGeoData* (smaller)**							
7	{*dbpedia:bundesland = Saarland*}	*lgd:OpenGeoDBLicensePlate-Number* ∈ { HOM, IGB, MZG, NK, SB, SLS, VK, WND}	0.93	46	49		46
	Our algorithm also produces interesting alignments between different properties. In this case, we find 8 of the 10 license plates in the state of Saarland						

Table 3. Example alignments from the *LinkedGeoData-DBpedia*, *Geospecies-DBpedia*

#	**{r_1}**	$p_2 \in \{v_2\}$	$R'_U = \frac{\vert U_A\vert}{\vert U_L\vert}$	$\vert U_A\vert$	$\vert U_L\vert$	***Outliers***	***# Explained Instances***
8	{*rdf:type*, *dbpedia:EducationalInstitution*}	*rdf:type* ∈ {lgd:Amenity, lgd:K2543, lgd:School, lgd:University, lgd:WaterTower}	0.9901	2609	2610		2609
	Educational Institutions in *DBpedia* can be explained with classes in *LinkedGeoData*. An example of an incorrent alignment, a water tower has been linked to as an educational institution.						
LinkedGeoData **(larger) -** ***DBpedia*** **(smaller)**							
9	{*lgd:gnisST_alpha = NJ*}	*dbpedia:subdivisionName* ∈ {Atlantic, Burlington, {Cape May, Hudson, Hunterdon, Monmoth, New Jersey, Ocean, Passaic}	1.0	214	214		214
	Due to missing instance alignments, this *union alignment* incorrectly claims that the state of New Jersey is composed of 9 counties while actually it has 21.						
10	{*rdf:type = lgd:Waterway*}	*rdf:type* ∈ dbpedia:River dbpedia:Stream}	0.97	33	34	dbpedia:Place(1/94989)	34
	Waterways in *LinkedGeoData* as equal to the union of streams and rivers from *DBpedia*						
DBpedia **(larger) -** ***Geospecies*** **(smaller)**							
11	{*rdf:type = dbpedia:Amphibian*} *dbpedia:Amphibian* }	*geospecies:hasOrderName* ∈ {Anura, Caudata, Gymnophionia}	0.99	90	91	Testudines (1/7)	91
	Species from *Geospecies* with the order names Anura, Caudata & Gymnophionia are all Amphibians We also find inconsistancies due to misaligned instances, e.g. one Turtle (Testidune) was classified as amphibian.						
12	{*rdf:type = dbpedia:Salamander*}	{*geospecies:hasOrderName* = Caudata}	0.94	16	17	Testudines (1/7)	17
	Upon further inspection of #11, we find that the culprit is a Salamander						
Geospecies **(larger) -** ***DBpedia*** **(smaller)**							
13	{*rdf:type = dbpedia:Plant*}	{*geospecies:inKingdom* = geospecies:kingdoms/Ab}	0.99	1874	1876	geospecies:kingdoms/Ac(1/8)	1875
	The Kingdom Plantae, from both sources, almost matches perfectly. The only inconsistant instance happens to be a fungus.						

Table 4. Example alignments from the *GeneID-MGI*

#	$\{r_1\}$	$p_2 \in \{v_2\}$	$R'_U = \frac{\|U_A\|}{\|U_L\|}$	$\|U_A\|$	$\|U_L\|$	***Outliers***	***# Explained Instances***
14	{*geospecies:inOrder = geospecies:orders/jtSaY*}	*dbpedia:ordo* ∈ {dbpedia:Carnivora, dbpedia:Carnivore}	0.99	247	247		247
	Inconsistancies in the object values can also be seen - Carnivores from *Geospecies* are aligned with both : Carnivora & Carnivore.						
15	{*geospecies:hasOrderName = Chiroptera*}	*dbpedia:ordo* ∈ {Chiroptera@en, dbpedia:Bat}	1	111	111		111
	We can detect that species with order Chiroptera correctly belong to the order of Bats. Unfortunatey, due to values of the property being the literal “Chiropta@en”, the alignment is not clean.						
***GeneID* (larger) - *MGI* (smaller)**							
16	{*bio2rdf:subType = pseudo*}	{*bio2rdf:subType* = Pseudogene}	0.93	5919	6317	Gene (318/24692)	6237
	Due to the absence of a clear hierarchy, we found only a few hierarchical relations. For example, alignments of the classes Pseudogenes.						
17	{*bio2rdf:xTaxon = taxon:10090*}	*bio2rdf:subType* ∈ {Complex Cluster/Region, DNA Segment, Gene, Pseudogene}	1	30993	30993		30993
	The Mus Musculus (house mouse) taxonomy is completely composed of complex clusters, DNA segments, Genes and Pseudogenes .						
***MGI* (larger) - *GeneID* (smaller)**							
18	{*bio2rdf:subType = Pseudogene*}	*bio2rdf:subType = pseudo*	0.94	5919	6297	other (4/230) protein-coding (351/39999) unknown(23/570)	6297
	Inconsistancies are also evident as the values pseudo and Pseudogene are used to denote the same thing.						
19	{*mgi:genomeStart = 1*}	*geneid:location* ∈ {1, 1 0.0 cM, 1 1.0 cM, 1 10.4 cM, ...}	0.98	1697	1735	“”(37/1048) 5 (1/52)	1735
20	{*mgi:genomeStart = X*}	*geneid:location* ∈ {X, X 0.5 cM, X 0.8 cM, X 1.0 cM, ...}	0.99	1748	1758	“”(10/1048)	1758
	We find interesting alignments like #19 & #20 , which align the genome start position in *MGI* with the location in *GeneID* As can be seen, the values of the locations (distances in centimorgans) in *GeneID* contain genome start value as a prefix. Inconsistancies are also seen, e.g. in #19 a gene that starts with 5 is misaligned and in #20, where the value is an empty string.						

Outliers. In alignments that had inconsistencies, we identified three main reasons: **(i)** *Incorrect instance alignments* - outliers arising out of possible erroneous equivalence link between instances (e.g. #4, #8, etc.), **(ii)** *Missing instance alignments* - insufficient support for coverage due to missing links between instances or missing instances (e.g. #9, etc.), **(iii)** *Incorrect values for properties* - outliers arising out of possible erroneous assertion for property (e.g. #5, #6, etc.). In the tables, we also mention the classes that these inconsistencies belong to along with their support.

5 Related Work

Ontology alignment and schema matching have been a well explored area of research since the early days of ontologies [1,3] and received renewed interest in recent years with the rise of the Semantic Web and Linked Data. Though most work done in the Web of Linked Data is on linking instances across different sources, an increasing number of authors have looked into aligning the source ontologies in the past couple of years. Jain et al. [4] describe the BLOOMS approach which uses a central forest of concepts derived from topics in Wikipedia. An update to this is the BLOOMS+ approach [5] that aligns Linked Open Data ontologies with an upper-level ontology called Proton. BLOOMS & BLOOMS+ are unable to find alignments because of the small number of classes in *GeoNames* that have vague declarations. The advantage of our approach over these is that our use of *restriction classes* is able to find a large set of alignments in cases like aligning *GeoNames* with *DBpedia* where Proton fails due to a rudimentary ontology. Cruz et al. [2] describe a dynamic ontology mapping approach called *AgreementMaker* that uses similarity measures along with a mediator ontology to find mappings using the labels of the classes. From the subset and equivalent alignment between *GeoNames*(10 concepts) and *DBpedia*(257 concepts), AgreementMaker was able to achieve a precision of 26 % and a recall of 68 %. We believe that since their approach did not consider unions of concepts, it would not have been able to find alignments like the Educational Institutions example (#1) by using only the labels and the structure of the ontology, though a thorough comparison is not possible. In our work, we find equivalent relations between a concept on one side and a union of concepts on another side. *CS*R [9] is a similar work to ours that tries to align a concept from one ontology to a union of concepts from the other ontology. In their approach, the authors describe how the similarity of properties are used as features in predicting the subsumption relationships. It differs from our approach in that it uses a statistical machine learning approach for detection of subsets rather than the extensional approach. An approach that uses statistical methods for finding alignments, similar to our work, has also been described in Völker et al. [10]. This work induces schemas for RDF data sources by generating OWL2 axioms using an intermediate associativity table of instances and concepts (called *transaction data-sets*) and mining associativity rules from it.

6 Conclusions and Future Work

We described an approach to identifying *union alignments* in data sources on the Web of Linked Data from the Geospatial, Biological Classification and Genetics domains. By extending our definition of *restriction classes* with the disjunction operator, we are able to find alignments of union concepts from one source to larger concepts from the other source. Our approach produce coverings where concepts at different levels in the ontologies of two sources can be mapped even when there is no direct equivalence. We are also able to find outliers that enable us to identify inconsistencies in the instances that are linked by looking at the alignment pattern. The results provide deeper insight into the nature of the alignments of Linked Data.

As part of our future work we want to try to find a more complete descriptions for the sources. Our preliminary findings show that the results of this paper can be used to find patterns in the properties. For example, the *countryCode* property in *GeoNames* is closely associated with the *country* property in *DBpedia*, though their ranges are not exactly equal. We believe that an in-depth analysis of the alignment of ontologies of sources is warranted with the recent rise in the links in the Linked Data cloud. This is an extremely important step for the grand Semantic Web vision.

References

1. Bernstein, P., Madhavan, J., Rahm, E.: Generic schema matching, ten years later. Proc. VLDB Endow. **4**(11), 695–701 (2011)
2. Cruz, I., Palmonari, M., Caimi, F., Stroe, C.: Towards on the go matching of linked open data ontologies. In: Workshop on Discovering Meaning On The Go in Large Heterogeneous Data, p. 37 (2011)
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
4. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 402–417. Springer, Heidelberg (2010)
5. Jain, P., Yeh, P.Z., Verma, K., Vasquez, R.G., Damova, M., Hitzler, P., Sheth, A.P.: Contextual ontology alignment of LOD with an upper ontology: a case study with proton. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 80–92. Springer, Heidelberg (2011)
6. Parundekar, R., Ambite, J.L., Knoblock, C.A.: Aligning unions of concepts in ontologies of geospatial linked data. In: Proceedings of the Terra Cognita 2011 Workshop in Conjunction with the 10th International Semantic Web Conference, Bonn, Germany (2011)
7. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Aligning geospatial ontologies on the linked data web. In: Proceedings of the GIScience Workshop on Linked Spatiotemporal Data, Zurich, Switzerland (2010)
8. Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 598–614. Springer, Heidelberg (2010)

9. Spiliopoulos, V., Valarakos, A.G., Vouros, G.A.: *CSR*: Discovering subsumption relations for the alignment of ontologies. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 418–431. Springer, Heidelberg (2008)
10. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 124–138. Springer, Heidelberg (2011)

Extraction of Historical Events from Wikipedia

Daniel Hienert(✉) and Francesco Luciano

GESIS – Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8,
50667 Cologne, Germany
{daniel.hienert,francesco.luciano}@gesis.org

Abstract. The DBpedia project extracts structured information from Wikipedia and makes it available on the web. Information is gathered mainly with the help of infoboxes that contain structured information of the Wikipedia article. A lot of information is only contained in the article body and is not yet included in DBpedia. In this paper we focus on the extraction of historical events from Wikipedia articles that are available for about 2,500 years for different languages. We have extracted about 121,000 events with more than 325,000 links to DBpedia entities and provide access to this data via a Web API, SPARQL endpoint, Linked Data Interface and in a timeline application.

Keywords: Historical events · Wikipedia · DBpedia · Linked data

1 Introduction

The Wikipedia project is a community-based encyclopedia with about 19.7 million articles in 268 languages.[1] The DBpedia project extracts the most relevant facts of Wikipedia articles with the help of infoboxes and gives access to 3.64 million things and their relations. Major historical events like the *Olympic Games in Sydney 2000* have its own Wikipedia article and are therefore also available in DBpedia. Historical events in Wikipedia are also collected in articles for each year like the article for the year 2011 (http://en.wikipedia.org/wiki/2011). The articles contain bullet-point lists with historical events categorized by month and/or categories and subcategories. The events themselves consist of a date and a description with links to other Wikipedia articles. All together, these articles and lists provide an outline of several thousand years of human history. Because the events are listed in the article body, they are not yet included in DBpedia and cannot be queried in a structured way.

Historical events are a good supplement for linked data as it involves persons, places and other entities available in DBpedia. It can therefore combine different entity types and add a historical component. In conjunction with data from disciplines like economy, social science or politics, historical events can provide added value, i.e. give background information for certain phenomena.

We have extracted these events from three different language versions of Wikipedia. In total, this results in 121,821 events with 325,693 links to other Wikipedia articles. Events can be queried via a Web API with results in different formats (XML,

[1] Zachte, Erik. Wikipedia-Statistik. http://stats.wikimedia.org/DE/TablesArticlesTotal.htm.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 16–28, 2015.
DOI: 10.1007/978-3-662-46641-4_2

JSON, N3) and via a SPARQL endpoint with links to DBpedia resources. They are applied in a Linked Data Interface and in a timeline application.

In this paper we describe related work in Sect. 2. Existing data sources for historical events and their granularity are discussed in Sect. 3. Our own approach to extract historical events from Wikipedia is presented in Sect. 4. In Sect. 5 we will give information about the access to historical events via Web API and SPARQL endpoint and the representation and application in a Linked Data Interface and in a timeline application. We will conclude in Sect. 6.

2 Related Work

Concepts, semantic relations, facts and descriptions from Wikipedia are used as a resource for several research areas like natural language processing, information retrieval, information extraction and ontology building. Medelyan et al. [11] give a comprehensive overview of research activity in these areas.

The field of information extraction tries to extract meaningful relations from unstructured data like raw article texts in Wikipedia. Methods can be grouped in (1) processing raw text (2) processing structured parts and (3) the recognition and typing of entities.

Processing raw text extracts relations from the articles full text to identify relations between Wikipedia entities. Known relations are used as seed patterns to identify new relations in a large text corpus. The use of linguistic structures [14], selectional constraints [17] and lexical features [18] can enhance the basic approach. WordNet [12, 21], Wikipedia infobox content [22] and web resources [20] can be used as positive examples for the classification to improve the extraction process.

Processing structured parts such as infoboxes and the category structure are used successfully for building large knowledge bases. Important outcomes are the YAGO and the DBpedia project. YAGO [15] is an ontology automatically build from Wikipedia's category system and matched to the WordNet [8] taxonomy with more than 10 million entities and about 80 million facts about these entities. The DBpedia project [1] extracts structured information like attribute-value pairs from Wikipedia infoboxes and transforms them into RDF triples. The resulting knowledge base contains classified entities like persons, places, music albums, films, video games, organizations, species and diseases organized in an ontology. The data set contains information from the English version of Wikipedia, but also from other language editions. The SPARQL endpoint allows querying the knowledge base.

Recognition and typing of entities is important for the fields of information retrieval and question answering. Work has been done on labeling entities like persons, organization, locations etc. with different approaches like machine learning or mapping to WordNet with different foci like the articles categorization, description or the whole article [3, 4, 16]. Dakka and Cucerzan [6] use different classifiers (bag-of-words, page structure, abstract, titles, entity mentions) to label Wikipedia pages with named entity tags like animated entities (human, non-human), organization, location or miscellaneous, among them events or works of art. In the system of Bhole [2] Wikipedia articles are classified as persons, places or organizations on the basis of Support Vector

Machines. Text mining is used to extract links and event information for these entities. To extract events, the system first extracts plain text from Wiki markup, then with the help of sentence boundary disambiguation extracts sentences containing a date. These are used and linked as events for the article and shown on a timeline. Exner and Nugues [7] extracted events based on semantic parsing from Wikipedia text and converted them into the LODE model. They applied their system to 10 % of the English Wikipedia and extracted 27,500 events with links to external resources like DBpedia and GeoNames. Several more work has been done extracting events from Wikipedia articles with the help of NLP and statistical methods like in [5] and [19].

There are various toolkits and applications for the visualization of time events, for example, the Google News Timeline,[2] the BBC Timeline[3] or the Simile Timeline.[4] The Google News Timeline aggregates data from various sources like news archives, magazines, newspapers, blogs, baseball scores, Wikipedia events/births/deaths and media information from Freebase. The formerly research project has now gone to a productive web service with mainly news categories that can be customized by users. The BBC Timeline shows historical events from the United Kingdom on a flash-based timeline that can be searched and browsed. The Simile Timeline is an open source web toolkit that allows users to create customized timelines and feed it with own data.

3 Historical Events in Wikipedia and DBpedia

In this section we give an overview of the availability and coverage of historical events in Wikipedia and DBpedia.

3.1 Wikipedia

Wikipedia organizes information about historical events in three different ways:

(1) Major historical events have their own article like for example the *Deepwater Horizon oil spill*[5] in 2010 or the *2008 Summer Olympics*[6] in Beijing. Descriptions, details, links and sub events are integrated in the article text like for example *"On July 15, 2010, the leak was stopped by capping the gushing wellhead…"*. This approach is common for all language versions, also if similar events may be differently labeled, have different URLs and content or maybe missing in some language versions depending on their importance.

(2) Historical events are also organized in a system of time units (i.e. years) and in a combination of topics and years (i.e. 2010 in architecture). The time units system is available in all language versions, the combination of topics and year only in

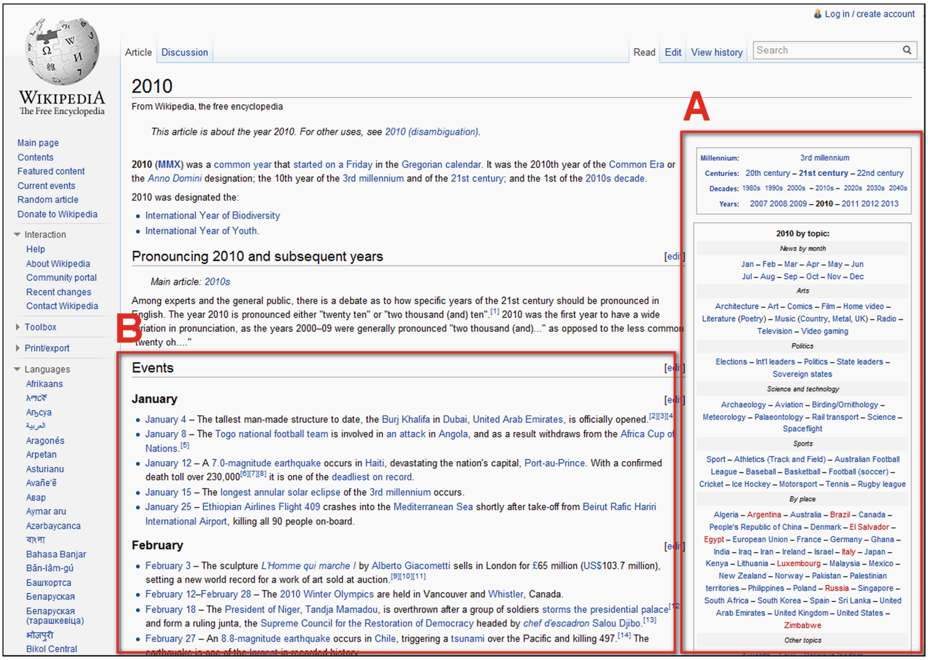
[2] Google. Google News timeline http://newstimeline.googlelabs.com/.

[3] BBC. History timeline. http://www.bbc.co.uk/history/interactive/timelines/british/index.shtml.

[4] SIMILE Project. SIMILE timeline. http://www.simile-widgets.org/timeline.

[5] http://en.wikipedia.org/wiki/Deepwater_Horizon_oil_spill.

[6] http://en.wikipedia.org/wiki/2008_Summer_Olympics.



2010

From Wikipedia, the free encyclopedia

This article is about the year 2010. For other uses, see 2010 (disambiguation).

2010 (**MMX**) was a common year that started on a Friday in the Gregorian calendar. It was the 2010th year of the Common Era or the *Anno Domini* designation; the 10th year of the 3rd millennium and of the 21st century; and the 1st of the 2010s decade.

2010 was designated the:

- International Year of Biodiversity
- International Year of Youth.

Pronouncing 2010 and subsequent years

Main article: 2010s

Among experts and the general public, there is a debate as to how specific years of the 21st century should be pronounced in English. The year 2010 is pronounced either "twenty ten" or "two thousand (and) ten".[1] 2010 was the first year to have a wide variation in pronunciation, as the years 2000–09 were generally pronounced "two thousand (and)..." as opposed to the less common "twenty oh...."

Events

January

- January 4 – The tallest man-made structure to date, the Burj Khalifa in Dubai, United Arab Emirates, is officially opened.[2][3][4]
- January 8 – The Togo national football team is involved in an attack in Angola, and as a result withdraws from the Africa Cup of Nations.[5]
- January 12 – A 7.0-magnitude earthquake occurs in Haiti, devastating the nation's capital, Port-au-Prince. With a confirmed death toll over 230,000[6][7][8] it is one of the deadliest on record.
- January 15 – The longest annular solar eclipse of the 3rd millennium occurs.
- January 25 – Ethiopian Airlines Flight 409 crashes into the Mediterranean Sea shortly after take-off from Beirut Rafic Hariri International Airport, killing all 90 people on-board.

February

- February 3 – The sculpture *L'Homme qui marche I* by Alberto Giacometti sells in London for £65 million (US$103.7 million), setting a new world record for a work of art sold at auction.[9][10][11]
- February 12–February 28 – The 2010 Winter Olympics are held in Vancouver and Whistler, Canada.
- February 18 – The President of Niger, Tandja Mamadou, is overthrown after a group of soldiers storms the presidential palace[12] and form a ruling junta, the Supreme Council for the Restoration of Democracy headed by *chef d'escadron* Salou Djibo.[13]
- February 27 – An 8.8-magnitude earthquake occurs in Chile, triggering a tsunami over the Pacific and killing 497.[14] The

Millennium:	3rd millennium
Centuries:	20th century – **21st century** – 22nd century
Decades:	1980s 1990s 2000s – 2010s – 2020s 2030s 2040s
Years:	2007 2008 2009 – **2010** – 2011 2012 2013

2010 by topic:

News by month

Jan – Feb – Mar – Apr – May – Jun
Jul – Aug – Sep – Oct – Nov – Dec

Arts

Architecture – Art – Comics – Film – Home video – Literature (Poetry) – Music (Country, Metal, UK) – Radio – Television – Video gaming

Politics

Elections – Int'l leaders – Politics – State leaders – Sovereign states

Science and technology

Archaeology – Aviation – Birding/Ornithology – Meteorology – Palaeontology – Rail transport – Science – Spaceflight

Sports

Sport – Athletics (Track and Field) – Australian Football League – Baseball – Basketball – Football (soccer) – Cricket – Ice Hockey – Motorsport – Tennis – Rugby league

By place

Algeria – Argentina – Australia – Brazil – Canada – People's Republic of China – Denmark – El Salvador – Egypt – European Union – France – Germany – Ghana – India – Iraq – Iran – Ireland – Israel – Italy – Japan – Kenya – Lithuania – Luxembourg – Malaysia – Mexico – New Zealand – Norway – Pakistan – Palestinian territories – Philippines – Poland – Russia – Singapore – South Africa – South Korea – Spain – Sri Lanka – United Arab Emirates – United Kingdom – United States – Zimbabwe

Other topics

Fig. 1. The Wikipedia article for the year *2010*: (A) shows two templates with (1) links to millennia, centuries, years and (2) to articles with combinations of topics and years. (B) shows a list of events for the year 2010.

the English version. Applied time units are millennia, centuries, decades, years, months and days. The titles and URLs of these articles are accordingly like *3rd_millennium*, *20th_century*, *2000s*, *2010, June_2010* or *June_10*. There exist also a series of articles for the combination of topic and time. Categories are for example, Arts, Politics, Science.

Science and Technology, Sports, By place etc. with a lot of different subcategories like Architecture, State leaders, Paleontology, Australian Football or Australia. Labels and URLs of these articles are then the combination like *2010_in_architecture* or *2010_in_baseball*. Figure 1a gives an overview of the templates for the year *2010*.

All these articles have in common that they can contain full text but also a list of events. The number of events per time unit (centuries, decades, years, months) differs and events are described on different abstraction levels. So, the article about the 20^{th} *century* contains mainly full text and describes main developments in brief over decades. The article about the *2010s* decade describes developments in politics, disasters, economics etc. in a multi-year view. The *2010* article contains a list of most important events for every month. The article for *February 2010* lists detailed events for every day of the month. Because events are not queried from a database, but are edited by users in the wiki page, they are not similar in different lists. The same event in the list of *2010*, *January_2010*, *January_4* may differ in description and links.

(3) There exist also articles for the collection of events for a specific topic. These articles have titles like *Timeline of …* and contain a list of events analog to year articles. Examples are the *Timeline of the Deepwater Horizon oil spill*, the *Timeline of World War II* or the *Subprime crisis impact timeline*.

In this paper we focus on the extraction of events from articles for years, because these are available in different languages and offer a good compromise between number and abstractness of events. The corresponding text section begins with a heading in the actual language like *Events*. Then, depending on the language version follow categories, subcategories, months and bullet-point lists with events. The events have a date at the beginning, followed by a short text describing the event with links to other Wikipedia articles (see Fig.1b).

3.2 DBpedia

DBpedia only extracts events from Wikipedia if the resource has its own article. All extracted events from Wikipedia can be queried with the DBpedia SPARQL Endpoint. Events can be identified either by (1) being of the DBpedia ontology type *Event* or (2) containing a date attribute (*date*, *startDate* etc.).

The ontology type *Event* lists eight subcategories like *Convention, Election, FilmFestival, MilitaryConflict, MusicFestival, SpaceMission, SportsEvent* and *YearInSpaceflight*. The category *SportsEvent* lists further eight sub types: *FootballMatch*, *GrandPrix*, *MixedMartialArtsEvent*, *Olympics*, *Race*, *SoccerTournament*, *WomensTennisAssociationTournament* and *WrestlingEvent*. Table 1 provides an overview of the number of resources for the type *Event* and its sub types.

Table 1. Number of DBpedia resources for the type *Event* and its sub types.

DBpedia ontology class	Number of resources
Event	20,551
MilitaryConflict	9,725
SportsEvents	5,020
Election	3,778
FootballMatch	1,536
MixedMartialArtsEvent	1,068
WrestlingEvent	946
MusicFestival	752
Convention	454
SpaceMission	419
Race	415
FilmFestival	350
Olympics	58
WomensTennisAssociationTournament	54
YearInSpaceflight	53

The type *Event* contains about 20,551 entities, whereby already the half are military conflicts and another quarter to a half are sport events. Typed events in DBpedia are thus distributed over only a few categories that have in most cases a military or sport character.

DBpedia resources that contain a date attribute or property could also be handled as an event. We have extracted all attributes from the DBpedia ontology that contain the term *date* and determined with SPAQRL queries the count of these resources. Table 2 gives an overview of the top ten properties. It can be seen that most entities containing a date attribute are distributed over only a few domains like person, work or battles analog to typed events. Furthermore, a lot of dates are handled as properties outside the ontology, as for example the *dbpprop:spillDate* from the before mentioned *Deepwater Horizon* resource.

Table 2. Top ten properties containing the string *date* with number of resources and their main type.

DBpedia ontology property	Number of resources	Main entity type
birthDate	468,852	Person
deathDate	187,739	Person
releaseDate	156,553	Work/Media/ Software
date	9,872	Battle
recordDate	9,604	Album
openingDate	8,700	Architecture
formationDate	7,577	Sport, Societies
startDate	3,508	Election
publicationDate	2,129	Book
latestReleaseDate	1,516	Software

Historical events in DBpedia are distributed over only a few categories: military battles and sport events for the type *Event* or *Person/Work/Battle* for entities having a date attribute. This is a very one-sided distribution and current events are totally missing. Depending on that, combined queries for events associated to a person, place or thing are not possible or with poor results. But, historical events already exist in Wikipedia with a wide temporal coverage, with links to other entities and for different languages. That is why we have chosen to implement our own software to make this data set available and link them to DBpedia entities.

4 Extracting Historical Events from Wikipedia

We have created software to crawl, parse and process lists of historical events from Wikipedia articles for years (see Fig. 2 for the overall extraction and transformation process). The software can be parameterized for specific Wikipedia language versions and temporal restrictions. Language-dependent specifications are loaded directly from a configuration file that holds all keywords and patterns used for parsing. This way, new

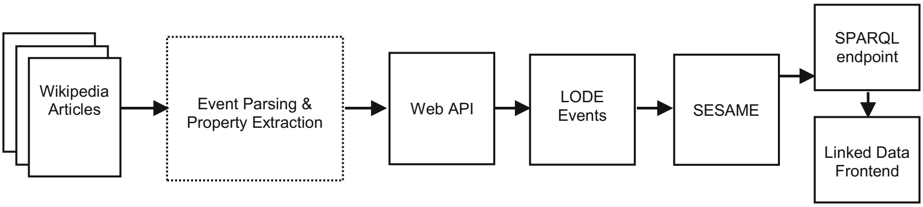


Fig. 2. Extraction and transformation pipeline.

language versions can easily be integrated and adapted without changing the program code.

4.1 Extraction Algorithm

Beginning from a start year the software retrieves all articles from the Wikipedia API. Articles are returned in Wiki markup, a lightweight markup language. Text sections with events are then identified by language-dependent patterns. Depending on the language also categories and subcategories are provided as headings for events. For example, in the German Wikipedia events have been categorized in politics and world events, economy, science and technology and culture. The individual events are then parsed, broken down into its components and saved in a MySQL database. The decomposition is done on the basis of regular expressions depending on the language and the settings in the config file. Altogether we have extracted 121,812 historical events in different languages like German (36,063), English (32,943), Spanish (18,436), Romanian (9,745), Italian (6,918), Portuguese (6,461), Catalan (6.442), Turkish (3,084) and Indonesian (1,720) with a temporal coverage from 300 BC to 2013.

For example, in the English Wikipedia Version the event section is identified by the entry pattern “== .*Events.* ==” and the exit patterns “== .*Deaths.* ==” or “== .*Births.* ==”. Individual events are recognized by four different patterns like i.e. “*.*\[\[\D*\s\d*\]\].*” for a standard event with enumeration sign, date as link and the following description. In the event, the date field and the description are separated by a “&ndash;” or a “&mdash;” character. The date field is decomposed by the pattern “Month Day”. The description can contain links to Wikipedia articles. These links are marked in the wiki markup with double brackets. If the link text is different to the Wikipedia title, text and Wikipedia title are divided by a pipe symbol. We have extracted about 325,693 links from all events, which means an average of about 2.7 links per event.

For each event an URL for an illustrating image from Wikipedia has been added. Lists of events very rarely contain images but nearly every event includes links to other Wikipedia articles. For all events, (1) the software iterates through the links of the event and queries the Wikipedia API for images included in the article. (2) The API returns all images in alphabetical order. (3) If the image is not a standard image (like images for disambiguation sites) the API is queried again to receive the URL of that image. The API offers the possibility to parameterize the URL with an image width or height.

The width string (i.e. -150px-) is included into the URL and resizes the image directly on server side. We choose the width of 150 pixels to have the same image size for all images and to reduce download time in applications. The size in the URL string can later be modified by any other size. This method for gathering images is very fast with a high confidence of getting an image. However, because images are returned in alphabetical order and not in the order of their appearance in the article, the image is not always fitting perfectly to the event. 63,158 image URLs (33,177 distinct ones) had been added to events with a coverage of 89 %.

4.2 Challenges and Evaluation

We faced various difficulties in the parsing of events from the wiki markup. Historical events are entered and edited by human users into the wiki page. In general events are entered analog to a given template, for example analog to the structure of the wiki page from the year before. But in some cases events are entered in another structure because users want to assemble several events under one heading and include the month in the heading but not in the events. Only the formatting of dates offers various variations, for example, for different country styles (English vs. German), structures (date or month in the heading), different formatting of time periods, missing date parts (i.e. only the month), missing separators etc. In a corpus of about 20,000 wiki pages nearly all variations exits and must be handled. Since our algorithm relies on fixed regular expression, each variation of these patterns needs to be included or withdrawn if only a few events are affected.

We faced these issues by different approaches: (1) outsourcing regular expression and keywords to a config file and providing different parts for different Wikipedia language versions. For each regular expression, for example, to parse the date, variations are allowed. This way, alternative patterns for each language versions can be included and applied. (2) Implementing a quality management in the source code that (a) wraps all critical parts like event detection, date and link extraction in the code with try/catch to catch all events that could not be handled and write them to a log file. (b) Calculating the extraction quotient by counting events that are written into the database and dividing them by the total number of events that exist for the language (by counting each line in the events section starting with an enumeration sign). (c) Checking the quality of atomized events in the database. This way, the extraction algorithm can be optimized outside the source code by checking which cluster of events failed in the log file, optimizing and adding variations of regular expression and therefore improve the overall recall. For the individual languages the algorithm achieved the following extraction quotients: German (98,97 %), Spanish (94,12 %), Turkish (91,87 %), Portuguese (91,74 %), English (86,20 %), Catalan (85,69 %), Indonesian (80,19 %), Italian (75,81 %) and Romanian (74,73 %). The quality of the events in the database turned out to be very good, as only positively evaluated events were written to the database.

5 Provision and Application of Historical Events

We provide access to historical events via a Web API, SPARQL endpoint, Linked Data Interface and in a timeline application.

5.1 Web API

The Web API[7] is a web service that returns results in standardized formats. Users can query the database with URL parameters like *begin_date*, *end_date, category*, *language*, *query, html, links, limit, order* and *format*. This allows the filtering of events by time or category, but also a free search for events that belong to a certain topic. Results are returned in XML, JSON or RDF/N3 format and can therefore be easily processed further. With the parameter *html* = *true* the description will include links to Wikipedia in HTML format, the parameter *links* = *true* return all links in a separate XML node. For example, the following URLs return historical events for (1) a specific time period, (2) for the keyword *Egypt* or (3) for the German category *Kultur*:

(1) http://www.vizgr.org/historical-events/search.php?begin_date=19450000&end_date=19501231
(2) http://www.vizgr.org/historical-events/search.php?query=Egypt
(3) http://www.vizgr.org/historical-events/search.php?category=Kultur

5.2 Modeling of Events, SPARQL Endpoint and Linked Data Interface

There are several ontologies for the modeling of events in RDF like EVENT,[8] LODE,[9] SEM,[10] EventsML,[11] and F,[12] see [9] for a comparison of these models. We have chosen the LODE [13] ontology, because it is domain-independent and a light weighted structure to represent events. All English events have been transformed with the help of the Web API to the LODE ontology in N3 format (see Fig. 3). Then, the data set has been imported into the Sesame repository,[13] which provides the SPARQL endpoint.[14] The linked data interface Pubby[15] is set upon the SPARQL endpoint to generate a HTML representation of the events via dereferencing the URIs. Because data is semantified users can make complex queries against the SPARQL endpoint. Users can query all events that are event of any specific DBpedia entity like a person, place or thing. For example, all events associated with *Barack Obama*, *The White House* or *Basketball* can be queried. With a federated SPARQL query or by integrating the data into the DBpedia data set, much more complex queries like "*Give me all events that are associated with Presidents of the United States between 1950 and 2000*".

[7] http://www.vizgr.org/historical-events.

[8] http://motools.sourceforge.net/event/event.html.

[9] http://linkedevents.org/ontology/.

[10] http://semanticweb.cs.vu.nl/2009/11/sem/.

[11] http://www.iptc.org/site/News_Exchange_Formats/EventsML-G2/.

[12] http://isweb.uni-koblenz.de/eventmodel.

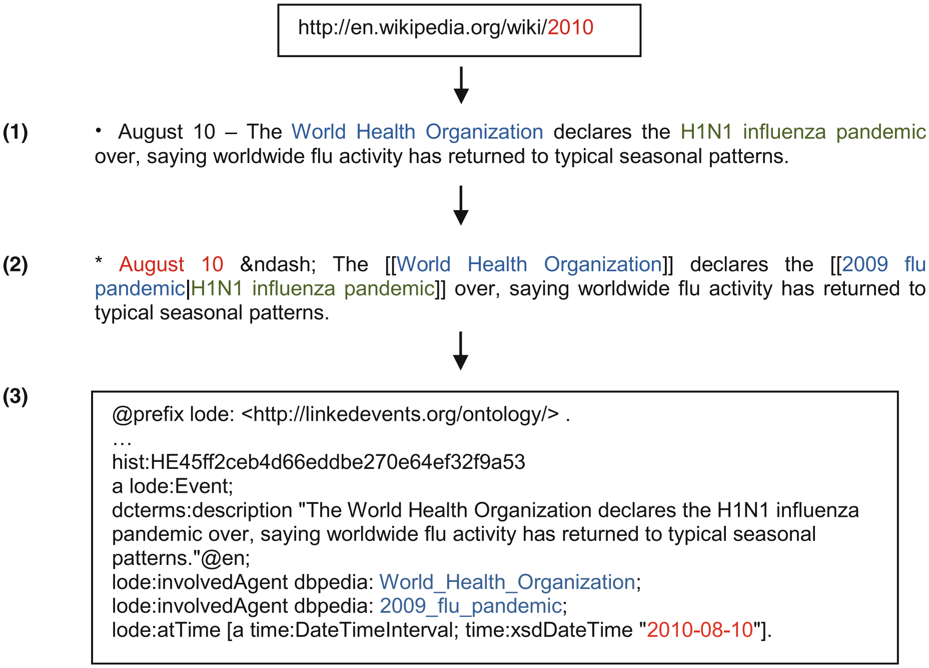
[13] http://lod.gesis.org/historicalevents/sparql.

[14] http://lod.gesis.org/historicalevents/.

[15] http://www4.wiwiss.fu-berlin.de/pubby/.

5.3 Timeline Application

We have developed an application that shows historical events on a timeline (compare Fig. 4). Compared to the presentation on distributed sites in Wikipedia it has the advantage that the user interface is more intuitive and user-friendly. Analog to the presented timelines in Sect. 2 users can easily scroll the years, zoom to a decade or search for a specific event with a keyword search. The timeline then scrolls to the matching event. We have created a simple version of a timeline[16] with the Vizgr visualization toolkit [10]. The timeline shows all extracted historical events with images from 300 B.C. to today from the English, German or Italian version of Wikipedia. Users can browse the timeline, can view events and can click on links to be forwarded to the Wikipedia article.



http://en.wikipedia.org/wiki/2010

(1) • August 10 – The World Health Organization declares the H1N1 influenza pandemic over, saying worldwide flu activity has returned to typical seasonal patterns.

(2) * August 10 &ndash; The [[World Health Organization]] declares the [[2009 flu pandemic|H1N1 influenza pandemic]] over, saying worldwide flu activity has returned to typical seasonal patterns.

(3)

```
@prefix lode: <http://linkedevents.org/ontology/> .
…
hist:HE45ff2ceb4d66eddbe270e64ef32f9a53
a lode:Event;
dcterms:description "The World Health Organization declares the H1N1 influenza
pandemic over, saying worldwide flu activity has returned to typical seasonal
patterns."@en;
lode:involvedAgent dbpedia: World_Health_Organization;
lode:involvedAgent dbpedia: 2009_flu_pandemic;
lode:atTime [a time:DateTimeInterval; time:xsdDateTime "2010-08-10"].
```

Fig. 3. Extraction of an event and transformation to LODE: (1) the event in the HTML view, (2) the wiki markup of the event, (3) in the LODE event model.

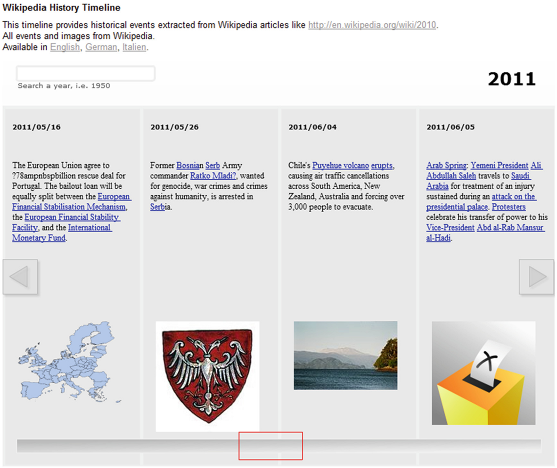
6 Discussion and Conclusion

So far, there are two approaches to extract events from Wikipedia: (1) extracting events from the main article text and (2) the creation of events from the article itself.

[16] http://vizgr.org/historical-events/timeline.

Extracting events from the Wikipedia main article as done by [2, 8] mainly uses NLP methods and semantic parsing to identify phrases containing a date attribute and relating them to the typed article entity. This has several disadvantages such as the time-consuming and complex processing and based on the methods there is a certain error-rate that events are extracted and connected correctly. The main outcome of this method is to extract events specifically for one topic, the article entity itself. That's why the resulting data set can't contain historical events important for one time unit like a year.

We have shown that existing events in the DBpedia data set are distributed over only a few categories: military battles and sport events for the ontology type *Event* or *Person/Work/Battle* for entities having a date attribute. This is a very one-sided distribution, the number of historical events is low and querying of events for a time period or for a specific resource is complex.



Wikipedia History Timeline

This timeline provides historical events extracted from Wikipedia articles like http://en.wikipedia.org/wiki/2010.
All events and images from Wikipedia.
Available in English, German, Italien.

Search a year, i.e. 1950

2011

2011/05/16

The European Union agree to ?78ampnbspbillion rescue deal for Portugal. The bailout loan will be equally split between the European Financial Stabilisation Mechanism, the European Financial Stability Facility, and the International Monetary Fund.

2011/05/26

Former Bosnian Serb Army commander Ratko Mladi?, wanted for genocide, war crimes and crimes against humanity, is arrested in Serbia.

2011/06/04

Chile's Puyehue volcano erupts, causing air traffic cancellations across South America, New Zealand, Australia and forcing over 3,000 people to evacuate.

2011/06/05

Arab Spring: Yemeni President Ali Abdullah Saleh travels to Saudi Arabia for treatment of an injury sustained during an attack on the presidential palace. Protesters celebrate his transfer of power to his Vice-President Abd al-Rab Mansur al-Hadi.

Fig. 4. The Wikipedia History Timeline.

In our own approach we rely on existing structures in Wikipedia. Wikipedia collects historical events of different granularity in lists for centuries, years, months and on a daily basis. These lists are user-maintained with a high actuality and correctness, so that events of yesterday are already available with links to other Wikipedia articles. However, because each of these events does not have its own Wikipedia article, they are not available in DBpedia. We have parsed and processed historical events on a yearly basis for different language versions and make them available by a Web API, a SPARQL endpoint and a Linked Data Interface. This allows the simple and fast querying, but also complex queries. The created data set has a wide temporal coverage from 300 BC to 2013 and exists for different languages. The granularity of events is more evenly distributed over years and not so much fixed on individual topics as users already have chosen which events are important for a year. This can make the use of these events easier in end-user applications. Historical events build an important typed category analog to persons, places, work or organizations already available in DBpedia.

Furthermore, they are a linking hub because each historical event involves dimensions like time and space, but also agents like persons, things and other entities. We have extracted over 325,000 links from 121,000 events. This means, combined with the DBpedia data set about 325,000 DBpedia entities are now connected by about 121,000 historical events.

In future work, we want to extract events for more language versions, on the basis of other time units like days, centuries or millennia and in a combination with topics. This enriches the data set to different granularity that can be used in end-user applications. We also want to add a live module that parses the latest added events and makes them available directly in our data set. The next step is to find relations between events among different languages and granularity levels.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) The Semantic Web. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Bhole, A., et al.: Extracting named entities and relating them over time based on wikipedia. Informatica (Slovenia) **31**(4), 463–468 (2007)
3. Buscaldi, D., Rosso, P.: A bag-of-words based ranking method for the wikipedia question answering task. In: Peters, C., et al. (eds.) Evaluation of Multilingual and Multi-modal Information Retrieval. LNCS, vol. 4730, pp. 550–553. Springer, Heidelberg (2006)
4. Buscaldi, D., Rosso, P.: A comparison of methods for the automatic identification of locations in wikipedia. In: Proceedings of the 4th ACM workshop on Geographical information retrieval, pp. 89–92. ACM, New York, NY, USA (2007)
5. Chasin, R.: Event and Temporal Information Extraction towards Timelines of Wikipedia Articles. Simile, pp. 1–9 (2010)
6. Dakka, W., Cucerzan, S.: Augmenting Wikipedia with Named Entity Tags. In: Proceedings of IJCNLP 2008 (2008)
7. Exner, P., Nugues, P.: Using semantic role labeling to extract events from Wikipedia. In: Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011). Workshop in Conjunction with the 10th International Semantic Web Conference 2011 (ISWC 2011). Bonn (2011)
8. Fellbaum, C. (ed.): WordNet An Electronic Lexical Database. The MIT Press, Cambridge (1998)
9. van Hage, W.R., et al.: Design and use of the simple event model (SEM). Web Semant. Sci. Serv. Agents World Wide Web **9**, 2 (2011)
10. Hienert, D., et al.: VIZGR: combining data on a visual level. In: Proceedings of the 7th International Conference on Web Information Systems and Technologies (WEBIST) (2011)
11. Medelyan, O., et al.: Mining meaning from wikipedia. Int. J. Hum.-Comput. Stud. **67**(9), 716–754 (2009)
12. Ruiz-Casado, M., et al.: Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from wikipedia. Data Knowl. Eng. **61**(3), 484–499 (2007)
13. Shaw, R., Troncy, R., Hardman, L.: LODE: Linking Open Descriptions of Events. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) The Semantic Web. LNCS, vol. 5926, pp. 153–167. Springer, Heidelberg (2009)

14. Suchanek, F.M., et al.: Combining linguistic and statistical analysis to extract relations from web documents. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 712–717. ACM, New York, NY, USA (2006)
15. Suchanek, F.M., et al.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp. 697–706. ACM, New York, NY, USA (2007)
16. Toral, A., Munoz, R.: A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In: EACL 2006 (2006)
17. Wang, G., Zhang, H., Wang, H., Yu, Y.: Enhancing relation extraction by eliciting selectional constraint features from wikipedia. In: Kedad, Z., Lammari, N., Métais, E., Meziane, F., Rezgui, Y. (eds.) Natural Language Processing and Information Systems. LNCS, vol. 4592, pp. 329–340. Springer, Heidelberg (2007)
18. Wang, G., Yu, Y., Zhu, H.: PORE: positive-only relation extraction from wikipedia text. In: Aberer, K., et al. (eds.) The Semantic Web. LNCS, vol. 4825, pp. 580–594. Springer, Heidelberg (2007)
19. Woodward, D.: Extraction and Visualization of Temporal Information and Related Named Entities from Wikipedia. Springs, pp. 1–8 (2001)
20. Wu, F., et al.: Information extraction from Wikipedia: moving down the long tail. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge discovery and data mining, pp. 731–739. ACM, New York, NY, USA (2008)
21. Wu, F., Weld, D.S.: Automatically refining the wikipedia infobox ontology. In: Proceeding of the 17th International Conference on World Wide Web, pp. 635–644. ACM, New York, NY, USA (2008)
22. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: Proceedings of the sixteenth ACM Conference on Information and Knowledge Management, pp. 41–50. ACM, New York, NY, USA (2007)

Capturing Interactive Data Transformation Operations Using Provenance Workflows

Tope Omitola[1](✉), André Freitas[2], Edward Curry[2], Séan O'Riain[2], Nicholas Gibbins[1], and Nigel Shadbolt[1]

[1] Web and Internet Science (WAIS) Research Group Electronics and Computer Science, University of Southampton, Southampton, UK
{tobo,nmg,nrs}@ecs.soton.ac.uk

[2] Digital Enterprise Research Institute (DERI), National University of Ireland, Galway, Ireland
{andre.freitas,ed.curry,sean.oriain}@deri.org

Abstract. The ready availability of data is leading to the increased opportunity of their re-use for new applications and for analyses. Most of these data are not necessarily in the format users want, are usually heterogeneous, and highly dynamic, and this necessitates data transformation efforts to re-purpose them. Interactive data transformation (IDT) tools are becoming easily available to lower these barriers to data transformation efforts. This paper describes a principled way to capture data lineage of interactive data transformation processes. We provide a formal model of IDT, its mapping to a provenance representation, and its implementation and validation on Google Refine. Provision of the data transformation process sequences allows assessment of data quality and ensures portability between IDT and other data transformation platforms. The proposed model showed a high level of coverage against a set of requirements used for evaluating systems that provide provenance management solutions.

Keywords: Linked Data · Public open data · Data publication · Data consumption · Semantic Web · Workflow · Extract-Transform-Load · Provenance · Interactive data transformation

1 Introduction

The growing availability of data on the Web and in organizations brings the opportunity to reuse existing data to feed new applications or analyses. In order to reuse existing data, users must perform data transformations to repurpose data for new requirements. Traditionally, data transformation operations have been supported by data transformation scripts organized inside ETL (Extract-Transform-Load) environments or by ad-hoc applications. Currently, users developing data transformation programs follow a typical software development cycle, taking data samples from datasets, and developing the transformation logic, testing and debugging. These approaches are problematic in emerging scenarios such

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 29–42, 2015.
DOI: 10.1007/978-3-662-46641-4_3

as the Linked Data Web where the reduction of the barriers for producing and consuming new data brings increasing challenges in coping with heterogeneous, high-volume, and dynamic data sources. In these scenarios, the process of *interactive data transformation* emerges as a solution to scale data transformation.

Recently platforms, such as Google Refine[1], are exploring user interface and interaction paradigms for defining data transformations. These platforms allow users to operate over data using Graphical User Interface (GUI) elements instead of scripts for data transformation. By providing a set of pre-defined operations and instant feedback mechanism for each iteration of the data transformation process, this model defines a powerful approach to data transformation and curation. However, despite its practical success, the assumptions behind platforms such as Google Refine have not been modeled nor formalized in the literature. As the need for curation increases in data abundant environments, the need to model, understand, and improve data curation and transformation processes and tools emerges as a critical problem. A foundational model of IDT that (a) brings out the underlying assumptions that IDT platforms use, (b) makes explicit how IDT relates to provenance, and (c) helps IDT platforms be comparable and thereby helping in defining interfaces for their interoperability, would be highly useful.

The process of data transformation is usually positioned as one element in a connected pipeline which needs to be integrated with different processes, to form a **workflow**. Currently, the set of transformation operations present in IDT platforms are not materialized in a way that could allow its use in contexts outside the IDT environment. A transformation process normally involves a data object that is being transformed, and the transformation procedure itself. *Provenance* is the contextual metadata describing the origin or source of that data. *Prospective provenance* provides mechanisms to describe generic workflows which could be used to materialize (future) data transformations. *Retrospective provenance* captures past workflow execution and data derivation information to provide useful context for what had happened up to the present state and time.

This paper investigates the two complementary perspectives described above, and an approach for expressively capturing and persisting provenance in IDT platforms. A provenance extension for the Google Refine platform is implemented and used to validate the proposed solution. The supporting provenance model focuses on the maximization of interoperability, using the three-layered data transformation model proposed in [6] and uses Semantic Web standards to persist provenance data.

The contributions of this work include: (a) a formal model of IDT; (b)the use of an ontology-based provenance model for mapping data transformation operations on IDT platforms; (c) the verification of the suitability of the proposed provenance model for capturing IDT transformations, using Google Refine as the IDT platform; and (d) the validation of the system against the set of requirements, from the literature, used for evaluating provenance management provided by IDT platforms.

[1] http://code.google.com/p/google-refine/.

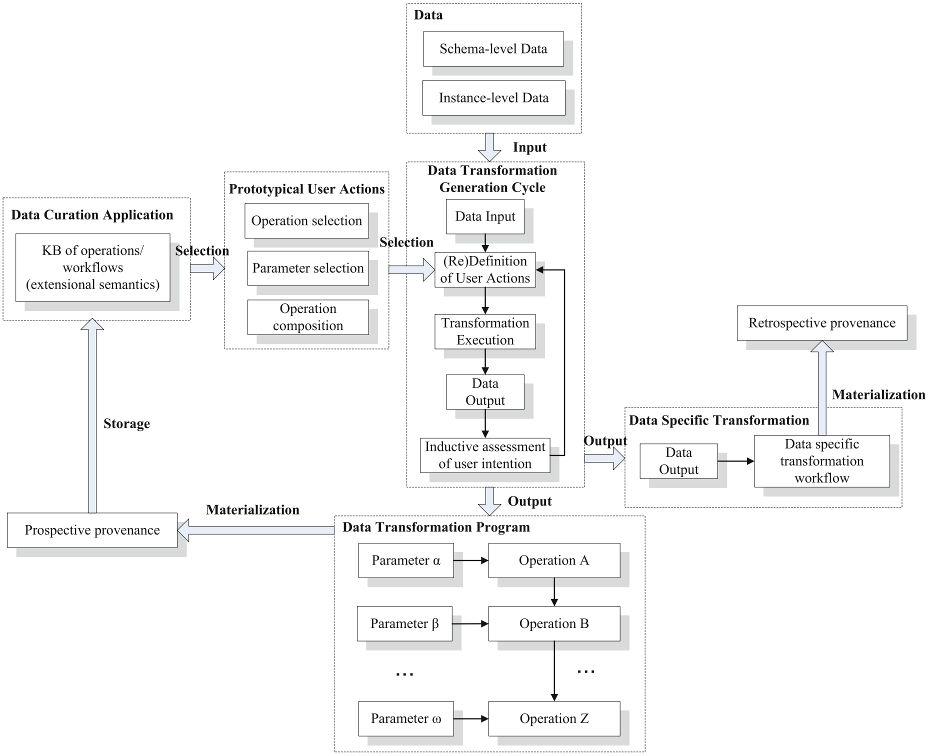


Fig. 1. A model of interactive data transformation

2 Interactive Data Transformations (IDT)

2.1 IDTs, ETL, and Data Curation

Governments across the globe are making their data available in a single Web-based dataspace, such as data.gov.uk, visible across different government agencies, which may also include third-party open data. In this data-rich and heterogeneous environment, data consumers and producers face major data management challenges. Data transformation tasks such as data cleaning, refinement, aggregation, and analysis are usually mediated by programming tasks (in the form of ETL scripts or in generic data transformation programs), making these transformations time consuming and expensive. The increasing availability of third-party datasets brings potential challenges to the centralized nature of traditional ETL environments. Analytical tasks are likely to strongly benefit from data which is present in large numbers of datasets, increasing the cost of the data transformation tasks. With more datasets (such as the open data Web, third-party datasets, etc.) at one's disposal, a large number of ad-hoc small data transformation tasks may emerge.

The concept of interactive data transformation is strongly related to data curation, where human intervention in the data aggregation, cleaning, and transformation is increased. According to Buneman et al. [1], "curated databases are databases that are populated and updated with a great deal of human effort". Curry et al. [2] defines data curation as "the active management and appraisal of data over its life-cycle of interest ... data curation is performed by expert curators responsible for improving the accessibility and quality of data". In a scenario where third-party data becomes abundant on the Web, the process of curating the data for a new intended use becomes an intrinsic part of the data consumption process.

IDTs naturally map to a workflow structure which makes its representation isomorphic to another critical dataspace concern: the capture and representation of provenance. In dataspaces, when datasets cross their original context of use (a specific government agency, for example), data consumers need additional information to determine the trustworthiness and general suitability of the data. For example, it will be very useful for the provenance data representing the processes, artifacts, and agents behind these data transformations to be made available so that data consumers can determine its quality (i.e. answering queries such as *Who modified this value?, From which table is the data derived?*, etc.). The interplay between IDT platforms and provenance addresses important challenges emerging in this new data environment, facilitating the process of data transformation and blurring the borders between ETL and IDT.

2.2 IDT Operations Overview

IDT is defined as the application of a pre-defined set of data transformation operations over a dataset. In IDT, after a transformation operation has been selected from the set of available operations (**operation selection**), users usually need the input of configuration parameters (**parameter selection**). In addition, users can compose different sets of operations to create a data transformation workflow (**operation composition**). Operation selection, parameter selection, and operation composition are the core user actions available for interactive data transformation as depicted in Fig. 1.

In the IDT model, the expected outputs are a data transformation program and a transformed data output (Fig. 1). The transformation program is generated through an iterative *data transformation program generation cycle* where the user selects and composes an initial set of operations, executes the transformation, and assesses the suitability of the output results by an inductive analysis over the materialized output. This is the point where users decide to redefine (*reselect*) the set of transformations and configuration parameters. The organization of operations as GUI elements minimizes program construction time by minimizing the overhead introduced by the need to ensure programming language correctness and compilation/deployment.

The transformation generation cycle generates two types of output: a data output with a data specific transformation workflow, and a data transformation program which is materialized as a prospective provenance descriptor. A **provenance descriptor** is a data structure showing the relationship between the inputs, the outputs, and the transformation operations applied in the process. While the provenance-aware data output is made available for further changes, the prospective provenance workflow is inserted into the KB of available workflows, and can be later reused.

In the next section, we shall present an algebra for Interactive Data Transformation.

2.3 Foundations - an Algebra for Provenance-Based Interactive Data Transformation (IDT)

Despite the practical success of IDT tools, such as Google Refine, for data transformation and curation, the underlying assumptions and the operational behaviour behind such platforms have not been explicitly brought out. Here, we present an algebra of IDT, bringing out the relationships between the inputs, the outputs, and the functions facilitating the data transformations.

Definition 1: Provenance-based Interactive Data Transformation Engine $\mathfrak{G}$. A provenance-based Interactive Data Transformation Engine, $\mathfrak{G}$, consists of a set of transformations (or activities) on a set of datasets generating outputs in the form of other datasets or events which may trigger further transformations.

$\mathfrak{G}$ is defined as a tuple,

$$\mathfrak{G} \; = < \mathbb{D}, (D \cup V), \mathbb{I}, \mathbb{O}, \Sigma, \sigma, \lambda >$$

where

1. $\mathbb{D}$ is the non-empty set of all datasets in $\mathfrak{G}$,
2. D is the dataset being currently transformed,
3. V is the set of views in $\mathfrak{G}$ (V may be empty),
4. $\mathbb{I}$ is a finite set of input channels (this represents the points at which user interactions start),
5. $\mathbb{O}$ is a finite set of output channels (this represents the points at which user interactions may end),
6. Σ is a finite alphabet of actions (this represents the set of transformations provided by the data transformation engine),
7. σ is a finite set of functions that allocate alphabets to channels (this represents all user interactions), and
8. $\lambda = < \; \mathbb{D} \times \mathbb{O} \rightarrow \Sigma \; >$ is a function, where, in a modal transformation engine, $\lambda(D, O) \in \sigma(O)$ is the dataset that is the output on channel O when D is the dataset being currently transformed.

Definition 2: Interactive Data Transformation Event. An Interactive Data Transformation Event is a tuple,

$$P_{TE} = < D_i, F_{trans}, (D_o \cup V), T_{trans} >$$

where

- D_i is the input dataset for the transformation event,
- D_o is the dataset that is the result of the transformation event,
- V is a view or facet that is a result of the transformation event,
- $D_i \cup D_o \cup V \subseteq \mathbb{D}$
- F_{trans} is the transformation function applied to D_i (applied element-wise), and
- T_{trans} is the time the transformation took place.

Definition 3: Run. A **run** can be informally defined as a function from time to dataset(s) and the transformation applied to those dataset(s). Intuitively, a run is a description of how $\mathfrak{G}$ has evolved over time.

So, a run over $\mathfrak{G}$ is a function, $\mathcal{P} : t \rightarrow < D, f >$ where t is an element in the time domain, D is the state of all datasets and views in $\mathfrak{G}$, and f is the function applied at time, t.

A system $\mathcal{R}$ over $\mathfrak{G}$, i.e. $\mathcal{R}(\mathfrak{G})$, is a set of all runs over $\mathfrak{G}$. We say that $< \mathcal{P}, t >$ is a point in system $\mathcal{R}$ if $\mathcal{P} \in \mathcal{R}$.

$\mathcal{P}$ captures our notion of"prospective provenance".

Definition 4: Trace. Let $\alpha =< \mathcal{P}, t > \in \mathcal{R}(\mathfrak{G})$. The trace of α, denoted by, $\overrightarrow{\alpha}$, is the sequence of pairs $< r_i, t_i >$ where r_i is the $i-$th run at time t_i. The set of all traces of $\mathfrak{G}$, $Tr(\mathfrak{G})$, is the set $\{ \overrightarrow{\alpha} \mid \alpha \in \mathcal{R}(\mathfrak{G}) \}$. An element of $Tr(\mathfrak{G})$ is a trace of $\mathfrak{G}$. A trace captures our notion of retrospective provenance.

3 Provenance-Based Data Transformation

3.1 Provenance Model

Community efforts towards the convergence into a common provenance model led to the Open Provenance Model (OPM)[2] OPM descriptions allow interoperability on the level of workflow structure. This model allows systems with different provenance representations to share at least a workflow-level semantics. OPM, however, is not intended to be a complete provenance model, demanding the complementary use of additional provenance models in order to enable uses of provenance which requires higher level of semantic interoperability. This work targets interoperable provenance representations of IDT and ETL workflows using a three-layered approach to represent provenance. In this representation, the bottom layer represents the basic workflow semantics and structure provided by OPM, the second layer extends the workflow structure provided by OPM with Cogs [6], a provenance vocabulary that provides a rich type structure for describing ETL transformations in the provenance workflow, and voidp

[2] http://openprovenance.org/.

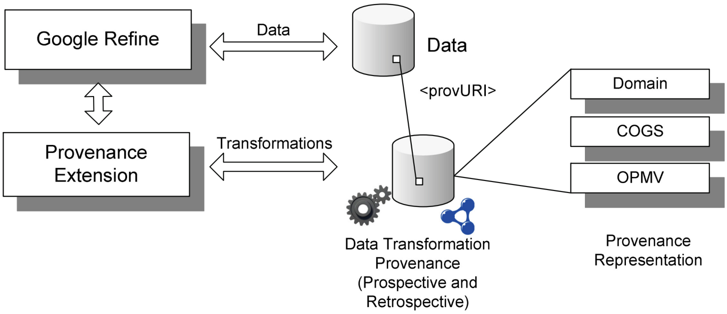

Google Refine
Data
Data
<provURI>
Provenance Extension
Transformations
Domain
COGS
OPMV
Data Transformation Provenance (Prospective and Retrospective)
Provenance Representation

Fig. 2. The system architecture as applied to Google Refine

[8], a provenance extension for the void[3] vocabulary, that allows data publishers to specify the provenance relationships of the elements of their datasets. The third layer consists of a domain specific schema-level information of the source and target datasets or classes/instances pointing to specific elements in the ETL process. In our implementation, the third layer contains the mapping of instances to source code elements. Fig. 2 shows how we applied the model (and architecture) to Google Refine, a popular IDT platform.

3.2 Provenance Capture

There are two major approaches for representing provenance information, and these representations have implications on their cost of recording. These two approaches are: (a) The (Manual) Annotation method: Metadata of the derivation history of a data are collected as annotation. Here, provenance is precomputed and readily usable as metadata, and (b) The Inversion method: This uses the relationships between the input data, the process used to transform and to derive the output data, giving the records of this trace.

The Annotation method is coarser-grained and more suitable for slowly-changing transformation procedures. For more highly-dynamic and time-sensitive transformations, such as IDT procedures, the Inversion method is more suitable. We map the provenance data to the three-layered data transformation model provenance model as described in Sect. 3.1.

After choosing the representation mechanism, the next questions to ask are: (a) what data transformation points would generate the provenance data salient to our provenance needs, and (b) what is the minimal unit of a dataset to attach provenance to. For our system, we choose an Interactive Data Transformation Event to capture these two questions and this is made up of the data object being transformed, the transformation operation being applied to the data, the data output as a result of the transformation, and the time of the operation (as stated in Sect. 2.3).

[3] http://vocab.deri.ie/void/guide.

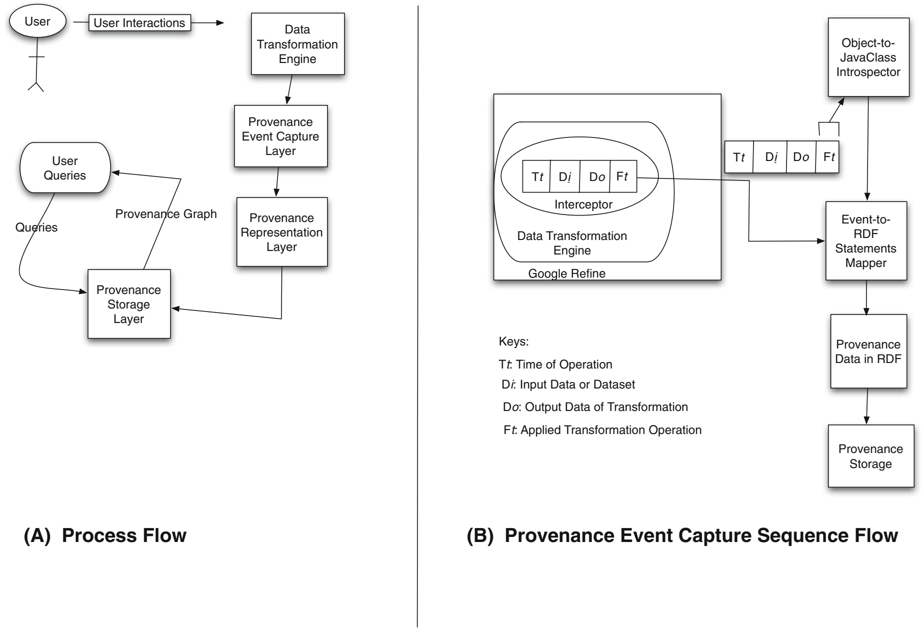

User
User Interactions
Data Transformation Engine
Provenance Event Capture Layer
User Queries
Queries
Provenance Graph
Provenance Representation Layer
Provenance Storage Layer
(A) Process Flow
Object-to-JavaClass Introspector
Tt
Di
Do
Ft
Interceptor
Data Transformation Engine
Google Refine
Event-to-RDF Statements Mapper
Provenance Data in RDF
Provenance Storage
Keys:
Tt: Time of Operation
Di: Input Data or Dataset
Do: Output Data of Transformation
Ft: Applied Transformation Operation
(B) Provenance Event Capture Sequence Flow

Fig. 3. Process and provenance event capture sequence flows

3.3 Process Flow

Fig. 3(A) depicts the process flow that we adopted, and it consists of the following:

1. The Provenance Event Capture Layer, which consists of the following layers and operations' sequence (Fig. 3(B)): (i) The Interceptor Layer: Here, user interactions are intercepted and provenance events extracted from these interactions; (ii) Object-to-JavaClass Introspector: The inputs to this layer are the Transformation Operation chosen to transform the data. We employ the Java language reflection mechanism to elicit the full class path of the operation performed. Eliciting the full class path is useful for the following reasons: (a) It allows us to have a pointer to the binary of the programs doing the transformation, and (b) This pointer to the program binary allows the connection between the full semantics of the program and the data layer. The outputs of this layer are the full class paths of the transformation operations; (iii) Event-to-RDF Statements Mapper: it receives the provenance event and is responsible for the generation of the RDF predicates.
2. These events are then sent to the **Provenance Representation Layer**, which encodes the captured events into RDF using the provenance representation described in Sect. 3.1.
3. These events, represented in RDF, are then sent to the **Provenance Storage Layer**, which stores them in its Knowledge Base (KB).

4 Google Refine: An Exemplar Data Transformation/Curation System

Google Refine[4] (GRefine) is an exemplar Interactive Data Transformation system, and some of its mechanisms include: (i) ability to import data into GRefine from different formats including tsv, csv, xls, xml, json, and google spreadsheets; (ii) GRefine supports faceted browsing[5], such as: Text Facets, Numeric Facets, and Text Filters; (iii) Editing Cells, Columns, and Rows, using GRefine's editing functions; and (iv) The provision of an extension framework API.

4.1 Making Google Refine Provenance-Aware

Some of the design decisions that must be made when making an application provenance-aware is to ask "what" type of provenance data to capture, "when" to collect the said provenance data, and "what" type of method to use for the capture.

As regards to "when" to collect provenance data: (a) provenance data can be collected in real-time, i.e. while the workflow application is running and the input dataset(s) are being processed and used, or (b) ex-post (after-the-fact), i.e. provenance data is gathered after a series of processing events or a sequence of activities has completed.

As regards to "what" type of method to use for collection, provenance data collection methods fall into three types: (a) Through "User annotation": A human data entry activity where users enter textual annotations, capturing, and describing the data transformation process. User-centered metadata is often incomplete and inconsistent [10]. This approach imposes a low burden on the application, but a high burden on the humans responsible for annotation; (b) An automated provenance instrumentation tool can be provided that is inserted into the workflow application to collect provenance data. This places a low burden on the user, but a higher burden on the application in terms of process cycles and/or memory, and (c) Hybrid method: This method uses an existing mechanism, such as a logging or an auditing tool, within the workflow application to collect provenance data.

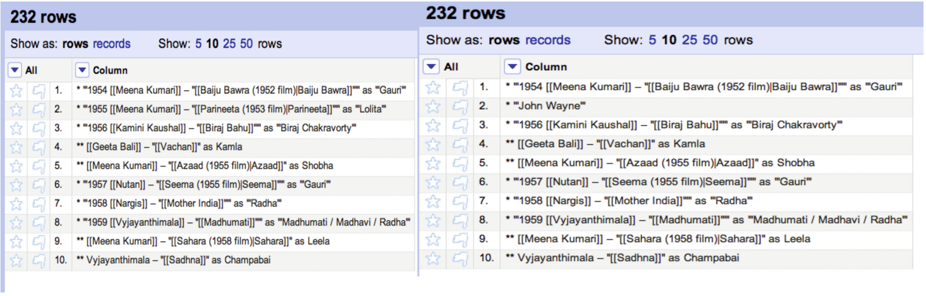
We built an automated instrumentation tool to capture provenance data from user operations as they use the system (Fig. 2). This approach incurs very little burden on the user.

4.2 Usage Scenarios and Data Transformation Experimentation Using Google Refine

Here, we describe how we have used GRefine to capture data transformation provenance events, how these events have been represented using our provenance models (described in Sect. 3.1), and how we have made use of the IDT algebra

[4] http://code.google.com/p/google-refine/.

[5] http://code.google.com/p/google-refine/wiki/FacetedBrowsingArchitecture.



232 rows

Show as: rows records Show: 5 10 25 50 rows

<table>
<tr><th>All</th><th>Column</th></tr>
<tr><td>1.</td><td>* '''1954 [[Meena Kumari]] – ''[[Baiju Bawra (1952 film)|Baiju Bawra]]''''' as '''Gauri'''</td></tr>
<tr><td>2.</td><td>* '''1955 [[Meena Kumari]] – ''[[Parineeta (1953 film)|Parineeta]]''''' as '''Lolita'''</td></tr>
<tr><td>3.</td><td>* '''1956 [[Kamini Kaushal]] – ''[[Biraj Bahu]]''''' as '''Biraj Chakravorty'''</td></tr>
<tr><td>4.</td><td>** [[Geeta Bali]] – ''[[Vachan]]'' as Kamla</td></tr>
<tr><td>5.</td><td>** [[Meena Kumari]] – ''[[Azaad (1955 film)|Azaad]]'' as Shobha</td></tr>
<tr><td>6.</td><td>* '''1957 [[Nutan]] – ''[[Seema (1955 film)|Seema]]''''' as '''Gauri'''</td></tr>
<tr><td>7.</td><td>* '''1958 [[Nargis]] – ''[[Mother India]]''''' as '''Radha'''</td></tr>
<tr><td>8.</td><td>* '''1959 [[Vyjayanthimala]] – ''[[Madhumati]]''''' as '''Madhumati / Madhavi / Radha'''</td></tr>
<tr><td>9.</td><td>** [[Meena Kumari]] – ''[[Sahara (1958 film)|Sahara]]'' as Leela</td></tr>
<tr><td>10.</td><td>** Vyjayanthimala – ''[[Sadhna]]'' as Champabai</td></tr>
</table>

232 rows

Show as: rows records Show: 5 10 25 50 rows

<table>
<tr><th>All</th><th>Column</th></tr>
<tr><td>1.</td><td>* '''1954 [[Meena Kumari]] – ''[[Baiju Bawra (1952 film)|Baiju Bawra]]''''' as '''Gauri'''</td></tr>
<tr><td>2.</td><td>* '''John Wayne'''</td></tr>
<tr><td>3.</td><td>* '''1956 [[Kamini Kaushal]] – ''[[Biraj Bahu]]''''' as '''Biraj Chakravorty'''</td></tr>
<tr><td>4.</td><td>** [[Geeta Bali]] – ''[[Vachan]]'' as Kamla</td></tr>
<tr><td>5.</td><td>** [[Meena Kumari]] – ''[[Azaad (1955 film)|Azaad]]'' as Shobha</td></tr>
<tr><td>6.</td><td>* '''1957 [[Nutan]] – ''[[Seema (1955 film)|Seema]]''''' as '''Gauri'''</td></tr>
<tr><td>7.</td><td>* '''1958 [[Nargis]] – ''[[Mother India]]''''' as '''Radha'''</td></tr>
<tr><td>8.</td><td>* '''1959 [[Vyjayanthimala]] – ''[[Madhumati]]''''' as '''Madhumati / Madhavi / Radha'''</td></tr>
<tr><td>9.</td><td>** [[Meena Kumari]] – ''[[Sahara (1958 film)|Sahara]]'' as Leela</td></tr>
<tr><td>10.</td><td>** Vyjayanthimala – ''[[Sadhna]]'' as Champabai</td></tr>
</table>

Fig. 4. Google Refine edit scenario (Before and After)

(in Sect. 2.3). We converted the contents of FilmAwardsForBestActress[6] into a GRefine project called "actresses".

We have applied our definition of a **Run** (as stated in Sect. 2.3) to actual implementations in the Usage Scenarios described below. Also here, we see the Google Refine system as an implementation of an IDT, $\mathfrak{G}$ (from Definition 1 of Sect. 2.3).

Edit Usage Scenario. If we want to change the entry in row 2 (of Fig. 4) from *”’1955 [[Meena Kumari]] ”[[Parineeta (1953 film)—Parineeta]]”””’ as ”’Lolita” to "1935 John Wayne" and would like our system to keep a provenance record of this transaction, we can achieve that, in GRefine, by viewing the column as a "Text Facet" and applying the GRefine's "Edit" operation on row 2. Figure 4 shows us the before and after pictures of our Edit operation.

The **Events-to-RDF Statements Mapper** automatically generates the RDF statements using the following mechanisms. A transformation function, e.g. "Edit", from GRefine is automatically mapped to a type (an "`rdf:type`") of `opmv:Process`. What the operation gets mapped to in the Cogs ontology depends on the attribute of the data item that was the domain of the operation. For example, if the data item attribute is a Column, this gets mapped to a Cogs "ColumnOperation" class, while if the item attribute is a Row, this gets mapped to a Cogs "RowOperation" class. Since this is a transformation process, we made use of "`cogs:TransformationProcess`" class. voidp has a single class of ProvenanceEvent and every transformation operation is mapped to "`voidp:ProvenanceEvent`".

The system's Object-to-JavaClass Introspector is used to elicit the actual Java class responsible for the transformation. We have made use of a Cogs property, "`cogs:programUsed`", to specify the full path to this Java class. It allows the generated provenance data to communicate with the program semantics, in this way the intensional program semantics is linked up with the provenance extensional semantics. The data item that is actually made use of in the transformation

[6] http://en.wikipedia.org/w/index.php?title=Filmfare_Award_for_Best_Actress&action=edit&section=3.

process is mapped to “`opmv:Artifact`”. To give us the process flow that is part of the transformation, we used two opmv properties: (a) “`opmv:wasDerivedFrom`”, which tells us from which artifact this present data item is derived from, and (b) “`opmv:wasGeneratedBy`”, which tells us from which process this data item is generated from. In order to store temporal information of when the derivation took place, we used “`opmv:wasGeneratedAt`”, an opmv property that tells us the time of transformation.

The result of the transformation operation as RDF statements is below:

```
@prefix id: <http://127.0.0.1:3333/project/1402144365904/> .
id:MassCellChange-1092380975 rdf:type opmv:Process,
cogs:ColumnOperation, cogs:TransformationProcess, voidp:ProvenanceEvent ;
opmv:used <http://127.0.0.1:3333/project/1402144365904/
                                      MassCellChange-1092380975/1_0> ;
cogs:operationName"MassCellChange"^^xsd:string;
cogs:programUsed "com.google.refine.operations.cell.
                                MassEditOperation"^^xsd:string;
rdfs:label  "Mass edit 1 cells in column ==List of winners=="^^xsd:string.

<http://127.0.0.1:3333/project/1402144365904/MassCellChange-1092380975/1_0>
rdf:type opmv:Artifact ;
rdfs:label  "*’’’1955 [[Meena Kumari]]
     ’[[Parineeta (1953 film)|Parineeta]]’’’’’ as’’’Lolita’’’"^^xsd:string.

http://127.0.0.1:3333/project/1402144365904/MassCellChange-1092380975/1_1>
rdf:type    opmv:Artifact ;
rdfs:label  "*’’’John Wayne’’’"^^xsd:string;
opmv:wasDerivedFrom <http://127.0.0.1:3333/project/1402144365904/
                  MassCellChange-1092380975/1_0>;
opmv:wasGeneratedBy <http://127.0.0.1:3333/project/1402144365904/
                 MassCellChange-1092380975>;
opmv:wasGeneratedAt "2011-11-16T11:2:14"^xsd:dateTime.
```

Row Removal Usage Scenario. Let us say we want to remove the rows that have lines starting with two equal signs such as this: “==”. One way to achieve this in GRefine is to do the following: “Column → Text Filter → == → All → Edit rows → Remove all matching rows”.

Our operation removed seven rows, and the places where they were removed are also shown by the RDF statements, in `rdfs:label`, below:

```
id:RowRemovalChange-1030462081
rdf:type    opmv:Process, cogs:RowOperation,
                  cogs:TransformationProcess, voidp:ProvenanceEvent ;
opmv:used   <http://127.0.0.1:3333/project/1402144365904/
                                 RowRemovalChange-1030462081/1_0> ;
cogs:operationName  "RowRemovalChange"^^xsd:string;
cogs:programUsed        "com.google.refine.operations.row.
                                 RowRemovalOperation"^^xsd:string;
rdfs:label  "Remove 7 rows"^^xsd:string.

<http://127.0.0.1:3333/project/1402144365904/
                                       RowRemovalChange-1030462081/1_0>
rdf:type    opmv:Artifact ;
```

```
rdfs:label  ""^^xsd:string.

http://127.0.0.1:3333/project/1402144365904/
                         RowRemovalChange-1030462081/1_1>
rdf:type    opmv:Artifact ;
rdfs:label  "[0, 11, 42, 84, 126, 174, 227]"^^xsd:string;
opmv:wasDerivedFrom <http://127.0.0.1:3333/project/1402144365904/
                                                    RowRemovalChange-1030462081/1_0>;
opmv:wasGeneratedBy <http://127.0.0.1:3333/project/1402144365904/
                                                    RowRemovalChange-1030462081>;
opmv:wasGeneratedAt "2011-11-16:11:4:58"^^xsd:dateTime.
```

In the next section, we will describe the requirements we used for analysis.

5 Analysis and Discussion

Our approach and system were evaluated in relation to the set of requirements listed in [3,6] and enumerated below:

1. **Decentralization:** deployable one database at a time, without requiring co-operation among all databases at once. Coverage: High. **Justification:** Use of Semantic Web Standards and vocabularies (OPMV + Cogs + voidp) to reach a decentralized/interoperable provenance solution.
2. **Data model independency:** should work for data stored in flat file, relational, XML, file system, Web site, etc., model. Coverage: High. **Justification:** Minimization of the interaction between the data level and the provenance level. Data is connected with its provenance descriptor by a provenance URI and the provenance representation is normalized as RDF/S.
3. **Minimum impact to existing IDT practice:** Provenance tracking is invisible to the user. Coverage: High. **Justification:** The provenance capture is transparent to the user. Provenance is captured by a lightweight instrumentation of the IDT platform, mapping program structures to the provenance model.
4. **Scalability:** to situations in which many databases cooperate to maintain provenance chain. Coverage: High. **Justification:** Usage of Semantic Web standards and vocabularies for provenance representation allows for the co-operation of multiple platforms for provenance management.

6 Related Work

We categorize related work into three: (i) provenance management for manually curated data [3], (ii) interactive data transformation models and tools [4,5], and (iii) data transformation models for databases [7].

Buneman et al. [3] propose a model for recording provenance of data in a single manually curated database. In their approach, the data is copied from external data sources or modified within the target database creating a copy-paste model for describing user actions in assimilating external datasources into curated database records.

Raman and Hellerstein [5] describe Potters Wheel, an interactive data cleaning system which allowed users to specify transformations through graphic elements. Similarly, Wrangler [4] is an interactive tool based on the visual specification of data transformations. Both systems are similar to Google Refine: [5], however, provides a more principled analysis of the interactive data transformation process, while [4] focuses on new techniques for specifying data transformations. The IDT and provenance model proposed in our work can be directly applied to both systems.

Davidson [7] analyses the requirements for the construction of a formalism for modeling the semantics of database transformations and propose a declarative language for specifying and implementing these transformations and other constraints. They conclude that approaches for modeling database transformations based on a fixed-set of well-defined transformations can inherently limit the expressivity of the transformation model. An additional conclusion is that a high-level language is necessary to express general transformations but that such language should have a well-defined formal semantics.

Our work is different from that of Davidson [7]. Our solution provides a best-effort and pay-as-you-go semantics for the data transformation, focusing on an improvement of the level of semantic interoperability across data provenance of data transformation representations. We also provide a formal model of the transformation system, on which a language can be constructed to generate such a data transformation system.

7 Conclusion

The world contains an unimaginably vast amount of data which is getting ever vaster ever more rapidly. These opportunities demand data transformation efforts for data analysis and re-purposing. Interactive data transformation (IDT) tools are becoming easily available to lower barriers to data transformation challenges. Some of these challenges will be solved by developing mechanisms useful for capturing the process flow and data lineage of these transformation processes in an interoperable manner. In this paper, we provide a formal model of IDT, a description of the design and architecture of an ontology-based provenance system used for mapping data transformation operations, and its implementation and validation on a popular IDT platform, Google Refine. We shall be making our provenance extension to Google Refine publicly accessible very soon.

Acknowledgements. This work was supported by the EnAKTing project, funded by EPSRC project number EP/G008493/1 and by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Buneman, P.: Curated databases. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD) (2006)

2. Curry, E., Freitas, A., O'Riain, S.: The role of community-driven data curation for enterprises. In: Wood, D. (ed.) Linking Enterprise Data, pp. 25–47. Springer, Boston (2010)
3. Buneman, P., Chapman, A., Cheney, J., Vansummeren, S.: A provenance model for manually curated data. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 162–170. Springer, Heidelberg (2006)
4. Kandel, S., Paepcke, A., Hellerstein, J., Heer, J.: Wrangler: interactive visual specification of data transformation scripts. In: ACM Human Factors in Computing Systems (CHI) (2011)
5. Raman, V., Hellerstein, J.: Potter's wheel: an interactive data cleaning system. In: Proceedings of the 27th International Conference on Very Large Data Bases (2001)
6. Freitas, A., Kämpgen, B., Oliveira, J.G., O'Riain, S., Curry, E.: Representing interoperable provenance descriptions for ETL workflows. In: Proceedings of the 3rd International Workshop on Role of Semantic Web in Provenance Management (SWPM 2012), Extended Semantic Web Conference (ESWC), Heraklion, Crete (2012)
7. Davidson, S., Kosky, A., Buneman, P.: Semantics of database transformations. In: Thalheim, B., Libkin, L. (eds.) Semantics in Databases 1995. LNCS, vol. 1358, pp. 55–91. Springer, Heidelberg (1998)
8. Omitola, T., Zuo, L., Gutteridge, C., Millard, I., Glaser, H., Gibbins, N., Shadbolt, N.: Tracing the provenance of linked data using voiD. In: The International Conference on Web Intelligence, Mining and Semantics (WIMS 2011) (2011)
9. Deelman, E., Gannon, D., Shields, M., Taylor, I.: Workflows and e-Science: an overview of workflow system features and capabilities. Future Gener. Comput. Syst. **25**(5), 528–540 (2009)
10. Newhouse, S., Schopf, J.M., Richards, A., Atkinson, M.: Study of user priorities for e-Infrastructure for e-Research (SUPER). In: UK e-Science Technical report Series Report UKeS-2007-01 (2007)

Representing Interoperable Provenance Descriptions for ETL Workflows

André Freitas[1](✉), Benedikt Kämpgen[2], João Gabriel Oliveira[1],
Seán O'Riain[1], and Edward Curry[1]

[1] Digital Enterprise Research Institute (DERI),
National University of Ireland, Galway, Ireland
andre.freitas@deri.org
[2] Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract. The increasing availability of data on the Web provided by the emergence of Web 2.0 applications and, more recently by Linked Data, brought additional complexity to data management tasks, where the number of available data sources and their associated heterogeneity drastically increases. In this scenario, where data is reused and repurposed on a new scale, the pattern expressed as Extract-Transform-Load (ETL) emerges as a fundamental and recurrent process for both producers and consumers of data on the Web. In addition to ETL, *provenance*, the representation of source artifacts, processes and agents behind data, becomes another cornerstone element for Web data management, playing a fundamental role in data quality assessment, data semantics and facilitating the reproducibility of data transformation processes. This paper proposes the convergence of these two Web data management concerns, introducing a principled provenance model for ETL processes in the form of a vocabulary based on the Open Provenance Model (OPM) standard and focusing on the provision of an interoperable provenance model for ETL environments. The proposed ETL provenance model is instantiated in a real-world sustainability reporting scenario.

Keywords: ETL · Data transformation · Provenance · Linked Data · Web

1 Introduction

Extract-Transform-Load (ETL) is a fundamental process in data management environments. In Data Warehousing, data preprocessing is crucial for reliable analysis, e.g., reporting and OLAP; data coming from large databases or data derived using complex machine-learning algorithms may hide errors created in an earlier step of the analysis process. As a result, the design of ETL processes such as retrieving the data from distributed sources, cleaning it from outliers, and loading it in a consistent data warehouse demands up to 80 % of data analysts' time [12].

The growing availability of data on the Web provided by Web 2.0 applications and, more recently through Linked Data, brought the computational

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 43–57, 2015.
DOI: 10.1007/978-3-662-46641-4_4

pattern expressed as ETL to reemerge in a scenario with additional complexity, where the number of data sources and the data heterogeneity that needs to be supported by ETL drastically increases. In this scenario, issues with data quality and trustworthiness may strongly impact the data utility for end-users. The barriers involved in building an ETL infrastructure under the complexity and scale of the available Web-based data supply scenario demands the definition of strategies which can provide data quality warranties and also minimise the effort associated with data management.

In this context, *provenance*, the representation of *artifacts*, *processes* and *agents* behind a resource, becomes a fundamental element of the data infrastructure. Given the possibility to represent ETL workflows both at design time (*prospective provenance*), and after execution (*retrospective provenance*), provenance descriptions can overcome challenges of today's ETL scenarios in a large spectrum of applications including documentation for reproducibility and reuse, data quality assessment to improve trustworthiness as well as automatic consistency checking, debugging and semantic reconciliation [14]. Additionally, the frequency and generality of simple and recurrent processes such as contained in many data transformation workflows in an environment with increasing data availability justifies the importance of a provenance descriptions for ETL.

However, in an environment where data is produced and consumed by different systems, the representation of provenance should be made interoperable across systems. Interoperability represents the process of sharing the semantics of the provenance representation among different contexts. Although some systems in the areas of data transformation [1] and databases [20] provide a historical trail of data, those descriptions cannot be easily shared or integrated. *Provenance* and *interoperability* walk together: provenance becomes fundamental when the borders of a specific system or dataset are crossed, where a representation of a workflow abstraction of the computational processes can enable reproducibility, improve data semantics and restore data trustworthiness. Ultimately, provenance can make the computational processes behind applications interpretable at a certain level by external systems and users.

Standardisation efforts towards the convergence into a common provenance model generated the Open Provenance Model [11] (OPM). OPM provides a basic provenance description which allows interoperability on the level of workflow structure. The definition of this common provenance ground allows systems with different provenance representations to share at least a workflow-level semantics, i.e., the causal dependencies between artifacts, processes and the intervention of agents. OPM, however, is not intended to be a complete provenance model, but demands the complementary use of additional provenance models in order to enable applications of provenance that require higher level of semantic interoperability. The explicit trade-off between the semantic completeness of a provenance model and its level of interoperability imposes challenges in specifying a provenance model.

This paper focuses on the provision of a solution that allows the improvement of the semantic completeness and interoperability for provenance descriptors in complex data transformation/ETL scenarios. To achieve this goal, a vocabulary focused on modelling ETL workflows is proposed. This model is built upon the workflow structure of OPM, designed to extend the basic semantics and structure of OPM-based provenance workflows. In this work, the ETL acronym is used in a broader context, focusing on generic data transformation patterns, transcending the original Data Warehouse associated sense. The contributions of this work are summarised in the following: **(i)** analysis of requirements for an interoperable provenance model for ETL workflows, **(ii)** provision of a solution in the form of a Linked Data ETL vocabulary, **(iii)** application of the proposed model in a real-world ETL scenario.

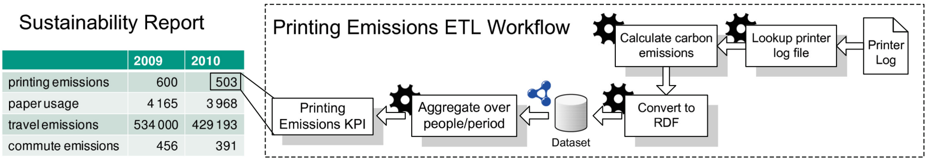
The paper is organised as follows: Sect. 2 presents an ETL motivational scenario, Sect. 3 analyses related work on the representation and formalisation of ETL provenance workflows; Sect. 4 provides a list of requirements for an ETL provenance model; Sect. 5 describes the construction of the ETL provenance model, describing *Cogs*, a provenance vocabulary for ETL. Section 6 describes the application of the ETL vocabulary in a case study for sustainable reporting. Section 7 finally provides conclusions and future work.

2 ETL Motivational Scenario

The ability to describe data transformation processes behind data resources plays a fundamental role while producing and consuming data, especially if done on heterogeneous data sources and by different parties. Applications need to become provenance-aware, i.e., attaching to the data the associated description of what has been done to generate the data. This brings provenance management as a key requirement for a wide spectrum of applications which publish and consume data on the Web and, in particular, to ETL activities.

As a concrete motivational scenario consider an organisation publishing a sustainability report on the Web (Fig. 1). The sustainability report contains Key Performance Indicators (KPIs) related to the environmental impact of the organisation which can be audited by external regulators, reused by customers to calculate their indirect environmental impact or used internally to minimise the company environmental impact. One example KPI is the total volume of CO2 emissions/per time period which is calculated by collecting indicators of energy consumption emissions, travel emissions, printing emissions, etc. The data used to build the indicators is collected from distributed and heterogeneous sources which include spreadsheets, log files and RDF data, and is processed through distinct ETL workflows into data cubes. An application queries the final sustainability KPIs from the data cubes, publishing them as a report on the Web.

The problem that is specifically introduced in this scenario is the fact that different values might have been produced by independently developed and executed ETL workflows. For instance, the value indicating a printing emissions of 503 kg of carbon dioxide as indicated in Fig. 1 is created by a lookup on the printer log file, a conversion to RDF, an aggregation over people and a filter on the year 2010. The printing emission for 2009, however, might have been produced by a crawl of RDFa from the organisation's website and a unit conversion by a constant factor. Each KPI should have an associated provenance trail describing the data processing steps from the original data sources, so that information consumers – both humans and machines – are able to make better sense of information generated by heterogeneous ETL processes.


Sustainability Report

	2009	2010
printing emissions	600	503
paper usage	4 165	3 968
travel emissions	534 000	429 193
commute emissions	456	391

Printing Emissions ETL Workflow
Calculate carbon emissions
Lookup printer log file
Printer Log
Printing Emissions KPI
Aggregate over people/period
Dataset
Convert to RDF

Fig. 1. Representing provenance behind the KPI of a sustainability report.

3 Related Work: The Gap of ETL Workflow Descriptions

Previous literature analysed and formalised conceptual models for ETL activities. In the center of these models is the concept of *data transformations*. This section describes previous data transformation models, analysing their suitability as interoperable provenance descriptions for our motivational scenario. Existing work can be grouped into two major perspectives: *ETL Conceptual Models*, which focus on the investigation of ontologies and serialisation formats for design, development, and management of ETL workflows, and *ETL Formal Models*, which concentrate on applications of ETL descriptions that require formal, logics- or algebra-based representations. Between these two groups we identify the gap of an interoperable ETL provenance model.

3.1 ETL Conceptual Models

Standardisation efforts by the W3C Provenance Incubator Group[1] and the later Provenance Working Group[2] have considered in-scope use cases of data integration and repeatable data analyses. Yet, their focus targets the determination of basic provenance descriptors allowing interoperability on an abstract level of workflow semantics and does not target more specific provenance descriptors from an ETL perspective.

Much work has been done in the usage of ontologies for automated design and standard descriptions of ETL tasks. Vassiliadis et al. [19] investigate generic properties present in ETL activities across different ETL implementations and, based on these properties, construct a taxonomy of ETL concepts. Vassiliadis et al. and Skoutas & Simitsis [15,19] use a categorisation of operations from

[1] http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/.

[2] http://www.w3.org/2011/prov/wiki/Main_Page.

different systems to capture the relationship between input and output definitions. Trujillo and Luján-Mora [17] propose to use UML for the specification of ETL processes in terms of operations such as the transformation between source and target attributes and the generation of surrogate keys. Similarly, Akkaoui & Zimani [5] propose a conceptual language for modelling ETL workflows based on the Business Process Model Notation (BPMN). The artificiality of the solution lies on the fact that BPMN is not intended to be a universal data representation format, bringing questions on its suitability as an interoperable representation. In general, although the mentioned conceptual models introduce common terms and structures for ETL operations and help with ETL-related communication and discussions, they do not aim at providing a machine-readable representation of heterogeneous ETL processes to be shared by data consumers.

Becker & Ghedini [2] describe a system to document data mining projects, including the data preprocessing step. Descriptions are manually captured by the analysts in a Web 2.0 fashion. The ETL representations include tasks as an abstraction for various preprocessing activities, distinguishing between prospective task definitions and retrospective task executions. Tasks can be annotated using free text and can be tagged with predefined concepts. Although useful for reproducibility and reuse in terms of knowledge management between ETL designers, those descriptions lack a minimum of ontological commitment for interoperability between heterogeneous ETL applications.

Other works specifically aims at sharing descriptions between systems. The Common Warehouse Metamodel (CWM) is an open OMG standard for data warehousing which defines a metadata model and an XML-based exchange standard. In CWM, a *transformation* is a basic unit in the ETL process which can be combined into a set of tasks. Thi & Nguyen [16] propose a CWM compliant approach over an ontology-based foundation for modelling ETL processes for data from distributed and heterogeneous sources; their approach does not model the types of transformations explicitly but only provides a basic mapping infrastructure which can be used to reference external classes. Also, the approach lacks a concrete use case where its benefits are demonstrated.

Kietz et al. [9] introduce a cooperative planning approach for data mining workflows using the support of a data mining ontology (DMO). DMO covers artifacts (I/O-Objects), processes (Operators), as well as descriptors to describe artifacts in more detail (MetaData); as such, it provides ETL descriptions and allows semantic interoperability between systems. However, DMO was not designed to track the history of data and lacks retrospective provenance of ETL workflows.

3.2 ETL Formal Models

Formal models use logics or algebras to describe ETL descriptions. Davidson et al. [4] analyse the requirements for the construction of a formalism for modelling the semantics of database transformations and propose a declarative language for specifying and implementing database transformations and constraints. The motivation of their work is to generate a transformation formalism which can be used to verify the correctness of transformations. Galhardas et al. [8]

propose another high-level declarative language for data transformations and describe the reasoning behind transformations. However, abstract languages for data cleaning and transformations are overly formal to be widely adopted for interoperable ETL descriptions.

Cui & Widom [3] formalise the lineage problem on general database environments proposing algorithms for lineage tracing. They restrict their model to specific transformation classes. The approach is not suited to describe general data transformation activities varying from simple filtering operations to complex procedural routines such as present in our motivational scenario.

Vassiliadis et al. [18] provide an abstract categorisation of frequently used ETL operations in order to introduce a benchmark of relational ETL systems. The benchmark documents measures such as data freshness and consistency, resilience to failures, and speed of workflows. In order to describe concrete ETL workflows such as given by our motivational scenario, both operations and measures are too abstract to help with problems of interpretability.

In summary, formal models of ETL workflows often explicitly limit their range of considered ETL workflows to fulfil specific tasks. Those models do not intend to provide interoperability across different ETL applications, but to achieve certain functionalities in their system, e.g., automatic debugging.

We have identified a gap regarding an interoperable ETL provenance model. Previous literature has either presented models with very high-level semantics lacking the ability to describe prospective and retrospective ETL provenance or presented rigorously formalised models that require to much ontological commitment for a broad adoption. As for ETL applications such as Kapow Software, Pentaho Data Integration, and Yahoo Pipes: currently, they either do not create and use provenance information or do not support sharing and integrating such provenance data with other applications.

4 Requirements of an Interoperable ETL Provenance Model

This section defines a list of requirements which summarises the core usability and model characteristics that should be present in an ETL provenance model. The requirements are defined to satisfy the two core demands which were found as gaps on the ETL literature (i) lack of a provenance representation from an ETL perspective and (ii) semantic interoperability across different ETL platforms and applications. An additional third demand is introduced: (iii) usability demand, i.e., the minimal effort and ontological commitment needed for an instantiation of a correct and consistent model. The requirements are described below:

1. *Prospective and retrospective descriptions:* Provenance descriptors represent both workflows specifications at design time (*prospective provenance*) and workflows which were already executed (*retrospective provenance*). Impacts: i, ii and iii.

2. *Separation of concerns:* ETL-specific elements are separated from the provenance workflow structure, allowing at least a minimum level of interoperability between ETL and non-ETL provenance descriptors. This requirement is aligned with the OPM [11] compatibility. Impacts: ii.
3. *Terminological completeness:* Terminological completeness of the provenance descriptor is maximised; there is a large terminological coverage of ETL elements. Impacts: i and ii.
4. *Common terminology:* Descriptors allow a common denominator of representations of ETL elements. Elements present in different ETL platforms can be mapped. Impacts: i and ii.
5. *Lightweight ontology structure:* A lightweight provenance model is provided; complex structures bring barriers for the instantiation and consumption of models, including consistency problems, scalability issues, interpretability problems and additional effort in the model instantiation. Impacts: iii.
6. *Availability of different abstraction levels:* The vocabulary allows users to express multiple abstraction levels for both processes and artifacts, varying from fine grained to coarse grained descriptions. Users are able to express multiple levels of abstraction simultaneously. This requirement is also present in the OPM specification [11]. Impacts: ii and iii.
7. *Decentralisation:* ETL provenance descriptors may be deployed on distributed database platforms without requiring cooperation among all databases. Impacts: ii and iii.
8. *Data representation independency:* Descriptors are able to refer to any data representation format including relational, XML, text files, etc. Impacts: iii.
9. *Accessibility:* The generated provenance descriptors are easily accessible for data consumers. Both machines and humans are able to query and further process provenance descriptors. Impacts: ii and iii.

5 Provenance Model for ETL Workflows

The following high-level approach was used to provide an ETL provenance model addressing the requirements:

- Construction of the provenance model based on the Open Provenance Model workflow structure, extending OPM with a hierarchical workflow structure, facilitating the representation of nested workflows.
- Design of a complementary vocabulary for expressing the elements present in an ETL workflow. The vocabulary can be extended to describe domain-specific objects.
- Usage of the Linked Data principles for representing, publishing and linking provenance descriptors on the Web in a machine-readable format.

In the following, we describe in more detail the two main features of the ETL provenance model: the multi-layered design and the ETL vocabulary *Cogs*.

5.1 Multi-layered Provenance Model

A three-layered approach is used, as depicted on the left side of Fig. 2, to provide interoperable provenance representations of ETL and generic data transformation workflows. OPM is a technology agnostic specification: it can be implemented using different representations or serialisations. This work uses the OPM Vocabulary[3] (OPMV) as the representation of OPM. In this representation, the bottom layer represents the basic workflow semantics and structure provided by OPMV, the second layer represents the common data extraction, transformation and loading entities and the third layer represents a domain specific layer.

The ETL provenance model layer is built upon the *basic workflow structure* of the OPMV layer. The ETL provenance model layer is designed to include a set of common entities present across different ETL workflows, providing a terminologically-rich provenance model instantiated as the *Cogs* vocabulary. The third layer consists of a domain specific layer which extends the second layer, consisting of domain-specific schema and instance-level information, e.g., of domain-specific source and target datasets or operations. An example of domain specific elements are references to e-Science operations from biological experiments that would further specialise classes of Cogs operators.

This paper defines a conceptual model for the second layer and describes its interaction with the two complementary layers. The separation of the provenance model into the three-layered structure supports the requirement *(2) separation of concerns*.

5.2 Cogs: A Vocabulary for Representing ETL Provenance

In the construction of Cogs, the core relationships are provided by *object properties* on the OPMV layer. The Cogs model specialises the core OPMV entities, artifacts and processes, with a rich taxonomic structure. The approach used in Cogs focuses on the design of a Linked Data vocabulary, a lightweight ontology, which minimises the use of logical features such as transitive, inverse properties as well as the consistency/scalability problems associated with the reasoning process (impacts requirement *(5) lightweight ontology structure*).

The methodology for building the Cogs vocabulary considered the following dimensions: (i) the requirements analysis (ii) the core structural definition of modelling ETL workflows using the structure of OPMV workflows, (iii) an in depth analysis of concepts expressed in a set of analysed ETL/data transformation tools (Pentaho Data Integration,[4] Google Refine[5]) and (iv) concepts and structures identified from the ETL literature [3,10,19]. The core of the Cogs vocabulary captures typical operations, objects and concepts involved in ETL activities, at different phases of the workflow.

[3] http://open-biomed.sourceforge.net/opmv/ns.html.

[4] http://kettle.pentaho.com.

[5] http://code.google.com/p/google-refine.

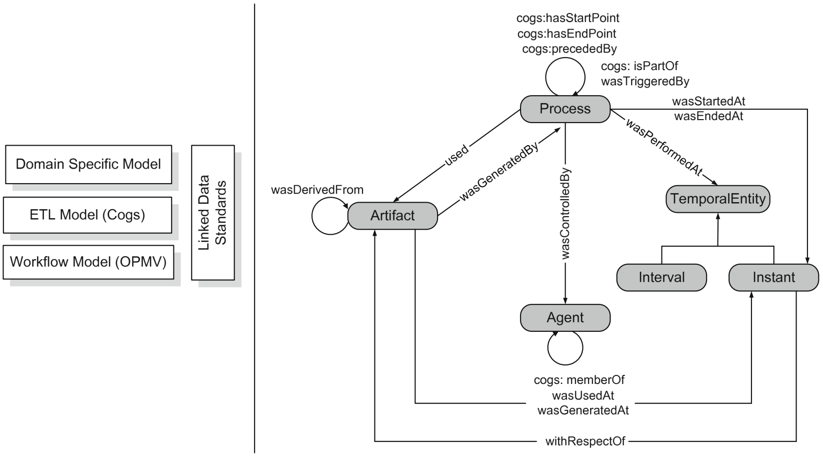


Fig. 2. OPMV workflow structure extended with additional Cogs properties.

Cogs also extends the workflow structure of OPMV with additional object properties targeting the creation and navigation of hierarchical workflow structures. Hierarchical workflow structures allow the representation of both fine grained (important for machine interpretation and automated reproducibility) and coarse grained (important for human interpretation) provenance representation. This feature impacts both requirements *(6) availability of different abstraction levels and (1) prospective and retrospective descriptions*. Similar hierarchical features extending OPM were also targeted in [6]. Figure 2 depicts the core of the OPMV workflow model and the workflow extension of the Cogs vocabulary (marked with the cogs namespace).

The Cogs vocabulary defines a taxonomy of 151 classes. In addition, 15 object properties and 2 data properties are included in the vocabulary. The large number of classes allows a rich description of ETL elements supporting an expressive ETL representation (impacts requirements *(3) terminological completeness and (6) availability of different abstraction levels*). The classes, extracted from the ETL literature and from available tools also cover the *(4) common terminology* requirement. The vocabulary taxonomy is structured with 8 high-level classes which are described below:

- *Execution:* Represents the execution job (instance) of an ETL workflow. Examples of subclasses include *AutomatedAdHocProcess* and *ScheduledJob*.
- *State:* Represents an observation of an indicator or status of one particular execution of an ETL process. These can range from execution states such as *Running* or *Success* to execution statistics, captured by the subclasses of the *PerformanceIndicator* class.
- *Extraction:* Represents operations of the first phase of the ETL process, which involves extracting data from different types of sources. *Parsing* is a subclass example. cogs:Extraction is an opmv:Process.

- *Transformation:* Represents operations in the transformation phase. Typically this is the phase which encompasses most of the semantics of the workflow, which is reflected on its number of subclasses. Examples of classes are *RegexFilter*, *DeleteColumn*, *SplitColumn*, *MergeRow*, *Trim* and *Round*. cogs:Transformation is an opmv:Process.
- *Loading:* Represents the operations of the last phase of the ETL process, when the data is loaded into the end target. Example classes are *ConstructiveMerge* and *IncrementalLoad*. cogs:Loading is an opmv:Process.
- *Object:* Represents the sources and the results of the operations on the ETL workflow. These classes, such as *ObjectReference*, *Cube* or *File*, aim to give a more precise definition of opmv:Artifact (every cogs:Object is an opmv: Artifact) and, together with the types of the operations that are generating and consuming them, capture the semantics of the workflow steps.
- *Layer:* Represents the different layers where the data can reside during the ETL process. *PresentationArea* and *StagingArea* are some of the subclasses.

In practice, it is not always possible to capture all data transformation operations into a fine-grained provenance representation. One important feature of the Cogs vocabulary is the fact that program descriptions (i.e. source code) or executable code can be associated with the transformations using the *cogs:programUsed* property. This feature impacts the requirements *(3) terminological completeness, (6) availability of different abstraction levels and (1) prospective and retrospective descriptions.*

The use of Linked Data principles strongly supports requirement *(10) accessibility* by allowing a unified standards-based publication and access layer to data. In the proposed model, the standards-based provenance representation is separated from the database representation (a relational database record or an element inside an XML file can have its provenance information represented using Linked Data principles). The use of (provenance) URIs to associate provenance information to data items is a generic solution which can be directly implemented to every data representation format, supporting the requirement *(8) data representation independency*. Additionally, by using RDF(S), HTTP and URIs, provenance can be persisted in a decentralised way (requirement *(7) decentralisation*). Users can access provenance through SPARQL queries, faceted-search interfaces, and follow-your-nose Linked Data browsers over dereferenceable URIs.

Table 1 summarises the requirements coverage by the proposed provenance model. The current version of the Cogs vocabulary is available at http://vocab.deri.ie/cogs and complementary documentation is available at: http://sites.google.com/site/cogsvocab/.

6 Vocabulary Instantiation

In order to analyse the suitability of the proposed vocabulary as a representation of ETL processes, we have implemented an instantiation of the Cogs vocabulary using as a case study a platform for collecting sustainability information at

Table 1. Requirements coverage of each element of the provenance model: '+' represents an effective impact on the requirements dimension while '−' represents the lack of impact.

Requirement	OPMV	Cogs	LD principles
Prospective and retrospective descriptions	+	+	−
Separation of concerns	+	+	−
Terminological completeness	+	+	+
Common terminology	+	+	−
Lightweight ontology structure	+	+	−
Availability of different abstraction levels	−	+	−
Decentralisation	−	−	+
Data representation independency	+	+	+
Accessibility	+	−	+

the Digital Enterprise Research Institute (DERI), similar to our motivational scenario. We first describe the application of the ETL provenance model in the use case and then discuss the results.

6.1 Use Case

The organisation-wide nature of sustainability indicators, reflecting the organisational environmental impact, means that potential information is scattered across the organisation within numerous existing systems. Since existing systems were not designed from the start to support sustainability analysis, heterogeneous data present in distributed sources need to be transformed into sustainability indicators following an ETL process. The correctness and consistency of each sustainability KPI needs to be auditable through the publication of the associated provenance information, which should be interpretable by different stakeholders.

The ETL process for the construction of sustainability indicators consists of four separate workflows, for printing emissions, paper usage, travel emissions and commute emissions. Data sources include RDF graphs for people, research units and different file formats containing raw data. The basic ETL workflow consists in a sequence of operations: file selection, filtering, transformation, CO2 emissions calculation and transformation into RDF conforming to the RDF Data Cube vocabulary. On the last step information in the data cubes is aggregated to generate a final report available on the Web. The ETL workflow is implemented in Java code. To make the ETL workflow provenance-aware, the Prov4J-Light framework was used, a lightweight version of [7], which is a Java framework for provenance management, that uses Semantic Web tools and standards to address the core challenges for capturing and consuming provenance information in generic Java-based applications. Core Java objects are mapped to artifacts and processes in the OPMV + Cogs provenance model. The set of generated instances is persisted in a separate provenance dataset. The connection between the final data, which is available in HTML format, and its provenance descriptor is given by a provenance URI (provURI) which is a reflection of the annotated artifact in the provenance store, pointing to its associated retrospective provenance workflow. Each element in the provenance store is represented by a dereferenceable provenance URI. Applications and users can navigate through the workflow structure by following the graph links or by executing SPARQL queries. Figure 3 depicts the high-level components of the provenance capture and storage mechanism.

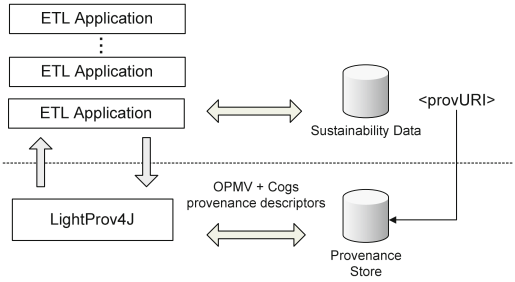


Fig. 3. High-level architecture of the provenance capture and storage mechanism.

6.2 Discussion

The purpose of the workflow usage should be determined in advance, where coarse grained data transformation representations are more suitable for human consumption (in particular, in the determination of human-based quality assessment) while fine grained representations provide a higher level of semantic interoperability which is more suitable for enabling automatic reproducibility. The proposed provenance model for ETL can serve both granularity scenarios. For our case study, since the main goal is to provide a human auditable provenance trail, a coarse grained implementation was chosen. Figure 4 depicts a short excerpt of the workflow in the provenance visualisation interface with both OPMV and Cogs descriptors. The user reaches the provenance visualisation interface by clicking in a value on an online financial report. Readers can navigate through a workflow descriptor for the printing CO2 emissions on the Web.[6] The final average linear workflow size of the high-level workflow consisted of 4 processes and 5 artifacts.

One important aspect for a provenance model is the expressivity of the queries supported by it. The OPMV layer allows queries over the basic workflow structure behind the data, such as *What are the data artifacts, processes and agents behind this data value?*, *When were the processes executed?*, *How long does each*

[6] http://treo.deri.ie/cogs/example/swpm2012.htm.

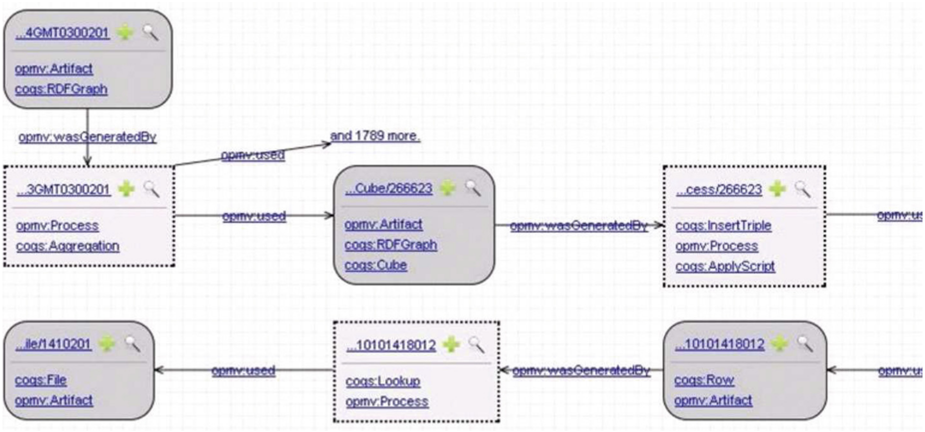


Fig. 4. Visualisation interface for the retrospective provenance of the implemented ETL workflow.

process take?. By adding a Cogs layer to the OPMV layer it is possible to define queries referring to specific classes within the ETL environment, such as *What are the RDF data sources used to generate this data value?*, *Which extractors are used in this workflow?*, *What are the schema transformation operations?*, *Which formulas were used to calculate this indicator?*, *Which is the source code artifact behind this data transformation?*. More specifically to the use case, queries such as *How long did all lookups take?*, *What scripts have been used to transform the data into RDF?*, *To which values constant factors have been applied?*, *Which aggregation functions were used to calculate this indicator?* could be answered to support the interpretation of different ETL executions. The third layer contains information which is domain-specific (not likely to be directly interoperable with other systems). It consists of specific operations (e.g., reference to specific data mining algorithms), schema-level information (such as table names and column names) and program code references (as in the example instantiation). This third layer specialises the classes of the Cogs layer: the presence of the Cogs classes can be used to facilitate the entity resolution among domain-specific layers of different contexts. The use of the Cogs vocabulary allows an increase of the query expressivity in relation to OPMV, allowing queries over the ETL elements. In addition to the direct interoperability increase provided by Cogs-compatible systems, the additional semantics of Cogs can facilitate knowledge discovery between provenance workflows, facilitating the inductive learning and semantic reconciliation of entities in the domain-specific layer.

Compared to previous works, the proposed provenance model focuses on providing a standards-based solution to the interoperability problem, relying on the structure of a community-driven provenance model (OPM) to build a provenance model for ETL. Linked Data standards are used for leveraging the accessibility of provenance descriptors. The proposed provenance model is able to provide a terminology-based semantic description of ETL workflows both in the prospective and retrospective provenance scenarios. The model is targeted towards a pay-as-you-go semantic interoperability scenario: the semantics of each workflow activity can be described with either a partial or a complete provenance descriptor.

7 Conclusion and Future Work

This work presented a provenance model for ETL workflows, introducing *Cogs*,[7] a vocabulary for modelling ETL workflows based on the Open Provenance Model (OPM). The proposed vocabulary was built aiming towards the provision of a semantically interoperable provenance model for ETL environments. The vocabulary fills a representation gap of providing an ETL provenance model, a fundamental element for increasingly complex ETL environments. The construction of the vocabulary is based on the determination of a set of requirements for modelling provenance on ETL workflows. The proposed provenance model presents a high coverage of the set of requirements and was applied to a realistic ETL workflow scenario. The model relies on the use of Linked Data standards.

A more thorough evaluation of the interoperability gained when using Cogs is planned. Future work include the refinement of the vocabulary based on feedback from users. The provenance model proposed in this paper was already implemented to describe interactive data transformations from the Google Refine platform [13]. The verification of the interoperability between Google Refine and an open source ETL platform is planned.

Acknowledgements. The work presented in this paper has been funded by Science Foundation Ireland under Grant No.SFI/08/CE/I1380 (Lion-2), by the German Ministry of Education and Research (BMBF) within the SMART project (Ref. 02WM0800) and the European Community's Seventh Framework Programme FP7/2007–2013 (PlanetData, Grant 257641).

References

1. Altinel, M., Brown, P., Cline, S., Kartha, R. Louie, E., Markl, V., Mau, L., Ng, Y.-H., Simmen, D., Singh. A.: Damia: a data mashup fabric for intranet applications. In: Proceedings of the 33rd International Conference on Very Large Data Bases (2007)
2. Becker, K., Ghedini, C.: A documentation infrastructure for the management of data mining projects. Inf. Softw. Technol. **47**, 95–111 (2005)
3. Cui, Y., Widom, J.: Lineage tracing for general data warehouse transformations. VLDB J. **12**, 41–58 (2003)
4. Davidson, S., Buneman, P., Kosky, A.: Semantics of database transformations. In: Thalheim, B. (ed.) Semantics in Databases 1995. LNCS, vol. 1358. Springer, Heidelberg (1998)

[7] http://vocab.deri.ie/cogs.

5. El Akkaoui, Z., Zimanyi, E.: Defining ETL worfklows using BPMN and BPEL. In: Proceedings of the ACM Twelfth International Workshop on Data Warehousing and OLAP, DOLAP 2009, New York, NY, USA, pp. 41–48 (2009)
6. Freitas, A., Knap, T., O'Riain, S., Curry, E.: W3P: building an OPM based provenance model for the Web. Future Gener. Comput. Syst. **27**, 766–774 (2010)
7. Freitas, A., Legendre, A., O'Riain, S., Curry, E.: Prov4J: a semantic Web framework for generic provenance management. In: Second International Workshop on Role of Semantic Web in Provenance Management (SWPM 2010), 2010
8. Galhardas, H., Florescu, D., Shasha, D., Simon, E., Saita, C.-A.: Declarative data cleaning: language, model, and algorithms. In: Proceedings of the 27th International Conference on Very Large Data Bases (2001)
9. Kietz, J.-U., Serban, F., Bernstein, A., Fischer, S.: Towards cooperative planning of data mining workflows. In: Proceedings of the ECML/PKDD 2009 Workshop on Third Generation Data Mining (SoKD 2009) (2009)
10. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting Cleaning. Wiley, New York (2004)
11. Moreau, L.: The open provenance model core specification (v1.1). Future Gener. Comput. Syst. **27**(6), 743–756 (2011)
12. Morik, K., Scholz, M.: The miningmart approach to knowledge discovery in databases. In: Zhong, N., Liu, J. (eds.) Intelligent Technologies for Information Analysis, pp. 47–65. Springer, Heidelberg (2003)
13. Omitola, T., Freitas, A., O'Riain, S., Curry, E., Gibbins, N., Shadbolt, N.: Capturing interactive data transformation operations using provenance workflows. In: Proceedings of the 3rd International Workshop on Role of Semantic Web in Provenance Management (SWPM 2012) (2012)
14. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. **34**, 31–36 (2005)
15. Skoutas, D., Simitsis, A.: Designing ETL processes using semantic Web technologies. In: Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP (2006)
16. Thi, A., Nguyen, B.T.: A semantic approach towards CWM-based ETL processes. In: Proceedings of I-SEMANTICS (2008)
17. Trujillo, J., Luján-Mora, S.: A UML based approach for modeling ETL processes in data warehouses. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 307–320. Springer, Heidelberg (2003)
18. Vassiliadis, P., Karagiannis, A., Tziovara, V., Simitsis, A.: Towards a benchmark for etl workflows. In: Ganti, V., Naumann, F. (eds.) QDB, pp. 49–60 (2007)
19. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for ETL processes. In: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP (2002)
20. Trio, J.W.: A system for integrated management of data, accuracy, and lineage. In: Innovative Data Systems Research (CIDR 2005) (2005)

Using SPIN to Formalise XBRL Accounting Regulations on the Semantic Web

Seán O'Riain[1], John McCrae[2](✉), Philipp Cimiano[2], and Dennis Spohr[2]

[1] Digital Enterprise Research Institute, NUI, Galway, Ireland
sean.oriain@deri.org
[2] Semantic Computing Group, University of Bielefeld, Bielefeld, Germany
{jmccrae,cimiano,dspohr}@cit-ec.uni-bielefeld.de

Abstract. The eXtensible Business Reporting Language (XBRL) has standardised consolidated financial reporting and through its machine readable format facilitates access to and consumption of financial figures contained within the report. Formalising XBRL as RDF facilitates the leveraging of XBRL with Open Financial Data. Previous XBRL to Semantic Web transformations have however concentrated on making the semantics of its logical model explicit to the exclusion of accounting regulatory validation rules and constraints found within the XBRL calculation linkbases. Using off-the-shelf Semantic Web technologies this paper investigates the use of the SPARQL Inferencing Notation (SPIN) with RDF to formalise these accounting regulations found across XBRL jurisdictional taxonomies. Moving beyond previous RDF to XBRL transformations we investigate how SPIN enhanced formalisation enables financial instrument fact inferencing and sophisticated consistency checking. SPIN formalisations are further used to evaluate the correctness of reported financial data against the calculation requirements imposed by accounting regulation. Our approach illustrated through the use of use case demonstrators outlines that SPIN usage meets central requirements for financial constraint regulatory modelling.

1 Introduction

XBRL has seen steady adoption due to its role as the required reporting format by regulatory authorities such as the U.S. Securities and Exchange Commission (SEC)[1] and the Committee of European Banking Supervisors[2], for consolidated financial filings and general business reporting. XBRLs[3] adoption has enhanced accessibility and availability of standardised financial reporting but interoperability with other standards such as RDF for linked data remain at early stages of realisation. Enhanced interoperability between multiple financial sources can be facilitated through the adoption of standard data representation formalisms

[1] See http://xbrl.sec.gov/.

[2] COmmon REPorting Framework, http://www.eba.europa.eu/Supervisory-Reporting/COREP.aspx/.

[3] XBRL V2.1 Taxonomy Specification http://www.xbrl.org/SpecRecommendations/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 58–72, 2015.
DOI: 10.1007/978-3-662-46641-4_5

but dependence on manual intervention directly impacts accuracy and timely analysis of financial reports [10].

XBRL, an XML-serialised format defines financial concepts and their relations based on jurisdictional Generally Accepted Accounting Practises or GAAPs. Relations, termed roles, define semantic views and include financial instrument calculation rules based on specific GAAPs. For example the value of financial instrument *Assets* is derived in part (calculated) from the summation of *Current assets* and *Non-current assets* found in the XBRL calculation linkbase[4], in addition to more complex accounting rules defined in the XBRL formula linkbases.

Attempts to make XBRL interoperable with other Web based information through the automated conversion of XBRL taxonomies and instances to their RDF [13] or OWL [1,3,6,9,15] equivalent representations have received interest in recent years. Financial statement contained within XBRL formatted financial reports are adequately transformed, but semantics inherent within the calculation and formula rule constraints have yet to be addressed as part of the formalisation activity [9]. Modelling these rule relationships remains important as they can be used directly to both assist hierarchical modelling and data validation. The rules themselves represent jurisdiction specific reporting requirements for financial instrument calculation and their verification an opportunity to assess the level of accounting standard conformance.

In the paper we present an approach to formalising XBRL calculation rule semantics as part of an XBRL to RDF transformation. Regulatory requirement are semantically modelled using SPIN[5], a de-facto industry standard used to represents SPARQL rules and constraints for Semantic Web models. SPIN is supported by the TopBraid Composer[6] technology offering, standardisation efforts are in progress and an open-source Java API exists[7]. Resulting representation are used to conduct initial experiments on the inferencing of financial instrument values and consistency checking of financial instrument reported values without the overhead of having to customised existing XBRL software. The experiments seek to establish whether the SPIN vocabulary developed specifically for rule representation, captures the intended semantics of accounting regulations in a transparent and intuitive manner.

The remainder of the paper is structured as follows. Section 2 provides a fundamental grounding in XBRL financial reporting using XBRL taxonomies Sect. 2.1, Semantic Web representation formalisms Sect. 2.2 and SPIN Sect. 2.3. Section 2.2 within an XBRL context looks at (i) existing work on XBRL to RDF transformation and (ii) the wider use of SPARQL for business information querying. Section 2.3 positions our efforts as relating to García and Gil [6] and Bao et al. [1]. Section 3 looks at XBRL to RDF transformations. The overall approach

[4] Allow the combination of concept labels and references and relationship definition between concepts. The calculation linkbase defines basic taxonomy validation rules applicable to taxonomy instances.

[5] Specification at http://www.spinrdf.org/.

[6] See http://topquadrant.com/products/TB_Suite.html.

[7] Refer to http://www.spinrdf.org/faq.html for further detail.

taken to RDF/OWL transformation is introduced in Sect. 3.1 with emphasis on the formalisation of XBRL calculations as SPIN rule constraints (Sect. 3.2), and the resulting representational use in inferring financial instrument values (Sect. 3.3), to determine regulatory accounting standard rule conformance. The experiments Sect. 4 reports on initial investigations into inferring these financial instrument values (Sect. 4.1) and their consistence checking (Sect. 4.2). Discussion on experimental results along with the SPARQL-based SPIN approach to generally available rule-based approaches is presented in Sect. 4.3 along with conclusions and future work in Sect. 5.

2 Background and Related Work

This section first introduces XBRL model fundamentals and GAAP taxonomy use within consolidated financial reporting (Sect. 2.1) to provide an understanding of wider XBRL use and utility. Previous efforts that addressed XBRL transformation into Semantic Web representations equivalent are discussed in (Sect. 2.2) and introduction to SPIN fundamentals given in Sect. 2.3. The use of SPARQL for business-related modelling issues is also mentioned.

2.1 XBRL Financial Reporting Fundamentals

XBRL standardises financial reporting and with its machine-interpretable format makes corporate reports easier to consume and integrate. It removes dependency on proprietary formats usage in financial filings and level of manual effort required for report preparation and compilation [7]. With regards to financial information, XBRL increased cross-company interoperability, figures comparison and more wider its transparency. XBRL has also been adopted outside of the financial domain for use as a generalised reporting format by the Global Reporting Initiative (GRI) to report on sustainability issues[8].

XBRL has as central the notion of *taxonomies* that describe core concepts use in particular reporting situations such as the International Financial Reporting Standard (IFRS) or jurisdictional GAAPs, and *instance documents* that represent financial report content (refer to Fig. 1). Instance documents provide financial instrument facts such as the concept monetary value and units. Facts have links to a reporting context that additionally specify entities (termed dimensions by XBRL) such as reporting company and time period to which the report contents are relevant for. The example (1), extracted from the 2009 SAP annual report, states that reporting period ending 31-12-2009, *Cash and cash equivalents* of €1.88 billion were reported. The example also illustrates how values in an instance document are linked through their context reference *(contextRef)* to the taxonomy monetary concept *Cash and cash equivalents*, which has a debit cash balance.

[8] Further details found at https://www.globalreporting.org/.

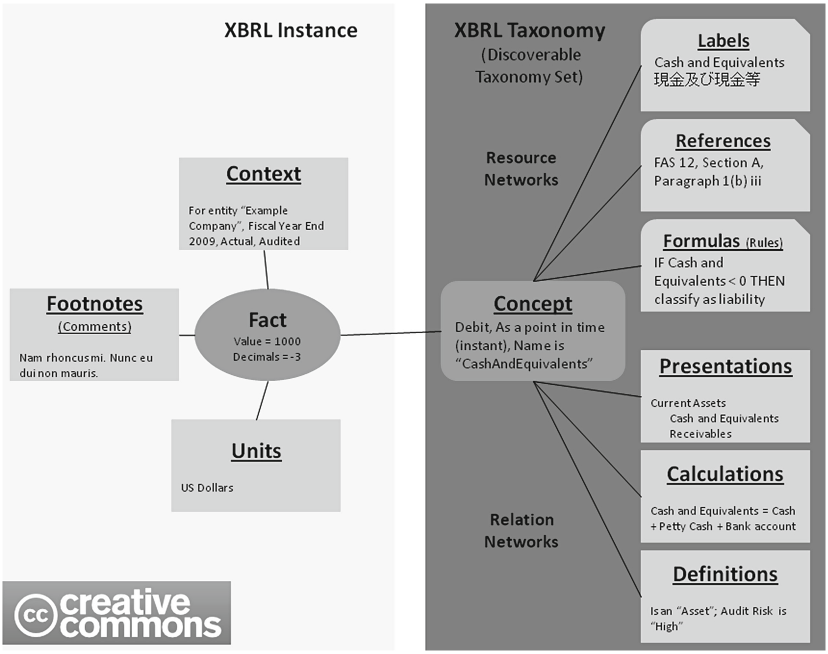

XBRL Instance
Context
For entity "Example Company", Fiscal Year End 2009, Actual, Audited
Footnotes
(Comments)
Nam rhoncus mi. Nunc eu dui non mauris.
Fact
Value = 1000
Decimals = -3
Units
US Dollars
creative commons
XBRL Taxonomy
(Discoverable Taxonomy Set)
Resource Networks
Labels
Cash and Equivalents
現金及び現金等
References
FAS 12, Section A, Paragraph 1(b) iii
Formulas (Rules)
IF Cash and Equivalents < 0 THEN classify as liability
Concept
Debit, As a point in time (instant), Name is "CashAndEquivalents"
Presentations
Current Assets
Cash and Equivalents
Receivables
Calculations
Cash and Equivalents = Cash + Petty Cash + Bank account
Relation Networks
Definitions
Is an "Asset"; Audit Risk is "High"

Fig. 1. High-level model of XBRL by Charles Hoffman (http://xbrl.squarespace.com)

```
(1) <context id="FYp0Qp0e">
     <entity>
      <identifier scheme="http://www.sec.gov/CIK">0000943042</identifier>
     </entity>
     <period>
      <instant>2009-12-31</instant>
     </period>
    </context>
     ...
    <ifrs:CashAndCashEquivalents contextRef="FYp0Qp0e" decimals="-6"
     unitRef="EUR">1884000000</ifrs:CashAndCashEquivalents>
```

XBRL also allows companies add their own specific taxonomy extensions to facilitate situations where reported value are not covered or their calculation deviates from standard vocabularies. SAP for example to report on *Software revenue*, a concept not defined in the IFRS taxonomy, first added a custom extension to the taxonomy before adding the financial instrument facts to the instance document. Each taxonomy concept is linked to a set of XLink *linkbases*[9] (referred to as *resource* and *relation networks* in the figure). These specify labels for the concepts, in addition to how concept values should be presented and displayed across differing statement types.

[9] e.g. the U.S. 2009 GAAP taxonomy contains 450 linkbases.

For example, the IFRS[10] *Statement of financial position, current/non-current* specifies that the concept *Assets* be hierarchically displayed above *Non-current assets* and *Current assets* in the consolidated filing, whereas *Statement of financial position, order of liquidity* hierarchically places *Assets* above a different financial instrument – *Property, plant and equipment*, *Investment property.* Such relationships within XBRL are defined using XML Linking Language (XLink:[11].) arcs and extended links, which can in turn be used to group and link any number of additional arcs to other resources. XBRL presentation linkbase responsible for specifying the report presentation specify how the concepts are linked using *parent-child* arcs, and that some set of presentation arcs are associated with a particular accounting standard using an extended link role.

Our investigations focus on the modelling of an XBRL calculation arc taken from the calculation linkbase of the IFRS taxonomy. The XBRL formula linkbase is considered outside current research scope and notes as future work. The calculation linkbases itself specifies how the IFRS accounting standard defines concept value calculations (Sects. 3.2). Example (2) below outlines such an XBRL calculation arc representation extracted from the calculation linkbase of the IFRS taxonomy used by SAP.

```
(2) <loc xlink:type="locator" xlink:label="ifrs_CashAndCashEquivalents"
      xlink:href="http://xbrl.iasb.org/taxonomy/2009-04-01/
                  ifrs-cor_2009-04-01.xsd#ifrs_CashAndCashEquivalents" />
    <calculationArc xlink:type="arc"
      xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"
      xlink:from="ifrs_CurrentAssets"
      xlink:to="ifrs_CashAndCashEquivalents" order="1" weight="1" />
```

The concept *CurrentAssets* is linked through the arc-role to *CashAndCashEquivalents* through a *summationitem* relation. Although an arc weight of 1 indicates that concept values be summed and conversely a value of −1 that they be subtracted, XBRL calculation links currently only cater for financial item summation. As calculation arc representation should form part of the wider XBRL to RDF transformation we look at previous converter tools and approaches that automate the conversion of XBRL taxonomies and instances to their RDF equivalents.

2.2 Semantic Web Representation of XBRL

Ontologies and Financial Data. – out in conclusion

XBRL to Semantic Web Transformation. Bao et al. [1] represents the most current work on XBRL transformation to a Semantic Web standard. Implicit XBRL semantics that concentrate on linkbase are preserved ([1], p. 144) and

[10] http://www.ifrs.org/XBRL/IFRS+Taxonomy/IFRS+Taxonomy+2011/IFRS+Taxonomy+2011.htm.

[11] http://www.w3.org/TR/xlink/.

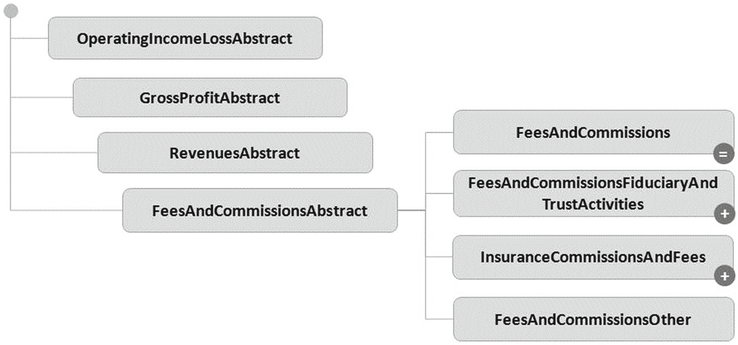

OperatingIncomeLossAbstract
GrossProfitAbstract
RevenuesAbstract
FeesAndCommissionsAbstract
FeesAndCommissions
=
FeesAndCommissionsFiduciaryAndTrustActivities
+
InsuranceCommissionsAndFees
+
FeesAndCommissionsOther

Fig. 2. US GAAP Presentation Hierarchy Extract [11]

an OWL-based model generated. Extended link roles are however considered "non-semantic" and omitted from transformation consideration. These link roles (Sect. 2.1) are used within XBRL linkbases to limit assertion scope, for example to a particular type of statement, making arcs globally applicable. The Bao et al. [1] method for linkbase arc representation also holds as the basic assumption that the interpretations of an arc is consistent between respective concept instances that it links. This usage deviates from the XBRL specification which only notes that arcs relate "one XBRL concept to one other XBRL concept"[12]. The assumption holds in situations where concepts have accompanying instances with values but XBRL also exhibits variation between what an instance document discloses and the taxonomy provides as hierarchy as illustrated by Fig. 2. The semantic hierarchy for *us-gaap:FeesAndCommissions* is established across presentation and calculation linkbases. Presentation details the parent-child relationships of *OperatingIncomeLossAbstract*, *GrossProfitAbstract*, *RevenuesAbstract* and *FeesAndCommissionsAbstract*. *FeesAndCommissionsAbstract* in turn has a parent-child relationship with financial instruments *FeesAndCommissions*, *FeesAndCommissionsFiduciaryAndTrustActivities*, *InsuranceCommissionsAndFees* and *FeesAndCommissionsOther*. The *FeesAndCommissionsAbstract* concept has no accompanying instance. Querying the calculation linkbase for *FeesAndCommissions* establishes thats its instance value as such is in fact derived from *FeesAndCommissionsFiduciaryAndTrustActivities* and *InsuranceCommissionsAndFees*. Transitive closure assessment of allowable calculations would determine what pairs exist and make querying straightforward. Although related through `parent-child` arcs, it is not clear now how [1] caters for these cases.

Bao et al. approach does not "mechanically" preserve XBRL structural properties while other approaches by Raggett [13], Declerck and Krieger [3] and García and Gil [6] do. Our approach also preserves XBRL structural information

[12] http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm#_3.5.3.9.

but adds further interpretation to specifically address the lack of mathematical relationship modelling inherent in Semantic Web vocabularies [9].

SPARQL usage in business information querying. Fürber and Hepp [5] propose the use of SPARQL for detection of data quality problems, SPIN to model consistency constraints and the TopBraid Composer to detect constraint inconsistencies. We note the usefulness of their approach to XBRL data and leverage it for SPIN constraint specification, extending it to cater for constraint checking on data which has been inferred through iterative rule application.

2.3 SPIN Fundamentals

The SPARQL Inferencing Notation (SPIN) has as its origin the necessity to perform constraints checking with closed-world semantics (cf. Sect. 4.3) and calculations on property values, a task largely unsupported in current Semantic Web formalisms [9,11,16]. SPIN allows SPARQL rules and constraints definition on Semantic Web models"[13]. By attaching SPARQL queries to resources using RDF properties `spin:rule`, `spin:constraint`, and superproperty `spin:query`. `spin:rule` accepts SPARQL CONSTRUCT queries as value and using statements in the queries WHERE clause new triples can be inferred. Basic example are provided in (3) with SPIN representation in (4).

```
(3) CONSTRUCT { ?this a ?c2 . }
    WHERE { ?c1 rdfs:subClassOf ?c2 .
            ?this a ?c1 . }
(4) [ a sp:Construct ; sp:templates ([ sp:subject spin:_this ;
                        sp:predicate rdf:type ;
                        sp:object _:b1 ])
         sp:where ([ sp:subject _:b3 ;
                        sp:predicate rdfs:subClassOf ;
                        sp:object _:b1
                     ]  [  sp:subject spin:_this ;
                       sp:predicate rdf:type ;
           sp:object _:b3 ]) ]
```

The example, based on that available from TopBraid's SPIN website[14], formalises the semantics of `rdfs:subClassOf` and illustrate variable use. In SPIN the variable `?this` refers to the resource `spin:_this`, whereas generally in a SPARQL query variables is mapped to blank nodes. The variable refers to an instance of the class to which the rule has been attached. If the rule in Example (3) was attached to `owl:Thing`, it would be applied to every instance of `owl:Thing` satisfying the WHERE clause statements. Using SPARQL ASK queries the `spin:constraint` property can be used to model consistency constraints as an evaluation to true indicates a constraints violation for the respective instance. Finally, the general property `spin:query` can be used to attach SPARQL queries to RDF resources (i.e. also SELECT queries).

[13] See http://www.spinrdf.org.

[14] http://topbraid.org/spin/owlrl-all.html.

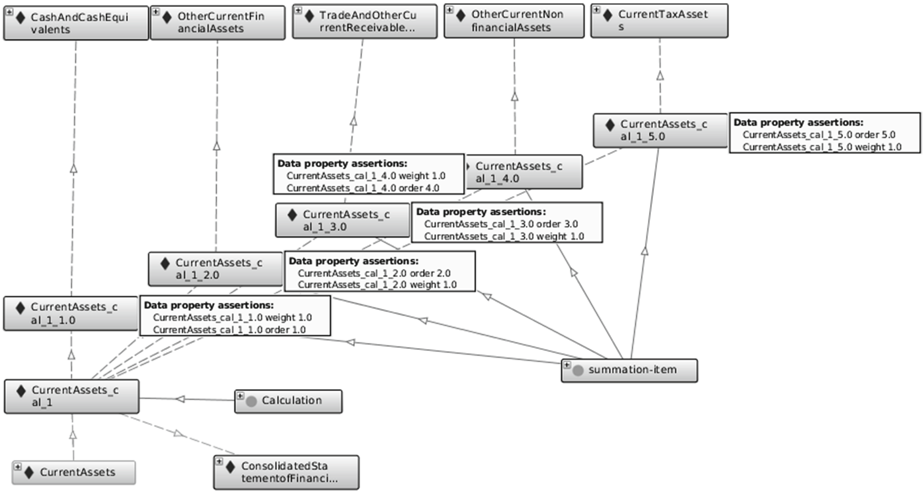

CashAndCashEquivalents
OtherCurrentFinancialAssets
TradeAndOtherCurrentReceivable...
OtherCurrentNonfinancialAssets
CurrentTaxAssets
CurrentAssets_cal_1_5.0
Data property assertions:
CurrentAssets_cal_1_5.0 order 5.0
CurrentAssets_cal_1_5.0 weight 1.0
Data property assertions:
CurrentAssets_cal_1_4.0 weight 1.0
CurrentAssets_cal_1_4.0 order 4.0
CurrentAssets_cal_1_4.0
CurrentAssets_cal_1_3.0
Data property assertions:
CurrentAssets_cal_1_3.0 order 3.0
CurrentAssets_cal_1_3.0 weight 1.0
CurrentAssets_cal_1_2.0
Data property assertions:
CurrentAssets_cal_1_2.0 order 2.0
CurrentAssets_cal_1_2.0 weight 1.0
CurrentAssets_cal_1_1.0
Data property assertions:
CurrentAssets_cal_1_1.0 weight 1.0
CurrentAssets_cal_1_1.0 order 1.0
summation-item
CurrentAssets_cal_1
Calculation
CurrentAssets
ConsolidatedStatementofFinanci...

Fig. 3. Structural representation of the calculation of `ifrs:CurrentAssets` (generated with OntoGraf Protégé)

The queries CONSTRUCT and ASK, detailed in Sect. 3, are primarily used to capture the intended semantics of accounting regulations. In addition to standard SPARQL operators like UNION, OPTIONAL and FILTER, SPIN supports SPARQL extensions such as the ARQ keyword LET[15], which allows variables value assignment as well as custom function definition.

3 XBRL Transformation to RDF

This section discusses the conversion of XBRL to RDF focusing on the SPIN-based representation of accounting regulations. After a brief introduction to the general underlying ideas in Sect. 3.1 the representation of calculation rules (Sect. 3.2) and consistency constraints used (Sect. 3.3) to transform the accounting regulations from XBRL to RDF are discussed.

3.1 Rule Generation Approach

Our approach like Bao et al. adheres to an XBRL “logical model” representation that preserves structural information from the original data [1]. The resulting RDF representation is interoperable with Open Linked Data sources, supports inferencing, allows consistency checking, and the model itself can be queried allowing investigations such as; what is the hierarchy for the *“Assets”* within *“Statement of financial position”*)? Figure 3 outlines the structural representation of the `ifrs:CurrentAssets` calculation. The `ifrs:CurrentAssets` instance

[15] For example http://jena.sourceforge.net/ARQ/assignment.html.

ifrs:CurrentAssets_cal_1 is linked to a Calculation class instance (bottom left), in addition to other instances, all of which form part of the calculation. Each instances represents an XBRL calculation arc reification with weight and order attributes as values of datatype property statements.

In addition, we include XBRL links between concepts as direct triple links, which enables the result to be easily queried using SPARQL. This leads to an issue in OWL as we indicate these concepts to be classes, hence under OWL2 DL this leads to *punning*, that is referring to a class or property also as an individual. This practise is widely used [8] and often, as here, this *punning* is more practical than modelling complex formal axioms.

3.2 SPIN Rules for XBRL Calculations

In order to model the regulatory calculation of monetary concepts in a Semantic Web compliant way, SPIN rules based on the data contained in XBRL calculation linkbases were generated. Specifically calculation arcs such as those shown in example (2), are converted into their SPIN representation (below), which represents the calculation of ifrs:CurrentAssets.

```
(5) CONSTRUCT { ?this xbrlrdf:calculatedValue ?cvalue . }
    WHERE { ?this xbrli:contextRef ?context; xbrli:unitRef ?unit .
     ?x0 a ifrs:CurrentTaxAssets ;
           xbrli:contextRef ?context; xbrli:unitRef ?unit;
           xbrlrdf:calculatedValue ?cv0 .
     ?x1 a ifrs:OtherCurrentNonfinancialAssets ;
           xbrli:contextRef ?context; xbrli:unitRef ?unit;
           xbrlrdf:calculatedValue ?cv1 .
     ?x2 a ifrs:TradeAndOtherCurrentReceivables ;
           xbrli:contextRef ?context; xbrli:unitRef ?unit;
           xbrlrdf:calculatedValue ?cv2 .
     ?x3 a ifrs:OtherCurrentFinancialAssets ;
           xbrli:contextRef ?context; xbrli:unitRef ?unit;
           xbrlrdf:calculatedValue ?cv3 .
     ?x4 a ifrs:CashAndCashEquivalents ;
           xbrli:contextRef ?context; xbrli:unitRef ?unit;
           xbrlrdf:calculatedValue ?cv4 .
    LET (?cvalue := 1.0 * ?cv0 + 1.0 * ?cv1 + 1.0 * ?cv2 +
                    1.0 * ?cv3 + 1.0 * ?cv4 ) . }
```

Each of the graph patterns in the query represents one calculation arc. References to URIs for the context and units, ensure that the values of relevant instances are only taken into account. This excludes cases where a particular value refers to different entities or different segments of the same entity, as well as cases in which values are reported for different time periods. This is a normal occurrence as financial statements generally contain figures for both the current and preceding reporting periods. Finally, the LET clause specifies how the values of the individual concept instances should be combined. For accounting rules, this is limited to summation and subtraction. XBRL provides a single arc role `summation-item` for this purpose, and uses the value of the *weight* attribute – either 1 or −1 (example (2) above) – to indicate whether the value of a particular concept should be added or subtracted. For further more complex calculations possible using SPIN we refer the reader to the SPIN vocabulary specification[16].

The example illustrates how rules can make use of previously calculated values to calculate further values. This allows value calculation for composite monetary concepts (i.e. those whose values are calculated on the basis of the values of other concepts) by specifying values for atomic concepts and then applying the rules iteratively (see Sect. 4.1 for more details).

Moreover, it should be noted that in our RDF representation, rules are not modelled as blank nodes, but instead carry a URI. This has the benefit of enabling other instances to dereference the rules, allowing a particular calculation rule be reused across different financial reports of the same accounting standard. It also further allows attachment of additional properties to a rule and a reference to the type of financial statement to which the rule applies. Therefore, in addition to the actual rule, the instance representing the rule in (5) is the subject of a triple relating it to the URI representing SAP's *Consolidated Statement of Financial Position* by means of `xbrlrdf:roleRef`.

As mentioned, rules can be executed iteratively, making use of previously inferred values. In order to make sure that a particular calculation rule with atomic concepts (i.e. those concepts which lack regulatory calculation rules attached) can also be applied, we add the default calculation shown in (6) to atomic monetary concepts. This rule then assigns the reported value of the respective instance to `xbrlrdf:calculatedValue`.

```
(6) CONSTRUCT { ?this xbrlrdf:calculatedValue ?value . }
    WHERE { ?this xbrlrdf:value ?value } .
```

Furthermore, we assume that each value is either the result of some calculation and/or is given in the initial report, and that the XBRL instance document is consistent if no concept generates two distinct calculated values.

3.3 SPIN Constraint Representation for XBRL Consistency Checking

On the basis of the SPIN rules presented above, each monetary concept which participates in some calculation and has reported values in the respective context is assigned a calculated value. The next step in modelling the accounting regulation is to specify that calculated values need to match reported values. As SPIN rules and constraints are applied to all instances of the class to which they have been attached, as well as to the instances of its subclasses, this can be achieved by attaching a single SPIN constraint to the top monetary concept:

```
(7) ASK WHERE { ?this xbrlrdf:value ?value ;
                      xbrlrdf:calculatedValue ?cvalue .
                FILTER (?value != ?cvalue) }
```

[16] http://www.spinrdf.org.

Additionally, SPIN constraints can be used to formalise more general constraints imposed by the XBRL specification. Below, we illustrate this using a constraint (restriction) which states that if two concepts have the same balance type (i.e. credit or debit), they can only be added to one another, not subtracted. In other words, the value of the XBRL weight attribute, which is preserved in our structural representation of XBRL, has to be positive:

```
(8) ASK WHERE { ?this xlink:from ?from ; xlink:to ?to ;
                     xbrlrdf:weight ?weight .
                ?from a ?balance . ?to a ?balance .
                ?balance rdfs:subClassOf xbrlrdf:BalanceType .
                FILTER (?weight < 0) }
```

4 Experiments

The SPIN rules and constraints given in Sect. 3 above can be used to infer values for instances of the respective reporting concepts, as well as check their consistency. We next illustrate how these representations integrate with TopBraid Composer, taking as example the SAP 2009 annual report reported against its custom extension of the IFRS 2009 taxonomy.

4.1 Inferring Values for Financial Instrument Figures

To investigate whether the rules outlined in Sect. 3 performed as anticipated we first modified the report ensuring that it only contained instance values for *atomic* monetary concepts. Composite monetary concepts were assigned values through iterative application of the SPIN calculation rules. This then allowed evaluation as to whether: the available information triggered the application of all rules necessary to calculate the missing information; and whether the calculated figures corresponded to those reported in the original filing. Table 1 summarises results, with the original report figures recorded in parentheses.

Tuples 3 and 4 of the table detail that the modified report contains 351 reported values for 129 monetary concepts, compared to 482 and 171, respectively, from the original report. After applying the calculation rules, values are inferred for 97.66 % of these 171 concepts, indicating that the modified report contains 458 values, as opposed to 482 original report values. 25.11 % of the 458 inferred values are due to regulatory rules, outlining that the remaining 343 values have been inferred for atomic concepts by means of the default rule. Over all 8.54 % of all the regulatory rules available in the IFRS 2009 taxonomy and the SAP 2009 extension have been applied.

The figures illustrate that for 4 of the monetary concepts no value could be inferred. Analysis revealed that this is due to values for `ifrs:BasicEarnings-LossPerShare` and `ifrs:DilutedEarningsLossPerShare`, being missing from the XBRL report instance, despite being part of a composite concept. As a result, the corresponding rules could not be applied (see Sect. 4.3 for a discussion regarding the optionality of calculation arcs).

Table 1. Reported and inferred values in the modified annual report 2009 of SAP

	Absolute	Relative
Concepts in IFRS 2009 taxonomy and SAP extension	3,021	100.00 %
Regulatory calculation rules	492	100.00 %
Reported monetary values	351 (482)	72.82 %
Monetary concepts with reported values	129 (171)	75.44 %
Inferred monetary values	458	95.02 %
Monetary concepts with inferred values	167	97.66 %
Monetary values inferred by default rule	343	74.89 %
Monetary values inferred by regulatory rules	115	25.11 %
Regulatory rules applied	42	8.54 %
Total number of monetary values	458 (482)	95.02 %
Total number of correct monetary values	458	100.00 %

4.2 Consistency Checking of Reported Financial Instrument Values

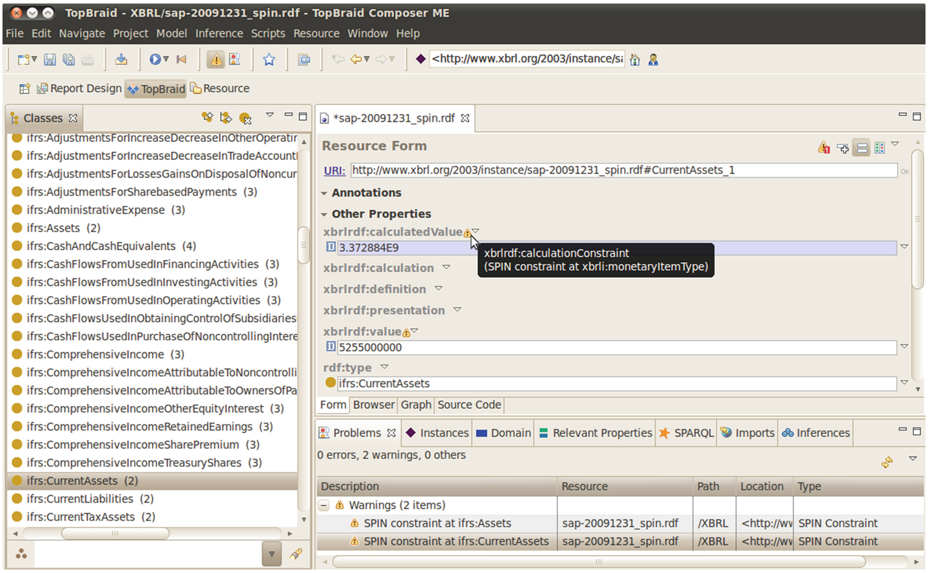
Table 1 results detail that all inferred values were correct, which in turn reflects XBRL's contribution to overall financial reporting consistency. The results also indicate that the functionality of the SPIN constraint were not evaluated by this method. To address this situation the reported value of the atomic concept `ifrs:CashAndCashEquivalents` in the original report was changed, and the change tracked to see whether the change would be propagated along the calculation "hierarchy", and yield inconsistent composite concepts. Result displaying an instance of `ifrs:CurrentAssets` after rule application and constraint checking in TopBraid Composer are reported in Fig. 4.

The figure demonstrates that TopBraid Composer correctly flags the instance where the calculated value differs from the reported one and where inconsistency have arisen at concept that has `ifrs:CurrentAssets` as calculation component (i.e. `ifrs:Assets`).

4.3 Default Values

The previous sections detail how the SPARQL CONSTRUCT rules and ASK queries capture the semantics of the XBRL data in an intuitive and transparent way. However the XBRL does allow calculation arcs to have an assumed value of zero. This usage of default values, cannot be naturally handled with monotonic logics such as OWL and would normally require the use of a formalism such as Reiter's default logic [14]. There is an extension of the Rule Interchange Format (RIF) called RIF-SILK[17] that allows such rules to be modelled, however the reasoning is much more complex and not supported by any SPARQL repositories.

[17] See http://silk.semwebcentral.org/RIF-SILK.html.


TopBraid - XBRL/sap-20091231_spin.rdf - TopBraid Composer ME
File Edit Navigate Project Model Inference Scripts Resource Window Help
<http://www.xbrl.org/2003/instance/s
Report Design
TopBraid
Resource
Classes
ifrs:AdjustmentsForIncreaseDecreaseInOtherOperatir
ifrs:AdjustmentsForIncreaseDecreaseInTradeAccount
ifrs:AdjustmentsForLossesGainsOnDisposalOfNoncur
ifrs:AdjustmentsForSharebasedPayments (3)
ifrs:AdministrativeExpense (3)
ifrs:Assets (2)
ifrs:CashAndCashEquivalents (4)
ifrs:CashFlowsFromUsedInFinancingActivities (3)
ifrs:CashFlowsFromUsedInInvestingActivities (3)
ifrs:CashFlowsFromUsedInOperatingActivities (3)
ifrs:CashFlowsUsedInObtainingControlOfSubsidiaries
ifrs:CashFlowsUsedInPurchaseOfNoncontrollingIntere
ifrs:ComprehensiveIncome (3)
ifrs:ComprehensiveIncomeAttributableToNoncontrolli
ifrs:ComprehensiveIncomeAttributableToOwnersOfPa
ifrs:ComprehensiveIncomeOtherEquityInterest (3)
ifrs:ComprehensiveIncomeRetainedEarnings (3)
ifrs:ComprehensiveIncomeSharePremium (3)
ifrs:ComprehensiveIncomeTreasuryShares (3)
ifrs:CurrentAssets (2)
ifrs:CurrentLiabilities (2)
ifrs:CurrentTaxAssets (2)
*sap-20091231_spin.rdf
Resource Form
Annotations
Other Properties
Form Browser Graph Source Code
Problems
Instances
Domain
Relevant Properties
SPARQL
Imports
Inferences

URI: http://www.xbrl.org/2003/instance/sap-20091231_spin.rdf#CurrentAssets_1

xbrlrdf:calculatedValue: 3.372884E9

xbrlrdf:calculationConstraint
(SPIN constraint at xbrli:monetaryItemType)

xbrlrdf:calculation

xbrlrdf:definition

xbrlrdf:presentation

xbrlrdf:value: 5255000000

rdf:type: ifrs:CurrentAssets

0 errors, 2 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (2 items)				
SPIN constraint at ifrs:Assets	sap-20091231_spin.rdf	/XBRL	<http://wv	SPIN Constraint
SPIN constraint at ifrs:CurrentAssets	sap-20091231_spin.rdf	/XBRL	<http://wv	SPIN Constraint

Fig. 4. View of an instance in TopBraid Composer after introducing an incorrect value

Instead, we simply require that missing values are explicitly marked as 0, generally applied as a pre-processing step.

5 Conclusion and Future Work

The paper outlines the capability of RDF-compatible SPARQL Inferencing Notation – SPIN, to transform the regulatory rules expressed by the XBRL calculation linkbase to their RDF equivalent. The resulting representation was evaluated against XBRL financial data, with respect to inferring values for instances of monetary concepts and also checking their consistency. Additionally the use of the representation to formalise additional constraints to address data quality issues was discussed.

The approach taken can be extended to cater for the more complex mathematical operations of the XBRL formula specification[18]. For example the formula specification defines that value calculations apply to instances that refer to *identical* contexts, and more generally to concept instances which are *p(arent)-equal*, *c(ontext)-equal* and *u(nit)*-equal. p-equality and u-equality have been previously shown through rule attachment to the composite class and including the reference to the unit in the rule. Alternatively c-equality could be inferred beforehand, by specifying that two contexts which share the same entity and period are linked by `owl:sameAs`. When applying the rules iteratively to a repository that is OWL-aware, the rules shown above can be applied as is.

[18] http://www.xbrl.org/Specification/formula/REC-2009-06-22/formula-REC-2009-06-22.html.

Although SPIN's viability has been demonstrated comparison with others rule languages such as the Semantic Web Rule Language[19] [16] would contribute towards selection and best practises recommendation for Semantic Web rule format representation.

Arguments for financial information integration include the ability to conduct financial metrics comparison [2] and querying of heterogeneous data sets to gain wider holistic insight [4]. For Linked Data driven information systems, data abstraction presents challenges for financial integration [12] and financial values comparison. Semantic Web offers a level of interoperability between data sources that would assist comparability based on the transformation of financial data, such as XBRL to RDF (e.g. [9,11,15]. Financial standards interoperability also faces additional challenges from different jurisdictional and regulatory rules. The business reporting community is actively considering the ontology as an architecture to accommodate multiple XBRL jurisdiction variations [16]. Within this context the ability of Semantic Web vocabularies to semantically model mathematical relations contained in the XBRL calculation and formula will become increasingly important.

Acknowledgements. The work presented in this paper has been funded by the Multilingual Ontologies for Networked Knowledge (MONNET) project under EU Grant No. 248458, and Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. Bao, J., Rong, G., Li, X., Ding, L.: Representing financial reports on the semantic web: a faithful translation from XBRL to OWL. In: Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.) RuleML 2010. LNCS, vol. 6403, pp. 144–152. Springer, Heidelberg (2010)
2. Debreceny, R.: Feeding the information value chain: deriving analytical ratios from XBRL filings to the SEC (2010)
3. Declerck, T., Krieger, H.U.: Translating XBRL into description logic. An approach using Protégé, Sesame & OWL. In: Business Information Systems (BIS), Klagenfurt, Germany, pp. 455–467 (2006)
4. Freitas, A., Curry, E., Oliveira, J.G., O'Riain, S.: Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. IEEE Internet Comput. **16**(1), 24–33 (2012)
5. Fürber, C., Hepp, M.: Using SPARQL and SPIN for data quality management on the semantic web. In: Abramowicz, W., Tolksdorf, R. (eds.) BIS 2010. LNBIP, vol. 47, pp. 35–46. Springer, Heidelberg (2010)
6. García, R., Gil, R.: Publishing XBRL as linked open data. In: Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW) (2009)
7. Hoffman, C., Strand, C.: XBRL Essentials. American Institute of Certified Public Accountants, New York (2001)
8. Jupp, S., Bechhofer, S., Stevens, R.: Skos with owl: don't be full-ish! In: Fifth International Workshop on OWL Experiences and Directions (2008)

[19] http://www.w3.org/Submission/SWRL/.

9. Lara, R., Cantador, I., Castells, P.: XBRL taxonomies and owl ontologies for investment funds. In: 1st International Workshop on Ontologizing Industrial Standards at the 25th International Conference on Conceptual Modelling, pp. 6–9 (2006)
10. Mueller, D., Raggett, D.: Report for the W3C Workshop on Improving Access to Financial Data on the Web (2009). http://www.w3.org/2009/03/xbrl/report.html
11. O'Riain, S., Curry, E., Harth, A.: XBRL and open data for global financial ecosystems: a linked data approach. Int. J. Account. Inf. Syst. **13**(2), 141–162 (2012). http://www.sciencedirect.com/science/article/pii/S1467089512000140, XBRL: Research Implications and Future Directions
12. O'Riain, S., Harth, A., Curry, E.: Linked data driven information systems as an enabler for integrating financial data. In: Information Systems for Global Financial Markets, Emerging Developments and Effects, pp. 239–270 (2011)
13. Raggett, D.: XBRL Import - An XBRL to RDF translator (2009). http://sourceforge.net/projects/xbrlimport
14. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**(1), 81–132 (1980)
15. Spies, M.: An ontology modelling perspective on business reporting. In: Information Systems, Vocabularies, Ontologies and Rules for Enterprise and Business Process Modeling and Management, vol. 35, pp. 404–416. Elsevier (2010)
16. Wenger, M., Thomas, M., Jr., J.B.: An ontological approach to XBRL financial statement reporting. In: 17th Americas Conference on Information Systems. Michigan, USA (2011)

An Ontology-Based Opinion Mining Approach for the Financial Domain

Juana María Ruiz-Martínez, Rafael Valencia-García,
and Francisco García-Sánchez(✉)

Facultad de Informática, Universidad de Murcia,
Campus de Espinardo, 30100 Espinardo (Murcia), Spain
{jmruymar, valencia, frgarcia}@um.es

Abstract. Opinion mining is a sub-discipline of computational linguistics that uses information retrieval techniques in order to determine whether a piece of text expresses a positive, negative or neutral opinion. In this paper, we present an approach for the opinion mining of financial news through the process of identifying their semantic polarity. Our approach relies on an algorithm that combines several gazetteer lists and leverages an existing financial ontology. The financial-related news are obtained from RSS feeds and then automatically annotated with positive or negative markers. The outcome of the process is a set of news organized by their degree of positivity and negativity. The preliminary experimental results seem promising as compared against traditional approaches.

Keywords: Opinion mining · Sentiment analysis · Financial news · Ontologies · Semantic web

1 Introduction

User participation is the primary driver of value in Web 2.0 applications. Such a simple idea has had a tremendous impact on the way in which users interact with the Web. While in traditional Web 1.0 sites, companies published content and users were just mere information consumers, in the Web 2.0 era users play a more active role in Web interactions, becoming not only consumers but also producers of information and media. In this new context, namely the Social Web, a key challenge is to understand the opinions and sentiments, not only of the general public and consumers, but also of companies, banks, and politics. Opinion mining is a recent sub-discipline at the crossroads of information retrieval and computational linguistics. The focus of opinion mining does not concern what topic a text is about, but rather what opinion that text expresses [1]. It determines whether the news and the comments in online forums, blogs or comments relating to a particular topic (financial asset, product, book, movie, etc.) are positive, negative or neutral.

Sentiment analysis is often used in opinion mining to identify sentiment, affect, subjectivity, and other emotional states in online texts [2]. In its conception, sentiment analysis referred to the processing of products' attributes from product reviews [3–5]. Nowadays, sentiment polarity analysis methods are employed in a variety of application domains including politics, sports, media, and finances [6–8]. A vast amount of

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 73–86, 2015.
DOI: 10.1007/978-3-662-46641-4_6

financial news is published regularly on the Web and the financial domain is constantly changing and evolving. Under these circumstances many authors emphasize the need for mechanisms to automatically extract sentiments from news rather than relying on the intuition of analyst as to what is good or bad news [6]. However, when dealing with automatic sentiment polarity analysis, existing techniques that involve checking the similarity between a text and a seed list of words are not sufficient. In this context, we believe that the already mature Semantic Web technology may be a valuable addition to traditional approaches.

The Semantic Web can be seen as an extension of the current Web, in which information is given a well-defined meaning, thus better enabling computers and people to work in cooperation [9]. The Semantic Web vision is based on the idea of explicitly providing the knowledge behind each Web resource in a manner that is machine processable. Ontologies [10] constitute the standard knowledge representation mechanism for the Semantic Web and can be used to structure information. The formal semantics underlying ontology languages enables the automatic processing of the information in ontologies and allows the use of semantic reasoners to infer new knowledge. In this work, we propose a semantically-enhanced mechanism for opinion extraction from natural language texts. The proposed algorithm has been extensively validated through thorough test-bed experiments in the financial domain. The methodology presented here is supported by natural language processing techniques capable of semantically annotating financial news texts complying with a financial ontology. The annotated financial news items are then further analyzed in order to group them into two separate sets, one with positive financial news and the other with negative financial news.

The rest of paper is organized as follows. Section 2 presents the technological background necessary for the development of the methodology. In Sect. 3, the platform and the way it works is described in detail. In Sect. 4, the experimental results of the evaluation are shown. Related work and the discussion are included in Sect. 5. Finally, conclusions and future work are put forward in Sect. 6.

2 Technological Background

The methodology proposed here is based on two main elements, namely, ontologies and natural language processing tools. In this section, the key features of these technologies are pointed out.

2.1 Ontologies and the Semantic Web

Ontologies constitute the standard knowledge representation mechanism for the Semantic Web [10]. The formal semantics underlying ontology languages enables the automatic processing of the information and allows the use of semantic reasoners to infer new knowledge. In this work, an ontology is seen as "a formal and explicit specification of a shared conceptualization" [10]. Ontologies provide a formal, structured knowledge representation, and have the advantage of being reusable and shareable. They also provide a common vocabulary for a domain and define, with different levels of formality, the meaning of the terms and the relations between them. Knowledge in ontologies is mainly formalized using five kinds of components: classes, relations, functions, axioms and instances [11].

Ontologies are thus the key for the success of the Semantic Web vision. The use of ontologies can overcome the limitations of traditional natural language processing methods and they are also relevant in the scope of the mechanisms related, for instance, with Information Retrieval [12], Semantic Search [13], Service Discovery [14] or Question Answering [15].

Next, the financial ontology that has been developed for the purposes of this work is described.

2.1.1 Financial Ontology

The financial domain is becoming a knowledge intensive domain, where a huge number of businesses and companies hinge on, with a tremendous economic impact in our society. Consequently, there is a need for more accurate and powerful strategies for storing data and knowledge in the financial domain. In the last few years, several finances-related ontologies have been developed. The BORO (Business Object Reference Ontology) ontology is intended to be suitable as a basis for facilitating, among other things, the semantic interoperability of enterprises' operational systems [16]. On the other hand, the TOVE ontology (Toronto Virtual Enterprise) [17], developed by the Enterprise Integration Laboratory from the Toronto University, describes a standard organization company as their processes. A further example is the financial ontology developed by the DIP (Data Information and Process Integration) consortium, which is mainly focused on describing semantic web services in the stock market domain [18]. Finally, the XBRL Ontology Specification Group, developed a set of ontologies for describing financial and economic data in RDF for sharing and interchanging data. This ontology is becoming an open standard means of electronically communicating information among businesses, banks, and regulators [19].

As part of this work, a financial ontology has been developed on the basis of the above referred ontologies, with the focus set on the stock exchange domain. The ontology, created from scratch, has been defined in OWL 2. This ontology covers three main financial concepts (see Fig. 1):

- A financial market is a mechanism that allows people to easily buy and sell financial assets such us stocks, commodities and currencies, among others. The main stock markets such as New York Stock Exchange, NASDAQ or London Stock Exchange have been modelled in the ontology as subclasses of the Stock_market class.
- The Financial Intermediary class represents the entities that typically invest on the financial markets. Examples of such entities are banks, insurance companies, brokers and financial advisers.
- The Asset class represents everything of value on which an Intermediary can invest, such as stock market indexes, commodities, companies, currencies, to mention a few. So, for instance, enterprises such as Apple Inc., General Electric or Microsoft belong to the Company concept and currencies such as US dollar or Euro are included as individuals of the Currency concept.

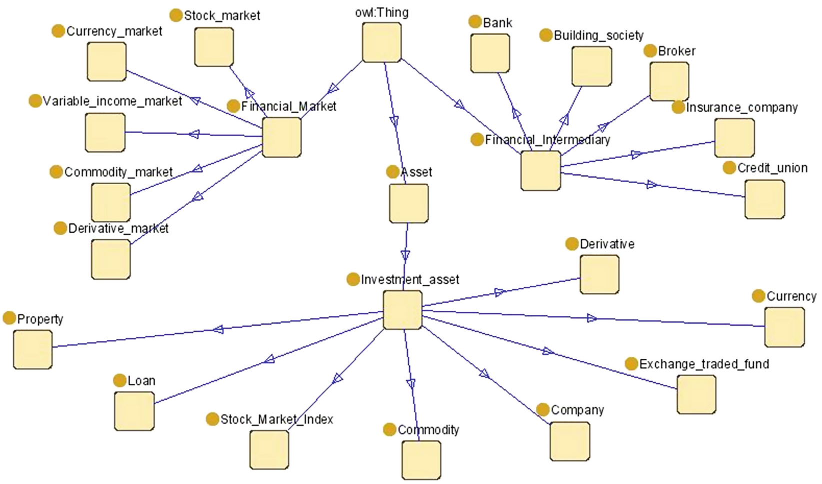


Fig. 1. An excerpt of the financial ontology

2.2 Natural Language Processing and Sentiment Analysis

Sentiment annotation can be seen as the task of assign positive, negative or neutral sentiment values to texts, sentences, and other linguistic units [20]. In this work, the values positive, negative and neutral have been assigned to general terms, which express some kind of sentiment (e.g. '*benefit*', '*positive*', '*danger*') and to financial terms (e.g. '*risk capital*', '*rising stock*', '*bankruptcy*'). Moreover, terms pertaining to the financial domain have been semantically annotated as '*risk premium*', '*capital market*' or '*Ibex35*' for example.

The open source software GATE[1] carries out sentiment and semantic annotation by means of gazetteers lists. GATE is an infrastructure for developing and deploying software components that process human language. One of the GATE's key components is gazetteer lists. A gazetteer list is a plain text file with one entry (a term, a number a name, etc.), which permits to identify these entries in the text. In this work, the lists have been developed using BWP Gazetteer.[2] This plugin provides an approximate gazetteer for GATE, based on Levenshtein's Edit Distance for strings. Its goal is to handle texts with noise and errors, in which GATE's default gazetteers may have difficulties. The implemented lists are based on the linguistic particularities of the financial domain.

Grishan and Kittredge [21] define a sublanguage as the specialized form of a natural language that is used within a particular domain or subject matter. A sublanguage is characterized by a specialized vocabulary, semantic relationships, and in many cases specialized syntax [22]. The boundaries of financial news domain are not very sharply defined [22]. For example, "*Euribor rates rise after ECB interest warnings*" or "*Portugal needs the luck of Irish*" are both headline of financial news, although the

[1] http://gate.ac.uk/.

[2] http://gate.ac.uk/gate/doc/plugins.html#bwp.

second one does not contain any financial term or a particular syntactic structure. Nevertheless, it is possible to define a wide set of financial specialized vocabulary (e.g. '*Euribor*', '*Ibex35*', '*investors*') which coexists with frequently used non-specialized terms (e.g. '*to rise*', '*unemployed*', '*construction*').

In this work, the semantic and sentiment gazetteers developed are employed to mark up all sentiment words and associated entities in our ontology. Six different kinds of gazetteers have been developed on the basis of the common characteristics and vocabulary of financial domain. The lists are used by the system in order to create three different types of annotations, that is, semantic annotations, sentiment annotations and modifier annotations. Semantic annotation refers to financial terms that are present in the financial ontology. Sentiment annotation indicates the polarity of selected terms. Modifiers annotation refers to elements that can invert or increase the polarity of the previously annotated terms. For each kind of annotation a gazetteer category has been created. Thus, semantic, sentiment and modifiers gazetteers have been developed. Each gazetteer category consists of one or more gazetteer lists, as explained below.

1. **Semantic gazetteer**
   a. Financial domain vocabulary gazetteer. This gazetteer contains the most relevant domain terms and entities. It has been directly mapped onto the ontology classes and individuals and their corresponding labels including synonyms. Examples in this category are '*Annual Percentage Rate*' (APR), '*Compound Interest*', '*Dividend*', '*Income Tax*', '*Apple*' and '*BBVA*'. This list is used for the semantic annotation and it does not contain any information related with opinions.
2. **Sentiment gazetteer**
   b. Positive sentiment gazetteer. It contains general terms that imply a positive opinion such as, for example, '*growth*', '*trust*', '*positive*' or '*rising*'.
   c. Negative sentiment gazetteer. It contains general terms that imply a negative opinion such as, for example, '*danger*', '*doubts*' or '*to cut*'.
   d. Financial positive sentiment gazetteer. It contains terms related to the financial domain that imply a positive opinion. For example, '*earning*', '*profitability*' or '*appreciating asset*'.
   e. Financial negative sentiment gazetteer. It contains terms related to financial domain that imply a negative opinion. For example, '*depreciation*', '*Insufficient Funds*' or '*creditor*'.
3. **Modifier gazetteer**
   f. Intensifier gazetteer. It contains terms that are used to change the degree to which a term is positive or negative such as, for example, '*very*', '*most*' or '*extremely*'.
   g. Negation gazetteer. It contains negation expressions such as, for example, '*no*', '*never*' or '*deny*'.
   h. Temporal sentiment gazetteers. They contain temporal expressions that imply a modification in the whole news. These expressions appear in conjunction with positive or negative linguistic expressions modifying their meaning. They usually increase or decrease negative or positive sentiment. There are two temporal gazetteers, one with long-term expressions and the other with short-term expressions. "*Last year*", "*trimester*" or "*several weeks*" are examples of the first

type, while “*this morning*”, “*today*” “*this week*” are examples of the second type. The following sentences show an example of the modification capacity of temporal terms in the financial domain:
(1) Apple shares have risen around 17 % in the last month.
(2) Apple shares have fallen 4.5 % this morning.
Here, “*last month*” and “*this morning*” can relativize the weight of the global meaning. In general, long-term positive or negative opinions are more reliable than short-term opinions. That is, if the user searches for the general status of Apple shares and the system retrieves these two entries, then the general opinion should be positive.

3 Platform Architecture

The architecture of the platform is shown in Fig. 2. The architecture is composed of four main components: the financial news extraction module, the semantic annotation module, the opinion-mining module and the search engine. Next, these components are described in detail.

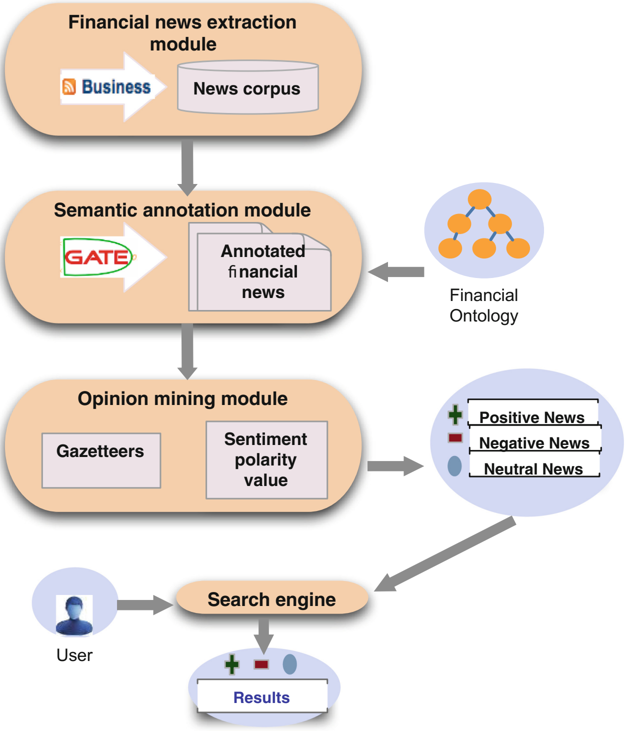


Fig. 2. Architecture of the system.

3.1 Financial News Extraction Module

This module manages the list of RSS feeds. RSS is a family of Web feed formats used for syndicating content from blogs or Web pages and is commonly used by newspapers. RSS is an XML file that summarizes information items and links to the information sources [23]. Once the resources have been selected, this module generates a set of abstracts, which will be used as input for the system. An example list of financial news-related RSS feeds is shown in Table 1.

Table 1. Example of RSS feeds

http://money.cnn.com/services/rss/
http://europe.wsj.com/xml/rss/3_7481.xml
http://www.ft.com/rss/home/europe
http://www.ibtimes.com/rss/

For each RSS source the last news are obtained and stored in a database. The information that is retrieved from each news is the date of publication, the information source, the url and the abstract. Abstracts constitute the corpus from which the system extracts the information. We only consider the abstract and the headline because they usually condense the polarity of news. Indeed, the analysis of the whole text can induce to error, since the sentiment polarity of an entire document is not necessarily the sum of its parts.

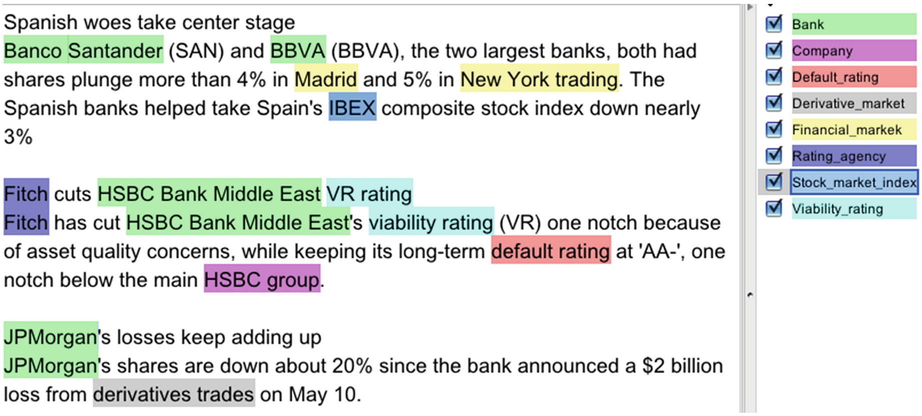
3.2 Semantic Annotation Module

This module identifies the most important linguistic expressions in the financial domain using the previously described semantic gazetteer. For each linguistic expression, the system tries to determine whether the expression under question is an individual of any of the classes of the domain ontology. Next, the system retrieves all the annotated knowledge that is situated next to the current linguistic expression in the text, and tries to create fully-filled annotations with this knowledge.

Each class in the ontology is defined by means of a set of relations and datatype properties. Then, when an annotated term is mapped onto an ontological individual, its datatype and relationships constitute the potential information which is possible to obtain for that individual. For example, a company has associate relationships such as '*Moody'sRate*', '*tradeMarket*' or '*isLegalRepresentativeFor*'. In Fig. 3, an example of the annotation process of financial news using the semantic gazetteer in GATE is depicted.

3.3 Opinion Mining Module

The main objective of this module is to classify the set of news obtained in the previous module according to their polarity: positive, negative or neutral. For any retrieved news which has been annotated, the sentiment orientation or sentiment polarity value is computed. For this, the module makes use of the previously described gazetteer lists.



Spanish woes take center stage
Banco Santander (SAN) and BBVA (BBVA), the two largest banks, both had shares plunge more than 4% in Madrid and 5% in New York trading. The Spanish banks helped take Spain's IBEX composite stock index down nearly 3%

Fitch cuts HSBC Bank Middle East VR rating
Fitch has cut HSBC Bank Middle East's viability rating (VR) one notch because of asset quality concerns, while keeping its long-term default rating at 'AA-', one notch below the main HSBC group.

JPMorgan's losses keep adding up
JPMorgan's shares are down about 20% since the bank announced a $2 billion loss from derivatives trades on May 10.

☑ Bank
☑ Company
☑ Default_rating
☑ Derivative_market
☑ Financial_markek
☑ Rating_agency
☑ Stock_market_index
☑ Viability_rating

Fig. 3. Example of knowledge entities identified in financial news.

The sentiment polarity (SP) value for each news item is calculated by summing the polarity values of all annotated terms in the news. In this process, the system must consider both the terms polarity included in the positive and negative gazetteers and the contextual valence shifters included in the negation and intensifier gazetteers.

For any annotated term (***at***) in a sentence s ∈ S, its SP value (**SP(at)**) is computed as follows:

1. If *at* ∈ GeneralPositive^k, SP(at) = Positive1
2. If *at* ∈ DomainPositive^k, SP(at) = Positive2
3. If *at* ∈ GeneralNegative^k, SP(at) = Negative1
4. If *at* ∈ DomainNegative^k, SP(at) = Negative2
5. If within the relevant cotext of *at*, there is a term *at*′ ∈ Negation, SP(at) = −SP(at)
6. If within the relevant cotext of *at*, there is a term *at*′ ∈ Intensifier, SP(at) = 2 × SP(at)
7. When within the relevant cotext of *at*, there is a term *at*′ ∈ Temporal, if …

7.1. *at*′ ∈ LongTerm, SP(at) = 2 × SP(at)
7.2. *at*′ ∈ ShortTerm + Negative(SP), SP(at) = 2 × SP(at)
7.3. *at*′ ∈ ShortTerm + Positive(SP), SP(at) = 1 × SP(at)

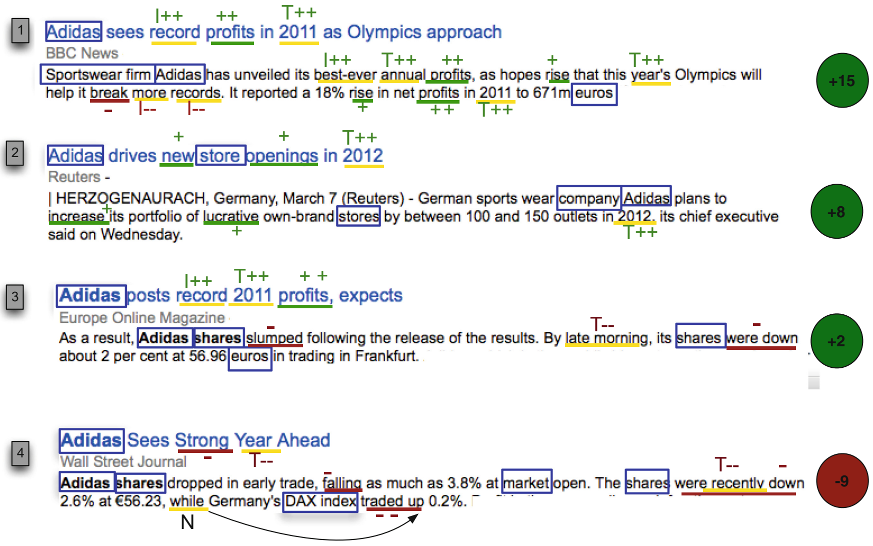
Then the polarity of each news item is represented as the sum of all SP(at) present in such news item (n):

$$f^k SP(n)^k = \sum_{at \in n} SP(at) \tag{1}$$

In the above algorithm, the term '*cotext*' refers to the linguistic set that surrounds an annotated term within the limit of a sentence, i.e. the rest of annotated terms present before and after it and pertaining to the same sentence. '*Positive1*' and '*Positive2*' refer to the degree of positivity of an annotated term, while '*Negative1*' and '*Negative2*' refer to the degree of negativity of an annotated term.

When a long-term temporal expression is found, its value is calculated taking into account the *at* pertaining to its cotext. If a positive *at* is found, then its value is 2. On the contrary, if a negative *at* is found its value is −2. Sort- term temporal expressions are calculated in the same way for negative value, i.e. adding −2. However, for positive value the system only adds 1positive. This is because we consider that financial short-term positive values change too frequently to consider them at the same level as long-term values.

Next, if the semantic polarity value of a news is less than 0, the news is labelled as negative. In contrast, if the value is higher than 0, the news is labelled as positive. Finally, if the sum of all values is 0 the news is labelled as neutral. An example of how the algorithm works is shown in Fig. 4.



1 Adidas sees record profits in 2011 as Olympics approach

BBC News

Sportswear firm Adidas has unveiled its best-ever annual profits, as hopes rise that this year's Olympics will help it break more records. It reported a 18% rise in net profits in 2011 to 671m euros

+15

2 Adidas drives new store openings in 2012

Reuters -

| HERZOGENAURACH, Germany, March 7 (Reuters) - German sports wear company Adidas plans to increase its portfolio of lucrative own-brand stores by between 100 and 150 outlets in 2012, its chief executive said on Wednesday.

+8

3 Adidas posts record 2011 profits, expects

Europe Online Magazine

As a result, Adidas shares slumped following the release of the results. By late morning, its shares were down about 2 per cent at 56.96 euros in trading in Frankfurt.

+2

4 Adidas Sees Strong Year Ahead

Wall Street Journal

Adidas shares dropped in early trade, falling as much as 3.8% at market open. The shares were recently down 2.6% at €56.23, while Germany's DAX index traded up 0.2%.

-9

Fig. 4. Semantic polarity annotation example

Let us suppose that a user searches for the company ‘*Adidas*’. In the example depicted in Fig. 4, four different news items are retrieved. In the figure, semantic annotations are the elements surrounded by a rectangle, which have been mapped onto ontology instances. GeneralPositive are indicated with one ‘+’ sign and DomainPositive with two, ‘++’. On the other hand, GeneralNegative are indicated with one ‘–’ sign and DomainNegative with two, ‘–’. The modifiers Negative, Temporal and Intensifier are indicated with ‘N’, ‘T’, ‘I’ respectively, together with the corresponding positive or negative symbol.

The outcome of the process is three positive and one negative news items. In this particular example, the presence of long-term temporal expressions, such as ‘*2012*’ or ‘*year*’, in conjunction with positive annotated terms, gives to the news a high positive value. The user can organize the final results in accordance with their degree of positivity and negativity.

3.4 Semantic Search Engine

In OWL-based ontologies, ‘*rdfs:label*’ is an instance of ‘*rdf:property*’ that may be used to provide a human readable version of a resource name. In this work, all the resources in the ontology have been annotated with the ‘*rdfs:label*’ descriptor. By considering that, the main objective of this module is to identify the financial news items that are related to the query issued by a user. Besides, this module is responsible for classifying and sorting the results in accordance with the sentiment classification that was described in the previous section.

The system is constantly crawling news information from RSS feeds and creating semantic annotations for the news pages. If no annotations are created for a news item, then such news item is not stored in the database. On the other hand, the news items that have been successfully annotated are processed to obtain their sentiment classification, which is also stored in the database. For example, let us suppose that the ontology contains the taxonomy presented in Fig. 3. There are two kinds of companies, namely, “Energy company” and “ICT company”. Each of these classes contains a set of individuals such as “Microsoft” and “GE energy”, respectively. If the user is searching for news about “Microsoft”, the system will certainly return all the news annotated with the individual Microsoft. Moreover, news related to other ICT companies could be relevant to the user, so the system also shows other news about companies such as Google, Apple and Nokia. If the user queries the system for “Energy companies”, then the result will include all the news that contains the concept “Energy company” and therefore the news related to the “GE Energy”, “Texaco” and “Shell” companies will be retrieved. Furthermore, if the query is such a general word as “Company”, the user is given the possibility of filtering the results according to the subclasses of “Company”, namely, “Energy company” and “ICT company”.

4 Evaluation

In this section, the experimental results obtained by the proposed method in the financial news domain are presented. The corpus of the experiment contains 57.210 words and comprises 900 abstracts of financial news (512 negative and 388 positive). This corpus has been extracted from the RSS feeds shown in Table 1 and each news item has been manually labelled, either as a positive news or a negative one, by two different annotators. This constitutes the baseline for the evaluation, which works as follows: if the result displayed by the system fits in with the manually annotated news, the result is considered correct, otherwise, incorrect. In the sentiment analysis field, it is agreed that human-based annotations are around 70–80 % precise (i.e. 2 different humans can disagree in 20–30 % of cases). However, for the purposes of this experiment, the news items that have been source of disagreement between annotators have been removed.

In the experiment, a total of five queries are issued to the system to find information in the financial domain. The results of the experiment are shown in Table 2. It is possible to observe that the sentimental analysis accuracy results are very promising, with an aggregate accuracy mean of 87 %. These results take into account the system’s final decision (positive or negative) and not the process that the system carries out to produce such decision.

Table 2. Hits results in information retrieval.

Query		Baseline	Our approach	Accuracy
1	Pos	33	28	84.85 %
	Neg	11	9	81.82 %
	Total	44	37	84.09 %
2	Pos	13	13	100 %
	Neg	36	34	94.44 %
	Total	49	47	95.92 %
3	Pos	15	14	93.33 %
	Neg	29	24	82.76 %
	Total	44	38	86.36 %
4	Pos	25	21	84 %
	Neg	97	86	88.66 %
	Total	122	107	87.70 %
5	Pos	66	55	83.33 %
	Neg	14	12	85.71 %
	Total	80	67	83.75 %
Total		678	592	87.32 %

5 Related Works and Discussion

In the literature, a number of methods for the automatic sentiment analysis from financial news streams have been described. The proposal of [7] uses theories of lexical cohesion in order to create a computable metric to identify the sentiment polarity of financial news texts. This metric is readapted in [6] to Chinese and Arabic financial news. The analysis of financial news is a particularly relevant topic in the prediction of the behavior of stock markets. For example, in [8] the authors use some simple computational linguistic techniques, such as bag of words or named entities, together with support vector machine and machine learning techniques to assist in making stock market predictions. In fact, in real life, stock market analysts' predictions are usually based on the opinions expressed in the news. Our approach does not focus solely on the stock market. It covers a wider range of financial news.

Semantic technologies have been around for a while, offering a wide range of benefits in the knowledge management field. They have revolutionized the way that systems integrate and share data, enabling computational agents to reason about information and infer new knowledge [10]. The accuracy results of opinion mining and sentiment polarity analysis can be improved with the addition of semantic techniques, as shown in [24]. In that work, some semantic lexicons are created in order to identify sentiment words in blog and news corpora. Then, a polarity value is attached to each word in the lexicon and such polarity is revised when a modifier appears in the text. The main problem of corpora-based methods is the cost of the annotation process. A further problem, namely, the obsolescence of resources, is present on constantly changing domains such as the financial domain. This challenge is partially overcome by using bag of words or gazetteer lists which are easy to update if required.

The FIRST project[3] provides an information extraction, information integration and decision making infrastructure for information management in the financial domain. The decision making infrastructure includes a module responsible for the sentiment annotation from financial news and blog posts. Its main aim is to classify the polarity of sentiment with respect to a sentiment object of interest [25]. These sentiment objects are classified by means of an ontology-guided and rule-based information extraction approach. Even though the ontology contains the financial-domain related relevant objects, the classification process is carried out entirely using JAPE rules. This work is the closest to the one presented here, but their approach does not leverage the reasoning capabilities of the ontology. In fact, none of the previous works considers the ontological capabilities in the task of sentiment annotation. Combining the usage of semantic gazetteers in a textual level with the ontology in a conceptual level makes our system adaptable to other languages with minimum cost.

6 Conclusions

The boom of the Social Web has had a tremendous impact on a number of different research topics. In particular, the possibility of extracting various kinds of added-value, informational elements from users' opinions has attracted researchers from the information retrieval and computational linguistics fields. This process is called opinion mining and is currently one of the most challenging research topics. More specifically, opinion mining is concerned with analyzing the opinions of a particular matter expressed by users in the form of natural language that appear in a series of texts. The opinion mining process makes it possible to figure out whether a user's opinion is positive, negative or neutral, and how strong it is [26]. Similarly, sentiment analysis deals with the computational treatment of opinions expressed in written texts.

The addition of the already mature semantic technologies to this field has proven to increase the results accuracy. In this work, a semantically-enhanced methodology for the annotation of sentiment polarity in financial news-related natural language texts is presented. The proposed methodology is based on an algorithm that combines several gazetteer lists and leverages an existing financial ontology. The sentiment algorithm assigns different degrees of positivity or negativity to relevant annotated terms and calculates what the polarity of the news is. The financial-related news items in our experiment are obtained from RSS feeds and then automatically annotated with positive or negative markers. The outcome of the process is a set of financial news texts organized by their degree of positivity and negativity.

This approach contributes to the research on financial sentiment annotation, and the development of decision support systems (1) by proposing a novel approach for financial sentiment determination in news which combines ontological resources with natural language processing resources, (2) by describing an algorithm for assigning different degrees of positivity or negativity to classify results into various categories

[3] http://project-first.eu/.

identified by the classifier, and (3) by proposing a set of resources, i.e. gazetteer lists and an ontology, for sentiment annotation.

Acknowledgements. This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Commission (FEDER / ERDF) through project SeCloud (TIN2010-18650).

References

1. Esuli, A., Sebastiani, F.: Sentiwordnet: a publicly available lexical resource for opinion mining. In: Proceedings of 5th Conference on Language Resources and Evaluation, LREC 2006 (2006)
2. Chen, H., Zimbra, D.: AI and opinion mining. Intell. Syst. IEEE **25**(3), 74–80 (2010)
3. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 339–346 (2005)
4. Ding, X., Liu, B.: The utility of linguistic rules in opinion mining. In Proceedings of 30th Annual International ACM Special Interest Group on Information Retrieval Conference (SIGIR 2007), Amsterdam, The Netherlands (2007)
5. Balahur, A., Montoyo, A.: Determining the semantic orientation of opinions of products-a comparative analysis. Procesamiento del lenguaje natural **41**, 201–208 (2008)
6. Ahmad, K., Cheng, D., Almas, Y.: Multi-lingual sentiment analysis of financial news streams. In: Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages, Linguistic Society of America, Linguistic Institute, Stanford University, pp. 1–12 (2007)
7. Devitt, A., Ahmad, K.: Sentiment analysis in financial news: a cohesionbased approach. In Proceedings of the Association for Computational Linguistics (ACL), pp. 984–991 (2007)
8. Schumaker, R.P., Chen, H.: Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Trans. Inf. Syst. **27**, 1–19 (2009)
9. Berners-Lee, T., Hendler, J.: Publishing on the semantic web. Nature **410**, 1023–1025 (2001)
10. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. Data Knowl. Eng. **25**(1–2), 161–197 (1998)
11. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
12. Valencia-García, R., Fernández-Breis, J.T., Ruiz-Martínez, J.M., García-Sánchez, F., Martínez-Béjar, R.: A knowledge acquisition methodology to ontology construction for information retrieval from medical documents. Expert Syst. Knowl. Eng. J. **25**(3), 314–334 (2008)
13. Lupiani-Ruiz, E., García-Manotas, I., Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T., Camón-Herrero, J.B.: Financial news semantic search engine. Expert Syst. Appl. **38**(12), 15565–15572 (2011)
14. García-Sánchez, F., Valencia-García, R., Martínez-Béjar, R., Fernández-Breis, J.T.: An ontology, intelligent agent-based framework for the provision of semantic web services. Expert Syst. Appl. Part 2 **36**(2), 3167–3187 (2009)
15. Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J.T.: OWLPath: an OWL ontology-guided query editor: IEEE Transactions on Systems. Man Cybern. Part A **41**(1), 121–136 (2011)

16. Partridge, C.: The role of ontology in integrating semantically heterogeneous databases. Report No.: LADSEB-CNR Technical Report 05/2002 (2002)
17. Fox, M.S., Gruninger, M.: Enterprise modeling. AI Mag. **19**(3), 109 (1998)
18. Corcho, O., Losada, S., Martínez Montes, M., Bas, J.L., Bellido, S.: Financial Ontology. DIP deliverable D10.3 (2004)
19. Bonsón, E., Cortijo, V., Escobar, T.: Towards the global adoption of XBRL using international financial reporting standards (IFRS). Int. J. Account. Inf. Syst. **10**(1), 46–60 (2009)
20. Andreevskaia, A., Bergler, S.: When specialists and generalists work together: overcoming domain dependence in sentiment tagging. In: Proceedings of ACL 2008: HLT, pp. 290–298 (2008)
21. Grishman, R., Kittredge, R.: Analyzing Language in Restricted Domains: Sublanguage Description and Processing. Lawrence Erlbaum, Hillsdale (1986)
22. Grishman, R.: Adaptive information extraction and sublanguage analysis. In: Kushmeric, N. (ed.) Proceedings of Workshop on Adaptive Text Extraction and Mining at Seventeenth International Joint Conference on Artificial Intelligence, Seattle, WA (2001). http://nlp.cs.nyu.edu/pubs/papers/grishman-ijcai01.pdf
23. Murugesan, S.: Understanding web 2.0. IT Prof. **9**(4), 34–410 (2007)
24. Godbole, N., Srinivasaiah, M., Skiena, S.: Largescale sentiment analysis for news and blogs: In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM) (2007)
25. Klein, A., Häusser, T., Altuntas, O., Grcar, M.: Large scale information extraction and integration infrastructure for supporting financial decision making. Deliverable: D4.1 First semantic information extraction prototype (2012). http://project-first.eu/content/d41-first-semantic-information-extraction-prototype
26. Zhao, L., Li, C.: Ontology Based Opinion Mining for Movie Reviews. In: Karagiannis, D., Jin, Z. (eds.) KSEM 2009. LNCS, vol. 5914, pp. 204–214. Springer, Heidelberg (2009)

Interacting with Statistical Linked Data via OLAP Operations

Benedikt Kämpgen[1](✉), Seán O'Riain[2], and Andreas Harth[1]

[1] Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany
{benedikt.kaempgen,harth}@kit.edu

[2] Digital Enterprise Research Institute, National University of Ireland,
Galway, Ireland
sean.oriain@deri.org

Abstract. Online Analytical Processing (OLAP) promises an interface to analyse Linked Data containing statistics going beyond other interaction paradigms such as follow-your-nose browsers, faceted-search interfaces and query builders. Transforming statistical Linked Data into a star schema to populate a relational database and applying a common OLAP engine do not allow to optimise OLAP queries on RDF or to directly propagate changes of Linked Data sources to clients. Therefore, as a new way to interact with statistics published as Linked Data, we investigate the problem of executing OLAP queries via SPARQL on an RDF store. First, we define projection, slice, dice and roll-up operations on single data cubes published as Linked Data reusing the RDF Data Cube vocabulary and show how a nested set of operations lead to an OLAP query. Second, we show how to transform an OLAP query to a SPARQL query which generates all required tuples from the data cube. In a small experiment, we show the applicability of our OLAP-to-SPARQL mapping in answering a business question in the financial domain.

Keywords: OLAP · Query · Linked data · Statistics · XBRL · SPARQL

1 Introduction

Linked Data provides easy access to large amounts of interesting statistics from many organizations for information integration and decision support, including financial information from institutions such as the UK government[1] and the U.S. Securities and Exchange Commission.[2] However, interaction paradigms for Linked Data such as follow-your-nose browsers, faceted-search interfaces, and query builders [12,14] do not allow users to analyse large amounts of numerical data in an exploratory fashion of "overview first, zoom and filter, then details-on-demand" [22]. Online Analytical Processing (OLAP) operations on data cubes for viewing statistics from different angles and granularities, filtering for specific features, and comparing aggregated measures fulfil this information seeking

[1] http://data.gov.uk/resources/coins.

[2] http://edgarwrap.ontologycentral.com/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 87–101, 2015.
DOI: 10.1007/978-3-662-46641-4_7

mantra and provide interfaces for decision-support from statistics [2,5,19]. However, OLAP on statistical Linked Data imposes two main challenges:

- OLAP requires a model of data cubes, dimensions, and measures. Automatically creating such a multidimensional schema from generic ontologies such as described by Linked Data is difficult, and only semi-automatic methods have proved applicable [18]. Although the RDF Data Cube vocabulary (QB)[3] is a Linked Data vocabulary to model statistics in RDF and several publishers have already used the vocabulary for statistical datasets, there is yet no standard to publish multidimensional models as Linked Data.[4]
- OLAP queries are complex and require specialised data models, e.g., star schemas in relational databases, to be executed efficiently [11]. The typical architecture of an OLAP system consists of an ETL pipeline that extracts, transforms and loads data from the data sources into a data warehouse, e.g., a relational or multidimensional database. OLAP clients such as JPivot allow users to built OLAP queries and display multidimensional results in pivot tables. An OLAP engine, e.g., Mondrian, transforms OLAP queries into queries to the data warehouse, and deploys mechanisms for fast data cube computation and selection, under the additional complexity that data in the data warehouse as well as the typical query workload may change dynamically [10,17].

As a first effort to overcome these challenges, in previous work [13], we have presented a proof-of-concept to automatically transform statistical Linked Data that is reusing the RDF Data Cube vocabulary (QB) into a star schema and to load the data into a relational database as a backend for a common OLAP engine. OLAP queries are executed not on the RDF directly but by a traditional OLAP engine after automatically populating a data warehouse which results in drawbacks: (1) although the relational star schema we adopted is a quasi-standard logical model for data warehouses, our approach requires an OLAP engine to execute OLAP queries and does not allow to optimise OLAP queries to RDF; (2) if statistical Linked Data is updated, e.g., if a single new statistic is added, the entire ETL process has to be repeated, to have the changes propagated; (3) integration of additional data sources for more expressive queries is difficult, since multidimensional models have a fixed schema.

Therefore, in this work, we approach the following problem as illustrated in Fig. 1: At the backend, given statistical Linked Data reusing QB, crawled into a triple store, and accessible via a SPARQL endpoint. SPARQL is a standard query language for issuing queries to Linked Data. On the frontend, given a common OLAP client capable of running OLAP operations on data cubes, and capable of communicating those OLAP operations as an OLAP query via a common OLAP query language such as MDX. The Multidimensional Expression Language (MDX), as far as we know, is the most widely used OLAP query language, adopted by OLAP engines such as Microsoft SQL Server, the Open Java API for OLAP, XML for

[3] http://www.w3.org/TR/2012/WD-vocab-data-cube-20120405/.

[4] http://wiki.planet-data.eu/web/Datasets.

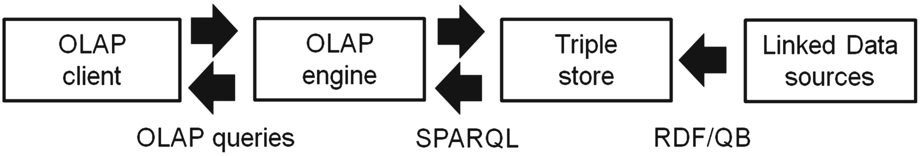


Fig. 1. Data flow for OLAP queries on statistical Linked Data in a triple store

Analysis (XMLA), and Mondrian. The question is how an OLAP engine can map MDX queries to SPARQL queries.

This paper presents a new way to interact with statistical Linked Data:

- We define common OLAP operations on data cubes published as Linked Data reusing QB and show how a nested set of OLAP operations lead to an OLAP query.
- We show how to transform an OLAP query to a SPARQL query which generates all required tuples from the data cube.

In the remainder of the paper, we first present a motivational scenario from the financial domain in Sect. 2. As a prerequisite for our contribution, in Sect. 3, we formally define a multidimensional model of data cubes based on QB. Then, in Sect. 4, we introduce OLAP operations on data cubes and present a direct mapping of OLAP to SPARQL queries. We apply this mapping in a small experiment in Sect. 5 and discuss some lessons learned in Sect. 6. In Sect. 7, we describe related work, after which, in Sect. 8, we conclude and describe future research.

2 Scenario: Analysing Financial Linked Data

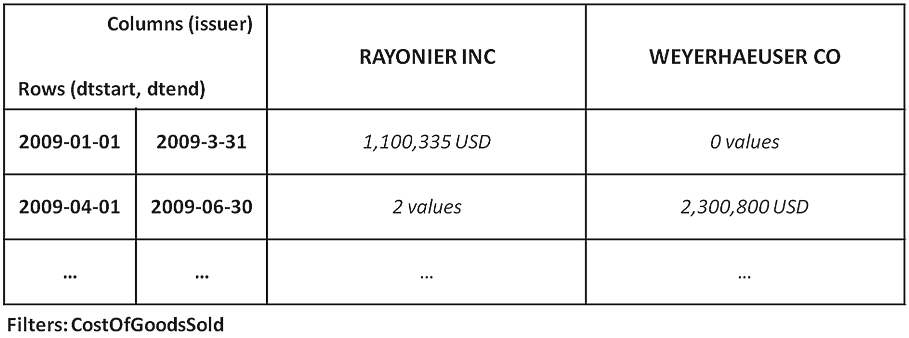
In this section we describe a scenario of analysing Linked Data containing financial information. XBRL is an XML data format to publish financial information.[5] The U.S. Securities and Exchange Commission (SEC) requires companies to provide financial statement information in the XBRL format. The Edgar Linked Data Wrapper[6] provides access to XBRL filings from the SEC as Linked Data reusing QB. Those filings disclose balance sheets of a large number of US organizations, for instance that *RAYONIER INC* had a *sales revenue net* of 377,515,000 USD from 2010-07-01 to 2010-09-30.[7]

Using LDSpider, we crawled Linked Data from the Edgar wrapper and stored a data cube *SecCubeGrossProfitMargin* into an Open Virtuoso triple store. The data cube contains single disclosures from financial companies such as *RAYONIER INC*. Each disclosure either discloses cost of goods sold (CostOfGoodsSold) or sales revenue net (Sales) as measures. The two measures have unit USD and an

[5] http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm.

[6] http://edgarwrap.ontologycentral.com/.

[7] http://edgarwrap.ontologycentral.com/archive/52827/0001193125-10-238973#ds.



Columns (issuer) Rows (dtstart, dtend)		RAYONIER INC	WEYERHAEUSER CO
2009-01-01	2009-3-31	*1,100,335 USD*	*0 values*
2009-04-01	2009-06-30	*2 values*	*2,300,800 USD*
...	...	...	...

Filters: CostOfGoodsSold

Fig. 2. Pivot table to be filled in our scenario

aggregation function that returns the number of disclosures, or – if only one – the actual number. Any disclosure is fully dependent on the following dimensions: the disclosing company (Issuer), the date a disclosure started (Dtstart) and ended (Dtend) to be valid, and additional meta information (Segment).

In our scenario, a business analyst wants to compare the number of disclosures of cost of goods sold for two companies. He requests a pivot table with issuers *RAYONIER INC* and *WEYERHAEUSER CO* on the columns, and the possible periods for which disclosures are valid on the rows, and in the cells showing the number of disclosed cost of goods sold, or – if only one – the actual number. Figure 2 shows the needed pivot table.

3 A Multidimensional Model Based on QB

In this section, as a precondition for OLAP queries on Linked Data, we formally define the notion of data cubes in terms of QB. The definition is based on a common multidimensional model, e.g., used by Gómez et al. [9], Pedersen et al. [20] and the Open Java API for OLAP.[8] Also, we base our definition on an RDF representation reusing QB, as well as other Linked Data vocabularies for publishing statistics, e.g., SKOS[9] and skosclass.[10]

Definition 1. (Linked Data store with RDF terms and triples). *The set of RDF terms in a triple store consists of the set of IRIs $\mathcal{I}$, the set of blank nodes $\mathcal{B}$ and the set of literals $\mathcal{L}$. A triple $(s, p, o) \in \mathcal{T} = (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$ is called an RDF triple, where s is the subject, p is the predicate and o is the object.*

Given a triple store with statistical Linked Data, we use basic SPARQL triple patterns on the store to define elements of a multidimensional model. Given a multidimensional element x, $id(x) \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$ returns its RDF identifier:

[8] http://www.olap4j.org/.

[9] http://www.w3.org/2004/02/skos/.

[10] http://www.w3.org/2011/gld/wiki/ISO_Extensions_to_SKOS.

Member defines the set of members as $Member = \{?x \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L}) | ?x \text{ a } \textit{skos:Concept} \vee x \in \mathcal{L}\}$. Let $\mathcal{V} = 2^{Member}$, $V \in \mathcal{V}$, $ROLLUPMEMBER \subseteq Member \times Member$, $rollupmember(V) = \{(v_1, v_2) \in V \times V | (id(v_1) \textit{ skos:broader } id(v_2) \vee id(v_2) \textit{ skos:narrower } id(v_1))\}$. Note, in case of literal members $rollupmember(V)$ is empty.

Level defines the set of levels as $Level = \{(?x, V, rollupmember(V)) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{V} \times ROLLUPMEMBER | (?x \text{ a } \textit{skosclass:ClassificationLevel} \wedge \forall v \in V(id(v) \textit{ skos:member } ?x)\}$. Let $\mathcal{L} = 2^{Level}$, $L \in \mathcal{L}$, $ROLLUPLEVEL \subseteq Level \times Level$, $rolluplevel(L) = \{(l_1, l_2) \in L \times L | (id(l_1) \textit{ skosclass:depth } x) \wedge (id(l_2) \textit{ skosclass:depth } y) \wedge x \leq y))\}$

Hierarchy defines the set of hierarchies as $Hierarchy = \{(?x, L, rolluplevel(L)) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{L} \times ROLLUPLEVEL | (?x \text{ a } \textit{skos:ConceptScheme}\ ) \wedge \forall l \in L(id(l) \textit{ skos:inScheme } ?x)\}$. Let $\mathcal{H} = 2^{Hierarchy}$.

Dimension defines the set of dimensions as $Dimension = \{(?x, H) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{H} | (?x \text{ a } \textit{qb:DimensionProperty}\ ) \wedge \forall h \in H(?x \textit{ qb:codeList } id(h))\}$. Let $\mathcal{D} = 2^{Dimension}$.

Measure defines the set of measures as $Measure = \{(?x, aggr) \in (\mathcal{I} \cup \mathcal{B}) \times \{UDF\} | (?x \text{ a } \textit{qb:MeasureProperty}\ )\}$ with $UDF : 2^{\mathcal{L}} \rightarrow \mathcal{L}$ a default aggregation function since QB so far does not provide a standard way to represent typical aggregation functions such as *SUM*, *AVG* and *COUNT*. If the input set of literals only contains one literal, *UDF* returns the literal itself, otherwise *UDF* returns a literal describing the number of values. *UDF* is an algebraic aggregation function in that it can be computed by distributive functions *COUNT* and *SUM* [10]. Conceptually, measures are treated as members of a dimension-hierarchy-level combination labelled "Measures". Let $\mathcal{M} = 2^{Measure}$.

DataCubeSchema defines the set of data cube schemas as $\{(?x, D, M) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{D} \times \mathcal{M} | (?x \text{ a } \textit{qb:DataStructureDefinition} \wedge \forall d \in D(?x \textit{ qb:component } ?comp \wedge ?comp \textit{ qb:dimension } id(d)) \wedge \forall m \in M(?x \textit{ qb:component } ?comp \wedge ?comp \textit{ qb:measure } id(m)))\}$.

Fact defines the set of possible statistical facts as $Fact = \{(?x, C, E) \in (\mathcal{I} \cup \mathcal{B}) \times 2^{Dimension \times Member} \times 2^{Measure \times Literal} | (?x \text{ a } \textit{qb:Observation}\ ) \wedge \forall (d_1, m_1), (d_2, m_2) \in C, id(m_1) \neq id(m_2)(?xid(d_1)id(m_1) \wedge id(d_1) \neq id(d_2)) \wedge \forall (m_1, v_1), (m_2, v_2) \in E, id(v_1) \neq id(v_2)(?xid(m_1)v_1 \wedge id(m_1) \neq id(m_2))$. Note that each fact is restricted to have for each dimension and measure at maximum one member and value, respectively. Let $\mathcal{F} = 2^{Fact}$.

DataCube defines the set of data cubes as $DataCube = \{(cs, F) \in DataCubeSchema \times \mathcal{F} | cs = (?x, D, M) \wedge \forall (?obs_1, C_1, E_1), (?obs_2, C_2, E_2) \in \mathcal{F}, ?obs_1 \neq ?obs_2(?obs_1 \textit{ qb:dataSet } ?ds \wedge ?ds \textit{ qb:structure } ?x) \wedge C_1 \neq C_2 \wedge \{d : \exists (d, m) \in C_1\} = D \wedge \forall (d, m) \in C_1(id(m) \textit{ skos:member } ?l.?l \textit{ skos:inScheme } ?h.id(d) \textit{ qb:codeList } ?h \vee ?m \textit{ skos:notation } id(m).?m \textit{ skos:member } ?l.?l \textit{ skos:inScheme } ?h.id(d) \textit{ qb:codeList } ?h)\}$ Note, the measure value is fully dependent on the dimension members, thus, any two facts need to have a different member on one of their dimensions. Also, any fact needs to have a member for each dimension mentioned in the schema. As a last requirement, each member needs to be contained in a level of a hierarchy of the dimension. A data cube may be sparse and not containing facts for each possible combination of dimension members. If the member is a literal, there will be a concept representing this member and linking to its literal value via *skos:notation*. Similar to data in a fact table of a star schema, we assume that all facts of the data cube are on the lowest granularity level, since then, all measures on any higher aggregation level can be computed from these facts.

We distinguish *metadata queries* and *OLAP queries* on data cubes. Whereas metadata queries return multidimensional objects such as the cube schema, the dimensions, and the measures, OLAP queries on a certain data cube return tuples. The number of tuples that possibly can be queried from a data cube is exponentially growing with the number of dimensions, as the following definition shows:

Definition 2. (Data Cube Tuples). *Adopting the concept of Gray et al. [10], we can compute all tuples* $(c_1, \ldots, c_{|D|}, t_1, \ldots, t_j)$*,* $j \leq |M|$ *with* $c_i \in Member \cup ALL$ *and* $t_i \in T$ *with* T *a numeric domain including the special* null *value in case of cube sparsity from a data cube represented in QB as follows: We extract a relational table containing measures for each possible combination of dimension members from the data cube* (cs, F) *with data cube schema* $cs = (?x, D, M)$*, dimensions* $D = \{D_1, \ldots, D_N\}$ *and facts on the lowest granularity level* $(?obs, C, E) \in F$*. We compute* 2^N *aggregations of each measure value by the measure's aggregation function over all possible select lists of dimensions* $sl \in 2^D$ *with a standard* GROUP BY*. Then, we merge all aggregation results with a standard* UNION*, substituting the special* ALL *value for the aggregation columns. If the number of possible members of a dimension is* $card(D_i)$*, then the number of resulting facts in the materialised data cube is* $\prod(card(D_i) + 1)$*. The extra value for each dimension is* ALL*.*

Subqueries and aggregation functions in SPARQL 1.1 make easily possible to apply the relational concept of Gray et al. [10] to a data cube represented as Linked Data reusing QB. However, such a SPARQL query would have an exponential number of subqueries and would take a long time to execute. Also, the query would fully materialise the data cube, i.e. compute all possible tuples, although OLAP queries may require only a small subset. Therefore, in the next section, we show how to evaluate OLAP queries directly without subqueries and without fully materialising the cube.

4 Mapping OLAP Operations to SPARQL on QB

In this section we show how to issue OLAP queries on a multidimensional model. We define common OLAP operations on single data cubes [19–21]. A nested set of OLAP operations lead to an OLAP query. We describe how to evaluate such an OLAP query using SPARQL on QB. Figure 3 illustrates the effect of common OLAP operations, with inputs and outputs.

Note, this paper focuses on direct querying of single data cubes, the integration of several data cubes through *Drill-Across* or set-oriented operations such as

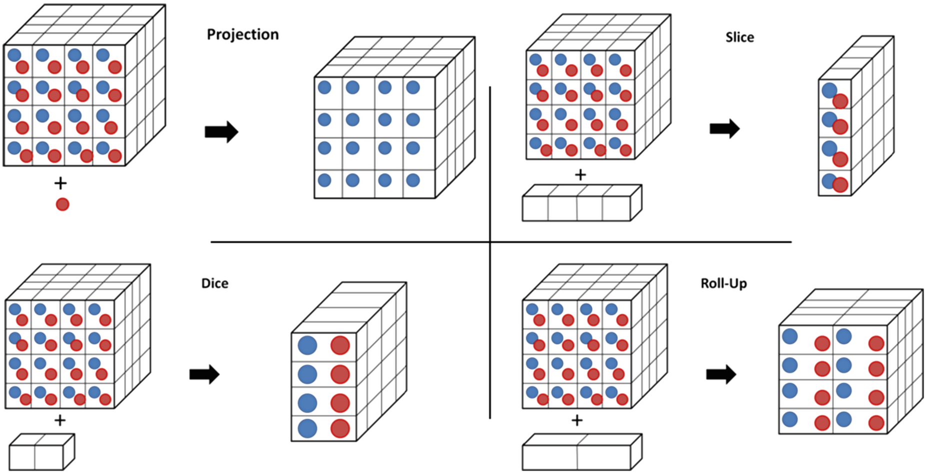


Fig. 3. Illustration of common OLAP operations with inputs and outputs (adapted from [21])

union, intersection, difference is out-of-scope. Multiple datasets can already be queried together if they are described by the same *qb:DataStructureDefinition*.

Each OLAP operation has as input and output a data cube. Therefore, operations can be nested. A nested set of OLAP operations lead to an OLAP query. For interpreting a set of OLAP operations as an OLAP query and evaluating the query using SPARQL on QB, we use and slightly adapt the notion of *subcube queries* [15].

Definition 3. (OLAP Query). *We define an OLAP query on a certain cube* $c = (cs, C)$*,* $cs = (?x, D, M)$ *as a subcube query* $Q = (c, q)$ *with* $c \in DataCube$ *and* q *a subcube query tuple* $(q_1, ..., q_{|D|}, m_1, ..., m_j)$*,* $j \leq |M|$ *[15]. The subcube query tuple contains for each dimension a tuple element* q_i*,* $Dimension(q_i) = D_i \in D$*,* $Hierarchy(q_i) = H_i \in Hierarchy$*,* $Level(q_i) = L_i \in Level$ *with* $dom(q_i) = \{?, ALL, x\}$*. The element* ? *marks a dimension as* inquired*, the* ALL *as aggregated, and* $x \in \mathcal{V}$ *fixes a dimension to a specific set of members. Also, for any queried measure the subcube query tuple contains an* $m_i \in M$*. For each dimension a granularity in the form of a hierarchy and level is specified. Note, for simplicity reasons we assume a fixed ordering of dimensions in the cube and in the subcube query tuple. An OLAP query returns a set of tuples from a data cube as defined by Definition 2.*

As examples, we describe three distinguishable subcube queries:

Full-Cube Query $(?, ?, ?, ..., m_1, ..., m_{|M|})$ returns the tuples on the highest granularity, i.e., the lowest level of each dimension, inside a data cube. //returns the tuples resulting from an aggregation over all dimensions. It contains all tuples described by the facts inside a data cube, plus any tuples not explicitly contained in the cube due to sparsity.

Point Query $(a_1, a_1, ..., a_{|D|}, m_1, ..., m_{|M|})$ with $a_i \in \mathcal{V}$, $|a_i| = 1, Dimension(a_i) = D_i$ returns one specific tuple from a data cube.

Fully-Aggregated Query $(ALL, ALL, \ldots, ALL, m_1, \ldots, m_{|M|})$ returns one single fact with measures aggregated over an empty select list.

In the following we describe how to evaluate each OLAP operation in terms of this query model and how a nested set of OLAP operations results in one specific OLAP subcube query. Given a data cube as input to an OLAP operation, the tuples from the cube are given as a full-cube query tuple $(?, ?, ?, ..., m_1, ..., m_{|M|})$.

Projection is defined as $Projection : DataCube \times Measure \rightarrow DataCube$ and removes a measure from the input cube and allows to query only for specific measures. We evaluate *Projection* by removing a measure from the subcube query tuple.

Slice is defined as $Slice : DataCube \times Dimension \rightarrow DataCube$ and removes a dimension from the input cube, i.e., removes this dimension from all selection lists over which to aggregate. We evaluate *Slice* by setting the tuple element of the dimension to *ALL*.

Dice is defined as $Dice : DataCube \times Dimension \times \mathcal{V} \rightarrow DataCube$ and allows to filter for and aggregate over certain dimension members. We evaluate *Dice* by setting the tuple element of that dimension to this particular set of members and aggregate over the set. Note, we regard dice not as a selection operation but a combined filter and slice operation.

Roll-Up is defined as $Roll - Up : DataCube \times Dimension \rightarrow DataCube$ and allows to create a cube that contains instance data on a higher aggregation level. We evaluate $Roll - Up$ on a dimension by specifying the next higher level of the specified hierarchy. Note, $Drill - Down$ can be seen as an inverse operation to $Roll - Up$.

As an example, consider an OLAP query on our *SecCubeGrossProfitMargin* cube for the cost of goods sold (*edgar:CostOfGoodsSold*) for each issuer (*edgar:issuer*) and each date until when each disclosure is valid (*edgar:dtend*), filtering by disclosures from two specific segments (*edgar:segment*). A nested set of OLAP operations that queries the requested facts can be composed as follows. In all our queries, we use prefixes to make URIs more readable:

```
Slice(
    Dice(
            Projection(
        edgar:SecCubeGrossProfitMargin,
        edgar:CostOfGoodsSold),
    edgar:segment,
    {edgar:segmentAHealthCareInsuranceCompany,
    edgar:segmentAResidentialRealEstateDeveloper}),
edgar:dtstart)
```

This query can then be represented as a subcube query with dimensions Issuer, Dtstart, Dtend, Segment:

```
(?, ALL, ?, {edgar:segmentAHealthCareInsuranceCompany,
edgar:segmentAResidentialRealEstateDeveloper}, edgar:
    CostOfGoodsSold)
```

Next, we describe how to evaluate such an OLAP query using a SPARQL query on QB. Since QB does not yet fully specify how to represent OLAP hierarchies, and since dimensions in our scenario were flat, in this paper, we simplify the problem of translating a subcube query to SPARQL and assume a queried data cube with only one hierarchy and level per dimension. A $Roll-Up$ has a similar effect to a *Slice* operation and can be added to our concept by a group-by not on the members of the lowest level of a dimension but on members of a higher level, specified via their $ROLLUPMEMBER$ and $ROLLUPLEVEL$ relations. An OLAP query $Q = (c, q)$ with $c \in DataCube$ and query tuple $q = (q_1, \dots, q_{|D|}, m_1, ..., m_j)$, $j \leq |M|$ can be translated into a SPARQL query using the following steps:

1. We initialise the SPARQL query using the URI of the data cube. We query for all instance data from the data cube, i.e., observations linking to datasets which link to the data structure definition.
2. For each selected measure, we incorporate it in the SPARQL query by selecting additional variables for each measure and by aggregating them using the aggregation function of that measure, using *OPTIONAL* patterns for cases of cube sparsity.
3. For each inquired dimension, we add query patterns and selections for all the instances of *skos:Concept* in the specified level and hierarchy of the dimension. We query for the observations showing property-value pairs for each of these variables, either directly using concepts or using literals linked from the concepts via *skos:notation*. We use *OPTIONAL* patterns for cases of cube sparsity. To display inquired dimensions in the result and correctly aggregating the measures, we *GROUP BY* each inquired dimension variable.
4. For each fixed dimension, we filter for those observations that exhibit for each dimension one of the listed members.

We transform our example from above to the following SPARQL query. Note, *UDF* represents the standard aggregation function from our scenario:

```
select ?dimMem0 ?dimMem1 UDF(?measureValues0) where {
?obs qb:dataSet ?ds.
?ds qb:structure edgar:SecCubeGrossProfitMargin.
?dimMem0 skos:member edgar:issuerRootLevel.
    OPTIONAL {?obs edgar:issuer ?dimMem0. }
?concept1 skos:member edgar:dtendRootLevel.
?concept1 skos:notation ?dimMem1.
    OPTIONAL {?obs edgar:dtend ?dimMem1. }

?obs edgar:segment ?slicerMem0.
     Filter(?slicerMem0 = edgar:
          segmentAHealthCareInsuranceCompany
    OR ?slicerMem0 = edgar:
        segmentAResidentialRealEstateDeveloper)

OPTIONAL {?obs egar:CostOfGoodsSold ?measureValue0. }
} group by ?dimMem0 ?dimMem1
```

5 Experiment: Evaluating an OLAP Query on Financial Linked Data

In this section, we demonstrate in a small experiment the applicability of our OLAP-to-SPARQL mapping to our scenario from the financial domain. In this experiment, we have a triple store with around 148,426 triples. The triple store describes a data cube *edgar:SecCubeGrossProfitMargin* that contains 17,448 disclosures that either disclose cost of goods sold or sales revenue net. The values of the measures fully depend on one of 625 different issuers (dimension *edgar:issuer*), the date a disclosure started (27 members of dimension *edgar:dtstart*) and ended (20 members of *edgar:dtend*) to be valid, and additional information (21,227 members of *edgar:segment*). The two measures (*edgar:CostOfGoodsSold* and *edgar:Sales*) have the unit USD and an aggregation function that returns the number of disclosures, or – if only one – the actual number. If fully materialised according to Definition 2, the cube contains $626 \cdot 28 \cdot 21 \cdot 21,228 = 7,813,772,064$ facts. To compute all of its facts, $2^4 = 16$ SPARQL subqueries would be needed.

In order to answer the OLAP question of our scenario, we use an OLAP client such as Saiku to compose OLAP operations to an OLAP query in MDX:

```
SELECT
 {edgar:cik1417907idConcept, edgar:cik106535idConcept} ON
    COLUMNS,
 CrossJoin(edgar:dtstartRootLevel.Members, edgar:
    dtendRootLevel.Members) ON ROWS
FROM [edgar:SecCubeGrossProfitMargin]
WHERE {edgar:CostOfGoodsSold}
```

The MDX query is sent to the OLAP engine, the resulting tuples will be visualised using a pivot table, a compact format to display multidimensional data [6]. Multidimensional elements are described in the MDX query using their unique URIs.[11] For an introduction to MDX, see its website.[12] A more detailed description of how to transform an MDX query into an OLAP query due to space constraints we leave for future work when we evaluate our OLAP-to-SPARQL mapping more thoroughly.

Now, we show that an MDX query can be transformed into an OLAP subcube query according to Definition 3 and evaluate the subcube query using SPARQL.

[11] Note, URIs need to be translated to an MDX-compliant format that does not use reserved MDX-specific characters, which is why we use the prefixed notation of URIs.

[12] http://msdn.microsoft.com/en-us/library/aa216770.

The result is a subset of all possible tuples from a data cube. The pivot table determines what dimensions to display on its columns and rows.

A nested set of OLAP operations to compose our OLAP query is as follows:

```
Slice(Projection(
     edgar:SecCubeGrossProfitMargin,
     edgar:CostOfGoodsSold),
edgar:segment)
```

This query can then be represented as a subcube query with dimensions Issuer, Dtstart, Dtend, Segment: $(?, ?, ?, ALL, CostOfGoodsSold)$. The resulting SPARQL query is as follows:

```
select ?dimMem0 ?dimMem1 ?dimMem2 count(xsd:decimal(?
    measureValue0)) sum(xsd:decimal(?measureValue0))
where {
?obs qb:dataSet ?ds.
?ds qb:structure edgar:SecCubeGrossProfitMargin.

?dimMem0 skos:member edgar:issuerRootLevel.
     OPTIONAL {?obs edgar:issuer ?dimMem0. }

?values1 skos:member edgar:dtstartRootLevel.
?values1 skos:notation ?dimMem1.
     OPTIONAL {?obs edgar:dtstart ?dimMem1. }

?values2 skos:member edgar:dtendRootLevel.
?values2 skos:notation ?dimMem2.
     OPTIONAL {?obs edgar:dtend ?dimMem2. }

OPTIONAL {?obs edgar:CostOfGoodsSold ?measureValue0. }
} group by ?dimMem0 ?dimMem1 ?dimMem2
```

UDF, our default aggregation function is algebraic, therefore, we had to compute the *SUM* and *COUNT* for the measure. We run the query after a reboot of the triple store. The query took 18sec and returned 58 tuples to be filled into the requested pivot table. The number of $7,813,772,064$ potential tuples in the cube does not have a strong influence on the query since the cube is very sparse, for instance, the triple store contains observations only for a fraction of segment members.

6 Discussion

In our experiment, we show the applicability of our mapping between OLAP and SPARQL queries. We correctly aggregate data on one specific granularity, defined by the mentioned inquired and fixed dimensions. Dimensions that are not mentioned will be automatically handled as having an *ALL* value [10], representing all possible values of the dimension. The aggregation results in correct calculations, since we assume that a QB data cube only contains facts on the lowest granularity level. Only an aggregation of observations from different granularities would result in incorrect numbers, e.g., a *SUM* over gender *male*, *female*, and *total*.

The SPARQL query created by our approach shows sufficiently fast in our small experiment but may not scale for larger datasets for the following reasons: First, data from the data cube is queried on demand, and no materialisation is done. Every OLAP query is evaluated using a SPARQL query without caching and reusing of previous results. Second, a *Dice* operation currently always includes a *Slice* operation; thus, all member combinations of inquired dimensions are calculated, even though only specific combinations might be required, as in the case of the two issuers in the OLAP query of our scenario. Third, OLAP clients and pivot tables require multidimensional data, i.e., data cubes containing facts linking to specific members of dimensions, but our SPARQL query returns relational tuples. Using the unique identifiers of dimensions, members, and measures, query result tuples need to be joined with the multidimensional data points as required by the OLAP client for filling the pivot table. Another possibility would be to use SPARQL *CONSTRUCT* queries to first materialise data cubes resulting from OLAP operations as RDF [8]. Populating the pivot table could then be done by simple SPARQL *SELECT* queries on this resulting multidimensional view. However, the applicability and performance of this approach to answer OLAP queries still needs to be evaluated.

In summary, though our OLAP algebra to SPARQL mapping may not result in the most efficient way to answer an business question and require additional efforts for usage in OLAP clients, it correctly computes all required tuples from the data cube with one SPARQL query, without the need for explicitly introducing the non-relational *ALL* member or using sub-queries [10].

7 Related Work

Kobilarov and Dickinson [14] have combined browsing, faceted-search, and query-building capabilities for more powerful Linked Data exploration, similar to OLAP, but not focusing on statistical data. Though years have passed since then, current literature on Linked Data interaction paradigms does not seem to expand on analysing large amounts of statistics.

OLAP query processing generally distinguishes three levels [4]: On the conceptual level, algebras of OLAP operations over data cubes are defined that are independent from a logical representation [1,3,9,19,21]. On the logical level, query processing mainly depends on the type of data structure on which to perform the computations and in which to store the results. Data structures can roughly be grouped into ROLAP, using relational tables and star or snowflake schemas, and MOLAP, using multidimensional arrays for directly storing and querying of data cubes [2,10,15,23]. The physical level is concerned with efficient execution of low-level executions such as index lookup or sorting over the data stored given a specific hardware and software.

The execution of OLAP operations mainly is concerned with the computation of the data cube and with storing parts of the results of that computation to efficiently return the results, to require few disk or memory space, and to remain easy to update if data sources change [15,17].

In this work we use the graph-based RDF data model and the QB vocabulary for querying and storing of multidimensional data. Both schema information and actual data is accessed using the Linked Data principles and managed using SPARQL on a triple store.

Other authors recognise the reduced initial processing and update costs of using triple stores and other dedicated query engines for query processing [18], but do not present approaches for executing OLAP queries directly over such Semantic Data Warehouses.

Etcheverry and Vaisman [8] present an algorithm to translate OLAP operations such as Roll-Up and Slice to SPARQL CONSTRUCT queries. Since results are cubes, one can nest operations. Different from our work, OLAP operations are executed for preprocessing of cubes from the Web that are then exported to a data warehouse for query processing.

Although there may be more efficient querying approaches such as special indexing and caching, to the best of our knowledge, this is the first work on executing OLAP queries on data cubes represented as RDF using SPARQL.

Several authors [20,24] motivate the integrating of data from the Web in OLAP systems. Yin and Pedersen [24] present a federated approach to "decorate" Data Cubes with virtual dimensions built from external data, which means that XML data can also be used in filter operations. For instance, Members of a level "Nation" in a Data Cube are linked to nations in an XML document providing additional information such the population which then can be used to filter for certain nations. Different from this approach, we directly execute multidimensional queries using SPARQL over RDF.

Diamantini and Potena [7] enrich a data cubes with a domain ontology represented using OWL as well as a mathematical ontology represented in XML standards for mathemtical descriptions such as MathML. Their goal is to provide analysts with useful background information and to possibly allow novel types of analyses, e.g., drill-down into single compound measures.

Mazón et al. [16] also motivate the use of external data to enhance Data Cubes; they propose the use of semantic relations such as hypernymy ("is-a-kind-of", generalization, e.g., cake is kind of baked goods) and meronymy (''is-a-part-of", aggregation, e.g., wheel is a part of car) between concepts provided by WordNet to enrich Dimension Hierarchies.

Although we so far only translate the common analytical operations, our approach could be extended with operations allowing filtering over decorations described in RDF. Also, OLAP client interfaces can be extended to display additional information related to analysed data cubes.

8 Conclusions and Future Work

We have presented an approach to interact with statistical Linked Data using Online Analytical Processing operations of "overview first, zoom and filter, then details-on-demand". For that, we define projection, slice, dice and roll-up operations on single data cubes in RDF reusing the RDF Data Cube vocabulary, map nested sets of OLAP operations to OLAP subcube queries, and evaluate those OLAP queries using SPARQL. Both metadata and OLAP queries are directly issued to a triple store; therefore, if the RDF is modified or updated, changes are propagated directly to OLAP clients. Though, our OLAP-to-SPARQL mapping may not result in the most efficient SPARQL query and require additional efforts in populating requested pivot tables, we correctly calculate required tuples from a data cube without inefficient full materialisation and without the need for explicitly introducing the non-relational *ALL* member or for using subqueries.

Future work may be conducted in three areas: 1) extending our current approach with OLAP hierarchies and OLAP operations over multiple cubes, e.g., drill-across; 2) implementing an OLAP engine to more thoroughly evaluate our current OLAP-to-SPARQL mapping and to investigate more efficient OLAP query execution plans, e.g., using RDF views; 3) investigating more Linked-Data-specific OLAP clients that allow external information to be used in queries and displayed.

Acknowledgements. The work presented in this paper has been funded in part by the German Ministry of Education and Research (BMBF) within the SMART research project (Ref. 02WM0800), the European Community's Seventh Framework Programme FP7/2007-2013 within the PlanetData project (Grant 257641) and MONNET project (Grant 248458), and the Science Foundation Ireland (Lion-2, Grant SFI/08/CE/I1380).

References

1. Agrawal, R.:, Gupta, A., Sarawagi, S.: Modeling multidimensional databases. In: Proceedings of the Thirteenth International Conference on Data Engineering (1997)
2. Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. ACM SIGMOD Record **26**, 65–74 (1997)
3. Chen, L., Ramakrishnan, R., Barford, P., Chen, B.C., Yegneswaran, V.: Composite subset measures. In: VLDB2006 Proceedings of the 32nd International Conference on Very Large Data Bases (2006)
4. Ciferri, C., Ciferri, R., Gómez, L., Schneider, M.: Cube algebra: a generic user-centric model and query language for OLAP cubes. IJDWM **9**, 39–65 (2012)
5. Codd, E., Codd, S., Salley, C.: Providing OLAP to user-analysts: an IT mandate. Technical report (1993)
6. Cunningham, C., Galindo-Legaria, C.A., Graefe, G.: PIVOT and UNPIVOT: optimization and execution strategies in an RDBMS. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, vol. 30 (2004)
7. Diamantini, C., Potena, D.: Semantic enrichment of strategic datacubes. In: Proceedings of the ACM 11th International Workshop on Data Warehousing and OLAP (2008)
8. Etcheverry, L., Vaisman, A.A.: Enhancing OLAP analysis with web cubes. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 469–483. Springer, Heidelberg (2012)

9. Gómez, L.I., Gómez, S.A., Vaisman, A.A.: A Generic data model and query language for spatiotemporal OLAP cube analysis categories and subject descriptors. In: EDBT 2012 (2012)
10. Gray, J., Bosworth, a., Lyaman, a., Pirahesh, H.: Data cube: a relational aggregation operator generalizing GROUP-BY, CROSS-TAB, and SUB-TOTALS. in: Proceedings of the Twelfth International Conference on Data Engineering (1995)
11. Harinarayan, V., Rajaraman, A.: Implementing data cubes efficiently. ACM SIGMOD Rec. **25**, 205–216 (1996)
12. Harth, A.: Visinav: a system for visual search and navigation on web data. J. Web Seman. **8**(4), 348–354 (2010)
13. Kämpgen, B., Harth, A.: Transforming statistical linked data for use in OLAP systems. In: I-Semantics 2011 (2011)
14. Kobilarov, G., Dickinson, I.: Humboldt: exploring linked data. In: Linked Data on the Web Workshop (LDOW2008) at WWW2008 (2008)
15. Li, X., Han, J., Gonzalez, H.: High-dimensional OLAP: a minimal cubing approach. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, vol. 30 (2004)
16. Mazón, J.-N., Trujillo, J., Serrano, M., Piattini, M.: Improving the development of data warehouses by enriching dimension hierarchies with WordNet. In: Collard, M. (ed.) Ontologies-Based Databases and Information Systems. LNCS, vol. 4623. Springer, Heidelberg (2007)
17. Morfonios, K., Konakas, S., Ioannidis, Y., Kotsis, N.: ROLAP implementations of the data cube. ACM Comput. Surv. **39**, 12 (2007)
18. Nebot, V., Berlanga, R.: Building data warehouses with semantic web data. Decis. Support Syst. **52**, 853–868 (2012)
19. Pardillo, J., Mazón, J.N., Trujillo, J.: Bridging the semantic gap in olap models: platform-independent queries. In: Proceedings of the ACM 11th International Workshop on Data Warehousing and OLAP (2008)
20. Pedersen, T.B., Gu, J., Shoshani, A., Jensen, C.S.: Object-extended OLAP querying. Data Knowl. Eng. **68**, 453–480 (2009)
21. Romero, O., Marcel, P., Abelló, A., Peralta, V., Bellatreche, L.: Describing analytical sessions using a multidimensional algebra. In: Cuzzocrea, A., Dayal, U. (eds.) Data Warehousing and Knowledge Discovery. LNCS, pp. 224–239. Springer, Heidelberg (2011)
22. Shneiderman, B.: The Eyes Have It: A task by data type taxonomy for information visualizations. In: Information Visualization (1996)
23. Vassiliadis, P., Sellis, T.: A survey of logical models for OLAP databases. ACM Sigmod Rec. **28**, 64–69 (1999)
24. Yin, X., Pedersen, T.B.: Evaluating XML-extended OLAP queries based on a physical algebra. Proceedings of the 7th ACM international workshop on Data warehousing and OLAP - DOLAP '04 p. 73 (2004)

Linguistic Modeling of Linked Open Data for Question Answering

Matthias Wendt(✉), Martin Gerlach, and Holger Düwiger

Neofonie GmbH, Robert-Koch-Platz 4, 10115 Berlin, Germany
{wendt,gerlach,duewiger}@neofonie.de
http://www.neofonie.de/Forschung

Abstract. With the evolution of linked open data sources, question answering regains importance as a way to make data accessible and explorable to the public. The triple structure of RDF-data at the same time seems to predetermine question answering for being devised in its native subject-verb-object form. The devices of natural language, however, often exceed this trFiple-centered model. But RDF does not preclude this point of view. Rather, it depends on the modeling. As part of a government funded research project named Alexandria, we implemented an approach to question answering that enables the user to ask questions in ways that may involve more than binary relations.

1 Introduction

In recent years, the Semantic Web has evolved from a mere idea into a growing environment of Linked Open Data (LOD)[1] sources and applications. This is due in particular to two current trends: The first is automatic data harvesting from unstructured or semi-structured knowledge that is freely available on the internet, most notably the DBpedia project [1]. The second notable trend is the evolution of linked data sources with possibilities of collaborative editing such as Freebase[2]. The growth of LOD gives rise to a growing demand for means of semantic data exploration. Question Answering (QA), being the natural device of querying things and aqcuiring knowledge, is a straightforward way for end users to access semantic data.

RDF[3] and other languages for triple-centered models, which are often used to model and describe linked data, seem to predetermine a specific way of thinking - and of asking questions. Many RDF sources offer information in the form "X birthplace Y" and "X birth-date Z", etc. Of course, in natural language, we are used to formulate more complex queries. It is natural to make statements like "X was born in Y on Z". While this does not matter as long as singular events like birth or death are involved, things become more complicated as soon as events are involved

[1] See http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

[2] http://www.freebase.com/.

[3] http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 102–116, 2015.
DOI: 10.1007/978-3-662-46641-4_8

that can occur more than once. For example, the question "Who was married to Angelina Jolie in 2006?" can only be answered if the temporal (and potentially limited) nature of a relation like marriage is taken into account.

In this paper we present the QA-driven ontology design behind Alexandria[4], a platform for exploring data of public interest comprising a system for answering questions in German. The domain consists of persons, organizations, locations as well as works such as books, music albums, films and paintings. Moreover, the Alexandria ontology is designed for holding information on events relating the various resources, including temporal information and relations involving more than two participants – so called N-ary Relations. Also, we describe the mapping algorithm used in our question answering system and how it benefits from the ontology design.

The ontology is built from and continuously being updated with data primarily from Freebase, and few parts from DBpedia, news feeds, and user generated content.

2 Related Work

Open domain question answering is of current research interest. There are several approaches to the subject based on linguistic analysis of natural language questions for generating queries against linked data.

FREyA [5] and PowerAqua [9,10] are both question answering systems that are to a certain degree independent of the underlying ontology schema. Both systems work on existing Linked Open Data *as is* and can be configured to use multiple ontologies. They rely on rather shallow approaches to query mapping, in favor of portability and schema-independence. However, this also limits them to the data structures and languages used by the schemas (e.g., DBpedia does not support N-ary relations).

Other systems are based on deeper, compositional mapping approaches. For example, ORAKEL [3,4] translates syntax trees constructed by lexicalized tree adjoining grammars (LTAGs) to a representation in first order logic which can be converted to F-Logic [8] or SPARQL[5] queries, depending on the target knowledge base. ORAKEL also principally supports N-ary relations. Though the system is in principle very similar to the one presented in this paper, it is not proven to scale up to a large data set.

In contrast to other projects that use Linked Open Data for question answering, our approach is an attempt to combine the advantages of availability of huge LOD sources and of tailoring the T-Box to the use case of QA. While the latter facilitates the fully automated mapping of natural language questions to SPARQL queries, we trade off the possibility to use existing labels for T-Box entities, which, combined with existing lexical resources such as WordNet[6], GermaNet[7], etc., boost lexical coverage.

[4] http://alexandria.neofonie.de/.

[5] http://www.w3.org/TR/rdf-sparql-query/.

[6] http://wordnet.princeton.edu/.

[7] http://www.sfs.uni-tuebingen.de/lsd/.

Another difference to the above-mentioned projects is that the focus of Alexandria is on answering questions in German, not English.

3 Design of the Alexandria Ontology

The design of the Alexandria Ontology was basically driven by practical demands of the application as well as linguistic considerations. According to [7], our approach can be seen as a unification of the "Type 4" and the "Type 3" approaches to ontology creation. The knowledge base has to meet the following requirements:

Linguistic Suitability. The data model needs to be suitable for natural language question answering, i.e. mapping natural language parse tree structures onto our data must be possible.

LOD Compatibility. Compatibility with existing LOD sources like Freebase and DBpedia needs to be maintained in order to facilitate mass data import for practical use.

Scalability. Large amounts of data need to be stored, maintained and updated while keeping the time for answering a question at minimum.

One of the major aspects relating to *linguistic suitability* in the Alexandria use case is that its target domain goes beyond what we refer to in the following as *attributive data*, i.e. data about things that are commonly known as named entities like persons, organizations, places, etc. In addition, the domain was designed to contain what we call *eventive data*, i.e. (historic) events and relations to participants within them.

As mentioned above, there are certain relations, such as *birth*, where this distinction is not important, because n-ary relations consisting of unique binary parts (like place and date of birth) can be covered by joining on a participant (the person) as proposed in [4]. The distinction between eventive and attributive data becomes important when relations are involved, which (may) occur repetitively and/or include a time span. For example, questions like "Who was married to Angelina Jolie in 2001?" and "Which subject did Angela Merkel major in at the German Academy of Sciences?" can not be answered by joining on binary facts in the general case.

It is possible to model such eventive n-ary facts as proposed in Pattern 1, use case 3 of the W3C Working Group Note on N-ary Relations on the Semantic Web[8]. This approach is also close to the semantic formalization advocated in Neo-Davidsonian theories [12], where participants in an event are connected to the event using roles. The sentence "Brutus stabbed Caesar." for example would be formalized like this:

$$\exists e[\text{stabbing}(e) \wedge \text{agent}(e, \text{brutus}) \wedge \text{theme}(e, \text{caesar})]$$

[8] http://www.w3.org/TR/swbp-n-aryRelations/.

The variable e in this case, ranges over entities of type ‘event’. By formalizing the model in terms of event semantics, a range of linguistic phenomena may be accounted for, such as:

- iterated adverbial modification (“Brutus stabbed Caesar {brutally, with a knife, in the back}”)
- quantification of events (“How often was Gerhard Schröder chancellor of Germany?”)
- nominalization (“Gerhard Schröder was elected chancellor of Germany” - “the election of Gerhard Schröder”)

As for the aspect of *LOD compatibility*, it is our aim to access existing large-scale sources to populate our knowledge base. DBpedia was the first LOD source to retrieve and constantly update its data repository by crawling Wikipedia[9]. Apart from its possibilities for end-users to add and update information, the majority of data contained in Freebase is obtained from Wikipedia as well. Therefore, using one (or both) of these sources is an obvious starting point for harvesting information on a broad range of popular entities, as it is required by Alexandria.

However, though DBpedia contains much valuable attributive data for entities of our interest, it does not offer eventive information as stated above. Also, DBpedia’s T-Box does not provide a model for adding such n-ary facts, either.

As opposed to DBpedia, which relies on the RDF standard, Freebase implements a proprietary format. Whereas in RDF, all information is abstractly represented by triples, Freebase abstractly represents information as links between topics. The Freebase data model incorporates n-ary relations by means of Compound Value Types[10], also called “Mediators”. A mediator links multiple topics and literals to express a single fact.

So Freebase’s data model suits our requirements, but we need to use RDF to be able to use Virtuoso Open Source Edition[11] which has proven to *scale well* for both loading and querying the amounts of data we expected.

Using the Freebase query API to pull a set of topics and links, an RDF based knowledge base can be built according to the Neo-Davidsonian model. The API also supports querying link updates for continuously updating the knowledge base.

There are straightforward mappings of Freebase topic and mediator types onto OWL[12] classes, and of Freebase link types onto OWL properties. For example, a marriage relation is imported from Freebase as follows:

```
nary:m_02t82g4 rdf:type       dom:Marriage ;
               dom:spouse     res:Angelina_Jolie ;
               dom:spouse     res:Brad_Pitt ;
               alx:hasStart   "2005"^^xsd:gYear .
```

[9] http://www.wikipedia.org/.

[10] http://wiki.freebase.com/wiki/Compound_Value_Type.

[11] http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/.

[12] Web Ontology Language, builds on RDF, see http://www.w3.org/TR/owl2- overview/.

The subject URI is generated from the Freebase mediator ID. The resource URIs are generated from Freebase topic names with some extra processing and stored permanently for each Freebase topic ID.

We differentiate the following three layers of our ontology (which correspond to three namespace prefixes alx:, dom:, and res: that appear in the examples):

The Upper Model (alx:) contains the abstract linguistic classes needed for a language-, domain- and task-independent organization of knowledge. The Alexandria upper model is inspired by [6].

The Domain Model (dom:) contains the concrete classes and properties for entities, events and relations of the modeled domain (e.g., Marriage, Study) as subclasses of the upper model classes and properties. Needed to make the domain-specific distinctions which are necessary for the task of question answering.

The A-Box (res:) consists of all "resources", i.e. entity, event and relation instances, known to Alexandria.

The examples of Angela Merkel's education and Angelina Jolie's and Brad Pitt's marriage, which we used above, would be represented as shown in Table 1.

Table 1. Upper and domain model

Upper model concept	Domain concept	U.m. role props.	Domain props.	Participant
Agentive process	Study	Agent	Student	Angela Merkel
Effective process	Study	Affected	Subject	Quantum Chemistry
Locative relation	Study	Location	Institution	German Academy of Sciences
		Located	Student	Angela Merkel
Attributive rel	Marriage[a]	Carrier	Spouse	Brad Pitt
		Attribute	Spouse	Angelina Jolie
Temporal concept	Marriage	Start	Wedding date 2005	

[a] In this example, marriage is modeled as a symmetric relation, expressing a spouse as attribute of the other spouse, i.e. carrier and attribute may be swapped.

Schematically, domain concepts are modeled as subclasses of one or more upper model concepts and domain properties are modeled as subproperties of one or more upper model properties, where the latter correspond to the semantic roles. Hints to the upper model concepts subsuming a target domain concept may be obtained by finding verbs that express the concept and then matching the roles defined for the upper model concepts to the respective surrounding constituents in a set of syntactic contexts (questions or other sentences).

4 Putting the Model in Action: Question Answering

As mentioned above, one of the major design goals of our ontology schema was to stay reasonably close to the phenomena and structure of natural language. Achieving this would facilitate the mapping of a natural language question to a SPARQL graph pattern that conveys the information need expressed in the question. The basic idea of the translation algorithm is to understand the problem of mapping of natural language to SPARQL as a graph mapping problem.

From a linguistic viewpoint, the syntactic structure of a sentence may be represented in the form of a dependency tree, as obtained using a dependency parser. A dependency graph is formalized like this:

Given a set L of dependency labels, a *dependency graph* for the sentence $x = w_1 \dots w_n$ is a directed graph $D = (V_D, E_D)$ with:

1. V_D is a set of vertices $\{w_0, w_1, \dots, w_n\}$ and
2. $E_D \subseteq V_D \times L \times V_D$ a set of labeled edges

The vertices are composed of the tokens in the sentence plus an artificial root node w_0. A well-formed dependency graph is a tree rooted at w_0.

Likewise, the structure of a SPARQL Select query basically consists of a graph pattern (in the Where clause) and a projection. Given a set of variable names N_V (`?x`, `?y` ...), the set of concept names N_C, a set of role names N_P, a set of resource names N_R and a set of literals N_L defined by the ontology, we define a SPARQL Select Graph $G = (V_G, E_G, P_G)$ as:

1. $V_G \subseteq N_V \cup N_R \cup N_C \cup N_L$
2. $E_G \subseteq N_V \cup N_R \times N_P \times V_G$
3. the projection $P_G \subseteq N_V$

Formally, we define the translation as a mapping $f(D)$ of a dependency graph D to a SPARQL Select Graph G.

4.1 Linguistic Processing

The dependency graph is the result of the application of a **linguistic analysis** to the input sentence. An example of a resulting dependency structure may be found on the left in Fig. 1. The analysis consists in tokenization, POS-tagging[13] and dependency parsing. Dependency parsing is conducted using the MaltParser [11], which was trained on the German Tiger corpus[14] [2]. The corpus has been slightly adapted by adding a small sub-corpus of German questions and a minor change to the set of role labels used.

To normalize surface form variation and identify morphosyntactic features, lemmatization and morphological analysis is applied to each of the tokens. This is roughly illustrated by the lemmata in square brackets at the verbal nodes (e.g. “verheiratet” has the lemma “verheiraten”).

[13] For German tokenization and POS-Tagging we use OpenNLP with some pre-trained models. (http://incubator.apache.org/opennlp/).

[14] http://www.ims.uni-stuttgart.de/projekte/TIGER/.

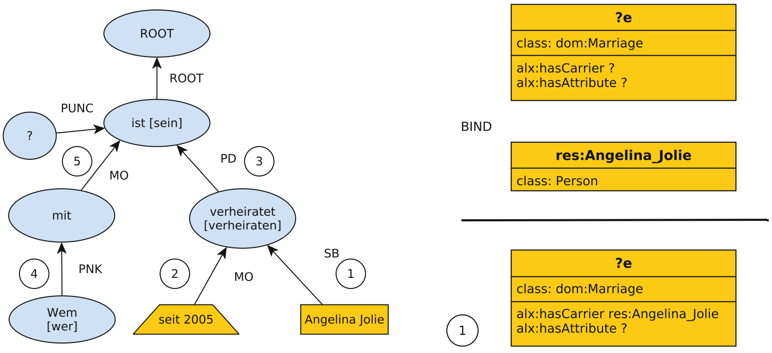


Fig. 1. Dependency parse of the sentence "Mit wem ist Angelina Jolie seit 2005 verheiratet?" ("Who is Angelina Jolie married to since 2005") and step (1) of the composition algorithm - binding the argument "Angelina Jolie" to the carrier property.

4.2 Compositional Semantics

The mapping of the dependency graph to the SPARQL query is largely done in two steps: *lexicalization* and *composition*. By *lexicalization* we refer to the process of mapping tokens (lemmata) or multi-word units to corresponding ontological units.

We refer to the identification of resources (identified by resource URIs) of the A-Box as *lexical named entity identification*. For this, we make use of the title (the name of an entity) and the alternative names (consisting of synonyms and different surface forms) that are imported from Freebase into a Lucene[15] index containing the resource URI in Alexandria (e.g. `res:Angelina_Jolie`), and the OWL classes it belongs to. While the user enters a question, matching entities are looked up in the index based on whole words already entered and a disambiguation choice is continuously updated. The user can select from the found entities at any time, whereupon the respective part of the question is updated.

The second noteworthy component in lexical named entity identification is the identification of dates (and time). For these, we have adapted the open source date parser provided by the Yago project[16] to German.

All other linguistic tokens or configurations (linguistic units) corresponding to T-Box concepts are mapped using hand-crafted lexica.

The complete set of mappings for the question shown in Fig. 1 is shown in Table 2.

Our syntax-semantics mapping is largely done by the composition of the lexical semantic entries attached to each dependency node. This lexicalized approach devises the notion of a *semantic description*. A semantic description represents the semantic contribution of a dependency node or (partial) dependency tree

[15] http://lucene.apache.org/.

[16] http://www.mpi-inf.mpg.de/yago-naga/javatools/.

Table 2. Types of lexical mappings

T-Box class	T-Box role	A-Box URI	Literal	T-Box class	
Custom Lexica		Lucene	Date, Literal Parser	Custom Lexica	
"wer"	"mit"	"Angelina Jolie"	"2005"	"verheiraten"	"sein"
`dom:Person`	`alx:hasAttribute`	`res:Angelina_Jolie`	`"2005"^^xsd:date`	`dom:Marriage`	`owl:Thing`

and encodes obligatory semantic complements (slots). During the composition, the slots are being filled by semantic descriptions (properties) until the semantic description is satisfied.

By virtue of the lexical mapping each linguistic unit is mapped to a set of semantic descriptions, also called *readings*.

Given a set of variable names N_V, the set of concept names N_C, a set of role names N_P, a set of resource names N_R and a set of literals N_L defined by the ontology, a semantic description S of an ontological entity $n \in N_V \cup N_L \cup N_R$ is defined as a five-tuple $S = (n, c, Sl, Pr, Fl)$ with:

1. $c \in N_C$ the concept URI of the semantic description
2. $Sl = [r_1, r_2, \ldots, r_n]$ a ordered set of slots ($r_i \in N_P$)
3. $Pr = \{(p_1, S_1), \ldots (p_m, S_m)\}$ with S_j a semantic description and $p_m \in N_P$
4. $Fl \subseteq \{proj, asc, desc\}$ a set of flags (with $proj$ indicating that n to be part of the projection of the output graph

For convenience, we define the following access functions for the semantic descriptions $S = (n, c, Sl, Pr, Fl)$:

1. $node(S) = n \Leftrightarrow S = (n, c, Sl, Pr, Fl)$
2. $pred(S) = \{p | (p, o) \in Pr\} \Leftrightarrow S = (n, c, Sl, Pr, Fl)$.

A semantic description $S = (n, c, Sl, Pr, Fl)$ is well-formed if the set of bound properties and the slots are disjoint $pred(S) \cap Sl = \emptyset$ and all bound properties are uniquely bound $\forall (p, o) \in Pr \rightarrow \neg\exists (p, o_1) \in Pr \wedge o \neq o_1$.

By definition, there is a strong correlation between a semantic description and a SPARQL Select query. A SPARQL Select query can recursively be built from a semantic description $S = (n, c, Sl, Pr, Fl)$ and an initially empty input graph $G_0 = (V_{G_0} = \emptyset, E_{G_0} = \emptyset, P_{G_0} = \emptyset)$:

```
**toSPARQL**$(S, G_0)$: $G_m$
$G_0 = (V_0 \cup \{n\}, E_0, P_0)$: the output SPARQL graph pattern
$E_0 = E_s \cup \{(n, \mathsf{a}, c)\}$
$P_0 \leftarrow P \cup \{n\} \Leftrightarrow proj \in Fl$ otherwise $P_o = P$

**foreach** $(p_i, o_i)$ **in** $(p_1, o_1), \ldots, (p_m, o_m) = Pr$
**begin**
    $E_i \leftarrow E_{i-1} \cup (n, p, node(o))$
    $G_i \leftarrow$ toSPARQL($o_i$, $G_{i-1}$)
**end**
**return** $G_m$
```

To give an example in an informal notation, the linguistic units of the sentence “Mit wem ist Angelina Jolie seit 2005 verheiratet?” are displayed in Table 3. The first row shows the linguistic unit, the ontological unit described corresponds to the variable or resource URI in the second row. The prefix `?!` in a variable designation (e.g. `?!x`) is equivalent to the flag *proj*, denoting that the variable will be part of the projection, i.e. $proj \in Proj$. Note that the verbal nodes “verheiratet” and “sein” are each mapped to a (distinct) variable `?e`, which corresponds to the Neo-Davidsonian event variable e.

Slots and bound properties are displayed in the third column. The slots are designated with the argument being just a `?`, whereas a variable is denoted by the prefix ‘`?`’ and a lower case letter (e.g. `?v`). In the example above the semantic descriptions for “verheiratet” and “mit” contain the slots `alx:hasCarrier` (verb only) and `alx:hasAttribute`.

Table 3. Semantic descriptions of lexical units.

“Angelina Jolie”	`res:Angelina_Jolie`	`a dom:Person`

“seit 2005”	`?e`	`a alx:TemporalConcept ;`
		`alx:hasStart 2005 .`

“mit”	`?e`	`a alx:AttributiveRelation ;`
		`alx:hasAttribute ?`
“wem”	`?!x`	`a dom:Person`
“sein”	`?e`	`a alx:AttributiveRelation`

“verheiratet”	`?e`	`a dom:Marriage ;`
		`alx:hasCarrier ? ;`
		`alx:hasAttribute ?`

4.3 Putting it all Together

The composition algorithm devises a fixed set of two-place composition operators, called actions. An action defines the mapping of two semantic descriptions related to an edge in the dependency graph to a composed semantic description, corresponding to the semantics of the subgraph of the dependency tree.

The two most important actions are BIND and MERGE. These two basic operations on the semantic descriptions involved in the composition intuitively correspond to (1) the mapping of the syntactic roles to semantic roles (also called semantic role labeling) and (2) the aggregation of two nodes to one in the output graph pattern. Given two semantic descriptions $S_1 = (n_1, c_1, Sl_1, Pr_1, Fl_1)$ and $S_2 = (n_2, c_2, Sl_2, Pr_2, Fl_2)$, the semantic operators are defined like this:

$$BIND(S_1, S_2) = \begin{cases} S(v, c_1, Sl_1 - \{\max(Sl_1)\}, Pr_1 \cup \{(\max(Sl_1), S_2)\}) \\ \hfill \text{if } \operatorname{range}(\max(Sl_1)) \sqcap c_2 \neq \bot \\ NULL \hfill \text{otherwise} \end{cases}$$

$$MERGE(S_1, S_2) = \begin{cases} S(v, lcs(c_1, c_2), Sl_1 \cup Sl_2, Pr_1 \cup Pr_2) & \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } c_1 \sqcap c_2 \neq \bot & \\ \qquad\qquad\qquad\qquad \wedge pred(S_1) \cap pred(S_2) = \emptyset & \\ NULL \qquad\qquad\qquad\qquad\qquad \text{otherwise} & \end{cases}$$

The function $lcs(c_1, c_2)$ gives the least common subsumer of two concepts, i.e. a concept c such that $c_1 \sqsubseteq c$ and $c_2 \sqsubseteq c$ and for all e such that $c_1 \sqsubseteq e$ and $c_2 \sqsubseteq e$ then $c \sqsubseteq e$.

The semantic role labeling implemented by BIND depends on a total order of semantic roles, which has to be configured in the system, e.g.:

`alx:hasAgent` > `alx:hasAffected` > `alx:hasRange`

This order determines the ordering of the slots Sl in a semantic description. It stipulates a hierarchy over the semantic roles of an n-ary node in the SPARQL graph pattern. It is reflected by an ordering over the syntactic role labels which is roughly equivalent to the linguistic notion of an *obliqueness hierarchy* [13], for example:

`SB` > `OC` > `OC2`

Formally, the obliqueness hierarchy defines a total order $>_L$ over the set of dependency labels L.

For the composition, each of the labels in the label alphabet is assigned one of the semantic operators. The algorithm chooses the operator that is defined to build the composition. The following table shows an excerpt of this mapping:

SB	OC	PNK	MO	PD	PUNC
BIND	BIND	BIND	MERGE	MERGE	IGNORE

The action IGNORE simply skips the interpretation of the subtree. The composition algorithm iterates over the nodes in the dependency graph in a top-down manner, for each edge applying the action defined for the edge label pairwise to each reading of the source and target node.

The algorithm works in a directed manner by sorting the outgoing edges of each node in the dependency graph according to a partial order $\geq_D$ derived from the order of the labels $>_L$.

$$(v_1, l_1, w_1) >_D (v_2, l_2, w_2) \Leftrightarrow l_1 >_L l_2$$

This ordering induces a hierarchy over the actual syntactic arguments in the dependency graph that corresponds to the hierarchy of role labels used for argument binding that was mentioned above. In cases where the order is not defined among an edge, we assume that the algorithm may choose for an arbitrary ordering. The correspondance between these orders controls the order in which the graph is traversed and therefore, in particular, the correlation of syntactic and semantic roles (semantic role labeling).

The mapping is implemented by the transformation algorithm sketched below. It takes as input a dependency graph $D(V_D, E_D)$ with the root node w_0 as the initial node c in the graph traversation. The nodes are traversed in the order of the hierarchy to assure the correct binding. Note that the transformation may have multiple semantic descriptions as its output. An output semantic description $S = (n, c, Sl, Pr, Fl)$ is only accepted, if all of its slots Sl have been filled. We then apply the toSPARQL operation to arrive at the final SPARQL query.

```
**transform**($D$,$c$): $S$
$c \leftarrow w_0$: the current node
$S \leftarrow \emptyset$: the set of output readings

**foreach** $(c, l, v)$ in sort(outgoing($c$), $\geq_D$)
**begin**
    $R_c \leftarrow$ readings($c$)
    $R_v \leftarrow$ transform($v$, $D$)
    **foreach** $(r_c, r_v)$ in $R_c \times R_v$
    **begin**
        $s \leftarrow$ apply(operator(l), $r_c$, $r_v$)
        **if**($s \neq$ NULL)
            $S \leftarrow S \cup \{s\}$
    **end**
**end**
```

Our working example is visualized in the Figs. 1, 2 and 3 in an informal representation. Note that two nodes of the dependency graph are compound nodes, corresponding to more than one token: “Angelina Jolie” and “seit 2005”. These correspond to multi-unit items that are already handled by lexical components - namely the named entity finder and the date finder respectively.

By means of the ordering upon the dependency labels, the algorithm traverses the tree in a depth-first, highest priority first manner. The left picture in

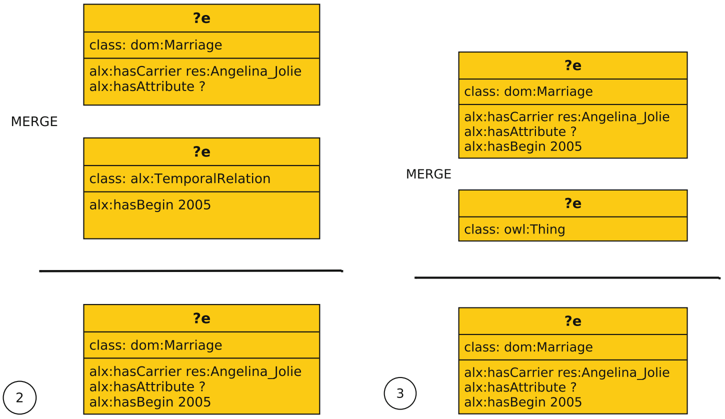


Fig. 2. Step (2) and (3) of the composition: merging the temporal modifier “seit 2005” (“since 2005”) with the result of (1) and (3) “sein” (“to be”) with the result of (2).

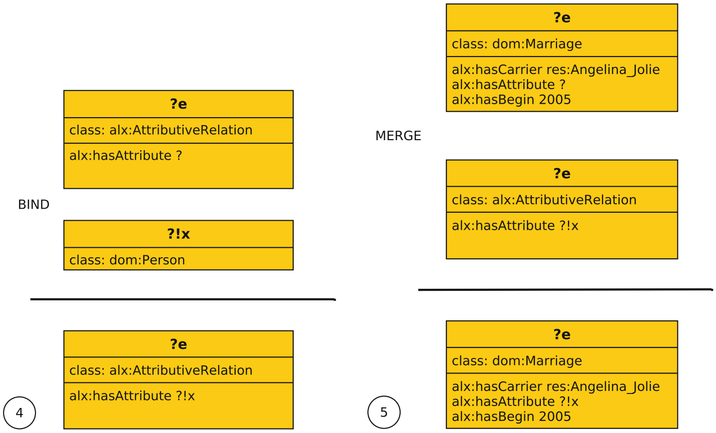


Fig. 3. Building the interpretation of the prepositional modifier phrase (4) by binding the interpretation of "wer" ("who") to the attribute role of the attributive interpretation of "mit" ("with") and (5) merging the results of (4) and (3).

Fig. 1 shows the dependency graph with encircled numbers for the transformation steps taken. The transformation starts by applying action number (1) (BIND) to the pair of semantic descriptions corresponding to the syntactic nodes of "verheiraten" and "Angelina Jolie". The result is a semantic description of class `dom:Marriage` with its `alx:hasCarrier` role bound by `res:Angelina_Jolie`.

In step number (2) in Fig. 2, the algorithm merges the temporal information of the node "seit 2005" with the marriage semantic description derived so far. Step (3) merges the result with the rather trivial description corresponding to "sein" ("to be"). In Fig. 3, step (4) consists in binding the semantic description corresponding to the interrogative pronoun "wer" ("who") to the attribute role of the attributive reading of the prepositition "mit" ("with"). The result is merged (5) with the semantic description of marriage derived so far, arriving at our final result (leaving aside the semantically void root and punctuation node).

5 Results

The N-ary modeling requires more triples for simple (binary) facts than using RDF/OWL properties like DBpedia, because there is always an instance of a relation concept comprising `rdf:type` and participant role triples.

At the time of writing, the Alexandria ontology contained approx. 160 million triples representing more than 7 million entities and more than 13 million relations between them (including literal value facts like amounts, dates, dimensions, etc.). We imported the triples into Virtuoso Open Source Edition, which scales as well as expected with respect to our goals.

80 % of all query types understood by the algorithm (i.e. mappable onto valid SPARQL queries) take less than 20 ms in average for single-threaded linguistic

processing on a 64 bit Linux system running on Intel® Xeon® E5420 cores at 2.5 GHz, and pure in-memory SPARQL processing by Virtuoso Open Source Edition on a 64 bit Linux system running on eight Intel® Xeon® L5520 cores at 2.3 GHz and 32 GB of RAM.

The question answering system works fast enough to be used in a multi-user Web frontend like http://alexandria.neofonie.de.

The performance of the algorithm is in part due to the high performance of malt parser with a liblinear model, which runs in less than 5 ms per question but performs slightly less accurate than a libsvm model.

Table 4. Quality of results of the Alexandria question answering on the QALD-2 training set translated to German.

Question subset	Avg. precision	Avg. recall	Avg. f-measure
All 100 QALD questions	0.25	0.27	0.25
Questions with SPARQL generated (qgen)	0.49	0.52	0.48
Questions on data without mismatch (qmm)	0.59	0.57	0.57
qgen ∩ qmm	0.92	0.89	0.89

The performance of the question answering system has been measured using the training set of the QALD-2 challenge[17] which consists of 100 English questions and is based on DBpedia. As the question answering in Alexandria covers only German, all questions were translated to German first. The results are shown in Table 4. 51 of the questions had a corresponding representation in terms of the Alexandria model and a SPARQL query could be generated. The second row shows the results for these questions.

As the Alexandria knowledge base relies on data imported from Freebase to a great extend, the comparability of the results with the gold standard coming from DBpedia is limited. The comparison of both datasets results in various mismatches. For example, the comparison of questions having a set of resources as answer type, is done via the indirection of using the labels. This is possible just because the labels are extracted from Wikipedia by both Freebase and DBpedia. However, some of the labels have been changed during the mapping.

Overall, we have identified the following error types:

1. different labels for the same entities
2. different number of results for aggregate questions
3. query correct but different results
4. training data specifies "out of scope" where we can provide results
5. question out of scope for Alexandria.

[17] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge\&q=2.

Type 1 applies particularly often to the labels of movies, most of which are of the form "Minority Report (film)" in DBpedia, and "Minority Report" in Alexandria. Another source of errors (2) results from a different number of resources associated with aggregate questions (involving a count). The question "How many films did Hal Roach produce?" for example yields 509 results in DBpedia and 503 results in Alexandria. The third type corresponds to a difference in the data set itself, that is when different information is stored. For example, in Alexandria the highest mountain is the "Mount Everest" whereas in DBpedia it is the "Dotsero".

The last two error types involve questions that are out of scope (4 and 5). On the one hand, Alexandria's data model is more expressive than that of DBpedia as a result of the n-ary modeling. On the other hand, Alexandria lacks information since we concentrate on a mapped subset of Freebase. According to the evaluation, the answer "out of scope" is correct if the question cannot be answered using DBpedia.

Out of the 82 questions containing (any) erroneous results 63 belong to one of the error classes mentioned above. The last two rows of Table 4 show the results for all questions that do not belong to any of these error classes.

6 Discussion and Future Work

We have shown a practical implementation of a question answering algorithm, making use of a task-tailored ontology model. The key to our approach is to view the mapping from natural language to SPARQL queries as a graph mapping problem. The algorithm benefits from this in being both simple and generic, while the semantic coverage may easily be extended by adding entries to the lexica. Also, the algorithm itself is easily extensible by adding new actions or adding flags to the set of flags in the semantic description. For example, simply by adding a "count" flag to the set of flags on a semantic description, we have implemented quantifying questions including counting over entities and counting over iterations of a relation, e.g. "Wie oft war Gerhard Schröder Bundeskanzler?" ("How often was G.S. Chancellor of Germany?").

Yet, the algorithm heavily depends on the structure of the parse. It is therefore fragile with respect to the parser's performance, which is not an issue most of the times, due to the short length of typical questions. Future work will improve on this by tuning the parser model more towards question sentences. Also, one might object that the stipulation of the correspondence of the input and the output graph is not correct when looking at coreference. In these cases our intuition would be that the node corresponding to the coreferring expression in the semantic graph is connected by more than one distinct incoming paths, whereas the dependency graph is always a tree. Future work will include accounting for coreference, which may be solved within our paradigm using equality filters between distinct variables.

Acknowledgements. Research partially funded by the German Federal Ministry for Economics and Technology as part of the THESEUS research program (http://theseus-programm.de/en/about.php.).

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia: a crystallization point for the web of data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 154–165 (2009)
2. Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G.: The TIGER treebank. In: Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol (2002)
3. Cimiano, P.: ORAKEL: a natural language interface to an F-Logic knowledge base. In: Meziane, F., Métais, E. (eds.) NLDB 2004. LNCS, vol. 3136, pp. 401–406. Springer, Heidelberg (2004)
4. Cimiano, P., Haase, P., Heizmann, J., Mantel, M.: Orakel: A portable natural language interface to knowledge bases. Technical report (2007)
5. Damljanovic, D., Agatonovic, M., Cunningham, H.: FREyA: an interactive way of querying linked data using natural language. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 125–138. Springer, Heidelberg (2012)
6. Elhadad, M., Robin, J.: Surge: a comprehensive plug-in syntactic realization component for text generation. Technical report (1998)
7. Hovy, E.: methodologies for the reliable construction of ontological knowledge. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 91–106. Springer, Heidelberg (2005)
8. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. J. ACM **42**, 741–843 (1995)
9. Lopez, V., Motta, E., Uren, V.S.: PowerAqua: fishing the semantic web. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 393–410. Springer, Heidelberg (2006)
10. Lopez, V., Nikolov, A., Sabou, M., Uren, V., Motta, E., d'Aquin, M.: Scaling up question-answering to linked data. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 193–210. Springer, Heidelberg (2010)
11. Nivre, J., Hall, J.: Maltparser: A language-independent system for data-driven dependency parsing. In: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories, pp. 13–95 (2005)
12. Parsons, T.: Events in the Semantics of English: A study in Subatomic Semantics. MIT Press, Cambridge (1990)
13. Pollard, C.J., Sag, I.A.: Information-Based Syntax and Semantics. CSLI Lecture Notes. CSLI Publications, Stanford University (1987). Distributed by University of Chicago Press

SPARTIQULATION: Verbalizing SPARQL Queries

Basil Ell(✉), Denny Vrandečić, and Elena Simperl

KIT, Karlsruhe, Germany
{basil.ell,denny.vrandecic,elena.simperl}@kit.edu

Abstract. Much research has been done to combine the fields of Databases and Natural Language Processing. While many works focus on the problem of deriving a structured query for a given natural language question, the problem of query verbalization – translating a structured query into natural language – is less explored. In this work we describe our approach to verbalizing SPARQL queries in order to create natural language expressions that are readable and understandable by the human day-to-day user. These expressions are helpful when having search engines that generate SPARQL queries for user-provided natural language questions or keywords. Displaying verbalizations of generated queries to a user enables the user to check whether the right question has been understood. While our approach enables verbalization of only a subset of SPARQL 1.1, this subset applies to 90 % of the 209 queries in our training set. These observations are based on a corpus of SPARQL queries consisting of datasets from the QALD-1 challenge and the ILD2012 challenge.

Keywords: SPARQL · Natural language generation · Verbalization

1 Introduction

Much research has been done to combine the fields of Databases and Natural Language Processing to provide natural language interfaces to database systems [22]. While many works focus on the problem of deriving a structured query for a given natural language question or a set of keywords [10,21,27,30], the problem of query verbalization – translating a structured query into natural language – is less explored. In this work we describe our approach to verbalizing SPARQL queries in order to create natural language expressions that are readable and understandable by the human day-to-day user.

When a system generates SPARQL queries for a given natural language question or a set of keywords, the verbalized form of the generated query is helpful for users, since it allows them to understand whether the right question has been asked to the queried knowledge base and, if the query is executed and results are presented, how the results have been retrieved. Therefore, verbalization of SPARQL queries may improve the experience of users of any such SPARQL

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 117–131, 2015.
DOI: 10.1007/978-3-662-46641-4_9

query generating system such as natural language-based question answering systems or keyword-based search systems.

In this paper we describe the current state of our SPARTIQULATION system,[1] which allows verbalization of a subset of SPARQL 1.1 SELECT queries in English.

The remainder of this paper is structured as follows. Section 2 gives an overview of the query verbalization approach in terms of the system architecture and the tasks that it performs. Section 3 presents the elements of our approach, Sect. 4 revisits existing work, and in Sect. 5 conclusions are drawn and an outlook is provided.

2 Query Verbalization Approach

2.1 Introduction

Our approach is inspired by the pipeline architecture for natural language generation (NLG) systems and the set of seven tasks performed by such systems as introduced by Reiter and Dale [19]. The input to such a system can be described by a four-tuple (k, c, u, d) – where k is a knowledge source (not to be confused with the knowledge base a query is queried against), c the communicative goal, u the user model, and d the discourse history. Since we neither perform user-specific verbalization nor do we perform the verbalization in a dialog-based environment, we omit both the user model and the discourse history. The communicative goal is to communicate the meaning of a given SPARQL query q. However, there are multiple options. Three basic types of linguistic expressions can be used: (i) statements that describe the search results where this description is based on the query only and not on the actual results returned by a SPARQL endpoint (e.g. *Bavarian entertainers and where they are born*), (ii) a question can be formulated about the existence of knowledge of a specified or unspecified agent (e.g. *Which Bavarian entertainers are known and where are they born?*), and (iii) a query can be formulated as a command (e.g. *Show me Bavarian entertainers and where they are born*). Thus, the communicative goal can be reached in three modes: *statement verbalization*, *question verbalization*, or *command verbalization*. Since the only communicative goal is to communicate the meaning of a query to a user, the various modes the system is built for and the omissions of both the user model and the discourse history, the input to our system is a tuple (k, m) where k is the SPARQL query and $m \in \{\mathit{statement}, \mathit{question}, \mathit{command}\}$ is a mode.

2.2 Components and Tasks

In this section we present our approach along the seven tasks involved in NLG according to Reiter and Dale [19]. This work is the first step towards the verbalization of SPARQL queries. So far we put a focus on *document structuring*, but

[1] The name is derived from joining *SPARQL* and *articulation*. A demo is available at http://km.aifb.kit.edu/projects/spartiqulator.

not on *lexicalization*, *aggregation*, *referring expression generation*, *linguistic realisation*, and *structure realisation*. Note that the modes in which a communicative goal can be reached are regarded in the task *linguistic realization* only.

The pipeline architecture is depicted in Fig. 1. Within the Document Planner the content determination process creates messages and the document structuring process combines them into a document plan (DP), which is the output of this component and the input to the Microplanner component. Within the Microplanner the processes lexicalization, referring expression generation and aggregation take place, which results in a text specification (TS) that is made up of phrase specifications. The Surface Realizer then uses this text specification to create the output text.

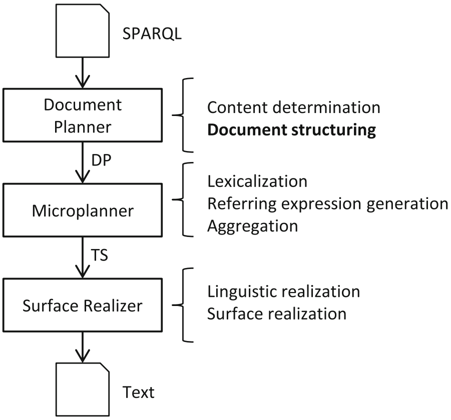


Fig. 1. Pipeline architecture of our NLG system

Content Determination is the task to decide which information to communicate in the text. In the current implementation we decided not to leave this decision to the system. What is communicated is the meaning of the input query without communicating which vocabularies are used to express the query. For example if *title* occurs in the verbalization and is derived from the label of a property, then it is hidden to the user whether this has been derived from http://purl.org/dc/elements/1.1/title or http://purl.org/rss/1.0/title.

Document Structuring is the task to construct independently verbalizable messages from the input query and to decide for their order and structure. These messages are used for representing information, such as that a variable is selected, the class to which the entities selected by the query belong to or the number to which the result set is limited. The output of this task is a document plan. Our approach to document structuring consists of the following elements:

1. Query graph representation
2. Main entity identification

3. Query graph transformation
4. Message types
5. Document plan.

These are the main contributions of this work. We continue this section with an introduction of the remaining tasks. In Sect. 3, each of the elements of our approach are presented in detail.

Lexicalization is the task of deciding which specific words to use for expressing the content. For each entity we dereference its URI and in case that RDF data is returned, we check if an English label is provided using one of the 36 labeling properties defined in [6]. Otherwise, we derive a label from the URI's local name. In case of properties, the 7 patterns introduced by Hewlett et al. in [11] are used. For example, Hewlett et al. provide the following pattern:

(is) VP P

- Examples: producedBy, isMadeFrom
- Expansions: X is produced by Y, X is made from Y.

The local name *producedBy* of a property *ex:producedBy* is expanded to *produced By* and its constituents are part-of-speech tagged. The expansion rule given for this pattern declares that a triple `ex:X ex:producedBy ex:Y` can be verbalized as *X is produced by Y.*

The main entity[2] is verbalized as *things*. If a constraint for the class of the main entity such as `?m rdf:type yago:AfricanCountries` is given, then it can be verbalized as *African countries*.[3] If the query is limited to a single result using `LIMIT 1` and no sort order is defined using `ORDER BY`, then it can be verbalized as *An African country.* Otherwise, if a sort order is defined such as `ORDER BY DESC(?population)`, then it can be verbalized as *The African country* as in *The African country with the highest population.* Other variables are also verbalized as *things* unless a type is either explicitly given using `rdfs:type` or implicitly given using `rdfs:domain` or `rdfs:range`. For example, this information is regarded when verbalizing the query `SELECT ?states ?uri WHERE { ?states dbo:capital ?uri .}` as *Populated places and their capitals.* Here, the domain of the property *dbo:capital* is defined as *dbpedia-owl:PopulatedPlace.*

Referring Expression Generation is the task of deciding how to refer to an entity that is already introduced. Consider the following two example verbalizations:

1. *Albums of things named Last Christmas and where available their labels.*
2. *Albums of things named Last Christmas and where available the labels of these albums.*

[2] The main entity is rendered as the subject of the verbalization.

[3] *African countries* is the rdfs:label of yago:AfricanCountries.

In the beginning of the verbalizations the entities albums and things are introduced. At the end labels are requested. In the first verbalization it is not clear whether the labels of the albums or the labels of the things are requested, whereas in the second verbalization it is clear that the labels of the albums are requested.

Aggregation is the task to decide how to map structures created within the document planner onto linguistic structures such as sentences and paragraphs. For example, without aggregation a query such as `SELECT ?m WHERE` {`?m dbo:starring res:Julia_Roberts . ?m dbo:starring res:Richard_Gere .` } would be verbalized as *Things that are starring Julia Roberts and that are starring Richard Gere.* With aggregation the result is more concise: *Things that are starring Julia Roberts as well as Richard Gere.*

Linguistic Realization is the task of converting abstract representations of sentences into real text. Thereby the modes *statement*, *question*, and *command* are regarded. As introduced in the next chapter, chunks of content of a SPARQL query are represented as messages given the list of message types (MT) from Fig. 5. For each of the message types (1)–(9) a rule is invoked that produces a sentence fragment, for example for the MT $MRVR_lL$ – which is an instance of the MT $M(RV)^*R_lL$ – the rule `article(lex(prop1)) + lex(prop1) + L` produces for two triples `?uri dbpedia:producer ?producer` and `?producer rdfs:label "Hal Roach"` the text `a producer named "Hal Roach"`. The function `article` choses an appropriate article (`a` or `an`) depending on the lexicalization `lex(prop1)` of the property.

Structure Realization is the task to add markup such as HTML code to the generated text in order to be interpreted by the presentation system, such as a web browser. Bontcheva [2] points out that hypertext usability studies [18] have shown that formatting is important since it improves readability. Indenting complex verbalizations, adding bullet points and changing the font size can help to communicate the meaning of a query.

3 Document Structuring

The elements our approach consists of can be summarized as follows. We transform textual SPARQL SELECT queries into a graphical representation – the query graph – which is suitable for performing traversal and transformational actions. Within a query graph we identify the main entity which is a variable that is rendered as subject of a verbalization. After a main entity is identified the graph is transformed into a graph where the main entity is the root. Then the graph is split into independently verbalizable parts called messages. We define a set of message types that allow to represent a query graph using messages. Message types are classified due to their role within the verbalization. The document plan is presented which orders the messages according to their classes and is the output of the Document Planner – the first component in our NLG pipeline.

Some observations in this chapter, namely regarding the main entity identification in Sect. 3.2 and the message types in Sect. 3.4, are based on a training set. This training set is derived from a corpus of SPARQL queries consisting of datasets from the QALD-1 challenge[4] and the ILD2012 challenge.[5] The full dataset contains 263[6] SPARQL SELECT queries and associated manually created questions. In order to derive a training set we used 80 % of each dataset as training data – in total 209 SELECT queries.[7] Since in our approach we cannot yet handle all features of the SPARQL 1.1 standard, we had to exclude some queries. Within this training set of 209 queries we excluded the queries with the UNION feature (20 queries) and those that were not parsable (1 query). This means that this subset – 188 queries in total – covers 90 % of the queries within the training set.

3.1 Query Graph Representation

We parse a SPARQL query into a query graph since this allows for easier manipulation of the query compared to its textual representation. Thereby each subject S and object O of a triple pattern $< S, P, O >$ within the query is represented by a node in the query graph. The nodes are connected by edges labeled with P. Since $< S, P, O >$ is a triple pattern and not an RDF triple, this means that each element can be a variable. Unless the subject or object is a variable, for each subject or object an own node is created. Therefore multiple non-variable nodes with the same label may exist. For every variable that appears within the query in subject or object position only one node is created.

As an example regard the SPARQL query in textual representation in Listing 1 and the visual representation of the query graph in Fig. 2. In this visual representation,[8] nodes are filled if they represent resources and not filled if they represent variables. Nodes are labeled with the name of the resource, variable, literal or blank node respectively. Labels of variables begin with a question mark. Labels of variables that appear in the SELECT clause of a query are underlined. Literal values are quoted. Edges are labeled with the name of the property and point from subject to object. Filters are attached to their respective variable(s) and parts of the graph that appear only within an OPTIONAL clause are marked as such.

[4] http://www.sc.cit-ec.uni-bielefeld.de/qald-1.

[5] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/.

[6] For nine questions no SPARQL query is given since they are out of scope regarding the datasets provided for the challenge. 28 queries are ASK queries.

[7] 37 queries from `2011-dbpedia-train`, 36 queries from `2011-musicbrainz-train`, 68 queries from `2012-dbpedia-train`, and 68 queries from `2012-musicbrainz-train`.

[8] Note that this not a complete visual representation of SPARQL SELECT queries and only solves for the purpose of visually representing the query graph examples.

```
SELECT DISTINCT ?uri ?string WHERE {
  ?states rdf:type yago:AfricanCountries .
  ?states dbo:capital ?uri .
  ?uri dbp:population ?population .
  FILTER ( ?population < 1000000 ) .
  OPTIONAL {
    ?uri rdfs:label ?string.
    FILTER (lang(?string) = 'en')
  }
}
```

Listing 1. Example SPARQL query

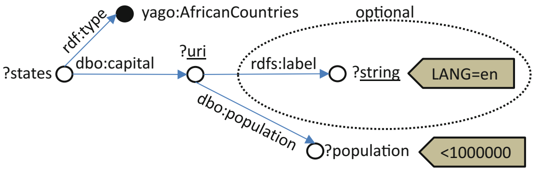


Fig. 2. Example query graph

3.2 Main Entity Identification

We perform a transformation of the query graph, since it reduces the number of message types that are necessary to represent information contained in the query graph thus simplifying the verbalization process. This transformation is based on the observation that in most queries one variable can be identified that is rendered as the subject of a sentence. For example, when querying for mountains (bound to variable `?mountain`) and their elevations (bound to variable `?elevation`), then `?mountain` is verbalized as the subject of the verbalization `mountains and their elevations`. We refer to this variable as the *main entity* of a query. However, for some queries no such element exists. Consider for example the query `SELECT * WHERE` { `?a dbpedia:marriedTo ?b .`}. Here a tuple is selected and in a possible verbalization *Tuples of married entities*[9] no single variable appears represented as a subject. In order to identify the main entity we define Algorithm 1 that applies the ordered list of rules shown in Fig. 3. These rules propose the exclusion of members from a candidate set. We derived them by looking at queries within the training set having multiple candidates. The candidate set C for a given query is initialized with variables that appear in

[9] DBpedia provides no `rdfs:domain` and `rdfs:range` information, such as `foaf:Person` for this property. Therefore here we give a generic verbalization to demonstrate the effect of missing domain and range information.

the SELECT clause[10] and the algorithm eliminates candidates step by step. Q denotes the set of triples within the WHERE clause of a query, R_t is the property `rdf:type` and R_l is a labeling property from the set of 36 labeling properties identified by [6]. The application of an exclusion rule R_i to a candidate set C, denoted by $R_i(C)$, results in the removal of the set E which is proposed by the exclusion rule.

We identified the ordered list of exclusion rules shown in Fig. 3. The numbers show how often a rule was successful in reducing the candidate set for the 188 queries within our training set. In some cases (61, 32.45 %) no rule was applied since the candidate set contained only a single variable. In the case that given the rules above the algorithm does not manage to reduce the candidate set to a single variable (21, 11.17 %), the first variable in lexicographic order is selected.

Rule 1 $E := \{x \in C \mid \text{"}x \textit{ appears in OPTIONAL only}\text{"}\}$
Rule 1 (71, 37.77%) proposes removing candidates that appear within the WHERE clause only within OPTIONAL blocks. For example in a query `SELECT ?a WHERE { ex:R1 ex:R2 ?a . OPTIONAL { ?a ex:R3 ?b . } }` the variable `b` is removed from the candidate set which contains `a` and `b`.

Rule 2 $E := \{z \in C \mid \neg\exists(z, R_t, u) \in Q\}$
if $\exists c_1 \in C : \neg\exists(c_1, R_t, x) \in Q \wedge \exists c_2 \in C : \neg\exists(c_2, R_t, y) \in Q$
Rule 2 (10, 5.32%) proposes removing candidates that represent subjects that are not constrained via `rdf:type` in the case that there are candidates that are constrained via *rdf:type*. For example in a query `SELECT ?a ?b WHERE { ?a rdf:type ex:R1 . ?b ex:R2 ex:R3 . }` with `ex:R2` $\neq$ `rdf:type`, the variable `b` is removed from the candidate set which contains `a` and `b`.

Rule 3 $E := \{z \in C \mid \neg\exists(z, R_l, u) \in Q\}$
if $\exists c_1 \in C : \neg\exists(c_1, R_l, x) \in Q \wedge \exists c_2 \in C : \neg\exists(c_2, R_l, y) \in Q$
Rule 3 (25, 13.3%) proposes removing candidates for which the existence of a label is not constrained or requested in the case that there are candidates for which the existence of a label is constrained or a label is requested. For example in a query `SELECT ?a ?b WHERE { ?a <`R_l`> ?l . ?b ex:R2 ex:R3 . }` where R_l is a labelling property, the variable `b` is removed from the candidate set which contains `a` and `b`.

Fig. 3. Exclusion rules

As an example regard the SPARQL query in Listing 1 which is visually presented in Fig. 2. The candidate set is initialized as $\{uri, string\}$. Rule 1 proposes to remove the variable *string* since it appears only within an OPTIONAL clause. Since the candidate set is reduced to $\{uri\}$ containing a single entity, this entity is the main entity.

3.3 Query Graph Transformation

Algorithm 2 transforms a query graph into a graph for which the main entity is the root and all edges point away from the root. Therefore, the algorithm

[10] In case of a `SELECT *` query, all variables within the WHERE clause are candidates.

Algorithm 1. Applying exclusion rules to candidate set.

```
if $|C| = 1$ then
    return $C$
while $|C| > 1$ do
    for all $R_i \in R$ do
        if $|R_i(C)| > 0$ then
            $C \leftarrow R_i(C)$
            if $|C| = 1$ then
                return $C$
return $\emptyset$
```

maintains three sets of edges: edges that are already processed (P), edges that need to be followed (F), and edges that need to be transformed (T) which means reversed. An edge is reversed by exchanging subject and object and by marking the property (p) as being reversed (p^r).

Algorithm 2. Graph transformation

```
$P \leftarrow \emptyset,\ F \leftarrow \{(s,p,o) \in Q | s = m\},\ T \leftarrow \{(s,p,o) \in Q | o = m\}$    (init)
while $F \neq \emptyset$ or $T \neq \emptyset$ do
    for all $(s_i, p_i, o_i) \in F$ do
        for all $(s_j, p_j, o_j) \in Q \setminus (P \cup F \cup T)$ do
            if $o_i = s_j$ then
                $F \leftarrow F \cup \{(s_j, p_j, o_j)\}$
            else if $o_i = o_j$ then
                $T \leftarrow T \cup \{(s_j, p_j, o_j)\}$
        Move $(s_i, p_i, o_i)$ from $F$ to $P$
    for all $(s_i, p_i, o_i) \in T$ do
        for all $(s_j, p_j, o_j) \in Q \setminus (P \cup F \cup T)$ do
            if $s_i = s_j$ then
                $F \leftarrow F \cup \{(s_j, p_j, o_j)\}$
            else if $s_i = o_j$ then
                $T \leftarrow T \cup \{(s_j, p_j, o_j)\}$
        $T \leftarrow T \setminus \{(s_i, p_i, o_i)\}$
        $P \leftarrow P \cup \{(o_i, p_i^r, s_i)\}$
return $P$
```

The query graph shown in Fig. 2 is transformed into the query graph shown in Fig. 4 where the main entity – the variable *uri* – is highlighted. Compared to the graph before the transformation, the edge *dbo:capital* was reversed. Therefore this edge now points away from the main entity and is marked as being reversed by the minus in superscript.

3.4 Message Types

We identified the set of 14 message types (MT), shown in Fig. 5 that allow us to represent the 209 queries from our training set. Here, $M(RV)*$ denotes a

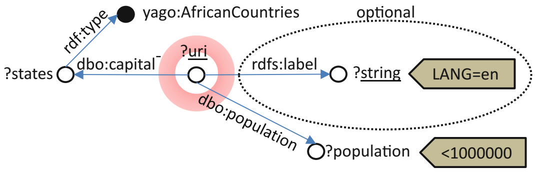


Fig. 4. Example query graph after transformation

path beginning at the main entity via an arbitrary number of property-variable pairs such as `?M <ex:R1> ?V1 . ?V1 <ex:R2> ?V2 .` The first 9 MTs represent directed paths in the query graph which means that for each directed path that begins at the main entity, we represent this path with a message. R_l denotes a labeling property and R_t the property `rdf:type`. The MTs $ORDERBY$, $LIMIT$, $OFFSET$ and $HAVING$ represent the respective SPARQL features.

As an example the SPARQL query in Listing 1 is represented using the 7 messages shown in Fig. 6. Note that due to the graph transformation the property `dbo:capital` is reversed which is denoted by `REV: 1` in message 2. This query can be verbalized as: *English names of African countries having capitals which have a population of less than 1000000 and the English names of these capitals.* Note that the plural form *capitals* instead of *capital* is used per default since no information is available that a country has exactly one capital. The filter for English labels is stored within message 6 representing the variable `string` as `lang: en`.

While this set of message types is sufficient for the given training set, which means that all queries can be represented using these message types, we extended this list with 7[11] more types in order to be prepared for queries such as `SELECT ?s ?p ?o WHERE { ?s ?p ?o. }` and `SELECT ?p WHERE { ?s ?p ?o. }` where instead of generating text, canned text is used, such as *All triples in the database* and *Properties used in the database.*

3.5 Document Plan

The document plan (DP), which is the output of the Document Planner and input to the Microplanner, contains the set of messages and defines their order. The verbalization consists of two parts. In the first part the main entity and its constraints are described, followed by a description of the requests (the variables besides the main entity that appear in the select clause) and their constraints. In a second part, if available and not already communicated in the first part, the selection modifiers are verbalized. According to these 3 categories – abbreviated with *cons*, *req*, and *mod* – we classify the message types (MT) as follows. The MTs (1), (2), (4), (6), (7), and (9) from Fig. 5 belong to the class *cons*, the MTs (3), (5), and (8) belong to the class *req*. MTs (1), (2), (4), (6), (7) and (9) may

[11] Given that all three variables can either be selected or not selected and at least one variable needs to be selected, this results in 7 combinations.

(1) $\boldsymbol{M(RV)^{*}RR}$ Messages of this type represent (RV)*-paths in the query that end with a resource in property position and a resource in object position, for example the path `?main ex:starring ex:Richard_Gere`.

(2) $\boldsymbol{M(RV)^{*}RL}$ Example (RV)*-path:
`?main ex:date "1960-12-29"^^xsd:dateTime`.

(3) $\boldsymbol{M(RV)^{*}RV}$ Example (RV)*-path: `?main ex:duration ?d`.

(4) $\boldsymbol{M(RV)^{*}R_{l}R}$ While a query such as `SELECT * WHERE {` `?m rdfs:label ex:R .` `}` is syntactically correct, semantically it is not valid since the range of the property *rdfs:label* is *rdfs:Literal* and not *rdfs:Resource*. We introduce this message type since we cannot assume that every query processed by our verbalizer is semantically correct.

(5) $\boldsymbol{M(RV)^{*}R_{l}V}$ Example (RV)*-path: `?main ex:label ?l`.

(6) $\boldsymbol{M(RV)^{*}R_{l}L}$ Example (RV)*-path: `?main ex:label "John Cage"@en`.

(7) $\boldsymbol{M(RV)^{*}R_{t}R}$ Example (RV)*-path: `?main rdf:type ex:SoloMusicArtist`.

(8) $\boldsymbol{M(RV)^{*}R_{t}V}$ Example (RV)*-path: `?main rdf:type ?t`.

(9) $\boldsymbol{M(RV)^{*}R_{t}L}$ Similarly to message type 4, a query such as `SELECT * WHERE` { `?m rdf:type "L" .` } is semantically not valid since the range of the property *rdf:type* is *rdfs:Class* and not *rdfs:Literal*.

(10) ***VAR*** A message of this type stores the name of the variable, whether it appears within the select clause, whether it is the main entity, whether it appears only within optional clauses, whether it is counted as in `SELECT COUNT(DISTINCT ?m)`, and the list of filters based on this variable.

(11) ***ORDERBY*** A message of this type stores the order of the solution sequence as specified by the ORDER BY clause, such as `ODER BY DESC(?main)`.

(12) ***LIMIT*** If a limit is specified using `LIMIT 10` then this integer value is stored.

(13) ***OFFSET*** If an offset is specified, for example using `OFFSET 10`, then this integer value is stored.

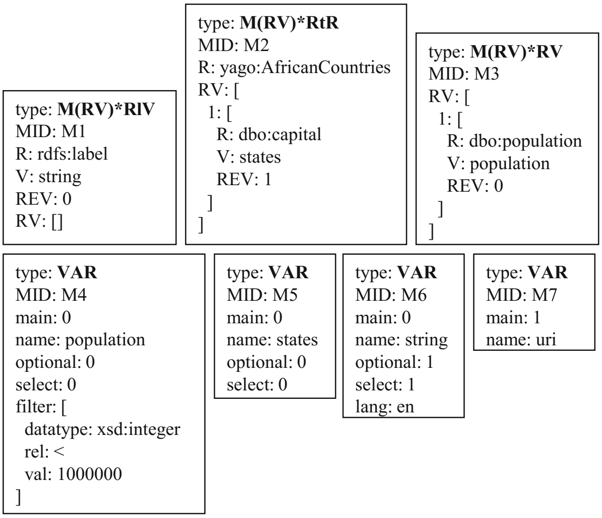
(14) ***HAVING*** A message of this type stores the order of the solution sequence as specified by the HAVING clause, such as in `GROUP BY ?x HAVING(AVG(?size) > 10)`.

Fig. 5. Message types

also belong to class *req* if they contain a variable besides the main entity that appears in the select clause. MTs (11) – (14) belong to the class *mod*. The VAR message is not classified since its only purpose is to store information about variables and variables are verbalized when verbalizing other messages. For the example query in Listing 1, which is represented using the messages shown in Fig. 6, the messages M1, M2, and M3 are classified as *cons*, the message M1 is classified as *req* and no message is classified as *mod*.

4 Related Work

While to the best of our knowledge no work is published on the verbalization of SPARQL queries, related work comes from three areas: verbalization of RDF data [5,15,24,25,29], verbalization of OWL ontologies [1,3,4,7–9,11,12,14,20, 23,26,28], and verbalization of SQL queries [13,16,17]. Although the first two



type: **M(RV)*RlV**
MID: M1
R: rdfs:label
V: string
REV: 0
RV: []

type: **M(RV)*RtR**
MID: M2
R: yago:AfricanCountries
RV: [
1: [
R: dbo:capital
V: states
REV: 1
]
]

type: **M(RV)*RV**
MID: M3
RV: [
1: [
R: dbo:population
V: population
REV: 0
]
]

type: **VAR**
MID: M4
main: 0
name: population
optional: 0
select: 0
filter: [
datatype: xsd:integer
rel: <
val: 1000000
]

type: **VAR**
MID: M5
main: 0
name: states
optional: 0
select: 0

type: **VAR**
MID: M6
main: 0
name: string
optional: 1
select: 1
lang: en

type: **VAR**
MID: M7
main: 1
name: uri

Fig. 6. Messages representing the SPARQL query in Listing 1.

fields provide techniques that we can apply to improve the lexicalization and aggregation tasks, such as the template-based approach presented in [5], the document structuring task, on which we focus here, is rarely explored. Compared to the SQL verbalization work by Minock [16,17], where they focus on tuple relational queries, our problem of verbalizing SPARQL queries is different in the sense that we strive for having a generic approach that can be applied to any datasource without being tied to any schema. Patterns need to be manually created to cover all possible combinations for each relation in the schema whereas in our work we defined a set of message types that are schema-agnostic. Koutrika et al. [13] annotate query graphs with template labels and explore multiple graph traversal strategies. Moreover, they identify a main entity (the *query subject*), perform graph traversal starting from that entity, and distinguish between *cons* (*subject qualifications*) and *req* (*information*).

5 Conclusions and Outlook

For the task of verbalizing SPARQL queries we focused on a subset of the SPARQL 1.1 standard which covers 90 % of the queries in a corpus of 209 SPARQL SELECT queries. Evaluation will have to show the representativeness of this corpus compared to real-life queries and the qualities of the verbalizations generated using our SPARTIQULATION system. While in our architecture 6 tasks are needed to generate verbalizations, our main focus has been the task of *document structuring* which we described in this work. In order to realize the full verbalization pipeline, 5 other tasks need to be explored in future work. Since the current approach is mostly schema-agnostic – only terms from the vocabularies RDFa and RDFS as well as a list of labeling properties from various vocabularies are regarded – we believe that this approach is generic in terms of being applicable to queries for RDF datasources using any vocabularies. However, in the future the tasks of lexicalization can be improved by regarding schemas such as FOAF and OWL. FOAF is interesting since if an entity is a *foaf:Person*, it can be treated differently. For example the person's gender can be regarded. OWL is interesting since if it is known that a property is functional, then the singular form can be used instead of, as per default, the plural form.[12] Having message types designed for specific vocabularies allows to tailor the verbalization to a specific use case and may lead to more concise verbalizations. In the current implementation, message types are hard-coded thus limiting the flexibility of the approach. Having the possibility to load a set of message types into the system would add the possibility to integrate automatically learned or application-specific message types.

Acknowledgements. The work presented here was partially supported by the project RENDER under the grant number FP7257790 funded by the Seventh Framework Program of the European Commission. We would like to thank the attendees at the ILD workshop at ESWC 2012 in Heraklion for their valuable feedback.

References

1. Aguado, G., Bañón, A., Bateman, J.A., Bernardos, S., Fernández, M., Gómez-Pérez, A., Nieto, E., Olalla, A., Plaza, R., Sánchez, A.: ONTOGENERATION: reusing domain and linguistic ontologies for Spanish text generation. In: Workshop on Applications of Ontologies and Problem Solving Methods, ECAI 1998 (1998)
2. Bontcheva, K.: Generating tailored textual summaries from ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 531–545. Springer, Heidelberg (2005)
3. Bontcheva, K., Wilks, Y.: Automatic report generation from ontologies: the MIAKT approach. In: Meziane, F., Métais, E. (eds.) NLDB 2004. LNCS, vol. 3136, pp. 324–335. Springer, Heidelberg (2004)
4. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL syntax - towards a controlled natural language syntax for OWL 1.1. In: Golbreich, C., Kalyanpur, A., Parsia, B. (eds.) OWLED, CEUR Workshop Proceedings, vol. 258. CEUR-WS.org (2007)
5. Davis, B., Iqbal, A.A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Handschuh, S.: RoundTrip ontology authoring. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 50–65. Springer, Heidelberg (2008)
6. Ell, B., Vrandečić, D., Simperl, E.: Labels in the web of data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 162–176. Springer, Heidelberg (2011)
7. Fliedl, G., Kop, C., Vöhringer, J.: Guideline based evaluation and verbalization of OWL class and property labels. Data Knowl. Eng. **69**, 331–342 (2010)

[12] For example the query `SELECT ?m WHERE { ex:PersonA ex:wife ?m . }` can then be verbalized as *The wife of PersonA* instead of *The wives of PersonA*.

8. Galanis, D., Androutsopoulos, I.: Generating multilingual descriptions from linguistically annotated OWL ontologies: the NaturalOWL system. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG 2007, Stroudsburg, PA, USA, pp. 143–146. Association for Computational Linguistics (2007)
9. Gareva-Takasmanov, L., Sakellariou, I.: OWL for the masses: from structured OWL to unstructured technically-neutral natural language. In: Kefalas, P., Stamatis, D., Douligeris, C. (eds.) BCI, pp. 260–265. IEEE Computer Society (2009)
10. Haase, P., Herzig, D., Musen, M., Tran, T.: Semantic Wiki search. In: Aroyo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 445–460. Springer, Heidelberg (2009)
11. Hewlett, D., Kalyanpur, A., Kolovski, V., Halaschek-Wiener, C.: Effective NL paraphrasing of ontologies on the semantic web. In: End User Semantic Web Interaction Workshop at the 4th International Semantic Web Conference (2005)
12. Kaljurand, K., Fuchs, N.E.: Verbalizing OWL in attempto controlled English. In: Golbreich, C., Kalyanpur, A., Parsia, B. (eds.) OWLED, CEUR Workshop Proceedings, vol. 258. CEUR-WS.org (2007)
13. Koutrika, G., Simitsis, A., Ioannidis, Y.E.: Explaining structured queries in natural language. In: ICDE 2010 (2010)
14. Liang, S.F., Stevens, R., Rector, A.: OntoVerbal-M: a multilingual verbaliser for SNOMED CT. In: Montiel-Ponsoda, E., McCrae, J., Buitelaar, P., Cimiano, P. (eds.) Multilingual Semantic Web, CEUR Workshop Proceedings, vol. 775. CEUR-WS.org (2011)
15. Mellish, C., Sun, X.: The semantic web as a Linguistic resource: opportunities for natural language generation. Knowl.-Based Syst. **19**(5), 298–303 (2006)
16. Minock, M.: A phrasal approach to natural language interfaces over databases. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 333–336. Springer, Heidelberg (2005)
17. Minock, M.: C-Phrase: a system for building robust natural language interfaces to databases. Data Knowl. Eng. **69**(3), 290–302 (2010)
18. Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Thousand Oaks (1999)
19. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Natural Language Processing. Cambridge University Press, Cambridge (2000)
20. Schütte, N.: Generating natural language descriptions of ontology concepts. In: Proceedings of the 12th European Workshop on Natural Language Generation, ENLG 2009, Stroudsburg, PA, USA, pp. 106–109. Association for Computational Linguistics (2009)
21. Shekarpour, S., Auer, S., Ngonga Ngomo, A.-C., Gerber, D., Hellmann, S., Stadler, C.: Keyword-driven SPARQL query generation leveraging background knowledge. In: International Conference on Web Intelligence (2011)
22. Simitsis, A., Ioannidis, Y.E.: DBMSs Should Talk Back Too. CoRR, abs/0909.1786 (2009)
23. Stevens, R., Malone, J., Williams, S., Power, R.: Automating class definitions from OWL to English. In: Proceedings of Bio-Ontologies 2010: Semantic Applications in Life Sciences SIG at the 18th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB 2010), July 2010
24. Sun, X., Mellish, C.: Domain independent sentence generation from RDF representations for the semantic web. In: Callaway, C., Corradini, A., Kreutel, J., Moore, J., Stede, M. (eds.) Proceedings of Combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems (Part of ECAI 2006) (2006)

25. Sun, X., Mellish, C.: An experiment on "free generation" from single RDF triples. In: Proceedings of the Eleventh European Workshop on Natural Language Generation, ENLG 2007, Stroudsburg, PA, USA, pp. 105–108. Association for Computational Linguistics (2007)
26. Third, A., Williams, S., Power, R.: OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies. In: Proceedings of the 10th International Semantic Web Conference (ISWC 2011) (2011)
27. Tran, D.T., Wang, H., Haase, P.: Hermes: Data Web search on a pay-as-you-go integration infrastructure. J. Web Semant. **7**(3), 189–203 (2009)
28. Wilcock, G.: Talking OWLs: towards an ontology verbalizer. In: Human Language Technology for the Semantic Web and Web Services, ISWC 2003, Sanibel Island, Florida, pp. 109–112 (2003)
29. Wilcock, G., Jokinen, K.: Generating Responses and Explanations from RDF/XML and DAML+OIL (2003)
30. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., Weikum, G.: Deep answers for naturally asked questions on the web of data. In: Proceedings of the 21st International Conference Companion on World Wide Web, WWW 2012 Companion, pp. 445–449. ACM, New York (2012)

Multilingual Ontology Matching Evaluation – A First Report on Using MultiFarm

Christian Meilicke[1](✉), Cássia Trojahn[2], Ondřej Šváb-Zamazal[3], and Dominique Ritze[1]

[1] University of Mannheim, Mannheim, Germany
christian@informatik.uni-mannheim.de
[2] INRIA and LIG, Grenoble, France
[3] University of Economics, Prague, Czech Republic

Abstract. This paper reports on the first usage of the MultiFarm dataset for evaluating ontology matching systems. This dataset has been designed as a comprehensive benchmark for multilingual ontology matching. In a first set of experiments, we analyze how state-of-the-art matching systems – not particularly designed for the task of multilingual ontology matching – perform on this dataset. These experiments show the hardness of MultiFarm and result in baselines for any algorithm specifically designed for multilingual ontology matching. We continue with a second set of experiments, where we analyze three systems that have been extended with specific strategies to solve the multilingual matching problem. This paper allows us to draw relevant conclusions for both multilingual ontology matching and ontology matching evaluation in general.

1 Introduction

Ontology matching is the task of finding correspondences that link concepts, properties or instances between two ontologies. Different approaches have been proposed for performing this task. They can be classified along the ontology features that are taken into account (labels, structures, instances, semantics) or with regard to the kind of disciplines they belong to (e.g., statistics, combinatorial, semantics, linguistics, machine learning, or data analysis) [4,8,12].

With the aim of establishing a systematic evaluation of matching systems, the Ontology Alignment Evaluation Initiative (OAEI)[1] [3] has been carried out over the last years. It is an annual evaluation campaign that offers datasets, from different domains, organized by different groups of researchers. However, most of the OAEI datasets have been focused on monolingual tasks. A detailed definition on multilingual and cross-lingual ontology matching tasks can be found in [14]. The multilingual datasets so far available contain single pairs of languages, as the MLDirectory dataset,[2] which consists of website directories in English and Japanese; and the VLCR dataset,[3] that aims at matching the thesaurus of the

[1] http://oaei.ontologymatching.org/.
[2] http://oaei.ontologymatching.org/2008/mldirectory/.
[3] http://www.cs.vu.nl/~laurah/oaei/2009/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 132–147, 2015.
DOI: 10.1007/978-3-662-46641-4_10

Netherlands Institute for Sound and Vision, written in Dutch, to the WordNet and DBpedia, in English. Furthermore, these datasets contain only partial reference alignments or are not fully open. Thus, they are not suitable for an extensive evaluation.

For overcoming the lack of a comprehensive benchmark for multilingual ontology, the MultiFarm dataset has been designed. This dataset is based on the OntoFarm [16] dataset, which has been used successfully in OAEI in the Conference track. MultiFarm is composed of a set of seven ontologies translated in eight different languages and the complete corresponding alignments between these ontologies.

In this paper, we report on the first usage of MultiFarm for multilingual ontology matching evaluation. In [10], we have deeply discussed the design of MultiFarm, focusing on its multilingual features and the specificities of the translation process, with a very preliminary report on its evaluation. Here, we extend this preliminary evaluation and provide a deep discussion on the performance of matching systems. Our evaluation is based on a representative subset of MultiFarm and a set of state-of-the-art matching systems participating in OAEI campaigns. Most of these systems have not particularly been designed for matching ontologies described in different languages. This hold for those systems participating in OAEI 2011. For these systems we have omitted testcases in which Russian and Chinese languages were involved. We also included three participants of OAEI 2011.5 that use specific multilingual components. These systems use basic translation components that are executed prior to the matching process itself. Here we also included Russian and Chinese testcases. To our knowledge, such a comprehensive evaluation has not been conducted so far in the field of multilingual ontology matching.

The rest of the paper is organised as follows. In Sect. 2, we first introduce the OntoFarm dataset and then we present its multilingual counterpart. We shortly discuss the hardness of MultiFarm and present the results that have been gathered in previous OAEI campaigns on OntoFarm. In Sect. 3, we present the evaluation setting used to carry out our experiments and list the tools we have evaluated. In Sect. 4, we finally describe the results of our experiments. We mainly focus on highly aggregated results due to the enormous amount of generated data. In Sect. 5, we conclude the paper and discuss directions for future work.

2 Background on MultiFarm

The MultiFarm dataset has been thoroughly described in [10]. It is available at http://web.informatik.uni-mannheim.de/multifarm/. The dataset is the multilingual version of the OntoFarm dataset [16], which has been used in previous OAEI campaigns in the Conference track. In the following, we shortly describe the OntoFarm dataset, explain how MultiFarm has been constructed, and roughly report about evaluation results of the OAEI Conference track.

2.1 OntoFarm

The OntoFarm dataset is based on a set of 16 ontologies from conference organisation domain. All contained ontologies differ in numbers of classes, properties, and in their DL expressivity. They are very suitable for ontology matching tasks since they were independently designed by different people who used various kinds of resources for ontology design:

- actual conferences and their web pages,
- actual software tools for conference organisation support, and
- experience of people with personal participation in organisation of actual conferences

Thus, the OntoFarm dataset describes a quite realistic matching scenario and has been successfully applied in the OAEI within the Conference track since 2006. In 2008, a first version of the reference alignments was created and then annually enriched and updated up to current 21 reference alignments built between seven (out of 16) ontologies. Each of them has between four to 25 correspondences. The relatively small number of correspondences in the reference alignments is based on the fact that the reference alignments contain only simple equivalence correspondences. Due to different modeling styles of the ontologies, for many concepts and properties thus no equivalent counterparts exist. This makes the matching task harder, however, it is also a typical characteristics of other matching scenarios.

2.2 MultiFarm

For generating the MultiFarm dataset, those seven OntoFarm ontologies, for which reference alignments are available, were manually translated into eight different languages (Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish). Since native speakers with a certain knowledge about the ontologies translated them, we do not expect any serious errors but of course they can never be excluded at all. Based on these translations, it is possible to re-create cross-lingual variants of the original test cases from the OntoFarm dataset as well as to exploit the translations more directly. Thus, the MultiFarm dataset contains two types of cross-lingual reference alignments.

We have depicted a small subset of the dataset shown in Fig. 1. This figure indicates the cross-lingual reference alignments between different ontologies, derived from original alignments and translations (type (i)), and cross-lingual reference alignments between the same ontologies, which are directly based on the translations or on exploiting transitivity of translations (type (ii)). Reference alignments of type (i) cover only a small subset of all concepts and properties. We have explained this above for the original test cases of the OntoFarm dataset. In contrast, for test cases of type (ii) there are (translated) counterparts for each concept and property.

Overall, the MultiFarm dataset has 36 × 49 test cases. 36 is a number of pairs of languages – each English ontology has its 8 language variants. 49 is the

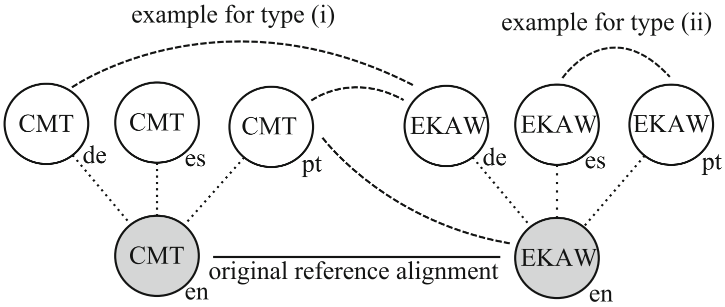


Fig. 1. Constructing MultiFarm from OntoFarm. Small subset that covers two ontologies and three translations. The solid line refers to a reference alignment of the OntoFarm dataset; dotted lines refer to translations; dashed lines refer to new cross-lingual reference alignments.

number of all reference alignments for each language pair. This is implied from the number of original reference alignments (21) which is doubled (42) due to the fact that there is a difference between CMT_{en}-$EKAW_{de}$ and CMT_{de}-$EKAW_{en}$ in comparison with the original test cases where the test cases CMT-$EKAW$ and $EKAW$-CMT are not distinguished. Additionally, we can also construct new reference alignments for matching each ontology on its translation which gives us seven additional reference alignments for each pair.

The main motivation for creating the MultiFarm dataset has been the ability to create a comprehensive set of test cases of type (i). We have especially argued in [10] that type (ii) test cases are not well suited for evaluating multilingual ontology matching systems, because they can be solved with very specific methods that are not related to the multilingual matching task.

2.3 Test Hardness

The OntoFarm dataset has a very heterogeneous character due to different modeling styles by various people. This leads to a high difficulty of the resulting test cases. For example, the object property `writtenBy` occurs in several OntoFarm ontologies. When only considering the labels, one would expect that a correspondence like `writtenBy` = `writtenBy` correctly describes that these object properties are equivalent. However, in ontology $\mathcal{O}_1$ the property indicates that a paper (domain) is written by an author (range), while in $\mathcal{O}_2$ the property describes that a review (domain) is written by a reviewer (range). Therefore, this correspondence is not contained in the reference alignment between $\mathcal{O}_1$ and $\mathcal{O}_2$. Similarly, comparing the English against the Spanish variant, there are the object properties `writtenBy` and `escrito por`. Pure translation would, similarly to the monolingual example, not result in detecting a correct correspondence. For that reason, the MultiFarm type (i) test cases go far beyond being a simple translation task.

The cross-lingual test cases of MultiFarm are probably much harder than the monolingual test cases of OntoFarm. Hence, it is important to know how matching systems perform on OntoFarm. These results can be understood as an upper bound that will be hard to top by results achieved for MultiFarm. In Fig. 2, we have depicted some results of previous OAEI campaigns in a precision/recall triangular graph. This graph shows precision, recall, and F-measure in a single plot. It includes the best (squares) and average (circles) results of the 2009, 2010, 2011 and 2011.5 Conference track as well as results of the three best ontology matching systems (triangles) from 2011.5. Best results are considered according to the highest F-measure which corresponds to exactly one ontology matching system for each year. In 2011, YAM++ achieved the highest F-measure that is why its cross sign overlaps with the light grey square depicting the best result of 2011.5. This matching system overcame 0.70 F-measure as a first system.

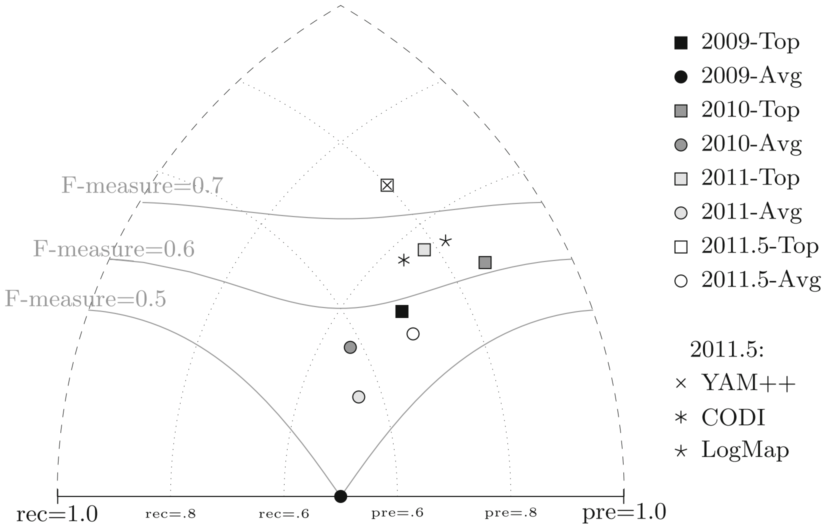


Fig. 2. Precision/recall triangular graph for the last four Conference tracks. Horizontal line depicts level of precision/recall while values of F-measure are depicted by areas bordered by corresponding lines F-measure = 0.[5|6|7].

On the one hand, Fig. 2 shows that there is an improvement every year, except the average results of 2011. Furthermore, average results of 2011.5 is almost the same as the results of the top matching system in 2009. A reason might be the availability of the complete dataset over several years. Since MultiFarm has not been used in the past, we expect that evaluation results also improve over the years. On the other hand, we can see that recall is not very high (0.63 in 2010, 0.60 in 2011 and 0.69 in 2011.5 for the best matching systems). This indicates that test cases of the OntoFarm dataset are especially difficult regarding recall measure.

3 Evaluation Settings

In the following, we explain how we executed our evaluation experiments and list the matching systems that have been evaluated.

3.1 Evaluation Workflow

Following a general definition, *matching* is the process that determines an *alignment* $\mathcal{A}$ for a pair of ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$. Besides the ontologies, there are other input parameters that are relevant for the matching process, namely: (*i*) the use of an input alignment $\mathcal{A}'$, which is to be extended or completed by the process; (*ii*) parameters that affect the matching process, for instance, weights and thresholds; and (*iii*) external resources used by the matching process, for instance, common knowledge and domain specific thesauri.

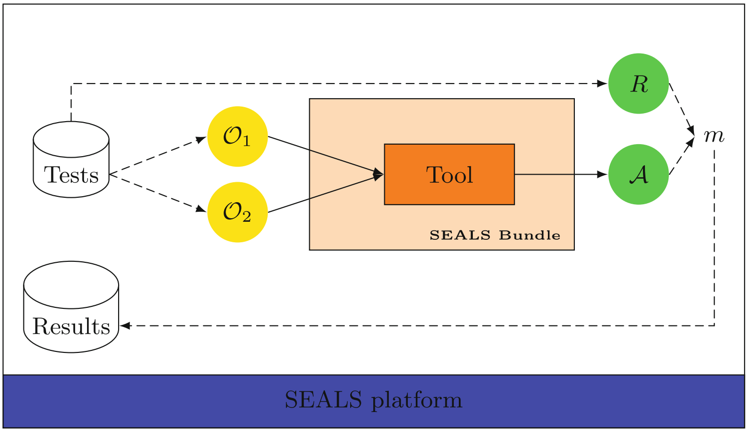


Fig. 3. Execution of tools.

In this paper, we focus on evaluating a standard matching task. (*i*) In most of our experiments, we do not modify the parameters that affect the matching process. For two systems, we made an exception from this rule and report very briefly on the results. (*ii*) We do not use an additional input alignment at all. Note that most systems do not support such a functionaility. (*iii*) We put no restriction on the external resources that are taken into account by the evaluated systems. Thus, we use the system standard settings for our evaluation. However, we obviously focus on the matching process where labels and annotations of $\mathcal{O}_1$ and $\mathcal{O}_2$ are described in different languages.

The most common way to evaluate the quality of a matching process is to evaluate the correctness (precision) and completeness (recall) of its outcome $\mathcal{A}$ by comparing $\mathcal{A}$ against a reference alignment $\mathcal{R}$. Since 2010, in the context of OAEI campaigns, the process of evaluating matching systems has been automated thanks to the SEALS platform (Fig. 3). For OAEI 2011 and OAEI 2011.5, participants have been invited to wrap their tools into a format that can be executed by the platform, i.e. the matching process is not conducted by the tool developer but by the organisers of an evaluation using the platform. For the purpose of this paper, we benefit from the large number of matching tools that become available for our evaluation. Furthermore, evaluation test cases are available in the SEALS repositories and can be used by everyone. Thus, all of our experiments can be completely reproduced.

3.2 Evaluated Matching Systems

As stated before, a large set of matching systems has already been uploaded to the platform in the context of OAEI 2011. We apply most of these tools to the MultiFarm dataset. In particular, we evaluated the tools AROMA [2], CIDER [5], CODI [6], CSA [15], LogMap and LogMapLt [7], MaasMatch [13], MapSSS [1], YAM++ [11] and Lily [17]. For most of these tools, we used the version submitted to OAEI 2011. However, some tool developers have already submitted a new version with some modifications between OAEI 2011 and OAEI 2011.5. This is the case for CODI, LogMap and MapSSS. Moreover, the developer of LogMap has additionally uploaded a lite version of their matching systems called LogMapLt.

We also included three OAEI 2011.5 participants: WeSeE and AUTOMSv2 [9] and the OAEI 2011.5 version of YAM++. WeSeE and YAM++ use Microsoft Bing to translate labels contained in the input ontologies to English. The translated English ontologies are then matched using standard matching procedures of WeSeE and YAM++. AUTOMSv2 re-uses a free Java API named WebTranslator to translate the ontologies to English. This process is performed before AUTOMSv2 profiling, configuration and matching methods are executed, so their input will consider only English-labeled copies of ontologies.

Since MultiFarm is based on the Conference dataset, we provide an overview table regarding performance of evaluated matching systems within last three editions of the Conference track, see Table 1. The last column (ML) of the table indicates whether systems uses multilingual components in the matching process. There have also been some systems participating in OAEI 2011 and OAEI 2011.5 that are not listed here. We have not added them to the evaluation for different reasons. Some of these systems cannot finish the MultiFarm matching process in less than several weeks while others generate empty alignments for nearly all matching tasks or terminate with an error. With respect to the OAEI 2011.5 participants, we have only added those systems that use mulitlingual techniques.

We have already explained that the MultiFarm data set is a comprehensive collection of testcases. For that reason we executed some of the tools in paralell on top of the SEALS platform. While systems as LogMap finished the MultiFarm dataset in less than 30 min, other systems required up to several days. However, reporting runtimes is beyond the scope of this paper.

Table 1. Performance of evaluated matching systems within last three editions of the Conference track (P = precision, R = recall, F = f-measure).

Matcher	OAEI 2010			OAEI 2011			OAEI 2011.5			ML
	P	F	R	P	F	R	P	F	R	
AROMA	.36	.42	.49	.35	.40	.46	*N/A*			
CIDER	*N/A*			.64	.53	.45	*N/A*			
CODI	.86	.62	.48	.74	.64	.57	.74	.64	.57	
CSA	*N/A*			.50	.55	.60	*N/A*			
LogMap	*N/A*			.84	.63	.50	.82	.66	.55	
LogMapLt	*N/A*			*N/A*			.73	.59	.50	
MaasMatch	*N/A*			.83	.56	.42	.74	.54	.42	
MapSSS	*N/A*			.55	.51	.47	.50	.50	.51	
YAM++	*N/A*			.78	.65	.56	.80	.74	.69	√
Lily	*N/A*			.36	.41	.47	*N/A*			
AUTOMSv2	*N/A*			*N/A*			.75	.52	.40	√
WeSeE	*N/A*			*N/A*			.67	.55	.46	√

4 Results

In the following, we discuss the results on different perspectives. First, we aggregate the results obtained for all pairs of test cases (and languages) in Sect. 4.1. Then we focus on different pairs of languages in Sect. 4.2. In both sections we report only on results for those systems that are not specifically designed for multilingual matching. In Sect. 4.3, we finally evaluate those three OAEI 2011.5 systems that use specific multilingual components.

4.1 Differences in Test Cases

As explained in Sect. 2, the dataset can be divided in (i) those test cases where the ontologies to be matched are translations of different ontologies and (ii) those test cases where the same original ontology has been translated into two different languages and the translated ontologies have to be matched. We display the results for test cases of type (i) on the left and those for type (ii) on the right of Table 2. We have ordered the systems according to the F-measure for the test cases of type (i). The best results, in terms of F-measure, are achieved by CIDER (18 %) followed by CODI (13 %), LogMap (11 %) and MapSSS (10 %). CIDER has both better precision and recall scores than any other system. Compared to the top-results that have been reported for the original Conference dataset (F-measure > 60 %), the test cases of the MultiFarm dataset are obviously much harder. However, an F-measure of 18 % is already a remarkable result given the fact that we executed CIDER in its default setting.

Table 2. Results aggregated per matching system.

Matcher	(i) Different ontologies				(ii) Same ontologies			
	Size	P	F	R	Size	P	F	R
CIDER	1433	.12	.18	.42	1090	.66	.12	.06
CODI	923	.08	.13	.43	7056	.77	.59*	.48
LogMap	826	.39	.11	.06	469	.71	.06	.03
MapSSS	2513	.16	.10	.08	6008	.97	.67*	.51
LogMapLt	826	.26	.07	.04	387	.56	.04	.02
MaasMatch	558	.24	.05	.03	290	.56	.03	.01
CSA	17923	.02	.03	.06	8348	.49	.42*	.36
YAM++$_{2011}$	7050	.02	.03	.03	4779	.22	.13*	.09
Aroma-	0	-	-	.00	207	.54	.02	.01
Lily	0	-	-	.00	11	1.00	.00	.00

The outcomes for test cases of type (ii) differ significantly. In particular, the results of MapSSS (67 % F-measure) are surprisingly compared to the results presented for test cases of type (i). This system can leverage the specifics of type (ii) test cases to cope with the problem of matching labels expressed in different languages. Similar to MapSSS, we also observe a higher F-measure for CODI, CSA, and YAM++. We have marked those systems with an asterisk. Note that all these systems have an F-measure of at least five times higher than the F-measure for test cases of type (i). For all other systems, we observe a slightly decreased F-measure comparing test cases of type (i) with type (ii).

Again, we have to highlight the differences between both types of test cases. Reference alignments of type (i) cover only a small fraction of all concepts and properties described in the ontologies. This is not the case for test cases of type (ii). Here, we have complete alignments that connect each concept and property with an equivalent counterpart in the other ontology. There seems to be a clear distinction between systems that are configured to generate complete alignments in the absence of (easy) usable label description, and other systems that focus on generating good results for test cases of type (i).

Comparing these results with the results for the OAEI 2011 Benchmark track, it turns out that all systems marked with an asterisk have been among the top five systems of this track. All Benchmark test cases have a similar property, namely, their reference alignments contain for each entity of the smaller ontology exactly one counterpart in the larger ontology. An explanation for this can be that these systems have been developed or at least configured to score well for the Benchmark track. For that reason, they generate good results for test cases of type (ii), while their results for test cases of type (i) are less good. MapSSS and CODI are an exception. These systems generate good results for both test cases of type (i) and (ii).

4.2 Differences in Languages

Besides aggregating the results per matcher, we have analysed the results per pair of languages (Table 3), for the case where different ontologies are matched (type (i) in Table 2). We have also compared the matchers with a simple edit distance algorithm on labels (edna).

Table 3. Results (F-measure) per pairs of languages for different ontologies.

pairs	edna	Aroma	CIDER	CODI	CSA	Lily	LogMap	LogLt	MaasMatch	MapSSS	YAM++$_{2011}$	average
cz-de	.01	-	.12	.11	.03	-	.09	.09	.02	.07	.03	.06
cz-en	.01	-	.20	.12	.04	-	.06	.04	.03	.08	-	.07
cz-es	.00	-	.14	.13	.02	-	.11	.11	-	.11	.03	.08
cz-fr	.01	-	.08	.01	.01	-	.01	.01	.01	.01	.03	.02
cz-nl	.00	-	.09	.09	.04	-	.04	.04	.04	.05	.03	.05
cz-pt	.00	-	.15	.15	.04	-	.13	.13	.02	.12	.04	.09
de-en	.01	-	.31	.22	.03	-	.22	.20	.20	.16	-	**.17**
de-es	.01	-	.25	.20	.02	-	.19	.06	-	.15	.03	**.11**
de-fr	.00	-	.18	.18	.01	-	.17	.04	.04	.13	.03	.09
de-nl	.01	-	.22	.08	.03	-	.05	.04	.04	.15	.03	.07
de-pt	.01	-	.10	.09	.03	-	.07	.07	.01	.06	.04	.05
en-es	.00	-	.25	.24	.03	-	.18	.04	.04	.18	-	**.12**
en-fr	.01	-	.20	.24	.03	-	.19	.04	.04	.13	-	**.11**
en-nl	.01	-	.22	.10	.04	-	.07	.10	.07	.15	-	**.10**
en-pt	.00	-	.15	.11	.06	-	.06	.06	.06	.07	-	.07
es-fr	.01	-	.29	.07	.02	-	.06	.01	.04	.06	.03	.07
es-nl	.01	-	.07	.01	.02	-	-	-	-	.01	.02	.02
es-pt	.01	-	.29	.26	.06	-	.27	.23	.09	.23	.03	**.16**
fr-nl	.01	-	.23	.14	.02	-	.13	.12	.13	.11	.03	**.10**
fr-pt	.00	-	.11	.06	.02	-	.06	-	.04	.02	.03	.04
nl-pt	.00	-	.02	.04	.03	-	.01	.01	.02	.02	.04	.02
average	.01		.17	.13	.03		.11	.08	.05	.10	.03	.08

With exception of Aroma and Lily, which are not able to deal with the complexity of the matching task, for most of the test cases no matcher has lower F-measure than edna. For some of them, however, LogMap, LogMapLt, MaasMatch and YAM++, respectively, have not provided any alignment. YAM++ has a specific behaviour and is not able to match the English ontologies to any other languages. For the other matchers, it (incidentally) happens mostly for the pairs of languages that do not share the same root language (e.g. es-nl or de-es). The exception is LogMapLt, which is not able to identify any correspondence between fr-pt, even if these languages have the same root language (e.g. Latin) and thus have a similar vocabulary. It could be expected that matchers should be able to find a higher number of correspondences for the pairs of languages where there is an overlap in their vocabularies because most of the matcher apply some label similarity strategy. However, it is not exactly the case in MultiFarm. The dataset contains many complex correspondences that cannot be found by a single translation process or by string comparison. This can be partially corroborated by the very low performance of edna in all test cases.

Looking at the results for each pair of languages, per matcher, the best five F-measures are obtained for de-en (31 %), es-fr/es-pt (29 %), de-es/en-es (25 %), all for CIDER, en-es/en-fr (24 %), for CODI, and fr-nl (23 %) again for CIDER. We could observe that 3 ahead pairs contain languages with some degree of overlap in their vocabularies (i.e., de-en, es-fr, es-pt). For each individual matcher, seven out of eight matchers have their best scores for these pairs (exception is YAM++ that scores better for cz-pt and de-pt), with worst scores in cz-fr, es-nl, which have very different vocabularies.

When aggregating the results per pair of languages, that order is mostly preserved (highly affected by CIDER): de-en (17 %), es-pt (16 %), en-es (12 %), de-es/en-fr (11 %), followed by fr-nl/en-nl (10 %). The exception is for the pair es-fr, where the aggregated F-measure decreases to 7 %. Again, the worst scores are obtained for cz-fr, nl-pt and es-nl. We can observe that, for most of the cases, the features of the languages (i.e., their overlapping vocabularies) have an impact in the matching results. However, there is no universal pattern and we have cases with similar languages where systems score very low (fr-pt, for instance). This has to be further analysed looking at the individual pairs of ontologies.

4.3 Translation Based Techniques

We have also analysed three matching systems (YAM++, AUTOMSv2 and WeSeE), participating in OAEI 2011.5, which first translate both source ontologies into English. The results, per pair of languages where different ontologies are matched (type (i) in Table 2), are reported in Table 4. These three matching systems clearly outperform all the other systems which do not use any specific method to deal with multilingual ontologies (cf. Table 3). Looking at the average results, the best results are achieved by YAM++ (0.41) followed by AUTOMSv2 (0.36) and WeSeE (0.27). However while YAM++ and WeSeE managed to match all eight different languages in the MultiFarm dataset, AUTOMSv2 did not manage to match ontologies in Chinese, Czech and Russian languages.

Looking at the results for each pair of languages, per matcher, the best five F-measures are obtained for en-fr (61 %), cz-en (58 %), cz-fr (57 %), en-pt (56 %), and en-nl/cz-pt/fr-pt (55 %). We can observe a positive effect of the translation step, since (besides the differences in the structure of the source ontologies) most of these language pairs do not have overlapping vocabularies (cz-pt or cz-fr, for instance). Furthermore, results with the same translator can differ depending on further matching system's components as demonstrated by YAM++ and WeSeE. YAM++ outperforms WeSeE for most of the pairs. When looking at the average of these three systems, we have the following pairs ranking: en-fr (47 %), en-pt

Table 4. Performance of the three OAEI 2011.5 systems that implemented specific multilingual methods.

Matcher/Pair	YAM++			AUTOMSv2			WeSeE			Average		
	P	F	R	P	F	R	P	F	R	P	F	R
cn-cz	.44	.32	.25	-	-	-	.14	.18	.24	.29	.25	.24
cn-de	.4	.32	.27	-	-	-	.11	.15	.21	.26	.23	.24
cn-en	.47	.38	.32	-	-	-	.41	.29	.22	.44	.33	.27
cn-es	.58	.2	.12	-	-	-	.1	.13	.2	.34	.17	.16
cn-fr	.43	.35	.3	-	-	-	.14	.17	.23	.28	.26	.26
cn-nl	.42	.35	.29	-	-	-	.09	.11	.17	.25	.23	.23
cn-pt	.34	.27	.22	-	-	-	.09	.12	.18	.22	.2	.2
cn-ru	.4	.3	.24	-	-	-	.09	.12	.17	.24	.21	.21
cz-de	.51	.47	.45	-	-	-	.16	.21	.3	.33	.34	.37
cz-en	.6	.58	.57	-	-	-	.51	.45	.4	.55	.51	.48
cz-es	.58	.2	.12	-	-	-	.21	.28	.43	.39	.24	.28
cz-fr	.58	.57	.57	-	-	-	.2	.26	.37	.39	.42	.47
cz-nl	.54	.53	.53	-	-	-	.16	.21	.3	.35	.37	.41
cz-pt	.54	.55	.56	-	-	-	.18	.24	.34	.36	.39	.45
cz-ru	.55	.55	.55	-	-	-	.2	.25	.35	.37	.4	.45
de-en	.55	.47	.42	.8	.39	.26	.52	.44	.37	.62	.43	.35
de-es	.58	.2	.12	.73	.37	.24	.16	.21	.33	.49	.26	.23
de-fr	.5	.46	.43	.86	.32	.2	.19	.24	.33	.51	.34	.32
de-nl	.5	.45	.41	.78	.39	.26	.18	.24	.35	.48	.36	.34
de-pt	.45	.39	.34	.82	.35	.22	.2	.26	.37	.49	.33	.31
de-ru	.56	.51	.47	-	-	-	.15	.19	.26	.36	.35	.37
en-es	.57	.21	.13	.61	.42	.32	.56	.5	.46	.58	.38	.3
en-fr	.6	.61	.62	.54	.3	.21	.58	.51	.45	.57	.47	.43
en-nl	.57	.55	.52	.6	.34	.24	.53	.45	.4	.57	.45	.38
en-pt	.58	.56	.54	.61	.37	.27	.53	.47	.43	.57	.47	.41
en-ru	.6	.55	.5	-	-	-	.58	.49	.43	.59	.52	.46
es-fr	.54	.2	.12	.62	.37	.26	.27	.35	.5	.47	.3	.3
es-nl	.48	.16	.1	.49	.35	.27	.16	.22	.37	.38	.24	.24
es-pt	.61	.25	.15	.54	.44	.37	.22	.3	.47	.46	.33	.33
es-ru	.55	.21	.13	-	-	-	.15	.21	.33	.35	.21	.23
fr-nl	.54	.53	.51	.54	.27	.18	.2	.26	.38	.43	.35	.36
fr-pt	.55	.55	.55	.63	.35	.24	.24	.31	.44	.47	.4	.41
fr-ru	.56	.52	.5	-	-	-	.19	.24	.34	.37	.38	.42
nl-pt	.52	.49	.46	.61	.38	.28	.17	.23	.35	.43	.36	.36
nl-ru	.56	.52	.5	-	-	-	.16	.22	.33	.36	.37	.41
pt-ru	.54	.53	.52	-	-	-	.16	.21	.3	.35	.37	.41
Average	**.52**	**.41**	**.37**	**.65**	**.36**	**.25**	**.25**	**.27**	**.34**	**.42**	**.34**	**.33**

(47 %), en-nl (45 %), de-en (43 %) and fr-pt (40 %). We can observe that for these pairs, the English language is in most of the pairs. It is somehow expected because these matchers use English as the pivot language in the translation process and the pure translation results are less penalised with regards to the lack of complementary strategies, such as translation disambiguation.

5 Discussion

Some of the reported results are relevant for multilingual ontology matching in general, while others help us to understand the characteristics of the MultiFarm dataset. The latter ones are relevant for any further evaluation that builds on the dataset. Moreover, we can also draw some conclusions that might be important for the use of datasets in the general context of ontology matching evaluation.

Exploiting structural information. Very good results for test cases of type (ii) can be achieved by methods non-specific to multilingual ontology matching. The result of MapSSS is an interesting example. This was also one of the main reasons why the MultiFarm dataset has been constructed as a comprehensive collection for test cases of type (i) and (ii). We suggest to put a stronger focus on test cases of type (i) in the context of evaluating multilingual ontology matching techniques. Otherwise, it remains unclear whether the measured results are based on multilingual techniques or on exploiting that the matched ontologies can interpreted as versions of the same ontology.

Finding a good configuration. Our results show that state-of-the-art matching systems are not very well suited for the tasks of matching ontologies described in different languages, especially when executed in their default setting. We started another set of experiments by running some tools (CODI, LogMap, Lily) in a manually configured setting better suited for the matching task. A first glimpse, the results shows that it is possible to increase the average F-measure up to a value of 26 %. Thus, we are planning to further investigate the influence of configurations on multilingual matching tasks within more extensive experiments.[4]

The role of language features. We cannot neglect certain language features (like their overlapping vocabularies) in the matching process. Once most of the matchers take advantage of label similarities it is likely that it may be harder to find correspondences between Czech and Portuguese ontologies than Spanish and Portuguese ones. In our evaluation, for most of the systems, the better performance where incidentally observed for the pairs of languages that have some degree of overlap in their vocabularies. This is somehow expected, however, we could find exceptions to this behavior. In fact, MultiFarm requires systems exploiting more sophisticated matching strategies than label similarity and

[4] We would like to thank Ernesto Jimenez-Ruiz (LogMap [7]) and Peng Wang (Lily [17]) for supporting us with a quick hint about a good, manually modified configuration for running their systems on MultiFarm.

for many ontologies in MultiFarm it is the case. To some extent we exploited automatic translation strategies by evaluating results from systems exploiting translations of ontologies into English language.

Test Difficulty. We can give the following simplified conclusion related to test difficulty. For the conference track top systems generate results with an average F-measure of 0.6 to 0.7 with better precision and worse recall. State of the art matching systems, without multilingual component, generate in their default setting in average an F-measure between 0 and 0.2 for the MultiFarm testcases. Using a well-chosen configuration, this value can increase up to 0.25 (based on very low recall values). A system that uses a preceding translation step, can achieve an F-measure between 0.3 and 0.4. These results are still based on slightly higher precision scores, however, the differences between precision and recall are less significant.

Implications on analyzing OAEI results. Aside from the topic of multilingual ontology matching, the results implicitly emphasise the different characteristics of test cases of type (i) and (ii). An example can be found when comparing results for the OAEI Benchmark and Conference track. The Benchmark track is about matching different versions (some slightly modified, some heavily modified) of the same ontology. The Conference dataset is about matching different ontologies describing the same domain. This difference finds its counterparts in the distinction between type (i) and type (ii) ontologies in the MultiFarm dataset. Without taking this distinction into account, it is not easy to draw valid conclusions on the generality of measured results.

6 Future Work

Even though we reported about diverse aspects, we could not analyse or evaluate all interesting issues. The following listing shows possible extensions and improvements for further evaluations based on MultiFarm:

- Executing matching systems with a specifically tailored configuration;
- Exploiting other approaches than pure translation strategies (disambiguation, use of multilingual lexicons, multilingual comparable corpora) and evaluate their impact on the matching process;
- Analysing the role of diacritics: in some languages, the same word written with or without accent can have a different meaning, e.g., in French 'où' (where) is different from 'ou' (or);
- Exploiting ontology population strategies for creating MultiFarm instances and take advantage of instance-level matching approaches; and evaluate how these approaches can help in the multilingual matching process.

Although we have many different ways to improve the multilingual matching task, we have shown that the MultiFarm dataset is a useful, comprehensive, and a difficult dataset for evaluating ontology matching systems. We strongly recommend to apply this resource and to compare measured results against the results presented in this paper. In particular, we encourage developers of ontology matching systems, specifically designed to match ontologies described in different languages, to make use of the dataset and to report about achieved results.

Acknowledgements. Some of the authors are partially supported by the SEALS project (IST-2009-238975). Ondřej Šváb-Zamazal has been partially supported by the CSF grant no. P202/10/0761.

References

1. Cheatham, M.: MapSSS results for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 184–189 (2011)
2. David, J.: Aroma results for OAEI 2009. In: Proceedings of the 4th ISWC Workshop on Ontology Matching (OM), pp. 147–152 (2009)
3. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Ontology alignment evaluation initiative: six years of experience. In: Spaccapietra, S. (ed.) Journal on Data Semantics XV. LNCS, vol. 6720, pp. 158–192. Springer, Heidelberg (2011)
4. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
5. Gracia, J., Bernad, J., Mena, E.: Ontology matching with CIDER: evaluation report for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 126–133 (2011)
6. Huber, J., Sztyler, T., Noessner, J., Meilicke, C.: Codi: Combinatorial optimization for data integration: results for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 134–141 (2011)
7. Jimenez-Ruiz, E., Morant, A., Grau, B.C.: LogMap results for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 163–170 (2011)
8. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowl. Eng. Rev. **18**(1), 1–31 (2003)
9. Kotis, K., Katasonov, A., Leino, J.: Aligning smart and control entities in the IoT. In: Andreev, S., Balandin, S., Koucheryavy, Y. (eds.) NEW2AN/ruSMART 2012. LNCS, vol. 7469, pp. 39–50. Springer, Heidelberg (2012)
10. Meilicke, C., Castro, R.G., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Tamilin, A., Trojahn, C., Wang, S.: Multifarm: A benchmark for multilingual ontology matching. J. Web Semant. **2**(1), 3–10 (2011)
11. Ngo, D., Bellasene, Z., Coletta, R.: YAM++ - results for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 228–235 (2011)
12. Rahm, E., Bernstein, P.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001)
13. Schadd, F., Roos, N.: Maasmatch results for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 171–178 (2011)

14. Spohr, D., Hollink, L., Cimiano, P.: A machine learning approach to multilingual and cross-lingual ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 665–680. Springer, Heidelberg (2011)
15. Tran, Q.-V., Ichise, R., Ho, B.-Q.: Cluster-based similarity aggregation for ontology matching. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 142–147 (2011)
16. Šváb, O., Svátek, V., Berka, P., Rak, D., Tomášek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. In: Poster Track of ISWC 2005 (2005)
17. Wang, P.: Lily results on SEALS platform for OAEI 2011. In: Proceedings of the 6th ISWC Workshop on Ontology Matching (OM), pp. 156–162 (2011)

Evaluating Semantic Search Systems to Identify Future Directions of Research

Khadija Elbedweihy[1](✉), Stuart N. Wrigley[1], Fabio Ciravegna[1],
Dorothee Reinhard[2], and Abraham Bernstein[2]

[1] University of Sheffield, Regent Court, 211 Portobello, Sheffield, UK
{k.elbedweihy,s.wrigley,f.ciravegna}@dcs.shef.ac.uk
[2] University of Zürich, Binzmühlestrasse 14, 8050 Zürich, Switzerland
{dreinhard,bernstein}@ifi.uzh.ch

Abstract. Recent work on searching the Semantic Web has yielded a wide range of approaches with respect to the style of input, the underlying search mechanisms and the manner in which results are presented. Each approach has an impact upon the quality of the information retrieved and the user's experience of the search process. This highlights the need for formalised and consistent evaluation to benchmark the coverage, applicability and usability of existing tools and provide indications of future directions for advancement of the state-of-the-art. In this paper, we describe a comprehensive evaluation methodology which addresses both the underlying performance and the subjective usability of a tool. We present the key outcomes of a recently completed international evaluation campaign which adopted this approach and thus identify a number of new requirements for semantic search tools from both the perspective of the underlying technology as well as the user experience.

1 Introduction and Related Work

State-of-the-art semantic search approaches are characterised by their high level of diversity both in their features as well as their capabilities. Such approaches employ different styles for accepting the user query (e.g., forms, graphs, keywords) and apply a range of different strategies during processing and execution of the queries. They also differ in the format and content of the results presented to the user. All of these factors influence the user's perceived performance and usability of the tool. This highlights the need for a formalised and consistent evaluation which is capable of dealing with this diversity. It is essential that we do not forget that searching is a user-centric process and that the evaluation mechanism should capture the usability of a particular approach.

One of the very first evaluation efforts in the field was conducted by Kaufmann to compare four different query interfaces [1]. Three were based on natural language input (with one employing a restricted query formulation grammar); the fourth employed a formal query approach which was hidden from the

This work was supported by the European Union 7th FP ICT based e-Infrastructures Project SEALS (Semantic Evaluation at Large Scale, FP7-238975).

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 148–162, 2015.
DOI: 10.1007/978-3-662-46641-4_11

end user by a graphical query interface. Recently, evaluating semantic search approaches gained more attention both in IR – within its most established evaluation conference TREC – [2] as well as in the Semantic Web community (SemSearch [3] and QALD[1] challenges).

The above evaluations are all based upon the Cranfield methodology [4][2]: using a test collection, a set of tasks and a set of relevance judgments. This leaves aside aspects of user-oriented evaluations concerned with the usability of the evaluated systems and the user experience which is as important as assessing the performance of the systems. Additionally, the above attempts are separate efforts lacking standardised evaluation approaches and measures. Indeed, Halpin et al. [3] note that "the lack of standardised evaluation has become a serious bottleneck to further progress in this field".

The first part of this paper describes an evaluation methodology for assessing and comparing the strengths and weaknesses of user-focussed Semantic Search approaches. We describe the dataset and questions used in the evaluation and discuss the results of the usability study. The analysis and feedback from this evaluation are described. The second part of the paper identifies a number of new requirements for search approaches based upon the outcomes of the evaluation and analysis of the current state-of-the-art.

2 Evaluation Design

In the Semantic Web community, semantic search is widely used to refer to a number of different categories of systems:

- *gateways* (e.g., Sindice [5] and Watson [6]) locating ontologies and documents
- approaches reasoning over data and information located within documents and ontologies (PowerAqua [7] and FREyA [8])
- view-based interfaces allowing users to explore the search space while formulating their queries (Semantic Crystal [9], K-Search [10] and Smeagol [11])
- mashups integrating data from different sources to provide rich descriptions about Semantic Web objects (Sig.ma [12]).

The evaluation described here focuses on user-centric semantic search tools (e.g. query given as keywords or natural language or using a form or a graph) querying a repository of semantic data and returning answers extracted from them. The tools' results presentation is not limited to a specific style (e.g., list of entity URIs or a visualisation of the results). However, the results returned must be answers rather than documents matching the given query.

Search is a user-centric activity; therefore, it is important to emphasise the users' experience. An important aspect of this is the formal gathering of feedback from the participants which should be achieved using standard questionnaires.

[1] http://www.sc.cit-ec.uni-bielefeld.de/qald-1.

[2] http://www.sigir.org/museum/pdfs/ASLIB%20CRANFIELD%20RESEARCH%20PROJECT-1960/pdfs/.

Furthermore, the use of an additional demographics questionnaire allows more in-depth findings to be identified (e.g., if a particular type of user prefers a particular search approach).

2.1 Datasets and Questions

Subjects are asked to reformulate a set of questions using a tool's interface. Thus, it is important that the data set would be from an understandable and well-known domain (and hence, easily understandable by non-expert users) and, preferably, already have a set of questions and associated groundtruths. The geographical dataset from the Mooney Natural Language Learning Data[3] satisfies these requirements and has been used in a number of usability studies [1,8]. Although the Mooney dataset is different from ones currently found on the Semantic Web such as DBpedia in terms of size, heterogeneity and quality, the assessment of the tools ability to handle these aspects is not the focus of this phase but rather the usability of the tools and the user experience.

The questions [13] used in the first evaluation campaign were generated based on the existing templates within the Mooney dataset. These contained questions with varying complexity and assessing different features. For instance, they contained simple with only 1 unknown concept such as "Give me all the capitals of the USA?" and comparative questions such as "Which rivers in Arkansas are longer than Aleghany river" as well as negation questions such as "Tell me which river do not traverse the state with capital nashville".

2.2 Criteria and Analyses

Usability. Different input styles (e.g., form-based, NL, etc.) can be compared with respect to the input query language's expressiveness and usability. These concepts are assessed by capturing feedback regarding the user experience and the usefulness of the query language in supporting users to express their information needs and formulate searches [14]. Additionally, the expressive power of a query language specifies what queries a user is able to pose [15]. The usability is further assessed with respect to results presentation and suitability of the returned answers (data) to the casual users as perceived by them. The datasets and associated questions were designed to fully investigate these issues.

Performance. Users are familiar with the performance of commercial search engines (e.g., Google) in which results are returned within fractions of a second; therefore, it is a core criterion to measure the tool's performance with respect to the speed of execution.

Analyses. The experiment was controlled using custom-written software which allowed each experiment run to be orchestrated and timings and results to be captured. The results included the actual result set returned by a tool for a

[3] http://www.cs.utexas.edu/users/ml/nldata.html.

query, the time required to execute a query, the number of attempts required by a user to obtain a satisfactory answer as well as the time required to formulate the query. We used post-search questionnaires to collect data regarding the user experience and satisfaction with the tool. Three different types of online questionnaires were used which serve different purposes. The first is the System Usability Scale (SUS) questionnaire [16]. The test consists of ten normalized questions and covers a variety of usability aspects, such as the need for support, training, and complexity and has proven to be very useful when investigating interface usability [17]. We developed a second, extended, questionnaire which includes further questions regarding the satisfaction of the users. This encompasses the design of the tool, the input query language, the tool's feedback, and the user's emotional state during the work with the tool. An example of a question used is *'The query language was easy to understand and use'* with answers represented on a scale from 'disagree' to 'agree'. Finally, a demographics questionnaire collected information regarding the participants.

3 Evaluation Execution and Results

The evaluation consisted of tools from form-based, controlled-NL-based and free-NL-based approaches. Each tool was evaluated with 10 subjects (except K-Search [10] which had 8) totalling 38 subjects (26 males, 12 females) aged between 20 and 35 years old. They consisted of 28 students and 10 researchers drawn from the University population. Subjects rated their knowledge of the Semantic Web with 6 reporting their knowledge to be advanced, 5 good, 9 average, 10 little and 8 having no experience. In addition, their knowledge of query languages was recorded, with 5 stating their knowledge to be advanced, 12 good, 8 average, 6 little and 7 having no experience.

Firstly, the subjects were presented with a short introduction to the experiment itself such as why the experiment is taking place, what is being tested, how the experiment will be executed, etc. Then the tool itself was explained to the subjects; they learnt about the type and the functionality of the tool and how to apply it's specific query language to answer the given tasks. The users were then given sample tasks to test their understanding of the previous phases. After that, the subjects did the actual experiment: using the tool's interface to formulate each question and get the answers. Having finished all the questions, they were presented with the three questionnaires (Sect. 3). Finally, the subjects had the chance to talk about important and open questions and give more feedback and input to their satisfaction or problems with the system being tested.

Table 1 shows the results for the four tools participating in this phase. The *mean number of attempts* shows how many times the user had to reformulate their query in order to obtain answers with which they were satisfied (or indicated that they were confident a suitable answer could not be found). This latter distinction between finding the appropriate answer and the user 'giving up' after a number of attempts is shown by the *mean answer found rate*. *Input time* refers to the amount of time the subject spent formulating their query using the tool's interface, which acts as a core indicator of the tool's usability.

Table 1. Evaluation results showing the tools performance. Rows refer to particular metrics.

Criterion	K-Search *Form-based*	Ginseng *Controlled NL-based*	NLP-Reduce *NL-based*	PowerAqua *NL-based*
Mean experiment time (s)	4313.84	3612.12	4798.58	2003.9
Mean SUS (%)	44.38	40	25.94	72.25
Mean ext. questionnaire (%)	47.29	45	44.63	80.67
Mean number of attempts	2.37	2.03	5.54	2.01
Mean answer found rate	0.41	0.19	0.21	0.55
Mean execution time (s)	0.44	0.51	0.51	11
Mean input time (s)	69.11	81.63	29	16.03

According to the ratings of SUS scores [18], none of the four participating tools fell in either the best or worst category. Only one of the tools (PowerAqua [7]) had a ‘Good’ rating with a SUS score of 72.25, other two tools (Ginseng [19] and K-Search [10]) fell in the ‘Poor’ rating while the last one (NLP-Reduce [20]) was classified as ‘Awful’. The results of the questionnaires were confirmed by the recorded usability measures. Subjects using the tool with the lowest SUS score (NLP-Reduce) required more than twice the number of attempts of the other tools before they were satisfied with the answer or moved on. Similarly, subjects using the two tools with the highest SUS and extended scores (PowerAqua and K-Search) found satisfying answers to their queries twice the times as for the other tools. Altogether, this confirms the reliability of the results and the feedback of the users and also the conclusions based on them.

4 Usability Feedback and Analysis

This section discusses the results and feedback collected from the subjects of the usability study. Figure 1 summarises the features most liked and disliked based on their feedback. The following discussion stems from this summary.

4.1 Input Style

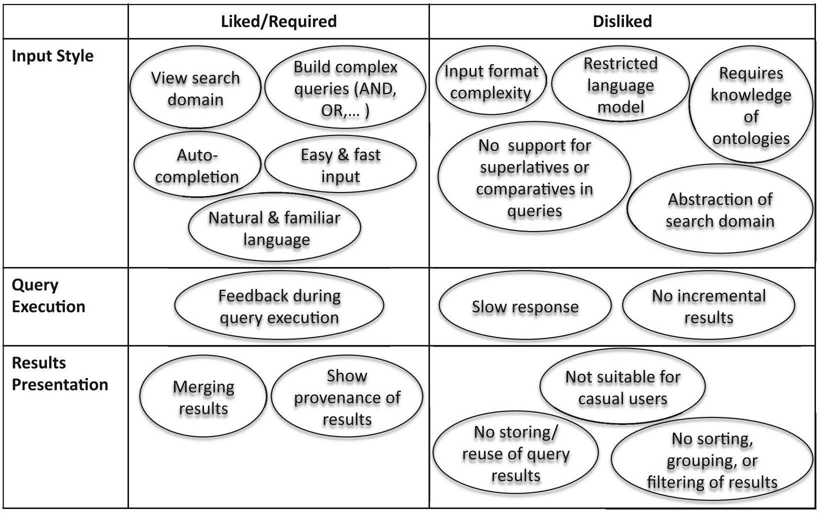
On the one hand, Uren et al. [14] state that forms can be helpful to explore the search space when it is unknown to the users. Additionally, Corese [21] – which uses a form-based interface to allow users to build their queries – received very positive comments from its users among which was an appreciation for its form-based interface. On the other hand, Lei et al. [22] see this exploration as a burden on users that requires them to be (or become) familiar with the underlying ontology and semantic data. The results of our evaluation and the feedback from the users support both arguments: positive comments for the form-based tool (K-Search) included ones such as “I liked to see the concepts and relations between them, it helped in knowing what sort of information is available to be retrieved from the system”. On the other hand, negative comments included ones such as “For me, it was complex to build some queries” and “It was hard to understand without explanation and it seemed less intuitive than NL-based tools”.

Additionally, we found that form-based interfaces allow users to build more complex queries than the natural language interfaces. However, building queries by exploring the search space is usually time consuming especially as the ontology gets larger or the query gets more complex. This was shown by Kaufmann et al. [1] in their usability study which found that users spent the most time when working with the graph-based system *Semantic Crystal*. Our evaluation supports this general conclusion: subjects using the form-based approach took between two to three times the time taken by users of natural language approaches. Also, feedback showed that most of the users found query formulation with the form-based tool (K-Search) to be laborious and requiring long time especially when they compared it to the NL-based tools (NLP-Reduce and PowerAqua). However, our analysis suggests a more nuanced behaviour. While freeform natural language interfaces are generally faster in terms of query formulation, we found this did not hold for approaches employing a very restricted language model. For instance, query formulation took longer using Ginseng (restricted natural language) than K-Search (form-based). This is further supported by users feedback: the most repeated positive comment for the free-NL-based tools was “It is quick and easy to use”. On the other hand, the more time required to formulate queries in Ginseng was due to its restrictive model which limited users expressivity and affected their satisfaction. Some of the most negative repeated comments for Ginseng included:

- It was unclear how individual terms suggested by the tool related to particular classes or relations.
- In most of the queries, I got stuck and could no longer complete the query in the way I wanted because it was restricted.
- It was very annoying and frustrating.

Kaufmann et al. [1] also showed that a natural language interface was judged by users to be the most useful and best liked. Their conclusion, that this was because users can communicate their information needs far more effectively when using a familiar and natural input style, is supported by our findings. The same study found that people can express more semantics when they use full sentences as opposed to simply keywords. Similarly, Demidova et al. [23] state that natural language queries offer users more expressivity to describe their information needs than keywords – a finding also confirmed by the user feedback from our study.

However, natural language approaches suffer from both syntactic as well as semantic ambiguities. This makes the overall performance of such approaches heavily dependent upon the performance of the underlying natural language processing techniques responsible for parsing and analysing the users’ natural



	Liked/Required	Disliked
Input Style	View search domain Build complex queries (AND, OR,…) Auto-completion Easy & fast input Natural & familiar language	Input format complexity Restricted language model Requires knowledge of ontologies No support for superlatives or comparatives in queries Abstraction of search domain
Query Execution	Feedback during query execution	Slow response No incremental results
Results Presentation	Merging results Show provenance of results	Not suitable for casual users No storing/ reuse of query results No sorting, grouping, or filtering of results

Fig. 1. Summary of evaluation feedback: features most liked and disliked by users categorised with respect to query format, query execution, and results presentation.

language sentences. This was shown by the feedback we received from users of the NL-based tool (NLP-Reduce), one of which was "the response is very dependent on the use of the correct terms in the query". This was also confirmed by that approach achieving the lowest precision. Another limitation faced by the natural language approach is the lack of knowledge of the underlying ontology terms and relations by the users due to the high abstraction of the search domain. The effect of this is that any keywords or terms used by users are likely to be very different from the semantically-corresponding terms in the ontology. This in turn increases the difficulty of parsing the user query and affects the performance.

Using a restricted grammar as employed by Ginseng is an approach to limit the impact of both of these problems. The 'autocompletion' provided by the system based on the underlying grammar attempts to bridge the domain abstraction gap and also resembles the form-based approach in helping the user to better understand the search space. Although it provides the user with knowledge regarding which concepts, relations and instances are found in the search space and hence can be used to build valid queries, it still lacks the power of visualising the structure of the used ontology. The impact of this 'intermediate' functionality can be observed in the users feedback with a lower degree of dissatisfaction regarding the ability to conceptualise the underlying data but still not completely eliminated. For instance, the positive comment "The autocompletion and suggestions was helpful to know the underlying data" was given by some of the users for Ginseng. The restricted language model also prevents unacceptable/invalid queries in the used grammar by employing a guided input natural language approach. However, only accepting specific concepts and relations – found in the grammar – limits the flexibility and expressiveness of the user queries. User coercion into following predefined sentence structures proves to be frustrating and too complicated [1,24]. Again, this was supported by the feedback showing that users were often annoyed by this restriction especially when they got stuck and did not know how to continue the query formulation.

The feedback from the questionnaires showed that using superlatives or comparatives in the user queries (e.g.: highest point, longer than) was not supported by any of the participating tools; an issue raised by 8 subjects in the answer of the SUS question "What didn't you like about the system and why?" and by others in the open feedback after the experiment. Only one provided a feature similar to this functionality: the ability to specify a range of values for numeric datatypes. A comparative such as *less than 5000* could then be translated to the range *0 to 5000*. However, this was deemed to be both confusing (since the user had to decide what to use as the non-specified bound) and, when the non-specified bounds were incorrect, having a negative impact on the results.

4.2 Query Execution and Response Time

Speed of response is an important factor for users since they are used to the performance of commercial search engines (e.g., Google) in which results are returned within fractions of a second. Many users in our study were expecting similar performance from the semantic search tools. Although the average response time of three of the tools (K-Search, NLP-Reduce, Ginseng) is less than a second (44 ms, 51 ms, and 51 ms respectively), users reported their dissatisfaction with these timings especially the ones who evaluated PowerAqua with response time of 11 s on average. The lack of feedback on the status of the execution process only served to increase the sense of dissatisfaction: no tool indicated the execution progress or whether a problem had occurred in the system. This lack of feedback resulted in users suspecting that something had gone wrong with the system – even if the search was still progressing– and start a new search. Furthermore, some tools made it impossible to distinguish between an empty result set, a problem with the query formulation or a problem with the search. This not only affected the users experience and satisfaction but also the approach's measured performance. The following list includes some of the most repeated comments given by the users for all the tools in this context:

- With some queries, I had no idea whether the search was in progress or failed since there was no feedback about it.
- When the response was delayed, I suspected that an error occurred and I restarted the search process.
- It was confusing when I got back no results and I didn't know whether this was due to an error in the system or my query, or there was actually no results for my question.

4.3 Results Presentation

Semantic Search tools are different from Semantic Web gateways or entry points such as Watson and Sindice. The latter are not intended for casual users but for other applications or the Semantic Web community to locate Semantic Web resources such as ontologies or Semantic Web documents and are usually presented as a set of URIs. For example, Sindice shows the URIs of documents and, for every document, it additionally presents the triples contained within the document, an RDF graph of the triples, and the used ontologies. Semantic Search tools are, on the other hand, used by casual users (i.e., users who may be experts in the domain of the underlying data but may have no knowledge of semantic technologies). Such users usually have different requirements and expectations of *what* and *how* results should be presented to them.

In contrast to these 'casual user' requirements, a number of the search tools did not present their results in a user-friendly manner and this was reflected in the feedback. Two approaches presented the full URIs together with the concepts in the ontology that were matched with the terms in the user query. Another used the instance labels to provide a natural language presentation; however, such labels (e.g., 'montgomeryAl') were not necessarily suitable for direct inclusion into a natural language phrase. Indeed, the tool also displayed the ontologies used as well as the mappings that were found between the ontology and the query terms. Although potentially useful to an expert in the semantic web field, this was not helpful to casual users. In this context, some of the negative comments repeated for most of the tools include:

- I found the URIs and sometimes ontology triples presented were technical and more targeted to experts.
- It would be good to change the way results are presented to allow non-experts to understand it.

The other commonly reported limitation of all the tools was the degree to which a query's results could be stored or reused. A number of the questions used in the evaluation had a high complexity level and needed to be split into two or more sub-queries. For instance, for the question "Which rivers in Arkansas are longer than the Allegheny river?", the users were first querying the data for the length of the Allegheny river and then performing a second query to find the rivers in Arkansas which are longer than the answer they got. Therefore, users often wanted to use previous results as the basis of a further query or to temporarily store the results in order to perform an intersection or union operation with the current result set. Unfortunately, this was not supported by any of the participating tools. However, this shows that users have very high expectations of the usability and functionalities offered by a semantic search tool as this requirement is not provided even by traditional search systems (e.g., Google and Yahoo). Another means of managing the results that users requested was the ability to filter results according to some suitable criteria and checking the provenance of the results; only one tool provided the latter. Indeed, even basic manipulations such as sorting were requested – a feature of particular importance for tools which did not allow query formulations to include superlatives. Again, some of the comments repeated in this context included:

- It would be nice to be able to sort the results.
- I often wanted to store previous answers and use them in subsequent queries or merge them with future result sets.

5 Future Directions

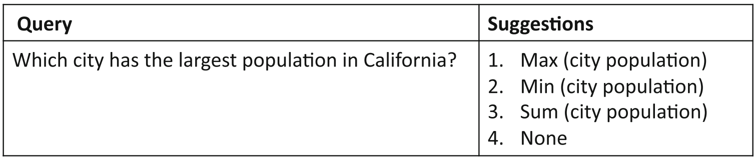
This section identifies a number of areas for improvement for semantic search tools from the perspective of the underlying technology and the user experience. It is motivated by the findings previously discussed in Sect. 4 and thus a similar structure is used.

5.1 Input Style

Usability. The feedback shows that it's very helpful for users – especially those who are unfamiliar with the underlying data – to explore the search space while building their queries using view-based interfaces which expose the structure of the ontology in a graphical manner. It gives users a much better understanding of what information is available and what queries are supported by the tool. In contrast, the feedback also shows that, when creating their queries, users prefer natural language interfaces because they are quick and easy. Clearly both approaches have their advantages; however, they suffer from various limitations when used separately as discussed in Sect. 4.1. Therefore, we believe that the combination of both approaches would help get the best of both worlds.

Users not familiar with the search domain can use a form-based or natural language-based interface to build their queries. Simultaneously, the tool should dynamically generate a visual representation of the user's query based upon the structure of the ontology. Indeed, the user should be able to move from one query formulation style to another – at will – with each being updated to reflect changes made in the other. This 'dual' query formulation would ensure a casual user correctly formulates their intended query. Expert users, or those who find it laborious to use the visual approach, would simply use the natural language input facility provided by the tool. The visualisation of the query structure is still useful for such users. An additional feature for natural language input would be an *optional* 'auto-completion' feature which could guide the user to query completion given knowledge of the underlying ontology. This not only helps the user but also alleviates the problem of mismatching the user terms with the correct ones in the underlying search space.

Expressiveness. The feedback also shows that the evaluated tools had difficulties with supporting complex queries such as the ones containing logical operators (e.g., "AND"). Allowing the user to input more than one query and combining them with their chosen logical operator from a list included in the interface would reduce the impact of this limitation. The tool would merge the results according to the used operator (e.g., "intersection" for "AND"). For instance, a query such as "What are the rivers that pass through California and



Query	Suggestions
Which city has the largest population in California?	1. Max (city population) 2. Min (city population) 3. Sum (city population) 4. None

Fig. 2. Suggestions generated by FREyA [8] for a datatype property, to handle superlatives and comparatives.

Arizona?" would be constructed as two subqueries: "What are the rivers that pass through California?" and "What are the rivers that pass through Arizona?" with the final results being the intersection of both result sets.

Furthermore, the evaluated tools faced similar difficulties with supporting superlatives and comparatives in users' queries. FREyA [8] deals with this problem by asking the user to identify the correct choice from a list of suggestions. To illustrate this we'll use the query *"Which city has the largest population in California?"*. If the system captures a concept in the user query that is a datatype property of type number, it generates maximum, minimum and sum functions. The generated suggestions for our query example are shown in Fig. 2. The user can then choose the correct superlative or comparative depending on their needs. A similar approach can be used to allow the use of superlatives and comparatives in natural language interfaces and form-based interface. In the case of the latter, whenever a datatype property is selected by the user, the tool can allow them to select from a list of functions that cover superlatives and comparatives (e.g., 'maximum', 'minimum', 'more than', 'less than').

5.2 Query Execution and Response Time

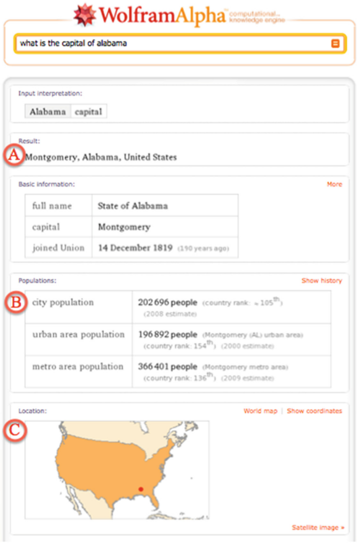
Several users reported dissatisfaction with the tools' response time to some of their queries. Users appreciated the fact that the tools returned more accurate answers than they would get from traditional search engines, however this did not remove the effect of the delay in response – even if it was relatively small. Additionally, the study found that the use of feedback reduces the effect of the delay; users showed greater willingness to wait if they were informed that the search is still being performed and that the delay is not due to a failure in the system.

The presentation of intermediate, or partially complete, results reduces the perceived delay associated with the complete result set (e.g., Sig.ma [12]). Although only partial results are available initially, it provides both feedback that the search is executing properly and allows the user to start thinking about the content of the results before the complete set is ready. However, it ought to be noted that this approach may induce confusion in the user as the screen content changes rapidly for a number of seconds. Adequate feedback is essential even for tools which exhibit high performance and good response times. Delays may occur at a number of points in the search process and may be the result of influences beyond the developer's control (e.g., network communication delays).

5.3 Results Presentation

Most of the users were frustrated by the fact that they didn't understand the results presented to them, feeling that too much technical knowledge was assumed. The evaluation showed that the tools underestimated the effect of this on user's experience and satisfaction.

Query answers ought to be presented to users in an accessible and attractive manner. Indeed, the tool should go a step further and augment the direct answer with associated information in order to provide a 'richer' experience for the user. This approach is adopted by WolframAlpha[4]. For example, as shown in Fig. 3, the response to the query '*What is the capital of Alabama?*' includes the natural language presentation of the answer (A) as well as various population statistics (B), a map showing the location of the city (C), and other related information such as the current local time, weather and nearby cities.


WolframAlpha computational.. knowledge engine
A
B
C
More
Show history
World map | Show coordinates
Satellite image »

what is the capital of alabama

Input interpretation:

Alabama	capital

Result:

Montgomery, Alabama, United States

Basic information:

full name	State of Alabama
capital	Montgomery
joined Union	14 December 1819 (190 years ago)

Populations:

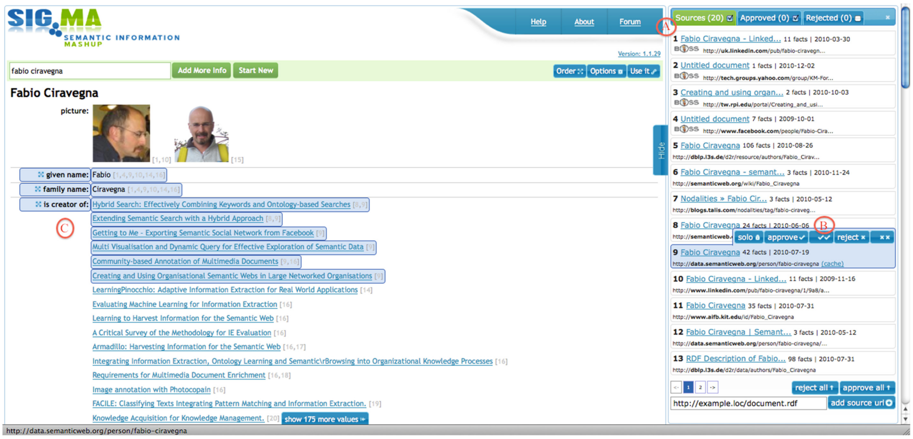
city population	202 696 people (country rank: ≈ 105th) (2008 estimate)
urban area population	196 892 people (Montgomery (AL) urban area) (country rank: 154th) (2000 estimate)
metro area population	366 401 people (Montgomery metro area) (country rank: 136th) (2009 estimate)

Location:

Fig. 3. Results presentation in *WolframAlpha*

An interesting requirement found by our study was the ability to store the result set of a query to use in subsequent queries. This would allow more complex questions to be answered which, in turn, improves the tools' performance. QuiKey [25] provides a functionality similar to this. QuiKey is an interaction approach that offers interactive fine grained access to structured information sources in a lightweight user interface. It allows a query to be saved which can later be used for building other queries. More complex queries can be constructed by combining saved queries with logical operators such as 'AND' and 'OR'.

[4] http://www.wolframalpha.com/.



fabio ciravegna

Fabio Ciravegna

picture: [1,10] [15]

given name: Fabio [1,4,9,10,14,16]

family name: Ciravegna [1,4,9,10,14,16]

is creator of:
Hybrid Search: Effectively Combining Keywords and Ontology-based Searches [8,9]
Extending Semantic Search with a Hybrid Approach [8,9]
Getting to Me - Exporting Semantic Social Network from Facebook [9]
Multi Visualisation and Dynamic Query for Effective Exploration of Semantic Data [9]
Community-based Annotation of Multimedia Documents [9,16]
Creating and Using Organisational Semantic Webs in Large Networked Organisations [9]
LearningPinocchio: Adaptive Information Extraction for Real World Applications [14]
Evaluating Machine Learning for Information Extraction [16]
Learning to Harvest Information for the Semantic Web [16]
A Critical Survey of the Methodology for IE Evaluation [16]
Armadillo: Harvesting Information for the Semantic Web [16,17]
Integrating Information Extraction, Ontology Learning and Semantic\rBrowsing into Organizational Knowledge Processes [16]
Requirements for Multimedia Document Enrichment [16,18]
Image annotation with Photocopain [16]
FACILE: Classifying Texts Integrating Pattern Matching and Information Extraction. [19]
Knowledge Acquisition for Knowledge Management. [20] show 175 more values

http://data.semanticweb.org/person/fabio-ciravegna

1 Fabio Ciravegna - Linked... 11 facts | 2010-03-30
BOSS http://uk.linkedin.com/pub/fabio-ciravegn...

2 Untitled document 1 facts | 2010-12-02
BOSS http://tech.groups.yahoo.com/group/KM-For...

3 Creating and using organ... 2 facts | 2010-10-03
BOSS http://tw.rpi.edu/portal/Creating_and_usi...

4 Untitled document 7 facts | 2009-10-01
BOSS http://www.facebook.com/people/Fabio-Cira...

5 Fabio Ciravegna 106 facts | 2010-08-26
http://dblp.l3s.de/d2r/resource/authors/Fabio_Cirav...

6 Fabio Ciravegna - semant... 3 facts | 2010-11-24
http://semanticweb.org/wiki/Fabio_Ciravegna

7 Nodalities » Fabio Cir... 3 facts | 2010-05-12
http://blogs.talis.com/nodalities/tag/fabio-ciraveg...

8 Fabio Ciravegna 24 facts | 2010-06-06
http://semanticweb.

9 Fabio Ciravegna 42 facts | 2010-07-19
http://data.semanticweb.org/person/fabio-ciravegna (cache)

10 Fabio Ciravegna - Linked... 11 facts | 2009-11-16
http://www.linkedin.com/pub/fabio-ciravegna/1/9a8/a...

11 Fabio Ciravegna 35 facts | 2010-07-31
http://www.aifb.kit.edu/id/Fabio_Ciravegna

12 Fabio Ciravegna | Semant... 3 facts | 2010-05-12
http://data.semanticweb.org/person/fabio-ciravegna/...

13 RDF Description of Fabio... 98 facts | 2010-07-31
http://dblp.l3s.de/d2r/data/authors/Fabio_Ciravegna

http://example.loc/document.rdf

Fig. 4. Example of a Sig.ma profile

Result management was also identified as being of importance to users with commonly requested functionality included sorting, filtering and more complex activities such as establishing the provenance and trustworthiness of certain results. For example, Sig.ma [12] – a system built on top of Sindice[5] – creates information aggregates called Entity Profiles and provides users with various capabilities to organise, use and establish the provenance of the results. Figure 4 shows part of the sigma for 'Fabio Ciravegna'. As shown in the figure, users can see all the sources contributing to a specific profile (A) and approve or reject certain ones (B), thus filtering the results. They can also check which values in the profile are given by a specific source as they get highlighted whenever the user scrolls over the source (C) thus checking provenance of the results. Sig.ma also supports the aspect of merging separate results by allowing users to view ones returned only from selected sources.

6 Conclusions

We have presented a flexible and comprehensive methodology for evaluating different semantic search approaches; we have also highlighted a number of empirical findings from an international semantic search evaluation campaign based upon this methodology. Finally, based upon analysis of the evaluation outcomes, we have described a number of additional requirements for current and future semantic search solutions.

In contrast to other benchmarking efforts, we emphasised the need for an evaluation methodology which addressed both performance and usability [24]. We presented the core criteria that must be evaluated together with a discussion of the main outcomes. This analysis identified two core findings which impact upon semantic search tool requirements.

[5] http://sindice.com/.

Firstly, we found that an intelligent combination of natural language and view-based input styles would provide a significant increase in search effectiveness and user satisfaction. Such a 'dual' query formulation approach would combine the ease with which a view-based approach can be used to explore and learn the structure of the underlying data whilst still being able to exploit the efficiency and simplicity of a natural language interface.

Secondly, (and perhaps of greatest interest to users) was the need for more sophisticated results presentation and management. Not only should fine grain feedback be provided on the progress of the search (such as the provision of intermediate results) but the final results should allow a large degree of customisability (sorting, filtering, saving of intermediate results, augmenting, etc.). Indeed, it would also be beneficial to provide data which is supplementary to the original query to increase 'richness'. Furthermore, users expect to be able to have immediate access to provenance information.

In summary, this paper has presented a number of important findings which are of interest both to semantic search tool developers but also designers of interactive search evaluations. Such evaluations (and the associated analyses as presented here) provide the impetus to improve search solutions and enhance the user experience.

References

1. Kaufmann, E.: Talking to the Semantic Web – Natural Language Query Interfaces for Casual End-Users. Ph.D. thesis, University of Zurich (2007)
2. Balog, K., Serdyukov, P., de Vries, A.P.: Overview of the TREC 2011 entity track. In: TREC 2011 Working Notes (2011)
3. Halpin, H., Herzig, D.M., Mika, P., Blanco, R., Pound, J., Thompson, H.S., Tran, D.T.: Evaluating Ad-Hoc object retrieval. In: Proceedings of the IWEST 2010 Workshop (2010)
4. Cleverdon, C.W.: Report on the first stage of an investigation onto the comparative efficiency of indexing systems. Technical report, The College of Aeronautics, Cranfield, England (1960)
5. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: weaving the open linked data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 552–565. Springer, Heidelberg (2007)
6. d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E.: Characterizing knowledge on the semantic web with watson. In: EON, pp. 1–10 (2007)
7. Lopez, V., Motta, E., Uren, V.S.: PowerAqua: fishing the semantic web. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 393–410. Springer, Heidelberg (2006)
8. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: combining syntactic analysis and ontology-based lookup through the user interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 106–120. Springer, Heidelberg (2010)

9. Bernstein, A., Kaufmann, E., Göhring, A., Kiefer, C.: Querying ontologies: a controlled english interface for end-users. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 112–126. Springer, Heidelberg (2005)
10. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid search: effectively combining keywords and semantic searches. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 554–568. Springer, Heidelberg (2008)
11. Clemmer, A., Davies, S.: Smeagol: a "specific-to-general" semantic web query interface paradigm for novices. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part I. LNCS, vol. 6860, pp. 288–302. Springer, Heidelberg (2011)
12. Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., Decker, S.: Sig.ma: live views on the web of data. In: Proceedings of the WWW 2010 (2010)
13. Wrigley, S.N., Elbedweihy, K., Reinhard, D., Bernstein, A., Ciravegna, F.: D13.3 Results of the first evaluation of semantic search tools. Technical report, SEALS Consortium (2010)
14. Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E., Giordanino, M.: The usability of semantic search tools: a review. Knowl. Eng. Rev. **22**, 361–377 (2007)
15. Angles, R., Gutierrez, C.: The expressive power of SPARQL. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 114–129. Springer, Heidelberg (2008)
16. Brooke, J.: SUS: a quick and dirty usability scale. In: Usability Evaluation in Industry, pp. 189–194. CRC Press (1996)
17. Bangor, A., Kortum, P.T., Miller, J.T.: An empirical evaluation of the system usability scale. Int. J. Hum. Comput. Interact. **24**(6), 574–594 (2008)
18. Bangor, A., Kortum, P.T., Miller, J.T.: Determining what individual SUS scores mean: adding an adjective rating scale. J. Usability Stud. **4**(3), 114–123 (2009)
19. Bernstein, A., Kaufmann, E., Kaiser, C.: Querying the semantic web with ginseng: a guided input natural language search engine. In: Proceedings of the WITS 2005 Workshop (2005)
20. Kaufmann, E., Bernstein, A., Fischer, L.: NLP-reduce: a "naïve" but domain-independent natural language interface for querying ontologies. In: Proceedings of the ESWC 2007 (2007)
21. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F.: Searching the semantic web: approximate query processing based on ontologies. IEEE Intell. Syst. **21**, 20–27 (2006)
22. Lei, Y., Uren, V.S., Motta, E.: SemSearch: a search engine for the semantic web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
23. Demidova, E., Nejdl, W.: Usability and expressiveness in database keyword search: bridging the gap. In: Proceedings of the VLDB Ph.D. Workshop (2009)
24. Wrigley, S.N., Elbedweihy, K., Reinhard, D., Bernstein, A., Ciravegna, F.: Evaluating semantic search tools using the SEALS platform. In: Proceedings of the IWEST 2010 Workshop (2010)
25. Haller, H.: QuiKey – an efficient semantic command line. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 473–482. Springer, Heidelberg (2010)

The DyKOSMap Approach for Analyzing and Supporting the Mapping Maintenance Problem in Biomedical Knowledge Organization Systems

Julio Cesar Dos Reis[3](✉), Cédric Pruski[1], Marcos Da Silveira[1], and Chantal Reynaud-Delaître[2]

[1] Luxembourg Institute of Science and Technology,
Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
{cedric.pruski,marcos.dasilveira}@list.lu
[2] LRI, University of Paris-Sud XI, Orsay, France
chantal.reynaud@lri.fr
[3] Institute of Computing, University of Campinas,
Av. Albert Einstein, 1251, Cidade universitária Zeferino Vaz,
13083-852 Campinas, SP, Brazil
julio.dosreis@ic.unicamp.br

Abstract. The ever-increasing quantity of data produced in biomedical applications requires the development of intelligent tools implementing Knowledge Organization Systems (KOS) like ontologies, thesauri, or classification schemes for a better integration and exploitation of this data. However, due to the size of this application field, element overlapping can occur between different KOS relying on various knowledge representation models. Therefore, mappings are necessary to improve the semantic interoperability of systems using KOS. Moreover, due to the dynamics of the biomedical domain, KOS constantly evolve over time, and new versions are released periodically forcing domain experts to revise the existing mappings affected by the evolution of the underlying heterogeneous KOS. In this paper, we provide an analysis of the mapping evolution problem. We highlight the lacks of existing approaches to deal with this problem and outline the sketch of the DyKOSMap framework as a prospective solution.

Keywords: Semantic mappings · Mapping maintenance · Mapping evolution · Mapping adaptation · Knowledge organization system evolution

1 Introduction

The development of biomedical applications contributes to the significant increase in the quantity of generated data. This is why there is an urgent need for intelligent tools that support the integration, harmonization and retrieval of this data. In this context, the exploitation of Knowledge Organization Systems (KOS) is promising [1]. According to

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 163–175, 2015.
DOI: 10.1007/978-3-662-46641-4_12

Hodge [2], KOS are intended to encompass all types of schemes: classifications and categorizations, taxonomies, thesauri, as well as semantic networks and ontologies. These schemes, which are defined using different knowledge representation models, are widely used in the biomedical field for various purposes. This is for instance the case for the ICD-9-CM classification that is used in billing systems for reporting diagnosis; the MedDRA terminology used to encode drug reports; the NCI Thesaurus (NCIT) implemented in the cancer research nomenclature; and SNOMED CT (SCT) which helps in organizing the content of Electronic Health Records.

This heterogeneity in the available resources for ensuring semantic interoperability in systems is one of the specificities of the biomedical domain. This is mainly due to the size of the domain. Since existing KOS are dedicated to a specific subfield, as it is the case for the Foundational Model of Anatomy [3] representing the phenotypic structure of the human body or Gene Ontology [4] for genetics, it forces users to implement several KOS in their system to optimize the coverage of the domain of interest. In addition, frontiers between the evoked subfields are quite fuzzy therefore some elements of the KOS overlap. It means that necessary characteristics or information to define the semantics of an object in the real world are duplicated in more than one KOS. For example, the code C80099 denoting the "Menopause, Premature" concept in the NCIT could be used to annotate a report dealing with this medical problem, however the same concept has a different code in ICD-9-CM (*e.g.,* 256.31).

This basic example underlines the necessity to define mappings, which express semantic correspondences between entities belonging to different KOS [5]. Mappings are important because users looking for reports annotated with C80099 may also be interested in documents tagged with the corresponding code in ICD-9-CM, since these are supposed to contain relevant information as well. In order to return such information, the system must be able to understand that both concepts, coming from different sources, are equivalent. To cope with this aspect of element overlapping in the biomedical domain, institutions like the *U.S. National Library of Medicine*[1] (NLM) define mappings between KOS, which are periodically released and distributed through applications like *Unified Medical Language System* (UMLS) [6] and *BioPortal* [7].

Another specificity of the biomedical domain concerns the rapid evolution of biomedical knowledge as pointed out by Baneyx & Charlet [8]. This implies that KOS must be updated in order to respect, as faithfully as possible, the evolution of the domain. As a consequence, new versions of the same KOS are frequently released, and the already established mappings between KOS have to be maintained each time a new version of a KOS is released. Even if the Semantic Web community has intensively studied ontology evolution [9], many aspects deserve a closer attention, in particular in the biomedical domain. An example might be the impact of changes in the established mappings.

In this context, the purpose of this paper is twofold. First of all, it aims at providing a quantitative and qualitative analysis of the mapping maintenance problem in the biomedical domain, with an emphasis on the lack of existing works in the field. This study has been realized on both the main KOS of the domain (*e.g.,* NCIT, ICD-9-CM,

[1] www.nlm.nih.gov.

SCT and MedDRA) and the mappings provided by the *BioPortal*. Second, based on this analysis, we sketch the outlines of the DyKOSMap approach. This approach aims at maintaining (adapting) mappings that exist between biomedical KOS in a more automatic way, by taking into account the semantic relationship type of the existing mappings combined with the evolution of these KOS.

The remainder of the paper is organized as follows: Sect. 2 presents both the results of our study regarding the mapping maintenance problem as well as the related works of the field. Section 3 introduces the DyKOSMap approach proposed in order to cope with the lack of existing approaches and the specificities of the problem previously introduced. We will also introduce the main characteristics of our approach through concrete examples borrowed from the biomedical field. We discuss the main strengths and weaknesses of the proposed approach in Sect. 4. Finally, Sect. 5 wraps up with concluding remarks and outlines future work.

2 The Mapping Maintenance Problem

Even if mapping maintenance has been recognized as an important problem to be tackled in the biomedical domain, little work has been proposed to solve it. As a consequence, it is necessary to have a clear understanding of the problem characteristics as well as of the elements that must be taken into account in the design of a solution. The analysis of the problem provided in this section is based on the observations we have made on the evolution of various KOS that are MedDRA, SCT, ICD-9-CM and the NCIT, and on the impact of their evolution on the mappings established between them.

2.1 Quantitative Analysis of the Problem

By virtue of the size of existing biomedical KOS (millions of elements in SCT), their dynamic nature and the quantity of mapped elements (*e.g.,* about 100.000 mappings between ICD-9-CM and SCT), the solutions provided to maintain mappings have to be as automatic as possible in order to support knowledge engineers in charge of this laborious and error prone task. To support this argument, we made a quantitative analysis of the KOS evolution and its impact on existing mappings. Our study consisted in observing all the changes that occurred in NCIT from March 2010 (version 10.01) to October 2011 (version 11.09) and identifying the amount of mappings between NCIT and ICD-9, and between NCIT and MedDRA (v. 12.0) affected by these changes. Actually, 1.162 different concepts were mapped from NCIT (v. 10.01) to ICD-9, and 6.195 different concepts were mapped from NCIT (v. 10.01) to MedDRA, according to information provided by the *BioPortal*[2] application. In October 2011 (after 19 months) it appears that 583 NCIT elements were modified (merged, split or removed). This was, for instance, the case of the concept code C80099 (Menopause, Premature), which was originally mapped to the concept code 256.31 (Premature menopause) of the ICD-9, and

[2] bioportal.bioontology.org.

to the concept labeled “Menopause” associated with the code 10027308 in MedDRA. In the new NCIT version, C80099 concept was merged with the concept C62595 (Premature_Ovarian_Failure). Consequently, due to modifications in NCIT, the previously established mappings are not valid anymore and therefore cannot be exploited by underlying biomedical systems unless they are updated. Our analysis shows that 0.8 % of the changes in NCIT had affected 0.4 % of the mappings established between NCIT and ICD-9. For the mappings between NCIT and MedDRA, 7.37 % of the changes in NCIT (between the considered versions) affected 0.69 % of the mappings. These observations clearly justify that the maintenance of mappings can hardly be done manually in an acceptable time due to the time spent on the identification of the impacted mapping, but it is completely irrelevant to re-compute the whole set of mappings each time a KOS evolves. It shows the real need for intelligent techniques and tools to cope with the maintenance of mappings affected by KOS evolution.

2.2 Qualitative Analysis of the Problem

We define mapping maintenance as:

> *“The modifications performed on the mappings established between*
> *KOS in order to keep them valid when these KOS evolve.”*

According to this definition, several dimensions have to be considered. We will highlight various facets of these dimensions through a set of experiments which is part of our study. The definition implies that mappings must already exist between KOS before they evolve in order to be maintained. Moreover, additional sets of experiments must show that not only modifications occurring in KOS impact the mappings, but also the types of modification (*e.g.,* the addition or removal of elements) and, in turn, the KOS model (*e.g.,* ontology, thesauri, classification, *etc*.).

Regarding this context, some attempts have been proposed in the literature. The majority of them have an emphasis on database schemas that are therefore strongly linked to the Entity-Relationship model. This is why they are inadequately coping with heterogeneous KOS models specificities. Fagin *et al.* [10] show how two fundamental operators on schema mappings, namely *composition* and *inversion*, can be used to address the mapping adaptation problem in the context of schema evolution. Due to the nature of database schemas, the adaptation process does not take into account the type of changes and their consequences. The work done by An *et al.* [11] concerns the maintenance of mappings established between ontologies and XML schemas. A formal solution is provided to (semi-) automatically adapt the mappings in order to maintain their validity when XML schemas evolve. An *et al.* present a plan which consists of several strategies to adapt a new semantic mapping. The provided solution clearly depends on the specificities of XML schemas and their limited possibilities of change. Additional investigation is required to figure out whether the proposed technique is applied to other KOS models.

In the case of ontologies, Martins & Silva [12] have proposed an original method for maintaining ontology mappings impacted by the element removal operation. Based on this method and depending on the nature of the ontology change strategy (elementary or composite changes), several cases have to be taken into consideration. In this approach, user intervention is necessary when the evolution leads to inconsistent ontologies. The mapping maintenance strategies are based on elementary descriptions of changes in a log, which may hardly support the maintenance process. Moreover, the proposal is too dependent on the Semantic Bridge Ontology (SBO) model, which is not a common way to define mappings. Recently, Khattak *et al.* [13] proposed an approach providing the benefits of mapping reconciliation between updated ontologies. This approach basically uses the change history of ontology to reduce the time required for reconciling mappings among ontologies. It considers the modified ontological elements, and not the nature of their changes and re-computes the mappings using matching algorithms with only the changed elements as input. In their work on ontology evolution, Hartung *et al.* [14] are interested in mappings, but between two versions of the same ontology. This implies that the KOS models do not interfere with the maintenance process. They provide a formally described set of changes that can occur within ontology. Nevertheless, to identify changes that have occurred, not all information provided in ontology is exploited, and the way similarity between ontology elements is computed to determine the type of changes is not explained.

Despite the improvements brought by these proposals from tackling the mappings maintenance problem, they fail by not taking into account the type of changes that occurred in a KOS, or they only support a specific KOS model and therefore must be adapted to cope with specificities of the biomedical domain. To overcome these gaps, we introduce the DyKOSMap framework that supports the mapping maintenance problem. The design of the framework is in line with the definition of mapping maintenance since it takes into account the type of changes in a KOS, the underlying heterogeneous KOS model and the information provided by existing mappings.

3 The DyKOSMap Approach for Biomedical KOS Mapping Maintenance

According to our study, changes occurring in a KOS, independent of its model, can have a direct impact on the mappings that have been established between the modified KOS and another one. This idea is the foundation of the approach supported by the DyKOSMap framework. Its utmost objective is to support, in an (semi-) automatic way, the maintenance of existing mappings between heterogeneous KOS through a coherent integration of the types of changes that affect a KOS, its underlying knowledge representation model and rules governing the overall process. The results of the process consist of a set of up-to-date mappings and the description of the history of mapping changes that is used to retrieve information about old mappings.

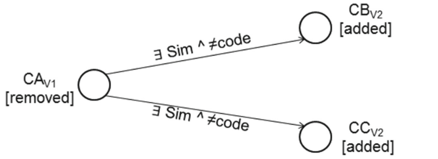
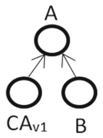
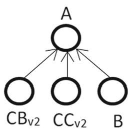
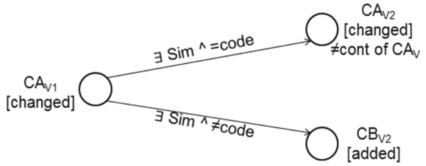
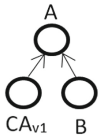
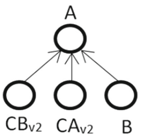
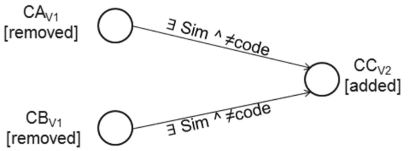
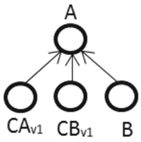
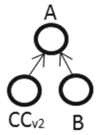
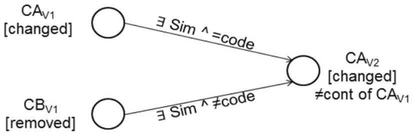
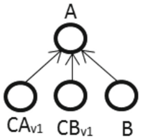
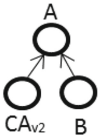
3.1 Influence of the Types of Changes in the Evolving KOS

The types of changes affecting a KOS have to be determined from the difference between two versions of the same KOS. These types of change are usually complex ones corresponding to sets of basic (atomic) change combinations [15]. Atomic changes are additions or removals of basic KOS entities; *e.g.*, the addition of a concept or an attribute, or the removal of a relation. A merge, a split or any other aggregation of these changes are examples of types of complex changes. For instance, the removal of the concept C62595 (Premature_Ovarian_Failure) from NCIT version 10.01 and the combination of its content with the concept C80099 (Menopause, Premature) from NCIT version 11.09 is a complex change corresponding to a merge operation.

The recognition from one version to another of types of complex changes operated in a KOS can be easier if these possible types are already known. We thus propose an approach based on the exploitation of generic descriptions of types of complex changes via change patterns (CPs) as it is done by Hartung *et al.* [14]. That way, in the previous example, the identification of the complex changes could be based on a merge pattern that applies at the moment a concept is deleted and most of its content appears in another (new) concept (*i.e.,* there is a similarity between them). The patterns are generic, and when applied, they are instantiated with entities of the evolving KOS.

CPs need to be defined and represented according to KOS models in order to take into account the underlying knowledge representation model. In order to improve the CP design, an empirical observation of old releases of different types of biomedical KOS is necessary over a significant period of time. Until today, we have studied CPs with regard to ICD-9-CM. Table 1 illustrates some examples of CPs designed observing ICD-9-CM. They present possible variations of merge and split complex changes. Besides those, other variations may also be found.

Table 1. Examples of change patterns

Change Pattern	**Pattern illustration**	**Before evolution**	**After evolution**
Split A	 CB_{V2} [added] ∃ Sim ∧ ≠code CA_{V1} [removed] ∃ Sim ∧ ≠code CC_{V2} [added]		
Split B	 CA_{V2} [changed] ≠cont of CA_V ∃ Sim ∧ =code CA_{V1} [changed] ∃ Sim ∧ ≠code CB_{V2} [added]		
Merge A	 CA_{V1} [removed] ∃ Sim ∧ ≠code CC_{V2} [added] CB_{V1} [removed] ∃ Sim ∧ ≠code		
Merge B	 CA_{V1} [changed] ∃ Sim ∧ =code CA_{V2} [changed] ≠cont of CA_{V1} CB_{V1} [removed] ∃ Sim ∧ ≠code		

In the examples of Table 1, C{x?}$_{v1}$ denote concepts from a KOS version 1, while C{x?}$_{v2}$ represent KOS elements, which were added, removed or modified in the same KOS version 2. "Merge A" describes a situation where CA$_{v1}$ and CB$_{v1}$ were removed from a KOS. CC$_{v2}$ appears in the next version of the KOS. All these 3 elements are semantically similar. The identifier code of CC$_{v2}$ is different from the code of CA$_{v1}$ and CB$_{v1}$. The semantic similarity value between these elements can be calculated using different methods. Actually, we measure the distance between strings used to describe concepts (*e.g.,* label, attribute values) and we check whether the applied changes are relevant from the semantic point of view using *MetaMAP*[3] application. In the same light, "Split B" pattern describes a situation where part of the information attached to the CA$_{v1}$ concept is used to create a new concept, CB$_{v2}$, while CA$_{v2}$ represents the modified CA$_{v1}$. This situation frequently occurs when the attributes of a concept are modified. These different change patterns may lead to divergent ways for maintaining the mappings. The possible patterns or their variations for other biomedical KOS models are under study.

There are many different approaches for identifying changes in the context of ontology evolution. One of them specifically deals with this identification by computing the difference between different versions of a same ontology. However, in this approach few proposals explicitly attempt to recognize complex changes [16–18], and KOS models specificities are not taken into account. The patterns that we described in Table 1 have been designed based on the observation of the evolution of the elements of ICD-9-CM. There are not many attempts in the literature to design CPs empirically, *i.e.,* based on experiments observing data evolution along the time. Javed *et al.* [19] proposed a graph-based pattern discovery algorithm analyzing ontology change logs to attempt an automatic derivation of possible change patterns. Shaban-Nejad [20] utilizes category theory for representing changes as patterns in a framework used for analyzing biomedical ontological changes. All these approaches do not define patterns empirically, and do not consider that changes may have distinct meanings according to different KOS models.

3.2 Techniques to Maintain Mappings

According to the selected CPs and KOS models, a set of actions (add, modify or delete) to update or refine mappings have to be done. Similarly, we propose to define the actions to be executed in a generic way. These actions are pre-defined. They are components of heuristics stored in a catalog. When a heuristics applies, then it is instantiated. Variables are replaced by concepts or relationships in the KOS.

Example of a heuristic:
"For the CP "Split A" (Table 1), add the "part of" relationship between CB$_{v2}$ (respectively CC$_{v2}$) and all concepts once linked by an equivalence relationship to CA$_{v1}$".

[3] metamap.nlm.nih.gov.

In this example, we assume that “part of” is a possible type of semantic relationship which can be established between the involved biomedical KOS elements. But, this heuristic does not indicate all the actions having to be executed. If the model of the KOS that is affected by evolution can represent hyperonymy or hyponymy relations between its elements, and if CA_{v1} was linked to a more general concept, then CB_{v2} and CC_{v2} will also be linked to this concept after evolution. In that case, new mappings that are a kind of heritage of the old mappings related to CA_{v1}, must also be added in relation to CB_{v2} and CC_{v2}. Therefore, the actions to make mappings evolve depend on the type of changes in the KOS (taken into account through patterns in the “For” part of the heuristics above) and also on the KOS model.

The approach that we propose aims at keeping the mappings valid after the evolution of a KOS, and also at keeping track of their history. This is done through a coherent integration of the instances of CPs, the corresponding set of actions suggested by the heuristics and the last (current) version of the mappings, and their associated history. More precisely, the system has to identify the mappings that will be reviewed first, *i.e.*, all mappings between the addressed KOS and the other KOS. Then, based on both the information from the instantiated CPs and the actions selected according to each CPs instance, the maintenance is performed on the impacted mappings only. In parallel, the system records the modifications of the mappings updating, thus of the mappings history. We obtain consistent and up-to-date mappings together with the history of mapping changes and KOS. How to manage the consistency of the mappings when they are updated is still an open research problem that will be tackled in future investigations.

To illustrate how CPs and actions from heuristics are combined to maintain mappings in practice, let us reconsider the example regarding “Premature Menopause”. In this example, C80099 (Menopause, Premature) concept code from NCIT is mapped to the concept code 256.31 (Premature menopause) in the ICD-9, and to the 10027308 (Menopause) in MedDRA. In the new version of NCIT, the concept C80099 is merged with the concept C62595 (Premature_Ovarian_Failure) and C62595 was removed from the KOS. The various CPs concerning NCIT is under study and may diverge from those studied for ICD-9-CM (Table 1). The mappings between the newly merged concept of NCIT and concepts from MedDRA and ICD-9 must be maintained but their relationships may change.

In this example, let us assume that the CP which applies considers that the remaining concept includes the removed concept. This CP is close to the “Merge B” pattern presented in Table 1. The heuristics associated to this NCIT CP must propose the following actions to maintain the mappings between NCIT and ICD-9:

(1) *Copy* the existing relationships between the concept C62595 and the concept of ICD-9 and *add* them to the modified concept C80099.

In this example, once the description of C62595 was added to C80099, the links to 256.31 and 10027308 still may be considered valid; in some cases more fine-grained constraints will need to be analyzed to observe whether the relationships of the mappings have to be reconsidered or not;

(2) *Delete* the mapping related to C62595 and *delete* possible redundant mappings related to C80099 (if there are).

The system has to interpret and transform these actions into basic tasks that are performed on the mappings. The execution of these tasks is dependent on the formalism (language) used to express mappings. The previous example shows a merged CP, and the performed adaptation actions are copy, add, and delete. A similar heuristic must be applied to update the mappings between NCIT and MedDRA.

In this example, the new mapping relations between the merged concept and the ancient mapped concepts will be the inverse of "part of". However, sometimes to define the exact semantics of the modified relationships is complex given the new definition of the concepts. Possible refinements of the relationships in the mappings could consider new attributes and relationships aggregated between the concepts when merging.

A real example that illustrates a complex change leading to a mapping refinement is related to the concept code C84382 (Sterilization) in the NCIT, which is mapped to ICD9/V25.2 (Sterilization) and to MedDRA/10054640 (Sterilization). Figure 1 presents an illustration of this case. After evolution, C84382 (Sterilization) in NCIT is split into C84382 and C95019 (Disinfectant) (as the "Split B" change pattern in Table 1). Consequently, the relationship of "Equivalence" type between C84382 (Sterilization) and ICD9/V25.2 (Sterilization) on one hand and MedDRA/10054640 (Sterilization) on the other hand have to be redefined. Mappings between some of these elements already existed but the relationships were not the same. The "equivalence" relation has to be converted into a "part-of" relation. In this case, the CP that has to be recognized is "split", and an example of instantiated action to be applied is (explained in natural language): "Modify the <equivalence> relation into <part-of> from <KOSa_C1> = NCIT_C84382 to <KOSb_C2> = ICD9_V25.2".

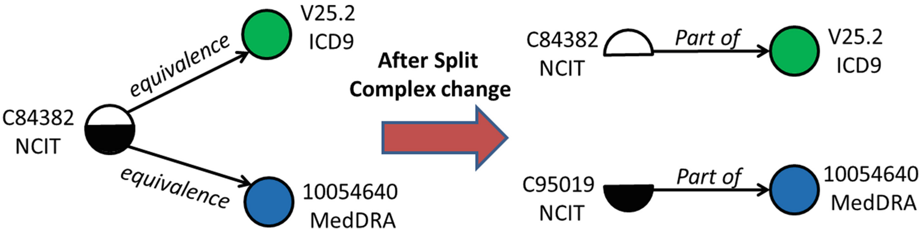


Fig. 1. Example of mapping maintenance

3.3 Towards the DyKOSMap Framework Definition

The general process proposed for mapping maintenance can be performed each time a new version of a KOS is released. In this section, we describe the workflow related to the mapping evolution process. An illustration of the DyKOSMap framework supporting the whole process is depicted in Fig. 2.

First, the changes between two different versions of a KOS are identified (CI in Fig. 2). To support this task, our approach relies on the use of previously designed sets of CPs. CPs aim at recognizing what types of changes (*e.g.,* merge, split) were performed on the elements of the two KOS versions. CPs are specific to the KOS models. The selection of the CPs depends on the KOS model and on the changes between the two versions of the KOS under study. Instances of CPs are generated by the CI module. The result serves as input to the Mapping Evolution (ME) module.

The Mapping Evolution (ME) module aims at maintaining mappings up-to-date with respect to the occurred changes, and at generating the histories (for keeping track) of both KOS (KOS history in Fig. 2) and mapping evolution (mapping history in Fig. 2). To support these tasks, the module relies on the use of previously designed sets of CPs and heuristics. Thus, modifying the CPs and/or the heuristics may also modify the way mappings are maintained in the ME process as a whole. The heuristics that have to be satisfied are different according to the possible changes of a KOS model identified by CPs. Heuristics selected by the ME module are coherent to the instantiated CPs obtained from the CI module. The decision of clearly separating CPs and heuristics is justified by the fact that they must be able to evolve over time, independent of the mapping maintenance process. Selected heuristics drive the overall mapping maintaining phase by providing the right actions to be done to maintain affected mappings without recalculating all the mappings.

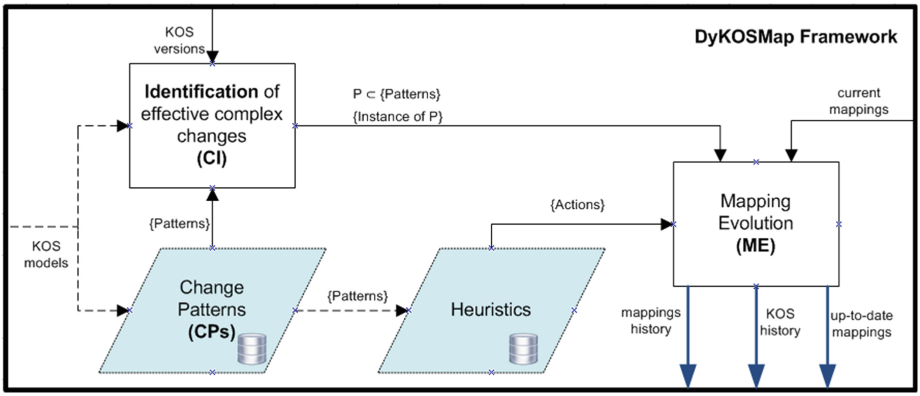


Fig. 2. The DyKOSMap framework

4 Discussion

Even though KOS evolution and mapping methods have been investigated in the literature, and several of these proposals have been applied to the biomedical domain, approaches for dealing with the mappings in the course of evolution are still lacking. There is no solution yet that could benefit from such proposals to maintain mappings according to the evolution aspects of biomedical KOS, which may contribute to decrease the time spent to review and keep mappings valid after KOS evolution.

Due to this lack, after each evolution step and KOS releases, software applications and humans in charge of maintaining the integration between the different KOS need to recalculate and validate all mappings. This strategy forces an efficient approach of placing the stress on the development of optimal matching algorithms and efficient validation strategies. Proposals in the literature are still not able to define how the evolution may impact the mappings, and how to deal with the mapping maintenance in the biomedical domain. Such an investigation must take into account the types of changes, which may be different according to the heterogeneous KOS models. However, a better understanding of the influences and impacts of evolution on mappings is needed to better accomplish it. Moreover, the current mapping maintenance solutions are not able to deal with the complexity brought by these heterogeneous KOS models and to take it into account in their solutions. They frequently focus on specific models, and mostly on database schemas. In addition, change patterns based on the KOS models are not considered in these solutions that may improve (better support) the way mappings are maintained over time.

Regarding this problematic we proposed an approach that handles mapping maintenance and KOS evolution as an integrated process. The framework supporting this approach is under development to accomplish the ME process and ensure the dynamic reconciliation of biomedical KOS. The idea is to split the entire ME process into several steps. For each one, approaches can be found in the literature and will be deeply evaluated in future work in order to identify the ones that could fit better to the framework. We will investigate their limitations and to use them within our approach and propose improvements and adaptations. However, at this moment, algorithms comparing results are beyond the scope of this paper. We focused on the general description of our approach, on the justification of the necessity of such solution and on the framework that will support it. The examples provided real illustrations on how the solution may work and on its possible outcomes.

In the proposed approach we assume that a first set of mapping was already built and validated, and we consider that these mappings will evolve over time based on the ME process. The final outcome provided by the framework (the evolved mappings and the history of the modifications) can be used to allow the retrieval and integration of information annotated with concepts from old versions of a KOS. This can avoid the update of every knowledge source that uses the modified KOS as annotation reference, or it can avoid end-users from being forced to store every version of a KOS in their system in order to perform retrieving activities.

Although this paper presents preliminary research results in suggesting a framework for maintaining mappings, it is important to consider the severity and urgency of the elucidated problem, of gaps in the literature which still need to be bridged, and of the benefits that a solution could have on applications. DyKOSMap is original for the issues we raised, and once first steps toward the problem are overcome, it might provide a way for performing a dynamic reconciliation of biomedical KOS.

5 Conclusion

Biomedical applications need to explore the evolution of the concepts and of the mappings between different KOS over time in order to enable publishing, integrating and retrieving data in an easier way. Dealing with the evolution effects of biomedical KOS is urgent, and mappings are the most affected elements. They deserve research attention for their adequate maintenance after evolution. Furthermore, biomedical vocabulary is frequently implemented in very heterogeneous KOS models, which increases the complexity of the issue. In this paper we presented the DyKOSMap approach for handling the changes and mapping evolution based on KOS complex changes identification. The proposed framework presents special phases based on the use of change patterns and on heuristics in a process designed to make the solution more adaptable. Our aim for future works is to develop a deeper investigation of each of the framework steps. Empirical experiments will be conducted in order to observe and understand the real effects of the evolution in the mappings established. The results from these experiments will lead to the heuristics definition. A tool based on this framework is envisioned, as well as case studies in real biomedical application scenarios.

Acknowledgments. This research is entirely funded by the National Research Fund (FNR) of Luxembourg under the DynaMO (www.santec.lu/project/dynamo) project. We thank São Paulo Research Foundation (FAPESP) (grant #2014/14890-0).

References

1. Eder, J., Dabringer, C., Schicho, M., Stark, K.: Data management for federated biobanks. In: Bhowmick, S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 184–195. Springer, Heidelberg (2009)
2. Hodge, G.: Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files (2000)
3. Rosse, C., Mejino, J.L.V.: The foundational model of anatomy ontology. In: Burger, A., Davidson, D., Baldock, R. (eds.) Anatomy Ontologies for Bioinformatics, pp. 59–117. Springer, London (2008)
4. The Reference Genome Group of the Gene Ontology Consortium: The gene ontology's reference genome project: a unified framework for functional annotation across species. PLoS Comput. Biol. **5**(7), e1000431 (2009)
5. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
6. Lindberg, D.A., Humphreys, B.L., McCray, A.T.: The unified medical language system. Methods Inf. Med. **32**(4), 281–291 (1993)
7. Noy, N.F., et al.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res. **37**(Web Server issue), W170–W173 (2009)
8. Baneyx, A., Charlet, J.: Évaluation, évolution et maintenance d'une ontologie en médecine: état des lieux et propositions méthodologiques. Revue Information - Interaction - Intelligence (2006)
9. Flouris, G., et al.: Ontology change: classification and survey. Knowl. Eng. Rev. **23**(2), 117–152 (2007)
10. Fagin, R., et al.: Schema mapping evolution through composition and inversion. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping, pp. 191–222. Springer, Heidelberg (2011)
11. An, Y., Borgida, A., Mylopoulos, J.: Discovering and maintaining semantic mappings between XML schemas and ontologies. J. Comput. Sci. Eng. (JCSE) **5**, 1–29 (2008)
12. Martins, H., Silva, N.: A user-driven and a semantic-based ontology mapping evolution approach. In: 11th International Conference on Enterprise Information Systems, Milano, Italy, p. 214–221 (2009)
13. Khattak, A.M., et al.: Reconciliation of ontology mappings to support robust service interoperability. In: IEEE International Conference on Services Computing (2011)

14. Hartung, M., Groß, A., Rahm, E.: COnto-Diff: generation of complex evolution mappings for life science ontologies. J. Biomed. Inf. **46**(1), 15–32 (2012)
15. Klein, M., Noy, N.F.: A component-based framework for ontology evolution. In: Workshop on Ontologies and Distributed Systems at IJCAI 2003, Acapulco, Mexico (2003)
16. Kirsten, T., et al.: GOMMA: a component-based Infrastructure for managing and analyzing life science ontologies and their evolution. J. Biomed. Semant. **2**, 6 (2011)
17. Gröner, G., Silva Parreiras, F., Staab, S.: Semantic recognition of ontology refactoring. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 273–288. Springer, Heidelberg (2010)
18. Hartung, M., Gross, A., Rahm, E.: Rule-based Generation of Diff Evolution Mappings between Ontology Versions, C. abs/1010.0122, Editor 2010, Cornel University Library (2010)
19. Javed, M., Abgaz, Y.M., Pahl, C.: Graph-based discovery of ontology change patterns. In: Joint Workshop on Knowledge Evolution and Ontology Dynamics (2011)
20. Shaban-Nejad, A.: A framework for analyzing changes in health care lexicons and nomenclatures in computer science and software engineering. Concordia University, Montreal, p. 357 (2010)

Effective Composition of Mappings for Matching Biomedical Ontologies

Michael Hartung[1,2](✉), Anika Gross[1,2], Toralf Kirsten[2], and Erhard Rahm[1,2]

[1] Department of Computer Science, University of Leipzig, Leipzig, Germany
{hartung,gross,rahm}@informatik.uni-leipzig.de
[2] Interdisciplinary Center for Bioinformatics, University of Leipzig, Leipzig, Germany
tkirsten@izbi.uni-leipzig.de

Abstract. There is an increasing need to interconnect biomedical ontologies. We investigate a simple but promising approach to generate mappings between ontologies by reusing and composing existing mappings across intermediate ontologies. Such an approach is especially promising for highly interconnected ontologies such as in the life science domain. There may be many ontologies that can be used for composition so that the problem arises to find the most suitable ones providing the best results. We therefore propose measures and strategies to select the most promising intermediate ontologies for composition. We further discuss advanced composition techniques to create more complete mappings compared to standard mapping composition. Experimental results for matching anatomy ontologies demonstrate the effectiveness of our approaches.

Keywords: Ontology matching · Mapping composition

1 Introduction

In recent years ontologies have become increasingly important in the life sciences [5,22]. For instance, Bio2RDF [3], the OBO Foundry [29] or BioPortal [24,33] distribute a growing number of biomedical ontologies from different domains such as anatomy and molecular biology. The ontologies are primarily used to annotate objects such as proteins, genes or literature to achieve a better information exchange. Often there are different ontologies from one domain containing overlapping or related information. As an example information about mammalian anatomy is available in NCI Thesaurus [23], Adult Mouse Anatomy [1] or the Unified Medical Language System [32]. In such cases *ontology mappings* can be used to express correspondences between different but related ontologies, e.g., which concepts of two different ontologies are equivalent.

Mappings between related ontologies are useful in many ways, in particular for data integration and enhanced analysis [18,24]. They are needed to merge ontologies [28], e.g., to create an integrated cross-species anatomy ontology such as the Uber ontology [31] or may also be useful to transfer knowledge from different experiments between species [6]. There are already numerous mappings

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 176–190, 2015.
DOI: 10.1007/978-3-662-46641-4_13

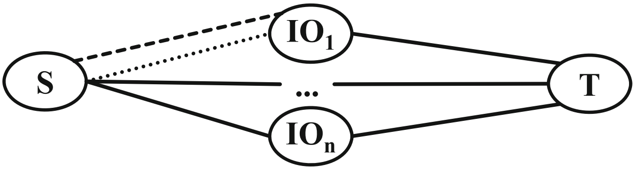


Fig. 1. Mapping composition with multiple intermediate alternatives

between ontologies available, e.g., BioPortal provides mappings between approx. 300 ontologies. However, there is still a strong need for increasing the number of mappings as most ontologies are interlinked to only one or a few other ontologies. Furthermore, new ontologies need to be connected to existing ones. The size of biomedical ontologies makes a manual generation of new mappings unfeasible, hence (semi-) automatic match algorithms are required.

We focus on the reuse and composition of existing mappings between ontologies to indirectly determine new ontology mappings and correspondences. Such an approach is especially promising for the life science domain where many mappings can be reused (e.g., from BioPortal). A main advantage of such a composition approach is its simplicity and high efficiency even for large ontologies. As shown in Fig. 1, one can use multiple alternatives (routes) to establish a new mapping between a source (S) and target (T) ontology using composition. First, there can be multiple intermediate ontologies IO ($IO_1 \ldots IO_n$) leading to questions like: "Is it better to use IO_1 instead of IO_2 or both?". Second, for one single intermediate ontology there can be several alternatives if there are multiple mappings between two ontologies (dotted/dashed lines between S and IO_1), e.g., determined by different match approaches. Considering a large number of possible composition alternatives we need an automatic approach to select the most suitable intermediates that likely result in the best composed mappings.

In this paper we study such selection methods and make the following contributions:

- We propose an efficiently computable measure to determine the effectiveness of composition routes via intermediate ontologies. For the case of composing two mappings, the effectiveness measure helps to find the most promising intermediate ontology.
- We describe two strategies using the proposed effectiveness measure to rank and select the top-k intermediates for mapping composition. Combining the derived mappings for the top-k routes helps to improve the overall mapping quality. We further discuss advanced composition techniques that may help to generate more complete ontology mappings compared to the standard technique.
- We evaluate the proposed approach on the OAEI [25] anatomy match task by using existing mappings determined by different match approaches. The obtained mapping quality results demonstrate the effectiveness of the proposed selection strategies.

This paper is an extended version of [15]. In Sect. 2 we introduce our ontology and mapping model. Section 3 presents the composition-based match approach. We describe our effectiveness measure and outline two strategies for selecting the most promising routes. In Sect. 4 we discuss advanced composition techniques. We evaluate the approach in Sect. 5. After a discussion of related work (Sect. 6), we summarize and outline possible future work.

2 Preliminaries – Ontologies and Mappings

An ontology $O = (C, R, A)$ consists of a set of concepts C which are interrelated by directed relationships R. Each concept has an unique identifier (e.g., accession number, URI) that is used to reference the concept, e.g., the concept 'Vertebra' in NCI Thesaurus is unambiguously referenced by `C12933`. A concept typically has further attributes $a \in A$ to describe the concept, e.g., `C12933` has the name 'Vertebra' and a synonym 'Vertebrae'. A relationship $r \in R$ forms a directed connection between two concepts and has a specific type, e.g., is_a or part_of. In our case `C12933` is a special 'Bone' (`C12366`): [`C12933`, is_a, `C12366`].

A *mapping* between two ontologies S and T, $M_{S,T} = \{(c_1, c_2, sim) | c_1 \in S, c_2 \in T, sim \in [0,1]\}$, consists of a set of correspondences between these ontologies, e.g., as determined by some ontology match method (see Related Work). Each correspondence interconnects two related concepts c_1 and c_2. Their relatedness is represented by a similarity value sim between 0 and 1 determined by the used match approach. The greater the sim value the more similar are the corresponding objects. Note that we focus on equality correspondences and leave the consideration of other correspondence types for future work. For already validated mappings we assume a similarity of 1 for each correspondence.

3 Rating and Selection of Composition Routes

In this section we present our approach to rate composition routes and to select the most promising ones. After introducing the concept of mapping composition, we propose an effectiveness measure to rate the value of routes in Sect. 3.2. Using this measure we describe the strategies *topKByEffectiveness* and *topKByComplement* for ranking and selecting the routes (Sect. 3.3). We finally describe in Sect. 3.4 the combined use of multiple selected routes to create a new mapping.

3.1 Composition for Generating New Mappings

The general idea behind mapping composition is to derive new mappings between two ontologies by reusing already existing mappings. Thus, new mappings are generated indirectly via one or more intermediate ontologies instead of a direct match between the two input ontologies. The typical situation for one intermediate is depicted in Fig. 2. The input consists of two ontologies S/T and two mappings $M_{S,IO}/M_{IO,T}$ w.r.t. an intermediate ontology IO. The `domain` and `range`

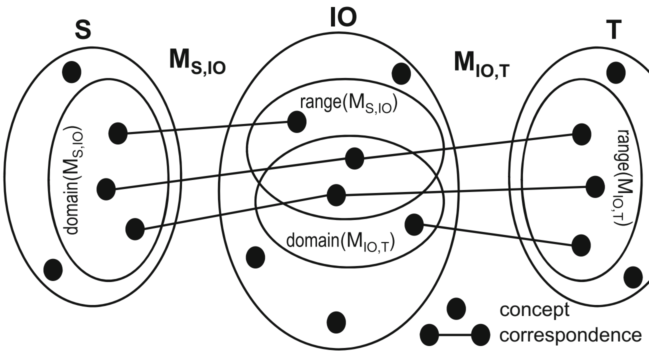


Fig. 2. General situation for mapping composition using one intermediate ontology

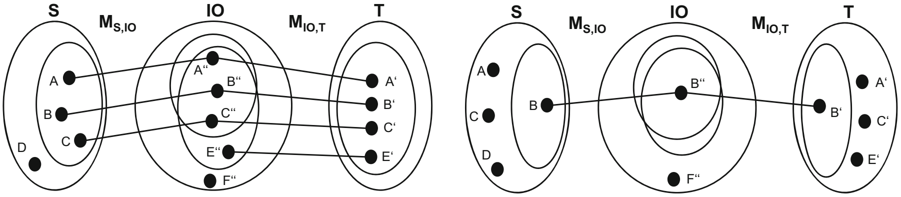
of the mappings can be used to find out which concepts are covered by the given mappings. For instance, all concepts of S covered by the mapping to IO are in its domain: $\texttt{domain}(M_{S,IO})$. Similarly, IO concepts covered by this mapping are in its range: $\texttt{range}(M_{S,IO})$. Mapping composition is then applied in the following way. A `compose` operator takes as input two mappings (from S/T to IO) and produces new correspondences between concepts of S and T if correspondences share the same concept in IO. The result is a new mapping $M_{S,T}$:

$$\begin{gathered} M_{S,T} = \texttt{compose}(M_{S,IO}, M_{IO,T}) = \\ \{(c_1, c_2, aggSim(sim_1, sim_2)) | c_1 \in S, c_2 \in T, b \in IO : \\ \exists(c_1, b, sim_1) \in M_{S,IO} \wedge \exists(b, c_2, sim_2) \in M_{IO,T}\} \end{gathered}$$

The similarity values of input correspondences are aggregated (aggSim) into new similarity values, e.g., by computing their maximum or average. In Fig. 2 we would create two correspondences between S and T since two concepts in IO overlap.

3.2 Effectiveness of Routes

The result of a mapping composition heavily depends on which intermediate ontologies are used and how the mappings to these intermediates look like. First, compose can at best create correspondences between concepts of S/T that are covered by the input mappings to an IO. The more concepts are covered by an input mapping the more likely it is that they can be interlinked to concepts in the other ontology. Thus, an intermediate for which mappings only cover a small portion of S/T are less effective compared to those covering larger portions. Second, there should be a high overlap of mapped objects in IO, i.e., many IO concepts should be in both $\texttt{range}(M_{S,IO})$ and $\texttt{domain}(M_{IO,T})$. This is because new correspondences can only be created if there are intermediate concepts for the composition. By contrast, a small overlap will only permit the creation of few correspondences, i.e., small and likely incomplete mappings. Based on these



(a) Positive example

(b) Negative example

Fig. 3. Examples for applying the effectiveness measure

observations we define a measure to rate the effectiveness of a route between sources S and T via an intermediate IO:

$$eff(S, IO, T) = \frac{2 \cdot |range(M_{S,IO}) \cap domain(M_{IO,T})|}{|S| + |T|}$$

The measure is largely based on the size of the overlap of concepts in the intermediate ontology, i.e., the larger the overlap the better the effectiveness. Second, we relate this overlap to the sizes of the ontologies to be matched S and T. Only mappings with many correspondences can produce a high overlap and a good coverage of concepts in S and T. Figure 3 shows two examples for applying the measure. The left example results in a good effectiveness of ($\frac{2 \cdot 3}{4+4} = 0.75$) because the overlap in the intermediate ontology covers a large part of S and T. By contrast, in the right example there is only one overlapping concept in the intermediate ontology resulting in a poor effectiveness of $\frac{2 \cdot 1}{4+4} = 0.25$. The compose operator would produce the following mappings (without similarity values): (a) $M_{S,T} = \{(A, A'), (B, B'), (C, C')\}$ and (b) $M_{S,T} = \{(B, B')\}$. This shows that the better rated intermediate ontology is able to produce more correspondences and thus a more complete mapping.

3.3 Ranking and Selection of Routes

Mapping composition using only one route may lead to insufficient (incomplete) match results. Composing mappings for several routes via different intermediates and combining their results is likely to improve the mapping to be determined. This is because other intermediate sources may provide additional correspondences between the input ontologies. The question thus arises which of the available routes should be selected for mapping composition. In the following, we describe two selection strategies that we will also evaluate later.

The first strategy *topKByEffectiveness* simply uses a ranking based on the effectiveness measure described in Sect. 3.2. Hence, we perform composition only on the k most effective routes and combine their results.

The second strategy *topKByComplement* also selects the most effective route but selects the remaining routes based on the number of complementary correspondences they can provide. The strategy determines how much additional gains can be achieved by considering further routes. For instance, if one has to match

Algorithm 1. topKByComplement

```
   Input: set of intermediates $all_{IO}$, input ontologies $S$ and $T$, number of
          intermediates to consider $k$
   Output: top intermediates $topK$
1  $topIO \leftarrow$ getMostEffectiveIntermediate($all_{IO}$);
2  $topK$.add($topIO$);
3  $all_{IO}$.remove($topIO$);
4  $cov_{all} \leftarrow$ domain($M_{S,topIO}$) $\cup$ range($M_{topIO,T}$);
5  while $|topK| < k$ do
6      $compl_{max} \leftarrow \emptyset$;
7      $topIO \leftarrow null$;
8      foreach $IO \in all_{IO}$ do
9          $compl_{IO} \leftarrow$ (domain($M_{S,IO}$) $\cup$ range($M_{IO,T}$)) $\setminus cov_{all}$;
10         if $|compl_{IO}| > |compl_{max}|$ then
11             $compl_{max} \leftarrow compl_{IO}$;
12             $topIO \leftarrow IO$;
13     $cov_{all} \leftarrow cov_{all} \cup compl_{max}$;
14     $topK$.add($topIO$);
15     $all_{IO}$.remove($topIO$);
16 return $topK$;
```

two anatomy ontologies, an ontology about the skeletal system would be complementary to one about the nervous system or blood circuit. Hence, it makes sense to consider intermediate ontologies that contain additional knowledge that others do not provide.

Algorithm 1 shows the implementation of this strategy. It first selects the most effective intermediate based on our effectiveness measure (lines 1–3). It then iteratively (while loop) adds the intermediate possessing the maximum complement ($compl_{max}$) compared to the already covered objects (cov_{all}) in S and T (lines 5–12). Particularly, we compare the covered concepts of the current intermediate with the covered concept set (cov_{all}) from already selected intermediates. In each round we select the intermediate which brings us the maximum complement. Note that the algorithm could be adapted to not only consider a fixed number (k) of intermediates. Instead we could stop taking further intermediates into account if their complement does not succeed a given threshold.

3.4 Overall Composition Algorithm

We use the algorithm *topKComposeMatch* (see Algorithm 2) to perform the composition for the k selected intermediates and to combine the composition results to obtain the overall mapping between two input ontologies.

We first apply our effectiveness measure on each route (line 1). Based on the given selection strategy (*topKByEffectiveness*, *topKByComplement*) we filter the top k promising intermediates (line 2). We then iteratively compose the

Algorithm 2. topKComposeMatch

```
Input: set of possible intermediates $all_{IO}$, input ontologies $S$ and $T$, selection
       strategy $selectionStrategy$, merge strategy $mergeStrategy$, number of
       intermediates to consider $k$
Output: mapping between $S$ and $T$ $M_{S,T}$
1 $all_{IO} \leftarrow$ computeEffectiveness($all_{IO}$,$S$,$T$);
2 $topK \leftarrow$ getTopRoutes($all_{IO}$,$selectionStrategy$,$k$);
3 $mapList \leftarrow empty$;
4 foreach $IO \in topK$ do
5     $M_{S,IO} \leftarrow$getMapping($S$,$IO$);
6     $M_{IO,T} \leftarrow$getMapping($IO$,$T$);
7     $mapList$.add(compose($M_{S,IO}$,$M_{IO,T}$));
8 return merge($mapList$, $mergeStrategy$);
```

mappings between S and T along each selected intermediate (lines 4–7). The generated mappings are temporarily stored in a *mapList* and are finally merged according to a specified merge strategy, such as union or intersection.

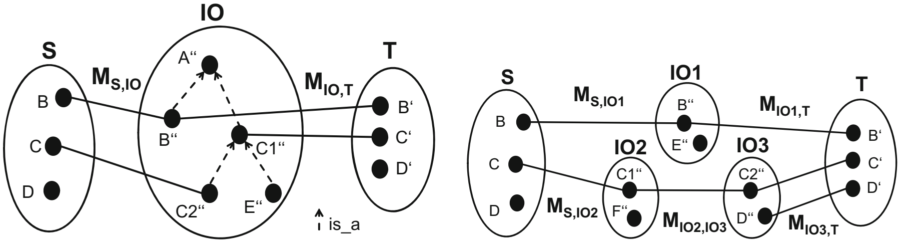
4 Advanced Composition Techniques

The compose operator described in Sect. 3.1 is most effective when many concepts in the intermediate ontology participate in both the first and the second input mapping. To improve the applicability of composition in less favorable settings we propose two generalized composition techniques to derive correspondences by reusing existing mappings. Such techniques can be applied incrementally, i.e., we would first use the standard `compose` operator to generate an initial mapping and then try to find further correspondences with the advanced techniques.

The two strategies we discuss in the following are *Semantic Neighborhood Composition* and *Multi-Step Composition*. We will not evaluate these strategies in this paper but leave this for future work.

4.1 Semantic Neighborhood Composition

Standard composition joins two correspondences $(c1, c2')$ and $(c2'', c3)$ only for $c2' = c2''$, i.e., if they share a concept in the intermediate ontology. With Semantic Neighborhood composition (*SNcompose*) we want to relax this condition by also composing correspondences where $c2'$ and $c2''$ are in close semantic neighborhood, e.g., if there are in a parent, child or sibling relationship. For the example in Fig. 4(a), standard composition only derives correspondence (B, B') via the shared concept B'' in the intermediate ontology IO. With SNcompose we can additionally find out that concept C in S and concept C' in T correspond to the closely related IO concepts $C1''$ and $C2''$ (that are in a parent-child relationship) so that the correspondence (C, C') may also hold. A similar kind of composition has already been applied by the taxonomy matcher of the COMA++ [2]



(a) SNcompose

(b) Longer route composition

Fig. 4. Examples for advanced composition techniques

matching tool where a taxonomy is used as an intermediate ontology for representing background knowledge.

In general, the `SNcompose` operator can be defined as follows:

$$M_{S,T} = \texttt{SNcompose}(M_{S,IO}, M_{IO,T}) =$$
$$\{(c_1, c_2, aggSim(sim_1, sim_2, sim_3)) | c_1 \in S, c_2 \in T, a, b \in IO :$$
$$\exists(c_1, a, sim_1) \in M_{S,IO} \wedge \exists(b, c_2, sim_2) \in M_{IO,T} \wedge \exists neighbor(a, b, sim_3) \in IO\}$$

It is assumed that the *neighbor* relation provides an intra-ontology distance or similarity (sim_3) between concepts depending on the type of relationship (parent of, child of, sibling of) and possibly further criteria such as cardinalities. This similarity is additionally used to compute the final similarity ($aggSim$) of the derived correspondence. However, one should be aware of that the resulting correspondences may no longer be of type ‘equality’, but that the relationship between the related concepts in the intermediate ontology may hold (e.g., an *is_a* relationship between C and C' in Fig. 4(a)).

4.2 Multi-Step-Composition

A second strategy to create additional correspondences is the adoption of a multi-step composition to combine multiple mappings within longer mapping paths over two or more intermediate ontologies. For the example in Fig. 4(b) we can only derive correspondence (B, B') when considering only the composition of two mappings via a single intermediate ontology such as $IO1$. Considering longer mapping routes via the intermediates $IO2$ and $IO3$ can help us to identify additional correspondences. In particular, composing the correspondences $(C, C1')$ and $(C1', C2')$ and the result with $(C2', C')$ leads to a second correspondence (C, C') between S and T. The idea thus is to apply the standard compose several times along a complete mapping path between the ontologies to match, i.e., to determine `compose`(`compose`(M_{S,IO_2},M_{IO_2,IO_3}),$M_{IO_3,T}$) in our example.

There may be many applicable mapping paths of length three or more so that it becomes even more important to select the most promising one. Our effectiveness measure introduced in Sect. 3.2 can be generalized to longer routes via several intermediates. In general such routes are mapping chains across ontologies $O_1, \ldots, O_n$ with $O_1 = S$ and $O_n = T$ and we can iteratively compute the overlap in each intermediate ontology. We determine the effectiveness as follows:

$$eff(O_1 \ldots O_n) = \frac{2 \cdot min_{i=2}^{n-1}[|range(M_{O_{i-1},O_i}) \cap domain(M_{O_i,O_{i+1}})|]}{|O_1| + |O_n|}$$

We take the minimal overlap since the intermediate with the smallest overlap restricts the overall effectiveness and a composition path must be represented in the overlaps of all intermediate ontologies. It is easy to see that the effectiveness measure for a single intermediate is a special case of the formula. In our evaluation, we will focus on routes with a single intermediate ontology and leave the evaluation of multi-step composition for future work.

5 Evaluation

We evaluate our approach by composing mappings between anatomy ontologies. In particular, we focus on generating mappings between the Adult Mouse Anatomy (MA) and the anatomy part of NCI Thesaurus (NCIT) which is a challenging task in the yearly OAEI [25] match contest. This has the advantage that we can use the publicly available OAEI gold standard (perfect mapping) to assess the quality of computed mappings (using precision, recall and F-measure) and to compare the achieved results with the published results of other approaches. Furthermore, we can reuse a lot of already existing mappings, in particular mappings provided by BioPortal [33] and mappings that we previously generated using our GOMMA ontology management infrastructure [20].

We first describe our experimental setup in more detail (Sect. 5.1). We then correlate the effectiveness measure with the achieved match results by composing the mappings according to different intermediate ontologies (Sect. 5.2). Finally, we adopt our selection strategies and present results of performing composition-based matching via the most promising intermediate ontologies (Sect. 5.4).

5.1 Experimental Setup

The experiment focuses on generating mappings between the ontologies MA (2,737 concepts) and NCIT anatomy part (3,298 concepts) as available in June 2011. We use 28 input mappings interrelating MA/NCIT via 11 different intermediate ontologies. The input mappings are separated in two different sets. The first mapping set (referred to as Mapping set 1) is taken from the community platform BioPortal [33] and comprises 20 mappings from MA or NCIT to 10 ontologies including BRENDA Tissue Ontology (BTO), Cell Line Ontology (CL), Foundational Model of Anatomy (FMA), Galen (Galen), Logical Observation Identifiers Names and Codes (LOINC), Medical Subject Headings (MeSH), RadLex, Uber Anatomy Ontology (Uber), Teleost Anatomy (TAO), and ZebraFish Anatomy (ZF). These mappings have been created with the LOOM match approach [12].

Table 1. Mappings between MA and NCIT included in the evaluation according to the two used mapping sets

	Mapping set 1										**Mapping set 2**			
Routes via intermediate	**Uber**	**FMA**	**CL**	**Galen**	**Radlex**	**MeSH**	**BTO**	**LOINC**	**ZF**	**TAO**	**UMLS**	**Uber**	**FMA**	**Radlex**
$\lvert M_{MA,IS} \rvert$	1882	1176	1131	699	679	569	513	403	250	206	2975	2300	1601	1082
$\lvert M_{IS,NCIT} \rvert$	1330	1804	1851	1083	902	941	684	650	375	334	4214	1703	2337	1347
$\lvert \texttt{range}(M_{MA,IS}) \cap \texttt{domain}(M_{IS,NCIT}) \rvert$	1048	825	793	564	479	438	372	364	169	140	2029	1320	1051	709
eff	0.35	0.27	0.26	0.19	0.16	0.15	0.12	0.12	0.06	0.05	0.67	0.44	0.35	0.23

LOOM takes all names and synonyms of the ontology concepts as input and returns concept pairs as matching when one of their name or synonym differ in at most one character. We use the mappings as provided by the BioPortal web page[1].

The second set of mappings (called Mapping set 2) consists of eight mappings interrelating MA and NCIT with four intermediate ontologies including Unified Medical Language System (UMLS), Uber, FMA, and Radlex. These mappings have been automatically created by a GOMMA match process. It uses a high trigram string similarity between concept name and synonyms to generate correspondences between concepts. Moreover, post-processing steps are applied to select only the best correspondence(s) per concept (MaxDelta selection (see [8])) and removal of crossing correspondences [19].

5.2 Route Effectiveness

We focus on routes involving a single intermediate ontology since there are many such routes. Typically, routes with chains of two or more intermediate ontologies may result in a reduced effectiveness. Table 1 shows selected statistics for the considered routes over different intermediates indicated in the columns. The routes are grouped by mapping set and ordered by the computed effectiveness (last row) starting with the route having the highest effectiveness. The first two rows characterize the input mappings for each route by showing the number of correspondences they comprise. These numbers are very different in both mapping sets ranging from approx. 1,900 (4,300) of the largest to about 200 (1,000) correspondences of the smallest mapping in Mapping set 1 (Mapping set 2). For the ontologies used in both mapping sets (Uber, FMA, and Radlex), the mappings in Mapping set 2 are larger than in Mapping set 1.

The third row displays the sizes of the mapping overlap in the intermediate ontology that is decisive for the effectiveness. In Mapping set 1, the route via Uber has the largest overlap (1,048 objects) and the highest effectiveness value of 0.35. In Mapping set 2, the number of referenced concepts in the intermediates is larger resulting in higher effectiveness values, but the relative order Uber, FMA,

[1] BioPortal: http://bioportal.bioontology.org, http://rest.bioontology.org.

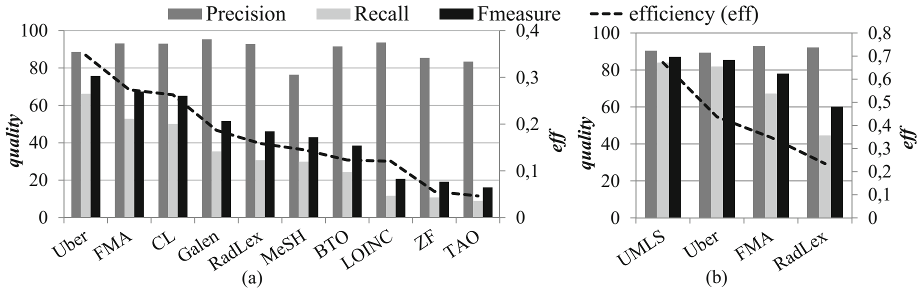


Fig. 5. Match quality for mapping compositions of routes with a single intermediate (sorted by effectiveness) for Mapping set 1 (a) and Mapping set 2 (b)

and Radlex remains. However, the route via UMLS has the highest effectiveness measure (0.67) and, is thus the most promising route for Mapping set 2.

5.3 Correlation of Routes Effectiveness and Composition Quality

Figure 5 correlates the effectiveness (dashed line, z-axis on the right) for each route with the match quality of the composed mapping in terms of precision, recall and F-measure (bars, y-axis). The routes are decreasingly ordered by their effectiveness from left to right and separated for both mapping sets. Overall, there is an excellent correlation between the effectiveness values and achieved match quality for both mapping sets. This means that the composed correspondences are indeed valuable and contribute to the match result so that higher effectiveness values translate into higher F-measure values. For instance, for Mapping set 1 the route via Uber has the best effectiveness and the highest F-measure of 0.76 whereas the route via TAO with the lowest effectiveness (0.05) results in the worst F-measure of only 0.16. The same holds for Mapping set 2: the route via UMLS (Radlex) with the highest (lowest) effectiveness generates a mapping with the best (worst) F-measure of 0.87 (0.6). Therefore, using the effectiveness metric is a valid and reliable means to select the intermediate ontology providing the best match quality.

5.4 Top K Selection and Composition

In the next experiment, we evaluate whether the match quality (F-measure) can be increased when using the proposed selection strategies *topKByEffectiveness* and *topKByComplement* for selecting k routes and combining their composition results. We set k to 3 and use union as merge operation in both selection strategies. According to the effectiveness values for each route (see Table 1 and Algorithm 1) we select routes via Uber, FMA, and CL (UMLS, Uber, and FMA) in Mapping set 1 (Mapping set 2) for the *topKByEffectiveness* strategy and routes via Uber, FMA, and Galen (UMLS, Uber, and FMA) in Mapping set 1 (Mapping set 2) for the *topKByComplement* strategy. For comparison, we consider several additional selection strategies. They include the single route with the highest

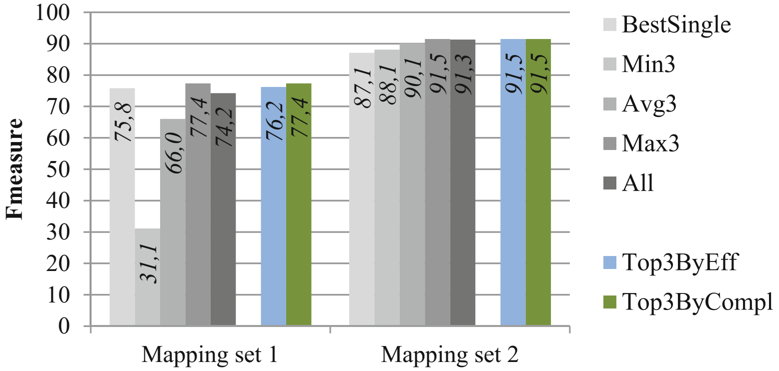


Fig. 6. Match results of combinations of multiple routes

F-measure in the mapping set (BestSingle) and the strategies resulting in the worst (Min3), average (Avg3), and best (Max3) F-measure result for combining any three routes. Moreover, we computed the combination of all routes per mapping set (All).

Figure 6 shows the F-measure for all selection strategies and both mapping sets. The results show that in both cases the *topKByComplement* strategy focusing on complementary mappings produces the max. possible match quality, i.e., it is able to identify the best and most effective composition routes. Interestingly, doing a compose-based match on only three out of the 10/4 possible routes results in better match quality than using all available routes since it apparently avoids wrong correspondences introduced by weaker routes. For instance, in Mapping set 1 F-measure is increased by 3 % (74.2 % → 77.4 %) compared to the 'All' strategy. For Mapping set 2, the F-measure is improved by 0.2 % compared to 'All'. Using this strategy we participated with GOMMA in the 2011.5 OAEI contest[2]. The achieved F-measure of 91.5 % is comparable to the best result in the OAEI contest (91.7 % F-measure of AgreementMaker [7] in 2011). While the OAEI contest poses certain restrictions, the participating prototypes did also exploit background knowledge for the Anatomy test case. Our *topKByEffectiveness* strategy shows marginally worse results compared to *topKByComplement* (76.2 % vs. 77.4 % for Mapping set 1), apparently since CL complements Uber and FMA less well than using Galen as intermediate ontology.

6 Related Work

Ontology matching is the process of determining a set of semantic correspondences (ontology mapping) between concepts of two ontologies. A manual matching by domain experts is very time-consuming and for large ontologies almost infeasible. Thus, many (semi-)automatic matching algorithms have been developed for ontology matching (see [10,26,27] for surveys). Common match approaches follow a

[2] http://oaei.ontologymatching.org/2011.5/.

direct matching by employing lexical and structural methods; some approaches also consider the similarity of ontology instances. State-of-the art match systems such as COMA++ [2], Falcon [16] or SAMBO [21] combine multiple matchers within a match strategy to achieve better match quality. Results of matching biomedical ontologies showed that linguistic matching methods based on the similarity of concept names and synonyms produce very good results [12,35]. To improve the runtime of matching (especially for large ontologies) some systems try to reduce the search space [17] or perform parallel matching on multiple compute nodes [13].

The composition of mappings has mainly been studied for schemas [9,11] and in model management [4]. Only a few approaches consider mapping composition for deriving new mappings in ontology matching. For instance, [34] utilizes FMA as an intermediate to indirectly generate a mapping between MA and NCIT. Similarly, the SAMBO system [21] utilizes background knowledge (e.g., UMLS) to find additional correspondences in the match process. Reference [30] presents an empirical analysis of mapping composition available in BioPortal. In own related work [14], we already studied mapping composition. The primary focus of this work was on match quality (F-measure) by a manual intermediate selection but not on automatic strategies to select the best intermediates according to their expected contribution to the overall match quality.

In contrast to these approaches this paper differs in the following points. First, we apply mapping composition with multiple routes, while most match approaches only consider one route or purely apply a direct match. Second, we focus on finding the most valuable routes for mapping composition out of a pool of possible routes in two different mapping sets. A ranking of routes w.r.t. their effectiveness allows us to compose mappings for a reduced number of routes saving time and possibly improving match quality as shown in the evaluation.

7 Conclusion and Future Work

We proposed a new approach to rank and select promising routes for composing mappings between biomedical ontologies. The introduced effectiveness measure can be easily computed and allows a reliable identification of the most promising intermediate ontologies for composition-based ontology matching. We further proposed the selection of the k top routes and the combination of their composition results for improved match quality. Our evaluation for an OAEI match task on large anatomy ontologies showed the effectiveness of the proposed approach. In particular we found that the effectiveness metric for different routes correlates excellently with their achievable F-measure quality. Furthermore, we found that the *topKByComplement* ranking strategy is most effective that combines the route with the best effectiveness with routes providing most complementary correspondences. Our approach could effectively exploit existing mappings and achieved an excellent 91.5 % F-measure for the challenging OAEI anatomy task. This shows that mapping composition is not only an efficient method to derive new mappings but can also increase the match quality, e.g., by finding additional correspondences compared to a direct match approach.

In future work we plan to apply and extend the approach for other domains, ontologies and data sources, e.g., matching Linked Data sources. In particular, we want to investigate inter-linking of instance objects and to consider further correspondence types. We further like to study the discussed advanced composition techniques in more detail, e.g., longer mapping chains via multiple intermediates.

Acknowledgment. This work is supported by the German Research Foundation (DFG), grant RA 497/18-1 ("Evolution of Ontologies and Mappings").

References

1. Adult Mouse Anatomy: http://www.informatics.jax.org/searches/AMA_form
2. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proceedings of the SIGMOD, pp. 906–908 (2005)
3. Belleau, F., et al.: Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J. Biomed. Inform. **41**(5), 706–716 (2008)
4. Bernstein, P., Melnik, S.: Model management 2.0: manipulating richer mappings. In: Proceedings of the SIGMOD, pp. 1–12 (2007)
5. Bodenreider, O., Stevens, R.: Bio-ontologies: current trends and future directions. Briefings Bioinform. **7**(3), 256–274 (2006)
6. Bodenreider, O., et al.: Of mice and men: aligning mouse and human anatomies. In: Proceedings of the AMIA Annual Symposium, pp. 61–65 (2005)
7. Cruz, I.F., et al.: Using AgreementMaker to align ontologies for OAEI 2011. In: Proceedings of the International Workshop on Ontology Matching, pp. 114–125 (2011)
8. Do, H., Rahm, E.: Matching large schemas: approaches and evaluation. Inform. Syst. **32**(6), 857–885 (2007)
9. Dragut, E., Lawrence, R.: Composing mappings between schemas using a reference ontology. In: Meersman, R. (ed.) CoopIS/DOA/ODBASE 2004. LNCS, vol. 3290, pp. 783–800. Springer, Heidelberg (2004)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag New York, Secaucus (2007)
11. Fagin, R., Kolaitis, P., Popa, L., Tan, W.: Composing schema mappings: second-order dependencies to the rescue. ACM Trans. Database Syst. (TODS) **30**(4), 994–1055 (2005)
12. Ghazvinian, A., Noy, N., Musen, M.: Creating mappings for ontologies in biomedicine: simple methods work. In: Proceedings of the AMIA Annual Symposium, pp. 198–202 (2009)
13. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: On matching large life science ontologies in parallel. In: Lambrix, P., Kemp, G. (eds.) DILS 2010. LNCS, vol. 6254, pp. 35–49. Springer, Heidelberg (2010)
14. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: Mapping composition for matching large life science ontologies. In: 2nd International Conference on Biomedical Ontology (ICBO), pp. 109–116 (2011)
15. Hartung, M., Gross, A., Kirsten, T., Rahm, E.: Effective mapping composition forbiomedical ontologies. In: eProcceedings of Semantic Interoperability in Medical Informatics @ ESWC (2012)
16. Hu, W., Qu, Y.: Falcon-AO: a practical ontology matching system. Web Semant. Sci. Serv. Agents World Wide Web **6**(3), 237–239 (2008)

17. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: a divide-and-conquer approach. Data Knowl. Eng. **67**(1), 140–160 (2008)
18. Jakoniene, V., Lambrix, P.: Ontology-based integration for bioinformatics. In: VLDB Workshop on Ontologies-Based Techniques for DataBases and Information Systems-ODBIS 2005, pp. 55–58 (2005)
19. Jean-Mary, Y., Shironoshita, E., Kabuka, M.: Ontology matching with semantic verification. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 235–251 (2009)
20. Kirsten, T., Gross, A., Hartung, M., Rahm, E.: GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. J. Biomed. Semant. **2**, 6 (2011)
21. Lambrix, P., Tan, H.: Sambo-a system for aligning and merging biomedical ontologies. Web Semant. Sci. Serv. Agents World Wide Web **4**(3), 196–206 (2006)
22. Lambrix, P., Tan, H., Jakoniene, V., Strömbäck, L.: Biological ontologies. In: Baker, C.J.O., Cheung, K.-H. (eds.) Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences, pp. 85–99. Springer, New York (2007)
23. NCI Thesaurus: http://ncit.nci.nih.gov/
24. Noy, N.F., et al.: Bioportal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res. **37**(Suppl. 2), W170–W173 (2009)
25. Ontology Alignment Evaluation Initiative: http://oaei.ontologymatching.org/
26. Rahm, E.: Towards large scale schema and ontology matching. In: Bellahsene, Z., Bonifati, A., Rahm, E. (eds.) Schema Matching and Mapping, Chap. 1, pp. 3–27. Springer, Heidelberg (2011)
27. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001)
28. Raunich, S., Rahm, E.: Atom: automatic target-driven ontology merging. In: ICDE, pp. 1276–1279 (2011)
29. Smith, B., et al.: The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. **25**(11), 1251–1255 (2007)
30. Tordai, A., et al.: Lost in translation? Empirical analysis of mapping compositions for large ontologies. In: Proceedings of the Ontology Matching Workshop (2010)
31. UBERON: http://obofoundry.org/wiki/index.php/UBERON:Main_Page
32. Unified Medical Language System: http://www.nlm.nih.gov/research/umls
33. Whetzel, P.L., et al.: Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Res. **39**(Suppl. 2), W541 (2011)
34. Zhang, S., Bodenreider, O.: Alignment of multiple ontologies of anatomy: deriving indirect mappings from direct mappings to a reference. In: AMIA Annual Symposium Proceedings, pp. 864–868 (2005)
35. Zhang, S., Bodenreider, O.: Experience in aligning anatomical ontologies. Int. J. Semant. Web Inform. Syst. **3**(2), 1–26 (2007)

Semantic Interoperability Between Clinical Research and Healthcare: The PONTE Approach

Anastasios Tagaris[1], Efthymios Chondrogiannis[1],
Vassiliki Andronikou[1], George Tsatsaronis[2](✉),
Konstantinos Mourtzoukos[1], Joseph Roumier[3], Nikolaos Matskanis[3],
Michael Schroeder[2], Philippe Massonet[3], Dimitrios Koutsouris[1],
and Theodora Varvarigou[1]

[1] National Technical University of Athens, 9, Heroon Polytechniou Str.,
15773 Athens, Greece
{tassos,dkoutsou}@biomed.ntua.gr,
{chondrog,vandro,kmour,doravarv}@mail.ntua.gr

[2] Biotechnology Center (BIOTEC), Technische Universität Dresden (TUD),
Tatzberg 47–49, 01307 Dresden, Germany
{george.tsatsaronis,ms}@biotec.tu-dresden.de

[3] Centre d' Excellence en Technologies de l' Information et de la
Communication (CETIC), Rue des Frères Wright 29/3, 6041 Chareroi, Belgium
{joseph.roumier,nikolaos.matskanis,
philippe.massonet}@cetic.be

Abstract. The adoption of ICT technologies in healthcare for recording patients' health events and progression in Electronic Health Records (EHRs) and Clinical Information Systems (CLIS) has led to a rapidly increasing volume of data which is, in general, distributed in autonomous heterogeneous databases. The secondary use of such data (commonly anonymised for privacy reasons) for purposes other than healthcare (such as patient selection for clinical trials) comprises an emerging trend. However, this trend encapsulates a great challenge; semantic interlinking of two different, yet highly related, domains (in terms of semantics) i.e., clinical research and healthcare. This paper aims at presenting an analysis of the heterogeneity issues met in this effort and describing the semantically-enabled multi-step process followed within the PONTE project for achieving the interlinking of these two domains for the provision of the size of the eligible patients for participation in a trial at the cooperating sites.

Keywords: Semantic interoperability · Electronic health records · Ontology mapping

1 Introduction

Clinical research aims, among others, at revealing the therapeutic potential of substances, methodologies and devices in order for them to be developed into real-world therapies. A critical step in this process lies in the design and conduction of clinical

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 191–203, 2015.
DOI: 10.1007/978-3-662-46641-4_14

trials, which comprise the investigation of the efficacy and safety of these candidate treatments. Over the years great debate has been taking place concerning the therapy development timeline, the devoted resources as well as the weakened R&D productivity; i.e. the number of therapies which reach the real-world vs the number of investigational therapies researched upon. The reported figures in drug development demonstrate that of every 5000 molecules which are pre-clinically tested, only 1 will in the end be approved and will enter the market [1]. In the meanwhile, the number of new active ingredients entering the market has been significantly reduced over the years [2], while the estimated average cost per drug candidate reaches € 900 million [3], with recently reported figures indicating that this cost may even reach € 9 billion [4] per drug approved. And what is more, the therapy investigation and development comprises a heavily prolonged process with the new drug development timeline being 11.3 years on average [3]. The latter poses significant limitations in the advancement of the domain, while thousands of patients anticipating for a new treatment remain untreated or are following treatments of low/medium efficacy, of not necessarily minor health risks and/or of reduced quality of life.

Key steps in the process of Clinical Trial Design and Implementation are (i) the definition of the inclusion and exclusion criteria (aka eligibility criteria) that describe the target population of the study and eventually the criteria which, patients should meet for participating in the trial and (ii) the recruitment of these patients. The eligibility criteria specification is among the most important steps of study design as it determines at a significant degree the feasibility and the applicability of the study conduction, as well as the value of the study outcome. Patient recruitment comprises another critical step in the clinical trial lifecycle. In fact, prolonged recruitment periods are reflected on study costs, while poor recruitment in trials (i.e. inability to reach the required sample size) is a significant bottleneck for the evaluation of new therapies and quite often leads to study findings of low external validity or even trial termination. One of the key reasons for the series of issues related to recruitment is the limited ability to reach patients due to the insufficient means used such as trial advertising and oral communication of the intent.

A new trend towards improving the process for the selection of eligible patients (based on these inclusion and exclusion criteria) involves taking advantage of ICT technologies and more specifically existing Clinical Information Systems (CLIS) or Electronic Health Record (EHR) systems currently in operation at hospitals and healthcare centres. Although the secondary use (i.e., other than healthcare provision purposes) of healthcare patient data sets has been investigated as of great importance and potential impact, their isolated and specific-focus development has led to significant variations in the organisation and representation of information, the technologies exploited and the implementation of these systems among others. These variations increase the complexity of such efforts, especially in the cases that interoperability needs to be achieved between these systems but also with external ones, such as clinical research information systems. In fact, an additional level of complexity is introduced due to the semantic distance between the different domains which are required to be interlinked.

Focus of this paper is at the challenges posed in the second step which stem from the native interoperability open issues in the world of CLIS and EHRs which becomes an even harder challenge to address, due to the different nature and way of view between clinical research and clinical practice. Moreover, in this paper we present the approach followed within the PONTE project [12] for overcoming the variety of heterogeneity issues between clinical research and clinical care domains. In fact, among the key objectives in PONTE is the semantic interlinking of clinical research, with particular focus on the clinical trial eligibility criteria, and the clinical patient data at healthcare for providing the researchers with an instant view of the size of the eligible population available at the cooperating hospitals.

More specifically, in Sect. 2, we analyse the great challenges faced in order to achieve interoperability among clinical research systems and healthcare clinical data sources which stem from the variety of heterogeneity issues at structural, syntactic, semantic and interfacing levels. In Sect. 3, we present the methodology developed and applied within the PONTE project for allowing communication with healthcare patient data sources for two purposes: (a) retrieving the size of the available population satisfying the eligibility criteria of a clinical trial during its design and (b) selecting the eligible patients for screening purposes after approval of the study protocol. Section 4 presents the key open issues while Sect. 5 summarises the main findings of the presented approach.

2 The Challenges

Given a set of eligibility criteria for patients to be selected and participate in a clinical trial, there is a need for a systematic communication with CLIS or EHRs of hospitals or healthcare centres in order to identify the patients that satisfy this set of criteria. To our knowledge there is no automated and systematic procedure to perform this process due to the heterogeneity of the multiple and different datasources. A methodological approach to overcome the challenges posed by high heterogeneity (at system, syntax, structure, semantics and interface/messaging level [5]) of datasources is presented in this paper.

According to [6, 7], the heterogeneity between data sources is classified into the following four categories:

i. *System Heterogeneity*: encapsulates the differences at the level of different hardware used and operating systems.
ii. *Syntactic Heterogeneity*: is related to the data models (relational, object oriented, hierarchical model, etc.) used to organize our knowledge. Based on the data model selected, a different language (SQL, OQL etc.) may be used to access the data. Moreover, many systems do not provide direct access to the data, but enable data retrieval using user/system specific queries. In such cases additional heterogeneity issues arise at interface/messaging level, concerning the means of accessing the data as well as the structure of the queries posed and the results retrieved.
iii. *Structural Heterogeneity*: is based on the way we organize our information. We can represent the same information in many different ways even if we use the same data model. Different schemas may be used for representing the same

information. For example, a many-to-many relationship can be implemented by using one-to-many and many-to-one relationships, but there is no standard way to find this implemented in the existing CLIS or EHRs. Different implementations may follow different approaches depending on the needs of the specific hospital department and applications.

iv. *Semantic Heterogeneity*: The semantic heterogeneity is related to the meaning of the elements of the schema. It represents the way people understand a specific domain of knowledge. They can use different terms to refer to the same concepts, while they may use the same concept to refer to different things. It can be found both at schema level and data/instance level. Thus, there might be cases of different terms referring to the same concepts (synonyms), the same term referring to different things (homonyms), missing data across EHRs, concepts used that have a broader, narrower or overlapping meaning. Also, differences could exist in the unit or scale of the measurements.

All the above have been depicted in Fig. 1.

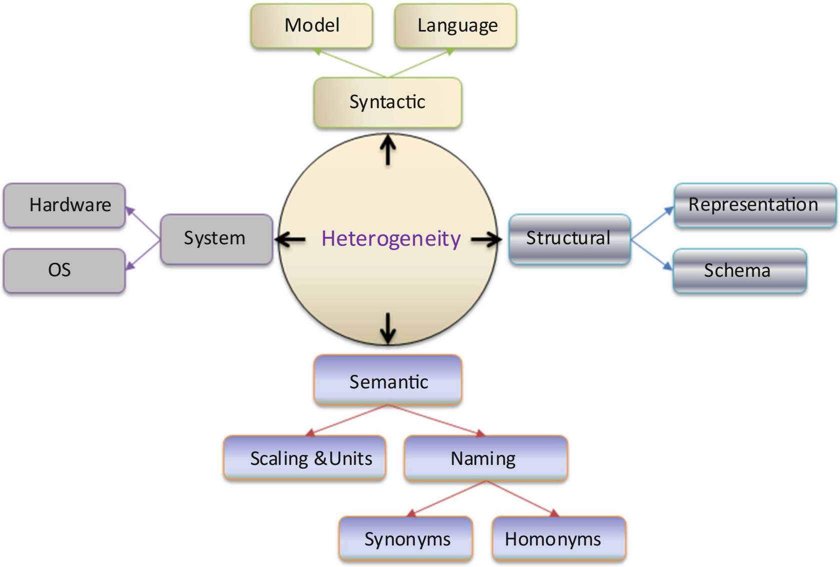


Fig. 1. CLIS and EHR datasources heterogeneity

A typical example of synonyms at the *schema level* of the different datasources relates to the name of the field, under which the "disease" of a patient is recorded, with terms such as "disorder", "diagnosis", "disease", "condition", being used across different EHRs. An example of *semantic heterogeneity* across EHRs at the *data/instance level* lies in the vocabularies used for recording the disorders for patients. For example, "myocardial infarction" may also be documented within a datasource as myocardial infarct or MI – an acronym of the latter ones - (synonym), acute myocardial infarction (narrower meaning) or heart disorder (broader meaning) with different codes being used in different coding systems, vocabularies and classifications.

The best case scenario, in which international terminology standards (such as the ICD-10 [8] or SNOMED-CT [9]) have been adopted for the documentation of a patient's disease, still imposes important challenges. Although certain mapping efforts across those terminologies can be used for at least overcoming this type of heterogeneity (a typical example of which is the NCI Metathesaurus [10]), handling terms of narrower, broader or overlapping meaning, even in this case, still remains a challenging task. The challenge gets even greater in cases that *local coding schemas and/or vocabularies* are used within the hospitals, which in turn requires mapping to an international standard vocabulary or coding system in order for any external application to query upon this data across hospitals in a transparent and efficient manner.

In order to achieve successful communication with the different data sources, we need to handle all the aforementioned heterogeneity issues and provide a way to the end user (who in most of the cases is expected to have limited IT background being a clinical researcher) to pose queries without needing to know the internal characteristics of these datasources. A safe way to overcome the challenges in the different interoperability levels is needed and the presented work aims at realising this through the use of semantics and appropriate ontologies to converge between the clinical research and clinical care domains.

Still several challenges exist, as clinical care systems are based mainly on RDBMS implementations with no semantic or ontological elements. A first challenge is to convert a relational-based model to an ontological model. Assuming that the clinical research domain (and in our case the eligibility criteria) is also represented using an ontological model, the next challenge is the *alignment* of the different ontologies. Hence, the eligibility criteria ontology and the CLIS or EHR ontology (derived from the corresponding CLIS or EHR system respectively) need to be aligned, while *mapping services* together with *transformation rules and mechanisms* (to address terminological and structural issues) have to come in place. This is depicted in Fig. 2.

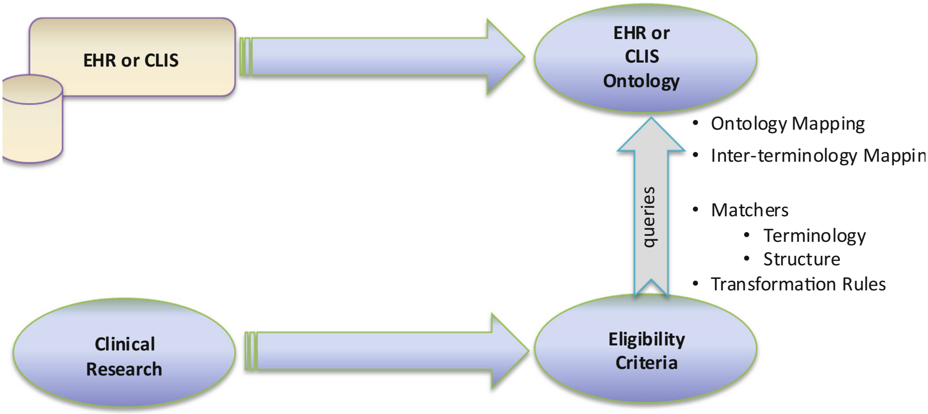


Fig. 2. Convergence of Clinical Research and Clinical Care using Ontologies

According to [11], combining and relating ontologies is by no means a straightforward procedure. On the contrary, many challenges are required to be overcome depending on the "mismatches" between those ontologies. These mismatches arise when two or more ontologies describe (partly) overlapping domains (apart from language mismatches which may also exist) (Fig. 3):

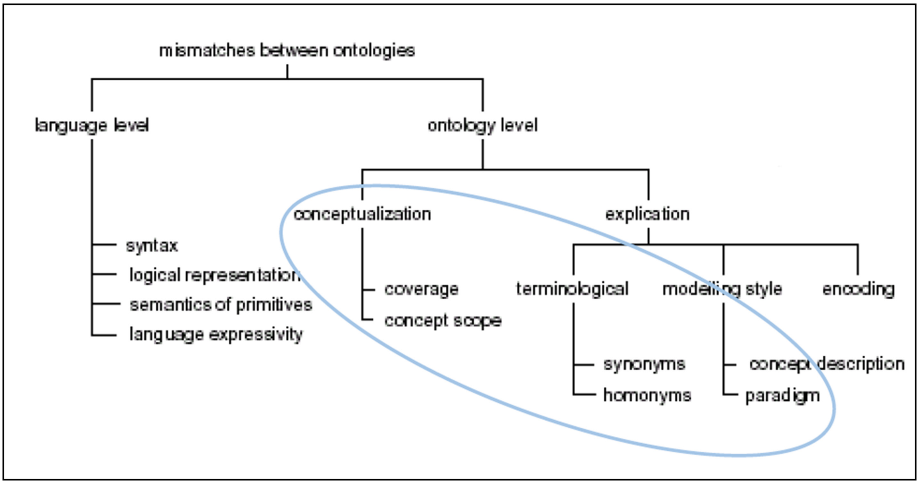


Fig. 3. Mismatches between ontologies [11]

i. *Language Level Mismatches*: This category of mismatches comes from the different languages that can be used to describe ontologies. Each one has its own syntax, representation, semantics and expressivity.
ii. *Conceptualization Mismatch*: These mismatches are met due to the different ways a domain of knowledge can be interpreted (conceptualized). They are further classified into "scope" and "model coverage and granularity" mismatches. In the first case, two classes which represent the same concepts may not have the same instances. In the second case, the part of the domain that is covered is different and some models are more detailed than others
iii. *Explication Mismatch*: These mismatches are driven by the way a domain of knowledge is specified. They are classified into the following subcategories:

  - *Paradigm*: Different paradigms can be used to represent concepts. Paradigm mismatch is the problem when two different approaches are to be used at the same time. The most common case of this idiom is in between ObjectModeling and RelationalModeling. The use of different "top-level" ontology is also an example of this kind of mismatch.
  - *Concept Description*: These mismatches arise from the choices made while conceptualizing a domain. For instance, the full name of a person can be specified using one property ("full_name") or using two properties ("first_name" and "last_name").

- *Terminological Mismatches*: The same concept may be represented using different names (*Synonym*) or the same term may have different meanings (*Homonym*).
- *Encoding*: These mismatches occur due to the variety of ways in which a value may be represented. For instance, a date may be represented as “dd/mm/yyyy” or “dd/mm/yy” or “dd-mm-yyyy”.

Another challenge that comes into place has to do with the alignment of datasources heterogeneity with the ontologies heterogeneity.

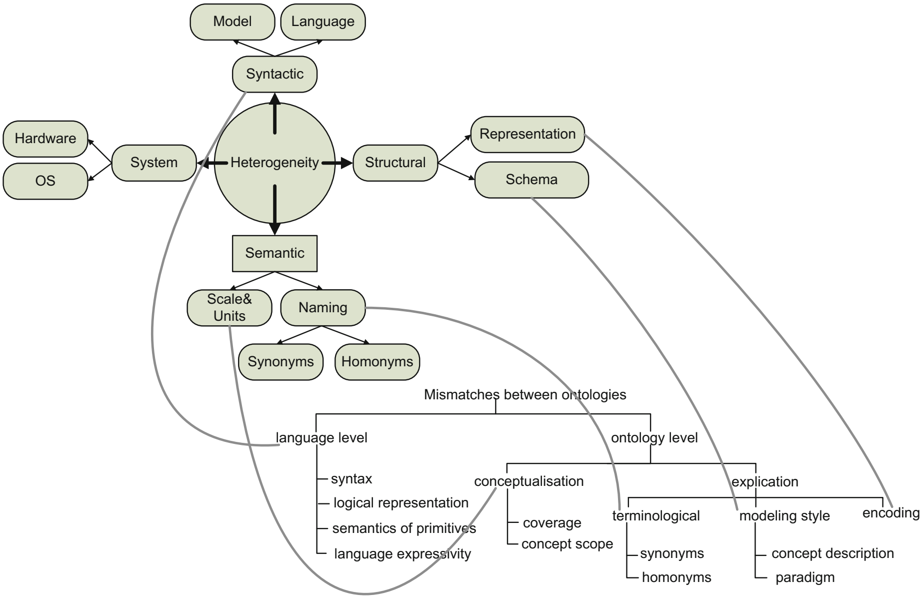


Fig. 4. Ontologies Heterogeneity and Data sources Heterogeneity

As we can see in Fig. 4, the Language Level mismatches are related to the Syntactic Heterogeneity. From the *Ontology Level* mismatches, the Conceptualization mismatches are related to the Scaling & Unit Heterogeneity (a subcategory of the Semantic Heterogeneity) whereas the *Explication mismatches* are associated with both Structural and Semantic Heterogeneity. More precisely, the Modelling Style and Encoding mismatches are linked with the Schema and Representation Heterogeneity accordingly (Structural Heterogeneity) while the Terminological mismatches are related to the Naming Heterogeneity (which comprises a subcategory of the Semantic Heterogeneity).

In the following section we present the methodological approach that we have followed in PONTE in order to address the aforementioned challenges.

3 Semantic Interoperability in PONTE

3.1 The PONTE Approach

Within PONTE we follow an ontological approach covering the whole representation of information requested and retrieved; from the clinical research-oriented information (i.e., the eligibility criteria) to the healthcare patient data on which the queries are posed. For this purpose two ontologies have been developed using the Web Ontology Language (OWL) namely: (i) the *Eligibility Criteria ontology* aiming at representing the eligibility criteria parameters (such as gender, life expectancy, contraindications to a treatment etc.) as well as the relationships among them and (ii) the *Global EHR ontology* which comprises the PONTE-side representation of the healthcare patient data. The latter include demographics, general characteristics and health-related parameters (such as administered medication, diagnosed conditions, operations scheduled or performed, etc.).

The semantic distance of the two types of information represented by the two aforementioned ontologies can be better understood through the following example: a potential inclusion criterion for a trial could be "*patients with adequate liver function*". Such information is not expected, in most cases, to be found within a CLIS or an EHR system. In order to be able to apply such a criterion in those systems, it needs to be converted into a query applicable to such systems based on their semantics; i.e., with parameters – at least partially - semantically correlated with this criterion. In the aforementioned example this corresponds to patients who have *never* been diagnosed with "*liver cirrhosis*" or are not *currently* suffering from "*Hepatitis*", etc. Hence, the ontological approach has been chosen in order to exploit the semantic relationships among the parameters for alignment purposes across the two domains (Fig. 5).

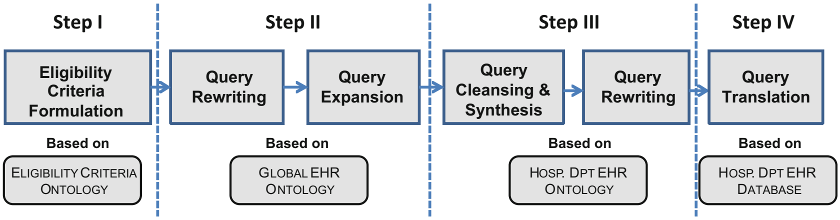


Fig. 5. PONTE approach for Interlinking with healthcare patient data

In the figure above, the main steps followed in the methodology developed and adopted within the PONTE platform are presented. Based on this approach, the initial set of eligibility criteria is expressed as SPARQL queries. Overall, a series of transformations from each initial SPARQL query representing the eligibility criteria and expressed over the Eligibility Criteria Ontology to one final SQL query applicable to the relational EHR database take place. This approach is driven by the effort to break down the heterogeneity issues so that they are almost individually addressed at a series of steps and thus reduce its complexity.

In Step I, the eligibility criteria are formulated as SPARQL queries expressed over the terms of the Eligibility Criteria ontology. Rather than generating a single SPARQL query for all the criteria, each criterion is formulated as a single query, so that it can be processed individually. One of the main reasons behind this decision is the fact that not all criteria are expected to be applicable at the different patient datasets (for example, information indicating that a patient will be willing to provide informed consent (a typical inclusion criterion) is not expected to be found in any EHR).

In Step II a query rewriting process takes place during which the initial SPARQL queries generated in Step I are transformed into SPARQL queries expressed over the terms of the Global EHR ontology. In order for this transformation to take place, an initialization phase has been set up during which the Eligibility Criteria ontology has been aligned with the Global EHR ontology and transformation rules have been built to be used at run-time. The heterogeneity issues addressed at this step are purely semantic and structural. For example, an expected eligibility criterion would be age range (e.g., patients from 20 to 50 years old), whereas the date of birth is expected to be found within an EHR. Applying such a criterion would require at this step alignment of age with date of birth and a transformation rule indicating the calculation of age from the date of birth. At this point it should be noted that the vocabularies and classifications (for disorders, active substances, laboratory examinations, race, etc.) used in these two ontologies are kept the same for semantic consistency reasons.

In this step, *query expansion* comprises an accompanying process, which might be required in some cases. Let's take for example the case that an exclusion criterion describes that "Patients suffering from any heart disorder" should not participate in the trial. This criterion, in fact, encapsulates a wide range of disorders, while in the EHRs (where the *diagnosed* diseases of patients are recorded) a possible record (which is semantically related to this criterion but not equivalent) would be "ST elevation (STEMI) myocardial infarction of anterior wall". Thus, in this case the query expansion process aims at collecting the terms of narrower meaning and expanding the SPARQL query (for any concept included in a criterion) so that all possible values, for the EHR parameter included in the query, are covered. After this step, as described above, the eligibility criteria are *semantically* much closer to the EHRs than when initially generated.

At this point it should be noted that, in most – if not all – of the cases, CLIS and EHRs are implemented using relational or object oriented database servers. In order to extract such database schemas to a semantic representation (ontological format) we have used the D2R server [13]. The D2R server takes as input a mapping file expressed in the D2RQ Language [14]. The D2R server provides also a tool that can be used to generate this mapping file based on the schema of the database. The reason for performing this process and extracting the ontology of the EHR database instead of directly transforming the Global EHR ontology-based SPARQL queries into the final SQL query for the EHR database is that such a transformation would require addressing a variety of heterogeneity issues, namely *semantic, structural, syntactic and interface heterogeneity*, *at the same time*.

Hence, in Step III the queries produced by Step II are synthesized into one SPARQL query which encapsulates all the eligibility criteria, each of which previously comprised a separate SPARQL query. The reason behind this is that in order for a patient to be eligible at the clinical trial s/he should meet all the inclusion and exclusion criteria specified for this study. Before the synthesis of the SPARQL query, a cleansing mechanism is applied. During this process the SPARQL queries representing eligibility criteria which cannot be applied on the healthcare patient data sets are removed. The need for this mechanism is driven by two main reasons. The first one lies in missing information at structural level of the patient data sets. For example, an eligibility criterion might exclude "Patients who have been smoking for the past 5 years", whereas no information about the tobacco use is being recorded for patients at the EHR or CLIS. The second reason involves missing information at data level. In this case, the healthcare department (e.g. cardiology) might be recording specific health-related events for patients, such as a specific subset of diseases (for example only heart and metabolism diseases). If an eligibility criterion aims at including "patients who suffer from liver cirrhosis", then the querying of such a data set will provide no results. *The analysis of this response, however, shows that it is of low confidence since it is based on the wrong assumption that the patient data sets provide a full description of the patients' health and thus lack of information means no such health event for the patient ever occurred.*

The resulting SPARQL query (which is still expressed over the terms of the Global EHR ontology) is then transformed into a SPARQL query expressed over the Hospital Department EHR *ontology* (aka Local EHR ontology) which comprises the ontological representation of the healthcare patient data sets. As in Step II, a mapping process between the Global EHR ontology and each one of the Local EHR ontologies has been held and a set of transformation rules has been developed during the initialization phase. The latter set is being used during the query rewriting process.

Taking into consideration that the vocabularies and coding systems used at the side of the EHRs are not expected to be the same as within the PONTE platform, the query rewriting process at this step also encapsulates *semantic mapping of the terms* used. For example, there might be the case that the diagnosis in an EHR are coded based on ICD9. In this case, transformation of the disorder values in the eligibility criteria from ICD10-CM (which is used within PONTE) to ICD9 would be required. For this purpose, the EVS Vocabulary Servers[1] are being used. This step focuses on overcoming semantic and structural heterogeneity issues. In Step IV, the SPARQL query synthesized during the previous step is translated into an SQL query based on the mapping file generated by the D2R server as described above.

3.2 The Mapping Issues

As mentioned above, during the initialisation phase of the presented methodology a set of mapping steps is required among the ontologies involved in each step. The mapping process itself poses a series of requirements on the language which will be used for representing it. In [15] the functionality/expressivity that should be provided from an ontology mapping language is investigated. According to this paper, the mapping is defined as a list of assertions (mapping rules) each of which defines the relation

[1] https://cabig.nci.nih.gov/community/concepts/EVS/.

between a set of ontological entities (concepts and relations). For each one we should determine whether it is bidirectional or not. Also we should keep information about the nature of the mapping rules (generated from an ontology alignment algorithm or defined by the user). The mapping language should be expressive enough to allow the definition of both simple and complex correspondences in order to handle the *syntactic and structural* differences (or conceptualization and explication mismatches). It should also contain the necessary operators which would allow for specifying that two concepts are equal or that one concept is more general than the other one or that they are partially overlapping. Additionally it should give us the functionality needed to define more complex relations such as that a concept is equal with the intersection of two or more concepts from the other ontology.

As an example let's take the full name of a person. This may be represented with the property "full_name" in one ontology, while the same information may be represented using the properties "first_name" and "last_name" in another ontology. The mapping language should allow us to specify this correspondence. In this case, simply defining that the full name is equal with the intersection of the first and last name is not enough. We should also specify that the first name is the first token of the full name whereas the last name is the second token of the full name.

Let's see another example. Suppose that we have in the first ontology the class "Person" in which the age is specified using the property "has Age". In the second ontology we have only the concept of the Adult. These ontologies don't have exactly the same information. In the first ontology the information for a person is the actual age, whereas in the second ontology the corresponding information describes whether s/he is an "adult" or not. As we know, an adult is a person whose age is greater than 18 years old. So, it is obvious that there is a correspondence between these ontologies. In this case, we should specify that the Person which has Age greater than 18 is considered an Adult person and vice versa.

Due to the variety of the correspondences between the two ontologies, their specification requires much more than what a declarative language can offer. Hence, we need a procedural language which enables us to specify every possible relation identified. These issues, which comprise real problems that we should overcome in order to achieve successful communication with existing CLIS or EHR systems, need to be considered when deciding upon the mapping language to be used. Concerning the specification of the correspondence between Global and Local EHR ontologies and in order to overcome the semantic and structural interoperability issues, we take advantage of the "Expressive and Declarative Ontology Alignment Language" EDOAL [16], which allows for representing correspondences between the entities of different ontologies. Its key strength and the main reason for its selection is the fact that it extends the ontology alignment format and it enables the representation of complex correspondences.

4 Challenges and Open Issues

As mentioned in Sect. 2, interoperability between clinical research systems and healthcare datasources comprises a great challenge due to the variety of heterogeneity issues primarily at the semantic level. The proposed approach followed within the PONTE project breaks down the process of communication with EHRs into different steps, each of which focuses on specific aspects of heterogeneity and aims at reducing the semantic distance between the eligibility criteria specified in clinical research and the healthcare patient data.

Among the greatest challenges met in this process rises from the fact that, although international vocabularies and classifications are being developed the past years, their adoption in the healthcare domain still remains limited. Mapping local vocabularies to the corresponding international ones, is a process which requires significant manual effort and involves experts both from the medical and the technical domains. A similar challenge is related to the structure of the EHRs. Despite efforts being made in the agreement on and provision of guidelines and specifications describing the information and data elements which should be recorded in an EHR, still great variations are found across EHRs in terms of parameters, their naming, structure, etc. In our approach, in order to reduce the effort required to perform the mapping between the Global EHR ontology and the ontology extracted from each EHR data and structure, one of the next steps in our work will involve the development of semi-automatic mechanisms allowing for the specification of the ontology mappings.

Moreover, given the wealth of eligibility criteria which are met across clinical trials and, consequently, a researcher would potentially specify, the enrichment of the Eligibility Criteria ontology and the Global EHR ontology comprises part of our ongoing work. This effort includes deep analysis of patterns and categories across eligibility criteria specified in trials registered in clinicaltrials.gov[2] and EU Clinical Trials Register[3] together with close interaction with the clinical experts of the PONTE project.

5 Conclusions

This paper presented an analysis of the different heterogeneity aspects met in the application of eligibility criteria in Electronic Health Records and Clinical Information Systems serving patient selection purposes for participation in clinical trials. The approach followed within the PONTE project for addressing the variety of the heterogeneity issues, including syntactic, structural, semantic ones, has been presented. In order to reduce the complexity of the process, a multi-step process has been adopted, with each step focusing on particular heterogeneity issues. At the same time each step aims at reducing the semantic distance between the eligibility criteria, expressed based on the semantics of the clinical research domain, and clinical care patient data, expressed based on the healthcare domain semantics. This approach requires an initial mapping process, which in cases that local vocabularies and health data structures are used, might become quite intensive. Hence, part of our future work will focus on building a tool allowing for the semi-automatic alignment of the Global EHR ontology and the produced local EHR ontologies. Moreover, in order to capture the richness of the eligibility criteria in clinical research, effort will be placed in further analysing the

[2] http://clinicaltrials.gov/.

[3] https://www.clinicaltrialsregister.eu/.

possible categories and parameters in inclusion and exclusion criteria and extending the Eligibility Criteria ontology as well as translating them into the respective EHR parameters.

Acknowledgements. This work is being supported by the PONTE project and has been partially funded by the European Commission's IST activity of the 7th Framework Programme under contract number 247945. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

References

1. Kraljevic, S., Stambrook, P.J., Pavelic, K.: Accelerating drug discovery. EMBO Rep. **5**, 837–842 (2004). doi:10.1038/sj.embor.7400236
2. Van den Haak, M.A., Sculthorpe, P.D., McAuslane, J.: New Active Substance Activities: Submission, Authorisation and Marketing 2001. CMR International, Epsom (2002)
3. Di Masi, J., Hansen, R., Grabowski, H.: The price of innovation: new estimates of drug development costs. J. Health Econ. **22**, 151–185 (2003)
4. Harper, M.: The Truly Staggering Cost of Inventing New Drugs (2012). http://www.forbes.com/sites/matthewherper/2012/02/10/the-truly-staggering-cost-of-inventing-new-drugs/
5. Sheth, A.: Changing focus on interoperability in information systems: from system, syntax, structure to semantic. In: Goodchild, M.F., Egenhofer, M.J., Fegeas, R., Kotman, C.A. (eds.) Interoperating Geographic Information Systems, pp. 5–29. Kluwer Academic Publishers, Norwell (1999)
6. Ghawi., R.: Ontology-based Cooperation of Information Systems, 15 March 2010
7. Sheth, A.P.: Changing Focus on Interoperability in Information Systems:From System, Syntax, Structure to Semantics. In: Goodchild, M., Egenhofer, M., Fegeas, R., Kottman, C. (eds.) Interoperating Geographic Information Systems, vol. 495, pp. 5–29. Kluwer Academic Publishers, Dordrecht (1999)
8. International Classification of Diseases (ICD). http://www.who.int/classifications/icd/en/
9. Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT). http://www.ihtsdo.org/snomed-ct/
10. NCI Metathesaurus. https://cabig.nci.nih.gov/tools/NCI_Metathesaurus
11. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: A. Gomez-Perz, M. Gruninger, H. Stuckenschmidt, and M. Uschold. (eds.) Workshop on Ontologies and Information Sharing, IJCAI 2001, Seattle, USA, pp. 309–327 (2001)
12. PONTE project. http://www.ponte-project.eu
13. Bizer, C., Cyganiak, R.: D2R server – publishing relational databases on the semantic web. In: Poster at the 5th International Semantic Web Conference (ISWC 2006) (2006). http://richard.cyganiak.de/2008/papers/d2r-server-iswc2006.pdf
14. Bizer, C., Seaborne, A.: D2RQ - treating non-RDF databases as virtual RDF graphs. In: ISWC 2004 (2004). http://www4.wiwiss.fu-berlin.de/bizer/pub/bizer-d2rq-iswc2004.pdf
15. Scharffe, F., Bruijn, J.A.: Language to specify mappings between ontologies. In: Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems (SITIS 2005), Yandoué, Cameroon. Dicolor Press, November 2005
16. Expressive and Declarative Ontology Alignment Language (EDOAL), available at:http://alignapi.gforge.inria.fr/edoal.html

A Linked Data Framework for Android

Maria-Elena Roşoiu, Jérôme David, and Jérôme Euzenat(✉)

INRIA and University Grenoble Alpes, Grenoble, France
{Maria.Rosoiu,Jerome.David,Jerome.Euzenat}@inria.fr

Abstract. Mobile devices are becoming major repositories of personal information. Still, they do not provide a uniform manner to deal with data from both inside and outside the device. Linked data provides a uniform interface to access structured interconnected data over the web. Hence, exposing mobile phone information as linked data would improve the usability of such information. We present an API that provides data access in RDF, both within mobile devices and from the outside world. This API is based on the Android content provider API which is designed to share data across Android applications. Moreover, it introduces a transparent URI dereferencing scheme, exposing content outside of the device. As a consequence, any application may access data as linked data without any a priori knowledge of the data source.

1 Introduction

Smartphones are becoming our main personal information repositories. Unfortunately, this information is stored in independent silos managed by applications, thus it is difficult to share data across them. One could synchronize application data, such as the contacts or the agenda using a central repository. However, these are not generic solutions and there is no mean to give access to data straight from the phone. The W3C Device API[1] covers this need across devices, but it offers specific APIs for specific applications, and not a uniform and flexible access to linked data. Nowadays, mobile operating systems, such as Android, deliver solutions to access application content, but they are restricted to some application database schemas that must be known beforehand.

Offering phone information as RDF data would allow application developers to take advantage of it without relying on specific services. Moreover, doing this along the linked data principles (use URIs, provide RDF, describe in ontologies, link to other sources) would integrate the phone information within the web of data and make it accessible from outside the phone. Our goal is to provide applications with a generic layer for data delivery in RDF. Using such a solution, applications can exploit device information in an uniform way without knowing, from the beginning, application schemas.

For example, an application may be used as a personal assistant: when one would like to know which of his contacts will participate to an event, he will

[1] http://www.w3.org/2007/uwa/Activity.html.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 204–218, 2015.
DOI: 10.1007/978-3-662-46641-4_15

consult the calendar of his contacts in order to retrieve the answer: are they participating to the event or will they be around the place? From data in linked data and the guest food preferences, it should be possible to select suitable nearby restaurants. Finally, from guest availability and restaurant opening hours, it could adequately plan for a meeting and deliver an invitation to these people. For sure, privacy and security concerns will have to be dealt with appropriately, but in a first step we are concerned by making these data available and interoperable.

The Android platform has several appealing features for that purpose:

- Applications are built and communicate in a service oriented architecture;
- Data sharing is built-in through the notion of ContentProviders.

We presented a first version of RDF content providers in [2]. This layer, built on top of Android content providers, allowed to share application data inside phones. In this paper, we extend it by adding capabilities to access external RDF data, and to share application data as linked data on the web. The mobile device information can then be accessed remotely, from any web browser, by any person who has been granted access to it. In this case, the device acts like a web server.

Early efforts were made to build an homogeneous XML repository from personal information [14]. Some pioneering attempts at marrying mobile information and semantic web technologies were made in [7], but do not aim to expose RDF data. The Nepomuk project[2] strived to produce RDF PIM ontologies, and to expose desktop data in RDF. OinkIt [6] exported phone contacts as FOAF files. OinkIt is restricted to contacts though Nepomuk covers a wide variety of PIM applications. Nepomuk replicates data in an RDF store and OinkIt generates a file, while we would rather only provide access to this data.

This paper is a comprehensive presentation of a framework for exposing Android application data as linked data. We first describe the context in which the Android platform stores its data (Sect. 2), and how it can be extended in order to integrate RDF (Sect. 3). Then, we present three types of applications that sustain its feasibility (Sect. 4): the first type of applications wrap several facilities of devices to expose their data as linked data, the second application allows one to annotate pictures stored inside the phone, and the last one is an RDF browser that acts like a linked data client. We then explain the behavior of a server which exposes phone information to the outside world (Sect. 5). We discuss some technical issues raised by this framework and the solutions we implemented for them (Sect. 6). Finally, we present future improvements and challenges in this field (Sect. 7).

2 The Android Architecture

Android is a Linux-based operating system for mobile phones. It allows for developing applications in Java [5,8] based on a service oriented architecture that we present here.

[2] http://nepomuk.semanticdesktop.org/.

2.1 Services and Intents

Android is built around different kinds of components that are provided by an application[3]:

- Activities are user interaction modalities (an application panel, a chooser, an alert);
- Services are processing tasks;
- Broadcast receivers are components that react to events;
- Content providers expose some data of an application to other applications.

An application implements any of these kinds of components. Applications and application components communicate through messages called "intent(s)". They are defined as:

```
Intent intent = new Intent( Action, Data );
```

such that Action is a Java like package name, e.g., `fr.inrialpes.exmo.rdfoid.GETRDF`, and Data can be anything, but would generally be a URI, e.g., `content://contacts/people/33`, and optionally a mime-type specifying the expected result. The intent must be called through:

```
startService( intent ).
```

The targeted component can be explicit, i.e., the components to deal with the request are explicitly identified, or Android can look for an application or component able to answer the Action on the Data, and pass it the call and the arguments.

2.2 Android Content Providers

Inside the Android system, each application runs in isolation from other applications. For each application, the system assigns a different and unique user. Only this user is granted access to the application data. This allows one to take advantage of a secure environment, however this tends to lock data in independent repositories, each of them with its own data representation. This prevents data sharing across applications.

Content providers overcome this drawback by enabling the transfer of structured data between device applications through a standard interface. This interface empowers one to query the data or to modify it.

A content provider[4] is a subclass of `ContentProvider` and implements the following interface:

```
Cursor query( Uri id, String[] proj, String select, String[] selectArgs, String orderBy )
Uri insert( Uri id, ContentValues colValueList)
int update( Uri id, ContentValues colValueList, String select, String[] selectArgs )
int delete( Uri id, String select, String[] selectArgs )
String getType( Uri id ) .
```

[3] http://developer.android.com/guide/topics/fundamentals.html.

[4] http://developer.android.com/guide/topics/providers/content-providers.html.

which allows one to query, to insert, to delete or to update the data. Queries are issued in an SQL manner and the results are returned as a cursor on a table.

Calling a `ContentProvider` is driven by the kind of content to be manipulated: the calling application indicates its desire to retrieve some content through its type and/or URI, but does not control which application will provide it. Android calls a `ContentResolver` which further looks into the query (the `id`) to find a suitable content provider on the phone for providing the required content. For that purpose, the resolver maps the query URIs to the declared providers. These providers are declared in application manifest file.

2.3 Android URIs

Android URIs have a specific structure:

`content://authority/path/to/data/optionalID.`

The `content` scheme indicates that the identified resource is delivered from a content provider, the `authority` identifies the provider, the `path/to/data` identifies a particular table, and the `optionalID` distinguishes a particular instance in the table, like in:

`content://contacts/people/33; content://fr.inrialpes.exmo.pikoid/picture/234.`

The URI `content://contacts/people` refers to all the people in the contact application, and the URI `content://contacts/people/33` identifies a specific instance of these, namely the instance having the id 33.

When an application requires access to a particular piece of data, it queries its URI. This is done through a request addressed to the `ContentResolver` which routes the query to the corresponding content provider.

The usage of URIs to identify data is a key strength from a linked data standpoint. However, these URIs use the specific `content` protocol instead of HTTP, and content providers do not return RDF.

Moreover, the URIs used by content providers are local to each device, i.e., not dereferenceable on the web, and not unique. This constraint is adequate only when the interface is used within the same Android device. However, when the interface connects to a wider context, this constraint does not maintain its validity: the `content://contacts/people/22` refers to different entries, i.e., contacts, stored inside the phone book agenda of different mobile devices.

3 The RDF Content Provider Framework

To allow Android applications to exchange RDF data, we need `ContentProviders` which deliver their data in RDF. For that purpose, we have designed an API which must be embedded inside applications that offer or access RDF content.

3.1 Framework Overview

In order to achieve this, we have primarily followed the same principles as the ones used in the original `ContentProvider` API. Therefore, our `RDFContentProvider` API delivers the following classes and interfaces:

- `RdfContentProvider:` An abstract class that should be extended if one wants to create an RDF content provider. It subclasses the `ContentProvider` class belonging to the Android framework;
- `RdfContentResolverProxy:` A proxy used by applications to send queries to the `RDFContentResolver` application. The `RDFContentResolver` application records all the RDF content providers installed on the device and routes queries to the relevant provider;
- `Statement:` A class used for representing RDF statements;
- `RdfCursor:` An iterator on a set of RDF statements;
- `RdfContentProviderWrapper:` A subclass of `RdfContentProvider` which allows for adding RDF content provider capabilities to an existing classical content provider.

Figure 1 gives an overview of the framework architecture.

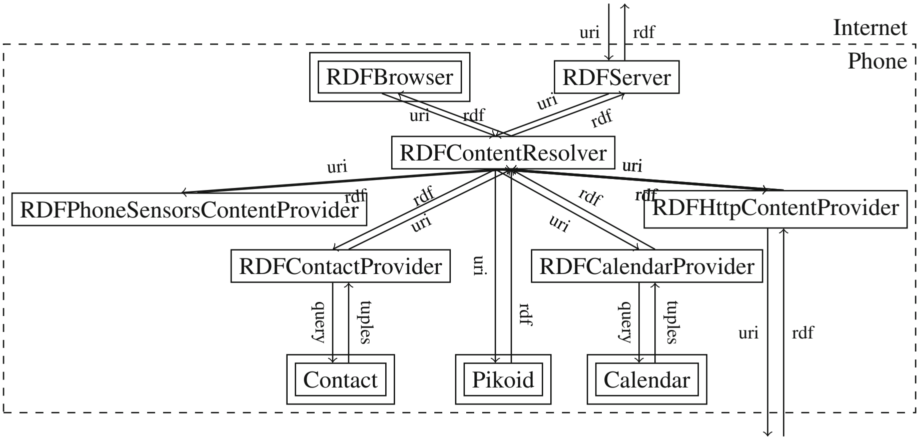


Fig. 1. The architecture components and the communication between them. Components with double square have a relevant graphic user interface.

The main components from a developer perspective are the `RDFContentProvider` API and the `RDFContentResolver` application.

3.2 The RDF Content Provider API

The goal of the `RDFContentProvider` API is to answer to two types of queries:

- Queries that request information about a particular individual, e.g., tell me what you know about contact 33. The provided answer is a set of triples which corresponds to the description of one object and its attribute values.
- Queries that request only the values for some variables that must satisfy a specific condition, i.e., SPARQL-like queries. In this case, the answer is a table of tuples, like in `ContentProviders` or SPARQL.

For the first type of queries we provide a minimal interface. This interface has to be implemented for linked data applications and has the following format:

- `RDFCursor getRdf( Uri id ).`

In this case, the cursor iterates on a table of subject-predicate-object (or predicate-object) which represents the triples involved in the description of the object given as a URI.

As for the second type of queries, they require a more elaborate semantic web interface, i.e., a minimal SPARQL endpoint. Thus, the following methods have to be also implemented:

- `Uri[] getTypes( Uri id ):` returns the RDF types of a local URI;
- `Uri[] getOntologies():` ontologies used by the provider;
- `Uri[] getQueryEntities():` classes and relations that can be delivered by the provider;
- `Cursor query( SparqlQuery query ):` returns tuple results;
- `Cursor getQueries():` triple patterns that can be answered by the provider.

The `RDFContentProviders` that we have developed so far only implement the first three primitives.

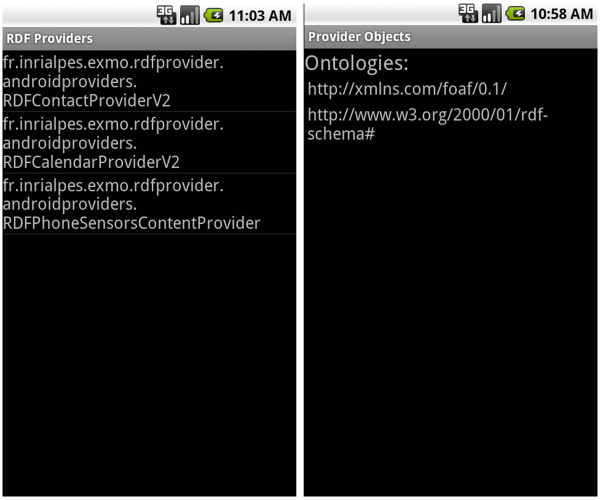
This interface corresponds to the one we required to web services in our work on ambient intelligence [4]. Indeed, to some extent this work is similar to the work in ambient intelligence except that instead of working in a building-like environment, it works within the palm of one's hand. But the problem is the same: applications which do not know each others can communicate through semantic web technologies.

3.3 The RDF Content Resolver Service

The `RDFContentResolver` service has the same goal as the `ContentResolver` belonging to the Android framework. It maintains the list of all installed `RDFContentProviders`, and forwards the queries it receives to the corresponding ones. Users do not have to interact with this application, therefore it is implemented as an Android service.

When an RDF content provider is instantiated by the system, a principle similar to the one from the Android content provider framework is used: this provider automatically registers to the `RDFContentResolver` (Fig. 2).

The `RDFContentResolver` can route both the local (`content:`) and external (`http:`) URI-based queries. In case of a local URI, i.e., starting with the `content` scheme, the resolver decides to which provider it must redirect the query. In case of an external URI, i.e., starting with the `http` scheme, the provider automatically routes the query to the `RDFHttpContentProvider` (see Fig. 1).



fr.inrialpes.exmo.rdfprovider.androidproviders.RDFContactProviderV2

fr.inrialpes.exmo.rdfprovider.androidproviders.RDFCalendarProviderV2

fr.inrialpes.exmo.rdfprovider.androidproviders.RDFPhoneSensorsContentProvider

Ontologies:

http://xmlns.com/foaf/0.1/

http://www.w3.org/2000/01/rdf-schema#

Fig. 2. RDFBrowser also allows for inspecting from the RDFContentResolver available RDFContentProviders and the ontologies they manipulate.

4 Applications

We developed a few applications as a proof of concept of this framework. They can be considered in different categories: wrappers for existing Android content providers and other services (Sect. 4.1), native applications (Sect. 4.2) and a client example for the framework (Sect. 4.3).

4.1 RDF Provider Wrappers for Phone Applications

The `RDFContentResolver` application is also bundled with several RDF content providers encapsulating the access to Android predefined providers. The Android framework has applications which can manage the address book and the agenda. These two applications store their data inside their own content provider.

In order to expose this data as RDF, we developed the `RDFContactProvider` and the `RDFCalendarProvider`. These providers are wrapper classes for the `ContactProvider` and the `CalendarProvider` residing inside the Android framework. The same could be obtained by bypassing the `ContentProvider` interface, and using instead the W3C Device APIs since they exist for each of these applications[5]. So doing should not be significantly more difficult and would ease porting to other platforms than Android.

`RDFContactProvider` exposes contact data using the FOAF ontology (Fig. 2). It provides data about the name of a person (display name, given name, family name), his phone number, email address, instant messenger identifiers, homepage and notes.

[5] http://www.w3.org/2009/dap/ or http://www.w3.org/TR/geolocation-API/.

`RDFCalendarProvider` provides access to Android calendar using the RDF Calendar ontology[6]. The data supplied by this provider is information about events, their location, their date (starting date, ending date, duration, and event time zone), the organizer of the event and a short description.

In addition to these content providers, two other RDF providers are the `RDFPhoneSensorsContentProvider` and `RDFHttpContentProvider`.

`RDFPhoneSensorsContentProvider` exposes sensor data from the sensors embedded inside the mobile device. Contrary to the others, they are not offered as content providers. At the present time, it only delivers the geographical position (retrieved using the Android LocationManager service). In order to express this information in RDF, we use the geo location vocabulary[7], which provides a namespace for representing lat(itude) and long(itude).

The `RDFHttpContentProvider` allows one to retrieve RDF data from the web of data. It parses RDF documents retrieved by dereferencing URIs through HTTP and presents them as `RDFCursors`. So far, only the minimal interface has been implemented, i.e., the `getRdf( Uri id )` method.

Developing a wrapper would consist, in general, of the following steps:

- Identify data exposed in the application content provider;
- Choose ontologies corresponding to this data;
- Provide a URI pattern for each ontology concept;
- Implement a dereferencing mechanism which, for each type of resource, extracts information from the content provider and generates RDF from this (generating URIs for related resources).

A native RDF content provider application may follow the same steps. However, it may be developed without any content provider. In this case, the analysis has to be carried out from the application data (or a corresponding API).

4.2 Pikoid

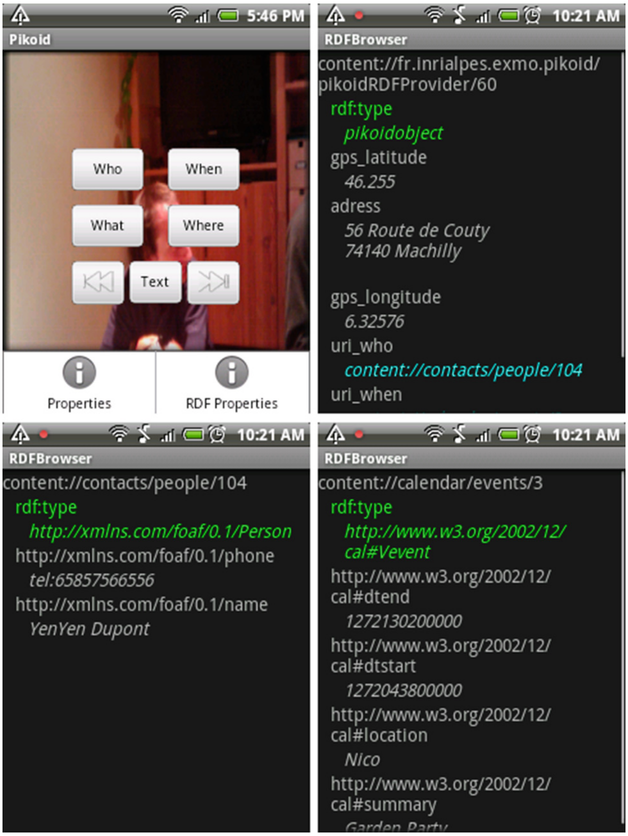
Pikoid is a native implementation of an RDF content resolver, i.e., an application that directly implements this interface.

It is a simple application allowing users to annotate pictures on the phone. The annotations answer the following simple questions: where and when (the picture was taken), who (is on the picture) and what (it represents). It is strongly integrated in the Android platform as it uses other content providers for identifying these annotations: people are taken from the address book, places from the map and events from the calendar.

Pikoid directly provides access to this data in RDF: each pikoid object offers these annotations as well as reference to the corresponding objects served by the wrapped content providers (`RDFContactProvider` and `RDFCalendarProvider`). Figure 3 illustrates browsing starting from Pikoid.

[6] RDF Calendar vocabulary: http://www.w3.org/TR/rdfcal/.

[7] Geo location vocabulary: http://www.w3.org/2003/01/geo/.



```
content://fr.inrialpes.exmo.pikoid/
pikoidRDFProvider/60
  rdf:type
    pikoidobject
  gps_latitude
    46.255
  adress
    56 Route de Couty
    74140 Machilly

  gps_longitude
    6.32576
  uri_who
    content://contacts/people/104
  uri_when
```

```
content://contacts/people/104
  rdf:type
    http://xmlns.com/foaf/0.1/Person
  http://xmlns.com/foaf/0.1/phone
    tel:65857566556
  http://xmlns.com/foaf/0.1/name
    YenYen Dupont
```

```
content://calendar/events/3
  rdf:type
    http://www.w3.org/2002/12/
    cal#Vevent
  http://www.w3.org/2002/12/
  cal#dtend
    1272130200000
  http://www.w3.org/2002/12/
  cal#dtstart
    1272043800000
  http://www.w3.org/2002/12/
  cal#location
    Nico
  http://www.w3.org/2002/12/
  cal#summary
    Garden Party
```

Fig. 3. The Pikoid application annotates images with metadata stored as RDF. RDFBrowser allows for querying this information to the Pikoid RDFContentProvider interface and displaying it. The current picture metadata is shown in the second panel (pikoidRDFprovider/60). From there, it is possible to browse the information available in the address book (people/104) and the calendar (events/3) through the corresponding RDF content providers wrapping them.

4.3 RDF Browser

The RDF Browser acts as a linked data client. Given a URI, either `http:` or `content:`, the browser issues a request to the `ContentResolver`. It then displays the resulting RDF cursor content as a simple page. If the data contains other URIs, the user can click on one of them and the browser will issue a new query with the selected URI.

An example can be found in Fig. 4. In this case, the user uses the `RDFBrowser` to get information about the contact having the id 4. When the browser receives the request, it sends it further to the `RDFContentResolver`. Since the URI starts with the `content://` scheme and has the `com.android.contacts` authority, the resolver routes the query to the `RDFContactProvider`. This provider retrieves the set of triples describing the contact and sends it to the calling application which displays it to the user. Thereupon, the user decides that he wants to continue browsing and selects the homepage of the contact. In this case, since the URI starts with the `http://` scheme, the resolver routes the query to the `RDFHttpContentProvider`. The same process repeats and the user can see the remote requested file, i.e., Tim Berners-Lee FOAF file.

5 RDF Server: Embedding a Phone in the Web of Data

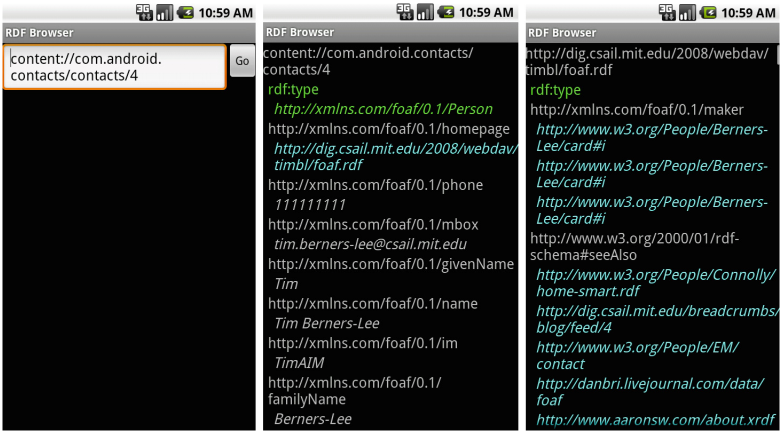
`RDFContentProviders` serve linked data within a single phone, `RDFServer` exposes this linked data to the wider web of data. This is based on three components:

- `RDFServer` is an RDF HTTP server that takes incoming HTTP queries (URIs) and returns RDF;
- the `RDFContentResolver` dereferences incoming URIs and externalizes local URIs within RDF;
- `RDFHttpContentProvider` allows for following HTTP URIs if necessary (see Fig. 1).

5.1 RDF Server

The `RDFServer` exposes the data stored into the device as RDF to the outside world. Because the server must permanently listen for new requests and does not require any user interaction, it is implemented as an Android service, i.e., a background process.

The server principles are quite simple. At launch time, it listens on port 80 for incoming requests. Once it receives a request from the outside, it dereferences the requested URI, i.e., it translates the external URI into an internal one, which

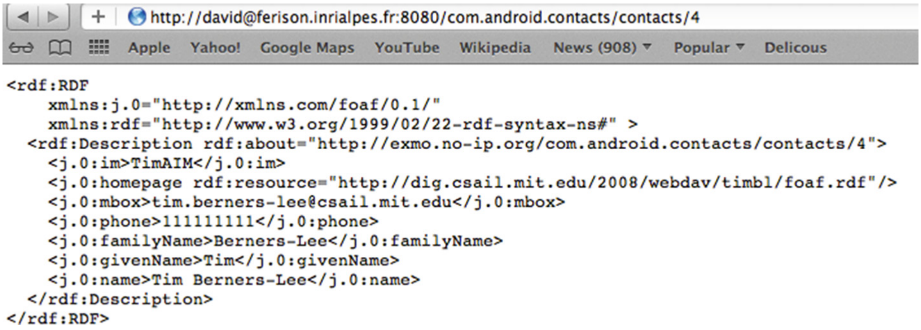


content://com.android.contacts/contacts/4

content://com.android.contacts/contacts/4
rdf:type
http://xmlns.com/foaf/0.1/Person
http://xmlns.com/foaf/0.1/homepage
http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf
http://xmlns.com/foaf/0.1/phone
111111111
http://xmlns.com/foaf/0.1/mbox
tim.berners-lee@csail.mit.edu
http://xmlns.com/foaf/0.1/givenName
Tim
http://xmlns.com/foaf/0.1/name
Tim Berners-Lee
http://xmlns.com/foaf/0.1/im
TimAIM
http://xmlns.com/foaf/0.1/familyName
Berners-Lee

http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf
rdf:type
http://xmlns.com/foaf/0.1/maker
http://www.w3.org/People/Berners-Lee/card#i
http://www.w3.org/People/Berners-Lee/card#i
http://www.w3.org/People/Berners-Lee/card#i
http://www.w3.org/2000/01/rdf-schema#seeAlso
http://www.w3.org/People/Connolly/home-smart.rdf
http://dig.csail.mit.edu/breadcrumbs/blog/feed/4
http://www.w3.org/People/EM/contact
http://danbri.livejournal.com/data/foaf
http://www.aaronsw.com/about.xrdf

Fig. 4. An example of using the RDF Browser for accessing remote RDF.


http://david@ferison.inrialpes.fr:8080/com.android.contacts/contacts/4
Apple
Yahoo!
Google Maps
YouTube
Wikipedia
News (908)
Popular
Delicous

```
<rdf:RDF
    xmlns:j.0="http://xmlns.com/foaf/0.1/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="http://exmo.no-ip.org/com.android.contacts/contacts/4">
    <j.0:im>TimAIM</j.0:im>
    <j.0:homepage rdf:resource="http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf"/>
    <j.0:mbox>tim.berners-lee@csail.mit.edu</j.0:mbox>
    <j.0:phone>111111111</j.0:phone>
    <j.0:familyName>Berners-Lee</j.0:familyName>
    <j.0:givenName>Tim</j.0:givenName>
    <j.0:name>Tim Berners-Lee</j.0:name>
  </rdf:Description>
</rdf:RDF>
```

Fig. 5. RDF Server response from externalized URIs.

has a meaning inside the Android platform. The `RDFServer` sends it further to the `RDFContentResolver`. In a manner similar to the one explained for the `RDFBrowser`, the set of triples is obtained. Before sending this set to the server, the URIs of the triples are externalized, i.e., transformed into `http:` URIs. The graph is then serialized using a port of Jena under the Android platform.

5.2 Dereferencing and Externalising URIs

One important issue appears when one want to get data from a device because the URIs used to query the content providers have a local meaning. URIs used to query the address book of two different devices are the same, but the content it identifies will likely be different.

The URI externalization process translates the local URI:

`content://authority/path/to/data`

into the dereferenceable one:

`http://deviceIPAddress:port/authority/path/to/data.`

Reversing the translation of such a URI is possible since both the authority and the path are preserved by the externalization process.

Usually, mobile devices do not have a permanent IP address and thus, the externalized URIs are not stable. To overcome this, a dynamic DNS client[8,9] may be used.

In addition, the server supports a minimal content negotiation mechanism. If one wants to receive the data in RDF/XML, it will set the MIME types of the Accept-type header of its request to “application/rdf+xml” or to “application/*”. In the opposite case or when the client sets the MIME type to “text/plain”, the data will be transmitted in the N-Triple format. Not only

[8] Dynamic DNS Client: https://market.android.com/details?id=org.l6n.dyndns\&hl=en.

[9] DynDNS: http://dyn.com/dns/.

the requester has the opportunity to express its preferences regarding the format of the received data, but the default format of the transmitted data can be specified in the server settings, as well the port on which the server can listen on and the domain name server for it.

An example can be found in Fig. 5. In this scenario, the user retrieves information about the fourth contact from the device address book. The request is processed by the RDF Server in a manner similar to the one of the RDF Browser.

6 Technical Issues: Application Size

The `RDFServer` included in our architecture eases the access of the user to the RDF data found on the web. For that purpose, we wanted to reuse an existing semantic web framework, such as Jena or Sesame. Yet these are not suitable to be employed under the Android platform (the code depends on some libraries that are unavailable under Android). There are a few ports of these frameworks to Android: Microjena[10] and Androjena[11] are ports of Jena and there exists a port of Sesame to the Android platform mentioned in [1]. We use Androjena.

A problem that arises when we use this framework is that the size of the application increases substantially. This is one of the constraints identified in [11]. This problem could have been avoided by reimplementing only the Jena modules that are needed in our architecture. Still, we would like to improve our architecture by adding more features (such as a SPARQL query engine) that require additional modules to those used to read/parse/write RDF, available in Jena.

We have used ProGuard for addressing this problem. ProGuard[12] is a code shrinker, optimizer, and obfuscator. It removes the unused classes, methods or variables, performs some byte-code optimizations and obfuscates the code. We only took advantage of the two former features. The tool proved to be efficient in reducing the size of our application (our framework including Androjena) by half, i.e., its initial size was 6.48 MB, and, after applying ProGuard, it was reduced to 3.15 MB. In Table 1, the effect of ProGuard can be observed on some of the applications that we developed.

Table 1. Size of applications with or without ProGuard.

Application	Size without ProGuard	Size with ProGuard
RDFContentResolver	6.49 MB	3.15 MB
RDFBrowser	368 KB	184 KB
RDFServer	100 KB	76 KB
Alignment API impl	254 KB	170 KB

[10] http://poseidon.ws.dei.polimi.it/ca/?page_id=59.

[11] http://code.google.com/p/androjena/.

[12] http://proguard.sourceforge.net/.

The existence of such tools as ProGuard, is a step forward in the continuous battle between applications that require a considerable amount of space for storing their code and devices with a reduced memory storage.

7 Perspectives

The current framework is only a first step towards a more comprehensive semantization of Android devices. Here are some further steps that we plan to take.

7.1 SPARQL Querying

One of these further steps would be to allow one to query the device data using SPARQL.

A double problem appears when one would like to achieve this: the distribution of the query across several content providers and its translation.

The distribution will require query partitioning and dispatching to different providers as performed in distributed query processing [10,13].

The translation of the query can be addressed in several manners:

- creating a new RDF content provider which relies on a triple store to deposit the data [9], and then using SPARQL to query it;
- translating SPARQL queries into more specific requests that may be answered by an RDF content provider;
- translating SPARQL queries into SQL queries and further decompose them into ones compatible with the ContentProvider interface.

Concerning the second option, there are several available tools that can make the translation from SPARQL to SQL, like Virtuoso or D2RQ. However, these tools solve only half of the problem because the SQL queries have to be adapted to the ContentProvider interface, i.e., the queries have a particular format, different from SQL. This interface allows for querying only one view of a specified table at a time, hence it is not possible to ask content providers to perform joins.

7.2 Query Mediation

Once one is able to query data, the heterogeneity of the ontology used by providers may be a problem. Overcoming this requires mediating queries, i.e., transforming query expressed into one ontology in another query expressed with an ontology understood by a content provider. For that purpose, we plan to use ontology alignments. We already provide a micro version of the Alignment API[13] [3] working under Android and able to retrieve alignments from an Alignment server.

[13] http://alignapi.gforge.inria.fr.

7.3 Security and Privacy

Challenges regarding security must be taken into account. The user of the application should be able to grant or to deny access to his personal data. A specific vocabulary, such as the one introduced in [12], should be used in order to express this. Moreover, the dangers of granting system access to a third-party user can be avoided by using a secure authentication protocol[14].

7.4 Resource Consumption

Finally, the problem of resource consumption is mentioned here for the record. Such resources may be related to bandwidth (WiFi or 3G) that are consumed by having the RDFServer working. In addition, such a server, and the use of our framework in general, may affect energy consumption. This will have to be precisely considered.

8 Conclusion

Involving Android devices in the semantic web, both as consumers and providers of data, is an interesting challenge. As mentioned, it faces the issues of size of applications and URI dereferencing in mobility situations. There remain other technical problems in implementing a full Android RDF framework encompassing distributed SPARQL querying.

So, our next step is to provide a more fine grained and structured access to data through SPARQL querying. This promises to raise the issue of computation, and thus energy, cost on mobile platform.

A further issue is the control of privacy in such a framework. In this particular domain too, we think that semantic technologies can provide more flexible and targeted solutions.

The framework and applications described here are available at http://swip.inrialpes.fr.

Acknowledgements. We thank Yu Yang and Loïc Martin who have programmed part of the Pikoid application. This work has been partially supported by the French National Research Agency under grant ANR-10-CORD-009 (Datalift).

References

1. d'Aquin, M., Nikolov, A., Motta, E.: Building SPARQL-enabled applications with android devices. In: Proceedings of the 10th ISWC Demonstration Track, Bonn (DE) (2011)

[14] http://www.w3.org/wiki/WebAccessControl and http://www.w3.org/wiki/Foaf+ssl.

2. David, J., Euzenat, J.: Linked data from your pocket: The Android RDFContent Provider. In: Proceedings of the 9th ISWC Demonstration Track, Shanghai (CN), pp. 129–132 (2010)
3. David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The alignment API 4.0. Semant. Web J. **2**(1), 3–10 (2011)
4. Euzenat, J., Pierson, J., Ramparani, F.: Dynamic context management for pervasive applications. Knowl. Eng. Rev. **23**(1), 21–49 (2008)
5. Gargenta, M.: Learning Android. O'Reilly Media Inc, Sebastopol (2011)
6. Lassila, O.: Semantic web approach to personal information management on mobile devices. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC), Santa Clara (CA US), pp. 601–607 (2008)
7. Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S.: Situational reasoning for task-oriented mobile service recommendation. Knowl. Eng. Rev. **23**(1), 7–19 (2008)
8. Meier, R.: Professional Android 2 Application Development. Wrox, Birmingham (2011)
9. Le Phuoc, D., Parreira, J.X., Reynolds, V., Hauswirth, M.: RDF on the go: an RDF storage and query processor for mobile devices. In: Proceedings of the 9th ISWC Demonstration Track, Shanghai (CN), November 2010
10. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 524–538. Springer, Heidelberg (2008)
11. Rietveld, L., Schlobach, S.: Semantic web in a constrained environment. In: Proceedings of the ESWC Downscaling the Semantic Web Workshop, Heraklion (GR) pp. 31–38 (2012)
12. Sacco, O., Passant, A.: A privacy preference ontology (PPO) for linked data. In: Proceedings of the WWW Linked Data on the Web Workshop (LDOW 2011), pp. 1–5 (2011)
13. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: optimization techniques for federated query processing on linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 601–616. Springer, Heidelberg (2011)
14. Walsh, N.: Generalized metadata in your Palm. In: Proceedings of the 2nd Extreme Markup Languages Conference, Montréal (CA) (2002)

The Web of Radios - Introducing African Community Radio as an Interface to the Web of Data

Anna Bon[1](✉), Victor de Boer[1], Pieter De Leenheer[1], Chris van Aart[1],
Nana Baah Gyan[1], Max Froumentin[2], Stephane Boyera[2],
Mary Allen[3], and Hans Akkermans[1]

[1] Network Institute, VU University, Amsterdam, The Netherlands
{a.bon,v.de.boer,pieter.de.leenheer,c.j.van.aart,n.b.gyan}@vu.nl,
hans.akkermans@akmc.nl

[2] World Wide Web Foundation, Sophia Antipolis, France
{maxf,boyera}@webfoundation.org

[3] Sahel Eco, ACI 200 Rue 402, 03 BP 259, Bamako, Mali
mary.saheleco@afribonemali.net

Abstract. The World Wide Web as it is currently deployed can only be accessed using modern client devices and graphical interfaces, within an infrastructure compassing datacenters and reliable, high-speed Internet connections. However, in many regions in developing countries these conditions are absent. Many people living in remote rural areas in developing countries will not be able to use the Web, unless they can produce and consume voice-based content using alternative interfaces such as (2G) mobile phone, and radio. In this paper we introduce a radio platform, based on a use case and requirements analysis of community radio stations in Mali. The voice-based content of this radio platform will be made publicly available, using Linked Data principles, and will be ready for unexpected re-use. It will help to bring the benefits of the Web to people who are out of reach of computers and the Internet.

Keywords: Community radio · Voice-based interfaces · Web of Data · Radio platform

1 Introduction

The World Wide Web is perfectly adapted for use by people in developed countries. It is visual, text-based, and mainly written in English or other world languages[1]. The Web depends on the availability of computers, datacenters, glass fiber backbones, fixed and wireless networks, 3G mobile telephony and transport of large volumes of data at high speed. In remote rural areas in many developing countries, conditions are different. Poor infrastructure, lack of equipment, low

[1] Wikipedia, http://en.wikipedia.org/wiki/Global_Internet_usage, Global Internet Usage.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 219–232, 2015.
DOI: 10.1007/978-3-662-46641-4_16

levels of literacy, and use of under-resourced local languages, seriously hamper the access to the Web for many people.

There is a general consensus that the global Information Society must benefit all people in the world. The United Nations Millennium Declaration contains a commitment for developing a *people-centered, inclusive and development-oriented Information Society so that people everywhere can create, access, utilize and share information and knowledge to attain the internationally agreed development goals and objectives, including the Millennium Development Goals.*[2]

Yet, in many rural regions in Africa community radio is the only source of information. People have radios at home and listen to programs broadcast in local languages every day. Many people have access to simple voice-based (2G) mobile phone, but text messaging is hardly used [2].

The availability of both mobile phone and radio is opening opportunities for new services. E.g. radio listeners phone to the radio station and leave voice messages that they want to have broadcasted, or react to popular radio programs leaving news, opinion, regional information etc. Community radio here operates as an important local information hub, where people bring information for further dissemination.

Radio stations in rural areas in Africa operate under harsh conditions. Only the largest and state financed radio stations have a computer and an internet connection. Due to lack of funds many radio stations still use old-fashioned, analogue equipment, such as tape recorders. Yet, it is in the line of expectation that more and more radio stations will have computers and an internet connection in the coming years.

In the current situation the information broadcast by the community radio is volatile: it is not stored and kept for later access or re-use. Radios do not have means to manage, reuse and index this voice-based content.

In this paper we introduce a radio platform as a new interface to the Web. It enables management of radio content in an efficient way, making it accessible and searchable, so that it can serve a broad audience, e.g. Africans in the diaspora, who want to have news from their home villages[3].

Additional, the voice-based radio content on this radio platform might be linked to other data sources on the Web, enabling community radios in Africa to become an interface to the Web of Data. An example of a system that manages market information based on Linked Data principles and produces voice-output as broadcasts for African community radios, is described by De Boer et al. [3]. In the future new applications providing locally relevant information from the Web of Data, such as pluviometric data, agricultural data, market prices etc. might become available through the radio platform.

[2] UNMD, United Nations Millenium Declaration, General Assembly resolution 55/2. United Nations, New York, 2000.

[3] Communication possibilities with people living in the diaspora, are described by Serigne Mansor Tall in: Les émigrés sénégalais et les nouvelles technologies de l'information et de la communication. http://www.unrisd.org.

The radio platform described in this paper not only facilitates production, consumption and management of voice-and web-based radio content, but it also enables access to the Web for people who do not have a computer or the Internet.

Contributions of this paper are:

- A radio platform with both a web and a voice-based mobile interface that allows content creation, retrieval and indexing of spoken radio content.
- African community radio, introduced as a new interface to the Web of Data

This paper is structured as follows. In Sect. 2 we describe related work. In Sect. 3 the architecture of the radio platform is described, the use cases collected from three different radios in Mali, as well as the principles used to manage the content. In Sect. 4 we describe challenges related to the organization of the voice-based radio content. In Sect. 5 we discuss future work that must be done on the Web of Radios, including the sustainability aspects.

2 Related Work

Related work on the development of a similar platform was done in the Freedom Fone[4]. Freedom Fone is a project initiated by The Kubatana Trust of Zimbabwe, a civil and information activist platform from Zimbabwe. Freedom Fone is open source software for creating audio content using phone. Freedom Fone provides a voice platform similar to the basic setup proposed in this paper, but without the Linked Data enabled data management.

Research on speech recognition started in the 1930s and resulted in commercial deployments of voice-based services in the 1970s. Major achievements on language recognition, mainly for English, took place in the 1980s and 1990s and culminated in the development of VoiceXML by the W3C Voice Browser group, in 1999, facilitating and standardizing the development of voice applications [4].

Sheetal Agarwal et al. from IBM Research India, developed a system to enable authorship of voice content for 2G phone in a web space, they named the WWTW or World-Wide Telecom Web. The system is not connected to the Web, therefore not allowing access by third party search engines. The system represents a closed web space, within the phone network. Especially the lack of open search possibility constrains its growth [5].

From Burundi a system has been reported [6] to use tagging software and multimedia mobile data collection. The software is named EthnoCorder[5]. The NGO that co-developed this app was Help Channel Burundi. However, because of the current unavailability of multimedia devices in the given rural context, this technical solution may be still out of reach of community radio stations targeted in this study.

A related project on the Semantic XO and Linked Data for developing countries is described by Guéret et al. [7]. The Semantic XO is a system that connects

[4] Freedom Fone, http://www.freedomfone.org.

[5] http://www.ethnocorder.com/.

rugged, low-power, low-cost robust small laptops (aka the XO promoted by the One Laptop Per Child organization) for the empowerment of poor communities, based on Linked Data principles in order to publish previously unpublished data.

De Boer et al. [3] describe a distributed voice- and web-based market information system, named Radio Marché, aimed at stimulating agricultural trade in rural areas of Africa. This system connects to regionally distributed market information systems, using Linked Data approaches.

3 The Radio Platform

The design of the radio platform described in this section is based on extensive use case and requirements analysis, performed in Mali, with the collaboration of radio journalists from community radio stations. The research was done as part of the Foroba Blon[6] project[7], funded by the International Press Institute, and the VOICES project[8], partially funded by the EU, within the 7th Framework Programme. The Foroba Blon project is aimed at supporting and promoting citizen journalism in developing countries. The VOICES project is aimed at developing innovative mobile voice services to support users in underprivileged communities in African countries (Fig. 1).

3.1 Operation of Community Radios in Mali

In Mali many community radio stations exist. Some are state funded and connected to the national broadcasting service ORTM (Office Radio Télévision du Mali). Others are privately funded or completely self-supportive. According to their business, funding scheme, size and location some radio stations do have computers and internet, some have computers without internet connection and some do not have any computer facilities at all. All these radio stations are situated within the coverage area of mobile telephony.

The Malian community radios have large bases of listeners and the radius of coverage ranges between 100 and 200 km. These radio stations create their own programs and broadcast local and regional news, music, informative programs, round table programs and paid announcements. Two radio station stations are involved in the projects described in this paper. These are: Radio ORTM Ségou, a state owned radio, that has computers and a 2 Mbps fixed line (DSL) internet connection. Radio ORTM Ségou broadcasts programs in French and Bambara, the most widely spoken language in Mali.

[6] Foroba Blon in Bambara language refers to a large space, where everyone has the right to speak in front of the village chief; the truth must be told here, but only respectfully, without insulting anyone.

[7] Foroba Blon, Citizen Journalism: http://worldplantage.wordpress.com/2012/01/14/community-radio-in-tominian-and-segou-mali/ and http://www.ipinewscontest.org/news/foroba-blon-plans-to-revolutionise-journalism-in-mali.html.

[8] VOICES: http://www.mvoices.eu.

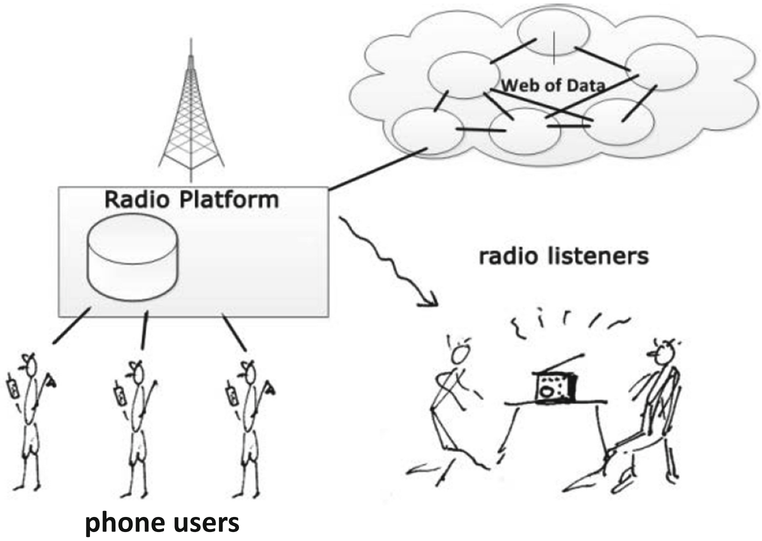


Fig. 1. Conceptual design of the radio platform as a voice-interface to the Web for people who are out of reach of computers and the Internet, but do have phone or radio.

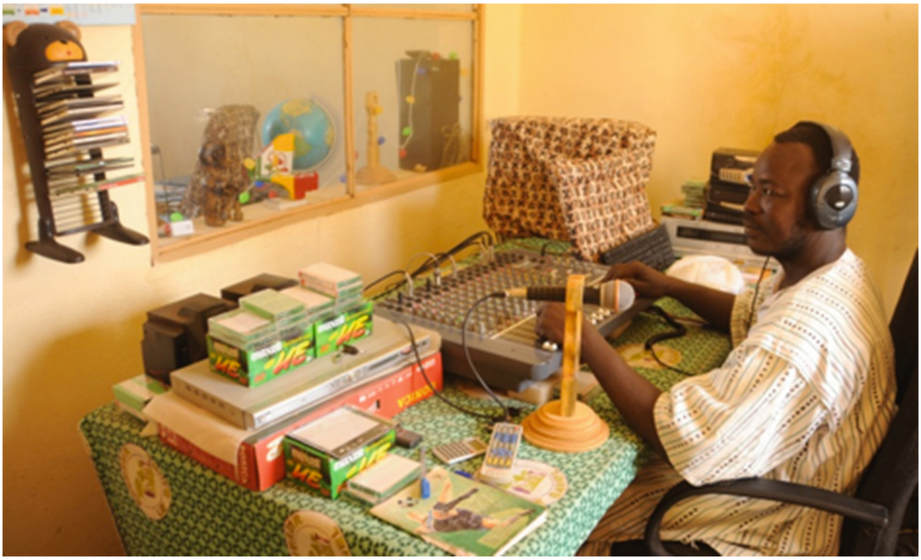


Fig. 2. Presenter Radio Moutian.

The second radio station is Radio Moutian, in Tominian, see Fig. 2. This radio is independent and its funding is based on paid airtime for announcements and private gifts from third parties. Radio Moutian has a computer but no internet connectivity. Programs are mainly broadcasted in Bomu, a local language fro the Tominian region. The third radio is Radio Seno in Bankass. This radio is independent from the Malian state and has only analogue equipment. There are no computers, there is no internet connection here, but the radio has many listeners in the region around Bankass. The main language spoken here is Dogon. The activities of the three above mentioned radio stations are related to three types of end-users or customers:

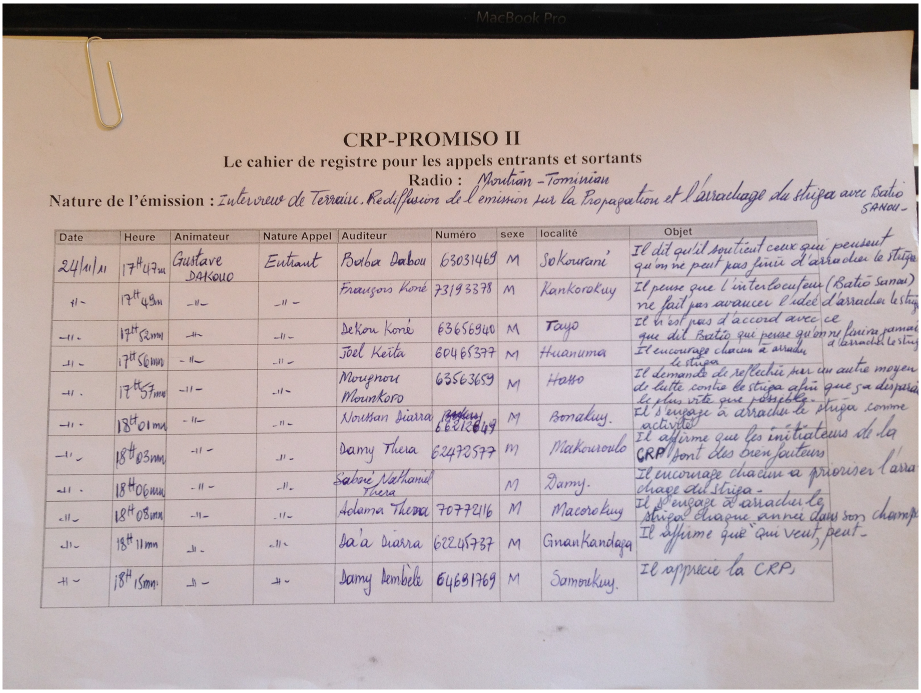
- NGOs that buy airtime to broadcast public announcements about informative and educational topics, such as agriculture and public health information. This type of service is usually based on fixed monthly subscriptions to airtime for recurring broadcasts.
- Non-commercial listeners from the region, who buy a few minutes of airtime and pay a broadcast fee per minute airtime. The information is usually brought to the radio, or communicated via phone and subsequently written down on paper by the radio staff. Some listeners call in on a given time slot (one hour per week) and leave a short voice message (few seconds only) as a reaction to a program that was broadcast on a certain popular topic. These messages are named letters to the editors (LTE).
- Journalists or trusted village reporters that phone to the radio and leave local news or interviews on a regular base. In the current situation, all incoming phone calls are attended by a radio staff member and annotated in tabular form on paper.

3.2 The Radio Platform Architecture

The proposed radio platform, which we named Foroba Blon (FB), consists of a data store containing recorded voice messages and related meta-information. FB will replace the existing caller log, see Fig. 3.

The interface to the FB radio platform for entering new content is either through mobile phone or via the web. Users of the mobile interface are the listeners from the region, who enter letters to the editor (LTE). These people only have mobile phones but no access to the Internet. Their calls are answered by the system with a pre-recorded welcome message in a local Malian voice inviting them to leave their message. For the sake of user-friendliness, the user interface and the dialogue for this category of users is kept as short and simple as possible, since the expected callers will be unfamiliar with interactive voice response systems and may not respond to a complex computer-generated dialogue asking to press buttons, etc.

Another category of users of FB are the trusted reporters calling from the field, and also using the mobile interface. They phone in and leave their spoken report for broadcasting. These users are previously registered, having their phone



CRP-PROMISO II

Le cahier de registre pour les appels entrants et sortants

Radio : Moutian - Tominian

Nature de l'émission : Interview de Terrain. Rediffusion de l'emission sur la Propagation et l'arrachage du striga avec Batio SANOU -

Date	Heure	Animateur	Nature Appel	Auditeur	Numéro	sexe	localité	Objet
24/11/11	17H47mn	Gustave DAKOUO	Entrant	Baba Dabou	63031469	M	Sokourani	Il dit qu'il soutient ceux qui pensent qu'on ne peut pas finir d'arracher le strig
-"-	17H49mn	-"-	-"-	François Koné	73193378	M	Kankorokuy	Il pense que l'interlocuteur (Batio Sanou) ne fait pas avancer l'idée d'arracher le strig
-"-	17H52mn	-"-	-"-	DeKou Koné	63656940	M	Tayo	Il n'est pas d'accord avec ce que dit Batio qui pense qu'on ne finira jamai d'arracher le strig
-"-	17H56mn	-"-	-"-	Joel Keita	60465377	M	Huanuma	Il encourage chacun à arracher le striga
-"-	17H57mn	-"-	-"-	Mougnou Mounkoro	63563659	M	Hasso	Il demande de reflechir sur un autre moyen de lutte contre le striga afin que ça disparai le plus vite que possible.
-"-	18H01mn	-"-	-"-	Noussan Diarra	~~Bonakuy~~ 66212649	M	Bonakuy.	Il s'engage à arracher le striga comme activité
-"-	18H03mn	-"-	-"-	Damy Thera	62472577	M	Makouroulo	Il affirme que les initiateurs de la CRP sont des bien fauteurs
-"-	18H06mn	-"-	-"-	Sabaré Nathaniel Thera		M	Damy.	Il encourage chacun à prioriser l'arrachage du striga -
-"-	18H08mn	-"-	-"-	Adama Thera	70772116	M	Macorokuy	Il s'engage à arracher le striga chaque année dans son champ
-"-	18H11mn	-"-	-"-	Da'a Diarra	62245737	M	GnanKandaga	Il affirme que "qui veut, peut" -
-"-	18H15mn	-"-	-"-	Damy Dembele	64631769	M	Samoukuy.	Il apprecie la CRP,

Fig. 3. Current situation: incoming messages are registered in a caller log.

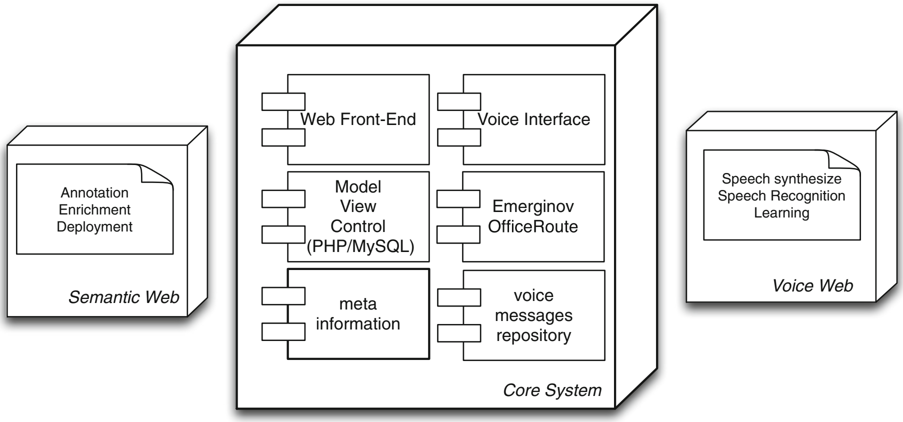


Fig. 4. Architceture of Foroba Blon, showing the core system and its connections to the Semantic Web and the Voice Web

number, name, address and preferred language in FB. These users will be trained to navigate the voice-menu, and use the IVR system, asking to press a button on the phone to confirm or answer a question about their current location, subject of the message, etc. The FB system always answers the registered caller in his/her preferential language (Fig. 4).

The voice messages are stored as audio files in the FB data store, together with meta-information being the date and time of the call, the length of phone call in seconds, the phone number of the caller. Messages from trusted users are linked to the owner, his/her address, and his/her preferred language. For all users of the system, confidentiality and anonymity will be ensured, according to the broadcast policies used by the radio stations in Mali.

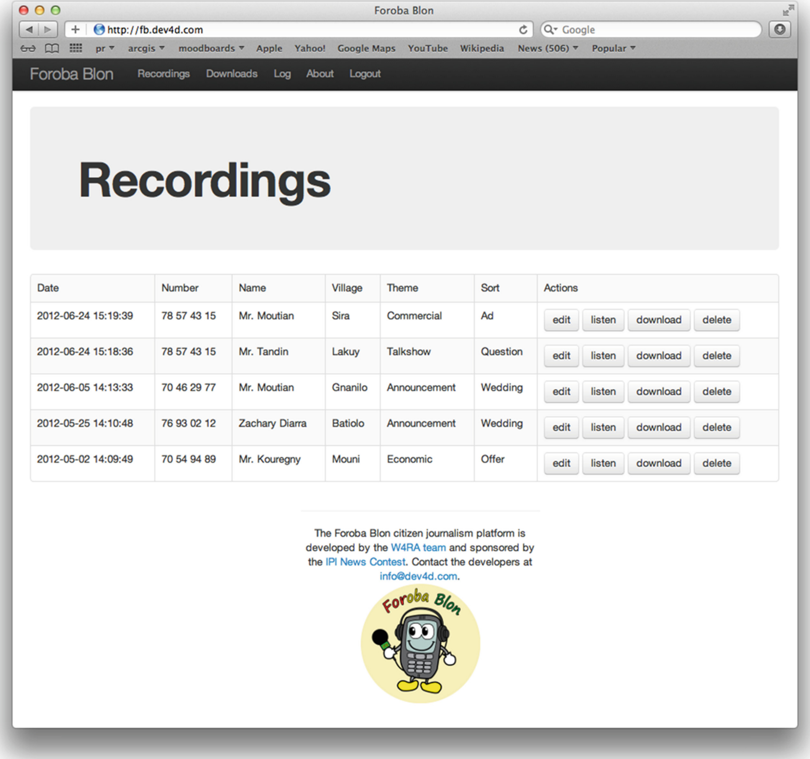
The FB radio platform also has a normal web interface, where internet-connected end-users/customers can access and upload a voice message. Depending of their customer relationship to the radio, they can login to the radio-platform as (i) registered users such as NGOs, and trusted reporters, or (ii) as unregistered users. There is an option to sign up and create a user account by registering the name, phone number, village and preferred language. Unregistered users can access former broadcasts since this is public information.

For the radio user, FB provides a web-based interface, enabling them to manage the data in the data store. It provides a file list where they can access, listen, broadcast, delete files, and add/update/delete meta-information, see Fig. 5.

The radio station that has no computer nor internet, only has a limited interface to the radio platform, since this is the constraint of a voice interface. The radio user receives a welcome message asking if she wants to hear the last ten messages, or if she wants to manage the welcome messages to the end-users. The FB radio platform is hosted either locally, on a stand-alone computer, or *in the cloud*. The FB consists of a voice platform running an open source web server and a local voice browser that handles the voice interaction. The FB radio platform uses a GSM gateway device, e.g. OfficeRoute,[9] a device that handles incoming and outbound calls and streams the voice messages to and from the phone.

The FB radio platform could in theory be physically hosted anywhere in the world, on any web server, connected to the Internet. However, in this actual case in rural regions of Mali, this is not possible. Firstly, the radio platform has to be accessible using an inexpensive local Malian phone number, so it must be connected to a Malian phone network. Secondly, the web service accessed over the Internet must also be accessible locally. Since the internet connectivity in Mali is usually of low bandwidth and of high latency, voice web services hosted in datacenters in the US or Europe, are too slow for proper deployment in Mali. For these two reasons, the system has to be preferably hosted locally in Mali. In the absence of good and reliable datacenters or hosting providers in Mali, the radios can decide to deploy the FB radio platform on a local computer at their own premises. Obviously before this can be done, the radio staff members have to be trained how to

[9] OfficeRoute: http://en.flossmanuals.net/freedom-fone/connecting-officeroute.



Recordings

Date	Number	Name	Village	Theme	Sort	Actions
2012-06-24 15:19:39	78 57 43 15	Mr. Moutian	Sira	Commercial	Ad	edit listen download delete
2012-06-24 15:18:36	78 57 43 15	Mr. Tandin	Lakuy	Talkshow	Question	edit listen download delete
2012-06-05 14:13:33	70 46 29 77	Mr. Moutian	Gnanilo	Announcement	Wedding	edit listen download delete
2012-05-25 14:10:48	76 93 02 12	Zachary Diarra	Batiolo	Announcement	Wedding	edit listen download delete
2012-05-02 14:09:49	70 54 94 89	Mr. Kouregny	Mouni	Economic	Offer	edit listen download delete

The Foroba Blon citizen journalism platform is developed by the W4RA team and sponsored by the IPI News Contest. Contact the developers at info@dev4d.com.

Fig. 5. Screenshot of the web-based interface of Foroba Blon for managing audio content generated by citizens

do operational maintenance of the FB platform, and especially how they can cope with frequent power outages, and bring the system back to a consistent state.

3.3 Semantic Datamodel

Figure 6 shows the semantic model used in Foroba Blon system. We use '`fb`' as abbreviation of the Foroba Blon namespace. We reuse existing schemas Dublin Core (`dct`) and FOAF (`foaf`) as well as the GeoNames dataset (`geonames`).

The main class is the `fb:recording` class. An instance of this class is linked to the URI of the actual audio file using the `fb:audiofile` property. The recording class has two literal properties: `dc:created` for the recording time and the `fb:caller_id` which is associated to the phone number of the caller. Both are extracted automatically by the system at recording time.

A recording is related to a specific radio station, which has a phone number predicate and uses FOAF properties to list its name and location. For the location, we use GeoNames entities where possible. If specific villages are not present in GeoNames, we provide our own resources, mapped to the GeoNames hierarchy.

A recording is also related to an instance of `fb:Annotation`. This instance holds all the information added by an radio station editor. The current version of the interface allows the editor to enter free text annotations in a number of fields. Through this, the system will gather persons, subjects, locations, products etc. associated with the radio message. We will enrich this free text data and map it to structured vocabularies, to be curated by the radio editors. Specific type of concepts will be linked to existing linked data sources (places to GeoNames[10], product types to Agrovoc[11] etc.). Figure 6 shows the design of the semantic model where this is the case.

The annotation lists the caller as a `foaf:Person` as well as the location from which the call originates as a GeoNames resource. The Dublin Core `dct:type` property is used to denote the type of the call (e.g., Announcement, Request, News Item, ...), which we model as SKOS concepts. The property `dct:subject` is used to further classify the subject of a call. This might be a literal value or (as is the case in Fig. 6, a SKOS concept. The radio editor is noted as the `fb:annotator` of the annotation which is also linked to a radio station. The annotation also has a separate automatically stored creation `datetime`. Lastly, the `fb:comment` predicate is used to link free text comments to the recording.

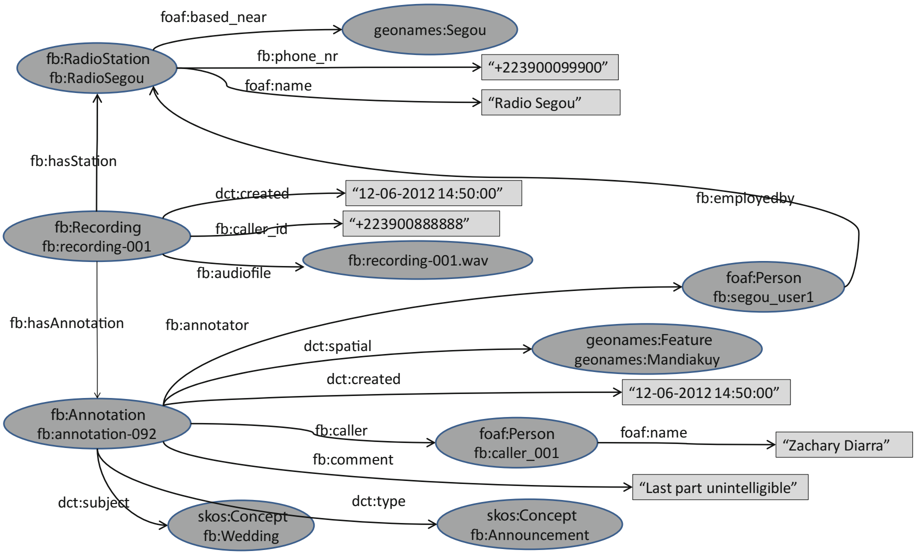


Fig. 6. Semantic Datamodel as used in the Foroba Blon system. The image shows a part of the semantic graph. Ellipses denote resources with their classes (italicized).

[10] http://www.geonames.org.

[11] FAO's food and agriculture thesaurus (http://fao.org/agrovoc).

4 Organizing the Radio Content

The next challenge is how to manage the spoken content of un-resourced languages such as Bomu, Bambara and Dogon. Since up to present no interactive voice response (IVR) systems exist for these languages, the voice-based content cannot be indexed by conventional search engines. Therefore collecting as much meta-information as possible is essential. Very simple ways of indexing the messages are based on owner (known through phone number) automatic language recognition, time slot, (e.g. *all messages collected on January 13 between 10 and 11 a.m. are related to the radio program on harvesting shea nuts).* The radio journalist can manually enter meta-information such as keywords, village region, language, name or any other attribute to an audio file using her radio-web interface. In the future existing tagging systems such as EthnoCorder may be considered, to facilitate meta-data collection.

4.1 Linked Data Sharing and Re-use

Initially, the data of the radio platform will be only used locally, but we explicitly designed the semantic datastore to ensure sharing and reuse of the data collected in the different Foroba Blon instances. We specifically envision reuse (1) across different instances of Foroba Blon (for different radio stations); (2) across information services, specifically the RadioMarché platform and (3) through aggregations of data, usable by third parties.

1. We are now in the process of installing Foroba Blon in two radio stations in the Tominian region in Mali. For specific recording types and for specific themes, sharing submissions can be of great value. Specifically, we envision that the sharing and spreading of local news reports can provide an efficient form of citizen journalism. The spreading of spoken news items in this way is akin to that of Twitter messages. In the region there are a great number of languages and dialects (57 in Mali alone). Linked Data is specifically well-suited to deal with multiple languages as its core concepts are resources rather than textual terms. A single resource, identified by a URI (i.e. http://example.org/shea_nuts) can have multiple labels (e.g. Shea Nuts@en and Amande de Karité@fr).
2. The Foroba Blon data will be linked to the data of Radio Marché. This platform is based on the same technology set and linked data principles as Foroba Blon to share and spread market information. As Foroba Blon will target mainly rural agricultural areas, we expect that much of the information and the stakeholders will overlap with that of RadioMarché. Specifically, the data will be linked through places (GeoNames), people (FOAF) and themes (SKOS concepts). These links can then be exploited by both platforms to enhance or augment the information provided. RadioMarché and Foroba Blon are only two examples. We are working towards an ecosystem of commercial and journalistic services running on the same radio platform where data is shared and reused across services.

3. Voice recordings can not only be shared across radiostations but also with other datasources and applications on the web. One possibility we are pursuing in the project is to open up certain messages to web users world wide through a web log. On this blog, people with access to Web infrastructure can listen to the journalist reports or announcements. Specifically, we foreseee that expatriates originating from the radio station's area will be interested in reports from their region of origin.

By exposing the produced data as linked data, we do not only open the possibility of expected, but also unexpected reuse. At all times, privacy issues related to the voice messages will be taken into account.

If the Foroba Blon radio platform proves to be a success, other instances of Foroba Blon may be installed at local radio stations in Mali, across borders, in neighbouring countries, Burkina Faso, Ghana, Senegal, Guinée where conditions with regards to illiteracy, local languages, mobile telephony and community radio are similar to those in Mali. This will create a Web of African community radios that are linked to each other and that will eventually become part of the Web of Data.

4.2 Organize an Open Source Community of Developers to Create Applications for the Radio Platform

In the VOICES and Foroba Blon projects one instance of the radio platform is developed by a small team of developers, in collaboration with end-users[12] sponsored by the International Press Institute as a pilot project. However, to enable further development of the radio platform, and to expand the scale of the web of radios, it is important to look at new ways of production and consumption of data and services. African community radios operate in a low-income region where the sustainability of a system relies on the underlying business model. Community radios do not have enough earnings to invest in new systems, and their listeners-base is large, but poor. Application development will therefore to be organized in a cost-effective way. We propose to organize an open source community of developers and to rely on commons-based peer production for the development of applications that will open the Web of Data to radio using voice-modality.

5 Discussion and Future Work

From this paper it becomes clear that the Web of African Radios can only emerge as an interface to the Web of Data, when sufficient applications are built, that link voice-based content. For the navigation of voice menus and other voice-based dialogues small subsets of the local languages such as Bambara and Bomu

[12] At the moment of writing, the use cases have been collected in Mali, and the FB platform is being built accordingly. However, no feedback has yet been received from the users.

have to be recorded and resourced using time-consuming techniques and efforts. The user interfaces have to be extensively tested and validated with end-users in the local situation, since these are culturally sensitive topics. For the resourcing of more local languages crowd-sourcing techniques may be applied. The issue of meta-information is another important topic. In the model presented for the FB radio platform in this paper, only a small amount of meta-data is collected. When the repositories of spoken content start to become larger, new innovative ways of describing spoken content have to be developed.

The annotation of the audio recordings will initially be done by radio employees. However, within the Web for Regreening in Africa initiative[13], we are currently developing Text To Speech (TTS) and Automated Speech Recognition (ASR) libraries for local dialects of French as well as local languages Bambara and Bomu. The TTS will allow us to a generate voice prompts in a dynamic way for the IVR system. The ASR system will be employed to automatically recognize parts of the audio content submitted by callers. This will further reduce the burden on the annotator. For larger languages such as English ASR is already used for automated call handling.

To contribute to a critical mass of content and applications that are necessary in this rural domain, a socio-technical network has to be put in place, that must be supported by a community of contributors: web developers, listeners that provide meta-information, local ICT-entrepreneurs, people who are willing to produce and consume data. According to Kazman and Hong-Mei Chen [9] organizing a community of developers around an open source service requires a consolidated kernel infrastructure, allowing peripheral services to be created by a de-centralized community of developers. Specific social and technical mechanisms are needed to ensure long-term participation and to encourage community engagement. In this case this is justified by the aim to open the Web of Data to people who are out of reach of computers and the internet.

References

1. Heine, B. (ed.): African Languages: An Introduction. Cambridge University Press, Cambridge (2000)
2. Akkermans, H., Grewal, A., Bon, A., Tuyp, W., Allen, M., Gyan, N.B.: W4RA-VOICES field report. Technical report, Web Alliance for Regreening Africa (2011)
3. de Boer, V., De Leenheer, P., Bon, A., Gyan, N.B., van Aart, C., Guret, C., Tuyp, W., Boyera, S., Allen, M., Akkermans, H.: RadioMarch: distributed voice- and web-interfaced market information systems under rural conditions. In: CAISE 2012, Gdansk, Poland (2012)
4. W3C: Voice Extensible Markup Language VoiceXML Version 2.0, W3C Recommendation 16 March (2004)
5. Agarwal, S.K., Jain, A., Kumar, A., Rajput, N.: The world wide telecom web browser. In: Proceedings of the First ACM Symposium on Computing for Development, ACM DEV 2010, New York, NY, USA (2010)

[13] http://www.w4ra.org/.

6. Horst, N.: EthnoCorder in Burundi: innovation, data collection and data use. In: Participatory Learning and Action IIED (2011). http://pubs.iied.org/pdfs/14606IIED.pdf
7. Guéret, C., Schlobach, S.: SemanticXO: connecting the XO with the worlds largest information network. In: ICeND 2011 (2011)
8. Berners-Lee, T.: Linked Data, the four rules (2006)
9. Kazman, R., Hong-Mei, C.: The metropolis model a new logic for development of crowd-sourced systems. Commun. ACM **52**(7), 76–84 (2009)

Social Media Matrix Matching Corporate Goals with External Social Media Activities

Harriet Kasper(✉), Iva Koleva, and Holger Kett

Fraunhofer Institute for Industrial Engineering IAO, 70569 Stuttgart, Germany
{harriet.kasper,holger.kett}@iao.fraunhofer.de,
kolevaiva@yahoo.de

Abstract. In this paper we introduce the Social Media Matrix, a practitioner-oriented instrument that supports companies to decide based on their corporate or communication goals which social media activities to execute. The matrix consists of three parts: 1. Social media goals and task areas have been identified and matched. 2. Five types of social media activities have been defined. 3. The matrix provides a structure to assess the suitability of each activity type on each social media platform for each goal. Whereas the first two parts can be generally used, the assessment must be conducted explicitly for an industry sector. A ready to use assessment for the German B2B sector has been exemplarily compiled from expert-interviews with practitioners and by reviewing concrete social media activities. The matrix is used as a basis for social media consultancy projects and evaluated thereby.

Keywords: Web 2.0 · Social media · Corporate social media · Marketing · Communication · Planning · Instrument · Tool · Decision tool · Practitioner-oriented · Goal-oriented · Business-to-business · B2B

1 Introduction

Facebook, founded in 2004, according to its own website[1] had 845 million monthly users at the end of December 2011. Together with platforms like Twitter,[2] LinkedIn[3] or YouTube[4] and other web 2.0 sites like blogs and forums, social media's impact on society and its relevance for business cannot be denied. The Social Media Governance Study 2011 [1], an empirical survey of communication managers and PR professionals in companies, governmental institutions and non-profit organizations in the German-speaking part of Europe: Germany, Austria and Switzerland, reveals that 71.4 percent of all organizations currently actively apply social media in their communications, which is a growth of 17 percent compared to the survey result of 2010 [2]. Only 7 percent of the respondents have neither applied nor planned to use social media. In the current project CLOUDwerker,[5] which is partially funded by the German Ministry

[1] www.facebook.com.

[2] www.twitter.com.

[3] www.linkedin.com.

[4] www.youtube.com.

[5] www.cloudwerker.de.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 233–244, 2015.
DOI: 10.1007/978-3-662-46641-4_17

of Economy and Technology, we experienced in several interviews that even small crafts enterprises are aware of the importance of social media and want to enter this field. Furthermore a survey of the University of Darmstadt [3] examined social media use by business-to-business (B2B) oriented companies, which seems to be different from consumer oriented businesses (B2C) and therefore is also a topic we are addressing with this paper. Structured approaches, methods and tools that support the process of choosing the `right` social media activities for a company have for a wide range of industries and businesses not been introduced yet. Our work aims to close this gap and suggests an instrument that can provide the necessary backing for corporate social media decisions. In this paper we concentrate on external social media activities, in contrast to such activities that are dealing with the internal use of social media also referred to as enterprise 2.0 [4].

Current literature concentrates on different aspects of social media and is examined in Sect. 3 of our paper, after introducing our methodology for developing the Social Media Matrix in Sect. 2. In Sect. 4 we will present the overall structure of the Social Media Matrix. Section 5 discusses the exemplary assessment of social media activities. We show how the Social Media Matrix can be used in Sect. 6 and conclude with describing future extensions in Sect. 7.

2 Methodology

The basic idea of matching corporate and communication goals with concrete social media activities on social media platforms arouse from talking to practitioners in companies across different industry sectors. Although it was our intention to create an instrument which is widely applicable, in the initial phase of our work we have chosen to narrow down our target group on business-to-business (B2B) oriented companies in Germany.

Besides considering relevant literature we have carried out six structured expert interviews [5] with practitioners who are in charge of social media activities in such B2B companies to identify their:

- overall approach, perceived potentials and challenges
- goals that should be met by social media activities
- best practices from experience and observation

The evaluation of the documented interviews confirmed the necessity of a structured approach, served as a review and completion of our goal list and formed the base for the assessment presented in Sect. 5. Furthermore by including the intended end user of the Social Media Matrix its relevance to practice could be ensured.

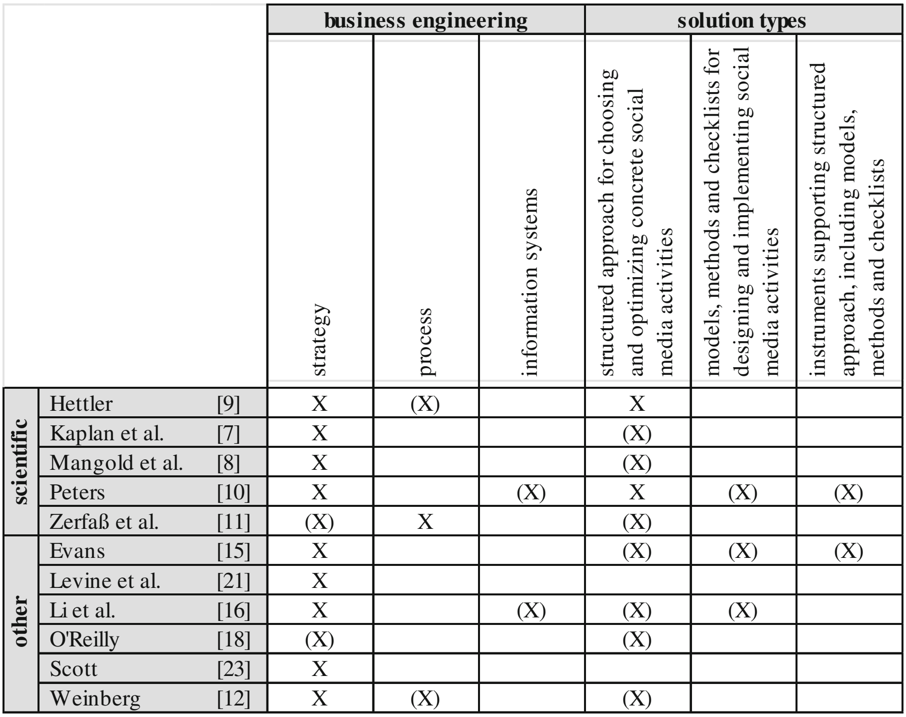
Our assumption that it is essential to assess social media activities and accumulate best practices and examples for an industry sector rather than only for goals has confirmed and today during the evaluation in consultancy projects our matrix can easily be adapted to other sectors e.g. business-to-consumer oriented industries.

3 Literature Review

Concerning social media, companies must answer the questions what to do (strategy), how to do this (organization: roles and processes) and which tools offer support to do so (information systems). This approach can be derived from Oesterle [6] who suggested this classification for business engineering tasks in general. Furthermore our practitioner-oriented view asks to distinguish between three solution types:

- structured approach for choosing and optimizing concrete social media activities
- models, methods and checklists for designing and implementing social media activities
- instruments supporting structured approach, including models, methods and checklists

Figure 1 gives an overview of social media publications and classifies them according to the above mentioned criteria.



			business engineering			solution types		
			strategy	process	information systems	structured approach for choosing and optimizing concrete social media activities	models, methods and checklists for designing and implementing social media activities	instruments supporting structured approach, including models, methods and checklists
scientific	Hettler	[9]	X	(X)		X		
	Kaplan et al.	[7]	X			(X)		
	Mangold et al.	[8]	X			(X)		
	Peters	[10]	X		(X)	X	(X)	(X)
	Zerfaß et al.	[11]	(X)	X		(X)		
other	Evans	[15]	X			(X)	(X)	(X)
	Levine et al.	[21]	X					
	Li et al.	[16]	X		(X)	(X)	(X)	
	O'Reilly	[18]	(X)			(X)		
	Scott	[23]	X					
	Weinberg	[12]	X	(X)		(X)		

Fig. 1. Classification of social media literature

Additional aspects of our literature review are summarized in the following paragraphs:

- **Scientific level:** A general distinction of literature about social media can be drawn between scientific studies e.g. Kaplan and Haenlein [7], Mangold and Faulds [8], Hettler [9], Peters [10], Zerfaß [11] and a tremendous number of key expert views on the use of social media in business e.g. Weinberg [12], Brogan [13, 14], Evans [15], Li and Bernoff [16] and Solis [17]. Due to being a fast moving new field of study, many of the experts and key authors in social media are current practitioners in this field, who have been widely-accepted as thought leaders e.g. key bloggers like Jeff Jarvis.[6]
- **Definition of social media:** There is a discussion about the concept of social media and how it differs from related concepts such as web 2.0 and user generated content [7, 12, 18–20]. The authors provide general advices for companies on how to engage in social media and how to communicate with their stakeholders on different platforms. In this regard Joel [19] contends that marketing of trust and transparency is important and focuses on the idea of 'building of a community based on trust'. Evans [15] shares this point of view and emphasizes the importance of involving customers in social media conversations with regard to social customer relationship management.
- **Impact of web-based activities:** Many early approaches consider not only technological development but also the changes in the ways organizations and end-users use the web. As one of the first books on web 2.0 'The Cluetrain Manifesto' [21] provides a set of 95 theses, which aim to examine the impact of the internet on both consumers and organizations. The manifesto contends that internet technologies enable people to have "human to human" conversations, which have the potential to transform traditional business practices. O'Reilly [18] describes the revolution of Web 2.0 as the era of participation and collective intelligence, and with his book "The wisdom of crowds" [22] James Surowiecki provided a standard work on the chances of this new development beyond using it for PR and marketing purposes.
- **Structured approach:** Few studies [9–11, 16] provide structured approaches, suggesting strategies and models for social media engagement and for implementation of social media activities. Forrester Research executives Li and Bernoff [16] develop a framework for engaging in the 'groundswell' based on following activities: listening, talking, energizing, and supporting. The framework attempts to help companies to understand and engage with their customers within the social media space. Although Li and Bernoff suggest a clear community engagement model and segment web 2.0 participators, it does not satisfactorily provide an understanding of the contribution of these activities to corporate objectives. Zerfaß [11] suggests the concept of 'Social Media Governance' as a regulatory framework for incorporating social media, based on a quantitative survey in Germany [1, 2].
- **Overall social media concept:** Although 'Social Media Governance' by Zerfaß is an overall concept concentrating on organizational issues, it focuses on the PR-perspective and omits that social media can be used to achieve goals in fields such

[6] www.buzzmachine.com/about-me/.

as product management or recruiting. Other studies analyze the use of social media from specific viewpoints, for example PR [23], social media marketing [12, 15], reputation management [10].

- **Social media marketing:** Some authors [12, 24] concentrate on social media marketing and understand social media as a chance to promote websites, products or services and 'to communicate with and tap into a much larger community that may not have been available via traditional advertising channels' [12]. Others are adamant that social media and marketing are distinctive disciplines and independent from each other. Brogan [14] for example suggests that "Marketing is NOT Social Media - Social Media is NOT Marketing". Focusing on the PR-perspective, Scott [23] believes that traditional public relations practices 'do not work anymore'. He contends that web 2.0 allows companies to take ownership of information and independently publish it instead of pushing it in the form of press releases to traditional media outlets. Although a lot of works analyze social media from a specific perspective, only a few studies focus on the integration of social media in the business practice. Mangold and Faulds [8] see social media as an element of the promotional mix within the marketing mix of a company and stress the need for integration of social media activities into communication strategies. They provide a set of examples for companies, which have successfully integrated social media. However, the study does not contain a well-structured approach, explaining how social media can be incorporated.
- **Social media management tools:** For monitoring and analysis of social media activities a new tool class has established which plays an important role in social media management. The provided opportunities for companies using these tools and their specific functions have been examined in several market studies [25–27]. The publications see a trend of these tools developing towards engagement platforms, which allow companies to directly react on found contributions in social media and a necessity to make the functions available for workgroups. In a whitepaper Owyang and Lovett suggest a social media measurement framework [28] and projects like Next Corporate Communications[7] research value and return on invest (ROI) of social media, but general, in depth and ready to use instruments and key performance indicators have not yet been revealed.
- **Model for social media planning:** Despite the high interest in the field of social media, well-structured approaches and models for realizing social media activities in the business practices are to our knowledge limited available. Although some scientific studies as well as some key experts suggest that planning social media activities has to be a part of a corporate communications strategy [8, 9, 11], still they do not satisfactorily provide solutions that help companies to plan and coordinate activities.

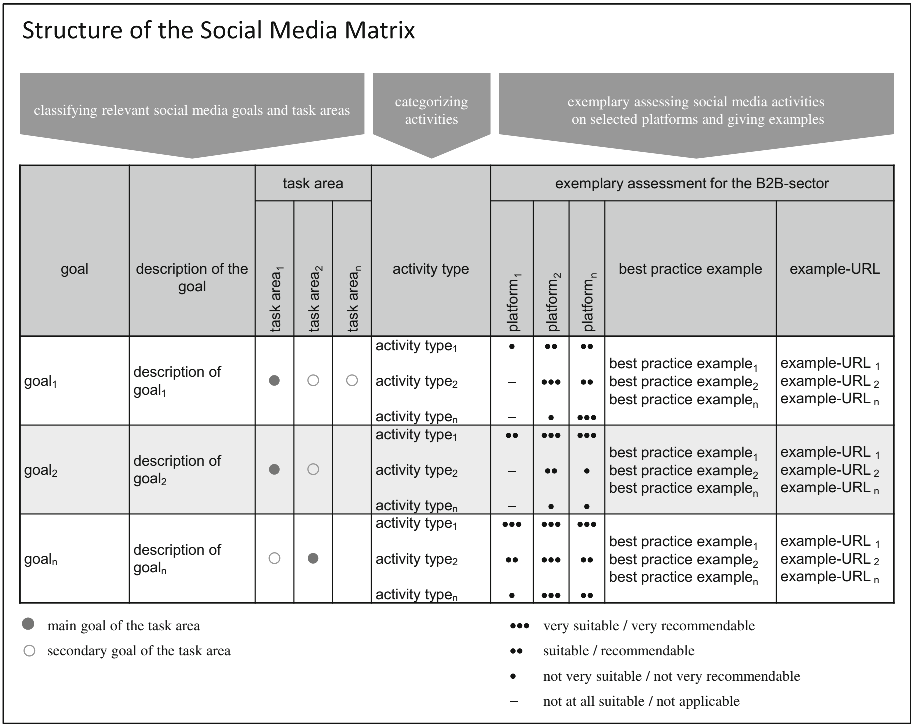
As shown in Fig. 1 current literature focuses on 'strategy' but when it comes to the more practice-oriented views on 'processes', 'information systems' and concrete 'models, methods and checklists' most generalizations do not work anymore. With the Social

[7] http://www.nextcc.de/.

Media Matrix we suggest an ‘instrument supporting structured approach, including models, methods and checklists’.

4 Social Media Matrix and Its Overall Structure

Our work supports the methodical selection of appropriate social media activities. The suggested instrument ‘Social Media Matrix’ aims at providing a match of suitable social media activities to relevant social media goals of companies. Thus, the matrix provides (1) a classification of social media goals and dedicated business tasks, (2) a categorization of social media activity types, and (3) the concrete assessment of each activity type on each platform for each goal, including a collection of examples. Figure 2 illustrates the overall structure of the Social Media Matrix.



Structure of the Social Media Matrix

classifying relevant social media goals and task areas
categorizing activities
exemplary assessing social media activities on selected platforms and giving examples

goal	description of the goal	task area			activity type	exemplary assessment for the B2B-sector				
		task area$_1$	task area$_2$	task area$_n$		platform$_1$	platform$_2$	platform$_n$	best practice example	example-URL
goal$_1$	description of goal$_1$	●	○	○	activity type$_1$	•	••	••	best practice example$_1$	example-URL $_1$
					activity type$_2$	–	•••	••	best practice example$_2$	example-URL $_2$
					activity type$_n$	–	•	•••	best practice example$_n$	example-URL $_n$
goal$_2$	description of goal$_2$	●	○		activity type$_1$	••	•••	•••	best practice example$_1$	example-URL $_1$
					activity type$_2$	–	••	•	best practice example$_2$	example-URL $_2$
					activity type$_n$	–	•	•	best practice example$_n$	example-URL $_n$
goal$_n$	description of goal$_n$	○	●		activity type$_1$	•••	•••	•••	best practice example$_1$	example-URL $_1$
					activity type$_2$	••	•••	••	best practice example$_2$	example-URL $_2$
					activity type$_n$	•	•••	••	best practice example$_n$	example-URL $_n$

● main goal of the task area
○ secondary goal of the task area

••• very suitable / very recommendable
•• suitable / recommendable
• not very suitable / not very recommendable
– not at all suitable / not applicable

Fig. 2. Structure of the social media matrix

(1) Classification of relevant social media goals by task areas

Social media can be used for achieving various goals of different departments. The matrix provides a classification of social media goals by task areas, in order to allow the user to focus on analysis, planning, execution and controlling of social media - away from the company’s functional perspective and toward the integrated communication management concept [8, 11]. To facilitate orientation and coordination with regard to the practical application of the instrument, we have identified different social media task areas and matched these with the goals. The following eleven task areas have been assigned to a total of 29 social media goals:

- media relations
- reputation management
- agenda setting
- issues management
- crises management
- branding & brand management
- customer relationship management & influencer relationship management
- trend, market & competition analysis
- social media marketing & social commerce
- market research & monitoring
- product & innovation management

A task area may pursue more than one goal. Furthermore the single goals may impact each other. For this reason, we have distinguished between main and secondary goals within a task area. For example, 'increasing brand awareness' is defined as a main goal within the task area 'branding & brand management'. Whereas 'building relationships to influencers in social media' is a secondary goal of 'branding and brand management' which will contribute to the achievement of the main goal.

(2) Categorizing activities

A major outcome of our work was to realize that before assessing the suitability of a platform for a goal the type of activity must be distinguished. We have therefore defined five types of social media activities, which cover all possibilities:

- **content:** Content refers to the publishing of information in the form of text, image, graphic, audio and video. This category includes only activities, which are based on one way communication, e.g. a company's info on the Facebook fan page or the channel on YouTube.
- **interaction/dialog:** This type comprises activities aimed at interacting and discussing with stakeholders. Such activities can be for example posts on the company's wall on Facebook or a group on LinkedIn.
- **listening and analyzing:** This activity type is about finding conversations about the company, its brands and products and analyzing these in order to turn them into valuable insights. This monitoring can be manual or automated and can furthermore include keyword search about competitors, current issues, important stakeholders etc.
- **application:** Expanding the possibilities of a social media platform by own applications to enable further interaction is summarized under this activity type. For example Facebook allows programming of own applications.
- **networking:** Taking active steps to find contacts and to build relationships is referred to as networking. This includes e.g. @-referencing other users on Twitter or inviting users to a group on LinkedIn.

(3) Assessing platforms and finding examples

Whereas goals, task areas and activity types are generally applicable this third part of the Social Media Matrix must be adapted both to the industry sector and the country/ regional context. It comprises a list of the most relevant social media platforms in one country and the assessment of the suitability of each activity type on each platform for each goal. Additionally best practices are collected and added as examples to the assessment.

The assessment can either rely on expert rating or derive from examples and best practices. For our exemplary assessment for B2B companies in Germany we used a combination of both approaches. A third possibility which we shortly discuss in the 'future research' section is crowdsourcing the assessment.

5 Assessing Suitability of Social Media Activities for German B2B Companies

To make the Social Media Matrix a ready to use instrument for planning corporate social media activities it must include specific assessment and examples to answer the question "How useful is a certain activity type on a certain platform to reach a certain goal?". In contrast to social media goals and to a certain extend also platforms which can both be generalized, the assessment in the matrix must be explicit, e.g. for an industry sector. In our work social media activities of German companies that predominantly follow business to business (B2B) transactions have been reviewed and evaluated according to goal, activity type and used platform to sample such a specific assessment. Together with the results of the expert interviews a directly applicable Social Media Matrix for the German B2B sector has been created which is at the same time through the examples a snap-shot of current social media use in B2B companies in Germany.

The social media platforms we considered in this use case were Facebook, Twitter, YouTube, Xing,[8] Blogs and 'community platforms or forums'. For transferring the results to other than German markets the platform Xing should be substituted by LinkedIn which is its international pendant. For most goals content, interaction/dialog and networking are the predominant activity types. Our assessment shows that social media activities can be used in German B2B companies best to achieve these goals:

- building and improving product and brand image
- identifying, contacting and bonding influencers
- improving ranking on search engines
- generating leads
- employer branding and recruiting

Social Media activities that aim towards vending products or providing service and support over social media platforms are less suitable, due to the complexity of B2B products. Nevertheless it could be a strategy to explicitly address such goals that seem

[8] www.xing.com.

less suitable and are therefore less represented to create unexpected social media activities that by their newness gain more attention which in social media means success.

6 Application Modes

Integrated communication distinguishes four management phases: analysis, planning, execution and controlling [29]. The Social Media Matrix is primarily designed to support planning activities, but can also be used for analyzing industry sectors beforehand or controlling own activities. A second dimension in applying the Social Media Matrix is the starting point of decision making. Besides goals - platforms and best practices can be used. The proposed structure allows all three starting points. Furthermore target groups/stakeholders are often used as a starting point for activity analysis, planning and controlling. This perspective has not (yet) been considered in the matrix, but since target groups are an implicit part of goals and best practices further detailing the matrix in this respect is possible.

Figure 3 illustrates the application modes of the Social Media Matrix.

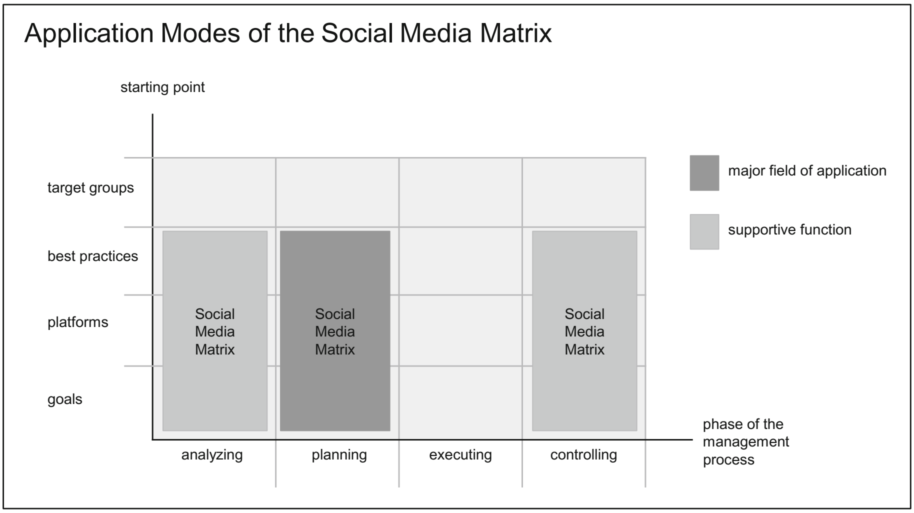


Fig. 3. Application modes of the social media matrix

We have initially implemented the social media matrix in Excel. The user can flexibly use the matrix for decision making by for example:

- prioritizing goals or filtering goals by task area
- using the assessment of the platforms to decide upon which platform to enter or to understand which optimization opportunities already used platforms offer
- using the assessment to identify which goals can best be achieved by social media activities
- using the given examples to better understand possibilities.

7 Conclusion and Future Research

We have introduced an instrument for companies to choose social media activities based on corporate goals and an assessment of the suitability of certain activities on selected platforms. The suitability assessment has been conducted exemplary for the German B2B-sector. Moreover our Social Media Matrix contains an example set of realized social media activities in the B2B-sector in Germany, which can be regarded as best practices and provide orientation for the user. Therefore this is a ready to use tool for German B2B-companies which is currently evaluated during consulting projects. It has been shown, that our practical integrated approach has not been presented in literature so far.

This work builds the foundation of a more sophisticated tool for planning and controlling corporate social media activities. To close this paper, three selected aspects of future work are presented in brief in the following paragraphs.

- **Crowdsourcing example collection and assessment of social media activities:** Especially in the field of social media the web community is a great source for information which can be used for optimizing the Social Media Matrix. A web interface could be used to collect examples for corporate social media activities. Each example should be assigned at least one goal, the industry-sector, the platform and the activity type it refers to. In an international context, country information should also be provided. Incentive to enter this data could be the possibility to compare the entered activity with other activities in the same industry sector. Assessing activities for different sectors could be either by experts, or everybody. Exceeding critical masses is essential here.
- **Detailing social media platform representation:** Since marketing and corporate communications is often driven by target group segmentation, this information must be modeled within the social media matrix. One approach is to add target group information to the description of platforms. Ideally the defined parameters can be easily used to describe any platform and new platforms thereby could derive their assessment from existing data.
- **Estimating costs:** To make the Social Media Matrix an effective tool in terms of Common Value Management the assessment of social media activities must be expanded by adding cost factors for execution of an activity. Cost factors could be distinguished into initial cost and operating cost, they contain time elements (e.g. singular vs. regular) and are interconnected with the available resources. It is a matter of research to what extend cost estimate and ROI of social media activities can be generalized.

Acknowledgements. The work published in this article is partially based on the diploma thesis of Iva Koleva filed in January 2012 with the University of Hohenheim in Stuttgart, Germany. The thesis has been initiated and supervised by Harriet Kasper, Fraunhofer IAO and examined by Prof. Dr. Frank Brettschneider-Vetter (1st examiner, University of Hohenheim) and Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath (2nd examiner, University of Stuttgart).

References

1. Fink, S., Zerfaß, A., Linke, A.: Social Media Governance 2011 – Expertise Levels, Structures and Strategies of Companies, Governmental Institutions and Non-Profit Organizations communicating on the Social Web. University of Leizpig/Fink & Fuchs Public Relations AG, Leipzig/Wiesbaden (2011). www.socialmediagovernance.eu
2. Fink, S., Zerfaß, A.: Social Media Governance 2010 – How companies, the public sector and NGOs handle the challenges of transparent communication on the Internet. University of Leipzig/Fink & Fuchs Public Relations AG, Leipzig/Wiesbaden (2010). www.socialmediagovernance.eu
3. Pleil, T.: Mehr Wert schaffen: Social Media in der B2B-Kommunikation. Books on Demand, Darmstadt (2010)
4. Spath, D. (ed.), Guenther, J.: Trendstudie Wissensmanagement 2.0 – Erfolgsfaktoren für das Wissensmanagement mit Social Software. Fraunhofer Verlag, Stuttgart (2010)
5. Bogner, A., Littig, B., Menz, W.: Das Experteninterview – Theorie Methode, Anwendung, 2nd edn. VS Verlag für Sozialwissenschaften, Wiesbaden (2005)
6. Oesterle, H.: Business Engineering – Prozess- und Systementwicklung. Band 1: Entwurfstechniken, 2nd edn. Springer, Berlin (1995)
7. Kaplan, A.M., Haenlein, M.: Users of the world, unite! the challenges and opportunities of social media. Bus. Horiz. **53**(1), 59–68 (2010)
8. Mangold, W.G., Faulds, D.J.: Social media - the new hybrid element of the promotion mix. Bus. Horiz. **52**(4), 357–365 (2009)
9. Hettler, U.: Social Media Marketing. Marketing mit Blogs, sozialen Netzwerken und weiteren Anwendungen des Web 2.0. Oldenbourg, München (2010)
10. Peters, P.: Reputationsmanagement im Social Web. Risiken und Chancen von Social Media für Unternehmen, Reputation und Kommunikation. Social Media Verlag, Köln (2011)
11. Zerfaß, A., Fink, S., Linke, A.: Social media governance: regulatory frameworks as drivers of success in online communications. In: IPRRC International Public Relations Research Conference, Miami (2011)
12. Weinberg, T.: Social media marketing. Strategien für Twitter. Facebook & Co., O'Reilly, Beijing (2010)
13. Brogan, C., Smith, J.: Trust economies: investigations into the new ROI on the web (2008). http://changethis.com/44.04.TrustEconomy
14. Brogan, C.: Marketing is NOT Social Media-Social Media is NOT marketing (2007). http://www.chrisbrogan.com/marketing-is-not-social-media-social-media-is-not-marketing/
15. Evans, D.: Social Media Marketing. An Hour a Day. Wiley, Indianapolis, Indiana (2008)
16. Li, C., Bernoff, J.: Groundswell: Winning in a World Transformed by Social Technologies. Harvard Business School Press, Boston (2008)
17. Solis, B.: Engage! The Complete Guide for Brands and Business to Build, Cultivate and Measure Success in the New Web. Wiley, Hoboken (2010)
18. O'Reilly, T.: What Is Web 2.0. O'Reilly Media, Sebastopol (2005). http://oreilly.com/web2/archive/what-is-web-20.html
19. Joel, M.: Trust economies: the new marketing ROI. Six Pixels of Separation (2007). http://www.twistimage.com/blog/archives/trust-economies—the-new-marketing-roi/
20. Joel, M.: Six Pixels of Separation. Everyone is Connected - Connect Your Business to Everyone. Business Plus, New York (2010)
21. Levine, R., Locke, C., Searls, D., Weinberger, D.: The cluetrain manifesto. Perseus Publishing, Cambridge (2000). http://www.cluetrain.com/

22. Surowiecki, J.: The Wisdom of Crowds - Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. Doubleday, Anchor (2004)
23. Scott, D.M.: The New Rules of Marketing & PR - How to Use News Releases, Blogs, Podcasting, Viral Marketing and Online Media to Reach Buyers Directly. Wiley, Hoboken (2008)
24. Barefoot, D., Szabo, J.: Friends with Benefits. A Social Media Marketing Handbook. No Starch Press, San Francisco (2010)
25. Kasper, H., Dausinger, M., Kett, H., Renner, T.: Marktstudie Social Media Monitoring Tools – IT-Lösungen zur Beobachtung und Analyse unternehmensstrategisch relevanter Informationen im Internet. Fraunhofer Verlag, Stuttgart (2010)
26. Plum, A.: Ansätze, Methoden und Technologien des Web-Monitorings – ein systematischer Vergleich. In: Brauckmann, P. (ed.) Web-Monitoring – Gewinnung und Analyse von Daten über das Kommunikationsverhalten im Internet, pp. 21–46. UVK, Konstanz (2010)
27. Gilliatt, N.: Social Media Analysis Platforms for Workgroups. Social Target LCC (2010)
28. Owyang, J., Lovett, B.: Social Marketing Analytics - A New Framework for Measuring Results in Social Media. Altimeter Group/Web Analytics Demystified (2010). http://www.slideshare.net/jeremiah_owyang/altimeter-report-social-marketing-analytics
29. Zerfaß, A.: Unternehmenskommunikation und Kommunikationsmanagement - Grundlagen, Wertschöpfung, Integration. In: Pilwinger, M., Zerfaß, A. (eds.) Handbuch Unternehmenskommunikation, pp. 21–70. Gabler, Wiesbaden (2007)

Knowledge Modeling of On-line Value Management

Dieter Fensel, Birgit Leiter, Stefan Thaler, Andreas Thalhammer, Anna Fensel, and Ioan Toma(✉)

STI Innsbruck, University of Innsbruck, Innsbruck, Austria
ioan.toma@sti2.at

Abstract. We discuss the challenge of scalable dissemination and communication approaches in a world where the number of channels is growing exponentially. The web, Web 2.0, and semantic channels have provided a multitude of interaction possibilities providing significant potential for yield, brand, and general reputation management. Our goal is to enable smaller organizations to fully exploit this potential. To achieve this, we have developed a new methodology based on distinguishing and explicitly interweaving content and communication as a central means for achieving content reusability, and thereby scalability over various heterogeneous channels.

1 Introduction and Motivation

In current times, it is (in principle) possible to instantly communicate with a large portion of the entire human population. Nevertheless, new means also generate new challenges. Take the world of the TV consumer as an example. Twenty-five years ago, there were around three channels. Therefore, selecting your program was a rather trivial task which required no more than a few seconds. Whilst hundreds of channels have been added, thousands of channels have been connected via the Internet, where extremely large libraries of videos (which go beyond the metaphor of a 'channel'), currently define the content. The consumer could now spend a lifetime in search of a program he or she wishes to watch. Obviously, consumers require new skills and more efficient access means to scale and filter the exponentially increased offer.

Precisely the same is needed for our overall approach to on-line (or Internet-based) communication. Assume the task of a small hotelier. How can it be ensured that the hotel is found by potential customers, i.e., how can she find them? The hotelier should have a website with high visibility on various search engines and must be present in a large number of on-line booking channels. We should find the hotel on the town's website, and the hotel should have a Facebook page, perhaps with a booking engine included. Bookings made through mobile platforms are increasingly popular, and the hotelier would want to be found there too. Why not add a video about the hotel on YouTube, a chat channel for instant communication, fast email and fax response capabilities, the old-fashioned telephone, and occasional tweets and emails that are clearly distinguishable from spam? Preferably the communication should be multidirectional, i.e., the hotelier should realize when one of his posts gets commented on (up to a full-fledged impact analysis), or even more importantly, the hotelier should

E. Simperl et al. (Ed.): ESWC 2012 Satellite Events, LNCS 7540, pp. 245–258, 2015.
DOI: 10.1007/978-3-662-46641-4_18

know when someone talks about the hotel, and how much the costumer liked it. As much as this is needed, this obviously does not scale and [15] calls it "*the growth of the multichannel monster*". Organizations of all sizes, commercial and not-for-profit, regularly face the challenge of communicating with their stakeholders using a multiplicity of channels, e.g. websites, videos, PR activities, events, email, forums, online presentations, social media, mobile applications, and recently structured data. The social media revolution has made this job much more complicated, because:

- the *number of channels* has grown exponentially,
- the communication has changed from a mostly unilateral "push" mode (one speaker, many listeners) to an increasingly fully *bilateral communication*, where individual stakeholders (e.g. customers) expect one-to-one communication with the organization, and the expected speed of reaction is shrunk to almost real-time, and
- the *contents of communication becomes more and more granular* and increasingly dependent on the identity of the receiver and the context of the communication.

Organizations need an integrated solution that provides management and execution of communication goals in a mostly automated fashion, with costs equivalent to mass-media communication, along with the granularity of individual experts, and at the pace of real-time social media. We are aiming to mechanize important aspects of these tasks, allowing scalable, cost-sensitive, and effective communication for small-or-medium sized business units and comparable organizations for which information dissemination is essential but resources are significantly limited. Additionally, it may also help intermediaries such as marketing agencies to extend their business scope by increasing the cost-effective ratio.

The remainder of this paper is structured as follows: Sect. 2 analyses the major goals that may underlie communicative interaction of an organization with a larger audience. Section 3 sketches the major technical elements that we developed to implement a common value management framework. Section 4 discusses some of the related work. Finally, conclusions are provided in Sect. 5.

2 The Aspects of Value Management

Scalable, multi-channel communication is a difficult challenge. In order to better understand it, we want to clarify the various underlying goals that it should achieve. Agents often connect (directly or indirectly) economic interests with their communication activities. In the following, we discuss different economic contexts for the communication approaches of organizations.

2.1 Yield Management

Yield or revenue management "is an economic discipline appropriate to many service industries in which market segment pricing is combined with statistical analysis to expand the market for the service and increase the revenue per unit of available capacity" [8].[1] Short-term increase of income is a valid target for a business entity;

[1] http://en.wikipedia.org/wiki/Yield_management, and http://en.wikipedia.org/wiki/Revenue_management.

however, it is quite tricky to realize in a multichannel world. For many channels, visibility is achieved through low prices. However, channels also often require price constraints on the price offers of other channels. Some channels generate costs without guaranteeing actual income. A hotel needs recommendations for what needs to be done and the support to do it, e.g. possible actions would be to reduce their price by 10 % or to include more amenities and supplements.

Many solutions to yield management are based on complex statistical methods and complex domain assumptions on how variation of the price can influence the number of bookings of a service. However, a multi-directional multi-channel approach must also rely on *Swarm intelligence*.[2] Observing in real time the reaction of customers and competitors will be a key to achieving successful on-line marketing. Adapting an offer dynamically in response to the behavior of the (on-line visible) environment will become critical for economic success.

2.2 Brand Management

Yield management tries to maximize the immediate revenue of an organization. However, communication is also very important in relation to the long-term value of a company. Actually, the reputation of a company can be viewed as one of its most important assets. Proper management, such as managing the value of brands, may be essential for its long-term economic success. This may conflict with revenue management. In many cases, it may be useful for short-term income management to reduce the price of the offering, which on the other hand can diminish and undermine the long-term income that is generated through a general price profile indicating quality and exclusivity.

2.3 Reputation Management

The economic impact of proper reputation management is evident when we talk about the reputation of economic entities. However, non-profit organizations also have a need for general *reputation management* and *public campaigns*.[3] "Reputation is the opinion (more technically, a social evaluation) of a group of entities toward a person, a group of people, or an organization on a certain criterion. It is an important factor in many fields, such as education, business, online communities or social status."[4] Here, it is not the direct and intermediate economic income that matters. It is rather about maintaining or increasing the appreciation an organization, topic, or certain approach gains in the public eye. However, even a campaign on a public issue has an immediate economic dimension to it: trying to use the available budget for it in the most effective way. Therefore, providing means to increase the effectiveness and efficiency of public campaigns is of high value.

[2] http://en.wikipedia.org/wiki/Swarm_intelligence.

[3] E.g. http://www.readwriteweb.com/archives/how_to_manage_your_online_reputation.php.

[4] http://en.wikipedia.org/wiki/Reputation.

2.4 Value Management

All of the issues above could be viewed as facets of Value Management, where value is defined as *the regard that something is held to deserve, i.e., its importance*. Online, multi-channel and bi-directional Value Management is about disseminating, communicating, and interacting with large, on-line communities to increase the value of a certain entity or issue. The value managed could cover issues such as importance, economic short-term income, or long-term value. Reference [10] identifies the following activities as part of an on-line based value management: Reputation management; Competitive Intelligence, i.e., Competitor Observation; Market Analysis; Influencer Detection; Trend Analysis; Market Analysis; Crisis Management; Issue Management; Campaign Monitoring; Product and Innovation Management; Customer Relationship Management; Risk Management; and Event Detection. Obviously, these activities overlap and share many common elements. It would be interesting to reduce these activities to the set of atomic tasks from which they are composed.

3 A Methodological Approach Towards Common Value Management

We start this section by introducing the underlying idea and major structure of our approach. We then discuss our information model, channel model, the weaver, and we sketch some applications of our approach.

3.1 Separating Content and Channel to Enable Various Dimensions of Reuse in Transactional Communication

The core idea of our approach is to introduce a layer on top of the various Internet based communication channels that is domain specific and ***not*** channel specific.[5] So one has:

- *information models*, that define the type of information items in a domain;
- a *channel model* (or communication model), that describes the various channels, the interaction pattern, and their target groups;
- *mappings* of information items to channels through weavers; and finally,
- a library of *implemented wrappers* for actual channel instances.

What is essential is to *distinguish* the communication or channel model from the conceptual descriptions of the information.[6] Our approach requires the creation of a communication model (i.e., an increasingly complete model of channels), and knowledge models for each vertical (such as research projects, research institutes, associations, hotels, restaurants, tourist events, medical doctors, etc.), and finally linking the

[5] See also as an excellent presentation on this idea: http://www.slideshare.net/reduxd/beyond-the-polar-bear.

[6] In analogy to style sheets that separate the contents from its presentation.

knowledge model with the communication model through a weaver that weaves concepts with channels. Data and information can be expressed at the conceptual level, which the domain expert understands. Mapping of the different communication means (where he is not at all an expert) is done automatically after the first implementation. The difficult dissemination through channels is done automatically through proper channels that are attached to these concepts.

Currently, all commercially available solutions are only channel centric and do not provide any built-in support for what needs to be disseminated or where to disseminate what piece. In our approach, a knowledge-model is built and explicitly linked with the channel model. This must be done once for a hotel, and can then be reused for millions of them. That is, we aim for the major elements of reusability:

1. The same information element can be *reused* for various channels through its channel independent formulation using the information model.
2. The information model is developed as domain ontology for a certain vertical area such as tourist accommodations, gastronomy, medical doctors etc. Therefore, it can be *reused* for various agents active in the same vertical domain.

These elements of reusability deliver the major contribution to the scalability of our approach.

3.2 Information Model

An information model is an ontology that describes the information items that are used in typical communication acts in a certain domain. Many methodologies for building such ontologies have been developed. Building ontologies can be a time-consuming and expensive process. Fortunately, we have a strong modeling bias that helps us to significantly guide and therefore reduce such an effort. We can focus on the major and typical information items that are used in the on-line dissemination and communication processes. Therefore, the size of these ontologies in our case studies (see Sect. 3.4.), were moderate (around 100 concepts and properties), and many of these concepts and properties could be reused between different use cases. As a result, there was a reduced effort in building informal domain models (less than one person month). After defining an informal model, we formalized this ontology (see [2] for more details) in a simple sublanguage of OWL-2, since we foresee little need for reasoning about it. We model structured information items as concepts and non-structured ones as properties, i.e., we assume simple non-structured values for properties.

In an intermediate phase of our journey, we also tried to directly use some LOD vocabularies to model these ontologies.[7] The conclusion from this experience is shared by [17]: “In contrast to the heterogeneity of the Web, it is beneficial in the application context to have all data describing one class of entities being represented using the same vocabulary … it is thus advisable to translate data to a single target vocabulary”. We draw an important conclusion from this: *For us, LOD vocabularies are not means to describe our content models, i.e. they were not really useful for deriving domain models.* That is,

[7] We had used a mélange of Dublin Core, FOAF, schema.org, and GoodRelations.

we model our information items in a Domain Ontology that is understandable by the domain experts. Interaction with them is essential to our approach and therefore understandability of our means towards these domain and communication experts. *For us, LOD vocabularies are means to disseminate and share information and not means to model information*. Ontologies are always on the brink of being a very specific and well-defined domain model derived from certain first principles, being very useful for a specific purpose in contrast to broadly used and consensually developed models used for sharing information between different viewpoints. Consequently, we live in a world of multiple ontologies. "We no longer talk about a single ontology, but rather about a network of ontologies. Links must be defined between these ontologies and this network must allow overlapping ontologies with conflicting – and even contradictory – conceptualizations [6]." We achieve this by weaving our models with LOD vocabularies when we see a gain in broadening our range of communication through them.

3.3 Channel Model

"In telecommunications and computer networking, a communication channel, or channel, refers either to a physical transmission medium such as a wire or to a logical connection over a multiplexed medium such as a radio channel."[8] In on-line communication, we take a broad definition of a channel. A channel is a means of exchanging information in the on-line space. There is a close relationship between URIs and channels as each URI can be used as a channel to spread or access information. However, not each channel directly refers to an URI. For example, Facebook provides around forty different methods of spreading information not distinguished by a URI. Additionally, individual information items spread through Facebook are not distinguished by URIs. In general, a channel can be interpreted as a "place" where one can find or leave information, whether it is unanimously referred by a URI or addressed through a service. However, even this is not broad enough. As described previously, a channel can also be the URI of a vocabulary (or the formalisms such as RDFa or microformats) that are used to publish the information. Through use of this URI, only humans or software agents that "speak" this dialect are able to access this information. Here, the communication channel cannot be interpreted as a place, but rather as a way to express or refer to the information. In the following, we want to distinguish channels by the communication mode they support.

Communication is based on the broadcasting of information. Therefore, we define the first category of our channel classification system as channels used for *broadcasting*. Here we make a distinction between the publication of mostly *static information* and *dynamic contents* that express the timeliness of an information item. One way of spreading information is to invite other people to use it. Therefore, *sharing* is another category we have identified. It reflects the insight that others are not passive consumers of our information but active prosumers that should be helped and supported in their information processing activities. Sharing is the first form of

[8] http://en.wikipedia.org/wiki/Communication_channel.

cooperation. Explicit *collaboration* through a shared information space is the next cooperation category we have identified. Collaboration between individuals leads to groups of people actively organizing their communication and cooperation. Social networking sites that support *groups of people* in their information needs are instances of this next category we have identified. Obviously, the boundaries between these categories are fluid and many channel providers try intentionally to establish services covering several of them. Still, it is often possible to identify a major category for them, which provides means for adding *machine-processable semantics* to information.

Broadcasting static information. Websites are an established means of providing (mostly) static information. Information that reflects the structure of the contents is provided through websites and they offer a smooth way for users to access this content. An important addition beyond the dissemination through an owned website is an entry on other sites such as Wikipedia, the world's leading encyclopedia.

Broadcasting dynamic information. With Web 2.0 technologies, dedicated means for publishing streams and interacting with information prosumers have been added. A first step in this direction is the inclusion of a News section in a website using blogging tools such as Wordpress. Such news can be further spread through a news ticker such as *RSS feeds* and *Twitter*. *Email* and *Email lists* are also well established means for news dissemination. Especially the latter are a proven means of broadcasting information and facilitating group discussions.

Sharing. There are a large number of Web 2.0 websites that support the sharing of information items such as: bookmarks, images, slides, and videos, etc.

Collaboration. A *wiki* is primarily a means for project internal collaboration. However, it also becomes a dissemination channel if external visitors have *read* access[9]. They may then follow the intensive internal interaction that can help to gain a better and more detailed understanding of externally published results and achievements.

Group communication. *Facebook* as a social networking site provides an additional community aspect, i.e., it forms a community that multi-directionally shares news, photos, opinions, and other important aspects. Notice that Facebook is actually not only one, but several channels. It offers more than 40 possibilities through which to disseminate information. These can also be tightly integrated into Web 1.0 pages, such as that of the New York Times.[10] *Google+* may have the potential to become a major competitor of Facebook. Therefore, it should also be included in a social networking site strategy. *LinkedIn* **and** *Xing* are focused on professional use and perfectly fit the purpose of research organizations.

[9] Write access cannot be provided due to spamming.

[10] http://www.nytimes.com/.

Semantic-based Dissemination. An important approach to broaden the scope of a dissemination activity is to add machine-processable semantics to the information. With this approach, search and aggregation engines can provide much better service in finding and retrieving this information. Semantic annotations injected in websites are used by search engines such as Google to provide a structured presentation of the contents of websites, such as that shown in Figure 3, which can be analyzed by the format and vocabulary used. "This data may be embedded within enhanced search engine results, exposed to users through browser extensions, aggregated across websites or used by scripts running within those HTML pages [21]." Already more than 60 million web domains are using machine-processable meta data.[11]

There are various *formats* of adding machine-processable semantics to data. First, there are three competing means of including semantics directly in HTML/XML files: (1) RDFa adds a set of attribute-level extensions to XHTML enabling the embedding of RDF triples; (2) Microformats directly use meta tags of XHTML to embed semantic information in web documents; (3) Microdata use HTML5 elements to include semantic descriptions into web documents aiming to replace RDFa and Microformats.[12] For the moment, we have three competing proposals that should be supported in parallel until one of them can take a dominant role on the web.[13]

Instead of including semantic annotations in XHTML documents, i.e., injecting machine-readable contents into content that is meant for direct human consumption, they can also be provided for direct machine consumption. A straight-forward way is to publish an RDF file containing the machine readable data. Instead of directly publishing an RDF file you can also provide a SPARQL endpoint allowing the querying RDF information. Instead of retrieving the entire RDF file, directed queries can be supported with this approach

In addition to predefined formats and technical means, we need to reuse predefined *LOD vocabularies* to describe our data to enable semantic-based retrieval of information.[14] Currently, we use Dublin Core, FOAF, GoodRelations, and schema.org.

Notice that we use each term of a vocabulary as a potential dissemination channel. For example, for the PlanetData fact sheet we publish pieces of the information using the following vocabulary terms: schema:url, foaf:topic, dc:creator, dc:date, dc:subject, and dc:title.

3.4 Weaver

The central element of our approach is the separation of content and communication channels. This allows reuse of the same content for various dissemination means. Through this reuse, we want to achieve scalability of multi-channel communication. The explicit modeling of content independent from specific channels also adds a second element of reuse: Similar agents (i.e., organizations active in the same domain) can reuse significant parts of such an information model.

[11] Compare http://webdatacommons.org/.

[12] See [21] for more details.

[13] Compare http://webdatacommons.org/.

[14] More than a hundred of them are listed at http://labs.mondeca.com/dataset/lov/index.html.

Separating content from channels also requires the explicit alignment of both. This is achieved through a weaver. Formally, a weaver is a set of tuples of nine elements:

1. An *information item:* As discussed in Sect. 2, it defines an information category that should be disseminated through various channels.
2. An *editor:* The editor defines the agent that is responsible for providing the content of an information item.
3. An *editor interaction protocol:* This defines the interaction protocol governing how an editor collects the content.

   Elements 1 to 3 are about the content. They define the actual categories, the agent responsible for them, and the process of interacting with this agent. Elements 4 to 9 are about the dissemination of these items.
4. An *information type:* We make a distinction between three types of content: an instance of a concept, a set of instances of a concept (i.e., an extensional definition of the concept), and a concept description (i.e., an intensional definition of a concept).
5. A *processing rule:* These rules govern how the content is processed to fit a channel. Often only a subset of the overall information item fits a certain channel.[15]
6. A *channel:* The media that is used to disseminate the information.
7. *Scheduling information:* Information on how often and in which intervals the dissemination will be performed which includes temporal constrains over multi-channel disseminations.
8. An *executor:* It determines which agent or process is performing the update of a channel. Such an agent can be a human or a software solution.
9. An *executor interaction protocol:* It governs the interaction protocol defining how an executer receives its content.

First, the information types distinguish whether one wants to disseminate a general description of the information item, an instance of the information item, or a set of all instances. For example, we want to find an overall description of scientific presentations (what is their general theme) and a set of all presentations at a defined place on the web. The former may be placed on the project website and the later may be placed on SlideShare as a means to share presentations. Finally, a single instance may be broadcast as news through the various news broadcasting channels. Now, take a single presentation as an example. The title, author, abstract, and event it was given may form the news. The title, author, and a short notion of the event may define a tweet, and the slides themselves may go to SlideShare. That is, the information item must be processed to fit the various dissemination channels. A channel is a URI or an API of an existing web service. Scheduling information defines temporal constraints for dissemination in a single channel and for dependencies between multi-channel dissemination. For example, a new presentation will be announced once. However, an event may be announced as soon at it is defined and a reminder may be sent out when certain deadlines (for submitting papers or for early registrations) are near. News may first be

[15] In case of LOD this can be an R2R mapping rule [4].

published on the website. Then, an excerpt of the news together with its URI will be published as a tweet.

A weaver is basically a large collection of tables specifying what is disseminated by whom to where. Interaction protocols, rules, and constraints further guide this process. Such a manual is of extreme importance to manage the on-line communication process.

3.5 Use Cases

We developed and applied our approach in three major case studies: the European Semantic Web Conference Series (ESWC)[16], the PlanetData project[17] and the Semantic Technology Institute (STI) International research association[18].

- The mission of the *Extended Semantic Web Conference (ESWC) series* is to bring together researchers and practitioners dealing with different aspects of semantics on the web. Founded in 2004, the ESWC builds on the success of the former European Semantic Web Conference series, but seeks to extend its focus by engaging with other communities within and outside ICT, in which semantics can play an important role.
- *PlanetData* is a semantic technology project funded by the European Commission. It aims to create a durable community made up of academic and industrial partners working on large-scale data management.
- *STI International* is a global network engaging in research, education, innovation and commercialization activities on semantic technologies working to facilitate their use and applicability within industries and society as a whole. STI International is organized as a collaborative association of interested scientific, commercial and governmental parties that share a common vision.

Around 80 % of the information items of ESWC, PlanetData, and STI International are interchangeable due to some simple renaming (e.g., core and associate partner versus partner and member). This is excellent news and a hint for scalability especially given the fact that we talk about a research *project* and a research *association*. This could imply that an even higher degree of reuse could be achieved when applying our information model to tens of thousands of European research projects (and hundreds of thousands of research projects or millions of projects) on the one hand, and millions of associations on the other. This is actually the second major assumption of our approach.[19] Reuse of the information model in a certain vertical area. The costs to build an information models are quickly paid back when applicable to several entities in a domain. These models empower simple non-IT users to communicate at the level of their domain knowledge rather than at the symbol level of various channels and these models can be reused between different players in the same vertical.

[16] http://eswc-conferences.org/.

[17] http://www.planet-data.eu/.

[18] http://www.sti2.org/.

[19] The first one is that is that it will pay back to model the information independent from the multitude of dissemination channels, ensuring reuse over them.

Based on our approach ESWC, PlanetData, and STI International are now managing their on-line appearance. In total, we have identified around *five hundred* different semantic and non-semantic channels in these case studies that are used to disseminate elements of the information model. Obviously, such a bandwidth requires a structured and mechanized approach. Based on our approach, around 300 concepts and properties, 500 channels, i.e., more than 100,000 potential content-to-channel mappings are run efficiently by a very small dissemination team.

4 Related Work

Many aspects of our work clearly relate to different fields that have been explored before. Generally, we see two specifically related areas: *Ontology-based content management systems (CMSs) for websites* and *Semantic matchmaking of senders and receivers of content*. Both areas will be briefly described and compared.

The field of *semantics-based or enhanced CMSs* has already been quite thoroughly explored. One of the earlier approaches to ontology-based website management is the OntoWebber system described in [9]. The proposed three-way approach of "explicit modeling of different aspects of websites", "the use of ontologies as foundation for Web portal design", and "semi-structured data technology for data integration and website modeling" presents an early but comprehensive approach to semantifying CMSs. OntoWebber introduces an integration layer which adapts to different data sources. This is related to our weaver concept introduced in Sect. 4, but, in contrast, the weaver adapts to different channels rather than to different information sources. A year later, in [18], Sheth et al. introduce the SCORE system, which defines four key features: semantic organization and use of metadata, semantic normalization, semantic search, and semantic association. Although written in the early days of the Semantic Web, the paper covers topics such as metadata extraction from unstructured text and automatic classification that may also become relevant to our approach. Reference [7] introduce "The Rhizomer Semantic Content Management System" which integrates services with metadata browsing, editing, and uploading, continuing their earlier work on the Knowledge Web portal. Reference [5] proposes a Linked Data extension for Drupal that enables content annotation with RDFa and provides a SPARQL endpoint. The British national broadcaster BBC started to integrate semantic technologies (i.e. Linked Data) in 2009 in order to integrate various data and content sources distributed throughout the enterprise [12]. As a result, as reported in [3], BBC's World Cup 2010 site[20] is based on semantic repositories that enable the publishing of metadata about content rather than publishing the content itself. While the data input is fixed, different schemas for the output are defined. However, as only one channel for output is considered, the mapping performed is quite straight-forward. In contrast, our system accounts for different information needs of various and heterogeneous channels and therefore enables the distribution of content through different portals. Finally, the European project Interactive Knowledge Stack (IKS)[21] focuses on porting semantic technologies to CMS software solutions.

[20] http://www.bbc.co.uk/worldcup.

[21] http://iks-project.eu/.

In a nutshell, all these approaches aim either to help the user publish semantic data or to use semantic methods to support the content management process for maintaining websites. We are taking these approaches and generalizing them to support the overall management of content dissemination in a multi-channel and bi-directional communication setting. Further, we augment the technical approach with a methodology and the approach of using vertical domain models, which are shared and reused in a vertical area instead of being used for a single application only.

Semi-automatic matchmaking is a well-studied field in Artificial Intelligence and related areas. Obviously we can only select a small sample of approaches in this area, which focus on matchmaking in regard to content. Reference [11] presents a selective information dissemination system that is based on semantic relations. In their paper, the terms in user profiles and terms in documents are matched through semantic relations that are defined using a thesaurus. Similarly, the approach taken by [14] introduces selective dissemination of information for digital libraries based on matching information items to user profiles. Obviously, user profiles correspond to our channels, however, we instead manually model their relationship with contents. The system introduced in [13] uses RDF, OWL, and RSS to introduce an efficient publish/subscribe mechanism that includes an event matching algorithm based on graph matching. Our approach, in contrast, matches information items to channels rather than events to users. Also, instead of graph matching, we use predefined weavers for channel selection. While [14] uses fuzzy linguistic modeling and NLP techniques for semiautomatic thesaurus generation and performs a matching based on statistical analysis, we use semantics to manually define the connections between information items and the channels.

Since we aim for high precision and professionalism in on-line communication, we see little use for statistical based semantic methods (natural language understanding, information extraction, etc.). We want to allow the user to abstract from the channel level to the content level, but we see the need for human involvement in defining the content-channel mapping and at the content level. However, as we expand towards a full-fledged value management approach that monitors the entire web space for important statements, such methods will be needed. Fortunately, a large number of such web analytical toolkits already exist, [10, 20][22] lists a large number of them that cover parts of these tasks. However, there is an important need for methods and integrated tools that cover the multi-channel bi-directional aspects of value management and provide highly scalable and effective solutions.

5 Conclusions

The following core features characterize our approach:

- We use ontologies to model content in order to have a representation layer independent from the communication channel. We want to achieve reuse of content over channels allowing small organizations to deal with an increasing number of

[22] See also [19] and http://www.somemo.at/?p=474.

communication channels and exploit their potential. The alignment of content and channel is achieved through a weaver that aligns ontological items with channels.

- These ontologies are not case-specific, but model a certain vertical domain such as research projects, associations, accommodations, restaurants, bars, touristic events and services, etc. Therefore, these ontologies and their channel alignments can be reused on a larger scale, providing a quick return of the investment necessary to build and maintain them.
- Our approach is bi-directional, i.e., in the same way that we disseminate through concepts we use these concept to aggregate feedback and impact found in various channels.
- We support in an integrated fashion, the dissemination via traditional web channels, Web 2.0, and semantic based channels, using various formats and vocabularies.

Based on our approach, ESWC, PlanetData, and STI International are now managing their on-line appearance. Currently, we are performing additional case studies. First, we use our approach in the dissemination of other research projects and associations. Second, we are entering more commercial areas such as eTourim, where millions of hotels are desperately waiting for a scalable dissemination strategy, given the fact that soon, around 50 % of all room bookings will be done on-line.

Acknowledgements. We would like to thank Anja Bunnefeld, Johannes Breitfuss, Carmen Brenner, Alice Carpenter, Michael Fried, Mark Greaves, Marko Grobelnik, Martin Hepp, Lyndon Nixon, Ina O' Murchu, Alexander Oberhauser, Simeona Pellkvist, Elena Simperl, Corneliu Valentin Stanciu, and Alexander Wahler for contributions to early drafts of this paper.

References

1. Amersdorffer, D., Bauhuber, F., Egger, R., Oellrich, J. (eds.): Social Web im Tourismus. Springer, Heidelberg (2010)
2. Bauereiß, T., Leiter, B., Fensel, D.: Effective and Efficient On-lin Communication, STI TECHNICAL REPORT 2011-12-01 (2011). http://www.stiinnsbruck.at/TR/Online Communication
3. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R.: OWLIM: A family of scalable semantic repositories, Technical report (2010)
4. Bizer, C., Schultz, A.: The R2R framework: Publishing and discovering mappings on the web. In: Proceedings of the 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010. http://www4.wiwiss.fu-berlin.de/bizer/r2r/
5. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Decker, S.: Produce and consume linked data with Drupal. Springer Constraints J. **1380**, 763–778 (2009)
6. Fensel, D.: Ontologies: Dynamic Networks of Formally Represented Meaning (2001). http://sw-portal.deri.at/papers/publications/network.pdf
7. Garcia, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M.: The rhizomer semantic content management system. In: Lytras, M.D., Damiani, E., Tennyson, R.D (eds.) WSKS 2008. LNCS (LNAI), vol. 5288, pp. 385–394. Springer, Heidelberg (2008)
8. THE BASICS OF REVENUE MANAGEMENT: Integrated Decisions and Systems, Inc. (2005). http://www.adhp.org/pdf/1-theBasicsofRM.pdf

9. Jin, Y., Decker, S., Wiederhold, G.: Ontowebber: Modeldriven ontology-based website management. In: Proceedings of the Semantic Web Working Symposium (SWWS), Stanford University (2001)
10. Kasper, H., Dausinger, M., Kett, H., Renner, T.: Marktstudie Social Media Monitoring Tools. Frauenhofer Verlag (2010)
11. Katzagiannaki, I.-E., Plexousakis, D.: Information dissemination based on semantic relations. In: CAiSE Short Paper Proceedings 2003 (2003)
12. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media meets semantic web – How the BBC uses DBpedia and linked data to make connections. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E (eds.) ESWC 2009. LNCS, vol. 5554, pp. 723–737. Springer, Heidelberg (2009)
13. Ma, J., Xu, G., Wang, J.L., Huang, T.: A semantic publish/subscribe system for selective dissemination of the RSS documents. In: Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC 2006), pp. 432–439 (2006)
14. Morales-del-Castillo, J.M., Pedraza-Jimenez, R., Ruiz, A.A., Peis, E., Herrera-Viedma, E.: A semantic model of selective dissemination of information for digital libraries. Inform. Technol. Libr. **28**(1), 21–31 (2009)
15. Mulpuru, S., Harteveldt, H.H., Roberge, D.: Five Retail eCommerce Trends to Watch in 2011. Forrester Research Report, 31 January 2011
16. Newell, A.: The knowledge level. Artif. Intell. **18**(1), 87–127 (1982)
17. Schultz, A., Matteini, A., Isele, R., Bizar, C., Becker, C.: LDIF – Linked data integration framework. In: Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany, October 2011. http://www4.wiwiss.fu-berlin.de/bizer/ldif/
18. Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Managing semantic content for the web. IEEE Internet Comput. **6**, 80–87 (2002)
19. Solis, B.: The Rise of Digital Influence, Altimeter Group (2012)
20. Stavrakantonakis, I., Gagiu, A.-E., Kasper, H., Toma, I., Thalhammer, A.: An approach for evaluation of social media monitoring tools (submitted)
21. J. Tennison: HTML Data Guide, W3C Editor's Draft 02 March 2012

Online Open Neuroimaging Mass Meta-Analysis with a Wiki

Finn Årup Nielsen[1](✉), Matthew J. Kempton[2], and Steven C.R. Williams[2]

[1] DTU Compute, Technical University of Denmark, Lyngby, Denmark
faan@dtu.dk
http://www.compute.dtu.dk/∼faan/

[2] Department of Neuroimaging, Institute of Psychiatry,
King's College London, London, UK

Abstract. We describe a system for meta-analysis where a wiki stores numerical data in a simple comma-separated values format and a web service performs the numerical statistical computation. We initially apply the system on multiple meta-analyses of structural neuroimaging data results. The described system allows for mass meta-analysis, e.g., meta-analysis across multiple brain regions and multiple mental disorders providing an overview of important relationships and their uncertainties in a collaborative environment.

1 Introduction

The scientific process aggregates a large number of scientific results into a common scientific consensus. *Meta-analysis* performs the aggregation by statistical analysis of numerical values presented across scientific papers. Collaborative systems such as wikis may easily aggregate text and values from multiple sources. However, so far they have had limited ability to apply numerical analysis as required, e.g., by meta-analysis.

Researchers have discussed the advantages and disadvantage of the tools for conducting systematic reviews from "paper and pencil", over spreadsheets to RevMan and web-based specialized applications [1]: Setup cost, versatility, ability to manage data, etc. In 2009 they concluded that "no single data-extraction method is best for all systematic reviews in all circumstances". For example, RevMan and Archie of the Cochrane Library provide an elaborate system for keeping track and analyzing textual and numerical data in meta-analyses, but the system could not import information from electronic databases [1]. Our original meta-analyses [2,3] relied on the Microsoft Excel spreadsheets later distributed on public web sites. Compared to an ordinary spreadsheet a wiki solution provides data entry provenance and collaborative data entry with immediate update. Shareable folders on cloud-based storage systems would help collaboration on spreadsheets. Online services, such as the spreadsheet of Google Docs,

Finn Årup Nielsen—Thanks to the Lundbeck Foundation for the funding of the *Center for Integrated Molecular Brain Imaging* (CIMBI).

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 259–271, 2015.
DOI: 10.1007/978-3-662-46641-4_19

may lack meta-analytic plotting facility. Web-based specialized applications for systematic reviews may have a high setup cost [1].

We have previously explored a simple online meta-analysis system—a "fielded wiki"—in connection with personality genetics [4]. As implemented specifically for this scientific area the web service lacks generality for other types of meta-analytic data. Furthermore the system relied on PubMed or Brede Wiki to represent bibliographic information.

Following Ward Cunningham's quote "What's the simplest thing that could possibly work?" we present a simple system that allows for mass meta-analysis of numerical data presented as comma-separated values (CSV) in a standard MediaWiki-based wiki, — the Brede Wiki: http://neuro.compute.dtu.dk/wiki/. We present the data in an Open science data fashion where other researchers can easily reuse the data.

2 Data and Data Representation

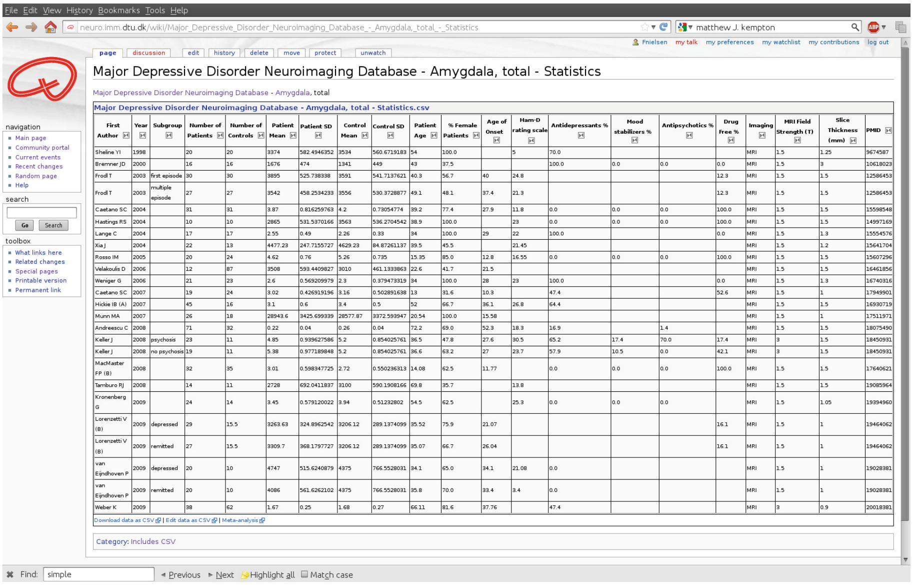
We use the MediaWiki-based Brede Wiki to represent the data [5]. For our neuroimaging data each data record usually consists of three values (number of subjects, the mean and standard deviation across subjects of the volume of brain structures). The individual study typically compares two such data records, e.g., from a patient and a control group. We also record labels for the data record, e.g., the biographic information, as well as extra subject information about the two groups, such as age, gender and clinical characteristics, so we get seven or more as the total number of data items for each study. Each meta-analysis will usually determine what extra relevant information we may include, and it may differ between studies, e.g., a *Y-BOCS* value has typically only relevance for obsessive-compulsive disorder (OCD) patients. The functional neuroimaging area has *CogPO* and *Cognitive Atlas* ontologies [6,7] enabling researchers to describe the topic of an experiment, but these efforts do not directly apply to our data. One CSV line carries the information for each study.

Separate wiki pages store—rather than uploaded files—the CSV data, so the MediaWiki template functionality can transclude the CSV data on other wiki pages. By convention pages with CSV information have the ".csv" extension as part of the title so external scripts can recognize them as special pages and the wiki pages have no wiki markup. Another approach instead of extensions could have used a dedicated MediaWiki namespace for the CSV pages, such that these pages would have the prefix `CSV:` or similar.

A few MediaWiki extensions can format CSV information: *SimpleTable*[1] and *TableData*. Figure 1 shows the transclusion of CSV data with a modified version of the SimpleTable extension. A MediaWiki template handles the transclusion and also establish web links to download and edit the CSV data.

The Brede Wiki uses the standard template system for recording structured bibliographic data about the publication, and a script that reads through the wiki dumps can represent the data in relational databases [5].

[1] http://www.mediawiki.org/wiki/Extension:SimpleTable.



Major Depressive Disorder Neuroimaging Database - Amygdala, total - Statistics

Major Depressive Disorder Neuroimaging Database - Amygdala, total

Major Depressive Disorder Neuroimaging Database - Amygdala, total - Statistics.csv

First Author	Year	Subgroup	Number of Patients	Number of Controls	Patient Mean	Patient SD	Control Mean	Control SD	Patient Age	% Female Patients	Age of Onset	Ham-D rating scale	Antidepressants %	Mood stabilizers %	Antipsychotics %	Drug Free %	Imaging	MRI Field Strength (T)	Slice Thickness (mm)	PMID
Sheline YI	1998		20	20	3374	582.4946352	3534	560.6719183	54	100.0		5	70.0				MRI	1.5	1.25	9674587
Bremner JD	2000		16	16	1676	474	1341	449	43	37.5			100.0	0.0	0.0	0.0	MRI	1.5	3	10618023
Frodl T	2003	first episode	30	30	3895	525.738338	3591	541.7137621	40.3	56.7	40	24.8				12.3	MRI	1.5	1.5	12586453
Frodl T	2003	multiple episode	27	27	3542	458.2534233	3556	530.3728877	49.1	48.1	37.4	21.3				12.3	MRI	1.5	1.5	12586453
Caetano SC	2004		31	31	3.87	0.816259763	4.2	0.73054774	39.2	77.4	27.9	11.8	0.0	0.0	0.0	100.0	MRI	1.5	1.5	15598548
Hastings RS	2004		10	10	2865	531.5370166	3563	536.2704542	38.9	100.0		23	0.0	0.0	0.0	100.0	MRI	1.5	1.5	14997169
Lange C	2004		17	17	2.55	0.49	2.26	0.33	34	100.0	29	22	100.0			0.0	MRI	1.5	1.3	15554576
Xia J	2004		22	13	4477.23	247.7155727	4629.23	84.87261137	39.5	45.5		21.45					MRI	1.5	1.2	15641704
Rosso IM	2005		20	24	4.62	0.76	5.26	0.735	15.35	85.0	12.8	16.55	0.0	0.0	0.0	100.0	MRI	1.5	1.5	15607296
Velakoulis D	2006		12	87	3508	593.4409827	3010	461.1333863	22.6	41.7	21.5						MRI	1.5	1.5	16461856
Weniger G	2006		21	23	2.6	0.569209979	2.3	0.379473319	34	100.0	28	23	100.0			0.0	MRI	1.5	1.3	16740316
Caetano SC	2007		19	24	3.02	0.426919196	3.16	0.502891638	13	31.6	10.3		47.4			52.6	MRI	1.5	1	17949901
Hickie IB (A)	2007		45	16	3.1	0.6	3.4	0.5	52	66.7	36.1	26.8	64.4				MRI	1.5	1.5	16930719
Munn MA	2007		26	18	28943.6	3425.699339	28577.87	3372.593947	20.54	100.0	15.58						MRI	1.5	1	17511971
Andreescu C	2008		71	32	0.22	0.04	0.26	0.04	72.2	69.0	52.3	18.3	16.9		1.4		MRI	1.5	1.5	18075490
Keller J	2008	psychosis	23	11	4.85	0.939627586	5.2	0.854025761	36.5	47.8	27.6	30.5	65.2	17.4	70.0	17.4	MRI	3	1.5	18450931
Keller J	2008	no psychosis	19	11	5.38	0.977189848	5.2	0.854025761	36.6	63.2	27	23.7	57.9	10.5	0.0	42.1	MRI	3	1.5	18450931
MacMaster FP (B)	2008		32	35	3.01	0.598347725	2.72	0.550236313	14.08	62.5	11.77		0.0	0.0	0.0	100.0	MRI	1.5	1.5	17640621
Tamburo RJ	2008		14	11	2728	692.0411837	3100	590.1908166	69.8	35.7		13.8					MRI	1.5	1.5	19085964
Kronenberg G	2009		24	14	3.45	0.579120022	3.94	0.51232802	54.5	62.5		25.3	0.0	0.0	0.0		MRI	1.5	1.05	19394960
Lorenzetti V (B)	2009	depressed	29	15.5	3263.63	324.8962542	3206.12	289.1374099	35.52	75.9	21.07					16.1	MRI	1.5	1	19464062
Lorenzetti V (B)	2009	remitted	27	15.5	3309.7	368.1797727	3206.12	289.1374099	35.07	66.7	26.04					16.1	MRI	1.5	1	19464062
van Eijndhoven P	2009	depressed	20	10	4747	515.6240879	4375	766.5528031	34.1	65.0	34.1	21.08	0.0				MRI	1.5	1	19028381
van Eijndhoven P	2009	remitted	20	10	4086	561.6262102	4375	766.5528031	35.8	70.0	33.4	3.4	0.0				MRI	1.5	1	19028381
Weber K	2009		38	62	1.67	0.25	1.68	0.27	66.11	81.6	37.76		47.4				MRI	3	0.9	20018381

Download data as CSV | Edit data as CSV | Meta-analysis

Category: Includes CSV

Fig. 1. Screenshot from the wiki showing CSV data transcluded on a page and formatted into a table by a MediaWiki extension.

To annotate the CSV information we also use templates. Figure 2 shows the markup of three different CSV pages with information about major depressive, bipolar and obsessive-compulsive disorders, where the title field indicates the title of the CSV page in the wiki and the topic fields link to items in the Brede Wiki topic ontology. Presently, no controlled vocabulary beyond the template fields describes the columns in the CSV. To generate an appropriate content-type (text/csv) a bridging web script functions as a proxy, so a download of the CSV page can spawn a client-side spreadsheet program.

The bulk of the data currently presented in the wiki comes from the large mass meta-analysis of volumetric studies on major depressive disorder reporting over 50 separate meta-analyses for individual brain regions [2]. Further data comes from mass meta-analyses across multiple brain regions on bipolar disorder [3] and first-episode schizophrenia [8], a meta-analysis on longitudinal development in schizophrenia [9] as well as data from individual original studies on obsessive-compulsive disorder.

Apart from neuroimaging studies the Brede Wiki also records data from meta-analyses from a few other studies outside neuroimaging [10,11], allowing us to test the generality of the framework. We distribute the data under *Open Data Commons Open Database License* (ODbL).

3 Web Script and Meta-Analysis

The web script for meta-analysis reads the CSV information by downloading it from the wiki, identifies the required columns for meta-analysis, performs the

```
{{Metaanalysis csv begin}}
{{Metaanalysis csv
 | title = Major Depressive Disorder Neuroimaging Database - Amygdala, total - Statistics
 | topic1 = Amygdala
 | topic2 = Major depressive disorder
 | topic3 = MaND
}}
{{Metaanalysis csv
 | title = Bipolar Disorder Neuroimaging Database - Amygdala
 | topic1 = Amygdala
 | topic2 = Bipolar disorder
 | topic3 = BiND
 | cn_ne = Number of Bipolars
 | cn_me = Bipolar Mean
 | cn_sde = Controls SD
}}
{{Metaanalysis csv
 | title = Obsessive-compulsive disorder Neuroimaging Database - Amygdala
 | topic1 = Amygdala
 | topic2 = Obsessive-compulsive disorder
 | topic3 = ObND
}}
{{Metaanalysis csv end}}
```

Fig. 2. Template to annotate the CSV data and define the links to the meta-analysis.

statistical computations and makes meta-analytic plots— the so-called forest and funnel plots—in the SVG format, see Fig. 3. From either the title information or a PubMed identifier the script generates back-links from the generated page to pages on the wiki. The script may also export the computed results as JSON or CSV. Furthermore, it may generate a small *R* script that sets up the data in variables and use the `meta` *R* library for meta-analysis. Finally the script itself can return its source code enabling other researchers to readily inspect the code and more easily reproduce the results [12].

The web script attempts to guess the separator used on the CSV page and also tries to match the elements of the column header, e.g., the strings "control n", "controls number", "number of controls", etc. match for number of control subjects. With no matches the user needs to explicitly specify the relevant columns via URL parameters, which in turn a wiki template can setup. Figure 2 displays three templates where the middle template explicitly specify which columns the web script should interpret as "number of experimentals", "mean of experimentals" and "standard deviation of experiments".

Standard meta-analysis computes an *effect size* from each result in a paper and computes a combined meta-analytic effect size and its confidence interval. Although the methodological development continues, there exist established statistical analysis approaches for ordinary meta-analysis [10]. Our system implements computations on the standardized mean difference for continuous variables and on the natural logarithm of the odds ratio for categorical variables with fixed and random effects methods using an inverse-weighted variance model. As an extra option we provide meta-analysis on the natural logarithm of the variance ratio [13], for comparison of the standard deviations between two groups of subjects. Meta-analysts seldom use this type of summary statistics compared to the standard effect size, but in a few areas such statistics generate interest, e.g., in the controversy over whether males show larger variance in intelligence than females [11].

With continuous data where we have $\bar{x}_e$, s_e and n_e as the mean, standard deviation and number of subjects for the experimental group (e.g., patients) and $\bar{x}_c$, s_c and n_c as the corresponding numbers for the control group (e.g., healthy controls) we compute an effect size d for each study included in the meta-analysis by

$$\begin{aligned} s_{\text{pooled}} &= \sqrt{\frac{(n_e - 1)s_e^2 + (n_c - 1)s_c^2}{n_e + n_c - 2}} \\ g &= \frac{\bar{x}_e - \bar{x}_c}{s_{\text{pooled}}} \\ d_{\text{smd}} &= \left(1 - \frac{3}{4(n_e + n_c) - 9}\right) g. \end{aligned}$$

With binary data we compute the logarithm of the odds ratio as

$$d_{\text{or}} = \ln \left[\frac{c(n_{ee})/c(n_e - n_{ee})}{c(n_{ce})/c(n_c - n_{ce})} \right], \qquad \text{where } c(x) = \begin{cases} 0.5 & \text{if } x = 0 \\ x & \text{else} \end{cases}$$

where n_{ee} and n_{ce} indicate the number of experimental events and control events (e.g., success of treatment in each of the groups), while the $c(\cdot)$ function handles the case where no events appear for either the experimental or control group, resulting in a division by zero. For the variance ratio we compute

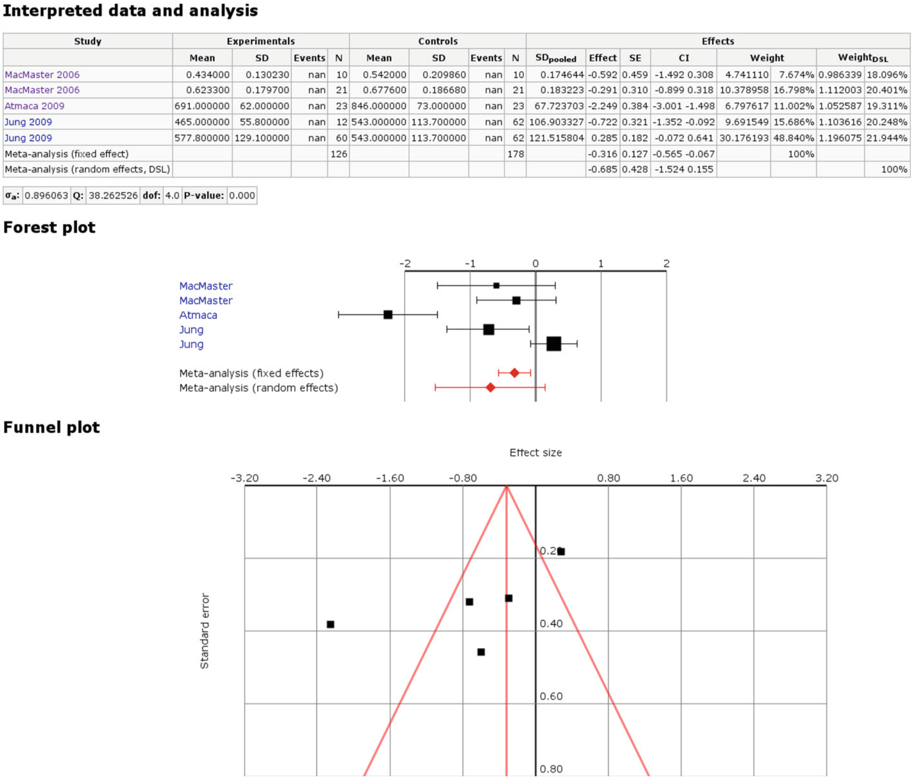
$$d_{\text{vr}} = \ln \left(\frac{s_e^2}{s_c^2} \right).$$

From the effect sizes and their variances we compute Z-scores, two-sided P-values and confidence intervals. Our random effects model gets estimated following the DerSimonian-Laird approach and we furthermore compute Cochran's homogeneity statistics that gives us a value for how much between-study variance affects each meta-analysis. The web script may return all these numerical results in the JSON format.

Certain types of neuroimaging studies report brain coordinates that may require specialized flexible multivariate meta-analysis [14,15], for a recent review see [16]. Our system does not yet implement these techniques.

4 Results

We have added 127 pages transcluded with CSV data, — most of which contain data suitable for meta-analysis. For mass meta-analysis we can presently consider 59 meta-analytic datasets. For a single meta-analyses the reading, computation and download finish within seconds. With multiple calls to the web script and JSON output another script can plot multiple meta-analytic results together



Interpreted data and analysis

Study	Experimentals				Controls				Effects							
	Mean	SD	Events	N	Mean	SD	Events	N	SD_{pooled}	Effect	SE	CI	Weight		$Weight_{DSL}$	
MacMaster 2006	0.434000	0.130230	nan	10	0.542000	0.209860	nan	10	0.174644	-0.592	0.459	-1.492 0.308	4.741110	7.674%	0.986339	18.096%
MacMaster 2006	0.623300	0.179700	nan	21	0.677600	0.186680	nan	21	0.183223	-0.291	0.310	-0.899 0.318	10.378958	16.798%	1.112003	20.401%
Atmaca 2009	691.000000	62.000000	nan	23	846.000000	73.000000	nan	23	67.723703	-2.249	0.384	-3.001 -1.498	6.797617	11.002%	1.052587	19.311%
Jung 2009	465.000000	55.800000	nan	12	543.000000	113.700000	nan	62	106.903327	-0.722	0.321	-1.352 -0.092	9.691549	15.686%	1.103616	20.248%
Jung 2009	577.800000	129.100000	nan	60	543.000000	113.700000	nan	62	121.515804	0.285	0.182	-0.072 0.641	30.176193	48.840%	1.196075	21.944%
Meta-analysis (fixed effect)				126				178		-0.316	0.127	-0.565 -0.067		100%		
Meta-analysis (random effects, DSL)										-0.685	0.428	-1.524 0.155				100%

σ_a:	0.896063	Q:	38.262526	dof:	4.0	P-value:	0.000

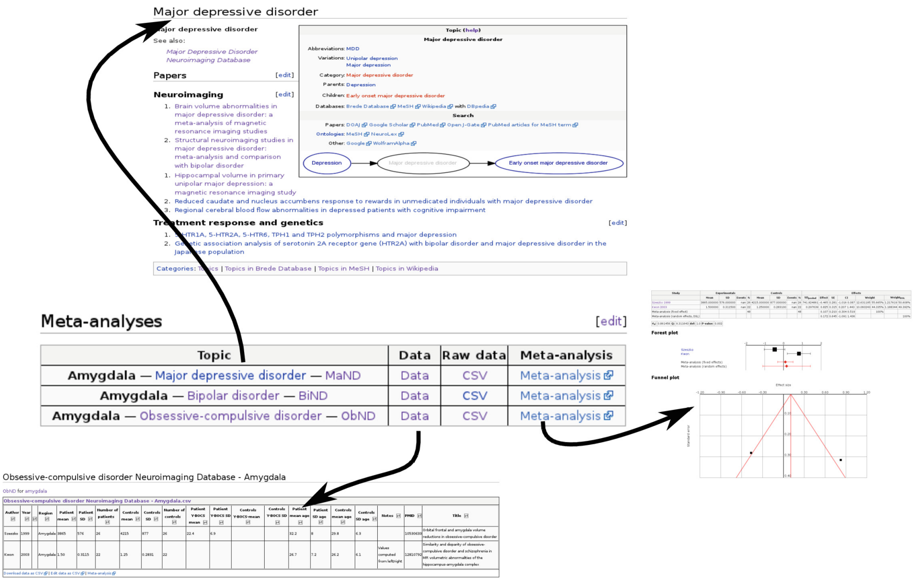
Forest plot

Funnel plot

Fig. 3. Screenshot of web script showing the meta-analytic results with forest and funnel plots on a dataset on pituitary gland volume differences between obsessive-compulsive disorder patients and healthy controls (Color figure online).

as in Fig. 5. Generating such a plot takes several minutes: First we use the MediaWiki API of Brede Wiki to get a list of pages which use the meta-analysis template (as shown in Fig. 2). We then download all these pages in the form of raw wiki markup and extract the templates with regular expressions. With this extracted information we call the meta-analytic web script multiple time, letting it return the results in the machine readable JSON format, to get the meta-analytic results that we can plot and tabularize. For generating the page shown in Fig. 3 with results from an individual meta-analysis we need only the CSV data and the web script, while the script that generated Fig. 5 with mass meta-analysis used information defined in templates, CSV data and the web script with no further adaption of MediaWiki.

The specific example shown in Fig. 3 displays results from 5 studies reported in 3 different papers so far entered in the Brede Wiki on the differences between obsessive-compulsive disorder patients and healthy controls in pituitary gland volume [17–19]. In this case the forest plot exposes a large variation between the



Major depressive disorder

ajor depressive disorder

See also:

Major Depressive Disorder Neuroimaging Database

Papers [edit]

Neuroimaging [edit]

1. Brain volume abnormalities in major depressive disorder: a meta-analysis of magnetic resonance imaging studies
2. Structural neuroimaging studies in major depressive disorder: meta-analysis and comparison with bipolar disorder
1. Hippocampal volume in primary unipolar major depression: a magnetic resonance imaging study
2. Reduced caudate and nucleus accumbens response to rewards in unmedicated individuals with major depressive disorder
3. Regional cerebral blood flow abnormalities in depressed patients with cognitive impairment

Tre tment response and genetics [edit]

1. HTR1A, 5-HTR2A, 5-HTR6, TPH1 and TPH2 polymorphisms and major depression
2. Ge etic association analysis of serotonin 2A receptor gene (HTR2A) with bipolar disorder and major depressive disorder in the Japa se population

Categories: T ics | Topics in Brede Database | Topics in MeSH | Topics in Wikipedia

Topic (help)

Major depressive disorder

Abbreviations:	MDD
Variations:	Unipolar depression Major depression
Category:	Major depressive disorder
Parents:	Depression
Children:	Early onset major depressive disorder
Databases:	Brede Database MeSH Wikipedia with DBpedia

Search

Papers:	DOAJ Google Scholar PubMed Open J-Gate PubMed articles for MeSH term
Ontologies:	MeSH NeuroLex
Other:	Google WolframAlpha

Depression
Major depressive disorder
Early onset major depressive disorder

Meta-analyses [edit]

Topic	Data	Raw data	Meta-analysis
Amygdala — Major depressive disorder — MaND	Data	CSV	Meta-analysis
Amygdala — Bipolar disorder — BiND	Data	CSV	Meta-analysis
Amygdala — Obsessive-compulsive disorder — ObND	Data	CSV	Meta-analysis

Study	Experimentals				Controls				Effects							
	Mean	SD	Events	N	Mean	SD	Events	N	SD_{pooled}	Effect	SE	CI		Weight		$Weight_{DSL}$	
Szeszko 1999	3865.000000	576.000000	nan	26	4215.000000	877.000000	nan	26	741.924861	-0.465	0.281	-1.016	0.087	12.631195	55.665%	1.217619	50.608%
Kwon 2003	1.500000	0.311500	nan	22	1.250000	0.283100	nan	22	0.297639	0.825	0.315	0.207	1.443	10.060243	44.335%	1.188344	49.392%
Meta-analysis (fixed effect)				48				48		0.107	0.210	-0.304	0.519		100%		
Meta-analysis (random effects, DSL)										0.172	0.645	-1.091	1.436				100%

τ_{d}^{2}:	0.861456	Q:	9.311643	dof:	1.0	P-value:	0.002

Forest plot

Funnel plot

Obsessive-compulsive disorder Neuroimaging Database - Amygdala

ObND for amygdala

Obsessive-compulsive disorder Neuroimaging Database - Amygdala.csv

Author	Year		Region	Patient mean	Patient SD	Number of patients	Controls mean	Controls SD	Number of controls	Patient Y-BOCS mean	Patient Y-BOCS SD	Controls Y-BOCS-mean	Controls Y-BOCS SD	Patient mean age	Patient SD age	Controls mean age	Controls SD age	Notes	PMID	Title
Szeszko	1999		Amygdala	3865	576	26	4215	877	26	22.4	6.9			32.2	8	29.8	6.3		10530633	Orbital frontal and amygdala volume reductions in obsessive-compulsive disorder
Kwon	2003		Amygdala	1.50	0.3115	22	1.25	0.2831	22					26.7	7.2	26.2	6.1	Values computed from left/right	12810792	Similarity and disparity of obsessive-compulsive disorder and schizophrenia in MR volumetric abnormalities of the hippocampus-amygdala complex

Download data as CSV | Edit data as CSV | Meta-analysis

Fig. 4. Connections between the different components of the meta-analysis wiki system: In the middle a table on the wiki generated from template information with links to topic ontology pages on the wiki (top), the transcluded data (bottom) and the web script with meta-analysis of the data (right).

studies. The colored components in the table and the forest plot link back to pages on the wiki associated with the three papers.

Figure 4 shows the links between the pages in the wiki and to the web script. The middle table gets generated from template definitions and data similar to the data displayed in Fig. 2. It provides links to topic ontology pages (with the page for "Major depressive disorder" shown), the data and result page generated by the meta-analysis web script.

Table 1 lists a part of the mass meta-analytic results sorted according to P-value of the effect size. Disregarding the non-neuroimaging results we see that results from meta-analyses on hippocampal and lateral ventricles volumes in first-episode schizophrenia and hippocampal volumes in major depressive disorder reach the highest significance. The original published meta-analyses [2,8] also mentioned these highly significant results prominently.

Whereas an offline script generated Fig. 5 an online web script has generated Fig. 6 with a mass meta-analysis of obsessive-compulsive disorder neuroimaging studies. This script can read a single wiki page with templates that describes meta-analytic CSV data and direct the meta-analytic web script to analyze all that data and return the results in a machine readable format. The mass meta-analytic web script then formats the results in a sortable table and plot it in the L'Abbé-like effect-uncertainty plot, here presently using the Bubble chart from

Table 1. Top 20 meta-analyses as of June 2012 sorted according to P-value of a total of 57 meta-analyses. The "ES" column displays the effect sizes.

#	P-value	ES	Meta-analysis
1	0.00000	1.90	Parkinson's disease and nicotinamide N-methyltransferase
2	0.00000	0.23	Sex differences in means and variability on the progressive matrices in university students: a meta-analysis - Table 1
3	0.00000	0.60	Brain volume in first-episode schizophrenia: systematic review and meta-analysis of magnetic resonance imaging studies - Table 4, part 1
4	0.00000	−0.47	Major Depressive Disorder Neuroimaging Database - Hippocampus, total - Statistics
5	0.00000	−0.44	Major Depressive Disorder Neuroimaging Database - Hippocampus, left - Statistics
6	0.00000	−0.52	Brain volume in first-episode schizophrenia: systematic review and meta-analysis of magnetic resonance imaging studies - Table 3, part 1
7	0.00000	0.39	Bipolar Disorder Neuroimaging Database - Lateral ventricles
8	0.00000	1.49	A refined method for the meta-analysis of controlled clinical trials with binary outcome - Table 1
9	0.00000	−0.53	Brain volume in first-episode schizophrenia: systematic review and meta-analysis of magnetic resonance imaging studies - Table 3, part 2
10	0.00001	−0.28	Brain volume in first-episode schizophrenia: systematic review and meta-analysis of magnetic resonance imaging studies - Table 2
11	0.00001	0.44	Major Depressive Disorder Neuroimaging Database - Lateral ventricles, total - Statistics
12	0.00008	−0.25	Systematic reviews in health care: meta-analysis in context - Streptokinase dataset
13	0.00014	0.54	Major Depressive Disorder Neuroimaging Database - Cerebrospinal fluid, total
14	0.00129	0.61	Obsessive-compulsive disorder Neuroimaging Database - Thalamus
15	0.00168	0.59	Obsessive-compulsive disorder Neuroimaging Database - Right thalamus
16	0.00222	0.58	Obsessive-compulsive disorder Neuroimaging Database - Left thalamus
17	0.00385	−0.07	Comparative efficacy of NaF and SMFP dentifrices in caries prevention: a meta-analysis overview - Data
18	0.00460	0.31	Progressive lateral ventricular enlargement in schizophrenia: a meta-analysis of longitudinal MRI studies - Table 3, part 1
19	0.00627	−0.22	Major Depressive Disorder Neuroimaging Database - Caudate, total - Statistics
20	0.01210	−0.34	Major Depressive Disorder Neuroimaging Database - Thalamus, total - Statistics

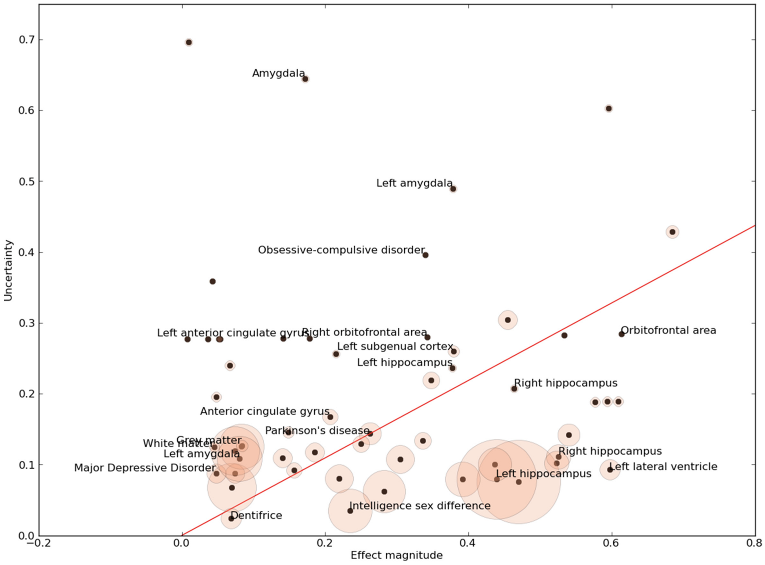


Fig. 5. Results from mass meta-analyses shown in a L'Abbé-like plot and constructed by calling the web script multiple times. Each dot corresponds to a meta-analysis. Uncertainty as a function of effect size with size of each dot determined by the number of studies in each meta-analysis. The line indicates 0.05-significance. To avoid overlapping texts labels appear only on a subset of meta-analyses and only with the primary topic.

Google Chart Tools.[2] It takes around 10 s to download and compute the 25 meta-analyses represented on the page for obsessive-compulsive disorder neuroimaging studies. In this case the meta-analysis for the pituitary gland has the largest magnitude of the effect size and thus placed to the extreme right in the plot. As redness indicates heterogeneity among the included studies in the meta-analysis the plot also shows that this particular region has the highest heterogeneity compared to the other 24 meta-analyses. The forest plot in Fig. 3 also display this heterogeneity.

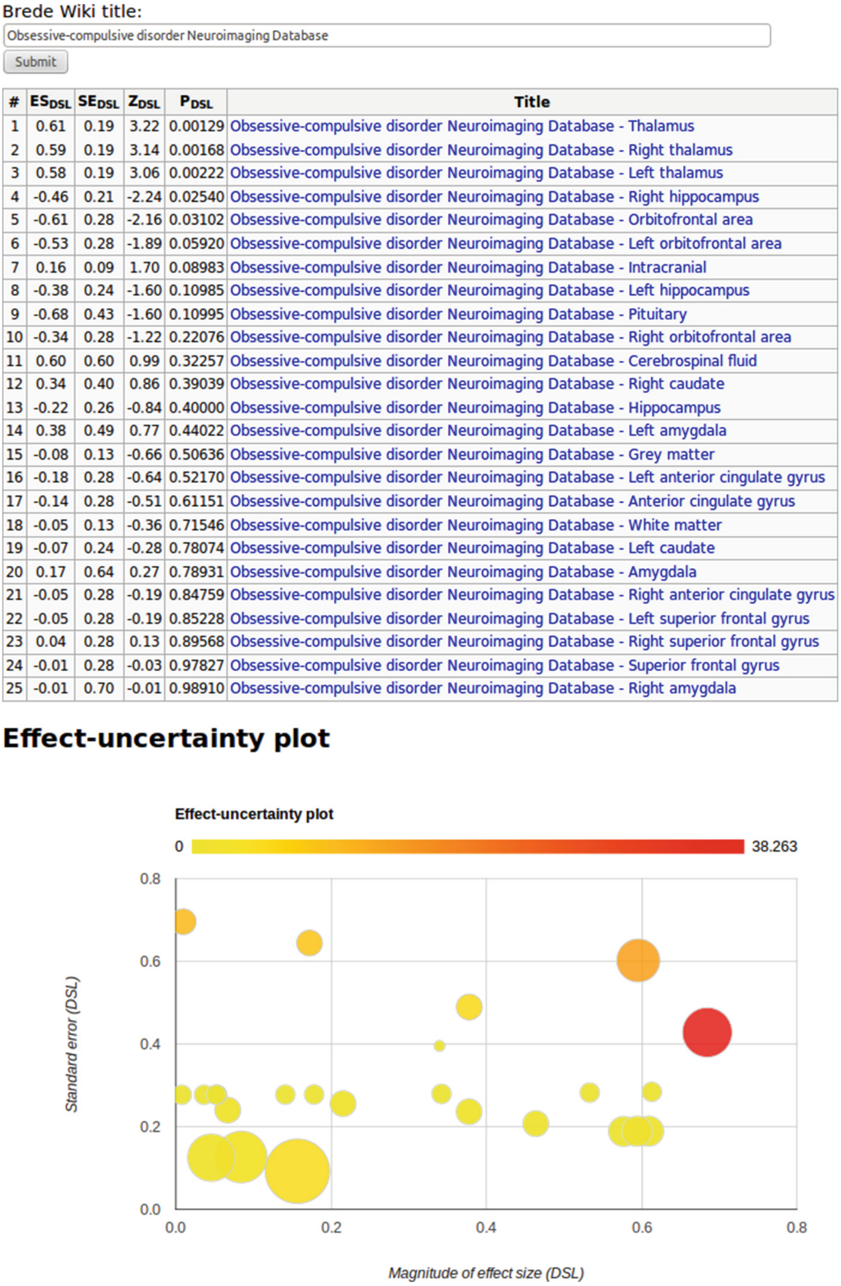
5 Discussion

By using MediaWiki in our present system we exploit the template facility to capture structured information and use free-form wikitext for annotation and comment on the individual scientific papers, — as in semantic academic annotation wikis *AcaWiki*[3] and *WikiPapers*.[4] We can also use the pages of the wiki as a simple means to keep track of the status of the papers considered for the meta-analysis: potentially eligible, eligible, partially entered and fully entered.

[2] https://developers.google.com/chart/interactive/docs/gallery/bubblechart.

[3] http://acawiki.org.

[4] http://wikipapers.referata.com.



Brede Wiki title:

Obsessive-compulsive disorder Neuroimaging Database

Submit

#	$\mathrm{ES}_{\mathrm{DSL}}$	$\mathrm{SE}_{\mathrm{DSL}}$	$\mathrm{Z}_{\mathrm{DSL}}$	$\mathrm{P}_{\mathrm{DSL}}$	Title
1	0.61	0.19	3.22	0.00129	Obsessive-compulsive disorder Neuroimaging Database - Thalamus
2	0.59	0.19	3.14	0.00168	Obsessive-compulsive disorder Neuroimaging Database - Right thalamus
3	0.58	0.19	3.06	0.00222	Obsessive-compulsive disorder Neuroimaging Database - Left thalamus
4	-0.46	0.21	-2.24	0.02540	Obsessive-compulsive disorder Neuroimaging Database - Right hippocampus
5	-0.61	0.28	-2.16	0.03102	Obsessive-compulsive disorder Neuroimaging Database - Orbitofrontal area
6	-0.53	0.28	-1.89	0.05920	Obsessive-compulsive disorder Neuroimaging Database - Left orbitofrontal area
7	0.16	0.09	1.70	0.08983	Obsessive-compulsive disorder Neuroimaging Database - Intracranial
8	-0.38	0.24	-1.60	0.10985	Obsessive-compulsive disorder Neuroimaging Database - Left hippocampus
9	-0.68	0.43	-1.60	0.10995	Obsessive-compulsive disorder Neuroimaging Database - Pituitary
10	-0.34	0.28	-1.22	0.22076	Obsessive-compulsive disorder Neuroimaging Database - Right orbitofrontal area
11	0.60	0.60	0.99	0.32257	Obsessive-compulsive disorder Neuroimaging Database - Cerebrospinal fluid
12	0.34	0.40	0.86	0.39039	Obsessive-compulsive disorder Neuroimaging Database - Right caudate
13	-0.22	0.26	-0.84	0.40000	Obsessive-compulsive disorder Neuroimaging Database - Hippocampus
14	0.38	0.49	0.77	0.44022	Obsessive-compulsive disorder Neuroimaging Database - Left amygdala
15	-0.08	0.13	-0.66	0.50636	Obsessive-compulsive disorder Neuroimaging Database - Grey matter
16	-0.18	0.28	-0.64	0.52170	Obsessive-compulsive disorder Neuroimaging Database - Left anterior cingulate gyrus
17	-0.14	0.28	-0.51	0.61151	Obsessive-compulsive disorder Neuroimaging Database - Anterior cingulate gyrus
18	-0.05	0.13	-0.36	0.71546	Obsessive-compulsive disorder Neuroimaging Database - White matter
19	-0.07	0.24	-0.28	0.78074	Obsessive-compulsive disorder Neuroimaging Database - Left caudate
20	0.17	0.64	0.27	0.78931	Obsessive-compulsive disorder Neuroimaging Database - Amygdala
21	-0.05	0.28	-0.19	0.84759	Obsessive-compulsive disorder Neuroimaging Database - Right anterior cingulate gyrus
22	-0.05	0.28	-0.19	0.85228	Obsessive-compulsive disorder Neuroimaging Database - Left superior frontal gyrus
23	0.04	0.28	0.13	0.89568	Obsessive-compulsive disorder Neuroimaging Database - Right superior frontal gyrus
24	-0.01	0.28	-0.03	0.97827	Obsessive-compulsive disorder Neuroimaging Database - Superior frontal gyrus
25	-0.01	0.70	-0.01	0.98910	Obsessive-compulsive disorder Neuroimaging Database - Right amygdala

Effect-uncertainty plot

Fig. 6. Screenshot from online mass meta-analyses with data from structural neuroimaging studies of obsessive-compulsive disorder in a table and a L'Abbé-like effect-uncertainty plot. Each row in the table and each dot in the chart correspond to a meta-analysis with the size of the dots controlled by the number of subjects in each meta-analysis and the color determined from Cochran's homogeneity test.

The *Internet Brain Volume Database* (IBVD)[5] also records structural neuroimaging data and presents the data online with visualization [20]. It does not entirely meet our needs as the database does not match associated patient and control groups. Nevertheless, the data recorded in IBVD and our system have many overlapping fields, and on pages for papers in the Brede Wiki we enter the IBVD paper identifier, so we can make deep links at the bibliographic level.

Why not Semantic MediaWiki? Semantic MediaWiki (SMW) may query text and numerical data, though has not had the ability to make complex computations. The *Semantic Result Formats*[6] extension includes average, sum, product and count result formats enabling simple computations of a series of numerical values, but insufficiently for the kind of computations we require. The data for meta-analysis form an n-ary data record (mean, standard deviation, number of subjects, labels) so either individual SMW pages should store each data record or we should invoke the n-ary functionality in *Semantic Internal Objects*[7] SMW extension, SMW *record* or the recently-introduced *subobject* SMW functionality. We have not investigated whether these tools provide convenient means for representing our data. The Brede Wiki can export its ontologies defined in MediaWiki template to SKOS. Our future research can consider RDFication of the CSV information through the SCOVO format [21].

Some methods for systematic reviews make two independent data-extractions that require consolidation and resolution of discrepancies to counter data extraction errors, that sometimes appear in meta-analyses [22]. Our simple system does not directly support such a process.

We wrote the web service in Python, where NumPy makes vector computation available and SciPy provides statistical methods necessary for the computation. In a future PHP implementation the script could more closely integrate with the wiki as either a MediaWiki or a SMW extension. Furthermore, new developments in 2012 in MediaWiki with support for the Lua programming language and Wikidata [23] for structured data in MediaWiki have relevance for our system. With the flexibility of Lua a wiki user can write templates that format CSV information into MediaWiki tables and perform numerical computations on the data, — possibly to such an extent that it will make the table rendering MediaWiki extension and the meta-analytic web script unnecessary.

6 Conclusion

A wiki built from standard components provides a inexpensive solution with means to manage meta-analytic data in a collaborative environment. The general framework allows not only the meta-analysis of neuroimaging-derived data but has the potential for managing and analyzing data from many other domains. The system gives an overview of the important effects and uncertainties across numerous meta-analyses that multiple wiki users may extend collaboratively.

[5] http://www.cma.mgh.harvard.edu/ibvd/.

[6] http://semantic-mediawiki.org/wiki/Semantic_Result_Formats.

[7] http://www.mediawiki.org/wiki/Extension:Semantic_Internal_Objects.

We regard the environment as an Open Science system that makes methods and data immediately available in human and machine readable form.

References

1. Elamin, M.B., Flynn, D.N., Bassler, D., Briel, M., Alonso-Coello, P., Karanicolas, P.J., Guyatt, G.H., Malaga, G., Furukawa, T.A., Kunz, R., Schunemann, H., Murad, M.H., Barbui, C., Cipriani, A., Montori, V.M.: Choice of data extraction tools for systematic reviews depends on resources and review complexity. J. Clin. Epidemiol. **62**(5), 506–510 (2009)
2. Kempton, M.J., Salvador, Z., Munafo, M.R., Geddes, J.R., Simmons, A., Frangou, S., Williams, S.C.R.: Structural neuroimaging studies in major depressive disorder: meta-analysis and comparison with bipolar disorder. Arch. Gen. Psychiatry **68**(7), 675–690 (2011)
3. Kempton, M.J., Geddes, J.R., Ettinger, U., Williams, S.C.R., Grasby, P.M.: Meta-analysis, database, and meta-regression of 98 structural imaging studies in bipolar disorder. Arch. Gen. Psychiatry **65**(9), 1017–1032 (2008)
4. Nielsen, F.Å.: A fielded wiki for personality genetics. In: Proceedings of the 6th International Symposium on Wikis and Open Collaboration, New York. ACM (2010)
5. Nielsen, F.Å.: Brede Wiki: Neuroscience data structured in a wiki. In: Lange, C., Schaffert, S., Skaf-Molli, H., Völkel, M. (eds.) Proceedings of the Fourth Workshop on Semantic Wikis - The Semantic Wiki Web. Volume 464 of CEUR Workshop Proceedings., Aachen, Germany, RWTH Aachen University, pp. 129–133, June 2009
6. Turner, J.A., Laird, A.R.: The Cognitive Paradigm Ontology: design and application. Neuroinformatics **10**(1), 57–66 (2011)
7. Poldrack, R.A., Kittur, A., Kalar, D., Miller, E., Seppa, C., Gil, Y., Parker, D.S., Sabb, F.W., Bilder, R.M.: The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. Front. Neuroinformatics **5**, 17 (2011)
8. Steen, R.G., Mull, C., McClure, R., Hamer, R.M., Lieberman, J.A.: Brain volume in first-episode schizophrenia: systematic review and meta-analysis of magnetic resonance imaging studies. Br. J. Psychiatry **188**, 510–518 (2006)
9. Kempton, M.J., Stahl, D., Williams, S.C.R., DeLisi, L.E.: Progressive lateral ventricular enlargement in schizophrenia: a meta-analysis of longitudinal MRI studies. Schizophr. Res. **120**(1–3), 54–62 (2010)
10. Hartung, J., Knapp, G., Sinha, B.K.: Statistical Meta-Analysis with Applications. Wiley Series in Probability and Statistics. Wiley, Hoboken (2008)
11. Irwing, P., Lynn, R.: Sex differences in means and variability on the progressive matrices in university students: A meta-analysis. Br. J. Psychol. **96**(4), 505–524 (2005)
12. Ince, D.C., Hatton, L., Graham-Cumming, J.: The case for open computer programs. Nature **482**(7386), 485–488 (2012)
13. Shaffer, J.P.: Caution on the use of variance ratios: a comment. Rev. Educ. Res. **62**(4), 429–432 (1992)
14. Nielsen, F.Å., Hansen, L.K.: Modeling of activation data in the BrainMap$^{\text{TM}}$ database: Detection of outliers. Hum. Brain Mapp. **15**(3), 146–156 (2002)
15. Turkeltaub, P., Eden, G.F., Jones, K.M., Zeffiro, T.A.: A novel meta-analysis technique applied to single word reading. NeuroImage **13**(6 (part 2)), S272 (2001)

16. Radua, J., Mataix-Cols, D.: Meta-analytic methods for neuroimaging data explained. Biol. Mood Anxiety Disord. **2**, 6 (2012)
17. MacMaster, F.P., Russell, A., Mirza, Y., Keshavan, M.S., Banerjee, S.P., Bhandari, R., Boyd, C., Lynch, M., Rose, M., Ivey, J., Moore, G.J., Rosenberg, D.R.: Pituitary volume in pediatric obsessive-compulsive disorder. Biol. Psychiatry **59**(3), 252–257 (2006)
18. Atmaca, M., Yildirim, H., Ozler, S., Koc, M., Kara, B., Sec, S.: Smaller pituitary volume in adult patients with obsessive-compulsive disorder. Psychiatry Clin. Neurosci. **63**(4), 516–520 (2009)
19. Jung, M.H., Huh, M.J., Kang, D.H., Choi, J.S., Jung, W.H., Jang, J.H., Park, J.Y., Han, J.Y., Choi, C.H., Kwon, J.S.: Volumetric differences in the pituitary between drug-naïve and medicated male patients with obsessive-compulsive disorder. Prog. Neuro-psychopharmacol. Biol. Psychiatry **33**(4), 605–609 (2009)
20. Kennedy, D.N., Hodge, S.M., Gao, Y., Frazier, J.A., Haselgrove, C.: The Internet Brain Volume Database: a public resource for storage and retrieval of volumetric data. Neuroinformatics **10**(2), 129–140 (2011)
21. Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., Ayers, D.: SCOVO: using statistics on the web of data. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 708–722. Springer, Heidelberg (2009)
22. Gøtzsche, P.C., Hróbjartsson, A., Maric, K., Tendal, B.: Data extraction errors in meta-analyses that use standardized mean differences. JAMA **298**(4), 430–437 (2007)
23. Vrandečić, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st international conference companion on World Wide Web, New York, NY, USA, Association for Computing Machinery, pp. 1063–1064 (2012)

Linked Data and Linked APIs: Similarities, Differences, and Challenges

Ruben Verborgh[1](✉), Thomas Steiner[2], Rik Van de Walle[1], and Joaquim Gabarro[2]

[1] ELIS – Multimedia Lab, Ghent University – iMinds,
Gaston Crommenlaan 8 Bus 201, 9050 Ledeberg-ghent, Belgium
{ruben.verborgh,rik.vandewalle}@ugent.be

[2] Department LSI, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
{tsteiner,gabarro}@lsi.upc.edu

Abstract. In an often retweeted Twitter post, entrepreneur and software architect Inge Henriksen described the relation of Web 1.0 to Web 3.0 as: "*Web 1.0 connected humans with machines. Web 2.0 connected humans with humans. Web 3.0 connects machines with machines.*" On the one hand, an incredible amount of valuable data is described by billions of triples, machine-accessible and interconnected thanks to the promises of Linked Data. On the other hand, REST is a scalable, resource-oriented architectural style that, like the Linked Data vision, recognizes the importance of links between resources. Hypermedia APIs are resources, too—albeit dynamic ones—and unfortunately, neither Linked Data principles, nor the REST-implied self-descriptiveness of hypermedia APIs sufficiently describe them to allow for long-envisioned realizations like automatic service discovery and composition. We argue that describing inter-resource links—similarly to what the Linked Data movement has done for data—is the key to machine-driven consumption of APIs. In this paper, we explain how the description format RESTdesc captures the functionality of APIs by explaining the effect of dynamic interactions, effectively complementing the Linked Data vision.

1 Introduction

1.1 The Web API Simplification Movement

The number of Web APIs has increased at a tremendous rate during the past few years. ProgrammableWeb, a major catalog of Web APIs and services, consisted of 6,000 entries as of May 2012 [13], twice the amount compared to the year before [12]. More than 4,000 of those entries carry the label "REST", meaning they are light-weight Web APIs, also called HTTP interfaces, as opposed to the more heavy-weight RPC-style Web services, often using SOAP. While there are certainly different viewpoints to take into account—especially when comparing enterprise

This paper is an extended version of the Linked APIs for the Semantic Web (LAPIS) workshop paper titled *"The Missing Links"* [34].

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 272–284, 2015.
DOI: 10.1007/978-3-662-46641-4_20

SOA architects and *mash-up* developers—Web developers in general welcome this simplification movement on the dynamic side of the Web. It is common practice to intermix Web APIs from different sources and, in the sense of emergence, create something new (commonly called mashup applications), where the whole is greater than the sum of its parts. If Web application development today was compared to the world of toys, the LEGO figures would ride the Playmobil horses and fight with Star Wars collectibles swords. However, a lot of manual plumbing is required to make this work: while Web APIs bare the potential to be composed straightforwardly, they lack the semantics to do this in an automated way [26].

In the past, we have introduced the Web API description method RESTdesc [35], which aims to provide the semantics necessary to enable automated API consumption and composition, in the same way that ontologies provide the semantics to static data. In this paper, we want to look at Web APIs from a Semantic Web perspective: investigating how APIs differ from data, what they have in common, and how they could work together on the Web. An important piece of the puzzle is to realize that the resource-oriented way of looking at APIs is similar to the Linked Data vision on data.

1.2 Linked Data Explains the Static Side, RESTdesc the Dynamic Side

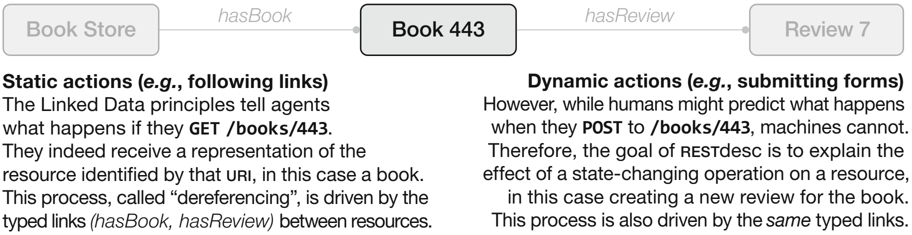
To clarify what we mean with this statement, we need to take a step back and think about the Web in its most abstract form. What we see are *resources*, with an unparalleled variety, and an ever increasing number of *links* between them [10]. Resources and their representations make up the essence of the Web [16], while the Linked Data vision made us all realize again the crucial role that links play therein. Indeed, links have been the catalysts of the success of the human Web, and they continue to prove their strengths on the Semantic Web [8]. The representations of resources—and therefore data—are given meaning by links, corresponding to well-defined RDF predicates.

Given the importance of links, one can wonder why they seem absent on the service-side of the Web, where interactions are mostly driven by static controls such as message templates and URI construction rules. These controls have to be known in advance, unlike Linked Data controls (*i.e.*, the links between resources), which are consumed at runtime. When Fielding redesigned the HTTP specification [15], he had a resource-oriented model in mind where hypermedia drives Web applications: Representational State Transfer (REST, [16]). He later clarified that the hypermedia constraint imposed by REST demands that representations of a resource should contain controls that guide hypermedia consumers to possible next steps or resources [14]. Consequently, modeling Web APIs the REST way leads to the same resources-and-links paradigm that is at the core of the human Web, which has HTML links and forms, and the Semantic Web, which has RDF links between resources.

In all fairness, REST APIs—as defined by Fielding—are scarce. While many APIs carry the "REST" label, few actually obey the hypermedia constraint, and, even worse, some of them do not correctly adhere to the defined HTTP semantics [26]. Hypermedia-driven APIs are vastly outnumbered by plain HTTP and RPC interfaces. However, this can be compared to the larger presence of unstructured and unlinked data on the Web compared to Linked Data. Therefore, the scarceness doesn't change the status of resource- and link-orientedness as well-suited model for automated agents to perform static *and* dynamic interactions.

Currently, the main obstacle for automated agents that want to consume Web APIs is that they cannot predict what effect a *state-changing* operation will have. Linked Data gives the answer for *information-retrieving* operations, known as *dereferencing*. Performing a `GET` operation on a resource's URI will provide the agent with information about that resource. But what happens when the agent performs a `POST` operation on the *same* resource? Reference [35] Since Fielding suggests the controls (*e.g.,* links and forms) should point to possible next steps or resources, it is obvious *how* the state change happens. However, *what* this state change will bring might be obvious to humans, but is still unknown to machines. Therefore, in this paper, we zoom in on how the description format RESTdesc explains to agents what will happen if *state-changing* operations are performed on a resource, complementary to the Linked Data principles that explain the same for static operations.

This complementary nature is illustrated in Fig. 1, which positions Linked Data and RESTdesc. The example shows a book store that offers several books, each of which can have several reviews. The resources might be available on the Web for human visitors as HTML representations with (possibly typed) hyperlinks in between. To make the store machine-accessible, the server might additionally serve RDF representations, in which the relations are RDF predicates, which eventually can lead to Linked Data. However, the Linked Data principles only explain how to *browse* books and reviews, whereas the HTML representations provide the controls to *add* reviews. RESTdesc bridges this gap by explaining the functionality of this Web API, in a representation-independent way.



Static actions (*e.g.*, following links)
The Linked Data principles tell agents what happens if they **GET /books/443**. They indeed receive a representation of the resource identified by that URI, in this case a book. This process, called "dereferencing", is driven by the typed links *(hasBook, hasReview)* between resources.

Dynamic actions (*e.g.*, submitting forms)
However, while humans might predict what happens when they **POST** to **/books/443**, machines cannot. Therefore, the goal of RESTdesc is to explain the effect of a state-changing operation on a resource, in this case creating a new review for the book. This process is also driven by the *same* typed links.

Fig. 1. RESTdesc complements Linked Data by explaining a hyperlink's *dynamic* functionality in machine-readable form. For instance, we can express with RESTdesc what happens when agents use `POST` on a linked resource instead of `GET`.

This paper starts with a description of related work in Sect. 2, then highlights the differences and similarities of Linked Data and hypermedia APIs in Sect. 3, zooming in on the gaps that need to be bridged. Section 4 continues with an illustration of the role RESTdesc can play herein by formally expressing the relationship between resources in a hypermedia API. Finally, Sect. 5 looks back on the discussed topics and ends by indicating the importance of hypermedia-driven APIs on the Web for autonomous agents.

2 Related Work

Description of Web services or APIs for automated use has been on the Web since before the advent of the Semantic Web (notably WSDL [11]), and played an important part during the beginning of the Semantic Web's inception. Several of the first initiatives are well-known: OWL-S [29], which evolved from DAML-S [3], and the conceptually different WSMO [24,32]. These formats target what are called "Big" Web services [31], which function in a message-passing or Remote Procedure Call (RPC) paradigm. While these models use Semantic Web elements such as ontologies, they predate the Linked Data vision and the recent revaluation of REST APIs. Neither OWL-S nor WSMO have stood the test of time, as extensive Web searches did not reveal substantial real-world usage. We therefore focus on more recent research projects that have design goals similar to RESTdesc, *e.g.*, a focus on functionality and/or hypermedia APIs.

Several methods aim to enhance existing technologies to deliver annotations of Web APIs. HTML for RESTful Services (hRESTS, [21]) is a microformat to annotate HTML descriptions of Web APIs in a machine-processable way. SA-REST [18] provides an extension of hRESTS that describes other facets such as data formats and programming language bindings. MicroWSMO [22,25], an extension to SAWSDL that enables the annotation of RESTful services, supports the discovery, composition, and invocation of Web APIs. The Semantic Web sErvices Editing Tool (SWEET, [27]) is an editor that supports the creation of mashups through semantic annotations with MicroWSMO and other technologies. A shared API description model, providing common grounds for enhancing APIs with semantic annotations to overcome the current heterogeneity, has been proposed in the context of the SOA4All project [28].

The Resource Linking Language (ReLL, [1]) features media types, resource types, and link types as first class citizens for descriptions. It offers a metamodel and an associated XML Schema to capture these aspects formally. The RESTler crawler [1] finds RESTful services based on ReLL descriptions. The authors also propose a method for ReLL API composition [2] using Petri nets to describe the machine-client navigation.

Linked Open Services (LOS, [23]) have an HTTP API approach, in which SPARQL graph patterns identify the offered functionality. Part of the project's scope concerns the lifting and lowering of existing services, since many of them do not expose their data in a semantic format yet. A difference with RESTdesc is that LOS APIs are not committed to the hypermedia constraint, whereas the hypermedia-driven consumption of APIs is a central concept in RESTdesc.

Linked Data Services (LIDS, [33]) have a similar notion of input and output graphs. They use the input data to construct a resource's URI, as opposed to

LOS, which sends input data in the request body. The result is an API whose interactions are thus in a sense solely *form*-based—the form structure being defined by the unbound variables in the input graph pattern. In addition to forms (not discussed in this paper), RESTdesc also aims to support the *link* part of the hypermedia control set.

3 A Joint Future for Linked Data and Hypermedia APIS

We start this section with an essential definition to avoid misunderstandings on the thin ice of REST, RESTlike, and unRESTful APIS:

Hypermedia Web APIS are interfaces to retrieve and manipulate *resources* according to the HTTP method semantics, serving *representations* of these resources along with the *controls* to advance through the interface [14].[1]

Striking parallels between Linked Data and hypermedia APIS exist—and this is not a coincidence, since both are closely tied to the original visions and architecture of the Web. One of the common elements are **resources**: concepts in Linked Data are identified by one or multiple URIS, which, when requested through HTTP `GET`, lead to information about that concept. Hypermedia APIS are similarly structured as concepts or resources, with the constraints that every URI should identify a resource and that the HTTP methods should be used conform to the HTTP specification [15]. The semantics of the `GET` method have therein been defined as "obtaining the information identified by the URI", which, unsurprisingly, matches the Linked Data purpose [19].

The other common element are **links**: as the name implies, they play a vital role in Linked Data, and they are at the heart of the Semantic Web. Links give a concept's data meaning beyond its own context. More concretely, if an agent does not understand what a data property means, it can look up that property because its link is an HTTP URI. The same applies to hypermedia APIS: the controls, telling us how other resources relate to the current resource, can be links. Details on the nature of the relation are conveyed by link types, which can have the same URIS as Linked Data properties [30].

In essence, one could see the whole Linking Open Data Cloud [9] as a large, distributed hypermedia application. This is in fact how its usage is encouraged: an agent starts from one resource and can make its way through the whole cloud, just by "following its nose", thanks to the links. However, it only provides a subset of the possibilities of what we expect from a hypermedia API: merely *retrieval* operations are supported. Yet, the role of links here remains important: browsing billions of triples in billions of resources would otherwise prove difficult.

An interesting aspect of REST is that it does not matter whether the resources and triples already exist. They can either be part of documents, or be the result

[1] Hypermedia APIS are synonymous to "REST APIS or services, *in the sense as defined by Fielding*" [16]. This last clarification is important, since many APIS that were given a "REST" label do *not*, or only partially, adhere to Fielding's definition, which is why we use the term "hypermedia API" to distinguish the *intended* meaning [20].

of a service invocation—but the agent does not have to know and does not have to care. For example, a huge dataset of natural numbers has been made available as Linked Data [38], yet the information of each number is not static, but instead generated dynamically when an agent dereferences its URI. This dataset is thus what we would traditionally consider a "service", but thanks to the REST principles, it manifests itself as just another set of linked resources.

Nevertheless, we often associate the concept of services additionally with action-driven behavior, for example, allowing us to post a comment or order tickets. In a REST architectural model, these actions are captured by the modification or creation of resources, linked to existing resources. While these and similar actions are very common on the human Web and on the Web of services, the Semantic Web still struggles with state-changing operations [7]. Several mechanisms are there (*e.g.,* SPARQL `UPDATE` [17]), but issues such as authorization and security still impede wide adoption. Consequently, the Linked Data vision must in the meantime assume that the publisher and consumer sides are distinct, *i.e.*, that consumers of Linked Data will not need to perform write operations. This simplifying assumption has its benefits—just look at the overwhelming amount of data—but will not be sufficient for the vision of autonomous agents that require actions in the real world. Indeed, as the comment and ticket examples indicate, many interactions we perform on a daily basis involve write actions. Therefore, in the next section, we will look at the requirements of agents for browsing full hypermedia APIs, which offer both information-retrieving and state-changing operations.

4 RESTdesc Describes Hypermedia Links

4.1 Example Scenario

As an example, let us consider the situation of Fig. 1. Starting from the book store's main URI, an agent discovers resources in a fully hypermedia-driven way. Its steps might be the following:

1. `GET` a representation of the index resource at `/`.
2. **Find** a `hasBook` link in this representation titled "*The Catcher in the Rye*".
3. `GET` a representation of this linked resource at `/books/443`.
4. **Find** a `hasReview` link in this representation.
5. `GET` a representation of this linked resource at `/books/443/reviews/7`.

This way of working is hypermedia-driven, because the agent only follows the representation-supplied controls (*e.g.*, links) to go from one step to the next.

4.2 Understanding the GET Operations

As an introduction to RESTdesc, we will know discuss the RESTdesc description that is associated with the action of retrieving a book's representation. RESTdesc descriptions are expressed in Notation3 (N3, [5]), a small superset of RDF put forward by Tim Berners-Lee. N3 adds support for quantification, necessary to create statements concerning *all* resources instead of only specific ones. Without this explicit support, the quantifications should have to be expressed indirectly. One other possibility to express this is to wrap SPARQL expressions inside string literals, which is the method used by LIDS [33]. The quantification constructs in N3 enable to integrate the semantics directly, whereas for instance SPARQL expressions have to be interpreted separately.

```
@prefix ex: <http://example.org/book-store#>.
@prefix http: <http://www.w3.org/2011/http#>.

{
  ?store ex:hasBook ?book. ❶
}
=>
{
  _:request http:methodName "GET"; ❷
            http:requestURI ?book;
            http:resp [ http:body ?book ].

  ?book ex:hasTitle ?title; ❸
        ex:hasAuthor ?author;
        ex:hasReview ?review.
}.
```

Listing 1. RESTdesc describes the act of retrieving a book by explaining the associated hypermedia link.

Listing 1 displays a description of the `GET` operation on the `hasBook` link type and serves as an illustration of several common aspects of RESTdesc descriptions. Every description is a logic implication. The logical foundations of N3 (N3Logic, [6]) define an operational semantics, *i.e.*, RESTdesc descriptions are N3 rules that can be instantiated and executed by a reasoner. As indicated in Listing 1, it is convenient to examine the description in three parts:

❶ **IF** you obtain a book's URI from a `hasBook` hyperlink
❷ **THEN** you can make a `POST` request to that URI
❸ to retrieve a representation of this book.

Below, we discuss some important aspects of this description.

Firstly, the explicit quantification makes agent understand that the book in the antecedent and the conclusion are the same. The `?book` variable can be instantiated with a concrete instance. For example, if an agent finds a `hasBook` link from the store `/` to the book `/books/443/`, it can instantiate the description of Listing 1 into the RDF fragment in Listing 2. This fragment details the instructions an agent needs to execute. Since this request in these instructions has not been executed, the resulting values are not known yet. However, the reasoner has instantiated them with blank nodes (`title1`, `author1`, and `review1`).

```
@prefix ex: <http://example.org/book-store#>.
@prefix http: <http://www.w3.org/2011/http#>.

</> ex:hasBook </books/443>.
_:request1 http:methodName"GET";
           http:requestURI </books/443>;
           http:resp [ http:body </books/443> ].

</books/443> ex:hasTitle _:title1;
             ex:hasauthor _:author1;
             ex:hasreview _:review1.
```

Listing 2. The instantiation of the RESTdesc description in Listing 1 yields an RDF fragment with instructions.

After a successful execution of the request, these blank nodes can be substituted by the actual data received from the server.

Secondly, it might seem strange at first sight that the request ❷ is part of the consequent, and not of the antecedent. After all, it is the existence of the link ❶ *and* the execution of the request ❷ that lead to obtaining information about the book ❸. However, RESTdesc adopts a different view here. In fact, it is because of the existence of the link ❶, that a request exists ❷ which will lead to the information ❸. The word *exists* is important here: the description indeed states that a request *exists* that will deliver the information, not that *all* requests with the given parameters will lead there. In other words: the request is existentially quantified, not universally. This notion is important, because it models the world more accurately. For example, a given request could fail because of connection issues or might require additional authentication.

Thirdly, RESTdesc does not specify what representations should look like. This is a central part of the RESTphilosophy. While RESTdesc describes the relations between resources and the result of actions performed on them, the selection of the right representation should happen at runtime by making use of the content negotiation mechanism of the HTTP protocol. For example, Listing 1 states that the retrieved resource will have a title and an author. The description does, however, *not* imply that this information will be supplied in RDF or any other format. While it seems logical that an agent would ask for an RDF representation (since the agent uses Semantic Web technologies internally), this is by no means a requirement. The actual representation could be in XML, JSON, HTML, or any other format. This opens possibilities to work with non-textual data, such as images and video [36]. However, the major benefit of RDF representations is that their contents are self-describing and can therefore be automatically interpreted by machines.

The final and most crucial remark is that the necessity of the description in Listing 1 can be questioned. After all, why would we want to describe a `GET` request? It seems unnecessary, because of the following two reasons: first, the Linked Data principles already tell us what happens with `GET` request—receiving a representation of the resource with the corresponding URI (which is called "dereferencing"). Second, even if these principles did not apply, an agent could safely execute the request, since the HTTP specification indicates `GET` should not change application state [15]. We fully agree here: RESTdesc is designed to describe state-changing operations whose result is resource-dependent, the primary verb being `POST`. Therefore, the next subsection illustrates a `POST` request, which fully illustrates RESTdesc's capabilities. RESTdesc *can* however be used for `GET`, which is interesting (*a*) for situations where ontological constructs are insufficient to describe a complex resource relationship and (*b*) to convey an expectation of what properties a representation will contain (*e.g.*, **`hasTitle`**, **`hasAuthor`**, ...).

```
@prefix ex: <http://example.org/book-store#>.
@prefix http: <http://www.w3.org/2011/http#>.

{
  ?store ex:hasBook ?book. ❶
  ?review ex:author _:author;
          ex:rating _:rating;
          ex:contents _:text.
}
=>
{
  _:request http:methodName "POST"; ❷
            http:requestURI ?book;
            http:body ?review;
            http:resp [ http:body ?book ].

  ?book ex:hasReview ?review. ❸
}.
```

Listing 3. RESTdesc describes the act of posting a review by explaining the associated hypermedia link.

4.3 Understanding POST Requests

The situation is completely different for `POST` requests because, unlike with `GET` and other safe requests, the agent cannot carelessly issue a `POST` request in one of the steps, since (*a*) it cannot predict what the result will be and (*b*) testing what the result is can have unwanted consequences, as `POST` is *unsafe* [15]. Furthermore, it cannot determine what body it should send along with the `POST` request. Although some representation formats provide forms (*e.g.*, HTML and Atom), others lack form functionality (*e.g.*, RDF, although proposals exist [4]), but in either case, it is unclear how the result relates to the submitted data.

Let us therefore examine the description in Listing 3, which can similarly be interpreted in three parts:

❶ IF you obtain a book's URI from a **`hasBook`** hyperlink
❷ THEN you can make a `POST` request to that URI
❸ to add a review with the supplied parameters to this book.

This enables agents to understand what data they can send along with a POST request and how this data will influence the outcome of the request.

Note how, in this example, the precondition is more restricting: the agent needs to have access to a review before the request can be executed. Also, this review is necessary to construct the request: it should be placed inside the HTTP request's POST body. Again, the exact representation of this body is not detailed, because agents and servers should be able to agree on the best representation at runtime. We do, however, get a suggestion of properties that should be present in the representation: an author, a rating, and a review text.

Now that the agent understands each of the steps, it is able to chain them together and actually execute each of the requests in the process.

4.4 Executing the Requests

Concretely, if the agent has been given the contents of a review *(author, rating, content)*, it can follow these hypermedia-driven steps:

1. GET the RESTdesc description of hasBook.[2]
2. GET a representation of the index resource at /.
3. **Find** a hasBook link in this representation titled "*The Catcher in the Rye*".
4. **Instantiate** the description with the review and found link.
5. POST the review, as instructed by the description, at /books/443.

Note again how only hypermedia controls are used to get from one step to the next. The added value of RESTdesc here is to explain the agent in advance what effect the POST request will have, so it can decide whether to execute this request. In real-world applications, RESTdesc descriptions can be used for goal-driven API compositions [37]. For instance, the user can supply the review parameters as input, and ask that it is submitted to a certain book.

5 Conclusion

The Linked Data vision strives to connect data on the Web, making it available in a machine-processable format. Hypermedia APIs similarly strive for connectedness of resources, but also consider the write side of interactions. Their goals are similar, and so are their tools: both make automated consumption of the Web available using the core principles of the HTTP architecture, featuring resources, representations, and links. However, dealing with state-changing operations requires automated agents to have expectations of what consequences their actions will have.

RESTdesc shows how existing Semantic Web technologies can be combined to explain the functionality of a Web API to those agents. It enables us to apply the Linked Data vision to hypermedia APIs by describing the meaning of links for

[2] RESTdesc discovery, *i.e.*, how to obtain RESTdesc descriptions, has been discussed earlier [36]. The agent could for example dereference the hasBook link.

state-changing operations. In that sense, it is a plea for more hypermedia APIs on the Web, as they beautifully incorporate the controls that future autonomous agents will need to browse the Web. Therefore, we believe it is time to transition today's services towards hypermedia APIs by adding the missing links.

Acknowledgments. The described activities were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research Flanders (FWO), and the European Union.

This work was partially supported by the European Commission under Grant No. 248296 FP7 (I-SEARCH project). Joaquim Gabarró is partially supported by TIN-2007-66523 (FORMALISM), and SGR 2009-2015 (ALCOM).

References

1. Alarcón, R., Wilde, E.: RESTler: crawling RESTful services. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1051–1052. ACM (2010). http://doi.acm.org/10.1145/1772690.1772799
2. Alarcon, R., Wilde, E., Bellido, J.: Hypermedia-driven RESTful service composition. In: Maximilien, E.M., Rossi, G., Yuan, S.-T., Ludwig, H., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6568, pp. 111–120. Springer, Heidelberg (2011). http://dx.doi.org/10.1007/978-3-642-19394-1_12
3. Ankolekar, A., et al.: DAML-S: web service description for the semantic web. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 348–363. Springer, Heidelberg (2002). http://eprints.soton.ac.uk/257342/1/ISWC2002-DAMLS.pdf
4. Baker, M.: RDF forms, 2003–2005. http://www.markbaker.ca/2003/05/RDF-Forms/
5. Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax. W3C Team Submission (2011). http://www.w3.org/TeamSubmission/n3/
6. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3LOGIC: a logical framework for the World Wide Web. Theory Pract. Log. Program. **8**(3), 249–269 (2008). http://arxiv.org/pdf/0711.1533v1.pdf
7. Berners-Lee, T., Cyganiak, R., Hausenblas, M., Presbrey, J., Seneviratne, O., Ureche, O.: Realising a read-write Web of Data, June 2009. http://web.mit.edu/presbrey/Public/rw-wod.pdf
8. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–43 (2001). http://www.scientificamerican.com/article.cfm?id=the-semantic-web
9. Bizer, C.: The emerging Web of Linked Data. Intell. Syst. **24**(5), 87–92 (2009). IEEE. http://lpis.csd.auth.gr/mtpx/sw/material/IEEE-IS/IS-24-5.pdf
10. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009). http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf
11. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S.: Unraveling the Web services Web: an introduction to SOAP, WSDL, AND UDDI. Internet Computing **6**(2), 86–93 (2002). IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=991449
12. DuVander, A.: 3,000 Web APIS: Trends from a quickly growing directory, March 2011. http://blog.programmableweb.com/2011/03/08/3000-web-apis/

13. DuVander, A.: 6,000 APIS: It's business, it's social and it's happening quickly, May 2012. http://blog.programmableweb.com/2012/05/22/6000-apis-its-business-its-social-and-its-happening-quickly/
14. Fielding, R.T.: REST APIS must be hypertext-driven. Untangled - Musings of Roy T. Fielding, October 2008. http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven
15. Fielding, R.T., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol - HTTP/1.1. Request for Comments: 2616, June 1999. http://tools.ietf.org/html/rfc2616
16. Fielding, R.T., Taylor, R.N.: Principled design of the modern Web architecture. Trans. Internet Technol. **2**(2), 115–150 (2002). http://dl.acm.org/citation.cfm?id=514185
17. Gearon, P., Passant, A., Polleres, A.: SPARQL 1.1 Update. W3C Working Draft, January 2012. http://www.w3.org/TR/sparql11-update/
18. Gomadam, K., Ranabahu, A., Sheth, A.: SA-REST: Semantic Annotation of Web Resources. W3C Member Submission. http://www.w3.org/Submission/SA-REST/
19. Hartig, O., Zhao, J.: Publishing and consuming provenance metadata on the web of linked data. In: McGuinness, D.L., Michaelis, J.R., Moreau, L. (eds.) IPAW 2010. LNCS, vol. 6378, pp. 78–90. Springer, Heidelberg (2010). http://www2.informatik.hu-berlin.de/~hartig/files/HartigZhao_Provenance_IPAW2010_Preprint.pdf
20. Klabnik, S.: REST is over, February 2012. http://blog.steveklabnik.com/posts/2012-02-23-rest-is-over
21. Kopecký, J., Gomadam, K., Vitvar, T.: hRESTS: an HTML microformat for describing RESTful Web services. In: Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology, pp. 619–625. IEEE Computer Society (2008). http://dx.doi.org/10.1109/WIIAT.2008.379
22. Kopecký, J., Vitvar, T.: MicroWSMO. WSMO Working Draft, February 2008. http://www.wsmo.org/TR/d38/v0.1/
23. Krummenacher, R., Norton, B., Marte, A.: Towards linked open services and processes. In: Berre, A.J., Gómez-Pérez, A., Tutschku, K., Fensel, D. (eds.) FIS 2010. LNCS, vol. 6369, pp. 68–77. Springer, Heidelberg (2010). http://www.linkedopenservices.org/publications/FIS2010.pdf
24. Lara, R., Roman, D., Polleres, A., Fensel, D.: A conceptual comparison of WSMO and OWL-S. In: (LJ) Zhang, L.-J., Jeckle, M. (eds.) ECOWS 2004. LNCS, vol. 3250, pp. 254–269. Springer, Heidelberg (2004). http://www.wsmo.org/2004/d4/d4.1/v0.1/20050106/d4.1v0.1_20050106.pdf
25. Maleshkova, M., Kopecký, J., Pedrinaci, C.: Adapting SAWSDL for semantic annotations of RESTful services. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 917–926. Springer, Heidelberg (2009). http://dx.doi.org/10.1007/978-3-642-05290-3_110
26. Maleshkova, M., Pedrinaci, C., Domingue, J.: Investigating Web on the World Wide Web. In: Proceedings of the 8th European Conference on Web Services, pp. 107–114. IEEE (2010). http://sweet-dev.open.ac.uk/war/Papers/mmaWebAPISurvey.pdf
27. Maleshkova, M., Pedrinaci, C., Domingue, J. Semantic annotation of web with APIS with SWEET, May 2010. http://oro.open.ac.uk/23095/
28. Maleshkova, M., Pedrinaci, C., Li, N., Kopecky, J., Domingue, J.: Lightweight semantics for automating the invocation of Web APIS. In: Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications, December 2011. http://sweet.kmi.open.ac.uk/pub/SOCA.pdf

29. Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., Mcguinness, D.L., Sirin, E., Srinivasan, N.: Bringing semantics to Web services with OWL-S. World Wide Web **10**, 243–277 (2007)
30. Nottingham, M.: Web Linking. Request for Comments: 5988, October 2010. http://tools.ietf.org/html/rfc5988
31. Pautasso, C., Zimmermann, O., Leymann, F.: RESTful Web services vs. "Big" Web services: making the right architectural decision. In: Proceedings of the 17th International Conference on World Wide Web, pp. 805–814. ACM, New York (2008). http://www.jopera.org/files/www2008-restws-pautasso-zimmermann-leymann.pdf
32. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. Appl. Ontol., 1:77–106, January 2005. http://dl.acm.org/citation.cfm?id=1412350.1412357
33. Speiser, S., Harth, A.: Taking the shape LIDS off data silos. In: Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS 2010, pp. 44:1–44:4. ACM, New York (2010). http://www.aifb.kit.edu/images/4/4a/Triplify-2010-ssp-aha-taking-the-lids-off-data-silos.pdf
34. Verborgh, R., Steiner, T., Van de Walle, R., Vallés, J.G.: The missing links - how the description format RESTdesc applies the linked data vision to connect hypermedia APIS. In: Proceedings of the First Linked workshop at the Ninth Extended Semantic Web Conference, May 2012. http://lapis2012.linkedservices.org/papers/3.pdf
35. Verborgh, R., Steiner, T., Van Deursen, D., Coppens, S., Vallés, J.G., Van de Walle, R.: Functional descriptions as the bridge between hypermedia APISand the Semantic Web. In: Proceedings of the Third International Workshop on restful Design. ACM, April 2012. http://www.ws-rest.org/2012/proc/a5-9-verborgh.pdf
36. Verborgh, R., Steiner, T., Van Deursen, D., De Roo, J., Van de Walle, R., Gabarro, J.: Capturing the functionality of Web services with functional descriptions. Multimedia Tools and Applications (2012). http://rd.springer.com/article/10.1007/s11042-012-1004-5
37. Verborgh, R., Van Deursen, D., Mannens, E., Poppe, C., Van de Walle, R.: Enabling context-aware multimedia annotation by a novel generic semantic problem-solving platform. Multimedia Tools and Applications (2012). http://rd.springer.com/article/10.1007/s11042-010-0709-6
38. Vrandečić, D., Krötzsch, M., Rudolph, S., Lösch, U.; Leveraging non-lexical knowledge for the Linked Open Data Web. In: 5th Review of April Fool's day Transactions, pp. 18–27 (2010). http://km.aifb.kit.edu/projects/numbers/linked_open_numbers.pdf

Hyperdata: Update APIs for RDF Data Sources (Vision Paper)

Jacek Kopecký([✉])

Knowledge Media Institute, The Open University,
Milton Keynes, UK
j.kopecky@open.ac.uk

Abstract. The Linked Data effort has been focusing on how to publish open data sets on the Web, and it has had great results. However, mechanisms for updating linked data sources have been neglected in research. We propose a structure for Linked Data resources in named graphs, connected through hyperlinks and self-described with light metadata, that is a natural match for using standard HTTP methods to implement application-specific (high-level) public update APIs.

1 Vision

A major function of Web APIs is to give users a way to contribute to data sources (whether they be social networks, photo sharing sites, or anything else) through rich scripted web sites, rather than through simple web forms, and also through external (even 3rd-party) tools. Facebook API, Flickr API and so on, support interactive Web interfaces as well as mobile apps or desktop tools.

Some of the data in these apps then gets published as Linked Data, a machine-friendly representation suitable for combining with other data. Commonly, there is a technologies disconnect, though, between the Linked Data read-only view on the data source (which employs RDF and URIs), and the update APIs (with JSON or XML, and non-URI identifiers).

In this paper, we describe a vision of *hyperdata*[1] — data that is not only hyperlinked and self-describing in terms of its schema, but also self-describing on how it can be updated.

As we've discussed in [1], update access cannot practically be provided through protocols such as SPARQL Update. Indeed, public update access should be through a data-source-specific application layer that enforces consistency and security. There are several reasons for this: (1) data dependencies, where an update needs to propagate into dependent data, (2) security, where low-level

This work has been funded by the European projects SOA4All (https://www.soa4all.eu) and PlanetData (https://www.planet-data.eu).

[1] The term "hyperdata", which predates the Web, has been used in connection with the Web of Data, for example in http://www.novaspivack.com/technology/the-semantic-web-collective-intelligence-and-hyperdata.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 285–290, 2015.
DOI: 10.1007/978-3-662-46641-4_21

access policies for RDF stores are harder to manage than if they were policies on the level of application-specific resources, (3) data constraints and consistency validation, and guiding the users in the structure of the accepted data (for example preventing well-meaning users from using the wrong ontology by mistake), and (4) creation of identifiers, because SPARQL Update does not (as yet) provide the equivalent of an `AUTO_INCREMENT` field in an SQL database, and leaving the creation of identifiers to clients is undesirable due to the potential for conflicts.

With self-describing read-write hyperdata, applications that consume Linked Data could easily add update functionalities, currently generally missing from mash-ups and other Linked-Data-based apps. Further, data browsers such as Tabulator, which currently supports SPARQL Update and WebDAV [2], would be able to provide edit/update capabilities over a wider range of resources.

Linked Data may be compared to Web 1.0: the latter was mostly read-only documents, and the former is mostly read-only RDF views on some databases. The Web of Data should be more like Web 2.0, with many sites allowing (and even relying on) contributions from their users. With hyperdata, that will be possible, because hyperdata is not only linked to other data, but also to its update APIs.

In our vision, the optimal update API should fit well with the structure of Linked Data (including the application of the principle of *following your nose* to discovering update capabilities), it should rely as much as possible on the methods of HTTP (adhering to REST's *uniform interface* constraint), and it should easily accommodate application-specific update authorization, validation and propagation logic.

2 Use Case Description

Our hyperdata approach was developed within a use case of the European research project SOA4All. The use case is an application called "Offers4All" that allows diverse companies to advertise offers to subscribers of the service (more detail in [1]). These offers might be "last-minute" travel deals, predefined campaign offers of restaurants, and so on. The Offers4All application allows an offer provider to create a new offer by describing what the offer is and who it is targeted at. An appropriate set of subscribers are then chosen and are made aware of the offer.

The application is backed by an RDF database that stores information about offer providers, their offers, and the users registered to receive the offers. Users can specify what offer categories they are interested in, and they can also choose to "like" some offers which allows social-networking-style recommendations to be used to increase the uptake of offers. Naturally, users can also specify various contact details, such as an email address and a mobile phone number.

For read and update access, the database is façaded by a custom API, whose functionalities can be seen as the following types of operations:

- listUsers() returns a list of the known users
- getUser(id) returns user data
- addUser(data) creates a new user record
- getUserInterests(id) returns the offer categories of interest to the user
- addUserInterest(id, uri) adds to the user's list of interests
- deleteUserInterest(interest-id) removes one from the user's list of interests
- deleteAllUserInterests(id) clears the list of interests
- *and so on for the various properties of the various objects in the database, incl. "likes" and contact information*

The granularity of these operations corresponds to the intended uses of the system: these are the types of operations that clients of such a database want to perform, and they are a good input for analyzing access control.

Following the principles of Linked Data and REST (useful even if the data is not published openly), the database is split into a number of resources: a single container *users resource*, multiple *user resources* (one per known user), container *user interests resources* (one per known user), and concrete *interest value resources* (one per a stated interest of a user), etc.

In a read-only data source, this fine level of granularity could be seen as too much, as retrieving all the data about a user does not present much overhead even if the client is only interested in the user's interests. However, with all these resources in place, update operations naturally map to HTTP methods.

Figure 1 shows the RDF graph of a user who likes two specific offers and has interest in one category. For brevity, the figure doesn't show the container resource for users. Along with the actual data triples, the figure also displays the self-description aspects, discussed in the next section.

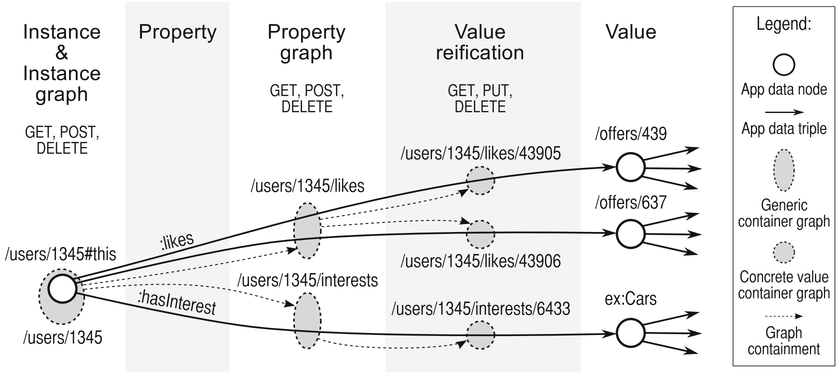


Fig. 1. Hyperdata structure of the API

```
1   </users/1345#this> a uc:User ;          uc: likes  </offers/439>, </offers/637> .
2
3   </users/1345> a g:Graph ;               g: defines </users/1345#this> ;
4     g: contains </users/1345/likes> ;     g: isContainedIn </users> .
5
6   </users/1345/likes> a g:Graph ;
7     g: contains </users/1345/likes/43905>, </users/1345/likes/43906> ;
8     g: defines [ a rdf:Statement ;
9       rdf: subject </users/1345#this> ; rdf: predicate uc: likes ; rdf: object [ ]
10    ] .
11
12  </users/1345/likes/43905> a g:Graph ;
13    g: defines [ a rdf:Statement ;
14      rdf: subject </users/1345#this> ; rdf: predicate uc: likes ; rdf: object </offers/439>
15    ] .
```

Listing 1. Example graph description triples (truncated)

3 Hyperdata Approach

The API in our use case consists of the following generic four types of resources: (1) containers of instances (users, offers etc.), (2) the instances themselves, (3) containers of property values, (4) concrete property values. Listing 1 illustrates the self-description metadata and hyperlinks, also shown in Fig. 1; it starts on line 1 with (a subset of) the actual data about the particular user.

Line 3 indicates the graph that is the description of the user instance, making it possible for a client to infer that an HTTP DELETE request can remove the instance. Line 4 links the instance graph with one of the property graphs (`/users/1345/likes`), and with the high-level class graph.

Lines 6–10 describe the property graph: it contains concrete value graphs, and a reified triple pattern (lines 8–10) that indicates that the graph includes statements of the form `/users/1345#this` uc:likes *something* (note the blank node as object). The triple pattern is meant to indicate what kind of data can be POSTed to the property resource to add a value, and what subset of the data about the user can be expected when GETting the property resource.

For adding a property value, the client can POST several kinds of data: an RDF graph, a list of URIs, or a literal value. Primarily, the POSTed data can be an RDF graph that contains a triple `/users/1345#this` uc:likes *something*, and any triples about the *something*. To prevent adding arbitrary statements, all the triples in the submitted data must be about the instance (the particular user) or about the values of other triples in the graph — we use the phrase that all the triples are "forward-reachable" from the instance. Further, the data must not contain any triples about other instances managed by this particular hyperdata store (for example about an offer) because submissions of such triples must go through that instance's update API.

Alternatively to submitting RDF data, there are media types that give the client a simpler way for submitting a new property value: if the client wants to add a property value that is some resource (e.g. `/offers/439#this`), the URI can be submitted as `text/uri-list` and it will be added as a direct value.

And finally, a new literal value (not appropriate for uc:likes but possible for other properties) can be submitted simply as `text/plain`.

The listing concludes on lines 12–15 with a description of a concrete value graph. The client can use PUT or DELETE here to update or remove a particular statement. PUT here has the same options for payload formats as POST for submitting new property values above — it can be an RDF graph, a URI list or a plain literal value.

The metadata uses a few very simple concepts to communicate much information: a Graph is a resource that besides GET may also accept update and delete requests (actually available methods can be discovered with HTTP OPTIONS).

The meaning of updates depends on the contents of the graph, described through reified statements. The reified statement may indicate a concrete triple like on line 14 (meaning that it represents a specific value, to be updated with PUT or removed with DELETE), or it may use blank nodes to indicate a collection (accepting POST with new items). The listing indicates a collection of property values for uc:likes on line 9. A reified statement of the form *something* rdf:type uc:User would be shown on the user container graph to indicate that it contains instances of the given ontology class, and that's what can be POSTed.

4 Prototype Implementation

We have developed a proof-of-concept triple-store wrapper (also described in [1]) that uses very simple configuration to realize the hyperdata API. Configured with a set of "classes of interest", whose instances the API manages, and "properties of interest" on those classes, the wrapper implements the necessary read and update resources to cover the structure of the hyperdata graph. The code generates and maintains all the self-description metadata as data is submitted and updated.

Currently, the metadata is stored in the underlying triple store along with the application data. If this overhead should become a performance or scalability issue, the metadata could be stored in a separate triple store (so that it does not affect reading and querying performance), or generated at runtime as the data is being accessed. On-the-fly generation of the metadata would decouple the data from the current configuration of the update API; however, it could itself present performance overhead. We have not evaluated performance and scalability issues in the scope of the use case.

The prototype does not address the issue of concurrent updates from multiple clients, beyond serializing the addition of instances; however, the HTTP specification defines the "entity tag" mechanism that supports conditional updates, performed only if the resource has not changed since the client has last seen it.

5 Conclusion

The Web included update capabilities from the start—the first browser[2] was also an editor—but still Web 1.0 was mostly read-only. A significant boom came with

[2] http://www.w3.org/People/Berners-Lee/WorldWideWeb.html.

the advent of the Web 2.0 with its attitude that anybody on the Web can—and should be allowed to—contribute. The Web of Data so far remains on the Web 1.0 level where contributions to it mostly happen outside it. Hyperdata APIs can bring update capabilities to the Web of Data, and make it more like Web 2.0. After all, Web 2.0 gave us Wikipedia, the heart of Linked Data.

Our prototype implementation is very basic, but the structure of the code and its configuration seems amenable to extensions towards access control policies, data validation and custom update propagation/processing (including logging and versioning). Also in future work, we would like to develop a client access library for hyperdata, and to extend Tabulator to support it. Client support will let us better evaluate the communication and client-side-processing overhead cost of all the metadata.

The hyperdata vision relies on the assumption that self-description of update capabilities can help clients adapt to changes in evolving hyperdata APIs, as hyperlinking allows clients to discover locations of new data, and to adapt to changed locations of expected data sources. This assumption needs to be evaluated on further case studies.

References

1. Kopecký, J., Pedrinaci, C., Duke, A.: RESTful write-oriented API for hyperdata in custom RDF knowledge bases. In: Proceedings of the International Conference on Next Generation Web Service Practices (NWeSP), Salamanca, Spain (2011)
2. Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., Prud'hommeaux, E., Schraefel, M.C.: Tabulator redux: browsing and writing linked data. In: Proceedings of the WWW 2008 Workshop on Linked Data on the Web, Beijing, China (2008)

Enabling Semantic Search in Large Open Source Communities

Gregor Leban(✉), Lorand Dali, and Inna Novalija

Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia
{gregor.leban, lorand.dali, inna.koval}@ijs.si

Abstract. This paper describes methodology used for building a domain specific ontology. Methods that allow automatic concept and relation extraction using domain-related glossaries are presented in this research. The constructed ontology contains vocabulary related to computer science and software development. It is used for supporting different functionalities in the ALERT project, which aims to improve software development process in large open source communities. One of the uses of the ontology is to provide semantic search functionality, which is a considerable improvement over the keyword search that is commonly supported.

1 Introduction

Open source communities and software development organizations in general are often using several different communication channels for exchanging information among developers and users. Beside a source code management system (SCMS), these software developing communities also frequently use an issue tracking system (ITS), a forum, one or more mailing lists and a wiki. Each of these channels typically serves a different purpose. Issue tracking system allows the users of the software to report to the developers issues they encountered or to suggest new features. Forums and mailing lists have a similar purpose which is to allow open discussions between the members of the community. Wikis are commonly used as platforms for providing software documentation, user guides and tutorials for the users.

A problem that is common for the present day and that is becoming more and more troubling for large and medium size open source communities is information overload. Users are generating large amounts of information on different communication channels and it is difficult to stay up-to-date. For illustration, consider the KDE community [1], which is offering a wide range of open source products. In April 2012 KDE had approximately 290.000 bug reports in their ITS, 126.000 posts on their forum and 163 active mailing lists where according to one of the KDE developers between 30–80 emails are exchanged per day.

Providing help in processing and managing such large amounts of information is one of the main goals of the ALERT[1] project [2]. ALERT is a European project that aims to develop a system, which will be able to help users (especially developers) in large open source communities. The system, once finished, will be able to collect and

[1] ALERT is acronym for Active support and reaL-time coordination based on Event pRocessing in open source software development.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 291–300, 2015.
DOI: 10.1007/978-3-662-46641-4_22

process all the posts (emails, issues, forum posts, etc.) that are generated in different communication channels used by the community. The information will be processed and stored in a way so that it will provide support for different features of the system.

The use case partners in the project are KDE, OW2 [3] and Morfeo [4]. At the beginning of the project they provided a set of most wanted features that should be supported by the ALERT system. One of the features that were identified as very important was advanced search functionality. Search that is supported on communication channels such as ITS and forums is only a simple Boolean keyword search. A fundamental problem with keyword search is that different people use different words (synonyms) to refer to the same concept. Therefore not all posts that discuss the same concept can be retrieved by a simple keyword-based search.

As an improvement to keyword search we would like to provide in the ALERT system *semantic* search. What we mean by semantic is that the search is performed using the actual concept that the search term represents. Consider, for example, that the user performs a search for term “dialog”. The concept of the dialog can be represented also with other terms, such as “window” or “form”. Instead of returning the results that directly mention “dialog” we therefore also want to return results that contain any of the term synonyms. Additionally, since search is based on the actual concepts we can also exploit the fact that concepts can be related to each other. When searching for one concept we can therefore also consider including results about some closely related concepts. In KDE domain, for example, searching for “email client” should also return posts containing concept “KMail” which is the KDE’s email application.

In order provide semantic search we have to use an ontology. Each class in the ontology should represent a concept that can have one or more labels (synonyms). When a new post is created in one of the communication channels, the ALERT system annotates or tags it with the concepts that are mentioned in the post. These annotations are stored in a knowledge base, which allows us then to quickly find all posts tagged with a particular concept.

The main question that needs answering is what ontology should be used for annotating the posts. Since we are providing support for software developing communities the important concepts that should be annotated are the ones related to computer science and software development. Since we were not able to find any such existing ontology we had to construct it ourselves. The process that we used to construct such an ontology is the main contribution of the paper. The steps in the process are general and can be reused also for constructing other domain specific ontologies.

The remainder of this paper is organized as follows. Section 2 provides details of the methodology used for building the Annotation ontology. In Sect. 3 we describe how the ontology can be used in ALERT to provide the semantic search functionality. Section 4 describes related work and Sect. 5 provides the conclusions.

2 Building the Annotation Ontology

In this section we describe the process that was used in order to construct the ontology used for annotating the posts. The process consists of two main steps – (a) identifying the computer science specific terminology that we wish to represent in the ontology, and (b) constructing the relations between the concepts.

2.1 Creating Ontology Concepts

As stated before, the concepts that we wish to have in the Annotation ontology are related to computer science and software development. In order to obtain a relevant set of terms we searched online for glossaries related to computer science. The two web sites that we found to be most up-to-date and relevant for our purpose were *webopedia.com* [5] and *whatis.techtarget.com* [6]. For each of the terms we were also able to obtain a description of the term which in most cases contained links to several related terms. To identify terms especially related to software development we used the *stackoverflow* website [7], which is a Q-A system with more than 2.5 million questions related to software development. *Stackoverflow* contains an up-to-date list of tags that are used to tag the questions. Most popular tags together with their descriptions were also included in the starting set of concepts.

After obtaining the set of terms, our first goal was to merge synonyms. Merging of terms was performed in two ways. First way was using a synonym list that we were able to obtain from the *stackoverflow* website and which contained around 1,400 synonym pairs. The second way was by using the term descriptions and searching in them for patterns such as “X also known as Y” or “X (abbreviated Y)”. In cases when such patterns were identified, terms X and Y can be considered as synonyms and be represented as the same concept. In this way we obtained a set of concepts where each concept has one or more labels that represent the concept.

In the next step we wanted to link the concepts to corresponding Wikipedia articles. This allows us to obtain more information about the concepts and potentially also extend the ontology with new related concepts. By using a semi-automatic approach, we make the repetition of the process relatively easy to do, such that future updates of the ontology are not too costly. By identifying a corresponding Wikipedia article we are also able to implicitly create links to well-known knowledge bases which are extracted from Wikipedia, such as DBpedia, Yago and Freebase.

Our approach for mapping concepts to Wikipedia articles has several steps. First, we link the concept labels to Wikipedia articles. We do this by automatically matching the labels to the titles of articles to see which article corresponds to each label. In this process, the following two challenges were identified:

a) The article with the matching title is a disambiguation page i.e. a page containing links to pages which each describe one of the meanings of the concept. For example *TTL* is mapped to a page which contains links to *Time to Live*, *Transistor Transistor Logic*, *Taiwan Tobacco and Liquor,* etc.
b) Some of the computer science concepts are so frequently used in common language that they are not considered ambiguous. In this case a computer science concept can be mapped on Wikipedia to something completely unrelated to computer science. An example of such a concept is *ant* which in computer science refers to *Apache ant*, a software tool for automatic build processes, but is mapped to the Wikipedia article about the ant insect.

The disambiguation pages were not difficult to identify since they typically contain phrases such as ‘*X may mean…*’ or ‘*X may refer to…*’, ‘*X is an abbreviation and can* *refer to…*'. We have defined rules to automatically match these patterns and exclude disambiguation pages from further analysis.

After mapping labels to the corresponding Wikipedia pages we used the content of these pages to identify new terms which were not covered by the glossaries. To do this, we only used the first paragraph of each article, which usually gives a short definition of the term. Often it also contains links to articles describing closely related concepts. We used the articles linked in the first paragraph as candidates for new terms and sorted them by their frequency. We expect that if an article was linked to by many articles that we know are about computer science, then this article is very likely about a computer science concept as well. Based on this assumption, titles of the frequently appearing articles were added to the ontology as new concepts.

After obtaining the final set of concepts we also wanted to organize the concepts into a hierarchy of categories. For this purpose we used text mining techniques and in particular the OntoGen [8] toolbox which interactively uses k-means clustering [9] to group the concepts into a hierarchy and extracts keywords to help the user in assigning a name to each category. In this way we were able to semi-automatically define 31 categories such as "operating systems", "programming languages" and "companies".

2.2 Creating Relations Between the Ontology Concepts

An important part of the ontology are also relations between the concepts. With regard to our task of semantic search, the relations allow us to expand the search to also include closely related concepts.

To create the relations between the concepts we can start by using the information that was available in the online glossaries. As we mentioned, the descriptions of the terms usually contained several links to other related terms. These links can be used to automatically create relations between the corresponding ontology concepts. Since hyperlinks don't contain any additional semantic information about the type of relation we can only use them to create some general kind of relation between the concepts. In our ontology we represented them using a *linksTo* relation.

In order to obtain more specific and usable relations we decided to apply natural language processing (NLP) techniques on term descriptions with the goal of identifying semantic relations between the concepts. Consider, for example, the following sentence from the "C#" concept description:

> "C# is a high level, general-purpose object-oriented programming language created by Microsoft."

If "C#" and "Microsoft" are concepts in the ontology then it is possible using NLP techniques to identify that the verb connecting the two concepts is "created by". The task of creating relations between concepts can then be reduced to defining a mapping from verbs to appropriate relations.

A detailed list of steps involved in creating the relations is as follows. The input to the procedure was the list of ontology concepts and all the descriptions of the concepts. First we identify in the descriptions sentences that mention two or more concepts. Next we use Stanford parser [10] to generate a dependency parse of the sentence.

The dependencies provide a representation of grammatical relations between words in a sentence. Using the dependency parse and the co-occurring ontology concepts, we can extract the path from one ontology concept to another one. As a next step, we used Stanford Part-Of-Speech (POS) tagger [11] to tag the words in the sentence. Of all the tags we are only interested in the verb (with or without preposition) that connects the two concepts. As a result we can obtain triples, such as:

XSLT, used by, XML schema
WSDL, describes, Web Service
Microsoft, created, Windows
Apple Inc., designed, Macintosh.

In the next step we use WordNet [12] and group the obtained verbs into synsets (synonym sets). From all the sentences we obtained verbs that can belong to around 750 different WordNet synsets. Of all these synsets we only considered those that can be mapped to relations *isPartOf, hasPart, creator* and *typeOf.* We chose to include these relations because they are mostly hierarchical and can be used to expand the search conditions. WordNet synsets that were used to obtain these relations were:

- *isPartOf* and *hasPart* relations were obtained from “include” and “receive have” synsets
- *creator* relations were obtained from “make create”, “form constitute make”, “implement”, “construct build make”, “produce bring forth”, “introduce present acquaint”, “make do” and “plan project contrive design” synsets
- *typeOf* relations were obtained from “establish base ground”, “include”, “exist be” and “integrate incorporate” synsets.

In addition to these relations we also included a few other types of relations:

- *subclass* and s*uperclass* relationships have been obtained by using the OntoGen [8] text mining tool
- *sameAs* relationships provide links to the identical Wikipedia and DBpedia resources.
- *linksTo* relations were used for all relations that we extracted from term descriptions but were not mapped to some more specific type of relation (like *isPartOf*, *creator*, etc.).

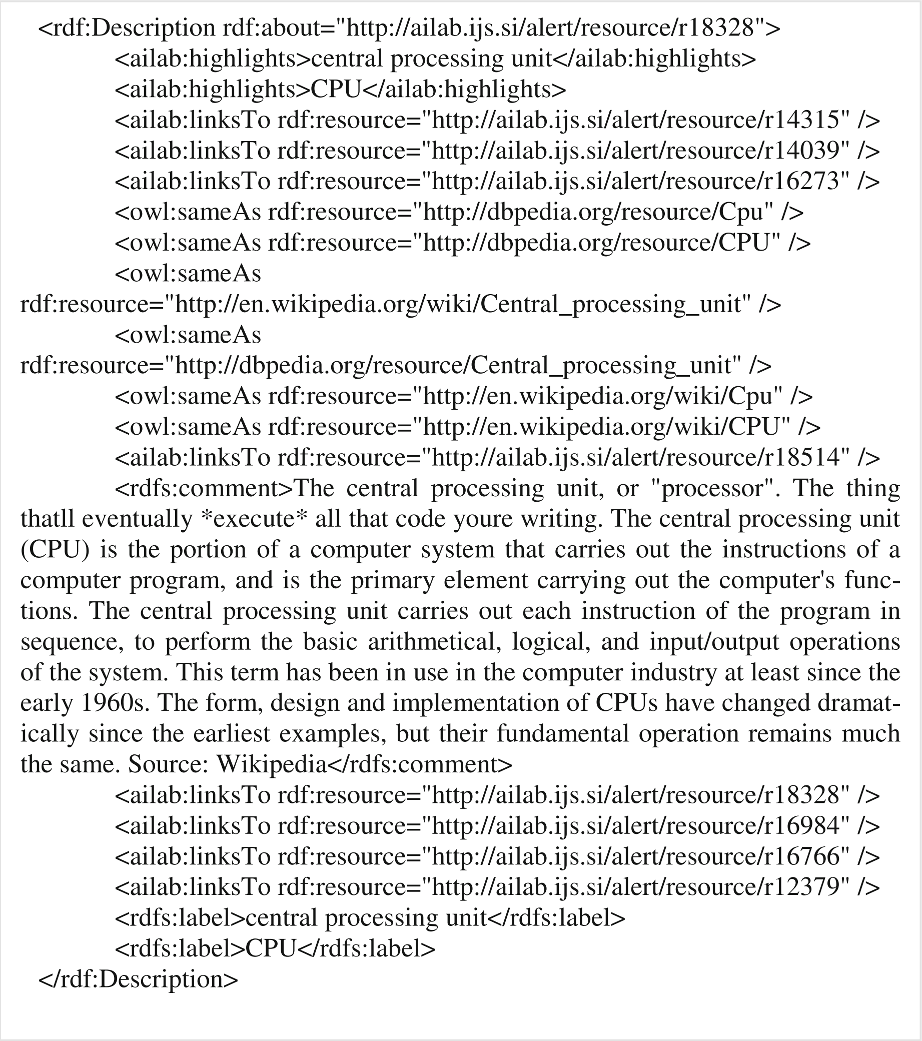
2.3 Filtering and Publishing the Ontology as RDF

Before the ontology was finished we wanted to make sure that it doesn’t contain any unnecessary concepts. By checking the terms on the online glossaries we noticed that some of them are obsolete and therefore irrelevant for our ontology. To determine if a concept is relevant or not we decided to again use the stackoverflow website. For each concept we searched in how many questions the concept is mentioned. If the concept was mentioned in less than 10 questions we decided to treat it as irrelevant and we removed the concept and its relations from the ontology. The value 10 was chosen experimentally by observing which concepts would be removed at different thresholds. An example of a concept that was removed by this procedure is HAL/S (High-order Assembly Language/ Shuttle) which was found only in one question on the *stackoverflow* website.

The final version of the generated ontology contains 6,196 concepts and 91,122 relationships and is published in the Resource Description Framework (RDF) format. In Fig. 1 we show an example of an ontology concept for the Central Processing Unit. It contains two labels, a description of the concept, links to equivalent Wikipedia and DBpedia resources and several relations to related concepts.

3 Using the Annotation Ontology for Semantic Search

The created ontology can be used to annotate all the posts that are generated in the communication channels monitored by the ALERT system. When a new post is created we annotate it with concepts that are mentioned in the text. We do this by checking the

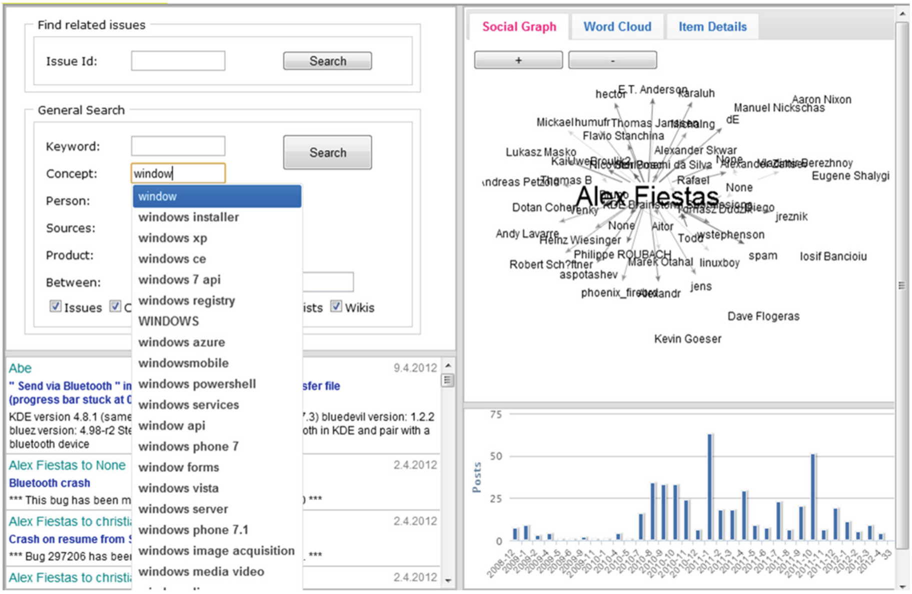


```
  <rdf:Description rdf:about="http://ailab.ijs.si/alert/resource/r18328">
          <ailab:highlights>central processing unit</ailab:highlights>
          <ailab:highlights>CPU</ailab:highlights>
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r14315" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r14039" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r16273" />
          <owl:sameAs rdf:resource="http://dbpedia.org/resource/Cpu" />
          <owl:sameAs rdf:resource="http://dbpedia.org/resource/CPU" />
          <owl:sameAs
rdf:resource="http://en.wikipedia.org/wiki/Central_processing_unit" />
          <owl:sameAs
rdf:resource="http://dbpedia.org/resource/Central_processing_unit" />
          <owl:sameAs rdf:resource="http://en.wikipedia.org/wiki/Cpu" />
          <owl:sameAs rdf:resource="http://en.wikipedia.org/wiki/CPU" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r18514" />
          <rdfs:comment>The central processing unit, or "processor". The thing
thatll eventually *execute* all that code youre writing. The central processing unit
(CPU) is the portion of a computer system that carries out the instructions of a
computer program, and is the primary element carrying out the computer's functions. The central processing unit carries out each instruction of the program in
sequence, to perform the basic arithmetical, logical, and input/output operations
of the system. This term has been in use in the computer industry at least since the
early 1960s. The form, design and implementation of CPUs have changed dramatically since the earliest examples, but their fundamental operation remains much
the same. Source: Wikipedia</rdfs:comment>
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r18328" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r16984" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r16766" />
          <ailab:linksTo rdf:resource="http://ailab.ijs.si/alert/resource/r12379" />
          <rdfs:label>central processing unit</rdfs:label>
          <rdfs:label>CPU</rdfs:label>
  </rdf:Description>
```

Fig. 1. Ontology concept for Central Processing Unit

labels of the concepts and determining if any of them appears in the text. The post with its annotations is then stored in the Knowledge base and can be used when performing the search.

Since the ALERT project is still in progress we currently only have a preliminary version of the search interface. A screenshot of the interface is shown in Fig. 2. The search form is located in the top left corner and allows the user to specify a rich set of search conditions. Beside the keyword search the conditions can also include:



Concept: window

Abe 9.4.2012
" Send via Bluetooth " in … sfer file (progress bar stuck at 0 …
KDE version 4.8.1 (same … .3) bluedevil version: 1.2.2 bluez version: 4.98-r2 Ste… oth in KDE and pair with a bluetooth device

Alex Fiestas to None 2.4.2012
Bluetooth crash
*** This bug has been m…) ***

Alex Fiestas to christi… 2.4.2012
Crash on resume from …
*** Bug 297206 has bee… ***

Alex Fiestas to christi… 2.4.2012

Fig. 2. Search interface provided by the ALERT system

- Concepts. The user can specify a concept from the Annotation ontology in order to find posts that are annotated with this concept.
- Authors of the posts. All posts from the communication channels have authors and they can be specified as a condition.
- Source code (files, classes, methods). By monitoring source code management systems used by the community we are aware of all the files, classes and methods developed in the project. Because the information is stored in the knowledge base we can easily find all the source code commits where a particular file/class/method was modified.
- Product. Issue tracking systems allow the issues to be assigned to particular products and components. When searching for issues we can limit the results only to issues that belong to a particular product or component.
- Time constraints. All posts have an associated time stamp that represents the time when the post was created. The search interface allows us to limit the search to a particular time period that we are interested in.

- Filtering by post type. Each post in the ALERT system is assigned a type based on the source where the post comes from (issues, emails, forum posts, etc.). The user can specify what type of posts he would like to see in the list of results.

After performing the search, the list of posts that match the query is displayed below the search form. Each item in the list contains the author of the post, the creation date, the subject and a short snippet of the post. Along with the list of results, the system also provides two visualizations of the results. Social graph of the people involved in the resulting posts is displayed on the right side of the screen. It shows who is corresponding with whom and highlights the most active people. Below the social graph is the timeline visualization that shows the distribution of results over time. It is an important aggregated view of the results since it can uncover interesting patterns. An example of such interesting pattern would be a spike in the number of submitted issues for a particular product. Such a pattern can be used to determine which source code commits should be analyzed to find the cause of the issues.

4 Related Work

The automatic and semi-automatic ontology learning methods usually include a number of phases. Most approaches define the set of the relevant ontology extension sources, preprocess the input material, build ontology according to the specified methodology, evaluate and reuse the composed ontology.

While developing ontologies, it is important to follow a number of ontology design criteria. Gruber [13] defines the following design criteria for ontology developing: Clarity, Coherence, Extendibility, Minimal encoding bias and Minimal ontological commitment. Uschold and King's method [14], Grüninger and Fox's methodology [15], METHONTOLOGY [16], On-To-Knowledge [17] represent the classic methodologies to ontology creation.

As Reinberger and Spyns [18] state, the following steps can be found in the majority of methods for ontology learning from text: collecting, selecting and preprocessing of an appropriate corpus, discovering sets of equivalent words and expressions, establishing concepts with the help of the domain experts, discovering sets of semantic relations and extending the sets of equivalent words and expressions, validating the relations and extended concept definition with help of the domain experts and creating a formal representation.

As suggested in [19], ontology learning from text is just one phase in the methodology for semi-automatic ontology construction preceded by domain understanding, data understanding and task definition and followed by ontology evaluation and ontology refinement.

In our approach we have utilized the traditional steps for ontology development, like terms extraction, synonyms extraction, concepts definition, establishment of concept hierarchies, relations identification [20].

Fortuna et al. [8] developed an approach to semi-automatic data-driven ontology construction focused on topic ontology. The approach combines machine learning and text mining techniques with an efficient user interface. The domain of interest is described by keywords or a document collection and used to guide the ontology construction. OntoGen [8] uses the vector-space model for document representation. In current work, the tool has been utilized for defining the hierarchical relationships between concepts.

Learning relations in the ontology was addressed by a number of researchers. Taxonomic relations have been extracted by Cimiano et al. [21]. Moreover, Cimiano et al. [22] suggested a method of learning concept hierarchies from text based on Formal Concept Analysis. Maedche and Staab [23] contributed to the approach, which allowed discovering conceptual relations from text.

In our work of building a domain specific ontology, we not only define ontology concepts, but also to specify the possible binary relations between the concepts using NLP techniques.

A number of researchers used ontologies for different tasks in software domain. Nallusamy et al. [23] describes how ontologies can be applied for software re-documentation. Rene Robin et al. [24] have designed an ontology suite for software risk planning, tracking and control.

5 Conclusions

In this paper we have proposed an approach for building a domain specific ontology related. The methods for concept and relation extraction have been suggested and applied in order to build an ontology related to computer science and software development. The generated ontology is used in the ALERT project among other things to provide semantic search functionality. The advantages of the semantic search over keyword search are (a) the avoidance of issues with synonyms, and (b) the ability for expanding the search by including related concepts in the search. The current version of the ALERT system provides a preliminary interface for performing the semantic search by entering the concept name. In future we plan to improve the interface to allow the user also to extend the search to related concepts.

Acknowledgements. This work was supported by the Slovenian Research Agency, European Social Fund and ALERT (ICT-2009.1.2).

References

1. KDE. http://www.kde.org
2. ALERT project. http://www.alert-project.eu
3. OW2. http://www.ow2.org
4. Morfeo project. http://www.morfeo-project.org
5. Webopedia. http://www.webopedia.com
6. Computer Glossary, Computer Terms. http://whatis.techtarget.com
7. Stack Overflow. http://www.stackoverflow.com
8. Gopal, D., Wang, Q., Gupta, G., Chitnis, S., Guo, H., Karshmer, A.I.: Winsight: Towards completely automatic backtranslation of nemeth code. In: Stephanidis, C. (ed.) HCI 2007. LNCS, vol. 4556, pp. 309–318. Springer, Heidelberg (2007)

9. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press
10. de Marneffe, M.-C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: LREC 2006 (2006)
11. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70 (2000)
12. WordNet. http://wordnet.princeton.edu
13. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum Comput Stud. **43**, 5–6 (1995)
14. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proceedings of Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95 Canada (1995)
15. Gruninger, M., Fox, M.: The role of competency in enterprise engineering. In: Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, IFIP (1994)
16. Corcho, Ó., Fernández-López, M., Gómez-Pérez, A., López-Cima, A.: Building legal ontologies with METHONTOLOGY and WebODE. In: Benjamins, V., Casanovas, P., Breuker, J., Gangemi, A. (eds.) Law and the Semantic Web. LNCS (LNAI), vol. 3369, pp. 142–157. Springer, Heidelberg (2005)
17. Sure, Y., Studer, R.: On-To-Knowledge Methodology - Final Version, On-To-Knowledge deliverable D-18, Institute AIFB. University of Karlsruhe (2002)
18. Reinberger, M.L., Spyns, P.: Unsupervised text mining for the learning of DOGMA-inspired ontologies. In: Buitelaar, P., Handschuh, S., Magnini, B. (eds.) Ontology Learning from Text: Methods. IOS Press, Evaluation and Applications (2005)
19. Grobelnik, M., Mladenic, D.: Knowledge discovery for ontology construction. In: Davies, J., Studer, R., Warren, P. (eds.) Semantic Web Technologies: Trends and Research in Ontology-Based Systems, pp. 9–27. Wiley, Chichester (2006)
20. Buitelaar, P., Cimiano, P., Magnini, B. (eds.): Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, Amsterdam (2005)
21. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. J. Artif. Intell. Res. **24**, 305–339 (2005)
22. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning taxonomic relations from heterogeneous evidence. In: Proceedings of ECAI 2004, Workshop on Ontology Learning and Population (2004)
23. Maedche, A., Staab, S.: Discovering conceptual relations from text. In: Horn, W. (ed.) ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, 21–25 August 2000, pp. 321–324. IOS Press, Amsterdam (2000)
24. Nallusamy, S., Ibrahim, S., Mahrin, M.N.: A software redocumentation process using ontology based approach in software maintenance. Int. J. Inf. Electron. Eng. **1**(2) (2011)
25. Rene Robin, C.R., Uma, G.V.: Design and development of ontology suite for software risk planning, software risk tracking and software risk control. J. Comput. Sci. **7**(3), 320–327 (2011)

An Approach for Efficiently Combining Real-Time and Past Events for Ubiquitous Business Processing

Laurent Pellegrino[1(✉)], Iyad Alshabani[1], Françoise Baude[1], Roland Stühmer[2], and Nenad Stojanovic[2]

[1] INRIA Sophia Antipolis Méditerranée, Valbonne, France
{laurent.pellegrino,iyad.alshabani,francoise.baude}@inria.fr
[2] FZI Forschungszentrum Informatik, Karlsruhe, Germany
{stuehmer,nstojano}@fzi.de

Abstract. Nowadays, datasets become larger and larger. As stated by Eric Schmidt, every two days now we create as much or more information as we did from the dawn of civilization up until 2003. Thus, the question is how to make good use of such big data. A solution is Complex Event Processing (CEP) engines that propose to correlate realtime, contextual and past information. In this paper we propose a new architecture that leverages existing research done in Publish/Subscribe systems and CEP engines with the idea to federate them in order to scale to the load that could be encountered in todays ubiquitous events workloads.

Keywords: Complex Event Processing · Publish/Subscribe · Resource Description Framework (RDF) · SPARQL protocol and RDF query language (SPARQL)

1 Introduction

Using Complex Event Processing this paper wants to outline a framework for dynamic and complex, event-driven interaction for the Web. Such an architecture will enable the exchange of contextual information (events) between heterogeneous services, providing the possibilities of optimizing and personalizing the execution of the services themselves, resulting in context-driven adaptivity. We will illustrate these goals by drawing a use case called "Contextual Latitude" in which we combine historical Twitter data with real-time location updates of mobile users.

The paper will begin by outlining the architecture used in our approach and implementation. It will then go into detail of explaining the components for real-time and historic events and propose an event format based on *Resource Description Framework* (RDF) with a matching SPARQL-based event pattern

This work was in part supported by the European Commission (Project PLAY, Grant 258659).

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 301–312, 2015.
DOI: 10.1007/978-3-662-46641-4_23

language syntax. After that the paper explains non-functional aspects of the architecture such as monitoring which is needed for contracts between event producers and consumers on the Web. The paper finishes by discussing related work and conclusions.

2 Proposed Solution

Nowadays, datasets become larger and larger. As stated[1] by Eric Schmidt, every two days now we create as much or more information as we did from the dawn of civilization up until 2003. Thus, the question is how to make good use of such big data. A solution is CEP engines that propose to correlate realtime, contextual and past information. The architecture we propose hereafter, leverage existing research done in pub/sub systems and CEP engines with the idea to federate them in order to scale to the load that could be encountered. The architecture we propose is depicted by Fig. 1.

The *EventCloud*s provide storage and event forwarding capabilities to interested subscribers. The role of an Event Cloud is a unified tool and corresponding API for manipulating realtime and historical events. Subscriptions may use a simple set of operators such as conjunctive queries to filter out an interesting event according to the numerous information it holds. It is also import to notice that each EventCloud may contain different kind of information and in a real usecase multiple EventCloud would be deployed either to differentiate data among organizations or to classify them. For example it is meaningless to put medical information with weather forecast data.

Complex queries are executed by the *Distributed Complex Event Processing* (DCEP) engine. As such, the *DCEP* component has the role of detecting complex events and reasoning over simple ones by means of event patterns defined in logic rules. To detect complex events, the DCEP subscribes to one or several EventClouds for any event defined in the original complex subscription and combines the results with historical ones thanks additional queries posted to EventClouds. DCEP supports traditional event operators such as sequence, concurrent conjunction, disjunction, negation, all operators from *Allen*'s interval algebra and windowing, filtering, enrichment, projection, translation, and multiplication operators. Out-of-order event processing is also supported (e.g., events that are delayed due to different circumstances such as network anomalies). More details about the expressivity and the features of both systems are given in Subsects. 3.1 and 3.2.

The *Query Dispatcher* serves as an entry point into the system to submit complex subscriptions. Its purpose is to decompose a complex subscription into pieces and to choreograph them to the DCEP and the EventCloud according to their expressiveness. In addition it may also pre-allocate EventClouds for outgoing complex events produced by the DCEP and it could ensure security aspects by contacting another component in charge of maintaining user authorizations (e.g., Is an Event Consumer X allowed to get information from EventCloud Y?).

[1] Eric Schmidt: http://techcrunch.com/2010/08/04/schmidt-data/.

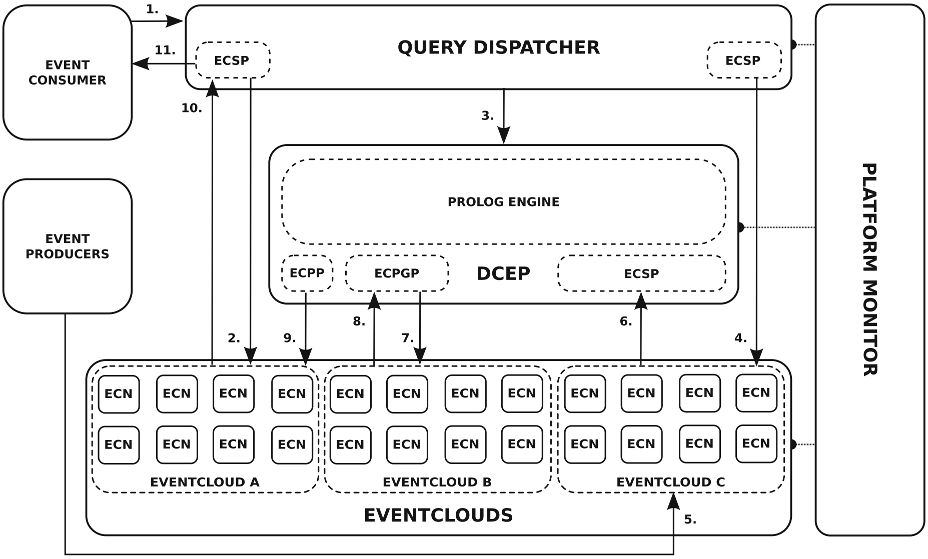


Fig. 1. Proposed architecture

Although this last point is not inside the scope of this paper, the architecture has been designed with it in mind.

A simple concrete scenario can better explain the interactions between all the components. Initially, an Event Consumer submits a complex subscription such as "notify me as soon as a friend is near me who tweeted in the last 2 weeks" (for the sake of conciseness we use english sentences). Once the pattern reaches the Query Dispatcher (interaction *1*) it is decomposed into pieces: a pattern and one or more subscriptions. These pieces are forwarded respectively to the DCEP and the EventClouds according to the expressivity they are supporting. One subscription is used to send back to the Event Consumer the complex events deduced by the DCEP as soon as they are published into the EventCloud *A* (interaction *2*). The other is a subscription sent on behalf of the DCEP to notify it about simple events that allow to resolve the original complex subscription (interaction *4*). Finally, the Query Dispatcher also conveys a pattern to the DCEP to inform which kind of events and how they have to be combined in order to deduce new complex events (interaction *3*).

One important point to notice here is that all communications with EventClouds go through proxies: *EventCloud Subscribe Proxy* (ECSP), *EventCloud Publish Proxy* (ECPP) and *EventCloud Putget Proxy* (ECPGP). Thanks to these proxies, the requests, the responses, the notifications, etc. can be monitored and handled according to QoS properties, see Subsect. 4.3 for Event Level Agreements.

Moments later, some Event Producers publish new events (interaction *5*). These events reach the EventCloud *C* where they are persisted. In addition, the events are forwarded to DCEP. At this point of time the DCEP can deduce the realtime part of its pattern but it still need to correlate this information with some historical one. Thus, it queries the EventCloud B for historical information about Tweets in the past (interactions *7* and *8*). In addition it could, if necessary, query several other EventClouds that contain knowledge information or previously published information. Querying historical information in the aftermath of a realtime notification imply several up and down communications between the DCEP and one or more EventCloud. This can be optimized by using the *Trigger* mechanism provided by the EventCloud described in Subsect. 3.2 below.

Now, the DCEP has all necessary data to deduce new complex events. These newly deduced events are pushed inside the EventCloud A (interaction *9*) and notified back to the final Event Consumer (interaction *10* and *11*).

3 Reasoning and Retrieval Model

3.1 DCEP Engine

At the API of any event-based system there is the *event format* and the query or *pattern language* to process events in meaningful ways. As the foundation for our event format we chose to use RDF for its extensibility, shared schemas and its semantics. These reseons are will be explained in more detail in this section below. As a corresponding pattern language we are building on the language from previous and ongoing work on our ETALIS system called *EP-SPARQL* which we extended to meet the needs of structured event models.

RDF Event Format. In a heterogeneous system such as the Internet, a common understanding of data is crucial. According to Rozsnyai et al. [12] this is especially true in a decoupled system such as an event-based system where the producer and consumer of an event might have no knowledge of each other. Therefore, a consumer must find a way to understand received events which entails the need for a universal event model [12].

Apart from the advantages of RDF as a data model, there is the advantage of having a lot of public data readily available in RDF that can be re-used[2]. This means that a lot of static data is available to be used as context of events. *Linked Data* is the methodology of publishing structured (static) data as RDF and to interlink the data to make it more useful. Examples of Linked Data are movies and their globally unique identifiers which can be found on-line. These identifiers are useful in identity management on the Web. We propose to apply this methodology[3] to streaming data and the principles apply just as well.

For our works, we are developing an RDF Streaming API to adapt the Linked Data principles to real-time applications. At the same time an event format is built using RDF so that our data modelling language fits seamlessly with the data dissemination.

[2] Data Sets: http://linkeddata.org/data-sets.

[3] Linked Data Principles: http://www.w3.org/DesignIssues/LinkedData.html.

```
@prefix :      <http://events.event-processing.org/types/> . @prefix
xsd:   <http://www.w3.org/2001/XMLSchema#> . @prefix telco:
<http://events.event-processing.org/uc/telco/> . @prefix geo:
<http://www.w3.org/2003/01/geo/wgs84_pos#> .

e:3363861111740 {
    e:3363861111740#event a :GeoLocation ;
    :endTime"1336554276709"^^xsd:dateTime ;
    :source <http://sources.event-processing.org/ids/Android#source> ;
    :stream <http://streams.event-processing.org/ids/GeoLocation#stream> ;
    :location [
        geo:lat"43.60843765"^^xsd:double ;
        geo:long"7.0582081500000005"^^xsd:double
    ] .
    telco:userType"Customer";
    telco:phoneNumber"33638xxxxxx";
}
```

Listing 1.1. Example of an RDF Event.

The example in Listing 1.1 shows several facts about our event schema: An event is **using quadruples** in TriG syntax. The graph name (a.k.a context) before the curly braces is used as a unique identifier per stream e.g., to enable efficient indexing of related contiguous triples in the storage backend for historic events. The event may **link to other events** in different streams which were used as input to create the event. These linked events could have further input events themselves. This allows modelling of *composite* events [9]. There is an **event ontology** from which type `GeoLocation` is inherited. This ontology can be extended by any user by referencing the RDF type `Event` as a super class. The ontology makes use of related work by reusing the class "Event" from Dolce Ultralight based on DOLCE [4]. The event links to a **stream** URI where current events can be obtained as they happen. We are reusing and creating domain vocabularies to subclass the class `Event`. For example in Listing 1.1 we use the W3C geo schema (i.e., `geo:lat` and `geo:long`) to add further well-structured data to the event. Moreover, to facilitate the creation of such data we are offering input adapters of which we already developed some for Pachube (sensor data), Facebook and Twitter and a generalized SDK to do the lifting (event transformation) to RDF.

In conclusion, we developed this event model to satisfy requirements of an open platform where data from the Web can be reused and which is extensible and can be used in reasoning and semantic search. Future updates to the event schema can be tracked on-line[4].

Event Processsing SPARQL 2.0. EP-SPARQL [2] is a language executed in our event processing engine ETALIS. The purpose of EP-SPARQL is matching the requirements of RDF events which are discussed in Subsect. 3.1. For this paper we are extending EP-SPARQL to EP-SPARQL 2.0 to deal with structured events consisting of many RDF triples as opposed to one triple per event. This greatly increases the ease-of-use of our implementation in environments where events have many attributes to be processed atomically as opposed to a single triple per event.

[4] Linked Data Streaming: http://km.aifb.kit.edu/sites/lodstream/.

EP-SPARQL 2.0 works with structured events which can combine many properties, whereas the previous design only looked at one triple at a time. An event is thus now a graph whereas it used to be a triple before in our earlier designs. This is much closer to real world systems which must understand events with multiple facets.

As part of allowing for multiple attributes, the timestamps in each event are now made explicit (as another event property) whereas they were merely second-class citizens in our previous work being added in the background to each tuple of subject-predicate-object. This allows us now to transmit the timestamp as part of the event RDF graph to support application time (where a source specifies the occurrence of an event). Additionally, we added more built-in functions to our language such as all functions like `fn:contains` from XPATH.

Unlike similar aims pursued in [10] we are supporting all of ETALIS' operators which explicitly include event-at-a-time operators and not only set-at-time operators. The latter can be explained as being only well-defined on windows (sets of events) whereas the former can select events individually.

Listing 1.2 shows an example query in EP-SPARQL. The purpose of the query is to find two people who are now close to a location where someone tweeted before.

```
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX
uctelco: <http://events.event-processing.org/uc/telco/> PREFIX geo:
<http://www.w3.org/2003/01/geo/wgs84_pos#> PREFIX :
<http://events.event-processing.org/types/>

CONSTRUCT {
    :e rdf:type :ContextualizedLatitudeEvent .
    :e :stream <http://streams.event-processing.org/ids/ContextualizedLatitudeFeed#
         stream>.
    :e :location [ geo:lat ?Latitude1; geo:long ?Longitude1 ] .
    :e uctelco:phoneNumber ?bob .
    :e uctelco:phoneNumber ?alice .
    :e :message"Alice and Bob are close to a where someone tweeted.".
}
WHERE {
    GRAPH ?id1 {
        ?e1 rdf:type :TwitterEvent .
        ?e1 :endTime ?time .
        ?e1 :stream <http://streams.event-processing.org/ids/TwitterFeed#stream> .
        ?e1 :location [ geo:lat ?Latitude1; geo:long ?Longitude1 ] .
    }
    FILTER ?time > (NOW() - xsd:duration("P14D"))
    WINDOW {
        EVENT ?id2 {
            ?e2 rdf:type :GeoLocation .
            ?e2 :stream <http://streams.event-processing.org/ids/GeoLocation#stream> .
            ?e2 :location [ geo:lat ?Latitude2; geo:long ?Longitude2 ] .
            ?e2 uctelco:phoneNumber ?alice .
            }
            FILTER fn:abs(?Latitude1 - ?Latitude2) < 0.1 && fn:abs(?Longitude1 - ?
                 Longitude2) < 0.5
        SEQ
        EVENT ?id3 {
            ?e3 rdf:type :GeoLocation .
            ?e3 :stream <http://streams.event-processing.org/ids/GeoLocation#stream> .
            ?e3 :location [ geo:lat ?Latitude3; geo:long ?Longitude3 ] .
            ?e3 uctelco:phoneNumber ?bob .
            }
            FILTER fn:abs(?Latitude2 - ?Latitude3) < 0.1 && fn:abs(?Longitude2 - ?
                 Longitude3) < 0.5
                && ?alice != ?bob
    } ("PT20M"^^xsd:duration, sliding)
}
```

Listing 1.2. Example EP-SPARQL Query.

This example demonstrates several facts about our query language: We modelled EP-SPARQL as close to SPARQL 1.1 [6] as possible. Exceptions are being made for necessary event operators and the denotations of events compared to non-event historic data. Historic data is processed by EventCloud. Event derivation (the modelling of the resulting complex event) is handled through a CONSTRUCT clause as according to SPARQL 1.1. The clause contains a template of triples for the complex event. Additional attributes like the time stamps are handled implicitly because they are derived automatically from the participating simple events which form the complex event according to the semantics of the event operators. The pattern in the WHERE clause contains several events combined with event operators (e.g., SEQ, cf. [2]) nested in a sliding time window defined by the `xsd:duration`. Each event is matched separately according to its content like a GRAPH clause does in SPARQL 1.1. The clause, however, is denoted with EVENT here to distinguish the real-time parts of the query from the optional historic parts. The latter works like traditional queries which are evaluated after or while the complex event is detected. We extended our underlying engine ETALIS to execute XPATH functions such as `fn:contains()` to be more expressive with regard to handling XML data types.

This concludes the description of EP-SPARQL 2.0, our revised design for continuously querying structured, rich events in RDF.

3.2 EventCloud

The EventCloud is a distributed datastore that allows to store quadruples (RDF triples with context) and to manage events represented as quadruples or set of quadruples (a.k.a., compound event). Compound events are essential because the number of elements contained by an event (quadruple) is limited. A compound event is a coarse grain event that is made of a non-limited number of quadruples. In addition, supposing that a quadruple is modeled by a 4-tuple $q = (c, s, p, o)$ and a compound event by a set $C = \{q_0, q_1, ..., q_n\}$ then each q of C shares the same context value c in order to have the opportunity to identify the quadruples that form this compound event.

To scale, the architecture is based on a structured Peer-to-Peer (P2P) network named Content Addressable Network (CAN) [11]. A CAN is a structured P2P network (structured in opposition to unstructured, another category of P2P networks better suited to high peer churn) based on a d-dimensional Cartesian coordinate space labeled $\mathcal{D}$. This space is dynamically partitioned among all peers in the system such that each node is responsible for indexing and storing data in a *zone* of $\mathcal{D}$ thanks to a traditional RDF datastore such as Jena [3]. According to our data model, we use a 4-dimensional CAN in order to associate each RDF term of a quadruple to a dimension of the CAN network. Thus, a quadruple to index is a point in a 4-dimensional space.

Retrieval Model. At the EventCloud level, an API is provided according to a retrieval model based on *pull* and *push* mechanisms. The *pull* or *put/get* mode refers to one-time queries; an application formulates a query to retrieve data which have been already stored. In contrast, the *push* or *pub/sub* mode is used to notify applications which register long standing queries and push back a notification each time an event that matches them occurs.

Both retrieval modes have their filter model based on SPARQL that is usually used to retrieve and manipulate data stored in RDF format with one time queries. This language is suitable to build a very expressive filter model. Even if it could be used as a pull retrieval model, for the push retrieval model we introduced some restrictions: we only allow SELECT query form and a pattern applies to one graph value at a time. As such SPARQL provides us the ability to formulate a subscription by associating several filter constraints to a quadruple (event), but also to a set of quadruples that belong to the same compound event. This means that several events of a same compound event that are published asynchronously at different times may participate to the matching of a subscription by using their common constraints. Also, due to the distributed nature of the EventCloud, each quadruple is possibly stored different peers.

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2
3 SELECT ?user ?name ?age WHERE {
4     GRAPH ?g {
5         ?user foaf:name ?name .
6         ?user foaf:age ?age
7     } FILTER (?age >= 18 && ?age <= 25)
8 }
```

Listing 1.3. Legal SPARQL subscription indicating that only events about users whose age is between 18 and 25 have to be notified.

Listing 1.3 shows an example of a subscription that is used to deliver a notification each time two events that belong to the same compound event (represented by the graph pattern and its associated variable `?g`) match the constraints. This subscription depicts two types of constraints that may be uttered. The first one is a join constraint (lines *5* and *6*). It consists in computing an equi-join condition on the variable `?user` with the events of the same graph that match the triple patterns (a triple that may contain variables for querying unknown values). The second type of constraints that may be formulated are filter constraints. Filter constraints are shown in the example by using the `FILTER` keyword on line *7*. This second type of constraint may contain several logically related predicates.

It is important to understand that our push retrieval mode is not supposed to act as a CEP engine correlating several compound events from several streams. Join queries apply on a single compound event with the purpose to deliver it to the CEP only if it is meaningful for the complex subscription.

Triggers. As introduced in Sect. 2, a client of an EventCloud such as the DCEP may have, depending of the complex subscription to handle, a need to execute one or more historical queries after having received notifications for simple subscriptions. When this scenario occurs, several round-trips between the DCEP and the EventCloud(s) are necessary to dispatch each historical query and to receive their results independently whereas only the last result is of importance.

Notice that as simple subscriptions are extracted from the original complex subscriptions, historical queries to perform in the future are known when they enter the system. The EventCloud is designed and optimized for such a scenario. It provides a trigger mechanism to register a simple realtime subscription along with a sequence of historical queries to perform. The historical queries can be parameterized with variables associated to the realtime subscription. Then, when the subscription is verified and before to execute the next historical query, the variables are resolved by using the values that have matched the subscription.

Thus, if there are S simple subscriptions to register and H historical queries to execute for each S, without triggers it implies to fire $S \times H$ round-trips between the DCEP and an EventCloud to retrieve the final result. With triggers only S round-trips are necessary to retrieve the same information.

4 Monitoring

Monitoring is essential in a platform such as the one we propose in Sect. 2. Firstly, to ensure that component managers are able to detect failures and conditions that may slow down the execution of a particular component. Secondly, to allow components to self-adapt according to the load they encounter, but also because Event Level Agreements (ELA) may be established between the platform and the users or between some components of the platform. In this way, to detect the violation and to ensure in some cases the property, a monitoring component dedicated to the platform must be able to gather information from other components and provide a high level view of what happens.

4.1 Communications

To allow to build a dynamic architecture, communications between components participating in the monitoring process are based on an Event Driven Architecture (EDA), as for the EventCloud. The different monitoring actors (Query Dispatcher, DCEP, EventClouds and Platform Monitor) can subscribe and publish asynchronously monitoring data in a dynamic way based on the needs. Regarding monitoring reports, they are based on OASIS Web Service Notification (WSN) for interoperability with existing tools.

4.2 Sensors

Monitoring data has to be collected in the runtime system and provided to interested parties by dedicated software components. We call these components sensors by analogy to hardware sensors. The Query Dispatcher, DCEP and the EventClouds embed soft “sensors” at each entry and exit point. Regarding an EventCloud, it means that each proxy but also each peer contains a sensor which is able to report activity about events handled (received, sent, stored, etc.).

In addition, each node contained by the EventCloud embeds a monitoring component dedicated to measure node activity (CPU consumption, memory, incoming/outgoing requests rate and more generally the load). Thanks to these information, the EventClouds are able to detect overloaded nodes and to react consequently by adapting themself.

These sensors are implemented by using GCM/ProActive; a distributed software component model. It allows to plug easily specific components for sensing monitoring information to functional components representing EventCloud nodes.

4.3 Event Level Agreement (ELA)

Service Level agreements have been introduced in Service Oriented Architecture as a contract between service providers and service clients. Here, we define agreements in the same manner for Event Driven Architecture. In this context, events are the core of the interactions between producers and consumers. As producers and consumers are decoupled, the contract could be defined either between producers and the platform or between consumers and the platform.

For example, a consumer may formulate its interest in some kind of events through a complex subscription deployed into the Query Dispatcher. Also, this consumer could require to receive at most X events per second. This is expressed through an ELA contract attached to the subscription.

Each component of the platform is involved in the process of a complex subscription reports. Once it receives a subscription from the Platform Monitor, it starts sending information about incoming and outgoing events (when it has been received, who is the provider, etc.). Thanks to these reports, the Platform Monitor has the ability to correlate these information to verify if the ELA contract is still valid at this point of time. Then, when the agreement is violated it is possible to further analyze the reason and to react by replaying a time windows. Indeed, an EventCloud provides storage capabilities and one or more can be dedicated to store monitoring information in addition to business events.

5 Related Work

There are already some approaches experimenting how to store and query RDF data, including with the aim to face a large volume of data and multiple concurrent (synchronous only) queries. For instance, BigOwlim[5] follows a master slave approach to query in parallel the RDF store, but the master remains a central bottleneck. Also, some contemplate the use of Cloud approaches, combined with NO-SQL database technologies. CumulusRDF [7] proposed to rely upon the Cassandra [8] key-value store, by leveraging its two levels indexing model in order to store and retrieve RDF triples, again in a only synchronous mode. The choices they make in CumulusRDF are driven by the need to retrieve RDF data by triple patterns only and not the full expressivity of SPARQL.

As for vocabulary for events, in Event Processing some attempts were made. A notable approach from research is the XML format of AMIT presented in [1].

[5] BigOwlim Replication: http://www.ontotext.com/owlim/replication-cluster.

It goes beyond the previous approaches, for example, by providing more detailed temporal semantics and by modelling not only events but generalization, specialization and other relationships between events which can be used in processing which we inherit in our current works. RDF-based event formats exist. Such schemas include E* [5] and others, all of which also rely on the DOLCE [4] top-level ontology as do we. However, they do not seem to be tailored to real-time processing of events because a lot of (e.g., temporal) expressivity such as relative and vague time is not supported by the state of the art in real-time processing engines. Our event format combines a large part of the expressivity and flexibility of the aforementioned formats with the execution model of our processing engine ETALIS.

6 Conclusion

In conclusion this paper outlines necessary technologies to realise a Web of streaming information. This includes an event format which is based on well-known notions of time and place, and an accompanying query language to filter and process these events. Using semantic technologies for these artefacts is a step in the direction of a grand challenge for the real-time Web which should be decentralized, global, Internet-like and built upon widely-accepted open standards.

References

1. Adi, A., Botzer, D., Etzion, O.: Semantic event model and its implication on situation detection. In: ECIS. Wirtschaftsuniversität Wien (WU) (2000)
2. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: Ep-sparql: A unified language for event processing and stream reasoning. In: WWW 2011: Proceedings of the Twentieth International World Wide Web Conference (2011)
3. Carroll, J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: World Wide Web Conference, pp. 74–83. ACM (2004)
4. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 166–181. Springer, Heidelberg (2002). http://dl.acm.org/citation.cfm?id=645362.650863
5. Gupta, A., Jain, R.: Managing Event Information: Modeling, Retrieval, and Applications. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, San. Rafael (2011)
6. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language, October 2010. http://www.w3.org/TR/sparql11-query/
7. Ladwig, G., Harth, A.: Cumulusrdf: Linked data management on nested key-value stores. In: Proceedings of the 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011) at the 10th International Semantic Web Conference (ISWC 2011), October 2011
8. Lakshman, A., Malik, P.: Cassandra: a structured storage system on a p2p network. In: Proceedings of the Twenty-First Annual Symposium on Parallelism in Algorithms and Architectures, SPAA 2009, p. 47. ACM, New York (2009). http://doi.acm.org/10.1145/1583991.1584009

9. Luckham, D.C., Schulte, R.: Event processing glossary - version 2.0. Online Resource (2011). http://www.complexevents.com/2011/08/23/event-processing-glossary-version-2-0/. Last visited: September 2011
10. Perry, M., Jain, P., Sheth, A.P.: Sparql-st: Extending sparql to support spatiotemporal queries. In: Ashish, N., Sheth, A.P. (eds.) Geospatial Semantics and the Semantic Web. Semantic Web and Beyond, vol. 12, pp. 61–86. Springer, US (2011). http://dx.doi.org/10.1007/978-1-4419-9446-2_3
11. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In: ACM SIGCOMM Computer Communication Review, vol. 31(4), pp. 161–172 (2001)
12. Rozsnyai, S., Schiefer, J., Schatten, A.: Concepts and models for typing events for event-based systems. In: DEBS 2007: Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems, pp. 62–70. ACM, New York (2007)

Context Management in Event Marketplaces

Yiannis Verginadis(✉), Ioannis Patiniotakis, Nikos Papageorgiou,
Dimitris Apostolou, and Gregoris Mentzas

Institute of Communications and Computer Systems,
National Technical University of Athens, Athens, Greece
{jverg,ipatini,npapag,dapost,gmentzas}@mail.ntua.gr

Abstract. This paper refers to methods and tools for enabling context detection and management based on events. We propose a context model that builds on top of previous efforts and we give details about the mechanisms developed for context detection in event marketplaces. In addition, we show how simple or complex events can be used in combination with external services in order to derive higher level context with the use of Situation-Action-Networks (SANs). Specifically, we present two different approaches, one for detecting low level context and another one for deriving higher-level contextual information using SANs. We present an illustrative scenario for demonstrating the process of specialization of our generic context model and its instantiation based on real-time events.

Keywords: Context · Event marketplace · Detecting context · Deriving context

1 Introduction

Context is "any information that can be used to characterize the situation of an entity, i.e., a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [1]. Context detection is considered important in the so-called event marketplaces [2] (see e.g., https://Xively.com, a platform offering a service based architecture, a range of graphing and visualization tools, event detection via triggers, along with cost-effective data storage) for enhancing the user's experience when interacting with the event marketplace.

Events from event marketplaces are an important source of context for service-based applications that consume them because they may convey important information, which is relevant for service execution and adaptation. To achieve the goal of injecting event processing results to context, an event-based context model is needed along with context detection and derivation mechanisms. In previous work [3] Situation-Action-Networks have been proposed as a hierarchical goal-directed modeling approach comprising nodes with specific semantics used to model goal decompositions, enriched with flow control capabilities. SANs provide means to decompose goals into subgoals and capabilities for seeking and achieving the high-level goals, involving situations (i.e. complex event patterns), context conditions and actions.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 313–326, 2015.
DOI: 10.1007/978-3-662-46641-4_24

The simplest SAN possible is a two level tree with a parent (root) node and three child nodes, each of them having specific semantics. A parent node models the Goal sought. The leftmost child node describes a situation that must occur, in order to start goal seeking. The middle child node corresponds to context update and requires that a specific contextual condition is true before continuing with the SAN traversal. The rightmost child node specifies the action to be taken in order to fulfil the goal. Rightmost node can also be a sub-goal node with its own three child nodes, or it can even be a construct joining several sub-goals in sequence or in parallel. As the SAN becomes more complex, involving several subgoals (Fig. 1), it deepens and reveals its hierarchical and goal-directed characteristics. In this work, SANs are extended so that they can be used for detecting and deriving context from events.

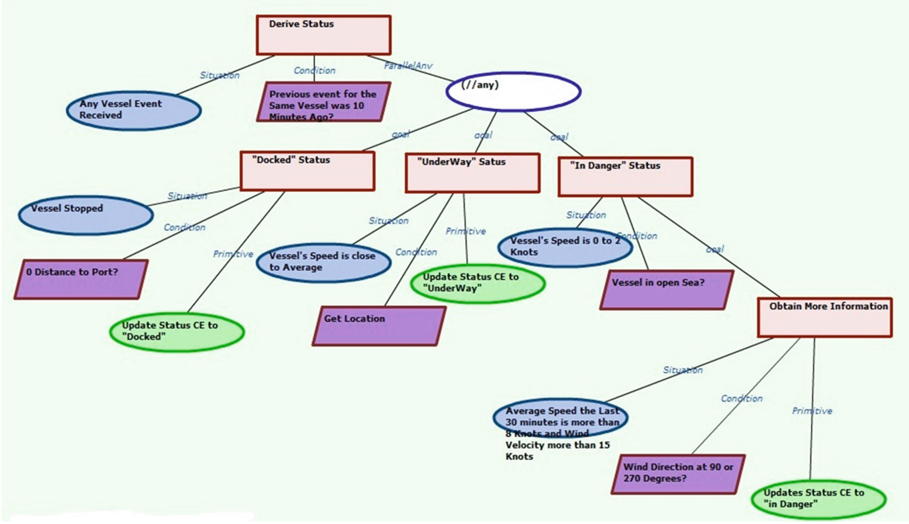

Derive Status
Situation
Condition
ParallelAnv
Any Vessel Event Received
Previous event for the Same Vessel was 10 Minutes Ago?
(//any)
goal
goal
goal
"Docked" Status
"UnderWay" Satus
"In Danger" Status
Vessel Stopped
Situation
Condition
Primitive
0 Distance to Port?
Update Status CE to "Docked"
Situation
Vessel's Speed is close to Average
Condition
Primitive
Get Location
Update Status CE to "UnderWay"
Situation
Vessel's Speed is 0 to 2 Knots
Condition
goal
Vessel in open Sea?
Obtain More Information
Situation
Condition
Primitive
Average Speed the Last 30 minutes is more than 8 Knots and Wind Velocity more than 15 Knots
Wind Direction at 90 or 270 Degrees?
Updates Status CE to "in Danger"

Fig. 1. SAN Illustration based on the marine vessel traffic scenario: Pink nodes denote Goals, Blue nodes denote Situations, Magenta nodes denote Context Condition and Green nodes denote Actions (Color figure online)

This paper continues with a discussion about related work in the domain of event-based context management, while in Sect. 3 it presents a generic context model that is considered appropriate for the needs of event marketplaces. In Sect. 4, we consider two different approaches, one for detecting low level context and another one for deriving higher-level contextual information using SANs. In Sect. 5, we show how the generic context model can be specialized so that it can be instantiated to support an example scenario. We conclude in Sect. 6 with a summary of our event-based context management approach.

2 Related Work

Context-awareness in service-oriented systems refers to the capability of a service or service-based application to be aware of its physical environment or situation and to respond proactively and intelligently based on such awareness; see e.g. [4]. Through the use of context, a new generation of service-based applications is expected to arise for the benefit of coping with the dynamic nature of the Internet; see e.g. [5, 6]. The multiplicity of applications and the surge of research activities in context-aware service systems are also evident in recent survey research; see e.g. [7–11] point out that in the case of service-based applications context has various different facets as it includes information ranging from the situation in which users exploit a service-based application to the conditions under which the component services can be exploited. Gartner analysts consider Context Delivery Architecture (CoDA) as the next step in the evolution of Service-Oriented Architecture; see [12–14]. In CoDA the functioning of software elements (services or event handlers) is determined not only by the input to the element, but also by the secondary sources of information – the context; two invocations of the same service with the same parameters may yield different results in different circumstances, i.e. within different contexts.

To reflect the varying nature of context and to ensure a universal applicability of context-aware systems, context is typically represented at different levels of abstraction [15]. At the first level of raw context sources there are context data coming from sensor devices, or user applications. At the next levels, context is represented using abstraction approaches of varying complexity. The work in [16] reviews models of context that range from key-value models, to mark-up schemes, graphical models, object-oriented models, logic-based models and ontology-based models. In [17] an ontological model of the W4H classification for context was proposed. The W4H ontology provides a set of general classes, properties, and relations exploiting the five semantic dimensions: identity (who), location (where), time (when), activity (what) and device profiles (how). The five dimensions of context have been also pointed out earlier in [18] where it was stated that context should include the 'five W': Who, What, Where, When, and Why. For example, by 'Who', they mean that it is not enough to identify a person as a customer; the person's past actions and service related background should also be identified for better service provision. 'What' refers to the activities conducted by the people involved in the context and interactions between them. 'Where' represents location data. 'When' is related to time. 'Why' specifies the reason for 'Who' did 'What'. 'Why' represents a complicated notion and acts as the driving force for context sensitive information systems.

An important aspect in the design of context-aware applications concerns modelling languages, which take context explicitly into account. The first such effort was ContextUML a UML-based modeling language which was specifically designed for context-aware Web service development and applies model-driven development principles; see [19]. In a Web-service-based environment ContextUML considers that context contains any information that can be used by a Web service to adjust its execution and output. Examples of contexts in ContextUML are: (i) contexts related to a service requester (mostly it is the client who invokes a service), including the requester's identification information, personal preferences, current situation (e.g., location), and other information (e.g., friends list, calendar); (ii) contexts related to a Web service, such as service location, service status (e.g., available, busy), and its QoS attributes (e.g., price, reliability); and (iii) other contexts like time and weather information. ContextUML has been adopted for the development of a model-driven platform, called ContextServ, which is used to develop context-aware Web applications; see [6]. There exist modelling efforts that attempt to treat service logic and context handling as separate concerns: the first is the work in [20], where they modified ContextUML using Aspect-Oriented Programming (AOP) principles the second is the work in [21] that leveraged ideas from model driven development (MDD) and AOP in order to define a conceptual context model and then mapped these ideas to a UML framework.

A problem that arises in context detection is related to imperfect observations (e.g., sensor readings) that lead to the estimation of the current user situation. Many researchers (see [22–24]) suggested filtering or repairing problematic contexts (e.g., inaccurate, incomplete or noisy contexts). For example, to deal with this shortcoming, Anagnostoloulos et al. in [25] propose the use of Fuzzy Logic theory with the purpose of determining (inferring) and reasoning about the current situation of the involved user. In this approach, captured, imperfect contextual information is matched against pre-developed situation ontologies in order to approximately infer the current user context. In [24] a hybrid approach is proposed in order to detect problematic contexts and resolve resulting context inconsistencies with the help of context-aware application semantics.

Our work focuses on detecting context changes which correspond to either atomic or complex events and use complex event processing to model and identify them. Similarly to [26], we focus on events as a source of context because they are snippets of the past activities; therefore event processing may be viewed as a context detecting technology. Event processing results may be transferred to other applications, injecting context related information into services and processes. Based on the context definition of Dey and Abowd [1] and the associated five dimensions of context expressed in ontological model of the W4H [17], we define a high-level context model following an object-based modelling approach which can be easily specialized for different applications. We use semantic querying to extract contextual information from event payloads. Moreover, we exploit the reasoning capabilities of Situation- Action- Networks to enable dynamic derivation of context from multiple event streams and external services.

3 Context Model

We propose a context model as a stepping stone for facilitating event-based context detection and derivation functionality, in order to better understand situations in dynamic service oriented environments that demand for new additional information sources or/and lead to a number of service adaptations as means for successfully coping with dynamic environmental changes. In order to achieve the goal of extracting contextual information, analyzing them and then deriving higher level context, we follow an event-based context modelling approach. In this section, we present such a Context Model (Fig. 2), expressed in UML 2.0 class diagram. This model is based on the W4H model [17] that describes the five main elements associated within a context; the five elements are arranged into a quintuple (When, What, Where, Who, How).

This Context Model expresses the temporal (i.e. When), spatial (i.e. Where), declarative (i.e. Who, What) and explanatory (i.e. How) dimensions of context having as central point of focus the notion of Entity. We refer to either physical or virtual entities with specific profiles and preferences that characterise them (e.g. vessel, port authority information system etc.). This way context obtains substance around the notion of an

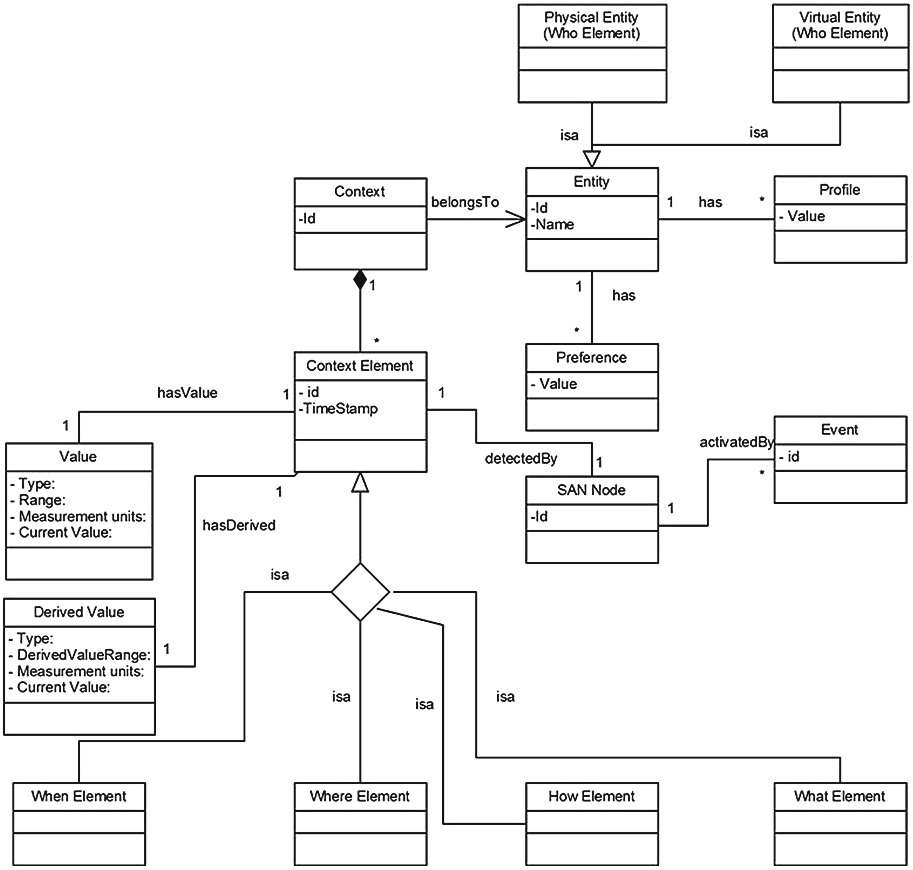


Fig. 2. Context Model

entity which can be a customer of an event marketplace system. The context class in our model constitutes the aggregation of several different context elements that may refer to five dimensions of context. Each Context element can have a value that can be acquired from the situation node of a SAN and/or a derived value that arises from any kind of reasoning process or call of external services. All context related information can be captured as objects which can store either a single scalar value or multiple values such as vectors, sets, lists etc. As any of the available context models [8], our model needs to become domain or application specific in order to be useful. Next, we show how SAN Editor can be used to specialize and instantiate the generic context model.

4 Event-Based Context Management

In this section we discuss our approach for both detecting and deriving contextual information based on events. Specifically, the context detection refers to a mechanism for querying events for updating contextual information while the context derivation involves the acquisition of higher level context compared to the lower level information that events carry. Both approaches use SANs expressed in a RDF-based language that is presented in the second subsection.

4.1 Detecting and Deriving Context from Events

In our context modelling approach and implementation, we consider entities as being able to own SAN trees. The scope of context elements is distinguished into three levels:

- "Local": Context elements can be updated and used only by a specific SAN instance, e.g., wind velocity value of a specific area can be updated only by the sensors of a specific vessel's event stream that is handled by a specific SAN.
- "Entity": Context elements can be updated and used by any of the SANs owned by the same entity, e.g., wind velocity value of a specific area can be updated by the sensors of different vessels' event streams that are handled by more than one SANs owned by a Port Authority Information System (i.e. entity that owns SANs).
- "Global": Context elements can be updated and used by all SANs independently to which entity they belong to, e.g., wind velocity value of a specific area can be updated by the sensors of different vessels' event streams but also from a National Meteorological Service (i.e. different entities).

Using the SAN Editor, i.e. a dedicated graphical tool for designing SANs, we can perform context model specializations based on the application scenario and can formulate the necessary queries to events for extracting contextual information. Using this editor, SANs are visualised in two different tree-like graphical representations and can be exported in an executable format, i.e. SAN language (presented below). We provide two approaches for acquiring context from simple or complex events and instantiating our context model. Both approaches use the SAN Editor for:

1. defining SPARQL queries to specific RDF event payload information that can update the values of an entity's context elements; and
2. defining SANs that can use information from several event streams, analyse them and/or combine them with external services, in order to update the derived values of context elements. In this way, we succeed in acquiring higher level context compared to the lower level information that events carry.

Regardless the approach used for detecting context, all contextual information is stored in a dedicated Context Repository. We provide an API for the use of different types of repositories. We have tested our implementation using a relational database and an in-memory (implemented in Java) context-repository for the purpose of our marine related experiments.

The application of both approaches is presented in the following Sect. 5 through the marine vessel traffic illustrative scenario, which uses events related to marine traffic control that can be used to detect potentially dangerous vessel movements informing a controller when two vessels are approaching each other.

4.2 SAN RDF-Based Language

SAN definitions, generated by SAN editor, are stored in files using an RDF-based language. Currently the RDF/N3 format is supported by SAN engine but more RDF formats can be added in the future.

A set of RDF classes have been defined and have special meaning to SAN engine. These classes describe the SAN node types, such as Goals, Situations, Actions, or describe their properties, for instance name, subsequent nodes or actions, complex event patterns (CEPAT) expressions. These SAN definitions must have the following outline:

- Prefix specifications– Prefixes are used as short names of namespace URIs in order to define the domain (and origin) of the various RDF elements
- one or more SAN entity specification blocks
  - one entity specification, referencing the SANs (Root Goals) contained in that particular entity
  - any number of SANs specification blocks
    - Root Goal specification block
    - Situation specification block
      - Defines interesting/critical Situations as Complex Event Patterns (CEPATS). CEPATS are expressed using EP-SPARQL [27]
    - Context Condition specification block
    - Action/Subgoal specification block
      - Primitive Action specification block
      - Complex Action specification block - Sequence Action, Selector Action, Parallel Any/All/Timeout specification blocks
      - Subgoal specification block - Same as Root Goal (without auto-start or other root specific features)

In the following Table 1 we present a part (due to page limitations) of the SAN language specification expressed in Backus Normal Form (BNF). This formal language description specifies the main concepts of SAN language, i.e. Root Goal, Goal, Situation, Context Condition and Action. The Decorators carry behavioural specifications on how the subordinate actions or goals should be executed.

In Table 2, we present a SAN language example using a simplified excerpt of the marine vessel related SAN. In this example the entity “Port Authority” owns a SAN that undertakes the task of monitoring the safe sailing of a high speed vessel.

5 Illustrative Scenario

A vast amount of real time events are available from portals connected to automatic identification systems (AIS) that contain important vessel information worldwide (e.g., speed, course, vessel type, wind conditions etc.) and the several different users/authorities that might be interested in them. In order to exploit efficiently all these information in an automated way we use our context model and present how it can be specialized for the specific application domain while we give a glimpse to its possible

Table 1. SAN language specifications in BNF.

```
# SAN SPECIFICATIONS
<NODE_NAME> ::= ";" "san:name" <STRING>
<ROOT GOAL> ::= <URI> "a" "san:RootGoal"
              <NODE_NAME>
              <GOAL_DETAILS>
<GOAL> ::= <URI> "a" "san:Goal"
              <NODE_NAME>
              <GOAL_DETAILS>
<GOAL_DETAILS> ::= ";" "san:hasSituation" <URI>
                        ";" "san:hasContextCondition" <URI>
                        ";" "san:hasAction" <URI>
                        "."
<SITUATION> ::= <URI> "a" "san:Situation"
                    <NODE_NAME>
                    ";" "san:dialect" " 'EP-SPARQL' "
                    ";" "san:defined-by" <EP-SPARQL-QUERY>
                    "."
<CONTEXT_CONDITION> ::= <URI> "a" "san:Situation"
                              <NODE_NAME>
                              ";" "san:dialect" <STRING>
                              ";" "san:defined-by" <STRING>
                              "."

# ACTION SPECIFICATIONS
<ACTION> ::= <PRIMITIVE_ACTION> | <ABSTRACT_ACTION> | <CALCULATION_ACTION> |
<MOUNT_ACTION> | <PARALLEL_ANY_ACTION> | <PARALLEL_ALL_ACTION> |
<SEQUENCE_ACTION> | <SELECTOR_ACTION>

# DECORATOR SPECIFICATIONS
<DECORATOR> ::= <LOOP_DECORATOR> | <COUNTER_DECORATOR> |
<TIMER_DECORATOR> | <SUCCESS_DECORATOR> | <FAILURE_DECORATOR> |
<PRINT_DECORATOR> | <BREAK_DECORATOR>
```

run time instantiations. For the purposes of our illustrative scenario we have used the AIS Hub portal (http://www.aishub.net/) that shares such vessel tracking information. These real-time data are fed into our system as RDF events, through an appropriate adapter that we have developed.

Context Model Specialization: Our context model needs to become application specific in order to be useful. We focus on context model specialisation which pertains the definition of entities along with their context elements necessary for capturing the context in terms of a specific application scenario. We use the marine vessel traffic scenario which is related to vessel and marine traffic control observing systems.

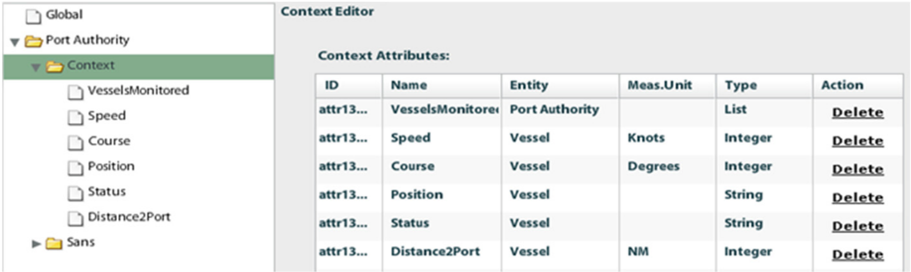
In this scenario, we consider the entity Port Authority as the owner of all SANs discussed below while the entity of interest is the Vessel. In order to capture contextual information related to Vessels' context, we have defined the following Context Elements (using SANs) that shape the specialization of our context model: Speed, Course, Position, Status, Distance2Port. In Fig. 3, a screenshot of the SAN Editor is presented that depicts this context model specialization.

Table 2. SAN language example.

Definition	Example
Entity	:_Port_Authority a san:SANEntity ; san:hasRootGoal :_Highspeed_Vessel_Safe_Sailing ; san:auto-start "yes"^^xsd:string ; san:name "Port Authority"^^xsd:string .
Root Goal	: _Highspeed_Vessel_Safe_Sailing a san:RootGoal ; san:hasSituation : _Vessel_Moved ; san:hasContextCondition :_Check_vessel_speed ; san:hasAction :_Warn_Vessel ; san:auto-start "yes"^^xsd:string ; san:name "Highspeed Vessel Safe Sailing"^^xsd:string .
Situation	:_Vessel_Moved a san:Situation ; san:dialect "default"^^xsd:string ; san:defined-by " http://streams.event-processing.org/ids/ VesselStream s "^^xsd:string ; san:hasContextualizer : _Vessel_Moved_Contextualizer; san:name "A ‘Vessel Moved’ Event received"^^xsd:string . :_Vessel_Moved_Contextualizer a san:Contextualizer ; san:type "RDF"^^xsd:string ; san:contextualizer-query :_Vessel_Moved_Cntxlzr_Qry1 . : _Vessel_Moved_Cntxlzr_Qry1 a san:ContextualizerQuery ; san:language "SPARQL"^^xsd:string ; san:context "LOCAL"^^xsd:string ; san:dialect "default"^^xsd:string ; san:defined-by " SELECT ?vessel_id ?name ?speed WHERE { ?uri :MMSI ?vessel_id . ?uri :Name ?name . ?uri :Speed ?speed } "^^xsd:string .

Context Condition	:_Check_vessel_speed a san:ContextCondition ;
	san:dialect "JS"^^xsd:string ;
	san:defined-by "/* JS code */
	// code executed right away.
	var id = ctx.getItem('vessel_id')[0];
	var speed = ctx.getItem('speed')[0];
	// a function definition (not-executed)
	function check(speed) {
	// check if speed over 50 knots
	return (speed > 50); }
	// calling a function
	check(); // the result of the last statement is also
	// the result of the Context Cond. node.
	// i.e. true or false in this case.
	"^^xsd:string ;
	san:name "Check Vessel Speed"^^xsd:string .

Primitive Action	: _Warn_Vessel a san:PrimitiveAction ;
	san:command "EXPRESSION"^^xsd:string ;
	san:dialect "JS"^^xsd:string ;
	san:defined-by "/* JS code */
	var name = ctx.getItem('name')[0];
	// write a message to console
	out.println('Contact vessel ‘+name+
	’ and ask to slow down');
	"^^xsd:string ;
	san:name "Warn vessel"^^xsd:string .



Global
Port Authority
Context
VesselsMonitored
Speed
Course
Position
Status
Distance2Port
Sans

Context Editor

Context Attributes:

ID	Name	Entity	Meas.Unit	Type	Action
attr13...	VesselsMonitore	Port Authority		List	Delete
attr13...	Speed	Vessel	Knots	Integer	Delete
attr13...	Course	Vessel	Degrees	Integer	Delete
attr13...	Position	Vessel		String	Delete
attr13...	Status	Vessel		String	Delete
attr13...	Distance2Port	Vessel	NM	Integer	Delete

Fig. 3. Context Model Specialisation using SAN Editor

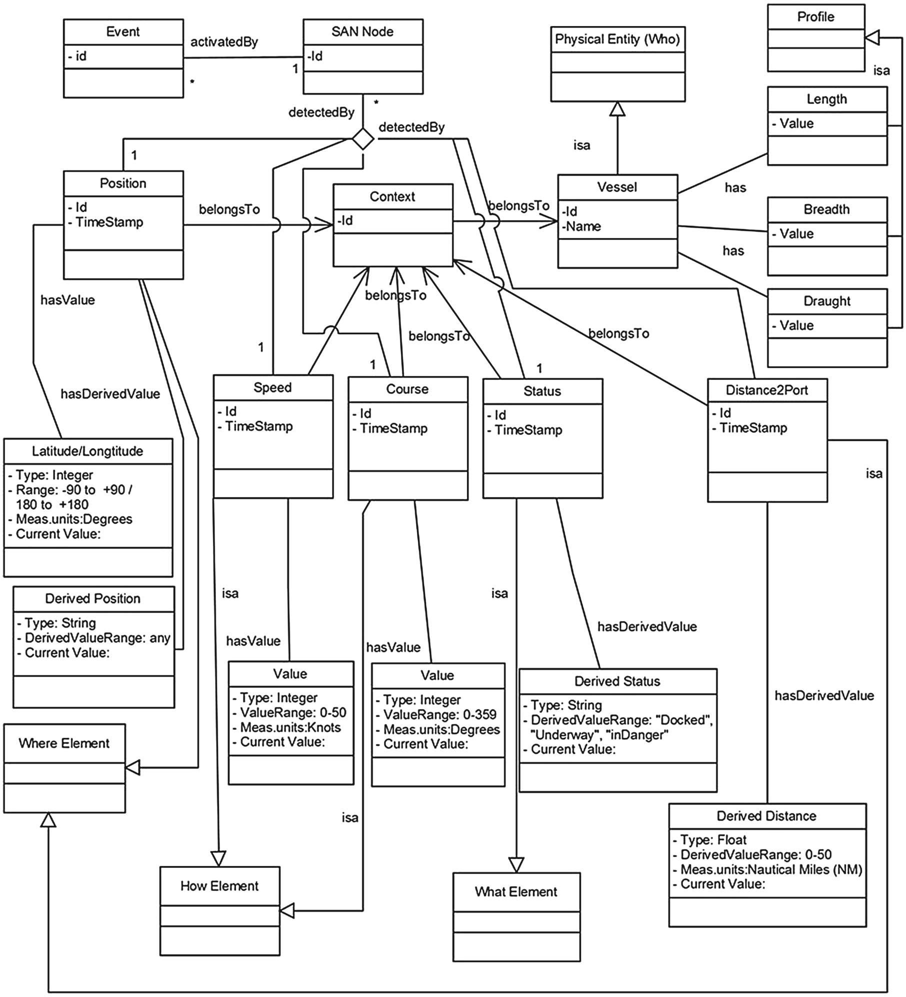
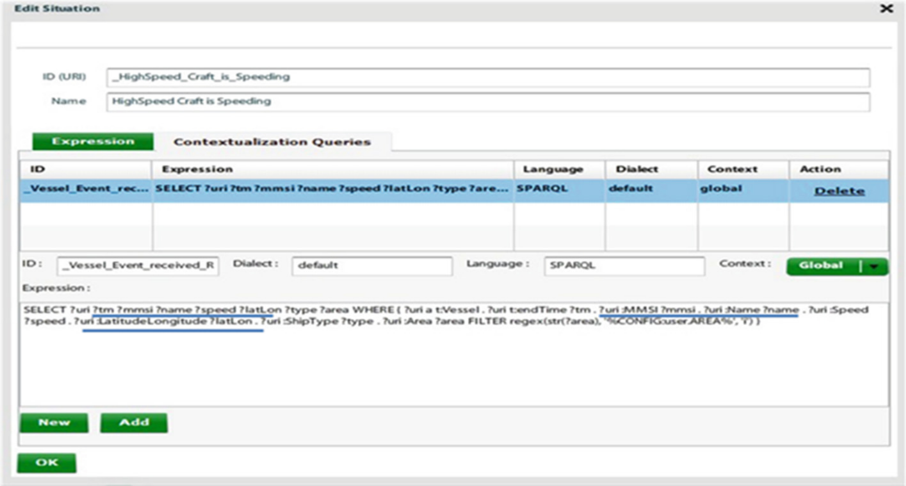


Fig. 4. Context Model Specialisation for the marine vessel traffic scenario



ID (URI) _HighSpeed_Craft_is_Speeding

Name HighSpeed Craft is Speeding

ID	Expression	Language	Dialect	Context	Action
_Vessel_Event_rec...	SELECT ?uri ?tm ?mmsi ?name ?speed ?latLon ?type ?are...	SPARQL	default	global	Delete

ID: _Vessel_Event_received_R Dialect: default Language: SPARQL Context: Global

Expression:

```
SELECT ?uri ?tm ?mmsi ?name ?speed ?latLon ?type ?area WHERE { ?uri a t:Vessel . ?uri t:endTime ?tm . ?uri :MMSI ?mmsi . ?uri :Name ?name . ?uri :Speed
?speed . ?uri :LatitudeLongitude ?latLon . ?uri :ShipType ?type . ?uri :Area ?area FILTER regex(str(?area), '%CONFIG:user.AREA%', 'i') }
```

Fig. 5 SAN Editor screenshot – performing SPARQL queries (about position)

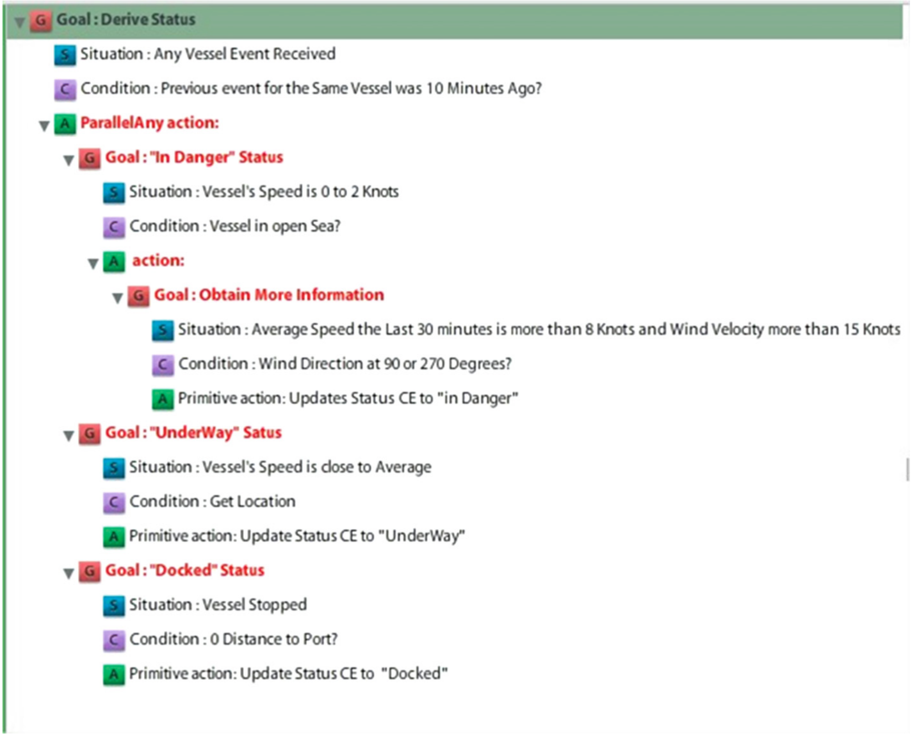
In Fig. 4, the reader can find the complete list of the five context elements associated with the Vessel entity, specialising the context model for the marine vessel traffic scenario. This model specialisation will be instantiated at run time through the context detection and derivation approaches that are presented below.

Detecting Context: In this section, we discuss our first approach for acquiring context from simple or complex events and instantiating our context model. Using SAN Editor, we are able to define SPARQL queries to specific event payload information that update the values of an entity's context elements. During our experiment we received events regarding a specific vessel called "Risoluto". Details regarding the entity such as profile information automatically update the context of this entity based on the detected events in the situation node of a SAN. Figure 5 depicts a screenshot of SAN editor with the required SPARQL queries for instantiating the "Position" context element of the vessel entity (Latitude/Longitude). Specifically, we query the vessel entity event payload with respect to the "LatLon" information. Similarly, other queries are used in the editor regarding the "Speed" and "Course" context elements and refer to event-based detection of low level context.

Deriving Context using SANs: this refers to our second approach that we apply for extracting context from simple or complex events and instantiating our context model using SANs. We define a number of SANs that can use information from several event streams and combine them with external services in order to update the derived value class of context elements. In this way, we succeed in acquiring higher level context compared to the lower level information that events carry.

This context derivation can be complex and may involve multi-level SANs. Figures 1 and 6 show a SAN that upon traversal will be able to update the derived value class of the Status context element. Specifically, the status of the vessel becomes "Docked" whenever we detect a vessel that has been stopped and its distance from any port is close to zero or "UnderWay" whenever vessel's speed is close to average and

"In Danger" when the system realizes that the vessel has almost stopped (away from any port) and strong winds are blowing from the side. The run-time execution of the specific SAN led to the derivation of a number of vessels' statuses. A pop up alert has been added in order to better demonstrate the context derivation regarding the Status context element for each vessel. Such contextual information can provide the basis for intelligent services in today's event marketplaces that will provide context-aware value added functionalities (e.g. dynamic event subscriptions, service/workflow adaptation).



- G Goal : Derive Status
  - S Situation : Any Vessel Event Received
  - C Condition : Previous event for the Same Vessel was 10 Minutes Ago?
  - A ParallelAny action:
    - G Goal : "In Danger" Status
      - S Situation : Vessel's Speed is 0 to 2 Knots
      - C Condition : Vessel in open Sea?
      - A action:
        - G Goal : Obtain More Information
          - S Situation : Average Speed the Last 30 minutes is more than 8 Knots and Wind Velocity more than 15 Knots
          - C Condition : Wind Direction at 90 or 270 Degrees?
          - A Primitive action: Updates Status CE to "in Danger"
    - G Goal : "UnderWay" Satus
      - S Situation : Vessel's Speed is close to Average
      - C Condition : Get Location
      - A Primitive action: Update Status CE to "UnderWay"
    - G Goal : "Docked" Status
      - S Situation : Vessel Stopped
      - C Condition : 0 Distance to Port?
      - A Primitive action: Update Status CE to "Docked"

Fig. 6. SAN for the marine vessel traffic scenario

6 Conclusions

In this paper we presented methods and tools for enhancing context detection and management based on events. This proposed context management approach presented here is considered appropriate for the needs of event marketplaces. We described a Context Model that was used by the developed mechanisms for performing event-based context detection and presented two different approaches for detecting low level context (using SAN Editor) and deriving higher-level contextual information using Situation-Action-Networks (SANs). We provided with a meaningful context model specialization and demonstrated how simple or complex events coming from an event marketplace can be used and combined with external services, in order to derive higher level context with the use of SANs. For this demonstration we used real-time data from the AIS Hub portal that were published as RDF events into our system, using the appropriate event adapter.

This proposed context management approach presented here is considered appropriate for the needs of value added services that can be developed in order to provide intelligent and cost efficient characteristics in today's event marketplaces. Such services can include dynamic event subscriptions recommenders that propose the appropriate times for subscribing and unsubscribing to heterogeneous event sources [28] or service adaptation frameworks that can detect and implement meaningful adaptations to business processes [3] based on situations and context-related information.

Acknowledgment. This work has been partly funded by the European Commission within the FP7 projects PLAY (258659) and REFLEX (63756). The authors would like to thank the project partners for their advices and comments regarding this work.

References

1. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: Proceedings of the PrCHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, pp. 304–307 (2000)
2. Stühmer, S., Stojanovic, N.: Large-scale, situation-driven and quality-aware event marketplace: the concept, challenges and opportunities. In: Proceedings of the 5th ACM International Conference on Distributed Event-Based System, NY, USA, pp. 403–404 (2011)
3. Verginadis, Y., Patiniotakis, I., Papageorgiou, N., Stuehmer, R.: Service adaptation recommender in the event marketplace: conceptual view. In: Garcia-Castro, R., Fensel, D., Antoniou, G. (eds.) ESWC 2011. LNCS, vol. 7117, pp. 194–201. Springer, Heidelberg (2012)
4. Abowd, G.D., Ebling, M., Hunt, G., Lei, H., Gellersen, H.W.: Context-aware computing. IEEE Pervasive Comput. **1**(3), 22–23 (2002)
5. Sheng, Q.Z., Nambiar, U., Sheth, A.P., Srivastava, B., Maamar, Z., Elnaffar, S.: WS3: international workshop on context-enabled source and service selection, integration and adaptation. In: Proceedings of the International Workshop on Context Enabled Source and Service Selection, Integration and Adaptation, Beijing, China, pp. 1263–1264 (2008)
6. Sheng, Q.Z., Yu, J., Dustdar, S.: Enabling Context-Aware Web Services: Methods, Architectures, and Technologies, 1st edn. Chapman and Hall/CRC, Boca Raton (2010)
7. Villegas, N.M., Müller, H.A.: Managing dynamic context to optimize smart interactions and services. In: Chignell, M., Cordy, J., Ng, J., Yesha, Y. (eds.) The Smart Internet. LNCS, vol. 6400, pp. 289–318. Springer, Heidelberg (2010)
8. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. Int. J. Web Inform. Syst. **5**(1), 5–31 (2009)
9. Hong, J., Suh, E., Kim, S.J.: Context-aware systems: a literature review and classification. Expert Syst. Appl. **36**(4), 8509–8522 (2009)
10. Kapitsaki, G.M., Prezerakos, G.N., Tselikas, N.D., Venieris, I.S.: Context-aware service engineering: a survey. J. Syst. Softw. **82**(8), 1285–1297 (2009)
11. Bucchiarone, A., Cappiello, C., di Nitto, E., Kazhamiakin, R., Mazza, V., Pistore, M.: A context-driven adaptation process for service-based applications. In: Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, pp. 50–56 (2010)

12. Natis, Y.V., Clark, W., Valdes, R.: Context delivery architecture: putting SOA in context. Gartner Research, ID Number: G00152306 (2007)
13. Clark, W., Lapkin, A.: Fundamentals of context delivery architecture: introduction and definitions. Gartner Research ID Number: G00161876 (2008)
14. Clark, W.: Fundamentals of context delivery architecture: provisioning context-enriched services. Gartner Research, ID Number: G00200649 (2010)
15. Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S.: Situational reasoning for task-oriented mobile service recommendation. Knowl. Eng. Rev. **23**(01), 7–19 (2008)
16. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput. **6**(2), 161–180 (2010)
17. Truong, H.L., Manzoor, A., Dustdar, S.: On modeling, collecting and utilizing context information for disaster responses in pervasive environments. In: Proceedings of the 1st International Workshop on Context-Aware Software Technology and Applications, pp. 25–28 (2009)
18. Abowd, G.D., Mynatt, E.D.: Charting past, present, and future research in ubiquitous computing. ACM Trans. Comput. Hum. Interact. **7**(1), 29–58 (2000)
19. Sheng, Q., Benatallah, B.: ContextUML: a UML-based modeling language for model-driven development of context-aware web services development. In: Proceedings of the International Conference on Mobile Business (ICMB 2005), Sydney, Australia, pp. 206–212 (2005)
20. Prezerakos, G.N., Tselikas, N.D., Cortese, G.: Model-driven composition of context-aware web services using ContextUML and aspects. In: Proceedings of the IEEE International Conference on Web Services, (ICWS), pp. 320–329 (2007)
21. Grassi, V., Sindico, A.: Towards model driven design of service-based context-aware applications. In: International Workshop on Engineering of Software Services for Pervasive Environments: in Conjunction with the 6th ESEC/FSE Joint Meeting, pp. 69–74 (2007)
22. Bu, Y., Gu, T., Tao, X., Li, J., Chen, S., Lu, J.: Managing quality of context in pervasive computing. In: QSIC 2006, pp. 193–200. IEEE Computer Society, Washington, DC (2006)
23. Xu, C., Cheung, S.C., Chan, W.K., Ye, C.: Heuristics-based strategies for resolving context inconsistencies in pervasive computing applications. In: ICDCS 2008, pp. 713–721. IEEE Computer Society, Washington, DC (2008)
24. Chen, C., Ye, C., Jacobsen, H.-A.: Hybrid context inconsistency resolution for context-aware services. In: Proceedings of 9th IEEE International Conference on Pervasive Computing and Communications, PerCom 2011 (2011)
25. Anagnostopoulos, C., Ntarladimas, Y., Hadjiefthymiades, S.: Situational computing: an innovative architecture with imprecise reasoning. J. Syst. Softw. **80**(12), 1993–2014 (2007)
26. Etzion, O., Skarbovsky, I., Magid, Y., Zolotorevsky, N., Rabinovich, E.: Context aware computing and its utilization in event-based systems. Tutorial presented in DEBS, Cambridge, UK (2010)
27. Anicic, D., Fodor, P., Rudolph, S., Stojanovic, N.: EP-SPARQL: a unified language for event processing and stream reasoning. In: WWW 2011, pp. 635–644 (2011)
28. Verginadis, Y., Patiniotakis, I., Papageorgiou, N., Apostolou, D., Mentzas, G.: A goal driven dynamic event subscription approach. In: The 6th ACM International Conference on Distributed Event-Based Systems (DEBS 2012), Berlin, Germany (2012)

SDDS Based Hierarchical DHT Systems for an Efficient Resource Discovery in Data Grid Systems

Riad Mokadem(✉), Franck Morvan, and Abdelkader Hameurlain

Institut de Recherche en Informatique de Toulouse (IRIT),
Paul Sabatier University, 118 Route de Narbonne, 31062 Toulouse, France
{mokadem,morvan,hameur}@irit.fr

Abstract. Most of the existing hierarchical Distributed Hash Table (DHT) systems, used for a resource discovery, generate considerable maintenance overhead which affects the routing efficiency in large scale systems. In this paper, we propose a Scalable Distributed Data Structures (SDDS) based Hierarchical DHT (SDDS- HDHT) solution for an efficient data source discovery in data Grid systems. Our solution deals with a reduced number of gateway peers running a DHT protocol. Each of them serves also as a proxy for second level peers in a single Virtual Organization (VO), structured as a SDDS. The performance evaluation of the proposed method proved the discovery cost reduction especially for intra-VO resource discovery queries. It also proved significant system maintenance save especially when peers frequently join/ leave the system.

Keywords: Resource discovery · Data grid · Peer to peer systems · Distributed hash table · Scalable distributed data structure

1 Introduction

A resource discovery constitutes an important step on a query evaluation in unstable and large scale environments [28], e.g., data Grid systems. It consists to discover resources (e.g., computers, data) that are needed to perform distributed applications [27]. Throughout this paper, we focus only on data source discovery in data Grid systems.

Classical resource discovery approaches in Grid systems are either centralized or hierarchical and were proved inefficient as the scale of Grid systems rapidly increases. In fact, excessive access to a centralized peer generates a bottleneck and its failure paralyzes the entire system and the using of web services, inspired from hierarchical models, has been explored in several research works as [42]. Although the advantage of being Open Grid Service Architecture (OGSA) [5] compliant, i.e., each resource is represented as a web service, this strategy is not adapted for grid environments since the dynamicity property in such environments [11]. Also, the using of the agents based hierarchical model gives partial results despite its capabilities to address scalability [34]. Several research works have adopted the Peer-to-Peer solutions to deal with resource discovery in Grid systems [24, 36]. P2P routing algorithms have been classified as structured or unstructured [37]. Although the good fault tolerance properties in

E. Simperl et al. (Ed.): ESWC 2012 Satellite Events, LNCS 7540, pp. 327–342, 2015.
DOI: 10.1007/978-3-662-46641-4_25

P2P unstructured systems (e.g., KaZaa [15]), the flooding –used in each search- is not scalable since it generates large volume of unnecessary traffic in the network. Structured Peer-to-Peer systems as DHT are self-organizing distributed systems designed to support efficient and scalable lookups in spite of the dynamic properties in such systems. Classical flat DHT systems organize peers, having the same responsibility, into one overlay network with a lookup performance of O(log(N)), for a system with N peers. However, the using of a flat DHT do not consider neither the autonomy of virtual organizations and their conflicting interests nor the locality principle, a crucial consideration in Grids [12]. Moreover, typical structured P2P systems as Chord [35] and Pastry [32] suffer not only from temporary unavailability of some of its components but also from churn effect. It occurs in the case of the continuous leaving and entering of peers into the system.

Recent research works as [27] proved that hierarchical overlays have the advantages of faster lookup times, less messages exchanged between peers, and scalability. They are valuable for small and medium sized Grids, while the super peer model is more effective in very large Grids [32]. In this context, several research works [6, 7, 14, 22, 23, 25, 33] and [43] proved that hierarchical DHT systems based on the super peer model [41] can be advantageous for complex systems. Hierarchical DHT solutions employ a multi level overlay network where peers are grouped according to a common property such as resource type or locality for a lookup service used in discovery [6]. In this context, a Grid can be viewed as a network composed of several, proprietary Grids, virtual organizations (VO) [23] where every VO is dedicated to an application domain (e.g., biology, pathology). Within a group, one or more peers are selected as super peers to act as gateways to peers in the other groups. Furthermore, most existing hierarchical DHT solutions neglect the churn effect and deal only with the improving performance of the overlay network routing. They mainly generate significant additional overhead to large scale systems [39]. Zöls et al. [43] gives a cost-based analysis of hierarchical DHT design. Performances depend on the ratio between super peers and the total number of peers. In fact, super peers are put more under stress especially if the leaf peers number increase. Several solutions have been proposed to reduce maintenance costs [6, 10, 16, 17, 21, 31] and [44]. Despite a good strategy to manage a churn in [34] through a lazy update of the network access points, inter-organizations lookups were expensive because of the complex addressing system. The SG-1 algorithm, proposed in [21], is based on the information exchange between super peers through a gossip protocol [1]. It aims to find the optimal number of super peers in order to reduce maintenance costs. However, most of these solutions add significant load at some peers which generates an additional overhead to large scale systems.

This paper deals with both improving lookup costs and managing churn while minimizing the overhead added to the system. We propose a Scalable Distributed Data Structure (SDDS) based Hierarchical DHT (SDDS- HDHT) solution for an efficient resource discovery in data Grids. It combines SDDS [20] and DHT routing schemes. Our solution employs a two level overlay network which deals with two classes of peers: super peers called also gateway peers (GP) and second level peers (SLP). Gateway peers establish a structured DHT based overlay. Only one peer, per VO, is considered as a gateway. Then, each of them serves as a proxy for second level peers in a single VO, structured as a SDDS. SDDS were among the first research works dealing with structured P2P systems. [40] noted numerous similarities between Chord and the best known SDDS scheme: LH* (Linear Hashing) [20]. Both implement key search and have no centralized components. Resource discovery queries, in our system, are classified into intra-VO and inter-VO queries. The intra-VO discovery consists to apply the principle of locality by favoring the discovery in a local VO through the efficient LH* routing system. Key based queries in LH*, in its $LH^*_{RS}{}^{P2P}$ versus, need at most two hops to find the target when the key search in a DHT needs $\log_B(N)$ hops [40], when N is the number of peers in the system and B typically equal to 4. In fact, super peers are not concerned by intra-VO queries unlike previous solutions as [43] which put super peers more under stress. Regarding Inter-VO queries, they are first routed to the reduced DHT overlay which permits to locate the gateway peer affected to the VO containing the resource to discover. Then, another $LH^*_{RS}{}^{P2P}$ lookup is done in order to discover metadata describing this resource. The proposed solution takes also into account the continuous leaving and joining of peers into the system (dynamicity properties of Grid environments). Only the arrival of a new VO requires the DHT maintenance. The connection/ disconnection of gateways do not require excessive messages exchanged between peers in order to maintain the system. This is done through a lazy maintenance which avoids high maintenance costs [16]. A simulation analysis evaluates performances of the proposed solution through comparison with previous hierarchical DHT solution performances. It shows the reduction of lookups costs especially for intra-VO queries. It also provides a significantly maintenance costs reduction, especially when peers frequently join/leave the system. The rest of the paper is structured as follows. Section 2 recalls hierarchical DHT and SDDS principles. Section 3 presents our resource discovery solution through the proposed protocol. It also describes the maintenance process. The simulation analysis study section shows the benefit of our proposition. The final section contains concluding remarks and future work.

2 Preliminaries

2.1 $LH^*_{RS}{}^{P2P}$ Scalable Distributed Data Structure

Many variant of SDDS have been proposed [20, 40]. In this paper, we deal with $LH^*_{RS}{}^{P2P}$ scheme which improves later LH* variants (LH^*_{RS}, LH^*_g…) and assume that the reader is familiar with a linear hashing algorithm LH* as presented in [19]. Scalable Distributed Data Structures (SDDS), designed for P2P applications, are a class of data structures for distributed systems that allow data access by key in constant time [40]. Each peer in SDDS stores records in a bucket which splits when the file grows. Every LH* peer is both client and, potentially, data or parity server which interacts with application using the key based record search, insert, update or delete query or a scan query performing non key operations. In the resource discovery process, we deal with the key search operations (e.g., a database relation) in order to search metadata describing the data source.

Basic linear hash functions are $H_i(C) = C \bmod 2^i$ with i the file level which determines the linear hash function H_i to be applied. Each record in LH * is identified by its key. The key *C* determines the record location according to the linear hashing

Algorithm [20]. The file starts with one data bucket and one parity bucket. It scales up through data bucket splits, as the data buckets get overloaded. It can be occurred when a peer splits its data bucket. In old SDDS scheme, one peer acted as a coordinator peer. It was viewed as the single peer knowing the correct state of the file or relation. However, [40] ameliorates this scheme. Split coordinator does not constitute a centralized peer for the SDDS scheme. It intervenes only to find a new data server when a split occurs and never in the query evaluation process. Any other peer uses its local view 'image', which may be not adjusted, to find the location of a record given in the key based query. Suppose that it send its key based query by using its image which can lead to an incorrect bucket. An outdated image could result. The peer server applies another algorithm $LH^{*}{}_{RS}{}^{P2P}$ [40]. It first verifies whether its own address is the correct one by checking its guessed bucket level in the received query against its actual level (the level of LH function used to split or create the bucket). If needed, the server forwards this query. The query always reaches the correct bucket in this step i.e., if forwarding occurs, the new address has to be the correct one. It sends an Image Adjustment Message (IAM) informing the initial sender that the address was incorrect and the sender adjusts its image reusing the LH* image adjustment algorithm described in [20]. It was not the case with LH* which may need an additional hop. Proof of this property is in [40]. So, the most important property here is that the maximal number of forwarding messages for key-based addressing is one. Moreover, [40] proved the efficient recovery of more than one peer without any duplication. $LH^{*}{}_{RS}{}^{P2P}$ scheme allows the correction possibility through parity calculus. It is done using Reed Solomon (RS) codes [20]. Another advantage of using SDDS is the possibility to support range queries very well and the less vulnerability in the presence of high churn [40].

2.2 Hierarchical Distributed Hash Tables

Structured P2P systems (e.g., Chord [35], Pastry [32], CAN [30] and Tapestry [45]) offer deterministic query search results. In this paper, we have focused on a Pastry DHT system [32] which requires Log_B (N) hops, where N is the total number of peers in the system and B typically equal to 4. Pastry system is also adapted for several different applications including a global persistent storage utility as PAST [2] and a scalable publish/subscribe system called SCRIBE [29]. Pastry peer permits also to easily locate both the right ad left neighbors in the DHT. These reasons motivate us to choose the Pastry routing system. But, our method can be applied to other DHT systems. Flat DHT solutions do not take into account the autonomy of organizations. Hierarchical DHT systems partition its peers into a multi level overlay network. Because a peer joins a smaller overlay network than in flat overlay, it maintains and corrects a smaller number of routing states than in flat structure. In such systems, one or more peers are often designated as super peers. They act as gateways to other peers organized in groups in second level overlay networks.

Throughout this section, we interest to two previous hierarchical DHT solutions which we consider comparable to our solution. In Fig. 1-left, super peers establish a structured DHT overlay network when second level peers (called leaf peers) maintain only connection to their super peers. This corresponds to the Super Peer HDHT

(SP-HDHT) solution [43]. However, [22] proved that this strategy can maintain super peers more under stress by maintaining pointers between super peers and their leaf peers. Furthermore, a super peer stores metadata of all leaf peers which it is responsible and acts as a centralized resource for them. Then, performances depend on the ratio between super peer's number and the total number of peers in the system. Multi-Gateway Hierarchical DHT (MG-HDHT) solution [23] is another example of 2-levels hierarchy system having multiple gateways per VO (Fig. 1- right). The system forms a tree of rings (DHTs in this example). Typically, the tree consists of two layers, namely a global ring as the root and organizational rings at the lower level. A group identifier (*gid*) and a unique peer identifier (*pid*) are assigned to each peer. Groups are organized in the top level as DHT overlay network. Within each group, peers are organized as a second level overlay. This solution provides administrative control and autonomy of the participating organizations. Unlike efficient intra-organization lookups, inter-organization lookups are expensive since the high maintenance cost of the several gateway peers. Hence, there is a trade-off between minimizing total network costs and minimizing the added overhead to the system.

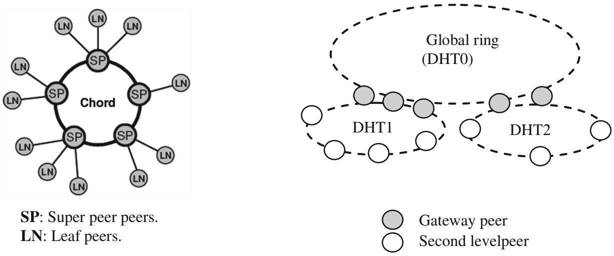


Fig. 1. SP-HDHT (left) and MG-HDHT (right) Solutions.

3 Resource Discovery Through SDDS Based Hierachical DHT Systems

A data source discovery is an important step in the query evaluation on large scale environments. It consists in search metadata describing data sources (e.g., the profile of a relation *Doctor* which is associated to a domain concept). In large scale environments, this task is complex since data sources are highly heterogeneous and constantly evolving due to data source autonomy. Furthermore, the dynamicity of peers is a major problem since the continuous joining/ leaving of peers generates prohibitive maintenance costs. In this section, we propose a SDDS based hierarchical DHT (SDDS-HDHT) solution for a resource discovery in data Grids. It aims to reduce both lookup and maintenance costs while minimizing overhead added to the system.

Resource Discovery through our solution deals with two different classes of peers: gateways (GP) and second-level peers (SLP). A Grid can be viewed as a network composed of several, proprietary Grids, virtual organizations (VO) [13] as shown in Fig. 2. Every VO is dedicated to an application domain (e.g., biology, pathology) [16]. It permits to take into account the locality principle of each VO [12]. Within a VO, one peer is selected as a gateway peer (GP). It acts as a super peer or a proxy for other peers, called second level peers (SLPs). GPs communicate with each other through a DHT overlay network. Each of them knows, through the $LH^*_{RS}{}^{P2P}$ routing system, how to interact with all SLPs belonging to the same VO. In this context, [6] proved that a DHT lookup algorithm required only minor adaptations to deal with groups instead of individual peers. In order to make a resource in VO_i visible through the DHT, hash join *H* is applied to this resource, when it joins the system, to generate a group identifier *gid*. Then, another hash function *h* is applied to this resource in order to generate a peer identifier *pid*. This permits to associate each resource to its VO [22]. We may assume that GPs are relatively more stable than SLPs. In contrast, GPs establish a structured DHT based overlay when each VO -regrouping SLPs- is structured as a SDDS. We consider here the peers as homogenous. Recall also that we have not interesting on the assignment of a joining SLP to an appropriate GP, i.e., loads balancing. We defer these issues to future work.

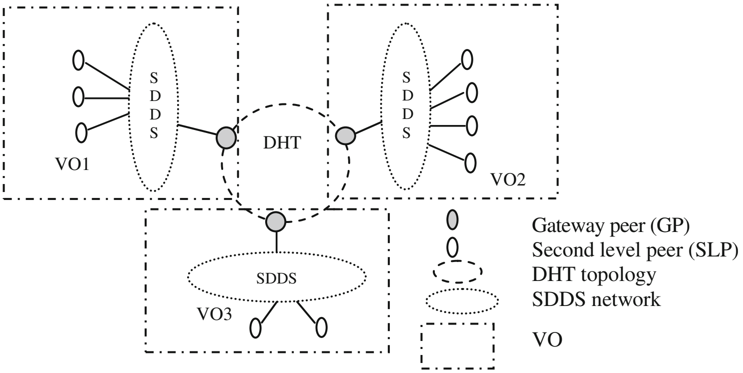


Fig. 2. SDDS based Hierarchical DHT Architecture.

3.1 Resource Discovery Protocol

After describing how VOs are connected, we present the resource discovery protocol used in the proposed SDDS-HDHT solution. Suppose that a SLP p_i Є VO_i wants to discover a resource *Res* through a resource discovery query *Q*. Let the peer p_J be the peer responsible for *Res*. Let Gp_i the GP responsible for VO_i, $Gp_{i\_}list$ the list of its neighbors in the top level DHT (e.g., the left and the right neighbors) and *Response* the metadata describing the data source *Res*. Thus, a lookup request for *Res* implies locating the peer responsible for *Res*. In our system, data source discovery queries can be classified into two types:

(i) Data source discovery within a single VO, i.e., peers p_i and p_j belong to the same VO. Then, the query Q corresponds to an intra-VO resource discovery query.
(ii) Data source discovery between VOs, i.e., peers p_i and p_j are in different VOs. Then, the query Q corresponds to an inter-VO resource discovery query.

Intra-VO resource discovery queries are evaluated through a classical $LH^*_{RS}{}^{P2P}$ routing system which is completely transparent to the top level DHT. Generally, users often access data in their application domain, i.e., in their VO. In consequence, it is important to search metadata source first in the local VO_i before searching in other VOs. This solution favors principle of locality [12]. Recall that finding a peer responsible of metadata of the searched resource requires only two messages. Finally, the peer p_J sends metadata describing Res (if founded) to p_i, the peer initiator of Q.

When the researched resource *Res* is not available in the local VO_i, resource discovery is required in other VOs. This corresponds to an inter-VO resource discovery process. Before introducing the resource discovery process, described by the algorithm shown in Fig. 3, let us recall that we have defined an interval of time noted RTT (Round- Trip Time) as in [27]. Hence, if a peer does not respond after one RTT, it is considered to be disconnected. The manner in which the RTT values are chosen during lookups can greatly affects performances under churn. [31] has demonstrated that a RTT is a significant component of lookup latency under churn. In fact, requests in peer to peer systems under a churn are frequently sent to a peer that has left the system. In our solution, a RTT is mainly useful to maximize time to discover resources when a failure occurred in a GP. In this case, p_i do not expect indefinitely. When RTT is exceeded, it considers that Gp_i is failed and consults the gateway neighbours list *Gp_i_list* received in the connection step. Then, p_i sends its query to one of the peers founded in *Gp_i_list*. Let us recall that in the connection step of any gateway peer Gp_i, this latter send its list neighbors *Gp_i_list* to p_0 in its VO. Then, p_0 forwards *Gp_i_list* to all other SLPs. It is done just on the connection step.

```
   Discover ($p_i$, Res, $VO_i$)
 { if ($LH*_{RS}^{P2P}$_Lookup(Res,$VO_i$)==1)then Send Response to $p_i$;
                                   Else forward Q to $Gp_i$
   if $Gp_i$ do not respond      then{ consult $Gp_i$_list;
                                Forward Q to neighbours of $Gp_i$;}
   if  DHT_route(Res)==$Gp_J$    then
         if ($LH*_{RS}^{P2P}$_Lookup(Res, $VO_J$)==1)
                     then send Response to $p_i$.
                     else Response = {}is sent to  $p_i$.
    else Response = {}is sent to $p_i$.
   }
```

Fig. 3. Data Source Discovery Algorithm.

Regarding the inter-VO lookup process in the SDDS-HDHT solution, it requires four steps. In the first step, a peer p_i routed the query to the gateway Gp_i. If a Gp_i failure is detected (RTT is elapsed), it requests one neighbor of Gp_i, already received. Once the query reaches a gateway peer Gp_i, a hash function *H* is applied to *Res* in order to discover the GP responsible for the VO that containing *Res*. The query arrives at some Gp_J. This is valid whenever a resource, matching the criteria specified in the query, is found in some VO_J. the third step consists to apply the $LH^*_{RS}{}^{P2P}$ routing system in the found VO_J. Gp_J routes the query to the peer p_J Є VO_J that is responsible for Res. Finally, metadata of *Res* are sent to Gp_j which forward it to p_i via the reversing path.

3.2 System Maintenance

The connection/ disconnection of a peer requires the update of the system. Furthermore, the continuous leaving and entering of peers into the system is very common in Grid systems (dynamicity properties). Peer departures can be divided into friendly leaves and peer failures. Friendly leaves enable a peer to notify its overlay neighbors to restructure the topology accordingly. Peer failures possibility seriously damages the structure of the overlay with data loss consequences. Remedying this failure generates additional maintenance cost. In structured peer-to-peer systems, such as Pastry [32] used in our system, the connection/ disconnection of one peer generates $2B*Log_B(N_T)$ messages [32]. Furthermore, the maintenance can concern the connection/ disconnection of one or more peers. Throughout this section, we explore the different factors that affect the behavior of hierarchical DHT under churn (super peer failure addressing, timeouts during lookups and proximity neighbor selection) [31]. Then, we discuss the connection/ disconnection of both GPs and SLPs.

Second Level Peer (SLPs) Connection/ Disconnection. The connection/ disconnection of a SLP p_i do not affect lookups in other peers except the possible split of a bucket if this latter gets overloaded. Let us discuss the only one required maintenance. When p_i joins some VO_i, it asks its neighbor about Gp_i and the Gp_i_list. In consequence, only two messages are required. This process avoid that several new arrival peers asked simultaneously the same GP which can constitute a bottleneck as in SP-HDHT solution. In other terms, when a new SLP arrives, it searches its GP (only one) and neighbors of this one. This process permits also to reduce messages comparing to the complex process in the MG- HDHT solution in which the new SLP should retrieve all GPs.

Gateway Peer (GP) Connection/ Disconnection. For this aim, we propose a protocol in order to reduce the overhead added to the system. When a GP connection/ disconnection occur, we distinguish two types of maintenance: (i) maintenance of the DHT and (ii) maintenance of the neighbour's lists. We will not discuss the first maintenance since it corresponds to a classical DHT maintenance [35]. In the other hand, without any maintenance protocol, a disconnection or a failure of a GP paralyzes access to all SLPs for which the Gp_i is responsible. Addressing this failure generates additional maintenance cost. Before describing the maintenance process, let us analyze the connection steps of Gp_i to VO_i. The gateway peer Gp_i send its list neighbors *Gp_i_list* (the left and right neighbor) to the nearest *SLP* p_0 in VO_i. Then, p_0 contacts peers in *Gp_i_list* to inform them about its existence (in order to have an entry to VO_i in the case of Gp_i failure). After that, p_0 send this list to all SLPs in VO_i via a multicast message. Other SLPs do not report their existence to neighbors of Gp_i.

Recall also that this process is done just once at the initial connection of Gp_i and only p_0 periodically executes a *Ping/Pong* algorithm with Gp_i. It sends a *Ping* message to Gp_i and this one answers with a *Pong* message in order to detect any failure in Gp_i. Let us discuss the case of a GP failure/ update. When a peer Gp_i is replaced by another, the process of maintenance (after the DHT maintenance) is:

(i) The new gateway Gp_{New} contacts the nearest (only one) SLP p_0 and gives him its neighbor's list *Gp_{New}_list*.

(ii) Peer p_0 inform peers in Gp_{New}*_list* about its existence. But, it does not inform other SLPs about Gp_{New}*_list*.

Remark that the peer p_0 do not send description of the new gateway peer Gp_{New} and its updated Gp_{New}*_list* to other SLPs at this moment. A lazy update is adopted. When Gp_i does not respond after a RTT period, a SLP consults its old Gp_i*_list* to reach other VOs. Thus, it rejoins the overlay network in spite of a GP_i failure. The update of this list is done during the reception of the resource discovery result as in [16]. Also, a failure of p_0 does not paralyze the system since the new GP always contacts its nearest SLP. The entry to the VO can also be done through peer p_0 since this one reported its existence in the connection step. This process allow a robust resource discovery process although the presence of dynamicity of peers. This is not the case in MG-HDHT solution when failures of all GPs in some VO paralyze the input/ output to/ from this VO. Recall also that one of the limitations that our solution suffers from: the failure of both GP_i and its neighbors in Gp_i*_list*. A solution consists in enriching the neighbors list of gateway peers with neighbors of neighbors.

4 Performance Analysis

In order to validate the proposed SDDS-HDHT solution, we evaluate its performances and compare them to those of three other resource discovery methods: (i) a flat DHT solution in order to measure the benefits of hierarchical system, (ii) the SP-HDHT solution [43] in which GPs establish a DHT overlay network when each leaf peers maintains a connection to its GP and (iii) the MG-HDHT solution in which several gateways are maintained between hierarchical levels [23].

We based on a virtual network as 10000 peers to prove the efficiency of our solution in large scale networks. We deal with a simulated environment since it is difficult to experiment thousands of peers organized as virtual organization in a real existing platform as Grid'5000 [9]. We based our experiments on a platform having four features: (i) emulation of peers, (ii) simulation of homogenous bandwidth networks and local network 100 Mb/s, (iii) using of FreePastry [4], an implementation of the Pastry DHT protocol and (iv) $LH^*{}_{RS}{}^{P2P}$ SDDS prototype implemented by Litwin's team in Dauphine University [18]. Variables used bellow are defined as follows: N_T is the number of peers in the system, N_G the number of GPs, N_{SL} the number of SLPs and α the gateway ratio. It is the ratio between GPs and the total number of peers ($N_G = \alpha.\ N_T$). Key of the searched data source corresponds to a database relation name in our experiments. For the detection of failed peers, we set a TTL to 1 s. Throughout this section, we deal with four classes of experiments: (i) Lookup performances experiments in which we are interested in elapsed times which includes the query processing and communication costs, (ii) maintenance overhead experiments in which we simulate a join/leave peers scenario and interest to the required update messages, (iii) experiments to find the optimal ratio between GPs and SLPs in order to evaluate the impact of the gateway ratio on performances (we have varied N_G but the total number of peers always stay constant) and (iv) experiments to measure the impact of the gateway session's length on the system maintenance [3].

4.1 Lookup Performances Analysis

The first experiment simulates a flat DHT solution. This equates to a configuration with $N_T/N_G = 1$ in the SDDS-HDHT solution. When we analyze the hops number required to discover one resource in both solutions, our results are always better when it concerns an intra-VO resource discovery query. In fact, $LH^*{}_{RS}{}^{P2P}$ lookup requires a maximum of two (2) messages when this number is always $log_B(N_T)$ in flat DHT solutions. For inter-Vo queries, we have showed in [26] that the theoretically worse case corresponds to $O(log_B(N_G)) + 4$ hops with SDDS-HDHT scheme. By a simple calculation, we deduce that flat DHT performances are better when our DHT overlay is composed by more than 1000 GPs. In other terms, from 10 SLPs per VO (i.e., $\alpha < 1$ %), our results are better. This is due to the fact that adding new SLPs do not influences $LH^*{}_{RS}{}^{P2P}$ lookup performances. However, these results correspond to theoretical numbers of hops for only one resource discovery query. In the case of simultaneous resource discovery messages, the results should take into account that all messages are forward to the same GP (in one VO). This generates some congestion in this peer. To confirm this, we have experimented systems with (i) 2000 gateways (5 SLPs/ VO, $\alpha = 20$ %) and (ii) 500 gateways (20 SLPs/ VO, $\alpha = -5$ %). We also interest to the number of simultaneous resource discovery queries. It is useful since it shows if the SDDS-HDHT solution is also scalable in the presence of high number of messages.

Figure 4 shows elapsed response times for resource discovery queries (intra and inter-VO queries). It confirms that SDDS-HDHT performances are always better when queries constitute intra-VO resource discovery queries. Elapsed response times are 50 % better than flat DHT solution. This is due to the reason mentioned above. Let analyze performances of inter-VO queries. When we experiment with $\alpha = 20$ %, performances are almost similar for a reduced simultaneous discovery queries. But, elapsed responses time increase from 20 queries/s. It is due to the fact that all queries transit by the same GP in each VO. However, a great leaf peers number ($\alpha = 5$ %) improves significantly our performances which are better. The save is close 10 % compared to the flat DHT solution in spite of the simultaneous messages. It provides from the gain in the DHT lookup. In fact, the probability to find the searched resource in a local VO is greater.

We have also compared SDDS-HDHT performances to both SP-HDHT and MG-HDHT performances. [43] proved that best performances are obtained with small number of gateways. We simulate a network with 100 VOs (with 100 level peers/ VO). Figure 4 shows that the SP-HDHT solution is slightly better for intra-VO queries when less simultaneous messages are used. From 70 messages/s, our solution is 10 % better than SP-HDHT solution. We explain this by the fact that intra-VO lookups are done without any GP intervention when a bottleneck is generated in each GP in the SP-HDHT solution. This is the reasons why the simultaneous messages influenced significantly the SP-HDHT results. We remark that the average response time is almost constant when we have several simultaneous messages in both SDDS-HDHT and MG-HDHT solution. We conclude that the save can be better if we experiment with great number of simultaneous discovery queries. Note that these experiments do not include the more costly connection step.

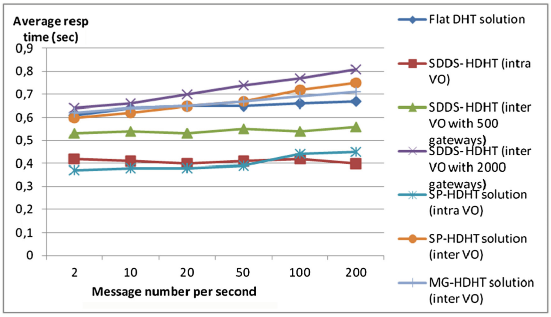


Fig. 4. SDDS-HDHT Performances vs. Flat DHT, SP-HDHT, MG-HDHT Performances.

For inter-V0 queries, simultaneous resource discovery queries influences performances of both SP-HDHT and SDDS-HDHT solutions. Bottleneck is generated since all queries transit by the same GP which increases response times. SP-HDHT response times are slightly better when we have less than 70 messages per second. From this value, results are almost close for the two solutions with slight advantage to SDDS-HDHT solution since intra-VO queries always precede inter-VO queries. We conclude that in inter-VO queries, we have dependence between performances and simultaneous queries for these two solutions. The same impact is observed with a reduced α. Regarding the MG-HDHT performances, they are better (rate of 5 %) especially for high simultaneous messages seen as inter-VO queries are propagated through several GPs.

4.2 Maintenance Analysis

We measure the impact of the peers connection/ disconnection on the system. We interest to the total messages number required when a peer joins/leaves the network. We tabulate churn in an event-based simulator which processes transitions in state (*down*, *available*, and *in use*) for each peer as in [8]. We simulate a churn phase in which several peers join and leave the system but the total number of peers N_T stays appreciatively constant. The maintenance costs are measured by the number of messages generated to maintain the system when peers join/leave the system.

Let a system with a peers distribution as $\{N_G = 100$ and 100 peers/ VO$\}$. This configuration corresponds to average results in inter-VO discovery queries performances. In these experiments, when a number of new connections/ disconnections exceed 20 peers, 10 % of them concern GPs. In this context, Flat DHT solution generates the greater number of messages. When only one SLP arrive into the system, 62 messages are required to update the Flat DHT system while this number is only 57 messages with our solution. The save is more important with the connection of 90 SLPs and 10 GPs. In this case, the messages number ratio is 4.5 (5200 messages for the DHT solution and 1600 messages with the SDDS-HDHT solution).

We compare these results to the SP-HDHT performances. Figure 5- left shows the impact of peers connection/ disconnection on the total messages number required to update the system. The numbers of update messages are closes when we have only SLPs connections/disconnections. It corresponds to the case when less than 10 peers

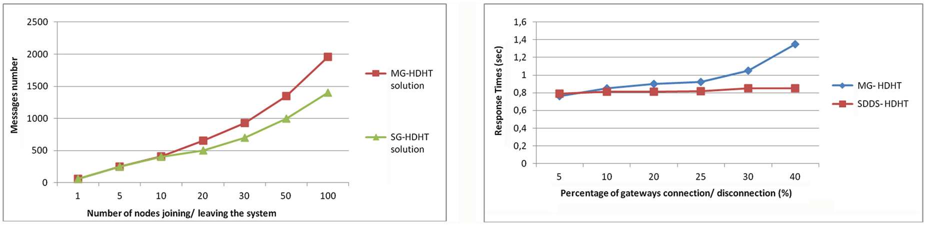


Fig. 5. Impact of: the peer connection/disconnection on the system maintenance (left) and the percentage of the GPs connection/disconnection on the response time (right).

join the system. In fact, all new peers must contact their super peer in SP-HDHT solution. Increasing the number of connection/ disconection of leaf peers can generates a bottleneck in this solution when SDDS-HDHT solution offers a significant maintenance cost gain including when the update occurs in GPs. As the number of GPs connection increase as the gain is important since the required update messages is less with our solution. The save is 59 % for the connection of 90 SLPs and 10 GPs. Certainly, updating the DHT concern both solutions. But, in the SP-HDHT solution experiments the new gateway establishes connections with all its leaf peers. It is also the case in the MG-HDHT solution. The fact that new SLPs in MG-HDHT must contact several gateways generates additional messages. It is not the case in our SDDS-HDHT solution. A new SLP contacts only its neighbour and the connection of a new GP generates only two additional messages with SLPs in the same VO.

We also experiment the impact of the percentage of the GPs arrival/ departure on the total response time as shown in Fig. 5- right. The worst case corresponds to a discovery process under a high churn. When only 5 % of gateways are replaced by other gateways, MG-HDHT solution has slightly better results than SDDS-HDHT performances. However, when this percentage increases, SDDS-HDHT performances remain stable since SLPs used the gateway neighbor's list to reach other GPs in the DHT. In MG-HDHT solution, they used the other not failed gateways in the same VO pending the update of the new GPs. From 25 % of GPs connection/ disconnection in the system, MG-HDHT curve increase significantly. Recall that we have deliberately ensured that not all GPs in the same VO are failed in MG-HDHT solution. Otherwise, a SLP in some VO$_i$ will be not able to contact any gateway of other VO$_j$ $(i \neq j)$ until. It is not the case in our solution in which SLPs can use the Gp_i_list. But, recognize that if all peers in the Gp_i_list failed, consequences are also the same as above.

4.3 Impact of the Gateway Ratio on Performances

Through these experiments, our goal is to determine optimal configurations on the three compared solutions. In first experiments, without any peer arrival/departure to the system (Fig. 6- left), a centralized overlay network with only one super peer in SP-HDHT solution generates the lowest traffic costs. The reason is that only lookup and *Ping/ Pong* messages are exchanged between the super peer and its leaf peers.

Same performances are obtained with the configuration (α = 100 %) in the three compared solution since all peers participate in a flat DHT overlay. If the number of GPs increases ($N_G > 1$), we notice increased lookup costs for the three compared solution as shown in Fig. 6- left. This cost is most important in SDDS-HDHT and SP-HDHT solution, mostly caused by the bottleneck in the only one GP. Indeed, it is due to the fact that all queries transit by the same GP when the several GPs are less in stress on the MG-HDHT solution. This cost decrease from α = 20 % in the SP-HDHT and SDDS-HDHT solutions. It is from α = 10 % in the MG-HDHT solution. We conclude that MG-HDHT solution constitutes the better solution when we have not or very little departures/ arrivals of peers in the system. However, good performances are obtained from α = 10 % with our solution which is close to real grid systems with several VOs.

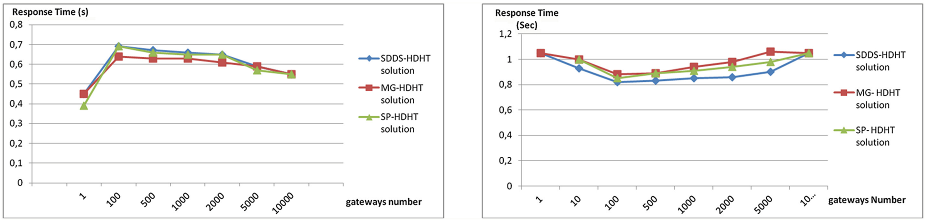


Fig. 6. Impact of the gateway number on performances: without maintenance costs (left) and including maintenance costs (right).

We also deal with experiments taking into account the arrival/ departure of peers to the system as shown in Fig. 6- right. We deal with the connection/ disconnection of 10 % of the GPs in the system and 10 % of SLPs in each VO. From α = 1 %, the maintenance cost of the MG- HDHT solution is always the most important since each GP inform all its SLPs in each arrival/ departure. It is also the case with the SP- HDHT solution with better results. This is not the case in the SDDS- HDHT solution which has the best results with α between 1 and 50 %. It is due to the fact that SLPs used a lazy update to update their neighbor's gateway list. For each value of α in this interval, the SDDS-HDHT solution generates the lowest total cost. It is valuable for the case when the major maintenance cost is generated by the departure/ arrival of SLPs but also for the case when the departure/ arrival of GPs constitute the major maintenance cost. We conclude that the best SDDS-HDHT performances are obtained with α ∈ [1 %, 20 %] which is close to real grid systems with several VOs.

5 Conclusion and Future Work

The proposed hierarchical DHT solution combines SDDS, in its $LH^*{}_{RS}{}^{P2P}$ variant, and DHT routing schemes for a data source discovery in data Grid systems. We aim to reduce lookup costs and manage churn while minimize the additional overhead to the system. We group all peers of a same domain in a same Virtual Organization (VO) in order to favor the content/path locality principle. Then, only one peer per VO runs a DHT protocol and acts as a gateway peer for second level peers in its VO. Within a VO, structured as a SDDS, the resource discovery process is based on a classical LH* routing system. Hence, the first contribution is the improvement of data source discovery query complexity especially for intra-VO queries since these queries are transparent to the DHT lookup. Regarding inter-VO resource discovery process, we have proposed a new protocol which aims to reduce the exchanged messages between peer. Only the arrival of a new VO requires the DHT maintenance and second level peers update theirs gateway peer neighbours during the resource discovery process. The performance analysis shows the benefit of our proposition through comparisons of our performances to those of previous solutions. It shows the improvement of lookup query performances especially as regards the intra-VO resource discovery queries. This is especially valid when comparing our solution to the flat DHT and SP-HDHT solutions since intra-VO queries do not require the gateway peer intervention. We have also compared our solution with another previous hierarchical DHT solution based on the using of several gateway peers (MG-HDHT solution). While sacrificing response times for simultaneous queries per second, we have important maintenance costs save when several peers join/ leave the system. Hence, it seems more reasonable to have a dynamicity of peers in a large scale environment. This save is mainly due to the lazy maintenance adopted in our solution.

Our solution can be useful in large scale environments since it generates less traffic network. Further work includes more performance studies in more realistic large grid environments with a high number of peers. Also, we would like to study the effects of alternate routing table neighbours as in [38].

References

1. Artigas, Marc S., García, Pedro, Skarmeta, Antonio FGómez: DECA: a hierarchical framework for decentralized aggregation in DHTs. In: State, Radu, van der Meer, Sven, O'Sullivan, Declan, Pfeifer, Tom (eds.) DSOM 2006. LNCS, vol. 4269, pp. 246–257. Springer, Heidelberg (2006)
2. Druschel, P., Rowstron, A.: PAST: a large-scale, persistent peer-to-peer storage utility. In: HotOS VIII, Germany (2001)
3. Fei, T., Tao, S., Gao, L., Guerin, R.: How to select a good alternate path in large peer-to-peer systems? In: Proceedings of the international conference on IEEE INFOCOM (2006)
4. http://Freepastry.org/FreePastry/
5. Foster, I., (ed.), Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Kishimoto, H., (ed.), Maciel, F., Savva, A., Siebenlist, F., Subramania, R., Treadwell, J., Von Reich, J.: The Open Grid Services Architecture, V 1.0. Global Grid Forum (2004)
6. Garcés-Erice, Luis, Biersack, Ernst W., Felber, Pascal, Ross, Keith W., Urvoy-Keller, Guillaume: Hierarchical peer-to-peer systems. In: Kosch, Harald, Böszörményi, László, Hellwagner, Hermann (eds.) Euro-Par 2003. LNCS, vol. 2790, pp. 1230–1239. Springer, Heidelberg (2003)
7. Ganesan, P., Gummadi, K., Garcia-Molina, H.: Canon in g major: designing DHTs with hierarchical structure. In: International Conference on Distributed Computing Systems 2004, pp 263–272 (2004)

8. Godfrey, P.B., Shenker, S., Stoica, I.: Minimizing churn in distributed systems. In: International Conference on SIGCOMM, pp 147–158, Italy (2006)
9. GRID'5000. www.grid5000.org
10. Gupta, I., Birman, K., Linga, P., Demers, A., Renesse, R.V.: Kelips: Building an Efficient and Stable P2P DHT through Increased Memory and Background Overhead. In: Frans Kaashoek, M., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 160–169. Springer, Heidelberg (2003)
11. Hameurlain, Abdelkader: Evolution of query optimization methods: from centralized database systems to data grid systems. In: Bhowmick, Sourav S., Küng, Josef, Wagner, Roland (eds.) DEXA 2009. LNCS, vol. 5690, pp. 460–470. Springer, Heidelberg (2009)
12. Harvey, N., Jones, M., Saoiu, S., Theimer, M., Wolman, A.: Skipnet: a scalable overlay network with practical locality properties. In: Proceedings of USITIS, Seattle, USA (2003)
13. Iamnitchi, A., Foster, I.: A peer-to-peer approach to resource location in grid environments. In: Proceedings of HPDC 2002, Edinburgh, UK (2002)
14. Joung, Y., Wang, J.-C.: Chord2: a two-layer chord for reducing maintenance overhead via heterogeneity. Comput. Netw. **51**(3), 712–731 (2007)
15. Kazaa. http://www.kazaa.com/
16. Ketata, Imen, Mokadem, Riad, Morvan, Franck: Resource discovery considering semantic properties in data grid environments. In: Hameurlain, Abdelkader, Tjoa, A.Min (eds.) Globe 2011. LNCS, vol. 6864, pp. 61–72. Springer, Heidelberg (2011)
17. Ketata, I., Mokadem, R., Morvan, Franck: Biomedical resource discovery considering semantic proprieties in data grid environments. In: Joy, M., et al. (eds.) INTECH'11, vol. 165, pp. 12–64. Springer, Heidelberg (2011)
18. http://lamsade.dauphine.fr/~litwin/default.html
19. Litwin, W.: Linear hashing: a new tool for file and table addressing. In: Stonebreaker, M. (ed.) VLDB 1980, 2nd edn. Morgan Kaufmann, San Fransisco (1995)
20. Litwin, W., Moussa, R., Schwarz, T.: LH*rs a highly available scalable distributed data structure. In: Jin, H., Rana, O.F., Pan, Y., Prasanna, V.K., et al. (eds.) Algorithms and Architectures for Parallel Processing, vol. 4494, pp. 188–197. Springer, Heidelberg (2005)
21. Montresor, A.: A robust protocol for building superpeer overlay topologies. In: IEEE International Conference on Peer-to-Peer Computing (P2P) (2004)
22. Martinez, I., Cuevas, R., Guerrero, C., Mauthe, A.: Routing performance in a hierarchical DHT-based overlay network. In: Euromicro International Conference PDP, pp. 508–515, Toulouse (2008)
23. Mislove, Alan, Druschel, Peter: Providing administrative control and autonomy in structured peer-to-peer overlays. In: Voelker, Geoffrey M., Shenker, Scott (eds.) IPTPS 2004. LNCS, vol. 3279, pp. 162–172. Springer, Heidelberg (2005)
24. Meshkova, E., et al.: A survey on Resource Discovery Mechanisms, Peer to Peer and Service Discovery Frameworks Computer Networks, pp. 2097–2128. Science Direct, Elsevier, New York (2008)
25. Mokadem, R., Hameurlain, A., Min Tjoa, A.: Resource discovery service while minimizing maintenance overhead in hierarchical DHT systems. In: International Conference on Information Integration and Web-based Applications & Services (iiWAS), Paris, France (2010)
26. Mokadem, R., Hameurlain, A.: An efficient resource discovery while minimizing maintenance overhead in SDDS based hierarchical DHT systems. Int. J. Grid Distrib. Comput. (IJGDC) **4**(3), 1–24 (2011)
27. Mastroianni, C., Talia, D., Verta, O.: Evaluating resource discovery protocols for hierarchical and super-peer grid information systems. In: 19th Euromicro International Conference (PDP) (2007)

28. Pacitti, E., Valduriez, P., Mattosso, M.: Grid data management: open problems and news issues. Int. J. Grid Comput. **5**, 273–281 (2007). Springer
29. Rajiv, R., et al.: Peer to peer based resource discovery in global grids: a tutorial. In: IEEE Communication Surveys, vol. 10, No 2, 2 nd Quarter (2008)
30. Ratnasamy, et al.: A scalable content-adressable network. In: Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, pp. 161–172 (2001)
31. Rhea, S., Geels, D., Roscoe, T., Kubiatowicz, J.: Handling churn in a DHT. In: Proceedings of the General Track: Usenix Annual Technical Conference, Boston, USA (2004)
32. Rowston, A., Druschel, P.: Pastry: scalable distributed object location and routing for large-scale peer-to-peer systems. In: Proceeding of the 18th IFIP/ACM International Conference on Distributed Systems Platforms, vol. 2218, pp. 329–350 (2001)
33. S´anchez-Artigas, M., Garc´ya, P., Pujol, J., Skarmeta, A.G.: Cyclone: a novel design schema for hierarchical DHTs. In: IEEE International Conference on Peer-to-Peer Computing (*P2P)* (2005)
34. Samad, M.E., Morvan, F., Hameurlain, A.: Resource discovery for query processing in data grids. In: Graham, J.H., et al. (eds.) 22nd International Conference on Parallel and Distributed Computing and Communication Systems, PDCCS 2009, 24–26 September 2009, Louisville, Kentucky, USA. ISCA (2009)
35. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishma, H.: CHORD: a scalable Peer to Peer Lookup Service for Internet Application. In: SIGCOMM'O, San Diego, USA (2001)
36. Trunfio, P., Talia, D., Papadakid, H., Fragoupoulou, P., Mordachini, M., Penanen, M., Popov, P., Valssov, V., Haridi, S: Peer-to-peer resource discovery in grids: models and systems. In: Future Generation Computer Systems (2007)
37. Valduriez, Patrick, Pacitti, Esther: Data management in large-scale P2P systems. In: Daydé, Michel, Dongarra, Jack, Hernández, Vicente, Palma, José MLaginha M. (eds.) VECPAR 2004. LNCS, vol. 3402, pp. 104–118. Springer, Heidelberg (2005)
38. Xiang, X., Jin, T.: Efficient secure message routing for structured peer-to-peer systems. In: International Conference on Networks Security, Wireless Communications and Trusted Computing Wuhan, China (2009)
39. Xu, Z., Min, R., Hu, Y.: HIERAS: a DHT based hierarchical P2P routing algorithm. In: Proceedings of International Conference on Parallel Processing (ICPP), pp 187–194 (2003)
40. Yakouben, H., Litwin, W., Schwarz, T.: LH*RSP2P: a scalable distributed data structure for the P2P environment.In: International Conference on New Technologies of Distributed Systems, France (2008)
41. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proceedings of International Conference on Data Engineering ICDE, Bangalore, India (2003)
42. The Web Services Resource Framework. http://www.globus.org/wsrf
43. Zöls, S., Despotovic, Z., Kellerer, W.: Cost-based analysis of hierarchical DHT design. In: International Conference, P2P'06. IEEE Computer Society, Cambridge, pp 233–239 (2006)
44. Zöls, S., Hofstatter, Q., Despotovic, Z., Kellerer, W.: Achieving and maintaining cost-optimal operation of a hierarchical DHT system. In: Proceedings of International Conference. ICC, Germany (2009)
45. Zhao, B.Y., Huang, L., Stribling, J., Rhéa, S.C.: Tapestry: a resilient global scale overlay for service deployment. IEEE Int. J. Sel. Areas Commun. **22**(1) (2004)

FaCETa: Backward and Forward Recovery for Execution of Transactional Composite WS

Rafael Angarita[1], Yudith Cardinale[1](✉), and Marta Rukoz[2]

[1] Departamento de Computación, Universidad Simón Bolívar,
Caracas 1080, Venezuela
{rangarita,yudith}@ldc.usb.ve
[2] LAMSADE, Université Paris Dauphine,
Université Paris Ouest Nanterre La Défense, Paris, France
marta.rukoz@lamsade.dauphine.fr

Abstract. In distributed software contexts, Web Services (WSs) that provide transactional properties are useful to guarantee reliable Transactional Composite WSs (TCWSs) execution and to ensure the whole system consistent state even in presence of failures. Failures during the execution of a TCWS can be repaired by forward or backward recovery processes, according to the component WSs transactional properties. In this paper, we present the architecture and an implementation of a framework, called FaCETa, for efficient, fault tolerant, and correct distributed execution of TCWSs. FaCETa relies on WSs replacement, on a compensation protocol, and on unrolling processes of Colored Petri-Nets to support failures. We implemented FaCETa in a Message Passing Interface (MPI) cluster of PCs in order to analyze and compare the behavior of the recovery techniques and the intrusiveness of the framework.

1 Introduction

Large computing infrastructures, like Internet, increase the capacity to share information and services across organizations. For this purpose, Web Services (WSs) have gained popularity in both research and commercial sectors. Semantic WS technology [20] aims to provide for rich semantic specifications of WSs through several specification languages such as OWL for Services (OWL-S), the Web Services Modeling Ontology (WSMO), WSDL-S, and Semantic Annotations for WSDL and XML Schema (SAWSDL) [15]. That technology supports WS composition and execution allowing a user request be satisfied by a Composite WS, in which several WSs and/or Composite WSs work together to respond the user query.

WS Composition and the related execution issues have been extensively treated in the literature by guaranteeing user *QoS* requirements and fault tolerant execution [7,11,16,18,21]. WSs that provide transactional properties are

This work was supported by the Franco-Venezuelan CNRS-FONACIT project N°22782.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 343–357, 2015.
DOI: 10.1007/978-3-662-46641-4_26

useful to guarantee reliable Transactional Composite WSs (TCWSs) execution, in order to ensure that the whole system remains in a consistent state even in presence of failures. TCWS becomes a key mechanism to cope with challenges of open-world software. Indeed, TCWS have to adapt to the open, dynamically changing environment, and unpredictable conditions of distributed applications, where remote services may be affected by failures and availability of resources [27].

Generally, the control flow and the order of WSs execution is represented with a structure, such as workflows, graphs, or Petri-Nets [3,6,7,14]. The actual execution of such TCWS, carried out by an Executer, could be deployed with centralized or distributed control. The Executer is in charge of *(i)* invoking actually WSs for their execution, *(ii)* controlling the execution flow, according to data flow structure representing the TCWS, and *(iii)* applying recovery actions in case of failures in order to ensure the whole system consistency; failures during the execution of a TCWS can be repaired by forward or backward recovery processes, according to the component WSs transactional properties.

In previous works [9,10] we formalized a fault tolerant execution control mechanism based on Colored-Petri Nets (CPN), which represent the TCWS and the compensation process. In [9] unrolling algorithms of CPNs to control the execution and backward recovery were presented. This work was extended in [10] to consider forward recovery based on WS replacement; formal definitions for WSs substitution process, in case of failures, were presented. In [10], we also proposed an Executer architecture, independent of its implementation, to execute a TCWS following our proposed fault tolerant execution approach.

In this paper, we present an implementation of our Executer framework, called FaCETa (FAult tolerant Cws Execution based on Transactional properties), for efficient, fault tolerant, and correct distributed execution of TCWSs. We implemented FaCETa in a Message Passing Interface (MPI) cluster of PCs in order to analyze and compare the efficiency and performance of the recovery techniques and the intrusiveness of the framework. The results show that FaCETa efficiently implements fault tolerant strategies for the execution of TCWSs with small overhead.

2 TCWS Fault-Tolerant Execution

This Section recall some important issues related to transactional properties and backward and forward recovery, presented in our previous works [7,9,10]. We consider that the Registry, in which all WSs are registered with their corresponding *WSDL and OWLS documents*, is modeled as a Colored Petri-Net (CPN), where WS inputs and outputs are represented by places and WSs, with their transactional properties, are represented by colored transitions - colors distinguish WS transactional properties [7]. The CPN representing the Registry describes the data flow relation among all WSs.

We define a query in terms of functional conditions, expressed as input and output attributes; *QoS* constraints, expressed as weights over criteria; and the required global transactional property as follows. A Query Q is a 4-tuple (I_Q, O_Q, W_Q, T_Q), where:

- I_Q is a set of input attributes whose values are provided by the user,
- O_Q is a set of output attributes whose values have to be produced by the system,
- $W_Q = \{(w_i, q_i) \mid w_i \in [0,1]$ with $\sum_i w_i = 1$ and q_i is a QoS criterion$\}$, and
- T_Q is the required transactional property; in any case, if the execution is not successful, nothing is changed on the system and its state is consistent.

A TCWS, which answers and satisfies a Query Q, is modeled as an acyclic marked `CPN`, called CPN-$TCWS_Q$, and it is a sub-net of the Registry `CPN`[1]. The *Initial Marking* of CPN-$TCWS_Q$ is dictated by the user inputs. In this way, the execution control is guided by an unrolling algorithm.

2.1 Transactional Properties

The transactional property (TP)of a WS allows to recover the system in case of failures during the execution. The most used definition of individual WS transactional properties ($TP(ws)$) is as follows [8,13]. Let s be a WS: s is **pivot** (p), if once s successfully completes, its effects remains forever and cannot be semantically undone (compensated), if it fails, it has no effect at all; s is **compensatable** (c), if it exists another WS s', which can semantically undo the execution of s, after s successfully completes; s is **retriable** (r), if s guarantees a successfully termination after a finite number of invocations; the **retriable** property can be combined with properties p and c defining **pivot retriable** (pr) and **compensatable retriable** (cr) WSs.

In [11] the following TP of TCWS have been derived from the TP of its component WSs and their execution order(sequential or parallel). Let tcs be a TCWS: tcs is **atomic** ($\boldsymbol{a}$), if once all its component WSs complete successfully, they cannot be semantically undone, if one component WS does not complete successfully, all previously successful component WSs have to be compensated; tcs is **compensatable** (c), if all its component WSs are compensatable; tcs is **retriable** (r), if all its component WSs are retriable; the retriable property can be combined with properties $\boldsymbol{a}$ and c defining **atomic retriable** ($\boldsymbol{a}r$) and **compensatable retriable** (cr) TCWSs.

According to these transactional properties, we can establish two possible recovery techniques in case of failures:

- *Backward* recovery: it consists in restoring the state (or a semantically closed state) that the system had at the beginning of the TCWS execution; i.e., all the successfully executed WSs, before the failure, must be compensated to undo their produced effects. All transactional properties (p, $\boldsymbol{a}$, c, pr, $\boldsymbol{a}r$, and cr) allow backward recovery;

[1] A marked `CPN` is a `CPN` having tokens in its places, where tokens represent that the values of attributes (inputs or outputs) have been provided by the user or produced by a WS execution.

- *Forward* recovery: it consists in repairing the failure to allow the failed WS to continue its execution. Transactional properties pr, $\boldsymbol{ar}$, and cr allow forward recovery.

2.2 Backward Recovery Process: Unrolling a Colored Petri-Net

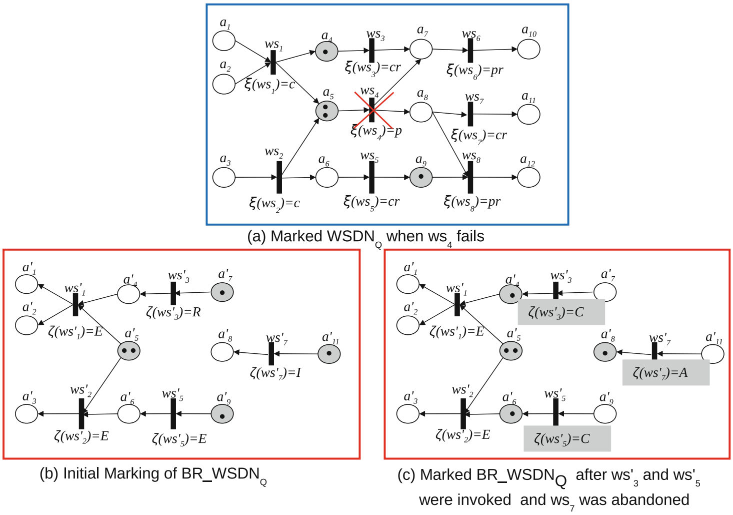
The global TP of CPN-$TCWS_Q$ ensures that if a component WS, whose TP does not allow forward recovery fails, then all previous executed WSs can be compensated by a backward recovery process. For modeling TCWS backward recovery, we have defined a backward recovery `CPN`, called BRCPN-$TCWS_Q$, associated to a CPN-$TCWS_Q$ [9]. The component WSs of BRCPN-$TCWS_Q$ are the compensation WSs, s', corresponding to all c and cr WSs in CPN-$TCWS_Q$. The BRCPN-$TCWS_Q$ represents the compensation flow, which is the inverse of the execution order flow. In BRCPN-$TCWS_Q$ a color of a transition s' represents the execution state of its associated transition s in the CPN-$TCWS_Q$ and is updated during CPN-$TCWS_Q$ execution. Color(s') $\in$ {I = 'initial', R = 'running', E = 'executed', C = 'compesated', A = 'Abandonned'} thus, if color(s') = 'E' means that its corresponding WS s is being executed. In [7,9] we propose techniques to automatically generate both `CPNs`, CPN-$TCWS_Q$ and BRCPN-$TCWS_Q$.

The execution control of a TCWS is guided by an unrolling algorithm of its corresponding CPN-$TCWS_Q$. A WS is executed if all its inputs have been provided or produced, i.e., each input place has as many tokens as WSs produce them or one token if the user provides them. Once a WS is executed, its input places are unmarked and its output places (if any) are marked.

The compensation control of a TCWS is also guided by an unrolling algorithm. When a WS represented by a transition s fails, the unrolling process over CPN-$TCWS_Q$ is halted, an *Initial Marking* on the corresponding BRCPN-$TCWS_Q$ is set (tokens are added to places associated to input places of the faulty WS s (if any), and to places representing inputs of BRCPN-$TCWS_Q$, i.e., places without predecessors) and a backward recovery is initiated with the unrolling process over BRCPN-$TCWS_Q$. We illustrate a backward recovery in Fig. 1. The marked CPN-$TCWS_Q$ depicted in Fig. 1(a) is the state when ws_4 fails, the unrolling of CPN-$TCWS_Q$ is halted, and the *Initial Marking* on the corresponding BRCPN-$TCWS_Q$ is set to start its unrolling process (see Fig. 1(b)), after ws'_3 and ws'_5 are executed and ws_7 is abandoned before its invocation, a new *Marking* is produced (see Fig. 1(c)), in which ws'_1 and ws'_2 are both ready to be executed and can be invoked in parallel. Note that only **compensatable** transitions have their corresponding compensation transitions in BRCPN-$TCWS_Q$.

2.3 Forward Recovery Process: Execution with WS Substitution

During the execution of TCWSs, if a failure occurs in an advanced execution point, a backward recovery may incur in highly wasted resources. On the other hand, it is hard to provide a **retriable** TCWS, in which all its components are **retriable** to guaranty forward recovery. We proposed an approach based



(a) Marked WSDN$_Q$ when ws$_4$ fails

(b) Initial Marking of BR_WSDN$_Q$

(c) Marked BR_WSDN$_Q$ after ws'$_3$ and ws'$_5$ were invoked and ws$_7$ was abandoned

Fig. 1. Example of backward recovery

on WS substitution in order to try forward recovery [10]. TCWS composition and execution processes deal with *service classes* [1], which group WSs with the same semantic functionality, i.e., WSs providing the same operations but having different WSDL interfaces (input and output attributes), transactional support, and QoS. When a WS fails, if it is not **retriable**, instead of backward recovery, a substitute WS is searched to be executed on behalf of the faulty WS.

In a *service class*, the functional equivalence is defined according the WSs input and output attributes. A WS s is a functional substitute, denoted by $\equiv_F$, to another WS s^*, if s^* can be invoked with at most the input attributes of s and s^* produces at least the same output attributes produced by s. s is an Exact Functional substitute of s^*, denoted by $\equiv_{EF}$, if they have the same input and output attributes. Figure 2 illustrates several examples: $ws_1 \equiv_F ws_2$, however $ws_2 \not\equiv_F ws_1$, because ws_1 does not produce output a_5 as ws_2 does. $ws_1 \equiv_F ws_3$, $ws_3 \equiv_F ws_1$, and also $ws_1 \equiv_{EF} ws_3$.

In order to guarantee the TCWS global TP, a WS s can be replaced by another WS s^*, if s^* can behave as s in the recovery process. Hence, if $TP(s)$=p, in which case s only allows backward recovery, it can be replaced by any other WS because all transactional properties allow backward recovery. A WS with $TP(s) = pr$ can be replaced by any other **retriable** WS (pr,$\boldsymbol{ar}$,cr), because all of them allow forward recovery. An **atomic** WS allows only backward recovery, then it can be replaced by any other WS which provides backward recovery. A **compensatable** WS can be replaced by a WS that also provides compensation as c and cr WSs. A cr WS can be only replaced by another cr WS because it is the only one allowing forward and backward recovery. Thus, a WS s is Transactional substitute of another WS s^*, denoted by $\equiv_T$, if s is a Functional substitute of s^* and their transactional properties allow the replacement.

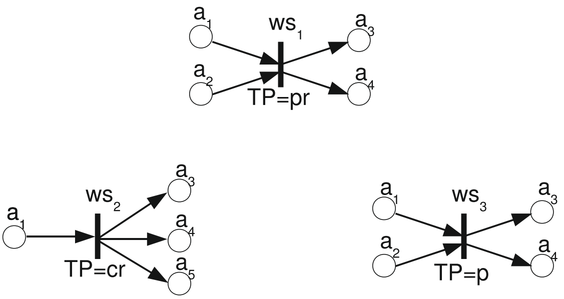


Fig. 2. Example of functional substitute WSs

In Fig. 2, $ws_1 \equiv_T ws_2$, because $ws_1 \equiv_F ws_2$ and $TP(ws_2) = cr$, then ws_2 can behave as a *pr* WS; however $ws_1 \not\equiv_T ws_3$, even $ws_1 \equiv_F ws_3$, because as $TP(ws_3) = p$, w_3 cannot behave as a *pr* WS. Transactional substitution definition allows WSs substitution in case of failures.

When a substitution occurs, the faulty WS s is removed from the CPN-$TCWS_Q$, the new s^* is added, but we keep the original sequential relation defined by the input and output attributes of s. In that way, the CPN-$TCWS_Q$ structure, in terms of sequential and parallel WSs, is not changed. For **compensatable** WSs, it is necessary Exact Functional Substitute to do not change the compensation control flow in the respective BRCPN-$TCWS_Q$. In fact, when a **compensatable** WS is replaced, the corresponding compensate WS must be also replaced by the new corresponding one in the BRCPN-$TCWS_Q$. The idea is to try to finish the TCWS execution with the same properties of the original one.

2.4 Protocol in Case of Failures

In case of failure of a WS s, depending on the $TP(s)$, the following actions could be executed:

- if $TP(s)$ is **retriable** (pr, $\boldsymbol{ar}$, cr), s is re-invoked until it successfully finishes (forward recovery);
- otherwise, another Transactional substitute WS, s^*, is selected to replace s and the unrolling algorithm goes on (trying a forward recovery);
- if there not exists any substitute s^*, a backward recovery is needed, i.e., all executed WSs must be compensated in the inverse order they were executed; for parallel executed WSs, the order does not matter.

When in a *service class* there exist several WSs candidates for replacing a faulty s, it is selected the one with the best quality measure. The quality of a transition depends on the user query Q and on its QoS values. WSs Substitution is done such that the substitute WS locally optimize the QoS. If several transitions have the same value of quality, they can be randomly selected to be the substitute. A similar quality measure is used in [7] to guide the composition process. Then, during the execution, we keep the same heuristic to select substitutes.

3 FaCETa: An TCWS Executer with Backward and Forward Recovery Support

In this Section we present the overall architecture of FaCETa, our execution framework. The execution of a TCWS in FaCETa is managed by an Execution Engine and a collection of software components called Engine Threads, organized in a three levels architecture. In the first level the Execution Engine receives the TCWS (represented by a CPN). It is in charge of initiating, controlling, and monitoring the execution of the TCWS. To do so, it launches, in the second layer, an Engine Thread for each WS in TCWS. Each Engine Thread is responsible for the execution control of its WS. They receive WS inputs, invoke the respective WS, and forward its results to its peers to continue the execution flow. In case of failure, all of them participate in the backward or forward recovery process. Actual WSs are in the third layer. Figure 3 roughly depicts the overall architecture.

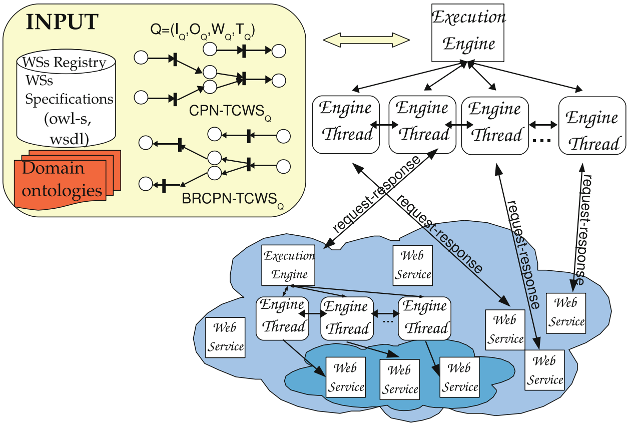
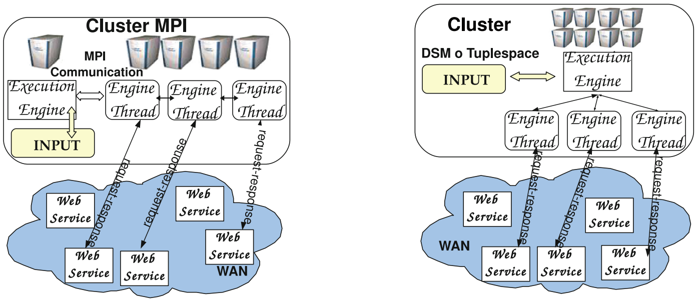


Fig. 3. FaCETa Architecture

By distributing the responsibility of executing a TCWS across several Engine Threads, the logical model of our Executer allows distributed execution of a TCWS and is independent of its implementation, i.e., this model can be implemented in a distributed memory environment supported by message passing (see Fig. 4(a)) or in a shared memory platform, e.g., supported by a distributed shared memory [22] or tuplespace [19] systems (see Fig. 4(b)). The idea is to place the Executer in different physical nodes (e.g., a highly available and reliable cluster computing) from those where actual WSs are placed. The Execution Engine needs to have access to the WSs Registry, which contains the *WSDL and OWLS* documents. The knowledge required at runtime by each Engine Thread (e.g., WS semantic and ontological descriptions, WSs predecessors and successors, transactional property, and execution control flow) can be directly extracted from the CPNs in a shared memory implementation or sent



(a) Distributed memory system

(b) Distributed shared memory system

Fig. 4. Implementation of FaCETa.

by the Execution Engine in a distributed memory implementation. In this paper, we have implemented a prototype of FaCETa in a distributed memory platform using MPI.

Typically, a component of a TCWS can be a simple transactional WS or TCWS. Thus, we consider that transitions in the CPN, representing the TCWS, could be WSs or TCWSs. WSs have their corresponding *WSDL and OWLS* documents. TCWSs can be encapsulated into an Executer; in this case the Execution Engine has its corresponding *WSDL and OWLS* documents. Hence, TCWSs may themselves become a WS, making TCWS execution a recursive operation (see Fig. 3).

3.1 Distributed Memory Implementation of FaCETa

We implemented FaCETa in a MPI Cluster of PCs (i.e., a distributed memory platform) following a Master/Slaves-SPDM (Single Process Multiple Data) parallel model. The Execution Engine runs in the front-end of the Cluster waiting user execution requests. To manage multiple client requests, the Execution Engine is multithreading. The deployment of a TCWS implies several steps: *Initial,* WS *Invocation*, and *Final* phases. In case of failures, recovery phases could be executed: *Replacing* phase, allowing forward recovery or *Compensation* phase for backward recovery.

Whenever the Execution Engine (master) receives a CPN-$TCWS_Q$ and its corresponding BRCPN-$TCWS_Q$, it performs the *Initial* phase: *(i)* start, in different nodes of the cluster, an Engine Thread (peer slaves) responsible for each transition in CPN-$TCWS_Q$, sending to each one its predecessor and successor transitions as CPN-$TCWS_Q$ indicates (for BRCPN-$TCWS_Q$ the relation is inverse) and the corresponding *WSDL and OWLS* documents (they describe the WS in terms of its inputs and outputs, its functional substitute WSs, and who is the compensation WS, if it is necessary); and *(ii)* send values of attributes in I_Q to Engine Threads representing WSs who receive them. Then the master waits for a successfully execution or for a message *compensate* in case of a backward recovery is needed.

Once Engine Threads are started, they receive the part of CPN-$TCWS_Q$ and BRCPN-$TCWS_Q$ that each Engine Thread concerns on, sent by the Execution Engine in the *Initial* phase. Then, they wait for the input values needed to invoke their corresponding WSs. When an Engine Thread receives all input values (sent by the master or by other peers) and all its predecessor peers have finished, it executes the WS *Invocation* phase, in which the actual WS is remotely invoked. If the WS finishes successfully, the Engine Thread sends WS output values to Engine Threads representing its successors and wait for a *finish* or *compensate* message. If the WS fails during the execution, the Engine Thread tries a forward recovery: if TP(WS) is **retriable**, the WS is re-invoked until it successfully finishes; otherwise the Engine Thread executes the *Replacing* phase: the Engine Thread has to determine the best substitute among the set of functional substitute WSs; it calculates the quality of all candidates according their QoS criteria values and the preferences provided in the user query; the one with the best quality is selected to replace the faulty WS; this phase can be executed for a maximum number of times ($MAXTries$). If replacing is not possible, the *Compensation* phase has to be executed: the Engine Thread responsible of the faulty WS sends the message *compensate* to Execution Engine and control tokens to successor peers of the compensation WS, in order to inform about this failure and start the unrolling process over BRCPN-$TCWS_Q$; once an Engine Thread receives all control tokens, it invokes the compensation WS; the unrolling of BRCPN-$TCWS_Q$ ensure the invocation of compensation WSs, s', in the inverse order in which their corresponding WS, s, were executed. Note that forward recovery is executed only by the Engine Thread responsible of the faulty WS, without intervention of the master neither other peers; while backward recovery need the intervention of all of them.

If the TCWS was successfully executed, in the *Final* phase the master receives all values of attributes of O_Q, in which case it broadcasts a *finish* message to all slaves to terminate them, and returns the answer to the user; otherwise it receives a *compensate* message indicating that a backward recovery has to be executed, as it was explained above, and returns an error message to the user.

Assumptions: In order to guarantee the correct execution of our algorithms, the following assumptions are made: *(i)* the network ensures that all packages are sent and received correctly; *(ii)* the Execution Engine and Engine Threads run in a reliable cluster, they do not fail; *(iii)* the Engine Threads receive all WS outputs when its corresponding WS finishes, they cannot receive partial outputs from its WS; and *(iv)* the component WSs can suffer silent or stop failures (WSs do not response because they are not available or a crash occurred in the platform); we do not consider runtime failures caused by error in inputs attributes (e.g., bad format or out of valid range) and byzantine faults (the WS still responds to invocation but in a wrong way).

4 Results

We developed a prototype of FaCETa, using Java 6 and MPJ Express 0.38 library to allow the execution in distributed memory environments. We deployed FaCETa in a cluster of PCs: one node for the Execution Engine and one node for each Engine Thread needed to execute the TCWS. All PCs have the same configuration: Intel Pentium 3.4 GHz CPU, 1 GB RAM, Debian GNU/Linux 6.0, and Java 6. They are connected through a 100 Mbps Ethernet interface.

We generated 10 **compensatable** TCWSs. All those TCWSs were automatically generated by a composition process [7], from synthetic datasets comprised by 800 WSs with 7 replicas each, for a total of 6400 WSs. Each WS is annotated with a transactional property and a set of *QoS* parameters, however for our experiments we only consider the response time as the *QoS* criteria. Replicas of WSs have different response times.

The OWLS-API 3.0 was used to parse the WS definitions and to deal with the OWL classification process.

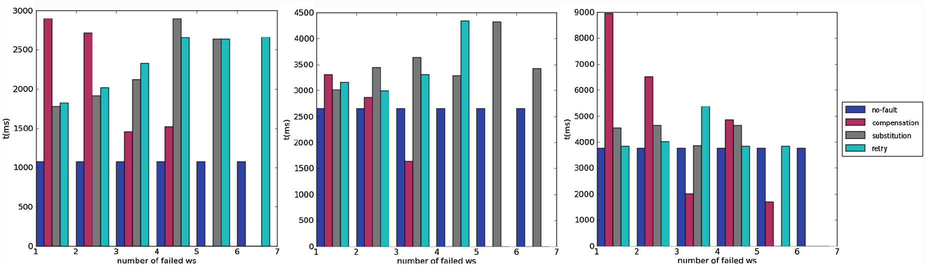
The first group of experiments were focussed on a comparative analysis of the recovery techniques. The second group of experiments evaluates the overhead incurred by our framework in control operations to perform the execution of a TCWS and to execute the fault tolerant mechanisms.

To simulate unreliable environments, we define five different conditions where in all WSs have the same probability of failure: 0.2, 0.15, 0.1, 0.005, and 0.001. The executions on these unreliable environments were done in three scenarios to support the failures: *(i)* backward recovery (compensation, red bars in Fig. 5), *(ii)* forward recovery because all WSs are retriable (retry, light blue bars in Fig. 5), and *(iii)* forward recovery (substitution, gray bars in Fig. 5). On each scenario all TCWSs were executed 10 times.

Each TCWS was also executed 10 times in a reliable environment, in which all WSs have 0 as probability of failures (no-faulty, blue bars in Fig. 5) in order to classify them according their average total execution time in three groups: less than 1500 ms (Fig. 5(a)), (ii) between 1500 ms and 3500 ms (Fig. 5(b), and (more than 3500 ms (Fig. 5(c)).

In Fig. 5 we plot the average of the total execution time according the number of failed WSs, in order to compare all recovery techniques. The results show that when the number of failed WSs is small (i.e., the probability of failures is less than 20 %) backward recovery (compensation) is the worst strategy because almost all component WSs had been executed and have to be compensated. Moreover, when the average of the total execution time of TCWSs is high (greater than 1500 ms) forward recovery with retry strategy is better than forward recovery with substitution due to the substitute normally has a bigger response time than the faulty WS. By the other side, in cases in which the probability of failure is greater than 30 %, backward recovery with compensation behaves better than the other ones (even if the final result is not produced) because there are many faulty services and only few have to be compensated.

Another issue that can be observed is the number of outputs received before the backward recovery mechanism has to be executed. In this experiment,



(a) Total Exec. Time less than 1500ms

(b) Total Exec. Time between 1500ms and 3500ms

(c) Total Exec. Timemore than 3500ms

Fig. 5. Executions on the unreliable environments

the average percentage of outputs received before compensation was 37 %. All these outputs are lost or delivered as a set of incomplete (and possibly meaningless and useless) outputs to the user. This percentage is related to the average percentage of compensated services, which is 80 %, confirming the overhead, the possible unfulfillment of QoS requirements, and the lost outputs. Definitely, backward recovery should be executed only in absence of another alternative, at early stages of execution of a TCWS, or high faulty environments.

To measure the intrusiveness of FaCETa incurred by control activities, we execute the same set of experiments describe above, but we set to 0 the response time of all WSs. Table 1 shows the average overhead under all different scenarios.

Table 1. Average overhead incurred by FaCETa

	Average overhead (ms)	% overhead increased
No fault	611.7	
Compensation	622.38	2 %
Substitution	612.82	0.2 %
Retry	612.01	0.05 %

The average overhead of FaCETa only depends on the number of component WSs in a TCWS. It does not depend on the total response time of TCWS. It means that while the total response time is higher the overhead % will decrease. It is clear that the reason behind the backward recovery strategy overhead (increased by 2 %) is the amount of coordination required to start the compensation and the fact that a new WS execution (the compensation WS execution) has to be performed for each successfully executed WS, in order to restore the consistent system state. Additionally, the compensation process has to be done following the unrolling algorithm of the respective BRCPN-$TCWS_Q$. We do not consider to wait before the retry of a failed WS execution; therefore, the increased overhead of retry a WS is almost imperceptible.

As the *service class* for each WS is sent by the Execution Engine in the *Initial* phase, each Engine Thread has the list of the functional substitute WSs sorted according their quality, then there is virtually no overhead when searching for a functional substitute WS to replace a faulty one.

Based on the results presented above, we can conclude that FaCETa efficiently implements fault tolerant strategies for the execution of TCWSs with admissible small overhead.

5 Related Work

Regarding fault tolerant execution of Composite WSs (CWSs), there exist centralized and distributed approaches. Generally centralized approaches [17,23,25] consider, besides compensation process, alternative WSs in case of failures or absent WSs, however they extend the classical 2PC protocol, which is time consuming, and they are not transparent to users and developers.

In distributed approaches, the execution process proceeds with collaboration of several entities. We can distinct two kinds of distributed coordination approach. In the first one, nodes interact directly based on a peer-to-peer application architecture and collaborate, in order to execute a CWS with every node executing a part of it [2,5,16,18,28]. In the second one, they use a shared space for coordination [4,12,19].

FENECIA framework [16] provides an approach for managing fault tolerance and *QoS* in the specification and execution of CWSs. FENECIA introduces WS-SAGAS, a transaction model based on arbitrary nesting, state, vitality degree, and compensation concepts to specify fault tolerant CWS as a hierarchy of recursively nested transactions. To ensure a correct execution order, the execution control of the resulting CWS is hierarchically delegated to distributed engines that communicate in a peer-to-peer fashion. A correct execution order is guaranteed in FENECIA by keeping track of the execution progress of a CWS and by enforcing forward and backward recovery. To manage failures during the runtime it allows the execution retrial with alternative candidates. FACTS [18], is another framework for fault tolerant composition of transactional WSs based on FENECIA transactional model. It combines exception handling strategies and a service transfer based termination protocol. When a fault occurs at runtime, it first employs appropriate exception handling strategies to repair it. If the fault cannot be fixed, it brings the TCWS back to a consistent termination state according to the termination protocol (by considering alternative services, replacements, and compensation). In [28] a fault handling and recovery process based on continuation-passing messaging, is presented. Nodes interpret such messages and conduct the execution of services without consulting a centralized engine. However, this coordination mechanism implies a tight coupling of services in terms of spatial and temporal composition. Nodes need to know explicitly which other nodes they will potentially interact with, and when, to be active at the same time. In [2] all replicas of a WS are simultaneously invoked. Only results of the first replica finished are accepted, other executions are halted or ignored. As our work, in [5] a rollback workflow is automatically created considering the service dependencies. Those frameworks support users and developers to construct CWS based on WS-BPEL technologies, then they are not transparent to users and developers.

Another series of works rely on a shared space to exchange information between nodes of a decentralized architecture, more specifically called a tuple space [12,19]. The notion of a tuplespace is a piece of memory shared by all interacting parties. Using tuplespace for coordination, the execution of a (part of a) workflow within each node is triggered when tuples, matching the templates registered by the respective nodes, are present in the tuplespace. Thus, the templates a component uses to consume tuples, together with the tuples it produces, represent its coordination logic. In [19] approach to replace a centralized BPEL engine by a set of distributed, loosely coupled, cooperating nodes, is presented. This approach presents a coordination mechanism where the data is managed using a tuplespace and the control is driven by asynchronous messages exchanged between nodes. This message exchange pattern for the control is derived from a Petri Net model of the workflow. In [19], the workflow definition is transformed into a set of activities, that are distributed by passing tokens in the Petri Net. In [12] an alternative approach is presented, based on the chemical analogy. Molecules (data) are floating in a chemical solution, and react according to reaction rules (program) to produce new molecules (resulting data). The proposed architecture is composed by nodes communicating through a shared space containing both control and data flows, called the multiset. Through a shared multiset, containing the information on both data and control dependencies needed for coordination, chemical WSs are co-responsible for carrying out the execution of a workflow in the CWS in which they appear. Their coordination is decentralized and distributed among individual WS chemical engine executing a part of the workflow. As this approach, in our approach the coordination mechanism stores both control and data information independent of its implementation (distributed or shared memory). However, none of these works manage failures during the execution.

Facing our approach against all these works, we overcome them because the execution control is distributed and independent of the implementation (it can be implemented in distributed or shared memory platforms), it efficiently executes TCWSs by invoking parallel WSs according the execution order specified by the `CPN`, and it is totally transparent to users and WS developers, i.e., user only provides its TCWS, that could be automatically generated by the composition process [7] and no instrumentation/modification/specification is needed for WSs participating in the TCWS; while most of these works are based on WS-BPEL and/or some control is sitting closely to WSs and have to be managed by programmers.

There exist some recent works related to compensation mechanism of CWSs based on Petri-Net formalism [21,24,26]. The compensation process is represented by Paired Petri-Nets demanding that all component WSs have to be compensatable. Our approach considers other transactional properties (e.g., pr, cr, $\boldsymbol{ar}$) that also allows forward recovery and the compensation Petri-Net can model only the part of the TCWS that is compensable. Besides, in those works, the Petri-Nets are manually generated and need to be verified, while in our approach they are automatically generated.

6 Conclusions and Future Work

In this paper we have presented FaCETa, a framework for ensuring *correct and fault tolerant execution order* of TCWSs. The execution model is distributed, can be implemented in distributed or share memory systems, is independent of implementation of WS providers, and it is transparent to users and developers. To support failures, FaCETa implements forward recovery by replacing the faulty WS and backward recovery based on a unrolling process over a `CPN` representing the compensation flow. We have presented a distributed memory implementation of FaCETa in order to compare the behavior of both recovery techniques. The results show that FaCETa efficiently implements fault tolerant strategies for the execution of TCWSs with small overhead.

We are currently working on implementing FaCETa in a distributed shared memory platform in order to test the performance of the framework in centralized and decentralized platforms. Our intention is to compare both implementations under different scenarios (different characterizations of `CPNs`) and measure the impact of compensation and substitution on QoS.

References

1. Azevedo, V., Mattoso, M., Pires, P.: Handling dissimilarities of autonomous and equivalent web services. In: Proceedings of Caise-WES (2003)
2. Behl, J., Distler, T., Heisig, F., Kapitza, R., Schunter, M.: Providing fault-tolerant execution of web-service-based workflows within clouds. In: Proceedings of the 2nd International Workshop on Cloud Computing Platforms (CloudCP) (2012)
3. Brogi, A., Corfini, S., Popescu, R.: Semantics-based composition-oriented discovery of web services. ACM Trans. Internet Techn. **8**(4), 1–39 (2008)
4. Buhler, P., Vidal, J.M.: Enacting BPEL4WS specified workflows with multiagent systems. In: The Workshop on Web Services and Agent-Based Engineering (2004)
5. Bushehrian, O., Zare, S., Rad, N.K.: A workflow-based failure recovery in web services composition. J. Softw. Eng. Appl. **5**, 89–95 (2012)
6. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: Web service selection for transactional composition. In: International Conference on Computational Science (ICCS). Elsevier Science-Procedia Computer Science Series, vol. 1(1), pp. 2689–2698 (2010)
7. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: CPN-TWS: a colored petri-net approach for transactional-qos driven web service composition. Int. J. Web Grid Serv. **7**(1), 91–115 (2011)
8. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: Transactional-aware web service composition: a survey. IGI Global - Advances in Knowledge Management (AKM) Book Series (2011)
9. Cardinale, Y., Rukoz, M.: Fault tolerant execution of transactional composite web services: an approach. In: Proceedings of the Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) (2011)

10. Cardinale, Y., Rukoz, M.: A framework for reliable execution of transactional composite web services. In: Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES) (2011)
11. El Haddad, J., Manouvrier, M., Rukoz, M.: TQoS: Transactional and QoS-aware selection algorithm for automatic web service composition. IEEE Trans. Serv. Comput. **3**(1), 73–85 (2010)
12. Fernandez, H., Priol, T., Tedeschi, C.: Decentralized approach for execution of composite web services using the chemical paradigm. In: IEEE International Conference on Web Services, pp. 139–146 (2010)
13. Gaaloul, W., Bhiri, S., Rouached, M.: Event-based design and runtime verification of composite service transactional behavior. IEEE Trans. Serv. Comput. **3**(1), 32–45 (2010)
14. Hogg, C., Kuter, U., Munoz-Avila, H.: Learning hierarchical task networks for nondeterministic planning domains. In: The 21st International Joint Conference on Artificial Intelligence (IJCAI 2009) (2009)
15. Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema. W3C Candidate Recommendation (January 2007). http://www.w3.org/TR/sawsdl/
16. Lakhal, N.B., Kobayashi, T., Yokota, H.: FENECIA: failure endurable nested-transaction based execution of compo site Web services with incorporated state analysis. VLDB J. **18**(1), 1–56 (2009)
17. Liu, A., Huang, L., Li, Q., Xiao, M.: Fault-tolerant orchestration of transactional web services. In: Aberer, K., Peng, Z., Rundensteiner, E.A., Zhang, Y., Li, X. (eds.) WISE 2006. LNCS, vol. 4255, pp. 90–101. Springer, Heidelberg (2006)
18. Liu, A., Li, Q., Huang, L., Xiao, M.: FACTS: a framework for fault tolerant composition of transactional web services. IEEE Trans. Serv. Comput. **3**(1), 46–59 (2010)
19. Martin, D., Wutke, D., Leymann, F.: Tuplespace middleware for petri net-based workflow execution. Int. J. Web Grid Serv. **6**, 35–57 (2010)
20. McIlraith, S., Son, T.C., Zeng, H.H.: Semantic web services. IEEE Intell. Syst. **16**(2), 46–53 (2001)
21. Mei, X., Jiang, A., Li, S., Huang, C., Zheng, X., Fan, Y.: A compensation paired net-based refinement method for web services composition. Adv. Inf. Sci. Serv. Sci. **3**(4), 169–181 (2011)
22. De Oliveira, J., Cardinale, Y., Federico, J., Chacón, R., Zaragoza, D.: Efficient distributed shared memory on a single system image operating system. In: Latin-American Conference on High Performance Computing, pp. 1–7 (2010)
23. Park, J.: A high performance backoff protocol for fast execution of composite web services. Comput. Ind. Eng. **51**, 14–25 (2006)
24. Rabbi, F., Wang, H., MacCaull, W.: Compensable WorkFlow nets. In: Dong, J.S., Zhu, H. (eds.) ICFEM 2010. LNCS, vol. 6447, pp. 122–137. Springer, Heidelberg (2010)
25. Schafer, M., Dolog, P., Nejdl, W.: An environment for flexible advanced compensations of web service transactions. ACM Trans. Web **2**, 1–36 (2008)
26. Wang, Y., Fan, Y., Jiang, A.: A paired-net based compensation mechanism for verifying web composition transactions. In: The 4th International Conference on New Trends in Information Science and Service Science (2010)
27. Qi, Y., Liu, X., Bouguettaya, A., Medjahed, B.: Deploying and managing web services: issues, solutions, and directions. VLDB J. **17**, 537–572 (2008)
28. Yu, W.: Fault handling and recovery in decentralized services orchestration. In: The 12th International Conference on Information Integration and Web-Based Applications & Services, iiWAS, pp. 98–105. ACM (2010)

Demonstration Session

Sgvizler: A JavaScript Wrapper for Easy Visualization of SPARQL Result Sets

Martin G. Skjæveland(✉)

Department of Informatics, University of Oslo, Oslo, Norway
martige@ifi.uio.no

Abstract. Sgvizler is a small JavaScript wrapper for visualization of SPARQL results sets. It integrates well with HTML web pages by letting the user specify SPARQL SELECT queries directly into designated HTML elements, which are rendered to contain the specified visualization type on page load or on function call. Sgvizler supports a vast number of visualization types, most notably all of the major charts available in the Google Chart Tools, but also by allowing users to easily modify and extend the set of rendering functions, e.g., specified using direct DOM manipulation or external JavaScript visualization tool-kits. Sgvizler is compatible with all modern web browsers.

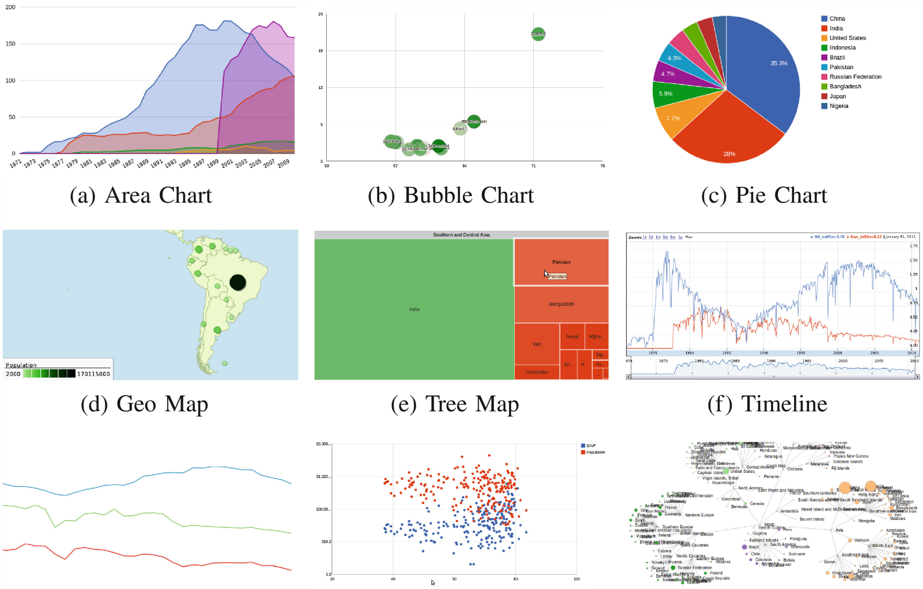
1 Introduction

The prominent way of displaying data residing in a SPARQL endpoint to a human user is to list the results of a DESCRIBE query for a given resource in a table. While tables are good for rendering heterogeneous data, needless to say, they do not convey the meaning of, e.g., geographical, temporal or quantitative data as well as respectively maps, timelines or different charts, such as pie charts and bar charts, do. There are different approaches to leverage this problem: The Data-gov Wiki project uses XSLT to convert the XML result of SPARQL SELECT queries into a format edible by the Query interface of the Google Chart Tools [1], which is used to render the data into charts and maps [2]. SPARQL Web Pages [3] uses a special-purpose vocabulary to specify how web pages are to be built from SPARQL query instructions. Spark [4] is a JavaScript library for processing SPARQL SELECT queries put directly into the HTML markup, which may be rendered into a list, a table, a line chart or a pie chart, or into other representations using custom-made visualization functions. Sgvizler [5], the topic of this paper, is a JavaScript wrapper of SPARQL result set visualization combining many of the ideas of the above-mentioned approaches into a powerful, easy-to-use and cross-browser visualization tool.

2 Overview of Sgvizler

Sgvizler is a JavaScript wrapper of SPARQL result set visualization, like Spark. What makes Sgvizler special is the ease with which it lets one integrate the visualization of SPARQL SELECT query result sets directly into web pages, *combined*

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 361–365, 2015.
DOI: 10.1007/978-3-662-46641-4_27



(a) Area Chart (b) Bubble Chart (c) Pie Chart

(d) Geo Map (e) Tree Map (f) Timeline

(g) Sparkline (h) Scatter Chart (i) Force-directed Graph

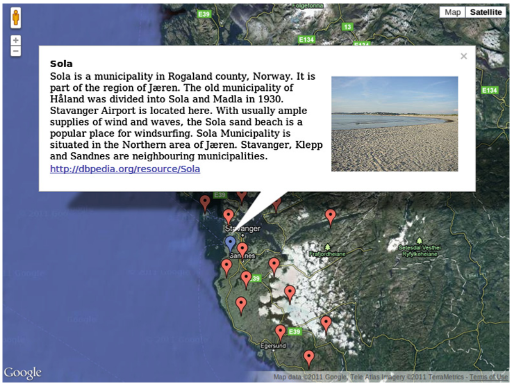
Fig. 1. Sgvizler chart examples.

with the large number of visualization types it supports and its compatibility of different origin SPARQL endpoint querying for all major modern browsers and most SPARQL endpoints. In addition, Sgvizler is built to be easily extensible by having a clear and simple API for adding user-defined rendering functions. A few samples of the available visualization types are found in Figs. 1 and 2, and Example 1 shows how simply visualizations can be added into web pages by the use of Sgvizler.

Example 1. The snippet below shows the source code of an HTML element which renders into the chart in Fig. 2 on page load. Input data to Sgvizler is set using HTML5 compatible `data-` prefixed attributes. The endpoint address and SPARQL query is given by the attributes `data-sgvizler-endpoint` and `data-sgvizler-query`,[1] respectively. The chart type to draw (`sMap`) is specified with `data-sgvizler-chart`, and additional options to the rendering function is listed in `data-sgvizler-chart-options`. The output format of the SPARQL endpoint is specified with `data-sgvizler-format` (permissible values are `xml`, `json`, `jsonp`) and `data-sgvizler-loglevel="2"` sets the level of feedback given to the user while preparing the chart.

```
<div id="id1" data-sgvizler-endpoint="http://dbpedia.org/sparql"
     data-sgvizler-query="SELECT ?lat ?long ?name ?text ?url ?image
      { ?url a dbpo:AdministrativeRegion ; dct:subject dbp:Rogaland ;
```

[1] The query is simplified to save space; this and other live examples are found at [5].


Map
Satellite
Folgefonna
E39
13
E134
Stavanger
Sandnes
Frafjordheiane
Setesdal Vesthei Ryfylkeheiane
Egersund
Google
Map data ©2011 Google, Tele Atlas Imagery ©2011 TerraMetrics - Terms of Use

Sola

Sola is a municipality in Rogaland county, Norway. It is part of the region of Jæren. The old municipality of Håland was divided into Sola and Madla in 1930. Stavanger Airport is located here. With usually ample supplies of wind and waves, the Sola sand beach is a popular place for windsurfing. Sola Municipality is situated in the Northern area of Jæren. Stavanger, Klepp and Sandnes are neighbouring municipalities.

http://dbpedia.org/resource/Sola

Fig. 2. An `sMap` of the municipalities in Rogaland, Norway—with a fact bubble about Sola.

```
        rdfs:label ?name ; geo:lat ?lat ; geo:long ?long .
        rdfs:comment ?text ; dbpo:thumbnail ?image }"
    data-sgvizler-chart="sMap"
    data-sgvizler-chart-options="dataMode=markers|showTip=true"
    data-sgvizler-format="jsonp"
    data-sgvizler-loglevel="2"
    style="width: 800px; height: 600px;"></div>
```

Sgvizler is lightweight: 31 KB in minified condition and 6 KB minified and gzipped. It relies on external libraries for visualization, endpoint communication and DOM manipulation. Currently, the following chart types and rendering functions are available: quantitative data charts (line chart, area chart, column chart, bar chart, bubble chart, scatter chart, sparkline, pie chart, candlestick chart, motion chart, gauge), hierarchical data charts (tree map, org chart), geographical visualizations (maps, geo chart, geo map), graphs, lists, tables and a generic text rendering function. It is released under an MIT License, and the complete source code, documentation, examples, issues are available at [5].

2.1 How Does It Work?

There are three intended ways of using Sgvizler: adding queries directly into HTML markup—as in Example 1, issuing a query and visualization options using an HTML form, or by direct JavaScript function call. For this explanation we will assume the first. On page load all HTML elements designated for Sgvizler visualization are collected and processed asynchronously. Each query is sent, together with the format the query results should be returned as—either XML or JSON, to the specified endpoint using jQuery's `ajax()` function [6]. The returned result set, expected by Sgvizler to be of one of the formats described by W3C [7,8], is parsed into a Google DataTable object. The DataTable object is then paired with drawing options and passed to the specified rendering function which fills the HTML element where the query was collected with its visualization results.

The DataTable class serves as input parameter type for all of the chart functions in Google's Chart Tools. This means that all these charts are readily available for Sgvizler visualization. Additionally, Sgvizler is designed such that any user-defined rendering functions must take a DataTable object as input. This makes it possible to use the same code for handling all visualization functions and easy to register new functions. Also, a DataTable object is equipped with many convenient functions for editing and querying its contents,[2] making it a helpful object to render into new representations.

The `sMap` function used in Example 1 is an example of a simple user-defined function. It extends the native Google Map function from three arguments to six, making it easier to create HTML formatted fact bubbles for points on the map, as shown in Fig. 2. The Force-directed Graph function (see Fig. 1i) is a different example, created entirely by use of the JavaScript visualization library D3 [9].

2.2 SPARQL Query Design

When designing SPARQL queries for visualization by Sgvizler, the order of the columns in the result set, i.e., the order of the variables in the `SELECT` block, is crucial. As indicated by the variable names in the query found in Example 1,[3] the `sMap` function expects the two first columns in the result set to contain respectively the latitude and longitude values of points to plot on the map. The remaining result set columns for `sMap` set respectively the heading, the body text, a clickable link and the link to an image to place in the fact bubble which appears when the point on the map is selected. Different rendering functions have different requirements on data input format. The data format for each function is described on the Sgvizler homepage [5].

2.3 Browser and Endpoint Compatibility

Disregarding SPARQL endpoint communication, Sgvizler has the same web browser compatibility as the external JavaScript libraries it uses. For jQuery and the Google Chart Tools this means compatibility with all reasonably new web browsers.[4] The compatibility of endpoint communication is a more complex matter. In general JavaScript has to abide by the *same origin policy*, a security measure which, for the purpose of Sgvizler, means that it cannot retrieve data from a SPARQL endpoint at a different domain, subdomain or port than where the script lives. This is a precaution that surely does not fit well with the idea of distributing data with SPARQL endpoints. However, *Cross-Origin Resource*

[2] See http://code.google.com/apis/chart/interactive/docs/reference.html.

[3] Even though the names of the variables in the example indicate their contents, the actual names are not important for the visualization function.

[4] A list of jQuery supported browsers are available at [6]. Google Chart Tools' information on the subject is: "Charts are rendered using HTML5/SVG technology to provide cross-browser compatibility (including VML for older IE versions) and cross platform portability to iPhones, iPads and Android" [1].

Sharing (CORS) [10] is a specification that aims to safely allow such requests and is supported by most modern browsers—the notable exception being Opera. In order for CORS to work the endpoint server must be *CORS-enabled*, meaning essentially that its header response must include a list of domains (or a wildcard $*$) with which it allows CORS communication. The specification is currently a W3C Working Draft, and not all SPARQL endpoints are CORS-enabled.

A second way to circumvent the same origin policy is to use *JSONP*. Data retrieved as JSONP is returned as a function call on the data in JSON format, thus exploiting that the HTML `<script>` tag is not required to respect the same origin policy. This, of course, requires that the endpoint can return data in JSONP format; luckily many do. Sgvizler supports receiving data in JSONP format. It has successfully been tested on CORS-enabled endpoints with output formats XML [7] and JSON [8] for the browsers Firefox 3.6, Chrome 12, Internet Explorer 8 and Safari 5.1, and, as expected, it does not work for Opera 11.51. On endpoints which return JSONP Sgvizler has been successfully tested also on Opera 11.51, including all the above-mentioned browsers. For same origin requests, endpoint communication is no longer a compatibility issue, thus in such cases Sgvizler works for all browsers compatible with the external JavaScript libraries used.

3 Future Work

The following future work items have been identified. *Technical issues:* Reduce page load time with more parallelization of tasks and selective library import. Improve Sgvizler's "external" API. *More graph visualizations:* RDF data naturally lends itself especially well for graph visualizations. However, the only graph visualization function available in Sgvizler is currently Force-directed Graph and it is in early development. *Linked Data tool integration:* Integrate Sgvizler with popular Linked Data SPARQL frontends. *Vocabulary sensitivity:* Make Sgvizler able to suggest good visualizations based on the vocabulary used by the dataset.

References

1. Google Chart Tools. http://code.google.com/apis/chart/
2. Guang Zheng, J., Ding, L.: How to render SPARQL results using Google visualization API. December 2011. http://iw.rpi.edu/wiki/How_to_render_SPARQL_results_using_Google_Visualization_API
3. Knublauch, H.: SPARQL Web Pages. http://uispin.org/
4. Vrandečić, D., Harth. A.: Spark. http://km.aifb.kit.edu/sites/spark/
5. Skjæveland, M.G.: Sgvizler. http://code.google.com/p/sgvizler/
6. jquery. http://jquery.com/
7. Beckett, D., Broekstra, J., (eds.): SPARQL Query Results XML Format. W3C Recommendation, W3C (2008). http://www.w3.org/TR/rdf-sparql-XMLres/
8. Grant Clark, K., Feigenbaum, L., Torres, E., (eds.): Serializing SPARQL Query Results in JSON. W3C Working Group Note, W3C (2008). http://www.w3.org/TR/rdf-sparql-json-res/
9. Bostock, M.: D3.js. http://mbostock.github.com/d3/
10. van Kesteren, A., (ed.): Cross-Origin Resource Sharing. W3C Working Draft, W3C, July 2010. http://www.w3.org/TR/cors/

Exploring History Through Newspaper Archives

Jasna Škrbec(✉), Marko Grobelnik, and Blaž Fortuna

Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana, Slovenia
{jasna.skrbec,marko.grobelnik,blaz.fortuna}@ijs.si

Abstract. This demo presents a web application which implements a pipeline for searching and browsing through newspaper archives. It uses a combination of information extraction, enrichment and visualization algorithms to help users to grasp a large amount of articles normally collected in archives. Illustrative results show appropriateness of the proposed pipeline for searching and browsing news archives.

Keywords: Newspaper archives · Data mining · Visualization

1 Introduction

Newspapers with a long tradition have gathered large news archives. In recent years some newspapers invested in the digitalization of archives, resulting in a million document corpora. The articles would typically be annotated with the metadata quality and quantity largely depending on the publishers and an archive type.

Typical search and browse interfaces do not work well with the archives. They are not specialized for news archives and as such do not take advantage of their inherent structure. Archives are not just a collection of articles, but they hide stories that can be presented in one or more articles, which happened in one or more locations during some period of time. News archives store rich historic information and can be turned into a valuable knowledge resource with a proper use of semantic technologies and data mining techniques. This demo introduces Archive Explorer, a system for annotating and presenting the archives in order to provide them an easier access to information and content connected with it.

The demo interface is designed to cover two common scenarios. The first scenario is the visualization of a particular article in the context of the overall archive. The second scenario is the summarization of a large collection of articles, providing glimpse of the events, entities, and topics covered by them. Interactive faceted search is used throughout the system to help with the navigation.

The system uses an annotation and contextualization pipeline, used to pre-process individual articles. In the annotation step, topics and entities occurring in the articles are recognized and linked to their corresponding resources in Linked Open Data (LOD) datasets. In the contextualization step, the articles are connected between each other based on their topicality, time, locations, events, important people, etc.

Connecting entities with LOD datasets brings additional relationships between entities into the archive. Disambiguation of an entity with its corresponding resource in

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 366–370, 2015.
DOI: 10.1007/978-3-662-46641-4_28

DBpedia or Freebase can be used to link articles talking about the same entity, but using different labels. For example, if someone would search for Princess Diana, they would not find articles where she is mentioned with her real name Diana Spencer or with name that people gave her, Lady Di. But because the archive disambiguated with resources from LOD, the user would find all relevant articles, even if the query string does not directly occur in the article. Providing users with a consolidated view of the information that is crumbled across different articles is one of the core components of Archive Explorer and this heavily relies on LOD resources.

Linking entities with LOD resources can also be used to provide additional descriptions about the entities. For example, what is Princess Diana's date of birth and death, that she was married to Prince Charles, and so on. Such information can provide an additional context to the users doing a research on a history of Princes Diana, letting their focus on specific events, and not worry about some general characteristics.

The final goal is that this additional data and information gathered together and presented in a nice way will help users to better understand the archive and save their precious time by reducing the amount of articles required to read in order to understand a particular story.

2 Archive Explorer

A news data flow in our system starts with a text mining of the articles one-by-one and creating a database from the newly extracted and contextualized data. The second part provides contextual browsing and querying capabilities of the archive. The architecture can be seen on Fig. 1.

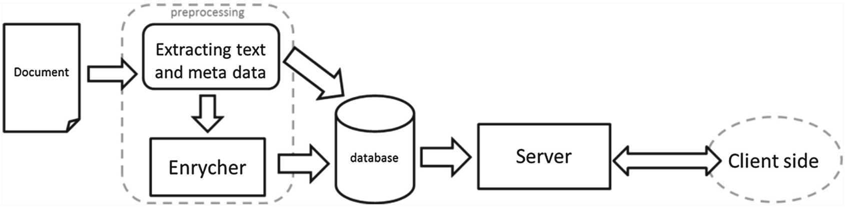


Fig. 1. The architecture of Archive Explorer.

2.1 Extracting Data

The system uses a service-oriented framework Enrycher [1] to pre-processes articles. It extracts information with pattern-based and supervised learning knowledge extraction techniques [2] and produces a list of entities, with some of them linked to resources from several LOD dataset: DBpedia, Freebase, New York Times Topics, OpenCyc and Yago. We use the extracted entities and links to LOD to provide information about people and organizations involved, information in which cities, countries or other known places events happened and information about other things, like important milestones that are included in viewed articles. Enrycher also provides a taxonomy categorization using DMoz categories, and a set of descriptive keywords.

The output of Enrycher provides a large part of articles' context. These are extended with information extracted from linked LOD resources.

2.2 Browse and Search

Once data is pre-processed and stored it needs to be presented properly. A regular search form is upgraded with a faceted search interface to help the users in browsing around and is appropriate for explorative analysis. If the users know more specifically what they are looking for, they can search across several dimensions and get the search results with all contextual information. One of goals of Archive Explorer is to put a power of the queries and advantages of the visualization together to make context useful and transparent. An application for dynamic re-ranking and visualization of search results Searchpoint [3] is used to let users sort the search results. It uses entities, connections between entities and articles for ranking and ordering.

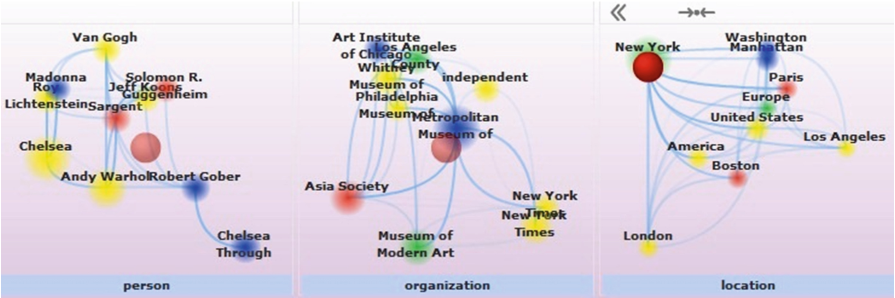
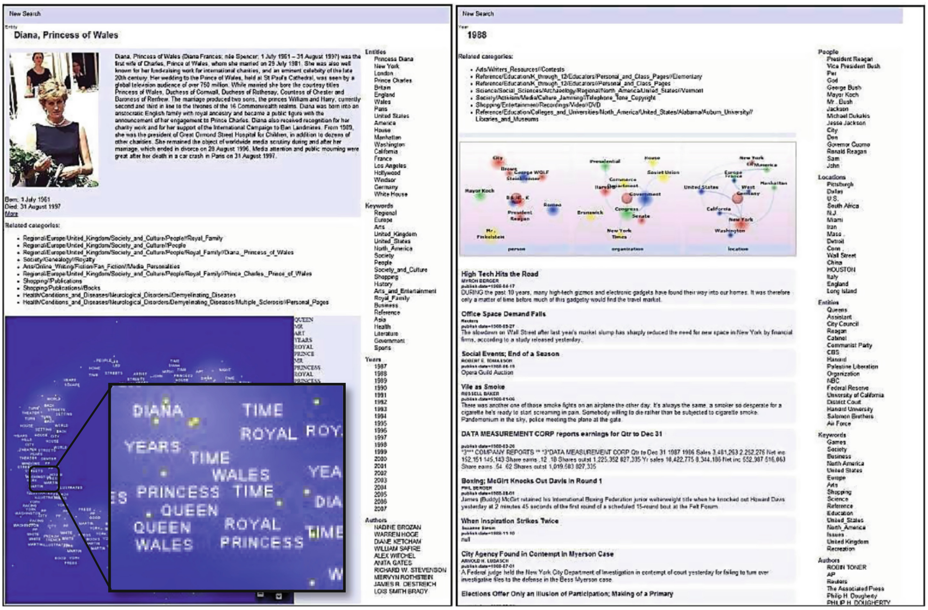


Fig. 2. Searchpoint visualization of entities. In location window red dot is dragged up to the Brooklyn.

The extracted entities are classified into several types (person, organization, location) and are used to divide the ranking criteria into three different parts of visualization. Every part is presented in its own window and entities are illustrated with spots of different colours. For choosing entity in specific window, users can drag a red dot around with a mouse, which results in a new ranking of articles. The order is changed in a way, so the articles most connected with entities nearest to the red dot are pushed to the top of list of the search results. With this users are narrowing down their search criteria without even knowing this in advance. In Fig. 2 a red dot in a location window is put on the entity New York, in other windows red dots are left at the centre position, which means that articles connected or related to New York are at the top of search results.

2.3 Contextualization and Summarization

A text of news can be short or very long and if we want to show more news articles, it can get really messy. What we want to show is an overview over a collection of articles without losing a user in endless text. Instead of a huge amount of text, we use just information that was extracted and gathered before. If someone is interested in certain topic, he can see which people, locations or other important things are present and how they are connected with each other.



New Search

Entity

Diana, Princess of Wales

Diana, Princess of Wales (Diana Frances; née Spencer; 1 July 1961 – 31 August 1997) was the first wife of Charles, Prince of Wales, whom she married on 29 July 1981. She was also well known for her fund-raising work for international charities, and an eminent celebrity of the late 20th century. Her wedding to the Prince of Wales, held at St Paul's Cathedral, was seen by a global television audience of over 750 million. While married she bore the courtesy titles Princess of Wales, Duchess of Cornwall, Duchess of Rothesay, Countess of Chester and Baroness of Renfrew. The marriage produced two sons, the princes William and Harry, currently second and third in line to the thrones of the 16 Commonwealth realms. Diana was born into an aristocratic English family with royal ancestry and became a public figure with the announcement of her engagement to Prince Charles. Diana also received recognition for her charity work and for her support of the International Campaign to Ban Landmines. From 1989, she was the president of Great Ormond Street Hospital for Children, in addition to dozens of other charities. She remained the object of worldwide media scrutiny during and after her marriage, which ended in divorce on 28 August 1996. Media attention and public mourning were great after her death in a car crash in Paris on 31 August 1997.

Born: 1 July 1961
Died: 31 August 1997
More

Related categories:

- Regional/Europe/United_Kingdom/Society_and_Culture/People//Royal_Family
- Regional/Europe/United_Kingdom/Society_and_Culture//People
- Regional/Europe/United_Kingdom/Society_and_Culture/People/Royal_Family//Diana,_Princess_of_Wales
- Society/Genealogy//Royalty
- Arts/Online_Writing/Fiction/Fan_Fiction//Media_Personalities
- Regional/Europe/United_Kingdom/Society_and_Culture/People/Royal_Family//Prince_Charles,_Prince_of_Wales
- Shopping//Publications
- Shopping/Publications//Books
- Health/Conditions_and_Diseases/Neurological_Disorders//Demyelinating_Diseases
- Health/Conditions_and_Diseases/Neurological_Disorders/Demyelinating_Diseases/Multiple_Sclerosis//Personal_Pages

QUEEN
MR
ART
YEARS
ROYAL
PRINCE
MR
PRINCESS
ROYAL
PRINCESS

Entities
Princess Diana
New York
London
Prince Charles
Britain
England
Wales
Paris
United States
America
House
Manhattan
Washington
California
France
Los Angeles
Hollywood
Windsor
Germany
White House

Keywords
Regional
Europe
Arts
United_Kingdom
United_States
North_America
Society
People
Society_and_Culture
Shopping
History
Arts_and_Entertainment
Royal_Family
Business
Reference
Asia
Health
Literature
Government
Sports

Years
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007

Authors
NADINE BROZAN
WARREN HOGE
DIANE KETCHAM
WILLIAM SAFIRE
ALEX WITCHEL
ANITA GATES
RICHARD W. STEVENSON
MERVYN ROTHSTEIN
JAMES R. OESTREICH
LOIS SMITH BRADY

New Search

Year

1988

Related categories:

- Arts/Writers_Resources//Contests
- Reference/Education/K_through_12/Educators/Personal_and_Class_Pages//Elementary
- Reference/Education/K_through_12/Educators//Personal_and_Class_Pages
- Science/Social_Sciences/Archaeology/Regional/North_America/United_States//Vermont
- Society/Activism/Media/Culture_Jamming//Telephone_Tone_Copyright
- Shopping/Entertainment/Recordings/Video//DVD
- Reference/Education/Colleges_and_Universities/North_America/United_States/Alabama/Auburn_University// Libraries_and_Museums

High Tech Hits the Road
MYRON BERGER
publish date=1988-04-17
DURING the past 10 years, many high-tech gizmos and electronic gadgets have found their way into our homes. It was therefore only a matter of time before much of this gadgetry would find the travel market.

Office Space Demand Falls
Reuters
publish date=1988-03-27
The slowdown on Wall Street after last year's market slump has sharply reduced the need for new space in New York by financial firms, according to a study released yesterday.

Social Events; End of a Season
ROBERT E. TOMASSON
publish date=1988-06-19
Opera Guild Auction

Vile as Smoke
RUSSELL BAKER
publish date=1988-01-06
There was another one of those smoke fights on an airplane the other day. It's always the same, a smoker so desperate for a cigarette he's ready to start screaming in pain. Somebody willing to die rather than be subjected to cigarette smoke. Pandemonium in the sky, police meeting the plane at the gate.

DATA MEASUREMENT CORP reports earnings for Qtr to Dec 31
publish date=1988-03-26
"3*** COMPANY REPORTS ** *3*DATA MEASUREMENT CORP Qtr to Dec 31 1987 1986 Sales 3,481,263 2,252,276 Net inc 152,151 145,143 Share earns .12 .18 Shares outst 1,225,352 827,335 Yr sales 10,422,775 8,344,186 Net inc 552,307 516,063 Share earns .54 .62 Shares outst 1,019,583 827,335

Boxing; McGirt Knocks Out Davis in Round 1
PHIL BERGER
publish date=1988-08-01
James (Buddy) McGirt retained his International Boxing Federation junior welterweight title when he knocked out Howard Davis yesterday at 2 minutes 45 seconds of the first round of a scheduled 15-round bout at the Felt Forum.

When Inspiration Strikes Twice
Suzanne Slesin
publish date=1988-11-10
null

City Agency Found in Contempt in Myerson Case
ARNOLD H. LUBASCH
publish date=1988-07-01
A Federal judge held the New York City Department of Investigation in contempt of court yesterday for failing to turn over investigative files to the defense in the Bess Myerson case.

Elections Offer Only an Illusion of Participation; Making of a Primary

People
President Reagan
Vice President Bush
Per
God
George Bush
Mayor Koch
Mr. Bush
Jackson
Michael Dukakis
Jesse Jackson
City
Don
Governor Cuomo
Ronald Reagan
Sam
John

Locations
Pittsburgh
Dallas
U.S.
South Africa
N.J.
Miami
Iran
Mass.
Detroit
Conn.
Wall Street
China
HOUSTON
Italy
England
Long Island

Entities
Queens
Assistant
City Council
Reagan
Cabinet
Communist Party
CBS
Harvard
Palestine Liberation Organization
NBC
Federal Reserve
University of California
District Court
Harvard University
Salomon Brothers
Air Force

Keywords
Games
Society
Business
North America
United States
Europe
Arts
Shopping
Science
Reference
Education
United_States
North_America
Issues
United Kingdom
Recreation

Authors
ROBIN TONER
AP
Reuters
The Associated Press
Philip H. Dougherty
PHILIP H. DOUGHERTY

Fig. 3. Connections between entities for search results visualized with graph.

Not just entities, also keywords are handy to visualize connections between contents of articles to understand, discover and summarize the topics in articles. This visualization Document Atlas [4] is used to show the whole picture of search results. With that picture it is also illustrated which articles belong together in the same story or in the same topic. Based on a similarity between documents, they are mapped onto a two-dimensional plan which represents a semantic space of articles and named-entities. Articles having very similar content have coordinates closer to each other than those that are less similar [5].

This visualization is good for bigger groups of articles where it is very essential that we do not put too many information in order to keep things clear and helpful. The same applies for presenting periods of time. From that kind of visualization users can quickly guess main topics that were important during that time. In Fig. 3 on the left picture we can see entities, keywords, authors and other things connected to Princess Diana, including a picture and other information from LOD seen on upper part of the left picture and on the right picture we can see things that were important for the whole year of 1988. The left part also demonstrates the use of Document Atlas. On a magnified part of it we can see yellow dots presenting articles and keywords presenting topics of nearby articles.

3 Demonstration

Archive Explorer is designed for all types of users. On demonstration, visitors will be able to take different perspectives and try our system either as a historian, a student working on his homework for history class or just a random user curious about events in the past. We will demonstrate usability and point out that a user is not just offered with some options but is getting help to find what he wants and hopefully encouraged to read and search more about related topics. Visitors will be also provided with information about parts of system that cannot be shown on the live demonstration.

4 Conclusions and Future Work

At this point Archive Explorer is a working system providing a framework for including additional visualization and summarization techniques to better show the content hidden in the archives. One particular area for improvement is the time component of news and its visualization. Additionally, search of articles can be improved with narrowing criteria using faceted search and query suggestions.

Acknowledgments. This work was supported by the Slovenian Research Agency and the IST Programme of the EC under PASCAL2 (IST-NoE-216886), RENDER (ICT-257790-STREP) and PlanetData (ICT-NoE-257641).

References

1. Enrycher. http://enrycher.ijs.si
2. Stajner, T., Rusu, D., Dali, L., Fortuna, B., Mladenić, D., Grobelnik, M.: A service oriented framework for natural language text enrichment. Informatica **34**, 3 (2010)
3. Pajntar, B., Grobelnik, M.: SearchPoint – a new paradigm of web search. In: 17th International World Wide Web Conference (WWW 2008) Developers Track (2008). http://searchpoint.si
4. Document Atlas. http://docatlas.ijs.si
5. Fortuna, B., Mladenić, D., Grobelnik, M.: Visualization of temporal semantic spaces. In: Davies, J., et al. (eds.) Semantic Knowledge Management, pp. 155–169. Springer, Heidelberg (2008)

Semantic Content Management with Apache Stanbol

Ali Anil Sinaci(✉) and Suat Gonul

SRDC Software Research and Development and Consultancy Ltd.,
ODTU Teknokent Silikon Blok No:14, 06800 Ankara, Turkey
{anil,suat}@srdc.com.tr

Abstract. Most of the CMS platforms lack the management of semantic information about the content although a lot of research has been carried out. The IKS project has introduced a reference architecture for Semantic Content Management Systems (SCMS). The objective is to merge the latest advancements in semantic web technologies with the needs of legacy CMS platforms. Apache Stanbol is a part of this SCMS reference implementation.

Keywords: Apache Stanbol · Interactive Knowledge Stack · IKS-project · Semantic content management systems

1 Introduction

Interactive Knowledge Stack (IKS) [1] is an FP7 research project targeting to Content Management System (CMS) providers in Europe so that current CMS frameworks gain semantic capabilities. Most of the CMS technology platforms do not address *semantic* information about the content, hence lack the intelligence [2]. Therefore, today's implementations cannot provide the interaction with the content at the users's knowledge level. The objective of IKS project is to bring semantic capabilities to current CMS frameworks. IKS puts forward the "Semantic CMS Technology Stack" which merges the advances in semantic web infrastructure with the needs of European CMS industry through coherent architectures which fit into existing technology landscapes.

Apache Stanbol [3] has been created within the Apache Software Foundation to meet the requirements addressed by the IKS project for *Semantic* Content Management Systems. Apache Stanbol is an open source modular software stack and reusable set of components for semantic content management. Each component provides independent and integrated services to be used by the CMS vendors/developers. The components are implemented as OSGi [4] components based on Apache Felix [5]. Moreover, all components can be accessed via RESTful service calls.

2 Research Background and Application Context of the Demonstration

CMSs needs to deal with huge amount of unstructured data. In recent years, many studies have focused on automatic extraction of *knowledge* from unstructured

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 371–375, 2015.
DOI: 10.1007/978-3-662-46641-4_29

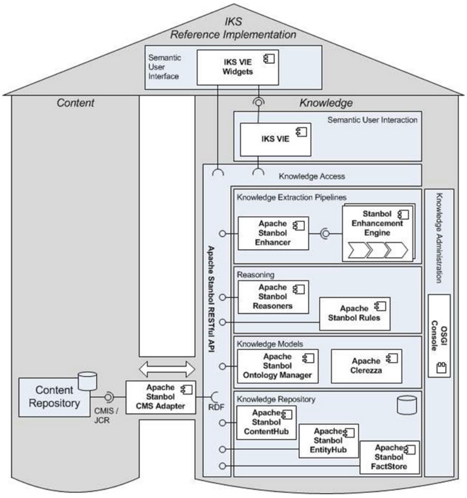


Fig. 1. SCMS reference implementation: Apache Stanbol [7]

content. Inline with this, several advancements and algorithms have been developed in the field of Information Extraction (IE) and Information Retrieval (IR). IKS project has focused on the integration of these latest technological foundations and proposed a reference architecture for Semantic Content Management Systems (SCMS) [6]. A SCMS is a CMS with the capability of extracting and managing semantic metadata of the content items through different components for specific tasks according to the layered architecture.

Apache Stanbol is the reference implementation for the Knowledge Access part of the reference architecture if a SCMS. The relation can be visualized as in Fig. 1. Several Stanbol components are implemented to serve in different layers of a SCMS.

In the application context of the demonstration, a legacy CMS (such as Apache Jackrabbit [8]) makes use of several componenets of Apache Stanbol (e.g. CMS Adapter, Enhancer) to semantify existing unstructured content and then manages the knowledge through other components (e.g. Contenthub, Entityhub). This enables intelligent content management, categorization, semantic search and powerful faceted search mechanisms, hence turns a CMS to a SCMS.

3 Key Technologies and Relation to Pre-existing Work

Components of Apache Stanbol implements latest advancements in semantic systems area and provides an integrated, comprehensive use for the users (CMS vendors/developers). The OSGi model which is adopted by Apache Stanbol supports elegant separation of different components required by the Knowledge column (Fig. 1). In addition, each component exposes its interfaces in terms of REST API.

Stanbol Enhancer implements Knowledge Extraction Pipelines through the Enhancement Engines. Enhancing the unstructured content stems from recent approaches [9] such as named-entity recognition, clustering and classification algorithms. Each Enhancement Engine processes the unstructured content (in addition to the results of other engines) and adds semantic information to the metadata of the content. Natural Language Processing [10] based engines extract valuable knowledge such as person, location and organization entities from the unstructured content. Extracted knowledge is represented in a triple-graph provided by Apache Clerezza [11] and persisted through Stanbol Contenthub.

Stanbol Entityhub is used to retrieve semantic information about the entities available through Linked Data sources such as DBpedia [12]. Independent domain ontologies can also be registered to Entityhub to create a new source for entities.

Stanbol Contenthub provides services to manage the knowledge on top of the content items. The Contenthub makes use of Apache Solr [13] as its backend to store the knowledge. Indexing through Apache Solr is performed through a number of configuration files (Solr cores). To give an example for simple semantic search, in the default index (default Solr core) of Contenthub, if a submitted document includes the keyword “Istanbul”, then the country information “Turkey” and the regional information “Marmara” are indexed along with this document as its knowledge. This leads to much more accurate search results over the knowledge of Contenthub.

LDPath [14] is a valuable outcome of Linked Media Framework (LMF) Project [15]. LDPath is a simple path based query language over RDF (similar to Xpath or SPARQL Property Paths) which is particularly designed for querying the Linked Data Cloud by following RDF links between resources. To be able to support domain specific indexing, LDPath has been integrated into the document submission and search processes of Contenthub. After LDPath integration, the semantic indexing performed by Contenthub is realized in a much smarter way so that adopters from CMS industry can enhance their legacy content with these semantic storage and search capabilities according to their specific needs.

Stanbol CMS Adapter acts as a bridge between existing unstructured content of legacy CMSs and Apahce Stanbol. CMS Adapter enables connecting content repositories through standard interfaces like JCR [16] and CMIS [17]. Furthermore it provides RESTful services to enable content repositories to submit their content models. CMS Adapter enables extracting already available semantics in the CMS into an ontology and store it in the knowledge base of Stanbol through Contenthub.

4 Demonstration and Benefits to the Audience

The demonstration presents the use of several Apache Stanbol components. A JCR based CMS connects to Stanbol through CMS Adapter and performs several semantic operations such as entity extraction from free text through Enhancement Engines, management of such entities from Linked Data cloud, management of ontologies in different formats (RDF, OWL) and management of the extracted knowledge.

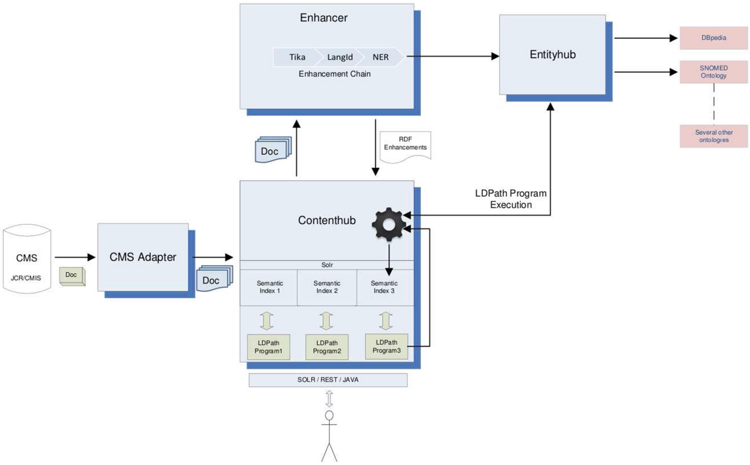


Fig. 2. Interaction between Apache Stanbol components

Since the adoption of semantic advancements are poor in the CMS arena, the demonstration provides a proof of what can be done with semantic technologies in real world. A use-case can be summarized as follows while the interaction between several Stanbol components can be followed from Fig. 2:

- A CMS (such as Apache Jackrabbit) gives the necessary information to Stanbol so that CMS Adapter can connect and retrieve the content items from the underlying JCR repository.
- CMS Adapter analyzes the underlying data model of the CMS and generates an RDF based ontology, and submits to Stanbol knowledge base.
- CMS Adapter submits all content items inside the CMS to Stanbol Contenthub.
- Contenthub submits the text-based content to Stanbol Enhancer, retrieves the enhancement results.
- Enhancer makes use of Entityhub while identifying the entities such as people (e.g. Dennis Ritchie), locations (e.g. Tokyo) and organization (e.g. European Commission). Stanbol comes with DBpedia as the default entity source. Any other ontology can also be easily registered to Stanbol system.
- Contenthub manages several *semantic* indexes on Apache Solr by means of LDPath programs. The knowledge to be indexed in the knowledge repository is extracted through the execution of LDPath.
- Contenthub provides *semantic search* on the content items. For example, if a keyword is related with an entity in a document, the document can be found even if the content does not include the keyword.
- Contenthub provides faceted search to refine search results.
- Contenthub provides a "tokenization" service for the queries. The entities are extracted from the query string and the search is directed in a more intelligent way with these tokens.
- Contenthub makes use of Wordnet, domain ontologies and referenced sites within Stanbol to suggest new query keywords to the user.

The audience learns about latest semantic technologies and more importantly be aware of their implementations. Demonstrating the capabilities of Stanbol will create such as realization that joining to the Apache Stanbol community and contributing to the implementation of latest semantic advancements is a good chance for the audience.

IKS project has an *Early Adopters* [18] programme. CMS vendors can get involved in this programme and turn their legacy CMS into a SCMS with the help of Apache Stanbol. The Early Adopters Programme provides grants to help developers evaluate and validate their software. The demonstration is a hands-on proof in this respect also; hence a CMS developer realizes the ease of integration with Apache Stanbol.

References

1. Interactive Knowledge Stack (IKS). http://www.iks-project.eu
2. Laleci, G.B., Aluc, G., Dogac, A., Sinaci, A., Kilic, O., Tuncer, F.: A semantic backend system to support content management systems. Knowl. Based Syst. J. **23**, 832–843 (2010)
3. Apache Stanbol. http://incubator.apache.org/stanbol/
4. OSGi Alliance. OSGi Service Platform - Core Service Specification Version 4.3 (2011). http://www.osgi.org/Release4/HomePage
5. Apache Felix. http://felix.apache.org
6. Christ, F., Nagel, B.: A reference architecture for semantic content management systems. In: Nttgens, M., Thomas, O., Weber, B. (eds.) Proceeding of the Enterprise Modelling and Information Systems Architectures Workshop 2011 (EMISA 2011), LNI, pp. 135–148. GI, Hamburg, Germany (2011)
7. Christ, F., Sinaci, A.A., Gonul, S.: Development of IKS Reference Architecture for Semantic Content Management Systems. Deliverable (2012). http://iks-project.googlecode.com/svn/doc/D5.0-Final
8. Apache Jackrabbit. http://jackrabbit.apache.org
9. Sarawagi, S.: Information Extraction. Found. Trends Databases **1**(3), 261–377 (2008)
10. Apache OpenNLP. http://incubator.apache.org/opennlp/
11. Apache Clerezza. http://incubator.apache.org/clerezza/
12. DBPedia. http://dbpedia.org/
13. Apache Solr. http://lucene.apache.org/solr/
14. LDPath. http://code.google.com/p/ldpath/
15. Linked Media Framework (LMF). http://kiwi-project.eu/
16. Content Repository for Java techonology API, Java Specification Request 170. http://jcp.org/en/jsr/detail?id=170
17. OASIS Content Management Interoperability Services. www.oasis-open.org/committees/cmis/
18. IKS Project Early Adopter Programme. http://www.iks-project.eu/projects/early-adopter-programme

RDFaCE-Lite: A WYSIWYM Editor for User-Friendly Semantic Text Authoring

Ali Khalili(✉) and Sören Auer

IFI/BIS/AKSW, Universität Leipzig, Johannisgasse 26, 04103 Leipzig, Germany
{khalili,auer}@informatik.uni-leipzig.de
http://aksw.org

Recently practical approaches for managing and supporting the life-cycle of semantic content on the Web of Data made quite some progress. However, the currently least developed aspect of the semantic content life-cycle is the user-friendly manual and semi-automatic creation of rich semantic content. In this demo we will present the RDFaCE-Lite editor and will show:

- how users can annotate textual content using vocabularies and named entities published on the Data Web
- how different NLP APIs can be combined in order to maximize precision and recall of the annotation process.
- how the RDFaCE-lite annotation environment can be used within existing applications such as Blogs, CMSs, etc.

RDFaCE-Lite combines WYSIWYG text authoring with the creation of rich semantic annotations. WYSIWYG text authoring is meanwhile ubiquitous on the Web and part of most content creation and management workflows. It is part of Content Management Systems, Weblogs, Wikis, fora, product data management systems and online shops, just to mention a few. Our goal with this work is to integrate the semantic annotation directly into the content creation process and to make the annotation as easy and non-intrusive as possible. The RDFaCE-Lite implementation is open-source and available for download together with an online demo at http://aksw.org/Projects/RDFaCE.

RDFaCE-Lite is developed as a plug-in for *TinyMCE Rich Text Editor* (http://tinymce.moxiecode.com). This open source HTML editor was chosen because it is very flexible to extend and is used in many popular Content Management Systems (CMS), blogs, wikis and discussion forums, etc. Therefore, by focusing efforts on this one particular editor, it is possible to quickly propagate accessible semantic content authoring practices to a number of other tools. As depicted in Fig. 1, RDFaCE-Lite provides one click text annotation by employing the following components:

NLP APIs Abstraction and Integration. Starting to annotate a document from scratch is very tedious and time consuming. There are already some Natural Language Processing (NLP) APIs available on the Web which extract specific entities and relations from the text. By using these APIs, we can provide a good starting point for further user annotations. Users then can modify and extend this automatically pre-annotated content. RDFaCE-Lite currently uses

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 376–380, 2015.
DOI: 10.1007/978-3-662-46641-4_30

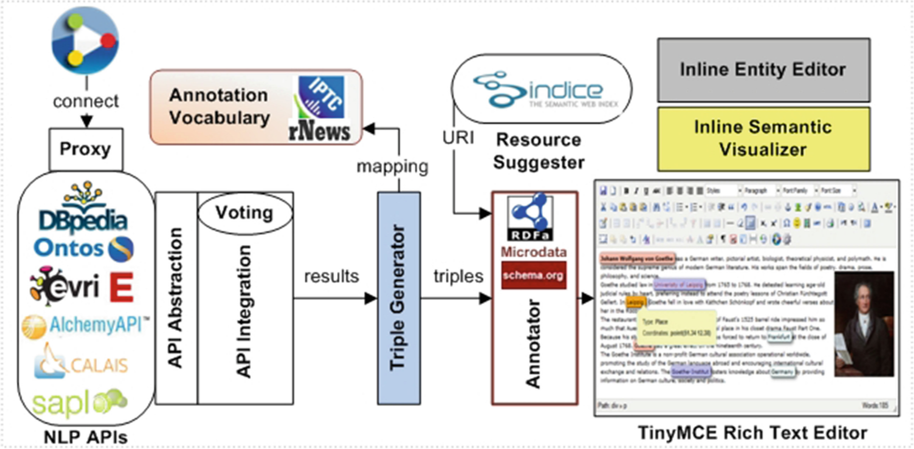


Fig. 1. RDFaCE-Lite system architecture.

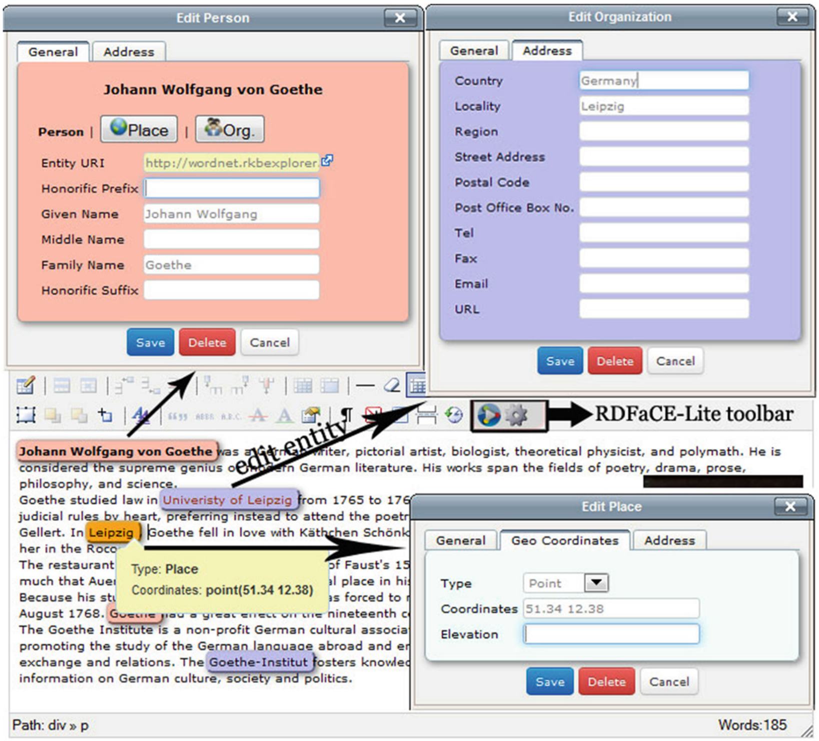
the *OpenCalais*, *Ontos*, *Alchemy*, *Extractiv*, *Evri*, *Lupedia*, *DBpedia Spotlight* and *Saplo*[1] APIs to enrich the text. Since each of these APIs use a different connecting interface as well as a different data structure for output we have implemented a *Proxy* and *API Abstraction* component. Proxy performs the connecting task by providing a separate adapter for each API. Abstraction component unifies the output of each API to a standard format[2] used by RDFaCE-Lite.

Besides annotation by each individual API, RDFaCE-Lite supports combining the results of multiple NLP APIs which yields superior performance compared to each individual (cf. http://rdface.aksw.org/samples/results.html). For this purpose, we have implemented an *API Integration* component which uses voting algorithm to integrate the results of different NLP APIs. This feature is fully configurable. Users can select their desired NLP APIs plus the number of agreements in their setting preferences.

Triple Generator. This component is responsible for generating the RDF triples to be embedded in the text. To achieve this goal, triple generator employs different existing vocabularies. In RDFaCE-Lite we use rNews 1.0 (http://dev.iptc.org/rNews) vocabulary as our annotation schema. *rNews* is a proposed standard to annotate HTML documents with news-specific metadata. rNews is proposed by International Press Telecommunications Council (IPTC) which is a consortium of the world's major news agencies, publishers and industry vendors. All the entities and properties extracted by NLP APIs are mapped to their corresponding ones in the rNews vocabulary.

[1] OpenCalais - http://www.opencalais.com, Ontos - http://www.ontos.com, Alchemy - http://www.alchemyapi.com, Extractiv - http://extractiv.com, Evri - http://www.evri.com, Lupedia - http://lupedia.ontotext.com/ and DBpedia Spotlight - http://dbpedia.org/spotlight.

[2] NLP Interchange Format (NIF) available at http://nlp2rdf.org/.



Johann Wolfgang von Goethe

Entity URI: http://wordnet.rkbexplorer
Honorific Prefix:
Given Name: Johann Wolfgang
Middle Name:
Family Name: Goethe
Honorific Suffix:

Country: Germany
Locality: Leipzig
Region:
Street Address:
Postal Code:
Post Office Box No.:
Tel:
Fax:
Email:
URL:

Johann Wolfgang von Goethe was a German writer, pictorial artist, biologist, theoretical physicist, and polymath. He is
considered the supreme genius of modern German literature. His works span the fields of poetry, drama, prose,
philosophy, and science. [redacted]
Goethe studied law in Univeristy of Leipzig from 1765 to 176
judicial rules by heart, preferring instead to attend the poetr
Gellert. In Leipzig , Goethe fell in love with Käthchen Schönk
her in the Roco
The restaurant
much that Auer
Because his stu
August 1768. Goethe
The Goethe Institute is a non-profit German cultural associa
promoting the study of the German language abroad and e
exchange and relations. The Goethe-Institut fosters knowled
information on German culture, society and politics.

Type: Place
Coordinates: point(51.34 12.38)

Type: Point
Coordinates: 51.34 12.38
Elevation:

Fig. 2. Editing entity annotations in RDFaCE-Lite.

Annotator. This component manipulates the Document Object Model (DOM) according to the generated triples. Annotator uses the resource suggester component to generate URIs for entities. Sindice (http://sindice.com) semantic search engine is employed by resource suggester. The annotator component supports *RDFa 1.1* and *Microdata* (based on Schema.org) annotation formats.

Inline Semantic Visualizer. This component provides a WYSIWYM (What-You-See-Is-What-You-Mean) view on top of the WYSIWYG (What-You-See-Is-What-You-Get) view. The WYSIWYG view is the classical interface for rich-text authoring and used by authors, journalists etc. WYSIWYG text authoring is meanwhile ubiquitous on the Web and part of most content creation and management workflows. Users authoring content are used to interact with a WYSIWYG views and there exists a wide variety of WYSIWYG editors and editing components, which can be used on the Web or offline.

The WYSIWYM view is an extension of the WYSIWYG view, which highlights named entities and other semantic information. The highlighting is realized with special CSS3 selectors for the RDFa annotations. They are thus easily configurable in terms of color borders, backgrounds etc. When pointing with the mouse on a highlighted annotation RDFaCE shows additional information concerning the particular annotation as a dynamic tooltip. RDFaCE also supports editing in the WYSIWYM view by letting a user select entities he wants to annotate and provisioning of respective annotation functionality either via the context menu or a specific form, which opens as an overlay.

Inline Entity Editor. Editing entity annotations by one click is the main task of this component. Figure 2 shows the inline editor window for Place, Person and Organization entities. Inline editor creates forms for each class of the recognized entity properties. For instance, Place entity has three categories of properties namely General, Geo-Coordinates and Address. For each of them users can fill in the values in the corresponding form.

The RDFaCE-Lite tool is very versatile and can be applied in a vast number of use cases. *Data-driven Journalism and Semantic Blogging* are two main use cases of RDFaCE-Lite. Data-driven journalism and semantic blogging deal with open data that is freely available online and analyzed with open source tools. Semantically annotated news and blog posts provided by RDFaCE-Lite facilitate a number of important aspects of information management:

- For *search and retrieval* enriching documents with semantic representations helps to create more efficient and effective search interfaces, such as faceted search or question answering.
- In *information presentation* semantically enriched documents can be used to create more sophisticated ways of flexibly visualizing information, such as by means of semantic overlays.
- For *information integration* semantically enriched documents can be used to provide unified views on heterogeneous data stored in different applications by creating composite applications such as semantic mashups.
- To realize *personalization*, semantic documents provide customized and context-specific information which better fits user needs and will result in delivering customized applications such as personalized semantic portals.
- For *reusability* and *interoperability* enriching documents with semantic representations facilitates exchanging content between disparate systems.

RDFace-Lite is published as a plug-in for WordPress blogging platform[3] thereby facilitates the semantic blogging process. Wordpress is often customized into a Content Management System (CMS) and is used by over 14 % of the 1,000,000 biggest websites (54.4 % of CMS market share) [7] which would potentially advance the promotion of semantic content even more. Furthermore, RDFaCE-Lite supports rNews standard which is getting adopted by the popular news providers such as NYTimes. Using this news-specific vocabulary to annotate entities and relationships between them will facilitate data journalism.

Regarding the related work, there are already many tools available for semantic text authoring. *WYMeditor*[4], *DataPress* [1], *Loomp* [3], *FLERSA* [4], *RDFauthor* [6] and *SAHA* 3 [2] are some examples of available tools. None of these tools

[3] Available at http://wordpress.org/extend/plugins/rdface/.

[4] http://www.wymeditor.org.

support Microdata annotation format Among the tools, RDFauthor and Loomp are adopting a similar approach as RDFaCE-Lite but do not provide any feature for automatic content annotation.

The RDFauthor approach is based on the idea of making arbitrary XHTML views with integrated RDFa annotations editable [6]. RDFauthor converts an RDFa-annotated view directly into an editable form thereby hiding the RDF and related ontology data models from novice users. The main difference between RDFaCE-Lite and RDFauthor is that RDFauthor assumes that the RDFa content is already existing while RDFaCE provides a complementary feature to create new RDFa annotations.

Loomp is another related tool representing a proof-of-concept for the *One Click Annotation* (OCA) strategy. The Web-based OCA editor allows for annotating words and phrases with references to ontology concepts and for creating relationships between annotated phrases. The main difference between Loomp and RDFaCE is that Loomp relies on the functionality of a server managing the semantic content while RDFaCE-Lite provides client-side annotation for modifying semantic content directly.

NERD [5] as a partially related work is an evaluation framework which records and analyzes ratings of Named Entity extraction and disambiguation tools. The main difference between RDFaCE-Lite and NERD is that RDFaCE-Lite employs the voting approach to combine the results of NLP APIs for automatic annotation but NERD expects a human being to manually compare the results of different NLP APIs and choose the right one for annotation. Furthermore, NERD does not focus on the annotation and authoring task but more on evaluating NLP APIs.

References

1. Benson, E., Marcus, A., Howahl, F., Karger, D.: Talking about data: sharing richly structured information through blogs and Wikis. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 48–63. Springer, Heidelberg (2010)
2. Frosterus, M., Hyvönen, E., Laitio, J.: DataFinland—a semantic portal for open and linked datasets. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 243–254. Springer, Heidelberg (2011)
3. Luczak-Roescht, R.H.M.: Linked data authoring for non-experts. In: Proceedings of the WWW 2009, Workshop Linked Data on the Web (2009)
4. Navarro-Galindo, J.L., Samos, J.: Manual and automatic semantic annotation of web documents: the FLERSA tool. In: iiWAS 2010, pp. 542–549. ACM, New York (2010)
5. Rizzo, G., Troncy, R.: NERD: a framework for evaluating named entity recognition tools in the web of data (2011)
6. Tramp, S., Heino, N., Auer, S., Frischmuth, P.: RDFauthor: employing rdfa for collaborative knowledge engineering. In: Cimiano, P., Pinto, H.S. (eds.) EKAW 2010. LNCS, vol. 6317, pp. 90–104. Springer, Heidelberg (2010)
7. W3Techs. Usage of content management systems for websites, June 2011

ParkJam: Crowdsourcing Parking Availability Information with Linked Data (Demo)

Jacek Kopecký(✉) and John Domingue

Knowledge Media Institute, The Open University, Milton Keynes, UK
{j.kopecky,j.b.domingue}@open.ac.uk

Abstract. This demo shows a mobile Android app that uses openly available geographic data and crowdsources parking availability information, in order to let its users conveniently find parking when coming to work or driving into town. The application builds on Linked Data, and publishes the crowdsourced parking availability data openly as well. Further, it integrates additional related data sources, such as events and services, to provide rich value-adding features that will act as an incentive for users to adopt the app.

1 Motivation

Managing parking in congested areas is a well-recognized problem. In the modern car-oriented world, many will experience difficulties finding parking places when driving to work or into a congested city, as discussed in [1], and drivers cruising in search of available parking can make up over 8 % of total traffic in urban areas. In [1], Shoup discusses the effects of free parking, and suggests that parking spaces should be dynamically priced at a level that would result in about 85 % utilization, with many benefits beside the improved availability. He acknowledges, however, that there are mainly political obstacles to charging for parking, and strong resistance to putting a price on previously free parking (Fig. 1).

Another approach to managing parking is to aim at improving the efficiency of the use of existing spaces, by informing drivers about available spaces, and by guiding them to alternate car parks. In some cases this is done with manually-placed “Car Park Full” signs; in better-equipped areas there are electronic systems in place. [2], a study published in 1993, reported how a real-time parking information system improved parking situation in Nottingham, England; and [3] studied the benefits of electronic parking information displays in Japan.

The figure on the right sketches a typical electronic display that shows the status of the main car parks in some town. The data displayed on such signs can easily be published online, so a user can then conveniently check it on the

The name of the app, “ParkJam”, may change when the app is released publicly, which is expected to happen before the conference. More information can be found at http://parking.kmi.open.ac.uk.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 381–386, 2015.
DOI: 10.1007/978-3-662-46641-4_31

Web or in a mobile app. SFPark, provided by the city of San Francisco, with real-time data on a number of car parks and on-street parking areas there, is an example of such an app.

Still, only a minority of car parks are monitored by electronic systems. With the growing popularity and affordability of internet-enabled smartphones, and with the wealth of data available online, especially in linked data, we can now take a step to address the parking problem in an inexpensive and efficient manner, by crowdsourcing parking availability information from drivers.

Much data about the location of car parks is available in the LinkedGeoData project, a Linked Data view on the geographical database of Open Street Map,[1] a global open collaborative mapping website. While little data on the up-to-date availability of car parks is publicly accessible online, a mobile app can make it effortless for users to contribute pieces of data (e.g. "this car park is full").

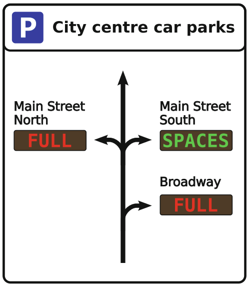


Fig. 1. Parking sign

This demo presents such a mobile app, ParkJam, in development for the Android smartphone operating system. The research focus of the ParkJam project is (1) on crowdsourcing near-real-time data, (2) on publishing such near-real-time data as Linked Open Data, and (3) on combining and using various semantic data sources and services in a mobile app. For crowdsourcing, we especially investigate how semantic data formats and the parking use case bear on the challenges listed in [4]: *How to recruit and retain users? What contributions can users make? How to combine user contributions to solve the target problem? How to evaluate users and their contributions?* Publishing near-real-time semantic data is closely related to work on semantic sensors [5], and we look into how the app, or its users, can be seen as sensors. Finally, the last aspect looks into the wider interplay of mobile client environments with the Web architecture and the access patterns for graph-structured semantic data.

2 ParkJam App Description

As shown in the screenshot in Fig. 2, the app is built around a map view that shows car parks located in the zoomed-in area, which by default follows the user's location. The app can show the availability status of the car parks, and notify the user if the availability of a watched car park changes; this will also be done with text-to-speech voice notifications, especially desirable when driving.

In a separate view, the app shows any available detailed information about the car park, such as its opening hours and pricing. Where information about car parks is missing, ParkJam users can add it, and the system will feed it back to Open Street Map.

The app allows its users to explicitly submit availability information of the currently selected car park. Additionally, to minimize the need for users to do

[1] http://linkedgeodata.org/, http://www.openstreetmap.org/.

anything, the app may also monitor conspicuous actions that imply something about availability of car parks: if a user enters a car park and quickly parks there, it is likely the car park has places available, whereas if a user drives around a car park and then moves on to another one, the first one is likely to be full.

All submissions from the users are aggregated to provide an up-to-date availability estimate for each car park. The aggregation formula must take into account the aging of information (it is seldom relevant that a car park was full six hours ago) and noisy data (submissions that look erroneous or malicious). In effect, the app crowdsources the creation and maintenance of parking location and availability data. The aggregate results are published as linked open data, to enable other third-party mashups and applications.

While the focus of ParkJam is to engage drivers, and to crowdsource parking availability data from them, we also recognize the usefulness of authoritative data sources, such as car park operators, who are encouraged to register with the app and submit their data. Any user, but especially an authoritative data provider, can make their car park availability submissions public (as a so-called "User's Data Source" — UDS); ParkJam then uses social features described below to recognize and promote authoritative sources of availability data.

Firstly, the detailed information view of a car park lists the applicable Users' Data Sources, and the user may select one(s) to trust, based on the name and other information — we recommend that authoritative sources include a phone number where users may confirm the UDS is indeed official. Reliable data sources from car park operators are likely to be trusted by many users, and such sources can be highlighted in the app to simplify discovery by new users; the app can even automatically give stronger weight to submissions from users recognized as reliable for the given car park.

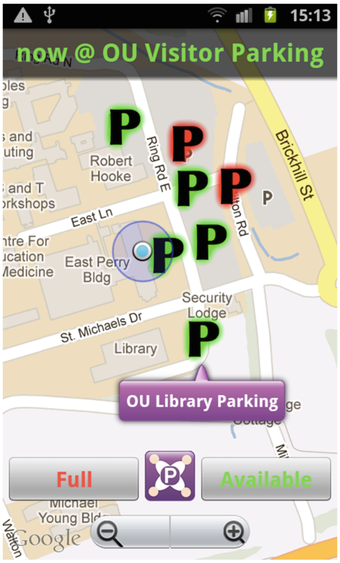


Fig. 2. ParkJam screenshot

Secondly, the app can generate a QR Code (a 2D barcode easily readable by smartphones) for a user's UDS, which the user (a car park operator) may print out and display, for instance, at the entrance to its car parks, or on Pay-and-Display machines, where drivers can scan them and readily accept as trusted.

Publishing their UDSs can also be meaningful for ordinary users/drivers. For example, a user may publish their submissions for the benefit of colleagues who happen to drive to work somewhat later, and who will be happy to know from the user what car parks are already full. As the general aggregation algorithm cannot judge a car park to be completely full after just one driver says so (partly because that would make the data prone to manipulation), marking a colleague's data source as trusted will allow a user to see the estimate of "full" early, while others may still see the car park as "nearly full".

Further, ParkJam integrates additional related data sources, such as events and services, to provide value-adding features that act as incentives for users to adopt the app. Where data is available, the detailed information view of a car park can show nearby events (which may affect parking situation in the area), and services associated with the car park, such as advance booking. These services, discovered from the registry iServe [6], will be invokable directly from the app, using the Semantic Web Service invocation engine OmniVoke [7].

3 Related Apps

There are many mobile apps that help with parking, too many to list here. We discuss two selected apps to show the novelty of our approach.

SFpark (`sfpark.org`) tracks the real-time availability status of on-street parking and parking garages in selected areas of San Francisco. Provided directly by the municipal transportation agency, the app has rich and highly-accurate data (also available to third-party developers), albeit only for a limited set of locations.

Another important app, Parkopedia (`parkopedia.com`), uses free public data and data licensed from third parties, including availability information for a small number of car parks, and even a direct booking interface for some car parks, hard-wired as the only supported type of service. Parkopedia users may submit information about car parks, with a manual review process for the submissions. The app is described as "think Wikipedia... but for parking!", but it is in fact a closed data silo — it does not make the user-submitted data freely and openly available, except through commercial licensing.

In contrast, ParkJam focuses on crowdsourcing of parking availability data from its users, which can be applied globally (but with a somewhat lower data quality), compared to the use of expensive sensor infrastructure in selected car parks in SFpark and Parkopedia; and on integration with sources of data on nearby businesses and services, particularly including generic service invocation.

4 Demo Contents

The demo starts with hands-on use of the application, on an emulator and on a smartphone device. It simulates a user driving to work at the campus of The Open University, with its 13 car parks, some of which are shown in Fig. 2. The user receives up-to-date estimates of car park availability and makes decisions on which car park to go to; then on the campus, the user submits information about car parks filling up.

Then, the demo proceeds to show additional features of the app: show detailed information about a car park, including dynamically discovered nearby events and relevant services associated with the car park, such as advance booking. Further, we show how a car park operator can publish its Users' Data Source

(UDS), and how users can find it using a QR code, or using the listing in the detailed information view. The demo highlights the effect that marking a UDS as trusted can have on the estimate of car park availability.

Finally, for interested viewers, we are prepared to show the behind-the-scenes workings of the system: the ontologies and the data sources, including how we integrate them; the architecture of the system, and the APIs and interactions between the mobile app and the back-end system.

5 Conclusions and Future Work

This demo shows ParkJam, an Android app that uses openly available geographic data and crowdsources parking availability information, in order to let its users conveniently find parking when coming to work or driving into town. The application builds on Linked Data (combining several data sources), and publishes the crowdsourced parking availability data openly as well.

As a crowdsourcing application, ParkJam must recruit users: capture their interest and give them incentives to use the app, and to submit data about car parks and their availability. The core value of the application is clear (users will spend less time looking for a parking space) but it depends on the quality and quantity of the user-submitted data. Therefore, user incentives will be a significant part of future work.

Currently, the project addresses user incentives along two axes: (1) the simplicity and efficiency of the user interface, which makes it effortless for users to submit the data that they know is valuable, while they already have the interface in front of them because they use it to look up parking availability; and (2) the added value of showing nearby events, services and businesses related to car parks. ParkJam can further integrate a number of data sources with relation to parking, such as the locations of businesses in the area, traffic and weather conditions, and even statistical information on car-related crime. By combining parking location data with business and service directories, the app can for example help the users select a car park that is near a desired business or other place of interest.

Acknowledgements. This work is funded by the project PlanetData, a European Network of Excellence on Large-Scale Data Management, http://planet-data.eu.

References

1. Shoup, D.: The High Cost of Free Parking. University of Chicago Press, Chicago (2005)
2. Khattak, A., Polak, J.: Effect of parking information on travelers' knowledge and behavior. Transportation **20**, 373–393 (1993). doi:10.1007/BF01100465
3. Asakura, Y., Kashiwadani, M.: Effects of parking availability information on system performance: a simulation model approach. In: Proceedings of Vehicle Navigation and Information Systems Conference, pp. 251–254 (1994)

4. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the world-wide web. Commun. ACM **54**, 86–96 (2011)
5. Sheth, A., Henson, C., Sahoo, S.: Semantic sensor web. IEEE Internet Comput. **12**(4), 78–83 (2008)
6. Pedrinaci, C., Liu, D., Maleshkova, M., Lambert, D., Kopecký, J., Domingue, J.: iServe: a linked services publishing platform. In: Proceedings of 1st International Workshop on Ontology Repositories and Editors for the Semantic Web, ORES 2010, colocated with 7th ESWC (2010)
7. Li, N., Pedrinaci, C., Maleshkova, M., Kopecky, J., Domingue, J.: OmniVoke: a framework for automating the invocation of Web APIs. In: ICSC 2011 Fifth IEEE International Conference on Semantic Computing (2011). http://oro.open.ac.uk/29272/

Nobody Wants to Live in a Cold City Where No Music Has Been Recorded

Analyzing Statistics with *Explain-a-LOD*

Heiko Paulheim(✉)

Knowledge Engineering Group, Technische Universität Darmstadt,
Darmstadt, Germany
paulheim@ke.tu-darmstadt.de

Abstract. While it is easy to find statistics on almost every topic, coming up with an explanation about those statistics is a much more difficult task. This demo showcases the prototype tool *Explain-a-LOD*, which uses background knowledge from DBpedia for generating possible explanations for a statistic (This demo accompanies the ESWC paper *Generating Possible Interpretations for Statistics from Linked Open Data* [1].).

1 Introduction

Every year, Mercer Research publishes a ranking of the most and the least livable cities in the world. For its current version, people in 221 cities have been interviewed and asked for the perceived quality of living in their city[1].

Statistics like these are widely spread and frequently cited, e.g., in the newspapers. However, what we are typically interested in is asking: *why* are the values in a particular statistics the way they are? Looking at the Mercer example, a typical question would be: *What is it that makes Vienna (which is at the top position) more livable than, e.g., Dubai (which is on position 74)?*

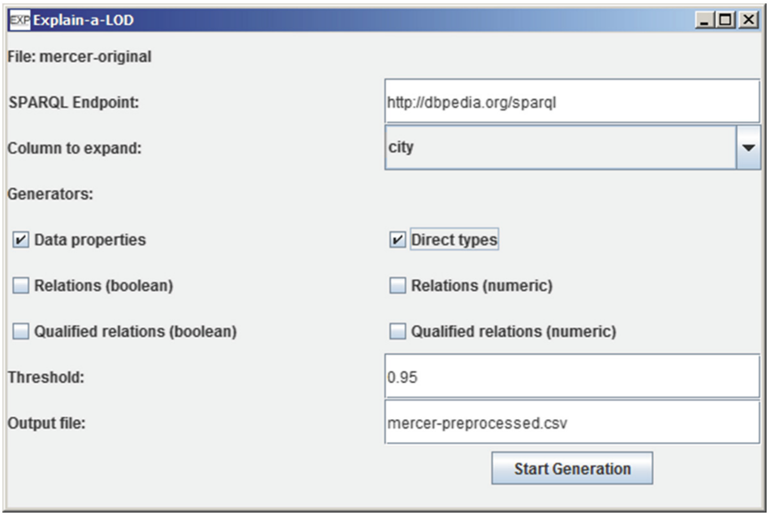
In order to come up with hypotheses for answering such questions, *background knowledge*, i.e., more information about the cities, is required. Factors that could be of interest for explaining the Mercer statistic could be dealing with the climate, the economy, the cultural live, the population density, and so on. Therefore, the first task is to enhance the statistics file at hand with more background information. Once this is done, tools for correlation analysis can be run on the enhanced file for finding possible hypotheses.

Compiling that background knowledge manually is a labour-intensive task, and it is prone to a priori biases – since we have an initial feeling for which information could be relevant, we are likely to include some pieces information and discard others. Thus, an automatic system for compiling the background information would be desirable. Explain-a-LOD, the tool introduced in this demo[2], uses data sources in Linked Open Data [2] for adding background knowledge to a statistic in a fully automated manner.

[1] http://www.mercer.com/articles/quality-of-living-survey-report-2011.

[2] http://www.ke.tu-darmstadt.de/resources/explain-a-lod.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 387–391, 2015.
DOI: 10.1007/978-3-662-46641-4_32



File: mercer-original

SPARQL Endpoint: http://dbpedia.org/sparql

Column to expand: city

Generators:

☑ Data properties ☑ Direct types

□ Relations (boolean) □ Relations (numeric)

□ Qualified relations (boolean) □ Qualified relations (numeric)

Threshold: 0.95

Output file: mercer-preprocessed.csv

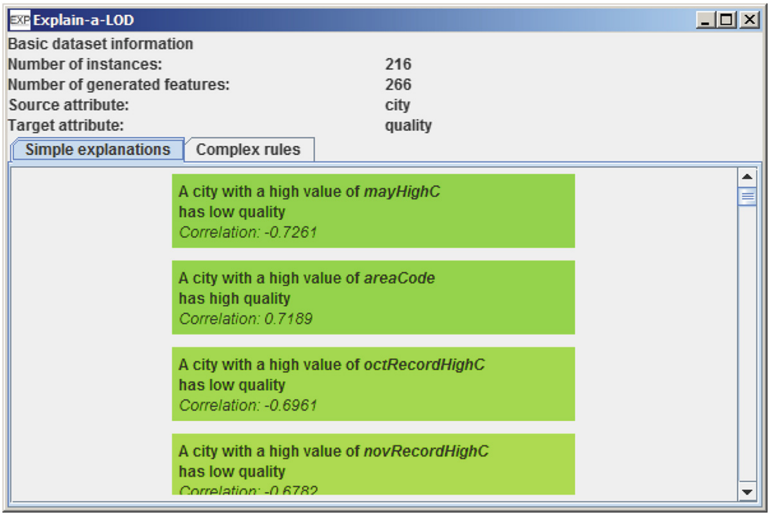
Fig. 1. Preprocessing a statistics file

2 The Explain-a-LOD Workflow

Statistics are most often tables, thus, the workflow of Explain-a-LOD starts with such a table, e.g., a CSV file. The user can import such a file, specify a column name which contains the entites to gather background information for (e.g., a column with city names), and select a couple of generators and a relevance threshold for the newly generated columns, as shown in Fig. 1. The preprocessed file may also be stored for later use.

Different generators are available for adding background information (see [3] for details):

- Data attributes can be added for all datatype properties. For example, a column *population* is introduced in each row of the cities statistics, which reflects the value of the `dbpedia:population` value of the respective entity.
- Direct types can be added as boolean columns. For example, the column `yago:EuropeanCapitals` is added with value *true* for Vienna, and with value *false* for Dubai.
- Incoming and outgoing relations can be added either as boolean or numeric columns. For example, if there are any albums recorded in a city, i.e., there are incoming relations of the type `recordedIn`, the column `recordedIn_in` is filled with *true* or a positive number, with *false* or zero otherwise.
- Qualified relations may also be added, taking into account the type of the related object. For example, since Vienna is the headquarter of the organization *OPEC*, a boolean or numeric column *headquarter_in_Organization* can be introduced, depicting whether the city is a `dbpedia:headquarter` of any `dbpedia:Organisation`, or the number of such organizations, respectively.



Basic dataset information

Number of instances:	216
Number of generated features:	266
Source attribute:	city
Target attribute:	quality

A city with a high value of *mayHighC*
has low quality
Correlation: -0.7261

A city with a high value of *areaCode*
has high quality
Correlation: 0.7189

A city with a high value of *octRecordHighC*
has low quality
Correlation: -0.6961

A city with a high value of *novRecordHighC*
has low quality
Correlation: -0.6782

Fig. 2. Hypotheses generated for a statistics file

Once that additional data is added to the original dataset, Explain-a-LOD will start analyzing the data and try to formulate hypotheses. Two strategies are used: simple correlations are sought by analyzing the correlation coefficient between each generated column and the statistic's target value (such as the quality of living score in the Mercer example), and different rule learners are run on the dataset for formulating more complex hypotheses, using the *Weka* machine learning framework [4].

The hypotheses found are presented to the user in two lists, using color codings for the machine's confidence in those hypotheses (the correlation coefficient or the confidence of a rule, respectively), as shown in Fig. 2. The colors range from green (high confidence) to red (low confidence).

3 Example Hypotheses

We have created a set of hypotheses, using the different generation strategies discussed above, and have had them rated in the form a questionnaire in a user study (see [1] for details on the user study). The top-rated hypotheses were[3]:

1. Cities where many things take place have a high quality of living.
2. European capitals of culture have a high quality of living.
3. African capitals have a low quality of living.
4. Host cities of olympic summer games have a high quality of living.
5. Cities where at least 73 things are located have a high quality of living.

[3] The full list of hypotheses and their ratings can be found at http://www.ke.tu-darmstadt.de/resources/explain-a-lod/user-study.

The first and the last hypothesis have been generated by exploiting unqualified relations, while the second, third, and fourth have been generated from direct types (e.g., *YAGO*, which is used for types in DBpedia, defines types such as *EuropeanCapitalsOfCulture* or *HostCitiesOfOlympicSummerGames*). The last hypothesis has been generated by a rule learning algorithm, which cannot only find a correlation between an attribute and the target, but also an optimal point for splitting the dataset into positive and negative examples (i.e., high and low quality cities).

While many of the hypotheses generated make sense to the users, the tool also produces some not-so well perceived hypotheses. Examples include:

1. Cities with a large latitude have a high quality of living.
2. Cities where many bands founded in 2004 originate have a high quality of living.
3. Cities where nothing has been recorded and where the maximum temperature in January does not exceed 16°C have a low quality of living.

Those examples point at challenging problems with the approach. The first hypothesis shows that the tool often cannot verbalize a hypothesis in a way that satisfies the end user. In fact, the latitude of a city is a good indicator for separating cities into cities in the first world and cities in the third world. It can be assumed that the rating for a re-formulated hypothesis like *First world cities have a high quality of living* would have been much higher, but the tool cannot detect which of the two variants will be more plausible to the user.

The second example points to a problem with DBpedia: it has a strong bias towards popular culture, especially Northern American and European popular culture. Thus, hypotheses with references to popular culture appear quite frequently, although they are in many cases not plausible. Due to that bias, that hypotheses mainly refers to the Northern American and European countries.

The third example also points to the bias problem, but also includes another challenge: at the moment, the tool is not capable of generating hypotheses that are coherent in themselves. For example, cultural life (expressed in music recorded in a city) and climate (expressed in the January temperature) may both influence the quality of living in a city, but in the users' perception, they are not interrelated. Thus, such hypotheses are ranked very low by users, although they may be quite accurate. Finding and implementing metrics for coherent rules could help remedying this problem.

4 Conclusion and Future Work

In this demo, we have introduced the *Explain-a-LOD*, which uses background information from Linked Open Data for enriching statistics, and which is capable of coming up with hypotheses for explaining a statistic in a fully automated way.

The tool and the hypotheses it creates have been tested with a larger number of users. In this paper, we have shown examples both for high and low ranked hypotheses, and discussed some reasons that lead to the generation of the latter.

While Explain-a-LOD is currently a prototype which can be used on various statistics datasets, there are many interesting research questions. Some of those have already been touched by the examples above: using data from the semantic web, dealing with biases, incompleteness and faultiness of data is an issue. Separating useful from useless, plausible from implausible hypotheses is also an issue which cannot be addressed trivially. Further research problems cover issues such as scalability, especially when using more complex generation strategies, or producing an intuitive verbalization and visualization of hypotheses.

Current plans of extending Explain-a-LOD include the combination of different datasets. For many statistical datasets, sources such as World Fact Book or Eurostat may be ideal candidates for generating background knowledge, while for others, such as statistics about the box office revenue of films, specialized data sets such as Linked Movie Database might be more suitable. Picking relevant datasets fully automatically for different statistics would be desirable, but requires some more in-depth research. Using table extraction mechanisms could be a way to also include tabular data from non-LOD sources.

In summary, Explain-a-LOD showcases an approach employing Linked Open Data for a use case which has not been addressed much in the past. The results prove that the approach is feasible and open up a number of interesting research questions. During the demo, the visitors will be able to try the prototype with different datasets by themselves.

Acknowledgements. This work was supported by the German Science Foundation (DFG) project FU 580/2 “Towards a Synthesis of Local and Global Pattern Induction (GLocSyn)”.

References

1. Paulheim, H.: Generating possible interpretations for statistics from linked open data. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 560–574. Springer, Heidelberg (2012)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
3. Paulheim, H., Fünkranz, J.: Unsupervised generation of data mining features from linked open data. In: International Conference on Web Intelligence, Mining, and Semantics (WIMS 2012) (2012)
4. Bouckaert, R.R., Frank, E., Hall, M., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: WEKA – Experiences with a Java open-source project. J. Mach. Learn. Res. **11**, 2533–2541 (2010)

ScienceWISE: A Web-Based Interactive Semantic Platform for Paper Annotation and Ontology Editing

Anton Astafiev[3](✉), Roman Prokofyev[4], Christophe Guéret[6],
Alexey Boyarsky[1,2,3], and Oleg Ruchayskiy[5]

[1] Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
alexey.boyarsky@epfl.ch
[2] Instituut-Lorentz for Theoretical Physics, Universiteit Leiden, Leiden, The Netherlands
[3] Bogolyubov Institute for Theoretical Physics, Kiev, Ukraine
anton.astafiev@cern.ch
[4] EXascale Infolab, University of Fribourg, Fribourg, Switzerland
roman.prokofyev@unifr.ch
[5] CERN TH-Division, PH-TH, Geneva 23, Switzerland
oleg.ruchayskiy@cern.ch
[6] Vrije Universiteit, Amsterdam, The Netherlands
c.d.m.gueret@vu.nl

Abstract. The ScienceWISE system is a collaborative ontology editor and paper annotation tool designed to help researchers in their discovery. In this paper, we describe the system currently deployed at sciencewise.info and the exposition of its data as Linked Data. During the "RDFization" process, we faced issues to encode the knowledge base in SKOS and find resources to link to on the LOD. We discuss these issues and the remaining open challenges to implement some target features.

1 Introduction

Organizing scientific knowledge in systematic ways becomes increasingly important. However, the creation of intra- and inter-disciplinary knowledge bases is hindered by the heterogeneity and the scale of the information to consider. This calls for *scientific community-run systems* (replacing classical publishers of encyclopedias) allowing to combine presentation of new results, in-depth discussions, "user-friendly" introductions for young scientists, and meta-data to relate semantically similar concepts or pieces of content. Today, there are no standard tools to insert, store and query such meta-data online, which mostly remains "in the heads of the experts". To address these pertinent issues and make a first step towards the creation of tools for the automated support of the scientific process, a group of physicists from EPFL and CERN together with computer scientists from EPFL, the University of Fribourg and VU created the *ScienceWISE system — a system for semantically importing, storing and searching scientific data.*

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 392–396, 2015.
DOI: 10.1007/978-3-662-46641-4_33

ScienceWISE[1] allows a community of scientists, working in a specific domain, to generate dynamically as part of their daily work an *interactive semantic environment*, *i.e.*, a field-specific ontology with direct connections to the text of research papers. ScienceWISE xphysicists and is connected with the ArXiv.org archive of papers. It is however designed not to be field specific and can be re-deployed to be used by other scientific communities. In the following, we will use ScienceWISE to refer to both the system and its deployed version at http://ScienceWISE.info.

The rest of the paper describes ScienceWISE general (Sect. 2) and the exposition of its data as Linked Open Data (Sect. 3). We conclude describing the problems faced while looking at exposing the data from ScienceWISE as Linked Data and sketch paths for future work (Sect. 4).

2 Architecture of ScienceWISE

The ScienceWISE system is an eco-system currently compromising three type of entities. The **Ontology**: the knowledge-graphs which captures the concepts and their complex relationships; The **Users**: the social community of experts in a field which give local, noisy and incomplete knowledge on some parts of the ontology; The **Portal**: the web application which consolidates all local inputs with the current ontology and attempts to create a comprehensive, global and dynamic knowledge system. There are plans to add a fourth item to the list [1]: an intelligent assistant able to leverage the knowledge expressed in the Ontology and assist the Users in their research activities. The exposition of the data from ScienceWISE as Linked Data (see Sect. 3) is a first step in this direction.

ScienceWISE Ontology. The ontology used to tag the papers is enriched and curated the users of the system. To create the initial version of the ontology, we have performed a semi-automated import from many science-oriented ontologies and online encyclopedias. After this initial step, ScienceWISE users (who are domain experts) are allowed to edit elements of the ontology (*e.g.*, adding new definitions or new relations) in order to improve its quality. Presently, the ScienceWISE ontology counts more than 60 000 unique entries, each with its own definitions, alternative forms, and semantic relations to other entries. The semantic relations are both of general (e.g., *is a part of*) and field-specific (*is a model of*, *is observed in*) nature.

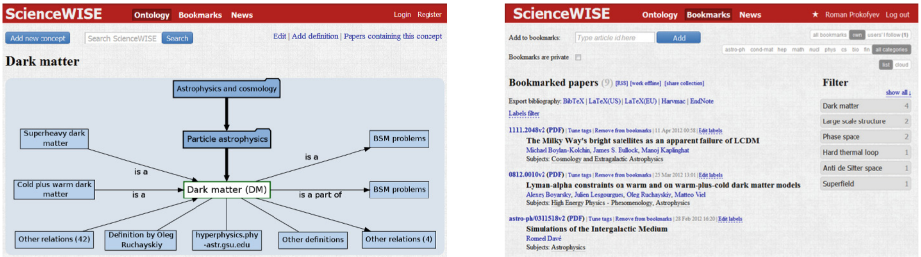
ScienceWISE Users. The system is public since April 2011 and accessible by scientists via ArXiv.org as well as via the CERN Document Server[2] and Inspire[3]. The system currently counts above 200 active users, thousands of conceptually indexed and annotated papers, and is now receiving several new registrations *daily*.

[1] Accessible at http://ScienceWISE.info.

[2] http://cds.cern.ch.

[3] http://inspirebeta.net: comprehensive bibliographic database in high-energy.

ScienceWISE Portal. The ScienceWISE portal is the main interface for interacting with the system. The ontology explorer (see Fig. 1a) shows information about concepts. Competing scientific viewpoints about the same concept are represented as alternative resources and definitions. In the tagging interface (see Fig. 1b) a user is presented with a automatically populated list of relevant concepts to pick from to annotate the paper. There is also the possibility to create and explore collection of papers. The portal is implemented in Python and uses PostgreSQL as a data back-end. It uses TeXpp[4] to browse the content of LaTeXfiles and spot concepts from the ontology.



(a) Representation of the ontology

Bookmarked papers (9) [RSS] [work offline] [share collection]

Export bibliography: BibTeX | LaTeX(US) | LaTeX(EU) | Harvmac | EndNote

Labels filter

1111.2048v2 (PDF) | Tune tags | Remove from bookmarks | 11 Apr 2012 00:58 | Edit labels
The Milky Way's bright satellites as an apparent failure of LCDM
Michael Boylan-Kolchin, James S. Bullock, Manoj Kaplinghat
Subjects: Cosmology and Extragalactic Astrophysics

0812.0010v2 (PDF) | Tune tags | Remove from bookmarks | 25 Mar 2012 13:01 | Edit labels
Lyman-alpha constraints on warm and on warm-plus-cold dark matter models
Alexey Boyarsky, Julien Lesgourgues, Oleg Ruchayskiy, Matteo Viel
Subjects: High Energy Physics - Phenomenology, Astrophysics

astro-ph/0311518v2 (PDF) | Tune tags | Remove from bookmarks | 28 Feb 2012 16:20 | Edit labels
Simulations of the Intergalactic Medium
Romeel Davé
Subjects: Astrophysics

Filter	
Dark matter	4
Large scale structure	2
Phase space	2
Hard thermal loop	1
Anti de Sitter space	1
Superfield	1

(b) Paper tagging interface

Fig. 1. Two screen captures of the ScienceWISE portal

3 ScienceWISE and Semantic Web Technologies

Already now ScienceWISE enable the construction of a highly-structured, collaboratively curated and expressive ontology. However, *the ultimate goal of our system* is to create a performant, user-friendly and customizable integrated environment to help scientists save time and effort in their daily work, while building complex ontological networks that capture their scientific findings [1]. This knowledge acquisition process will be conducted in a pervasive way, "in the background", harvesting data from different source and combining it with the ScienceWISE ontology.

The need for smooth data integration capabilities with other data sets on the web and the necessity for reasoning processes able to help scientists drove us toward considering the usage Semantic Web technologies. The Semantic Web enhanced version of ScienceWISE is depicted on Fig. 2.

"RDFization". To leverage on the existing Semantic Web technologies (deductive reasoning, relation finding, ontology matching) we have performed an "RDFization" of the ScienceWISE ontology. D2R periodically exports the content from the relational database as RDF and pushes it to an OWLIM triple store. The vocabulary used is mainly a combination of SKOS, RDF and RDFS.

[4] http://code.google.com/p/texpp/.

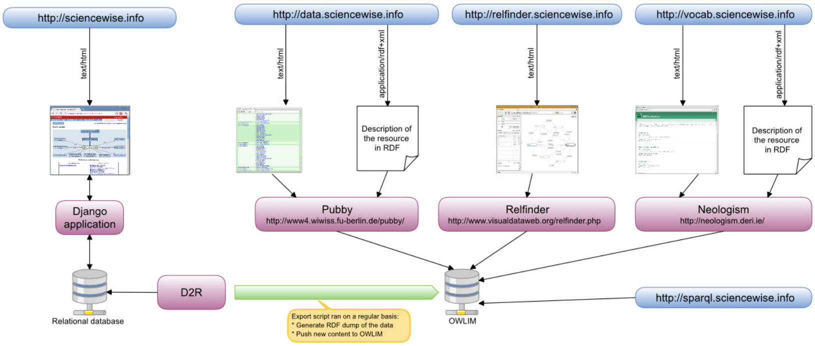
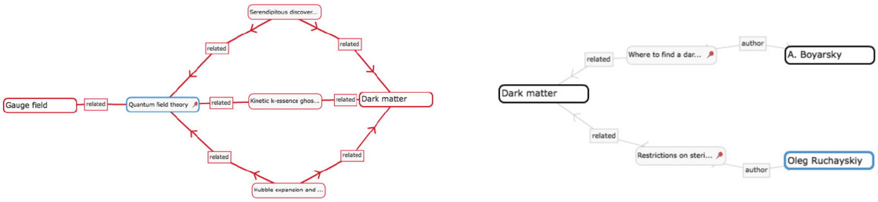

http://sciencewise.info
text/html
Django application
Relational database
D2R
http://data.sciencewise.info
text/html
application/rdf+xml
Description of the resource in RDF
Pubby
http://www4.wiwiss.fu-berlin.de/pubby/
http://relfinder.sciencewise.info
text/html
Relfinder
http://www.visualdataweb.org/relfinder.php
http://vocab.sciencewise.info
text/html
application/rdf+xml
Description of the resource in RDF
Neologism
http://neologism.deri.ie/
OWLIM
http://sparql.sciencewise.info
Export script ran on a regular basis:
* Generate RDF dump of the data
* Push new content to OWLIM

Fig. 2. Global architecture of ScienceWISE showing the "RDFization" of the ScienceWISE data. The data exposed describe papers, concepts and authors

However, some of the field-specific relations (such as "is a mechanism of") can not be directly mapped to SKOS so we created a vocabulary which extends SKOS to match our needs. This vocabulary is published using Neologism and is available at http://vocab.sciencewise.info/ontology. Finally, Pubby is used to serve de-referencable URIs for the resources at http://data.sciencewise.info/.

Initial Outcomes. Having the data from ScienceWISE transcoded in a graph-db enables finding non trivial paths between the different nodes of this graph. In ScienceWISE, the nodes are papers, concepts and authors. We deployed RelFinder[5] on top the SPARQL end point to rapidly obtain a tool able to find and display these paths (*c.f.* Fig. 3). We have extended it by adding a possibility to ignore some nodes, defined via configuration file, and have plans to extend it further more.



(a) Scientific concepts ("*Dark matter*" and "*Gauge field*") related through their co-occurrence in research papers

(b) Authors who did not co-author but had written about the same subject ("*Dark Matter*")

Fig. 3. Finding relation between entities of different nature with Relfinder

[5] http://www.visualdataweb.org/relfinder.php.

4 Current Challenges and Open Questions

The publication of the data from ScienceWISE as Linked Data and the usage of Relfinder are a first step and a number goals can reached if we are able to utilize and extend beyond-the-state-of-the-art existing Semantic Web technologies. However we are facing some major issues that slow us down or block some aspects of the development:

Semantic structures: The level of abstraction of scientific concepts is highly non-trivial. For most of the concepts (apart from the "named entities": proteins, particles, celestial objects) it is typical to be an *instance* of one class and a *subclass* of the other class at the same time. Extensions of SKOS and reasoners aware of this are desired;

Resource discovery in LOD: One of the main problems, that we have encountered in bringing the ScienceWISE data to LOD was the absence of any "browsing" capabilities for resources in the LOD cloud (beyond clickable version of the picture[6] and some basic unstructured tags). Without an easy way to find re-usable URIs we had to mint URIs for entities which are actually not described in the system.

Resource matching: The matching between resources from various data sources and those from ScienceWISE is important. Our attempts at using semi-automated tools like SiLK resulted in too many false positive, even with a conservative strategy. We need more flexible matching tool and we need to integrate them within the Portal to ensure human validation of the links.

ScienceWISE is a working system that is being used by a growing amount of Physicists to annotate paper, discuss related concepts and express diverging opinions. In order to further develop the capabilities of the system and share its data, we have started using Semantic Web technologies and applied Linked Data publication principles. Our first results are promising, showing already some added value, but are limited by a number of problems and challenges we are facing. We described them in the paper and a path for future work and a call for guidance from the Semantic Web research community.

Acknowledgments. This project is carried out as part of the "AAA/SWITCH–e-infrastructure for e-science" program under the leadership of the Swiss National Research and Education Network. It has been supported (in part) by funds from the ETH Board, the Swiss National Science Foundation under grant number PP00P2_128459, and the European Community's Seventh Framework Programme (FP7/20072013) under Grant Agreement n^{o}256975, LOD Around—The Clock (LATC) Support Action.

Reference

1. Aberer, K., Boyarsky, A., Cudré-Mauroux, P., Demartini, G., Ruchayskiy, O.: ScienceWISE: a web-based interactive semantic platform for scientific collaboration. In: Proceedings of ISWC2011 - "Outrageous ideas" track (2011)

[6] http://richard.cyganiak.de/2007/10/lod/imagemap.html.

Developing an Incomplete Reasoner in Five Minutes: The Large Knowledge Collider in Action

Alexey Cheptsov(✉)

High-Performance Computing Center Stuttgart,
Nobelstr. 19, 70569 Stuttgart, Germany
cheptsov@hlrs.de

Abstract. The Large Knowledge Collider (LarKC) is a prominent development platform for the Semantic Web reasoning applications. Guided by the preliminary goal to facilitate the incomplete reasoning, LarKC has evolved in a unique platform, which can be used for the development of robust, flexible, and efficient semantic web applications, also leveraging the modern grid and cloud resources. As a reaction on the numerous requests coming from the tremendously increasing user community of LarKC, we set up a demonstration package for LarKC that is intended to present the main subsystems, development tools and graphical user interfaces of LarKC. The demo aims for both early adopters and experienced users and serves the purpose of promoting Semantic Web Reasoning and LarKC technologies to the potentially new user communities.

Keywords: Larkc · Reasoning · Workflow

1 Introduction

Development of the internet-scale data-centric applications has been recognized as the primary challenger in the Semantic Web community. As a reaction on this challenge, several leading Semantic Web research organizations and technological companies have joined their efforts around the project of the Large Knowledge Collider (LarKC), supported by the European Commission. The LarKC's main value is twofold. On the one hand, it enables a new approach for large-scale reasoning based on the technique for interleaving the identification, the selection, and the reasoning phases. On the other hand, through over the project's life time (2008–2011), LarKC has evolved in an outstanding, service-oriented platform for creating very flexible but extremely powerful applications, based on the plug-in's realization concept. The LarKC plug-in marketplace has already comprised several tens of freely available plug-ins, which implement new know-how solutions or wrap existing software components to offer their functionality to a much wider range of applications as even originally envisioned by their developers. LarKC is an open source development, which encourages collaborative application development for Semantic Web. Despite being quite a young solution, LarKC has already established itself as a very promising technology in the Semantic Web world. Some evidence of its value was a series of Europe- and world-wide

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 397–401, 2015.
DOI: 10.1007/978-3-662-46641-4_34

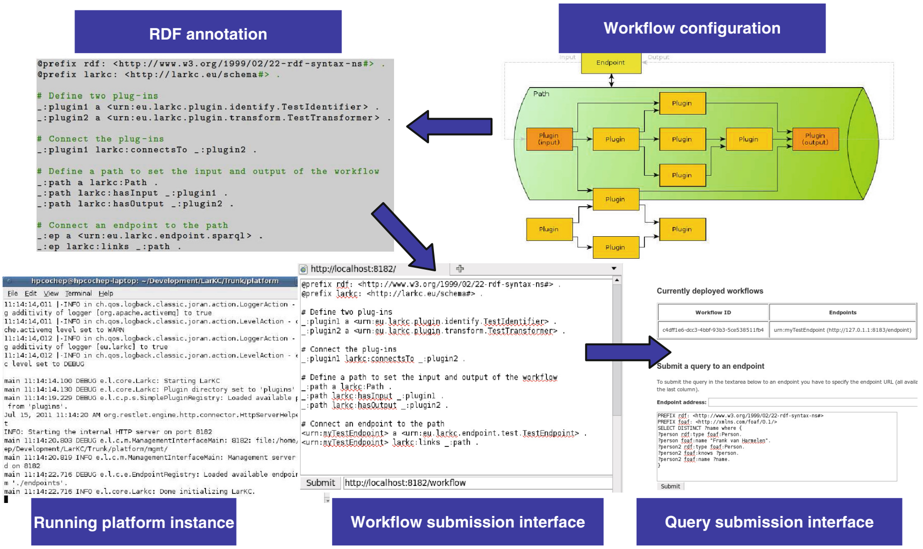
Semantic Web challenges won by the LarKC applications. As the most successful LarKC-based developments can be referred Bottari – the Semantic Challenge winner in 2011, and WebPIE – the Billion Triple Challenge winner in 2010.

In order to promote the reasoning ideas and practical solutions of LarKC to new development communities, we set up a demonstration package that is guiding the early adopter through the main steps towards creation of a scalable reasoning application. We believe that our demo will allow the LarKC-based service-oriented reasoning technology to be promoted to several new user and developer communities.

2 Contents of the Demo

The demo will guide the user through the main steps towards creation of an exemplarily LarKC application (Fig. 1). The application will be used to answer queries executed against the Linked Life Data, Sindice, and other chosen RDF knowledge bases. The main steps include:

1. Downloading and getting started with the LarKC platform.
2. Browsing the LarKC Market Place and identification of the needed plug-ins.
3. Creation and execution of a simple workflow based on the identified plug-ins with the Workflow Designer application.
4. Making the basic workflow more complicated by merging and splitting the application data flow (e.g. merging results acquired from multiple data bases).


RDF annotation
Workflow configuration
Input
Endpoint
Output
Path
Plugin (input)
Plugin
Plugin (output)
Running platform instance
Workflow submission interface
Query submission interface
hpcochep@hpcochep-laptop: ~/Development/LarKC/Trunk/platform
File Edit View Terminal Help
http://localhost:8182/
Submit

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define two plug-ins
_:plugin1 a <urn:eu.larkc.plugin.identify.TestIdentifier> .
_:plugin2 a <urn:eu.larkc.plugin.transform.TestTransformer> .

# Connect the plug-ins
_:plugin1 larkc:connectsTo _:plugin2 .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin2 .

# Connect an endpoint to the path
_:ep a <urn:eu.larkc.endpoint.sparql> .
_:ep larkc:links _:path .
```

```
11:14:14,011 |-INFO in ch.qos.logback.classic.joran.action.LoggerAction -
g additivity of logger [org.apache.activemq] to true
11:14:14,011 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - c
che.activemq level set to WARN
11:14:14,012 |-INFO in ch.qos.logback.classic.joran.action.LoggerAction -
g additivity of logger [eu.larkc] to true
11:14:14,012 |-INFO in ch.qos.logback.classic.joran.action.LevelAction - e
c level set to DEBUG

main 11:14:14.100 DEBUG e.l.core.Larkc: Starting LarKC
main 11:14:14.130 DEBUG e.l.core.Larkc: Plugin directory set to 'plugins'
main 11:14:19.229 DEBUG e.l.c.p.s.SimplePluginRegistry: Loaded available p
 from 'plugins'.
Jul 15, 2011 11:14:20 AM org.restlet.engine.http.connector.HttpServerHelpe
t
INFO: Starting the internal HTTP server on port 8182
main 11:14:20.803 DEBUG e.l.c.m.ManagementInterfaceMain: 8182: file:/home/
ep/Development/LarKC/Trunk/platform/mgmt/
main 11:14:20.819 INFO e.l.c.m.ManagementInterfaceMain: Management server
d on 8182
main 11:14:22.716 DEBUG e.l.c.e.EndpointRegistry: Loaded available endpoir
m './endpoints'.
main 11:14:22.716 INFO e.l.core.Larkc: Done initializing LarKC.
```

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define two plug-ins
_:plugin1 a <urn:eu.larkc.plugin.identify.TestIdentifier> .
_:plugin2 a <urn:eu.larkc.plugin.transform.TestTransformer> .

# Connect the plug-ins
_:plugin1 larkc:connectsTo _:plugin2 .

# Define a path to set the input and output of the workflow
_:path a larkc:Path .
_:path larkc:hasInput _:plugin1 .
_:path larkc:hasOutput _:plugin2 .

# Connect an endpoint to the path
<urn:myTestEndpoint> a <urn:eu.larkc.endpoint.test.TestEndpoint> .
<urn:myTestEndpoint> larkc:links _:path .
```

http://localhost:8182/workflow

Currently deployed workflows

Workflow ID	Endpoints
c4dff1e6-dcc3-4bbf-93b3-5ce538511fb4	urn:myTestEndpoint (http://127.0.1.1:8183/endpoint)

Submit a query to an endpoint

To submit the query in the textarea below to an endpoint you have to specify the endpoint URL (all availa the last column).

Endpoint address:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?name where {
?person rdf:type foaf:Person.
?person foaf:name "Frank van Harmelen".
?person2 rdf:type foaf:Person.
?person2 foaf:knows ?person.
?person2 foaf:name ?name.
}
```

Fig. 1. Life-cycle of a LarKC reasoning application.

3 Getting Started with LarKC

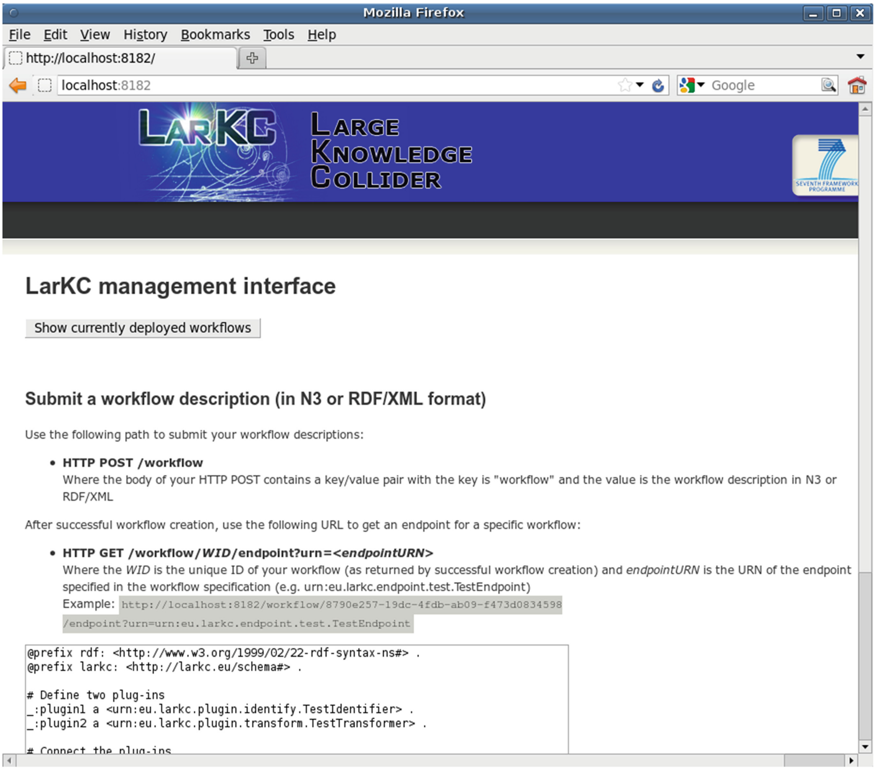
LarKC is an open-source software architecture available for downloading at SourceForge.[1] The contents of the newest LarKC-release contains the following components:

- Platform (/platform) - the core of LarKC, a software infrastructure and run-time environment that enables flexible development and serves a deployment environment for the LarKC plug-ins (see more details below).
- Exemplary collection of the LarKC Plug-Ins (/paltform/plugins) - self-contained, loosely-coupled and functionally-interoperable modules grouped in the following categories (according to the functionality, please refer to the documentation for more details), including:
  - LLDReasoner - executes a SPARQL query against Linked Life Data repository
  - SOStoVBtransformer - used for transformation of a SetOfStatements to the VariableBinding representation
- Extra tools
  - Workflow Designer - a tool for visual workflow construction (/extra_tools/workflow_designer)
  - Plug-in Wizard - a tool for creating new plug-ins in the Eclipse environment (/extra_tools/eclipse_wizard)
  - Archetypes for plug-in and end-point development with Maven (/extra_tools/larkc-plugin[endpoint]-archetype)
- Exemplary workflows (/workflows)
  - LLD_Reasoning - executes a SPARQL query against the Linked Life Data repository
  - User and developer manual (/doc)

Once the LarKC platform has been installed properly, it can be started by executing a special start-up script provided in the release package. In case of the successful start, the platform is ready for submission and execution of workflows. This is performed by means of a Management Interface – a web service based frontend to the LarkC platform (Fig. 2). In order to access the Management Interface, the user only needs to point the browser to the local link http://localhost:8182/. Using the Management Interface, the user can submit new workflows (see [1] for more information about the LarKC workflows and corresponding platform's services) and execute queries against the previously submitted workflows.

The user can execute one of the basic workflows provided with the LarKC installation, e.g. the LLD_Reasoning one. This can be done by copying the annotated worklow's text into the Management Interface and submitting it. In case of the successfully workflow's deployment, the newly-submitted workflow should appear in the Management Interface's list of the submitted workflows.

[1] http://sourceforge.net/projects/larkc/.


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8182/
localhost:8182
Google
LARGE
KNOWLEDGE
COLLIDER
SEVENTH FRAMEWORK PROGRAMME
Show currently deployed workflows

LarKC management interface

Submit a workflow description (in N3 or RDF/XML format)

Use the following path to submit your workflow descriptions:

- **HTTP POST /workflow**
  Where the body of your HTTP POST contains a key/value pair with the key is "workflow" and the value is the workflow description in N3 or RDF/XML

After successful workflow creation, use the following URL to get an endpoint for a specific workflow:

- **HTTP GET /workflow/*WID*/endpoint?urn=<*endpointURN*>**
  Where the *WID* is the unique ID of your workflow (as returned by successful workflow creation) and *endpointURN* is the URN of the endpoint specified in the workflow specification (e.g. urn:eu.larkc.endpoint.test.TestEndpoint)
  Example: `http://localhost:8182/workflow/8790e257-19dc-4fdb-ab09-f473d0834598/endpoint?urn=urn:eu.larkc.endpoint.test.TestEndpoint`

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix larkc: <http://larkc.eu/schema#> .

# Define two plug-ins
_:plugin1 a <urn:eu.larkc.plugin.identify.TestIdentifier> .
_:plugin2 a <urn:eu.larkc.plugin.transform.TestTransformer> .

# Connect the plug-ins
```

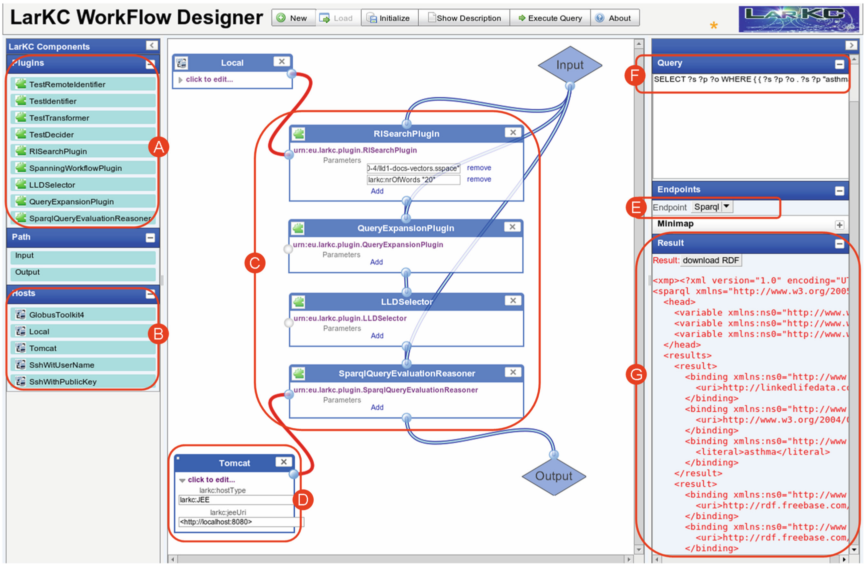
Fig. 2. LarKC management interface

4 Using the Workflow Designer Tool

Workflow Designer is a graphical front-end to LarKC. Workflow Designer is a middleware used on top of the management interface that allows users to construct, configure and execute workflows more efficiently. The designer is a JavaScript-based web application, offering an intuitive GUI for intelligent workflow design. Using Workflow Designer, the users can do the following:

- Retrieve the list of all available plug-ins from the plug-in registry, done through the platform's management interface (Fig. 3a).
- Retrieve the list of all available remote host templates, done through the platform's management interface (Fig. 3b).
- Create workflows by visually dragging and dropping LarKC plug-ins, specifying the dataflow dependencies (Fig. 3c).
- Easily and intuitively specify the deployment host for the plug-in (Fig. 3d).
- Convert the graphical workflow representation into RDF/XML or N3 formats, submit the created workflow description to the management interface, and instantiate the workflow within the platform.

- Choose the needed endpoint (Fig. 3e) and submit a query (e.g. SPARQL) to the chosen end-point (Fig. 3f).
- Visualize the query results in a separate window as an XML document (Fig. 3g).



)-4/lld1-docs-vectors.sspace"
larkc:nrOfWords "20"

larkc:hostType
larkc:JEE
larkc:jeeUri
<http://localhost:8080>

SELECT ?s ?p ?o WHERE {{ ?s ?p ?o . ?s ?p "asthm

Endpoint Sparql

Result: download RDF

```
<xmp><?xml version="1.0" encoding="U
<sparql xmlns="http://www.w3.org/2005
  <head>
    <variable xmlns:ns0="http://www.v
    <variable xmlns:ns0="http://www.v
    <variable xmlns:ns0="http://www.v
  </head>
  <results>
    <result>
      <binding xmlns:ns0="http://www
        <uri>http://linkedlifedata.co
      </binding>
      <binding xmlns:ns0="http://www
        <uri>http://www.w3.org/2004/0
      </binding>
      <binding xmlns:ns0="http://www
        <literal>asthma</literal>
      </binding>
    </result>
    <result>
      <binding xmlns:ns0="http://www
        <uri>http://rdf.freebase.com/
      </binding>
      <binding xmlns:ns0="http://www
        <uri>http://rdf.freebase.com/
      </binding>
```

Fig. 3. The workflow designer's interface

5 Conclusion

The proposed demo is intended to give a brief introduction to the usage of LarKC. It guides the early adopter of the LarKC platform through all the steps needed to develop a simple, plug-in based reasoning application. It also provides an overview of the main LarKC front-ends, namely Management Interface and Workflow Designer.

Reference

1. Assel, M., Cheptsov, A., Gallizo, G., Celino, I., Dell'Aglio, D., Bradeško, L., Witbrock, M., Della Valle, E.: Large knowledge collider: a service-oriented platform for large-scale semantic reasoning. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'2011) (2011)

Did You Validate Your Ontology? OOPS!

María Poveda-Villalón(✉), Mari Carmen Suárez-Figueroa,
and Asunción Gómez-Pérez

Departamento de Inteligencia Artificial, Ontology Engineering Group,
Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain
{mpoveda,mcsuarez,asun}@fi.upm.es

Abstract. The application of methodologies for building ontologies can improve ontology quality. However, such quality is not guaranteed because of the difficulties involved in ontology modelling. These difficulties are related to the inclusion of anomalies or bad practices within the ontology development. Several authors have provided lists of typical anomalies detected in ontologies during the last decade. In this context, our aim in this paper is to describe OOPS! (OntOlogy Pitfall Scanner!), a tool for detecting pitfalls in ontologies.

Keywords: Pitfalls · Bad practices · Ontology evaluation · Ontology engineering

1 Introduction

The growing interest during the last decades of practitioners in ontology development methodologies has facilitated major progress, transforming the art of building ontologies into an engineering activity. The correct application of such methodologies benefits ontology quality. However, such quality is not totally guaranteed because developers must tackle a wide range of difficulties and handicaps when modelling ontologies [1, 2, 5, 8]. These difficulties can imply the appearance of the so-called anomalies or bad practices in ontologies. Therefore, it is important to evaluate the ontologies before using or reusing them in other ontologies or semantic applications.

One of the crucial issues in ontology evaluation is the identification of anomalies in the ontologies. In this regard, it is worth mentioning that Rector et al. [8] describe a set of common errors made by developers during the ontology modelling. Moreover, Gómez-Pérez [4] proposes a classification of errors identified during the evaluation of different features such as consistency, completeness, and conciseness in ontology taxonomies. Finally, Poveda et al. [7] identify an initial catalogue of common pitfalls.

In this context, our goal within this paper is to present an automated tool to help ontology practitioners by detecting common pitfalls during the ontology development. This tool is called OOPS! (OntOlogy Pitfall Scanner!) and represents a new option for ontology developers within ontology evaluation tools as it enlarges the list of errors detected by most recent and available works (e.g. MoKi[1] [6] and XD Analyzer[2]). In addition, OOPS! can be executed independently of the ontology development platform

[1] https://moki.fbk.eu/moki/tryitout/index.php/Main_Page (Last visit on 14-04-2012).

[2] http://neon-toolkit.org/wiki/XDTools (Last visit on: 14-04-2012).

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 402–407, 2015.
DOI: 10.1007/978-3-662-46641-4_35

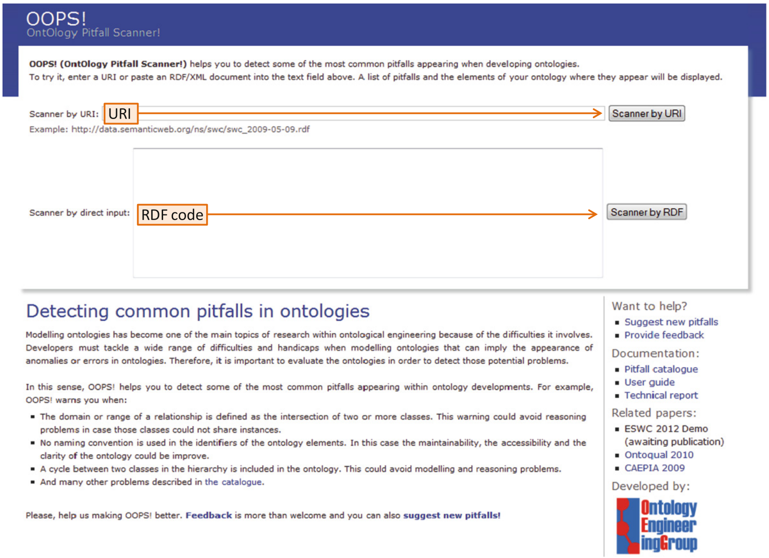
without configuration or installation and it also works with main web browsers (Firefox, Chrome, Safari and Internet Explorer[3]).

The remainder of this paper is structured as follows: Sect. 2 presents the main OOPS! features while Sect. 3 describes its architecture. Finally Sect. 4 outlines some conclusions and future steps to improve OOPS!.

2 OOPS! Features

OOPS! scans ontologies looking for potential pitfalls that could lead to modelling errors [7]. OOPS! is intended to be used by ontology developers during the ontology validation activity, particularly during the diagnosis task. Its main functionality is to analyze ontologies[4] (a) via URL in which an ontology is located or (b) via text input containing the RDF code of the ontology. As a result of the analysis, OOPS! informs developers about which elements of the ontology are possibly affected by pitfalls.

Figure 1 shows OOPS! home page[5] where a user can enter an ontology to be analyzed via URL or by pasting RDF code in the box. This page also presents a brief description of OOPS!.



OOPS! (OntOlogy Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.
To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Example: http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

Detecting common pitfalls in ontologies

Modelling ontologies has become one of the main topics of research within ontological engineering because of the difficulties it involves. Developers must tackle a wide range of difficulties and handicaps when modelling ontologies that can imply the appearance of anomalies or errors in ontologies. Therefore, it is important to evaluate the ontologies in order to detect those potential problems.

In this sense, OOPS! helps you to detect some of the most common pitfalls appearing within ontology developments. For example, OOPS! warns you when:

- The domain or range of a relationship is defined as the intersection of two or more classes. This warning could avoid reasoning problems in case those classes could not share instances.
- No naming convention is used in the identifiers of the ontology elements. In this case the maintainability, the accessibility and the clarity of the ontology could be improve.
- A cycle between two classes in the hierarchy is included in the ontology. This could avoid modelling and reasoning problems.
- And many other problems described in the catalogue.

Please, help us making OOPS! better. **Feedback** is more than welcome and you can also **suggest new pitfalls!**

Fig. 1. OOPS! home page

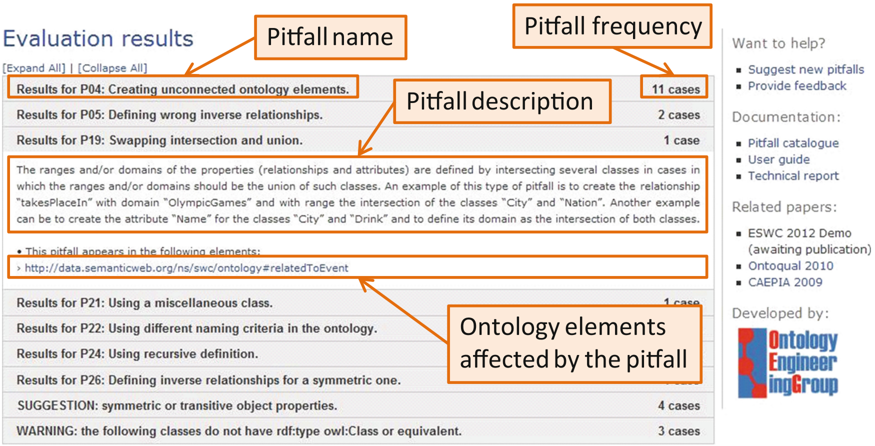
[3] You may experience some layout strange behaviours with Internet Explorer.

[4] The ontology to be analyzed must be implemented in OWL (http://www.w3.org/TR/2009/REC-owl2-primer-20091027/) or RDF (http://www.w3.org/TR/2004/REC-rdf-primer-20040210/).

[5] http://www.oeg-upm.net/oops.

As result of analyzing the ontology provided by the user, OOPS! generates, as it is shown in Fig. 2, a new web page listing the pitfalls appearing in the ontology. This list provides information about (a) how many times a particular pitfall appears, (b) which specific ontology elements are affected by such a pitfall, and (c) a brief description about what the pitfall consist on.

Up to the moment of writing this paper, OOPS! helps to detect a subset of 21 pitfalls of those included in the catalogue.[6] Among others, appearances of pitfalls related to obtaining unexpected inferences (e.g., P6 and P19), to obtaining no inference (e.g., P12 and P13), and to usability issues (e.g., P8 and P11) are considered in OOPS!.


Evaluation results

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements. 11 cases

Results for P05: Defining wrong inverse relationships. 2 cases

Results for P19: Swapping intersection and union. 1 case

The ranges and/or domains of the properties (relationships and attributes) are defined by intersecting several classes in cases in which the ranges and/or domains should be the union of such classes. An example of this type of pitfall is to create the relationship "takesPlaceIn" with domain "OlympicGames" and with range the intersection of the classes "City" and "Nation". Another example can be to create the attribute "Name" for the classes "City" and "Drink" and to define its domain as the intersection of both classes.

• This pitfall appears in the following elements:

› http://data.semanticweb.org/ns/swc/ontology#relatedToEvent

Results for P21: Using a miscellaneous class. 1 case

Results for P22: Using different naming criteria in the ontology.

Results for P24: Using recursive definition.

Results for P26: Defining inverse relationships for a symmetric one.

SUGGESTION: symmetric or transitive object properties. 4 cases

WARNING: the following classes do not have rdf:type owl:Class or equivalent. 3 cases

Fig. 2. Example of evaluation results generated by OOPS!

The current pitfall catalogue is included in the OOPS! web site. It is worth mentioning that the catalogue is continuously revised, since new kinds of modelling mistakes could appear as new ontologies are developed and evaluated. For example, pitfalls from P25 to P29 have been implemented in OOPS! extending the previous catalogue published in [6]. In addition, a form to suggest new pitfalls[7] is provided so that users can contribute enlarging the pitfall catalogue.

It is worth mentioning that OOPS! output points to ontology elements identified as potential errors but not necessarily factual errors and it depends on the type of pitfall detected. There are pitfalls that OOPS! detects in an automated way (e.g., P8 and P28) which means that they should be repaired; while others are detected in a semi-automated way (e.g., P13 and P24), which means that they must be manually checked in order to discern whether the elements identified actually contain errors.

[6] http://www.oeg-upm.net/oops/catalogue.jsp.

[7] http://www.oeg-upm.net/oops/submissions.jsp.

3 OOPS! Architecture

In this section OOPS! underlying architecture is presented (see Fig. 3) as well as some technical details. Basically, OOPS! is a web application based on Java EE,[8] HTML,[9] jQuery,[10] JSP[11] and CSS[12] technologies. The web user interface consists on a simple view where the user enters the URL pointing to or the RDF document describing the ontology to be analyzed. Once the ontology is parsed using the Jena API[13] the model is scanned looking for pitfalls, from those available in the pitfall catalogue. During this phase, the ontology elements involved in potential errors are detected as well as warnings regarding RDF syntax and some modelling suggestions are generated. Finally, the evaluation results are displayed by means of the web user interface showing the list of pitfalls appearing, if any, and the ontology elements affected as well as explanations describing the pitfalls.

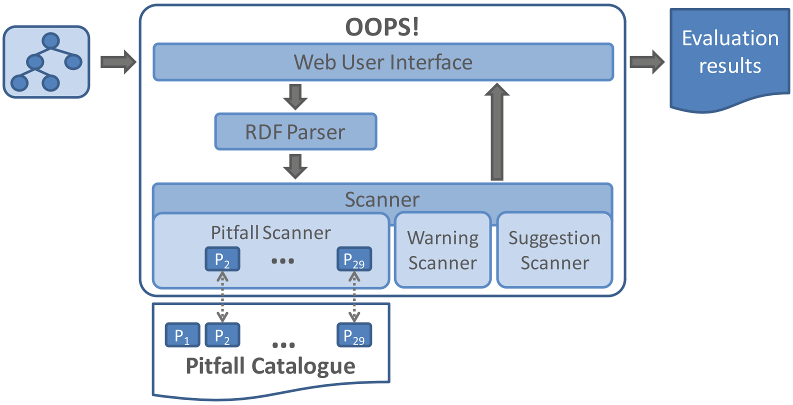


Fig. 3. OOPS! architecture

4 Conclusions and Future Work

In this paper we have presented OOPS! main features and architecture and how this tool represents a step forward within ontology evaluation tools as (a) it enlarges the list of errors detected by most recent and available works (e.g. MoKi [6] and XD Analyzer), (b) it is fully independent of any ontology development environment and (c) it works with main web browsers (Firefox, Chrome, Safari and Internet Explorer).

[8] http://www.oracle.com/technetwork/java/javaee/overview/index.html.

[9] http://www.w3.org/html/wg/.

[10] http://jquery.com/.

[11] http://www.oracle.com/technetwork/java/javaee/jsp/index.html.

[12] http://www.w3.org/Style/CSS/.

[13] http://jena.sourceforge.net/.

OOPS! is currently being tested by Ontology Engineering Group[14] members in order to debug it and extend its functionality. However, OOPS! has been already used by other ontology developers who belong to different organizations (such as AtoS, Tecnalia, *Departament Arquitectura, La Salle* at *Universitat Ramon Llull* and Human Mobility and Technology Laboratory at CICtourGUNE). In fact, OOPS! is freely available to users on the Web. It includes a link to a feedback form[15] so that everyone can test it and provide feedback and suggestions to be included in the tool.

As long as we discover new pitfalls during our research, they will be included in the current pitfall catalogue and implemented in OOPS!. In addition, we plan to improve and extend OOPS! features in the following lines:

- To group and classify pitfalls by categories according to previous ontology quality criteria identified in [3] and [4]. This feature will provide more flexibility to the ontology evaluation, since it will allow users to diagnose their ontologies just with respect to the dimensions they are interested in.
- To increase OOPS! features with guidelines about how to solve each pitfall. This information will ease the task of repairing the ontology after the diagnosis phase.
- To associate priority levels to each pitfall according to their different types of consequences they can convey when appearing in an ontology. This feature will be useful to prioritize actions to be taken during the repairing task.
- To make REST services available in order to allow other developments to use and integrate the pitfall scanner functionalities within their applications.
- To allow users to define their own pitfalls, according with their particular quality criteria, in order to use OOPS! in a customized fashion.

Acknowledgments. This work has been partially supported by the Spanish projects *BabelData* (TIN2010-17550) and *BuscaMedia* (CENIT 2009-1026). We would also like to thank Miguel Ángel García for his technical support.

References

1. Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., Suárez-Figueroa, M.C.: Natural language-based approach for helping in the reuse of ontology design patterns. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 32–47. Springer, Heidelberg (2008)
2. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on pattern-based ontology design. In: Proceedings of K-CAP 2009, pp. 41–48 (2009)
3. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 140–154. Springer, Heidelberg (2006)
4. Gómez-Pérez, A.: Ontology evaluation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, pp. 251–274. Springer, Heidelberg (2004)

[14] http://www.oeg-upm.net/.

[15] http://www.oeg-upm.net/oops/form.jsp.

5. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical report SMI-2001-0880, Standford Medical Informatics (2001)
6. Pammer, V.: PhD Thesis: Automatic Support for Ontology Evaluation Review of Entailed Statements and Assertional Effects for OWL Ontologies. Engineering Sciences. Graz University of Technology. http://know-center.tugraz.at/wp-content/uploads/2010/12/Dissertation_Viktoria_Pammer.pdf
7. Poveda, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: A double classification of common pitfalls in ontologies. In: OntoQual 2010 - Workshop on Ontology Quality at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010). Proceedings of the Workshop on Ontology Quality - OntoQual 2010. CEUR Workshop Proceedings, Lisbon, Portugal, pp. 1–12, 15 October 2010. ISBN: ISSN 1613-0073
8. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 63–81. Springer, Heidelberg (2004)

Does It Fit? KOS Evaluation Using the ICE-Map Visualization

Kai Eckert[1](✉), Dominique Ritze[1], and Magnus Pfeffer[2]

[1] University of Mannheim, University Library, Mannheim, Germany
{kai.eckert,dominique.ritze}@bib.uni-mannheim.de
[2] Stuttgart Media University, Stuttgart, Germany
pfeffer@hdm-stuttgart.de

Abstract. The ICE-Map Visualization was developed to graphically analyze the distribution of indexing results within a given Knowledge Organization System (KOS) hierarchy and allows the user to explore the document sets and the KOSs at the same time. In this paper, we demonstrate the use of the ICE-Map Visualization in combination with a simple automatic indexer to visualize the semantic overlap between a KOS and a set of documents.

1 Introduction

Hierarchical Knowledge Organization Systems (KOS), like thesauri, taxonomies, or other kinds of (lightweight) ontologies are widely used to describe all kinds of resources, large document corpora amongst others. In the Semantic Web, these KOSs are usually described in SKOS (Simple Knowledge Organization System[1]). The public availability of diverse KOSs on the web leads to new possibilities regarding the reuse of existing KOSs, but at the same time raises the question which KOS is suitable for the resources to be described. Thus, measuring the overlap of the subject coverage of a given document set and multiple KOSs is a necessary task before starting further, possibly time-consuming and costly efforts to annotate the documents. For these measurements, any one-dimensional analysis in the line of "summing the number of concepts that appear in the documents" is not sufficient. First, there is no baseline to compare the generated numbers with and second, all hierarchical information is lost and it is not possible to compare the results in their subject context. Instead, we propose to use a graphical visualization that preserves the hierarchical context as well as a statistical measure. The numerical results which are provided by the measure are intuitive to understand and well suited for a graphical representation. They are combined in the ICE-Map Visualization.

In this paper, we use the ICE-Map Visualization [2] to visualize the overlaps between KOSs and document sets. This visualization is based on a treemap and allows the user to browse a KOS hierarchy interactively. The colors indicate which parts of the KOS fit the documents. To create the visualization, the

[1] http://www.w3.org/TR/skos-reference/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 408–412, 2015.
DOI: 10.1007/978-3-662-46641-4_36

ICE-Map Visualization requires that the documents are annotated with KOS concepts. In the discussed use case, there are no annotations yet and manual assignment is obviously not feasible. Thus, it is necessary to automatically generate them. We show that the ICE-Map Visualization in combination with our automatic indexing approach is suitable to calculate and visualize the overlap between a KOS and a document set in a way that users can make informed decisions on whether the document set fits to the KOS.

2 Setup

We apply a KOS-based indexing approach to determine which concepts of the KOS occur in a given document. For this purpose, we developed a pure linguistic indexer called LOHAI [1] which is free and open source. It uses part-of-speech tagging, stemming, and word-sense disambiguation. It is especially important that the indexer does not rely on any additional knowledge sources and is kept simple to ensure usability as well as comprehensibility of the results. The reference implementation is available online[2]. The weighted concept annotations created by LOHAI form the basis for the ICE-Map Visualization.

The ICE-Map Visualization is an approach for visual datamining (VDM) specifically designed for the purpose of maintenance and use of concept hierarchies in various settings. In this paper, we use it to visualize the number of documents associated with the concepts in the thesaurus. The ICE-Map Visualization is described in detail by Eckert [2]. Here, we briefly recapitulate the basic idea and introduce the weight function employed in this paper.

The usage of a concept c is determined by a weight function $w(c) \in \mathbb{R}_0^+$ that assigns a non-negative, real weight to it. Based on this weight function, we further define:

$$w^+(c) = w(c) + \sum_{c' \in \text{Children}(c)} w^+(c') \tag{1}$$

with Children(c) being the direct child concepts (narrower concepts) of c. $w^+(c)$ is a monotonic function on the partial order of the concept hierarchy H, i.e., the value never increases while walking down the hierarchy. This gives the value of the root node root(c) a special role as the maximum value[3] of w^+, which we denote as $\hat{w}^+$: $\hat{w}^+(c) = w^+(\text{root}(c)) = \max_H w^+(c)$.

If we use the number of annotations made for a given concept as the weight function $w(c)$, we can calculate the likelihood that a concept is assigned to a random document as follows[4]:

$$L(c) = \frac{w^+(c) + 1}{\hat{w}^+(c) + 1} \qquad L(c) \in (0, 1] \tag{2}$$

[2] https://github.com/kaiec/LOHAI.

[3] The root node is defined as the only concept c in H for which holds that Parents(c) $= \emptyset$. Note that we require H to have a single root concept. Otherwise, we introduce an artificial single root concept that becomes the parent of all former root concepts.

[4] The addition of 1 is necessary to allow a value of 0 for $w(c)$. Otherwise, the logarithm of $L(c)$ (cf. Eq. 3) would not be defined for $w(c) = 0$.

In information theory, the *information content* or *self-information* of an event x is defined as $-\log L(x)$, i.e., a higher information content means a more unlikely event. Together with a normalizing factor, we get the following definition for the information content $IC(c) \in [0, 1]$ of a concept c:

$$IC(c) = \frac{-\log L(c)}{\log(\hat{w}^{+}(c) + 1)} \qquad \hat{w}^{+}(c) \neq 0 \tag{3}$$

This is again a monotonic function on the partial order of H and assigns 0 to the root concept and 1 to concepts with $w(c) = 0$. The ICE-Map Visualization always visualizes the difference of two information contents based on two different weight functions or two different data sets: $D(c) = IC_1(c) - IC_2(c)$. The power of the ICE-Map Visualization lies in the possibility to choose arbitrary weight functions for IC_1 and IC_2. To calculate the weight of a concept regarding its usage in a document set, we use:

$$w_1(c) = \sum_{a \in \mathrm{Aset}(c)} \mathrm{Weight}(a) \tag{4}$$

with $\mathrm{Weight}(a)$ denoting the weight of a single annotation a as calculated by LOHAI[5] and $\mathrm{Aset}(c)$ being the set of annotations assigned to a concept c.

To evaluate the suitability, we compare the information content based on Eq. 4 to the intrinsic information content [3] – a heuristic for the expected information content of a concept based on its position in the hierarchy. In our statistical framework, we obtain the intrinsic information content by employing the following weight function:

$$w_2(c) = |\mathrm{Children}(c)| \tag{5}$$

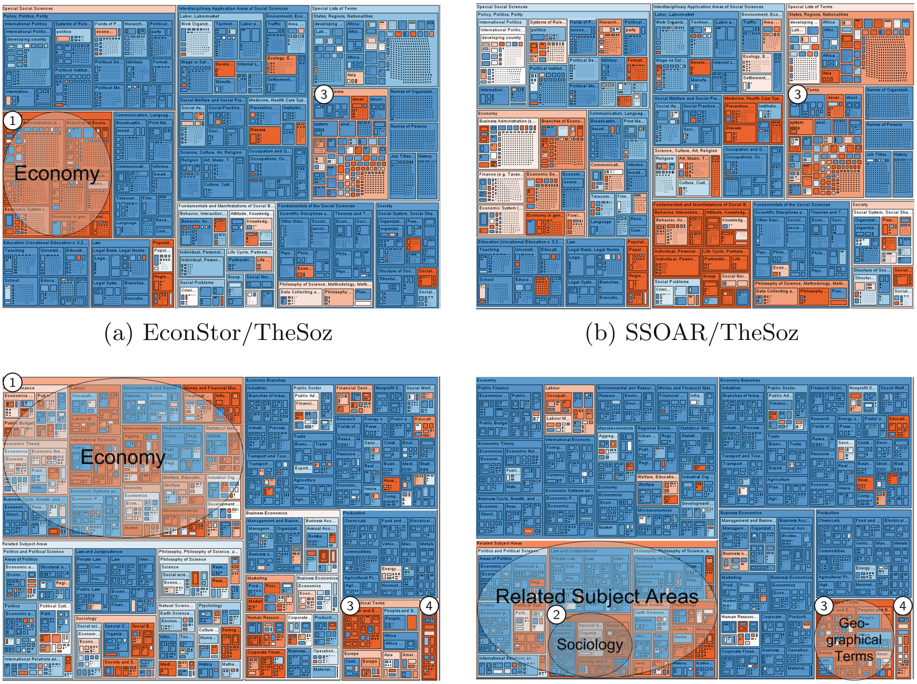
The ICE-Map Visualization uses a treemap to visualize the concept hierarchy together with the results of the analysis. It gives a broad overview of the whole document set with the annotated concepts and supports zooming and navigating the hierarchy of the KOS to get a detailed view. The automatic indexer LOHAI and the ICE-Map Visualization are included in our KOS analysis software SEMTINEL[6].

3 Experiments

To demonstrate the usefulness of the ICE-Map Visualization together with LOHAI to measure the suitability of thesaurus and document collection, comprehensive document sets and KOSs are needed. The KOSs need to have a significant overlap without describing the same topic and we also need at least one

[5] Strictly speaking, from an information-theoretic perspective, this function interprets the *tf-idf* weight of the annotation as the likeliness of being an annotation for the document. This interpretation is not correct, as *tf-idf* is no probability value.

[6] http://www.semtinel.org/.



(a) EconStor/TheSoz

(b) SSOAR/TheSoz

(c) EconStor/STW

(d) SSOAR/STW

Fig. 1. EconStor and SSOAR indexed with TheSoz and STW

document set for each KOS where we can assume that it fits to the KOS. Furthermore, we would prefer to use well-established KOSs that are freely available and widely used. They need to have a significant size and at least one language in common.

For the experiments we chose TheSoz[7] (Thesaurus for the Social Sciences) version 0.86 and STW[8] (Standard Thesaurus Wirtschaft) version 8.08 in our experiments. Both KOSs are available as SKOS vocabularies and have a comparable size of about 7000 concepts with English labels. While TheSoz covers all social science disciplines, STW focuses on economical topics. As document sets, we apply SSOAR[9] and EconStor[10]. SSOAR as well as EconStor are open-access servers, maintained by GESIS and ZBW, respectively, the organisations that also publish the KOSs. Of both sets, we take a random subset of 2700 documents to ensure comparable results. As for the KOSs, SSOAR has its focus on social science and EconStor on economy. Despite of some deviations, we can assume that SSOAR naturally fits to TheSoz and EconStor fits to STW.

[7] http://lod.gesis.org/thesoz/.

[8] http://zbw.eu/stw/versions/latest/download/about.en.html.

[9] http://www.ssoar.info/.

[10] http://www.econstor.eu/.

In Fig. 1, we show the resulting visualization for all combinations of KOSs and document sets. The coloring represents the value of the weight function. It ranges from blue which means the weight for this concept is really low over white and finally to red which indicates a very high weight, compared to the reference weight determined by the heuristic. This economical bias of Econstor can clearly be seen in Fig. 1a since most concepts which are used in the documents are narrower concepts of *Economy* ①. In contrast, the results of SSOAR/TheSoz (Fig. 1b) do not point out such a clear focus on one specific field. It is interesting that Economy is still very visible, an indicator that both sciences indeed have an overlap reflected in the document sets. Moreover, the *General Terms* section ③ is used similarly by both document sets. When the STW is used as KOS, it can be seen in Fig. 1c that EconStor documents contain concepts of several parts (especially *Economy* ①) while SSOAR documents use concepts which are narrower ones of *Related Subject Areas* and especially of *Sociology* (Fig. 1d, ②). Other parts that are used well by both document sets are again general parts like *Geographical Terms* ③ and *General Terms* ④. All in all, the semantic overlaps of the document sets with the KOSs are clearly visible. Without any further information, we evaluated two document sets and two KOSs and were able to develop a deeper understanding of them by just browsing through the ICE-Map Visualization.

4 Conclusion

We presented an approach to visualize the semantic overlap of a KOS and a document set. We combined the ICE-Map Visualization with a very simple automatic indexer called LOHAI. We chose two KOSs and two document sets with a significant topical overlap to demonstrate the usefulness of our approach. Based on the resulting visualization, we could show that it is possible to identify whether KOS and document set topically fit together. Thus, the choice of a suitable KOS or the maintenance of an already used KOS is strongly simplified.

References

1. Eckert, K.: LOHAI: Providing a baseline for KOS based automatic indexing.In: Proceedings of the First International Workshop on Semantic Digital Archives (SDA) at the International Conference on Theory and Practice of Digital Libraries (TPDL) 2011, Berlin, 29 September 2011
2. Eckert, K.: The ICE-Map Visualization. Technical Report TR-2011-003, University of Mannheim, Department of Computer Science (2011)
3. Seco, N., Veale, T., Hayes, J.: An intrinsic information content metric for semantic similarity in wordnet. In: Proceedings of the 16th European Conference on Artificial Intelligence, Valencia, Spain, pp. 1089–1090 (2004)

A Demo for Efficient Human Attention Detection Based on Semantics and Complex Event Processing

Yongchun Xu[1](✉), Ljiljana Stojanovic[1], Nenad Stojanovic[1], and Tobias Schuchert[2]

[1] FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
{yongchun.xu, ljiljana.stojanovic, nenad.stojanovic}@fzi.de
[2] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), FraunhoferStr. 1, 76131 Karlsruhe, Germany
tobias.schuchert@iosb.fraunhofer.de

Abstract. In this paper we present a demo for efficient detecting of visitor's attention in museum environment based on the application of intelligent complex event processing and semantic technologies. Semantics is used for the correlation of sensors' data via modeling the interesting situation and the background knowledge used for annotation. Intelligent complex event processing enables the efficient real-time processing of sensor data and its logic-based nature supports a declarative definition of attention situations.

Keywords: Sensor · Human attention · Complex event processing · Ontologies

1 Introduction

In this paper we describe a demo, which shows a semantic-based system providing personalized and adaptive experience for the visitor, in which the digital contents react depending on the artwork and the user's engagement/attention state. In the demo we use semantic technologies for the correlation of sensors' data via modeling the so-called interesting situation and use complex-event processing to recognize the attention patterns in the event stream.

2 Problem Overview

In order to enable an adaptive experience for the visitor to a museum, the demo is constructed around a four-phase OODA (Observe, Orient, Decide, Act) as shown on Fig. 1. In the Observe phase, our approach is concerned with the measurement of covert cues that may indicate the level of interest of the user. In order to consider how a user perceives an artwork, different sensors have been considered: The monitoring of visual behavior will allow the system to identify the focus of attention. The acoustic

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 413–418, 2015.
DOI: 10.1007/978-3-662-46641-4_37

module should provide important information about environmental influences on patterns of visual attention or psychophysiology. Finally, a video-based hand gesture recognition provides an additional input modality for explicit interaction with the system (e.g., for selecting certain visual items, navigating through menus).

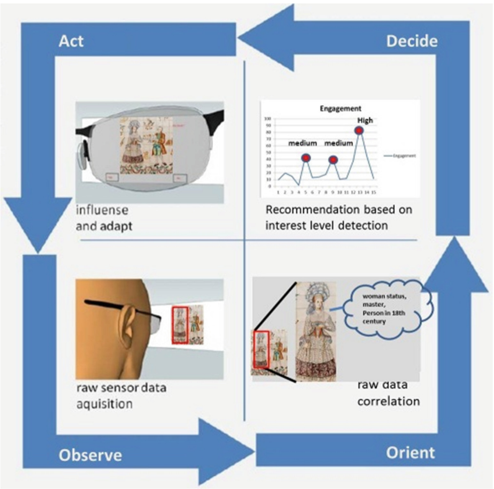


Fig. 1. OODA cycle

All data streams are collected and analyzed in real-time in order to yield a dynamic representation of the user attention state (phase Orient). In the Decide phase, covert physiological cues are used to to measure the level of interest or engagement with artwork or with augmented content presented via the AR device. Based on the interpretation of this complex state, the provision of augmented content from a repertoire of available content is made. The presentation of selected content via the AR device (e.g. visual, audio) is subsequently executed during the final Act stage.

3 Demo Challenge: Semantic-Based Attention Detection

The challenge of this demo is how to detect the attention of the visitor in the museum. In most situations the attention of the visitors can be determined according to the gaze behavior of the visitors. In some cases the observed object is the attention object of the visitor, while in other cases the visitors pay attention to the information behind the observed objects. Thus, we distinguish between visual attention and content-related attention. Figure 2 summaries categories of attentions that is relevant in the museum context, including visual attention and content-based attention.

Sustainable attention (Fig. 2(a)) means that the attention is focused over extended periods of times. Similarly, if during at least 3 seconds the acoustic noise level is large (e.g. a mobile phone of a visitor is ringing), then selective attention and shifting is detected (see Fig. 2(b)). Finally, divided attention (see Fig. 2(c)) means sharing of attention by focusing on more than one relevant object at one time. One possible way of "calculating" similarity is to consider semantics of the topics behind the artworks.

The presented visitor attention model puts some requirements on what has to be modeled: different types of sensors data and their fusion in order to detect visual attention and semantic about artwork in order to profit from content-based attention.

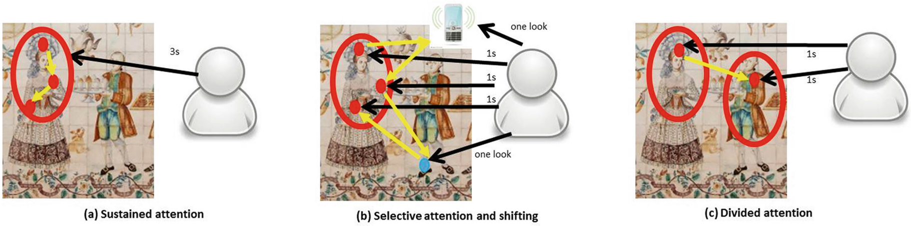


Fig. 2. Different attention categories relevant for museums: (a) Sustained attention; (b) Selective attention and shifting; and (c) Divided attention

4 The Role of Semantic Processing

The demo is based on knowledge-rich, context-aware, real-time artwork interpretation aimed at providing visitors with a more engaging and more personalized experience. Indeed, we propose to combine annotation of artworks with the time-related aspects as key features to be taken into account when dealing with interpretation of artworks.

Thus, the aspects of the museums modeled by ontologies are classified into:

- Static aspects which are related to the structuring of the domain of interest, i.e. describing organization of an artwork and assigning the metadata to it;
- Dynamic aspects which are related to how a visitor's interpretation the elements of the domain of interest (i.e. artworks) evolve over time.

5 Demo Setting

The demo will be performed using following hardware equipment (Fig. 3):

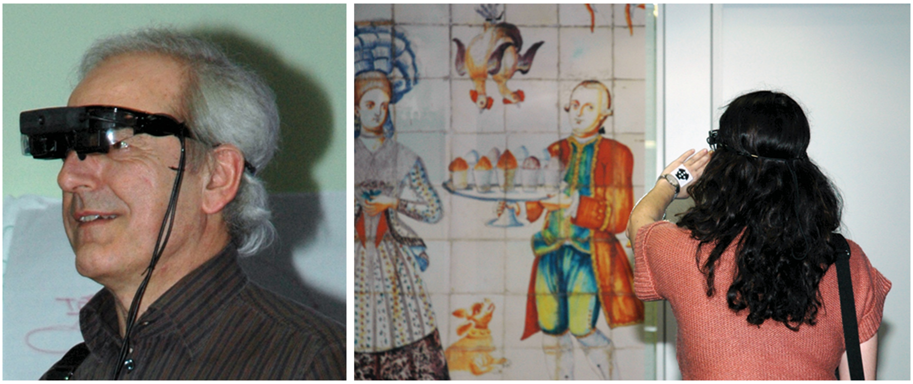


Fig. 3. The demo setting and equipment

- A Poster of Valencia Kitchen in MNAD (Museo Nacional de Artes Decorativas, Madrid Spain) as artwork
- Vuzix Star 1200 AR glasses with camera
- M-Audio Fast Track Pro audio card and BEYERDYNAMIC MCE 60.18 mic
- Bio sensors

6 Demo Workflow

Figure 4 shows the concrete workflow of the demo. The whole workflow is based on the OODA model.

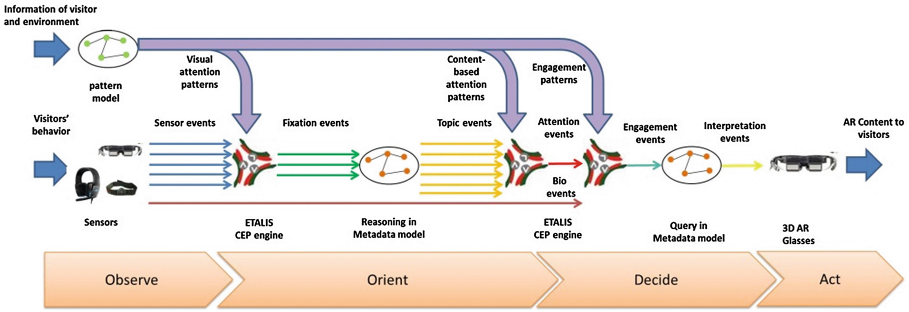


Fig. 4. Demo workflow

Observe phase: After the visitor got the equipment (e.g. sensors and AR glasses), the patterns have been deployed according to the visitor's information and actual museum environment. When the visitor stands before Valencia Kitchen and starts looking at the kitchen wall, the visual sensor detects the gaze of the visitor. Meanwhile the acoustic sensor monitors the environment sound for possible disturbance and the bio sensors monitors the physiological signals of the visitor. Assuming that the visitor has interest on the "person in 18th century", in the situation of divided visual attention the visitor looks at the head of the Lady for one second firstly, after that the visitor looks somewhere else and then the visitor looks at the servant for another one second. A sequence of gaze events and some acoustic events and bio events is published.

Orient phase: The CEP engine ETALIS receives the sensor events and use the visual attention pattern. Two short fixations (the first one on the lady and the second one on the servant) are detected and published as events.

The knowledge base receives the two short fixation events and uses SPARQL to find all the related topics of the observed object in the metadata According to the annotation of the Valencia Kitchen (Fig. 5) for the fixation on lady three direct topics: "woman status", "master", "person in 18th century" and two indirect topics: "social stratification", "human status" are found. For the short fixation on servant we got direct topics: "man status", "party servant", "person in 18th century" and indirect topics: "human

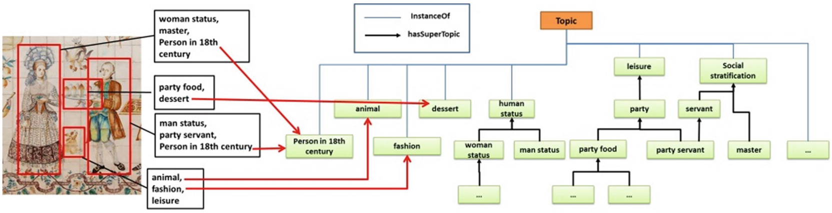


Fig. 5. Annotation of the Valencia Kitchen

status", "servant", "social stratification". All these topics are published as topic events. ETALIS detects the attention event according to these topic events.

Decide phase: The CEP engine detects the interest and engagement of visitors based on the attention events and the bio signal events. If the bio signal shows the visitor the interesting level is high and meanwhile an attention is detected, we can conclude that the visitor has interest on such topic. In our example the visitor has interest on the topics: "person in 18th century", "human status" and "social stratification". This discovery of engagement is sent out as event by CEP engine.

The knowledge base receives this engagement event and finds the related metadata (guide content) about the topic through reasoning and publishes the metadata as Interpretation event.

Act phase: Lastly in the act phase the AR glasses get the interpretation event and show the metadata (guide content) as Augmented Reality content on the glasses to the visitor.

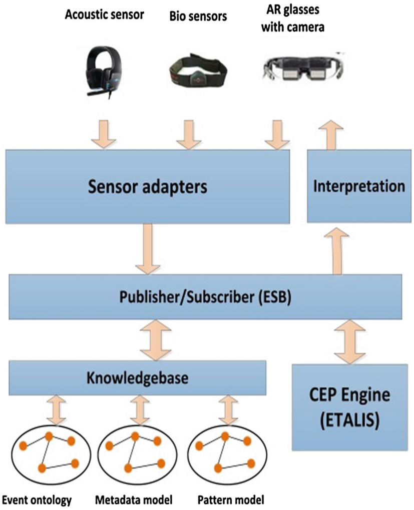


Fig. 6. System architecture

7 Demo Implementation

Figure 6 shows the architecture of our system. The following sensors are used: see-through glasses with integrated camera that can track the gaze of visitors and display the augmented reality (AR) content to visitors; acoustic sensor senses the acoustic information surrounding visitors such as environment noise or the content that visitors are listening to, and bio sensor observes the biological signals of visitors like heart rate.

All components communicate through ActiveMQ ESB by publishing and/or subscribing to events. The sensor adapters connect to the sensor hardware, collect the physical signals of visitors such as gaze, sound, heart rate and other bio signals from sensors and translate them into meaningful sensor events to be processed by the CEP engine. The complex event processing part detects the situations of interests based on predefined patterns and real-time sensor data. Semantic technologies are used to store the annotation of artworks, semantically-enriched sensor data and patterns. The knowledge base manages the background knowledge and provides the query function to other parts. The interpretation part recommends AR content to visitors based on their engagement and query results.

Domain-Specific OWL Ontology Visualization with OWLGrEd

Karlis Cerans(✉), Renars Liepins, Arturs Sprogis, Julija Ovcinnikova, and Guntis Barzdins

Institute of Mathematics and Computer Science,
University of Latvia, Riga, Latvia
{Karlis.Cerans,Renars.Liepins,Arturs.Sprogis,
Julija.Ovcinnikova,Guntis.Barzdins}@lumii.lv

Abstract. The OWLGrEd ontology editor allows graphical visualization and authoring of OWL 2.0 ontologies using a compact yet intuitive presentation that combines UML class diagram notation with textual Manchester syntax for expressions. We present an extension mechanism for OWLGrEd that allows adding custom information areas, rules and visual effects to the ontology presentation thus enabling domain specific OWL ontology visualizations. The usage of OWLGrEd and its extensions is demonstrated on ontology engineering examples involving custom annotation visualizations, advanced UML class diagram constructs and integrity constraints in semantic database schema design.

Keywords: OWL · UML/OWL profile · OWLGrEd · Domain-specific ontology visualization · Semantic databases · Integrity constraints

1 Introduction

Intuitive ontology visualization is a key for their learning, exchange, as well as their use in conceptual modeling and semantic database schema design. A number of tools and approaches exist for rendering and/or editing OWL [1, 2] ontologies in a graphical form, including UML Profile for OWL DL [3], ODM [4], TopBraid Composer [5], Protégé [6] plug-in OWLViz [7], OWLGrEd [8, 9]. The approaches of [3, 4, 8, 9] use UML [10, 11] class diagrams to visualize OWL ontologies. A core principle here is to visualize an independent hierarchy of ontology classes and then structure the data and object property visualizations along the property domain and range classes. Depicting OWL classes as UML classes, OWL object properties as association roles and OWL data properties as attributes allows for easy graphical visualization also of subclass assertions, simple cardinality constraints and inverse-of relations. Further OWL ontology constructions (e.g. class expressions, properties with more than one domain

K. Cerans, J. Ovcinnikova, and G. Barzdins—Partially supported by European Union via European Regional Development Fund project 2011/0009/2DP/2.1.1.1.0/10/APIA/VIAA/112.
R. Liepins and A. Sprogis—Partially supported by Latvian 2010.-2013. National Research Program Nr.2 project Nr.5.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 419–424, 2015.
DOI: 10.1007/978-3-662-46641-4_38

assertion, sub-property relations etc.) are then handled by some auxiliary means in the notation and the editor. The design choice for OWLGrEd is to use textual OWL Manchester syntax [12] for class expressions where the graphical notation is not available or is not desired thus allowing compact and comprehensible presentation of up to medium-sized ontologies[1] within a single diagram.

Although UML-style class diagram notation for basic OWL constructs can be successfully used in ontology rendering and authoring, there are further features that would be welcome in a graphical ontology editor. Since annotations in OWL 2.0 [2] may carry substantial model information that just does not fit into the "logical" part of the ontology, it would be important to offer means for domain-specific visualization of annotation assertions via specific textual presentation or graphical effects, e.g. as outlined in [13]. As a special case, a UML-style modeling in OWL would benefit from graphical composition or property derived union notation (modeled semantically as annotation assertions to the respective properties).

With the advent of semantic OWL-based databases, such as StarDog [14], an important issue is rising about incorporating integrity constraints [15, 16], also expressed in OWL syntax, in graphical database schema design. As an example of our technology application we provide a domain-specific ontology visualization profile for axiom-level annotations that separate "proper" (i.e. open-world) OWL axioms from integrity constraints, depicted within the same graphical ontology diagram.

The demonstration shows (i) working with OWLGrEd tool to render and author OWL ontologies (ii) OWLGrEd extension mechanism for creating domain-specific ontology visualization tools and (iii) created domain-specific tools, including OWLGrEd/S for integrity constraint specification, at work.

2 OWLGrEd Notation and Editor

OWLGrEd[2] provides a complete graphical notation for OWL 2 [2], based on UML class diagrams. We visualize OWL classes as UML classes, data properties as class attributes, object properties as associations, individuals as objects, cardinality restrictions on association domain class as UML cardinalities, etc. We enrich the UML class diagrams with the new extension notations, e.g. (cf. [8, 9]):

- fields in classes for *equivalent class*, *superclass* and *disjoint class* expressions written in Manchester OWL syntax [12];
- fields in associations and attributes for *equivalent*, *disjoint* and *super* properties and fields for property characteristics, e.g., *functional*, *transitive*, etc.;
- anonymous classes containing *equivalent class expression* but no name (we show graphically only those anonymous classes that need to have graphic represen-tation in order to be able to describe other ontology concepts in the diagram);
- connectors (as lines) for visualizing binary *disjoint*, *equivalent*, etc. axioms;
- boxes with connectors for n-ary *disjoint*, *equivalent*, etc. axioms;

[1] Please see http://owlgred.lumii.lv/examples for some ontology presentations.

[2] http://owlgred.lumii.lv/.

- connectors (lines) for visualizing object property restrictions *some*, *only*, *exactly*, as well as cardinality restrictions.

OWLGrEd provides option to specify class expressions in compact textual form rather than using separate graphical element for each logical item within class expression. If an expression is referenced in multiple places, it can optionally be shown as an anonymous class. An anonymous class is also used as a base for property domain/range specification, if this domain/range is not a named class.

Figure 1 illustrates some basic OWLGrEd constructs on simple mini-University ontology, including different notation options for *EquivalentClasses* assertion, object property restriction and a comment. The notation is explained in more detail in [8].

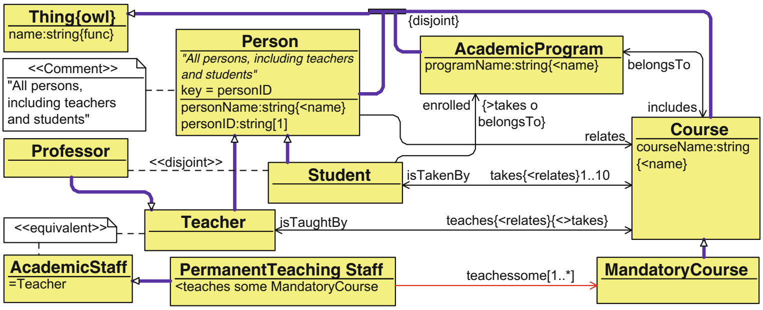


Fig. 1. Example: OWLGrEd notation for a mini-University ontology

The OWGrEd editor offers ontology interoperability (import/export) functionality with Protégé 4.1. ontology editor [6]. The principal OWLGrEd usage tool chains are:

– ontology authoring (create and edit an ontology in OWLGrEd, then export it to Protégé to analyze and possibly submit it to other ontology processing tools)
– ontology visualization (an ontology that is imported from Protégé is displayed graphically to obtain a comprehensible visual view on it).

Any combination of these two OWLGrEd usage patterns, including ontology round-trip engineering between OWLGrEd and Protégé are possible, as well.

3 Creating Domain-Specific Ontology Visualizations

Domain-specific ontology visualizations in OWLGrEd ontology editor are defined by means of ontology visualization profiles. Each ontology visualization profile consists of a set of visual item (= abstract field) specifications, where each field comprises:

(i) field type (e.g. textual/boolean(= check box)/combo box field)
(ii) field appearance (e.g. visibility and text font style)
(iii) visual effects on ontology diagram symbols and other fields (e.g. symbol color and shape)

(iv) field semantics (what OWL axioms or axiom annotations a value in the field corresponds to).

For an ontology to be visualized in OWLGrEd in a domain-specific way, the corresponding ontology visualization profile has to be created or imported using OWLGrEd visualization profile plug-in. When the ontology created in such domain-specific extension of OWLGrEd is exported to Protégé ontology editor, the ontology diagram node and edge fields that correspond to profile visual items generate the OWL axioms or axiom annotations, as specified in field semantics description.

Consider an ontology *A* fragment visualized in a domain-specific way, as in Fig. 2. The graphical notation has a new class field "DB" rendered textually with prefix "*{DB:*" and suffix "*}*", a class field "isImportant" whose value "true" is rendered as orange background and 3D shape of the class symbol, and association role sub-field "isComposition" whose value "true" is rendered as diamond symbol on opposite association end. We desire to have these fields correspond to the following axioms:

AnnotationAssertion(A:DBExpr A:AcademicProgram "XProgram")
AnnotationAssertion(A:DBExpr A:Course "XCourse")
AnnotationAssertion(A:isImportant A:Teacher "true")
AnnotationAssertion(A:isComposition A:includes "true")

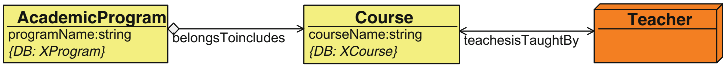


Fig. 2. Simple domain-specific ontology annotation visualization

This is achieved by semantics declarations: *AnnotationAssertion(:DBExpr $subject $value)* for the field "DB", *AnnotationAssertion(:isImportant $subject "true")* for the value "true" in the boolean-typed field "isImportant", and *AnnotationAsser-tion(:isComposition $subject "true")* for the value "true" in "isComposition".

When an ontology that uses the *A:isImportant*, *A:DBExpr* and *A:isComposition* annotations (or other OWL built-in or user defined annotations whose visual image is foreseen in a loaded ontology visualization profile) is imported into OWLGrEd, the editor is able to create the domain-specific visualization (like Fig. 2) automatically.

4 Integrity Constraints in Semantic Database Schema Design

Using the ontology of Fig. 1 as a schema for semantic database would be problematic due to the standard OWL axiom interpretation in "open-world" sense.[3] The solution we are offering is to mark explicitly the axioms whose interpretation in the open-world sense is undesirable, as integrity constraints.[4]

[3] This interpretation would allow to infer e.g. that a person is a student, if he/she has been entered into the database as taking (instead of teaching) a course, or that a student is enrolled in two academic programs just because of taking courses that belong to both of them.

[4] We refer to [15, 16] for integrity constraint discussion in the context of StarDog databases, noting that our integrity constraint encoding is easily interconvertible with that of StarDog's.

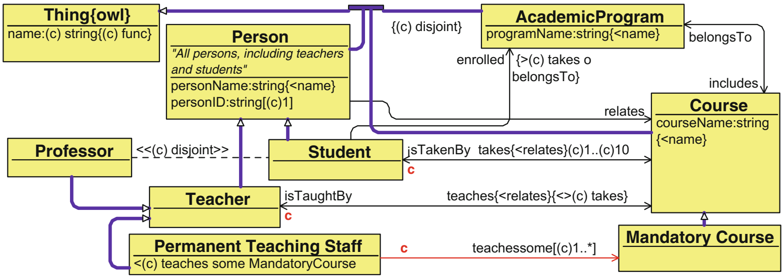


Fig. 3. Integrity constraint specification for mini-University ontology

The OWLGrEd editor is extended by "integrity constraint" visualization profile[5] that foresees a possibility to attach a (c)-mark ("c" for constraint) to visual places that can be identified as "holding" the concrete axioms, as in Fig. 3 for mini-University.

In the example, for instance, the axiom *ObjectPropertyDomain(A:takes A:Student)* is annotated to become *ObjectPropertyDomain(Annotation(C:isConstraint "true") A: takes A:Student)* for a suitable namespace *C* holding the *isConstraint* annotation property. The visual **c**-notation placed at the beginning of *takes*-role link is obtained from a "DomainMode" field under the association role *takes*. The corresponding semantics specification for the "DomainMode" field causing the considered *ObjectPropertyDomain*-axiom annotation is *Annotation(C:isConstraint "true").*

The considered examples outline the potential of domain-specific ontology visualization using OWLGrEd and invite the reader either to apply the demonstrated ontology visualization profiles, or design his/her own ontology visualization tools.

References

1. Smith, M.K., Welty, C., McGuiness, D.: OWL Web Ontology Language Guide (2004)
2. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (2009)
3. Brockmans, S., Volz, R., Eberhart, A., Löffler, P.: Visual modeling of OWL DL ontologies using UML. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 198–213. Springer, Heidelberg (2004)
4. ODM UML profile for OWL. http://www.omg.org/spec/ODM/1.0/PDF/
5. TopBraid Composer. http://www.topquadrant.com/products/TB_Composer.html
6. Protégé 4. http://protege.stanford.edu/
7. OWL Viz. http://www.co-ode.org/downloads/owlviz/
8. Barzdins, J., Barzdins, G., Cerans, K., Liepins, R., Sprogis, A.: OWLGrEd: a UML style graphical notation and editor for OWL 2. In: Proceedings of OWLED 2010 (2010)
9. Barzdiņš, J., Barzdiņš, G., Čerans, K., Liepiņš, R., Sprogis, A.: UML style graphical notation and editor for OWL 2. In: Forbrig, P., Günther, H. (eds.) BIR 2010. LNBIP, vol. 64, pp. 102–114. Springer, Heidelberg (2010)

[5] The extended editor is available as OWLGrEd/S from http://owlgred.lumii.lv/s.

10. Unified Modeling Language: Infrastructure, version 2.1. OMG Specification ptc/06-04-03. http://www.omg.org/docs/ptc/06-04-03.pdf
11. Unified Modeling Language: Superstructure, version 2.1. OMG Specification ptc/06-04-02. http://www.omg.org/docs/ptc/06-04-02.pdf
12. OWL 2 Manchester Syntax. http://www.w3.org/TR/owl2-manchester-syntax/
13. Barzdins, J., Cerans, K., Liepins, R., Sprogis, A.: Advanced ontology visualization with OWLGrEd. In: Proceedings of OWLED 2011 (2011)
14. Stardog. http://stardog.com/
15. Tao, J., Sirin, E., Bao, J., McGuinness, D.: Integrity constraints in OWL. In: Proceedings of AAAI 2010 (2010)
16. Sirin, E., Smith, M., Vallace, E: Opening, closing worlds – on integrity constraints. In: Proceedings of OWLED 2008 (2008)

Product Customization as Linked Data: Demonstration

Edouard Chevalier and François-Paul Servant(✉)

Renault SA, 13 Avenue Paul Langevin, 92359 Plessis Robinson, France
{edouard.chevalier,francois-paul.servant}@renault.com

Abstract. Exposing data about customizable products is a challenging issue, because of the number of features and options a customer can choose from, and the many constraints that exist between them. These constraints are not tractable without automatic reasoning. But the configuration process, which helps a customer to make her choice, one step at a time, is a traversal of a graph of partially defined products - that is, Linked Data. This natural yet fruitful abstraction for product customization results in a generic configuration API, in use at Renault, who has begun publishing data about its range in this way. Current achievements and prototypes of forthcoming developments are presented.

Keywords: Configuration · Customizable product · Linked data · Goodrelations · Automotive

1 Introduction

Publishing product data to improve e-business performance and visibility on the web is gaining momentum, thanks to vocabularies such as GoodRelations [1], or to the Schema.org initiative.

But exposing data about customizable products is a challenging issue. In industries practicing "Build to Order" of fully customizable products, ranges are huge, because of the number of features and options a customer can choose from: more than 10^{20} different cars are for sale at Renault. Ranges are also complex, because of the many constraints between features which invalidate some of their combinations: if every combination of distinctive features and options were possible, there would be 10^{25} different Renault cars, not our mere 10^{20} - meaning you have only one chance upon 100,000 to define an existing Renault car, if you choose its specifications without taking the constraints into account. Automatic reasoning is required to handle the constraints - a computationally hard task. It is possible to publish such range definitions on the web, including the constraints, by means of Semantic Web languages [2]. But it would not bring many practical results soon, as one cannot expect strong reasoning capabilities from client agents. This could hinder the publishing of descriptions of customizable product on the web of e-business data.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 425–429, 2015.
DOI: 10.1007/978-3-662-46641-4_39

This question is the subject of the paper "Product Customization as Linked Data", that will be presented in the "In Use and Industrial" track of this ESWC conference, and which the demonstration herein intends to illustrate. The paper describes how product configuration can be seen as a linked data application, and shows the benefits of this approach as to web e-business.

The crux of the paper centers around modeling the configuration process as the traversal of a graph of "Partially Defined Products" (PDP), or "Configurations", each configuration linking to those that refine it: for reference, a configurator is an application that helps a user interactively define a product step by step, each step describing a valid partially defined product, with a list of remaining choices given all previous selections. Each of these choices links to another PDP until completion. Thus, the configuration process traverses a graph whose nodes are PDPs. Now identify each PDP with a URI returning, among other relevant information, the list of the PDPs it is linked to: what you get is a description of the range as Linked Data.

Renault has begun to publish data about its range in this way, and this is the main subject of this demonstration, which intends:

- to show how to use the data returned by the Linked Data based configuration API to easily implement full-featured web configurators, such as the one which will be presented;
- to highlight some of the benefits of Linked Data modeling, in e-business related use cases.

As work at Renault is still in progress, the demonstration will mix presentations of systems that are in production and prototypes built upon the data returned by these systems. In both cases, we will focus on the (linked) data, which are the cornerstone of all this work and of services and applications that have been made available.

2 Linked Data Based Configuration at Renault

The Linked Data based Configuration API is implemented as a REST web service using Jersey[1].

This service is a facade in front of the configuration engine that provides the reasoning capabilities. Implementation of the latter, developed at Renault, is based on a compiled representation of the Constraint Satisfaction Problem that models the description of the range. Problems involving CSP are well-known to be computationally hard, but the hard part of the problem is fully solved in an offline compilation phase, guaranteeing bounded and fast response times for configuration related queries: time is linear on the size of the compiled representation, which happens to remain small enough [3].

We won't dwell any longer on this topic, as it is not the subject here: this demonstration, as well as the paper it refers to, are not about reasoning and

[1] http://jersey.java.net/.

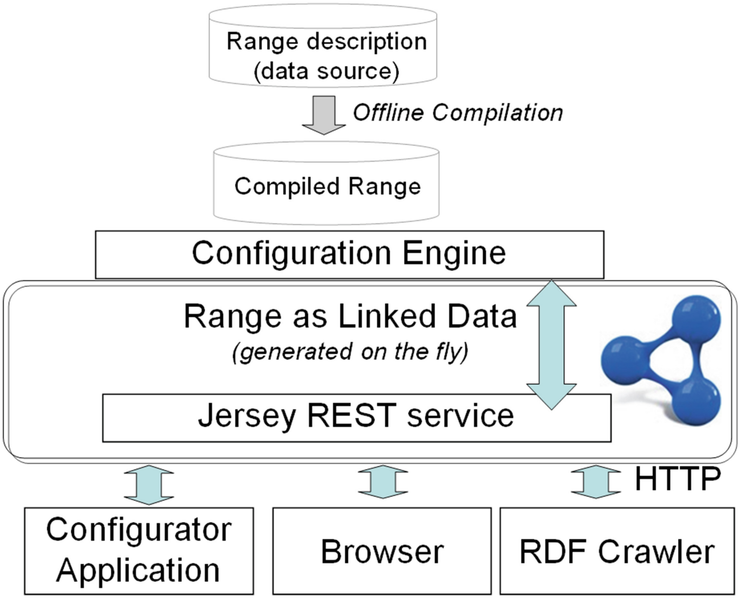


Fig. 1. Architecture.

the way it is implemented. In fact, one of the main points of this work is precisely about hiding the complexity of reasoning from clients. We'd like instead to remark that most, if not all, configurator applications on the web could be (re-)implemented as Linked Data: it is just a matter of wrapping in a REST service the configuration engine they use.

2.1 Related Work

The exact context of this work is the borderline where product configuration meets e-Commerce applications of Linked Data. The main work in the same context that we know of is Volkswagen's "Car Option Ontology"[2]. Their approach is different: they publish the constraints, in a proprietary vocabulary. We prefer to host the reasoning on the server, and free clients from the burden, ensuring maximal usability of the data that we provide.

2.2 Architecture of the Solution

It is presented in Fig. 1

The published data is the current commercial offer. It is managed by upstream systems, then "compiled" into the binary data used by the configuration engine (size: <100 MB). Linked Data is materialized on the fly when PDPs are queried (30 KB per PDP).

[2] http://purl.org/coo/ns.

3 Content of the Demonstration

3.1 Systems in Production

As of this writing (April 2012), the Linked Data service in production only returns JSON data, and only for German and Italian markets[3]. Depending on the pace of evolutions and deployments, more markets and/or data formats (RDF-XML, turtle) may be available by the time of the conference.

3.2 Accessing PDP Data and Implementation of Client Applications

Nevertheless all functionalities of Renault's configuration engine are already made accessible through the JSON data, including querying in free order, pricing information, filtering on a maximum price, negative choices, conflict resolution, completion, etc.,

This API - which, for the largest part, can be identified with the JSON data returned by the server when dereferencing the URIs of PDPs - therefore allows for the development of client applications that include configuration functionalities, and this was the first use of it envisioned at Renault. The development of such applications (in particular, a salesman assistant) are on their way.

One of the first goal of this demonstration, therefore, is to explain how to construct such applications from this data. This happens to be rather simple: for the web application developer, it is just a matter of displaying and/or following the links included in the data. As this data is published on the web, it can be used by developers outside of Renault: when such data is available from other automotive manufacturers, this is an opportunity to develop innovative mashups, such as range comparators.

3.3 RDF Data

A simple and generic ontology[4] describes the classes and properties involved in the modeling of the configuration process as Linked Data. We will show the inclusion of corresponding RDFa markup in the HTML pages of our prototype configurator application. This is interesting, because a configuration is a commercial offer (it can indeed always be completed to a valid product), and it can therefore be seamlessly described using GoodRelations [1], the "Web Vocabulary for E-Commerce".

3.4 Benefits in e-business Use Cases

Then, we intend to illustrate some of the benefits in e-business use cases. There, we will use a Proof Of Concepts implementation that has been widely used at Renault when promoting the solution. The main idea is the fact that identifying

[3] http://de.co.rplug.renault.com/docs, http://it.co.rplug.renault.com/docs.

[4] http://purl.org/configurationontology.

the PDPs with URIs allows their easy sharing between media and applications. In the demonstration, we'll show how a potential customer can begin a configuration by clicking on an ad or decoding a QR code in a billboard, modify it on her smartphone or PC; exchange it with members of her family; share it on FaceBook; have it transferred to the salesman's assistant when she finally goes to a shop, and have it converted to an order. See Fig. 2.

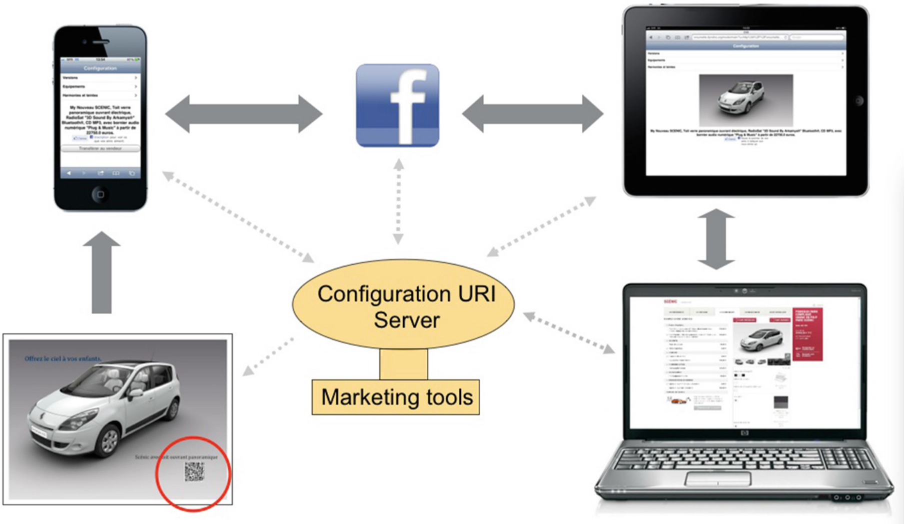


Fig. 2. Sharing PDPs between applications, devices and media.

4 Conclusion

The demonstration should succeed in showing that the challenging issue of publishing data about customizable products can be solved elegantly using Linked Data principles. We hope to convince people of the benefits of the method for e-business. They should be able to get started implementing such a solution, or developing applications using such data.

References

1. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. http://www.heppnetz.de/files/GoodRelationsEKAW2008-crc-final.pdf
2. Badra, F., Servant, F.P., Passant, A.: A semantic web representation of a product range specification based on constraint satisfaction problem in the automotive industry. In: OSEMA Workshop ESWC (2011). http://ceur-ws.org/Vol-748/paper4.pdf
3. Pargamin, B.: Vehicle sales configuration: the Cluster Tree Approach. In: ECAI Workshop on Configuration (2002)

Karma: A System for Mapping Structured Sources into the Semantic Web

Shubham Gupta, Pedro Szekely, Craig A. Knoblock(✉), Aman Goel,
Mohsen Taheriyan, and Maria Muslea

Information Sciences Institute and Department of Computer Science,
University of Southern California, Los Angeles, USA
{shubhamg,pszekely,knoblock,amangoel,mohsen,mariam}@isi.edu

1 Introduction

The Linked Data cloud contains large amounts of RDF data generated from databases. Much of this RDF data, generated using tools such as D2R, is expressed in terms of vocabularies automatically derived from the schema of the original database. The generated RDF would be significantly more useful if it were expressed in terms of commonly used vocabularies. Using today's tools, it is labor-intensive to do this. For example, one can first use D2R to automatically generate RDF from a database and then use R2R to translate the automatically generated RDF into RDF expressed in a new vocabulary. The problem is that defining the R2R mappings is difficult and labor intensive because one needs to write the mapping rules in terms of SPARQL graph patterns.

In this work, we present a semi-automatic approach for building mappings that translate data in structured sources to RDF expressed in terms of a vocabulary of the user's choice. Our system, Karma, automatically derives these mappings, and provides an easy to use interface that enables users to control the automated process to guide the system to produce the desired mappings. In our evaluation, users need to interact with the system less than once per column (on average) in order to construct the desired mapping rules. The system then uses these mapping rules to generate semantically rich RDF for the data sources.

We demonstrate Karma using a bioinformatics example and contrast it with other approaches used in that community. Bio2RDF [7] and Semantic MediaWiki Linked Data Extension (SMW-LDE) [2] are examples of efforts that integrate bioinformatics datasets by mapping them to a common vocabulary. We applied our approach to a scenario used in the SMW-LDE that integrate ABA, Uniprot, KEGG Pathway, PharmGKB and Linking Open Drug Data datasets using a

This research is based upon work supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 430–434, 2015.
DOI: 10.1007/978-3-662-46641-4_40

common vocabulary. In this demonstration, we first show how a user can interactively map these datasets to the SMW-LDE vocabulary, and then we use these mappings to generate RDF for these sources.

2 Application: Karma

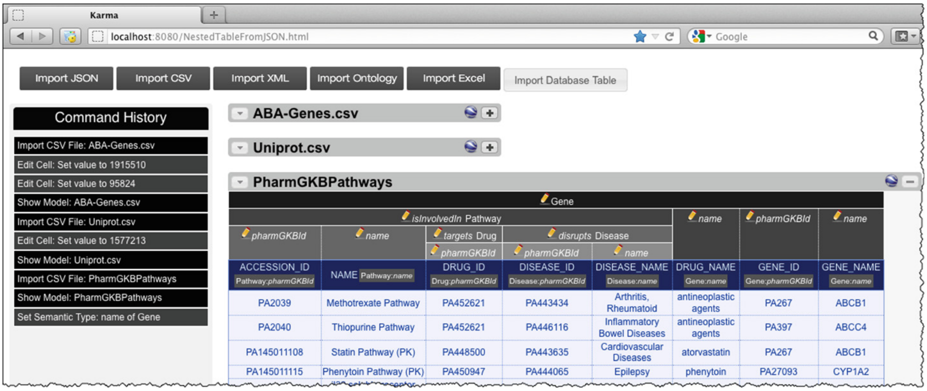
Karma[1] is a web application that enables users to perform data-integration tasks by example [8]. Karma provides support for extracting data from a variety of sources (relational databases, CSV files, JSON, and XML), for cleaning and normalizing data, for modeling it according to a vocabulary of the user's choice, for integrating multiple data sources, and for publishing in a variety of formats (CSV, KML, and RDF). In this demonstration we focus on the capabilities to interactively model sources according to a chosen vocabulary and to publish data in RDF.

The modeling process takes as input a vocabulary defined in an OWL ontology, one or more data sources to be modeled, and a database of semantic types learned in previous modeling sessions. It outputs a formal mapping between the source and the ontology that can be then used to generate RDF. The key technologies that this process exploits are the learning of semantic types using conditional random fields (CRF) [6] and a Steiner tree algorithm to compute the relationships among the schema elements of a source.

Semantic types characterize the meaning of data. For example, consider a dataset with a column containing PharmGKB accession identifiers for pathways. The syntactic type of the values is *String*. In our formulation, we represent their semantic type as a pair consisting of the class Pathway and the property pharmGKBId to capture the idea that these values are a particular type of pathway identifier. In RDF terms, the values are the objects of triples whose subject is of type Pathway and whose property is pharmGKBId. Karma infers semantic types automatically using the semantic types it has been trained to recognize. When Karma is unable to infer the semantic type for a column, users can interactively assign the desired type; Karma uses the assigned type and the data in the column to train a CRF model to recognize the type in the future [4]. The semantic types are used by our Steiner tree algorithm to compute the source model as the minimum tree that connects the assigned semantic types via properties in the ontology (the details of the approach are published elsewhere [5]). Because the minimum model is not always the desired model, Karma provides a user interface to enable users to force this algorithm to include specific properties in the model.

Most of the existing mapping generation tools, such as Clio [3], Altova MapForce (altova.com), or NEON's ODEMapster [1], rely on the user to manually specify the mappings in a graphical interface. In contrast, Karma provides a semi-automatic approach to achieve the same objective, enabling domain experts (and not just DB administrators or ontology engineers) to specify the mappings.

[1] https://github.com/InformationIntegrationGroup/Web-Karma-Public.



Command History

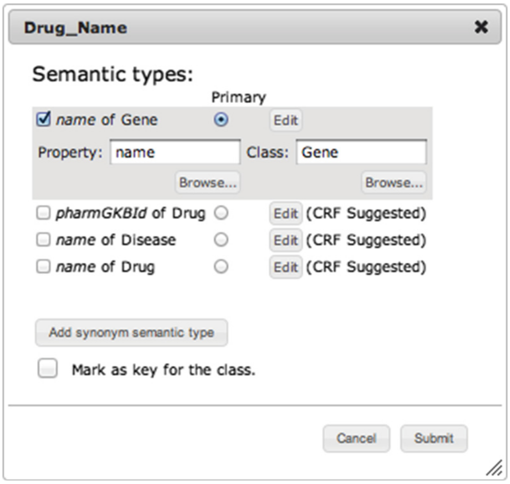
- Import CSV File: ABA-Genes.csv
- Edit Cell: Set value to 1915510
- Edit Cell: Set value to 95824
- Show Model: ABA-Genes.csv
- Import CSV File: Uniprot.csv
- Edit Cell: Set value to 1577213
- Show Model: Uniprot.csv
- Import CSV File: PharmGKBPathways
- Show Model: PharmGKBPathways
- Set Semantic Type: name of Gene

Gene							
isInvolvedIn Pathway					name	pharmGKBId	name
pharmGKBId	name	targets Drug	disrupts Disease				
		pharmGKBId	pharmGKBId	name			
ACCESSION_ID Pathway:pharmGKBId	NAME Pathway:name	DRUG_ID Drug:pharmGKBId	DISEASE_ID Disease:pharmGKBId	DISEASE_NAME Disease:name	DRUG_NAME Gene:name	GENE_ID Gene:pharmGKBId	GENE_NAME Gene:name
PA2039	Methotrexate Pathway	PA452621	PA443434	Arthritis, Rheumatoid	antineoplastic agents	PA267	ABCB1
PA2040	Thiopurine Pathway	PA452621	PA446116	Inflammatory Bowel Diseases	antineoplastic agents	PA397	ABCC4
PA145011108	Statin Pathway (PK)	PA448500	PA443635	Cardiovascular Diseases	atorvastatin	PA267	ABCB1
PA145011115	Phenytoin Pathway (PK)	PA450947	PA444065	Epilepsy	phenytoin	PA27093	CYP1A2

Fig. 1. Karma workspace showing a bioinformatics source and its model (color figure online).

3 Demonstration

In this demonstration, we first show how users model structured sources according to an ontology they select; then we show how Karma can use the model to generate RDF represented using the classes and properties defined in the ontology. We will illustrate the process using a bioinformatics example.



Semantic types:

Primary

☑ *name* of Gene ● Edit

Property: name Class: Gene

Browse... Browse...

□ *pharmGKBId* of Drug ○ Edit (CRF Suggested)

□ *name* of Disease ○ Edit (CRF Suggested)

□ *name* of Drug ○ Edit (CRF Suggested)

□ Mark as key for the class.

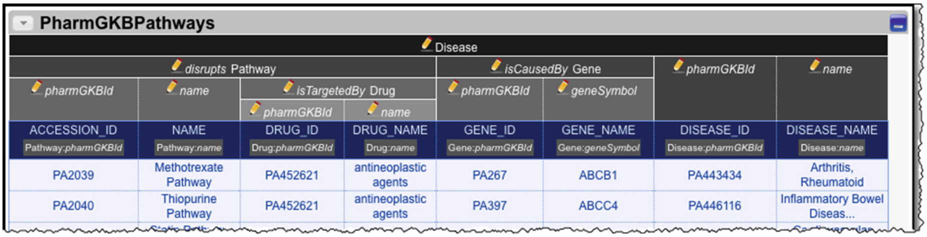
Fig. 2. Semantic type selection dialog box.

In the first part of the demonstration we provide an overview of the Karma workspace (Fig. 1) and show how to import data into Karma.

In the second part we show the model that Karma automatically infers for a source. Karma builds the initial model using the existing database of semantic types and visualizes it as hierarchical headings over the worksheet data. The inferred semantic types are shown in the grey boxes nested inside the dark blue boxes that show the column names.

In the third part we show how users can adjust the automatically generated model. We show how users can fix incorrectly assigned semantic types, and how users can adjust the model when Karma infers incorrect relationships between columns.

In our example shown in Fig. 1, when the user loads the source, Karma incorrectly assigns the semantic type Gene.name to the DRUG_NAME column. To correct the problem, users click on the semantic type to bring up the semantic



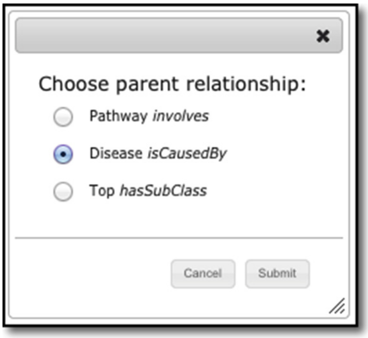
PharmGKBPathways

Disease							
disrupts Pathway				*isCausedBy* Gene		*pharmGKBId*	*name*
pharmGKBId	*name*	*isTargetedBy* Drug		*pharmGKBId*	*geneSymbol*		
		pharmGKBId	*name*				
ACCESSION_ID Pathway:*pharmGKBId*	NAME Pathway:*name*	DRUG_ID Drug:*pharmGKBId*	DRUG_NAME Drug:*name*	GENE_ID Gene:*pharmGKBId*	GENE_NAME Gene:*geneSymbol*	DISEASE_ID Disease:*pharmGKBId*	DISEASE_NAME Disease:*name*
PA2039	Methotrexate Pathway	PA452621	antineoplastic agents	PA267	ABCB1	PA443434	Arthritis, Rheumatoid
PA2040	Thiopurine Pathway	PA452621	antineoplastic agents	PA397	ABCC4	PA446116	Inflammatory Bowel Diseas...

Fig. 3. Source model for PharmGKB Pathways data before model refinement.

type specification dialog (Fig. 2). The dialog shows the top options computed by the CRF model. When the correct option is in the list, users can select it with a single click. Otherwise, users specify the class and property by typing it (with completion) or by selecting the appropriate class or property from an ontology browser. In our example, the correct semantic type Drug.name is the fourth option. After each adjustment to the semantic types, Karma retrains the CRF model and invokes the Steiner tree algorithm to recalculate the set of properties that tie together the semantic types. Figure 3 shows the updated model incorporating the user changes.

The model proposed by Karma in Fig. 3 is not correct because it specifies that the Gene columns contain information about genes that *cause* the disease described in the Disease columns (it models the relationship using the isCausedBy property). The correct model is that the genes are involved in the pathways that are disrupted by the disease. Users can specify the correct properties by clicking on the pencil icons.



Choose parent relationship:

○ Pathway *involves*

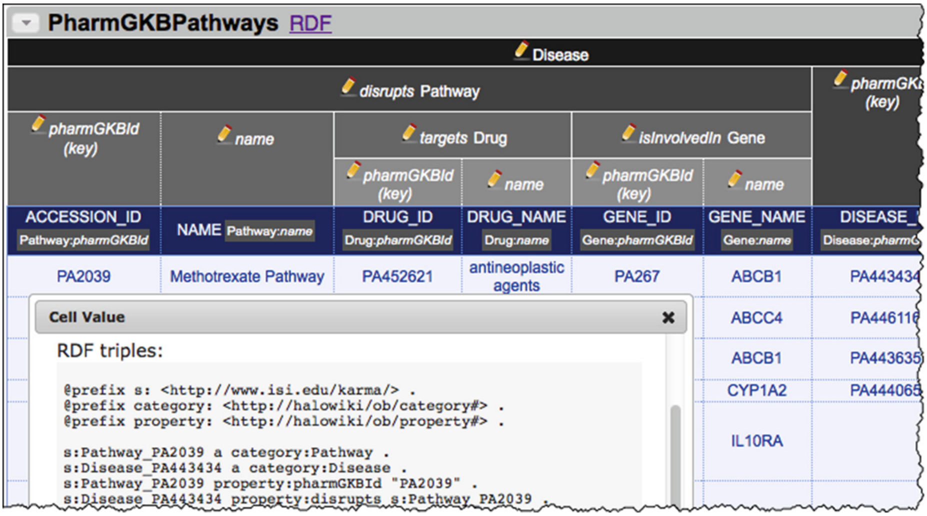
● Disease *isCausedBy*

○ Top *hasSubClass*

Fig. 4. Relationship selection dialog box.

Figure 4 shows the pop-up that appears by clicking on the pencil icon on the isCaused By Gene cell. The pop-up shows *domain/ property* pairs that satisfy two conditions. First, the class *domain* is a valid domain for the *property* and second, the class the user clicked (Gene in our example) is a valid range for the *property*. In our example, the correct choice is the first one because the information in the table is about Pathways that involve our Gene. After users make a selection, Karma recomputes the Steiner tree, which is now required to include the class/property selections users make [5]). Figure 5 shows the correct, updated model.

In the last part of the demonstration we show the RDF generation process. Once users are satisfied with the source model, they can generate and download



Disease						
disrupts Pathway						pharmGK (key)
pharmGKBId (key)	name	targets Drug		isInvolvedIn Gene		
		pharmGKBId (key)	name	pharmGKBId (key)	name	
ACCESSION_ID Pathway:pharmGKBId	NAME Pathway:name	DRUG_ID Drug:pharmGKBId	DRUG_NAME Drug:name	GENE_ID Gene:pharmGKBId	GENE_NAME Gene:name	DISEASE_ Disease:pharmG
PA2039	Methotrexate Pathway	PA452621	antineoplastic agents	PA267	ABCB1	PA443434
					ABCC4	PA44611
					ABCB1	PA443635
					CYP1A2	PA444065
					IL10RA	

RDF triples:

```
@prefix s: <http://www.isi.edu/karma/> .
@prefix category: <http://halowiki/ob/category#> .
@prefix property: <http://halowiki/ob/property#> .

s:Pathway_PA2039 a category:Pathway .
s:Disease_PA443434 a category:Disease .
s:Pathway_PA2039 property:pharmGKBId "PA2039" .
s:Disease_PA443434 property:disrupts s:Pathway_PA2039 .
```

Fig. 5. RDF generation with Karma.

the RDF for the whole source or view the RDF generated for a single cell (Fig. 5). A movie of the whole user-interaction process is available online[2].

References

1. Barrasa-Rodriguez, J., Gómez-Pérez, A.: Upgrading relational legacy data to the semantic web. In: Proceedings of WWW Conference, pp. 1069–1070 (2006)
2. Becker, C., Bizer, C., Erdmann, M., Greaves, M.: Extending smw+ with a linked data integration framework. In: Proceedings of ISWC (2010)
3. Fagin, R., Haas, L.M., Hernández, M., Miller, R.J., Popa, L., Velegrakis, Y.: Clio: schema mapping creation and data exchange. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Mylopoulos Festschrift. LNCS, vol. 5600, pp. 198–236. Springer, Heidelberg (2009)
4. Goel, A., Knoblock, C.A., Lerman, K.: Using conditional random fields to exploit token structure and labels for accurate semantic annotation. In: Proceedings of AAAI-11 (2011)
5. Knoblock, C.A., et al.: Semi-automatically mapping structured sources into the semantic web. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 375–390. Springer, Heidelberg (2012)
6. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)
7. Peter, A.: Model and prototype for querying multiple linked scientific datasets. Future Gener. Comput. Syst. **27**(3), 329–333 (2011). http://www.sciencedirect.com/science/article/pii/S0167739X10001706
8. Tuchinda, R., Knoblock, C.A., Szekely, P.: Building mashups by demonstration. ACM Trans. Web (TWEB) **5**(3), 1–50 (2011)

[2] http://www.isi.edu/integration/videos/karma-source-modeling.mp4.

Personalized Environmental Service Configuration and Delivery Orchestration: The PESCaDO Demonstrator

Leo Wanner[1,2], Marco Rospocher[8](✉), Stefanos Vrochidis[6], Harald Bosch[3], Nadjet Bouayad-Agha[2], Ulrich Bügel[4], Gerard Casamayor[2], Thomas Ertl[3], Desiree Hilbring[4], Ari Karppinen[5], Ioannis Kompatsiaris[6], Tarja Koskentalo[7], Simon Mille[2], Jürgen Moßgraber[4], Anastasia Moumtzidou[6], Maria Myllynen[7], Emanuele Pianta[8], Horacio Saggion[2], Luciano Serafini[8], Virpi Tarvainen[5], and Sara Tonelli[8]

[1] Catalan Institute for Research and Advanced Studies, Barcelona, Spain
[2] Department of Information and Communication Technologies,
Pompeu Fabra University, Barcelona, Spain
pescado@upf.edu
http://www.pescado-project.eu
[3] Visualization Institute, University of Stuttgart, Stuttgart, Germany
[4] Fraunhofer Institute for Optronics, System Technologies and Image Exploitation,
Karlsruhe, Germany
[5] Finnish Meteorological Institute, Helsinki, Finland
[6] Centre for Research and Technology Hellas, Informatics and Telematics Institute,
Thessaloniki, Greece
[7] Helsinki Region Environmental Services Authority, Helsinki, Finland
[8] Fondazione Bruno Kessler, Trento, Italy
rospocher@fbk.eu

Abstract. Citizens are increasingly aware of the influence of environmental and meteorological conditions on the quality of their life. This results in an increasing demand for personalized environmental information, i.e., information that is tailored to citizens' specific context and background. In this demonstration, we present an environmental information system that addresses this demand in its full complexity in the context of the PESCaDO EU project. Specifically, we will show a system that supports submission of user generated queries related to environmental conditions. From the technical point of view, the system is tuned to discover reliable data in the web and to process these data in order to convert them into knowledge, which is stored in a dedicated repository. At run time, this information is transferred into an ontology-based knowledge base, from which then information relevant to the specific user is deduced and communicated in the language of their preference.

1 Research Background

Citizens are increasingly aware of the influence of environmental and meteorological conditions on the quality of their life. One of the consequences of this

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 435–440, 2015.
DOI: 10.1007/978-3-662-46641-4_41

awareness is the demand for high quality environmental information that is tailored to one's specific context and background (e.g. health conditions, travel preferences, etc.), i.e., which is personalized. Personalized environmental information may need to cover a variety of aspects (such as meteorology, air quality, pollen, and traffic) and take into account a number of specific personal attributes (health, age, allergies, etc.) of the user, as well as the intended use of the information. For instance, a pollen allergic person, planning to do some outdoor activities, may be interested in being notified whether the pollen situation in the area may trigger some symptoms, or if the temperature is too hot for doing physical exercise, while a city administrator has to be informed whether the current air quality situation requires some actions to be urgently taken.

So far, only a few approaches have been proposed with a view of how this information can be facilitated in technical terms. All of these approaches focus on one environmental aspect and only very few of them address the problem of information personalization [2,7,9]. We aim to address the above task in its full complexity.

In this work, carried on in the context of the PESCaDO EU project, we take advantage of the fact that nowadays, the World Wide Web already hosts a great range of services (i.e. websites, which provide environmental information) that offer data on each of the above aspects, such that, in principle, the required basic data are available. The challenge is threefold: first, to discover and orchestrate these services; second, to process the obtained data in accordance with the needs of the user; and, third, to communicate the gained information in the users preferred mode.

The demonstration will aim, in particular, at showing how semantic web technologies are exploited to address this challenges in PESCaDO.

2 The PESCaDO Platform: Main Modules and Key Semantic Technologies Used

The challenges mentioned in Sect. 1 require the involvement of an elevated number of rather heterogeneous applications addressing various complex tasks: discovery of the environmental service nodes in the web, distillation of the data from webpages, orchestration of the environmental service nodes, fusion of environmental data, assessment of the data with respect to the needs of the addressee, selection of user-relevant content and its delivery to the addressee, and, finally, interaction with the user. Thus, in PESCaDO we developed a service-based infrastructure to integrate all these applications.

For a general overview of the running PESCaDO service platform[1], and the type of information produced, see: http://www.youtube.com/watch?v=c1Ym7ys3HCg. In this section, we focus on presenting three tasks we addressed by applying semantic web technologies.

The back-bone of the PESCaDO service platform, exploited in each of these three tasks, is an ontology-based knowledge base, the PESCaDO Knowledge

[1] A more comprehensive description of the system workflow can be found in [10].

Base (PKB), where all the information relevant for a user request are dynamically instantiated. The ontology, partially built exploiting automatic key-phrases extraction techniques [8], formalizes a variety of aspects related to the application context: environmental data, environmental nodes[2], user requests, user profiles, warnings and recommendations triggered by environmental conditions, logico-semantic relations (e.g. cause, implication) between facts, and so on. The current version of the ontology consists of 241 classes, 672 individuals, 151 object properties, and 43 datatype properties.

2.1 Discovery of Environmental Nodes

The first step towards the extraction and indexing of environmental information is the discovery of environmental nodes, which can be considered as a problem of domain specific search. To this end, we implement a node discovery framework, which builds upon state of the art domain specific search techniques, advanced content distillation, ontologies and supervised machine learning. The framework consists of three main parts: (a) Web search (b) Post processing and (c) Indexing and storage. Web search is realized with the aid of a general-purpose search engine, which accesses large web indices. In this implementation we employ Yahoo! Search BOSS API. In order to generate domain specific queries, we apply two complementary techniques. First we use the ontology of the PKB and we extract concepts and instances referring to types of environmental data (e.g. temperature, birch pollen, PM_{10}) and we combine them with geographical city names automatically retrieved by geographical resources. In addition, the queries are expanded by keyword spices [6], which are domain specific keywords extracted with the aid of machine learning techniques from environmental websites.

During the post-processing step we perform supervised classification with Support Vector Machines to separate relevant from irrelevant nodes and we crawl each website to further expand our search in an iterative manner. The determination of the relevance of the nodes and their categorization is done using a classifier that operates on a weight-based vector of key phrases and concepts from the content and the structure of the webpages. Subsequently, we parse the body and the metadata of the relevant webpages in order to extract the structure and the clues that reveal the information presented.

Finally, the information obtained with respect to each relevant node is indexed in a Sensor Observation Service (SOS) [5] compliant repository, which can be accessed and retrieved by the system when a user request is submitted.

The whole discovery procedure is automatic, however an administrative user could intervene through an interactive user interface, in order to select geographic regions of interest to perform the discovery, optimize the selection of keyword spices, and parametrize the training of the classifiers.

[2] An environmental node is a provider of environmental data values, like for instance a web-site, a web-service, or a measuring station.

2.2 Processing Raw Environmental Data to Obtain Content

The user interface of the PESCaDO system guides the user in formulating a request, which is instantiated in all its details (e.g. type of request, user profile, time period, geographic location) in the PKB. By exploiting Description Logics (DL) reasoning on the PKB, the system determines from the request description which are the types of environmental data which constitute the raw content necessary to fulfil the user needs. A specific component of the system is then responsible of selecting from the SOS repository the actual values (observed, forecasted, historical) for the selected types of environmental data, and to appropriately instantiate them in the PKB.

At this stage, the raw data retrieved from the environmental nodes are processed to derive additional personalized content from them, such as data aggregations, qualitative scaling of numerical data, and user tailored recommendations and warnings triggered by the environmental data relevant for the specific user query. Logico-semantic relations are also instantiated at this stage, for instance to represent whether a certain pollen concentration value causes the triggering of a recommendation to the user, due to its sensitiveness to that pollen.

The computation of this inferred content is performed by the *decision support* service of the PESCaDO Platform by combining some complementary reasoning strategies, including DL reasoning and rule-based reasoning. A two layer reasoning infrastructure is currently in place. The first layer exploits the HermiT reasoner for the OWL DL reasoning services. The second layer is stacked on top of the previous layer. It uses the Jena RETE rule engine, which performs the rule-based reasoning computation.

2.3 Generating User Information from Content

As is common in Natural Language Generation, our information generator is divided into two major modules: the text planning module and the linguistic generation module (with the latter taking as input the *text plan* produced by the former).

Text Planning. The text planning module is divided into a content selection module and discourse structuring module. As is common in report generation, our content selection is schema- (or template-) based. Therefore, the ontology of the PKB introduced above defines a class `Schema` with an n-ary schema component object property whose range can be any individuals of the PKB itself.

Similar to [1], we assume the output of the discourse structuring module to be a well-formed text plan which consists of (i) elementary discourse units (EDUs) that group together individuals of the PKB, (ii) discourse relations between EDUs and/or individuals of the PKB, and (iii) precedence relations between EDUs. This structure translates into two top classes of the ontology of the PKB: `EDU` with an n-ary EDU component relation and a linear precedence property, and `Discourse Relation` with nucleus and satellite relation. A set of SPARQL query rules are defined to instantiate the various concepts and relations.

Content Selection (CS) operates on the output of the decision support service. It selects the content to be included in the report and groups it by topic, instantiating a number of schemas for each topic. The inclusion of a given individual in a schema can be subject to some restrictions defined in the queries; for example, if the minimum and maximum air quality index (AQI) values are identical, or if the maximum AQI value triggers a user recommendation or warning, then only the maximum AQI value is selected (the minimum AQI rating is omitted).

Discourse structuring is carried out by a pipeline of three rule-based submodules: (i) Elementary Discourse Unit (EDU) Determination, which groups topically related PKB individuals into propositional units starting from the schemas determined during CS; (ii) Mapping logico-semantic relations to discourse relations; and (iii) EDU Ordering, which introduces a precedence relation between EDUs using a number of heuristics derived from interviews with domain communication experts.

Linguistic generation. Our linguistic generation module is based on a multilevel linguistic model of the Meaning-Text Theory (MTM) [4], such that the generation consists of a series of mappings between structures of adjacent strata (from the conceptual stratum to the linguistic surface stratum): *Conceptual Structure* (*ConStr*) ⇒ *Semantic Structure* (*SemStr*) ⇒ *Deep-Syntactic Structure* (*DSyntStr*) ⇒ *Surface-Syntactic Structure* (*SSyntStr*) ⇒ *Deep-Morphological Structure* (*DMorphStr*) ⇒ *Surface-Morphological Structure* (*SMorphStr*) ⇒ *Text*. Starting from the conceptual stratum, for each pair of adjacent strata $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$, a transition grammar $\mathcal{G}^i_{i+1}$ is defined; see [3].

The ConStr is derived from each text plan produced by the text planning component. In a sense, ConStr can thus be considered a projection of selected fragments of the ontologies onto a linguistically motivated structure. ConStrs are language-independent and thus ideal as starting point of multilingual generation.

3 System Demonstration

The system demonstration will show how the PKB is instantiated and exploited by the different services composing the PESCaDO Platform in the context of two different application scenarios, one about health safety decision support for end users and one about administrative decision support. In particular, the demo attendees will have the chance to see how the raw environmental data are dynamically processed with ontology-based techniques to obtain reports. Furthermore, we will demonstrate how to use and set-up the tool for environmental node discovery.

Acknowledgments. The work described in this paper has been partially funded by the European Commission under the contract number FP7-248594, PESCaDO (Personalized Environmental Service Configuration and Delivery Orchestration) project.

References

1. Bouayad-Agha, N., Casamayor, G., Wanner, L., Díez, F., López Hernández, S.: FootbOWL: using a generic ontology of football competition for planning match summaries. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 230–244. Springer, Heidelberg (2011)
2. Karatzas, K.D.: State-of-the-art in the dissemination of aq information to the general public. In: Proceedings of EnviroInfo, pp. 41–47 (2007)
3. Lareau, F., Wanner, L.: Towards a generic multilingual dependency grammar for text generation. In: King, T., Bender, E.M. (eds.) Proceedings of the GEAF07 Workshop, pp. 203–223. CSLI, Stanford (2007)
4. Mel'čuk, I.A.: Dependency Syntax: Theory and Practice. SUNY Press, Albany (1988)
5. North. Sensor observation service (sos) (2004)
6. Oyama, S., Kokubo, T., Ishida, T.: Domain-specific web search with keyword spices. IEEE Trans. Knowl. Data Eng. **16**(1), 17–27 (2004)
7. Peinel, G., Rose, T., San José, R.: Customized information services for environmental awareness in urban areas. In: Proceedings of the 7th World Congress on Intelligent Transport Systems, Turin, Italy (2000)
8. Tonelli, S., Rospocher, M., Pianta, E., Serafini, L.: Boosting collaborative ontology building with key-concept extraction. In: Proceedings of 5th IEEE International Conference on Semantic Computing, (September 18–21, 2011 - Palo Alto, USA) (2011)
9. Wanner, L., Bohnet, B., Bouayad-Agha, N., Lareau, F., Nicklaß, D.: MARQUIS: generation of user-tailored multilingual air quality bulletins. Appl. Artif. Intell. **24**(10), 914–952 (2010)
10. Wanner, L., Vrochidis, S., Tonelli, S., Moßgraber, J., Bosch, H., Karppinen, A., Myllynen, M., Rospocher, M., Bouayad-Agha, N., Bügel, U., Casamayor, G., Ertl, T., Kompatsiaris, I., Koskentalo, T., Mille, S., Moumtzidou, A., Pianta, E., Saggion, H., Serafini, L., Tarvainen, V.: Building an environmental information system for personalized content delivery. In: Hřebíček, J., Schimak, G., Denzer, R. (eds.) Environmental Software Systems. IFIP AICT, vol. 359, pp. 169–176. Springer, Heidelberg (2011)

Supporting Rule Generation and Validation on Environmental Data in EnStreaM

Alexandra Moraru[1,3](✉), Klemen Kenda[1], Blaž Fortuna[1],
Luka Bradeško[1], Maja Škrjanc[1], Dunja Mladenić[1,3],
and Carolina Fortuna[2]

[1] Artificial Intelligence Laboratory, Jožef Stefan Institute, Ljubljana, Slovenia
{alexandra.moraru,klemen.kenda,blaz.fortuna,
luka.bradesko,maja.skrjanc,dunja.mladenic}@ijs.si
[2] Department of Communication Systems, Jožef Stefan Institute,
Ljubljana, Slovenia
carolina.fortuna@ijs.si
[3] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

Abstract. Detection rules represent one of the components of the rule models in event processing systems. These rules can be discovered from data using data mining techniques or domain experts' knowledge. We demonstrate a system that provides its users the means for creating and validating such rules. The system is applied on real-life environmental scenarios, where the main source of data comes from sensors. Based on historical data about events of interest, the scope is to formulate rules that could have caused these events. Using a scalable infrastructure the rules can be tested on massive amount of data in order to observe how past events would fit to these rules. In addition, we create semantic annotations of the dataset and use them in the system outputs in order to support interoperability with other systems.

Keywords: Visual analytics · Sensor data · Rule model · Semantic annotations

1 Introduction

The avalanche of data which information systems have to face nowadays influences their evolution and characteristics. One such family of systems, called information flow processing (IFP) systems [1], refers to data stream management systems and complex event processing systems. Such systems are able to handle multiple data sources, often streams, by applying a set of processing rules in order to derive new knowledge. These rules can be discovered using data mining and machine learning techniques from a vast research area [2, 3] or they can be defined by domain experts based on their knowledge. For the second case, an example can be related to landslides phenomena, for which an expert already knows the causes producing landslides. Many of these situations follow specific patterns which can be expressed through rules. The next step to represent these rules in a format which can be used by information systems is to provide to the experts an environment where they can create and validate the rules.

We demonstrate a system which can be used by domain experts to explore large datasets in order to define processing rules for environmental data. The rules can be

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 441–446, 2015.
DOI: 10.1007/978-3-662-46641-4_42

created and validated on real datasets through a graphical user interface (GUI). Similar work can be found on visual pattern discovery [4], where the focus is set on time series visualization for detection of unknown events. In contrast, we consider the situation when the events of interest are already known and the possible causes of these events can be explored.

Our system uses EnStreaM infrastructure which is based on tightly integrated and scalable custom software modules. In addition, the indexing of the data is application-oriented, specific and therefore extremely efficient, allowing development of various applications, such as real-time mashups [5]. For interoperability with other systems, rules created in our system are exported in RuleML[1] format, using concepts from OpenCyc[2] for relations' names. Furthermore, the datasets to which the rules apply are exported in RDF format and annotated with OpenCyc concepts.

The demonstration of the system consists of live running EnStreaM platform with which the visitors can interact through the GUI. For illustrating the functionalities of the system a landslide use case is prepared as presented in Sect. 2.2. Furthermore, the standardized exports of the systems can be presented for those interested.

2 EnStreaM

EnStreaM is a scalable system which implements efficient storage and retrieval methods for handling large amounts of data, both static and dynamic [6]. It is used in the ENVISION[3] project for data stream mining tasks, for several environmental scenarios

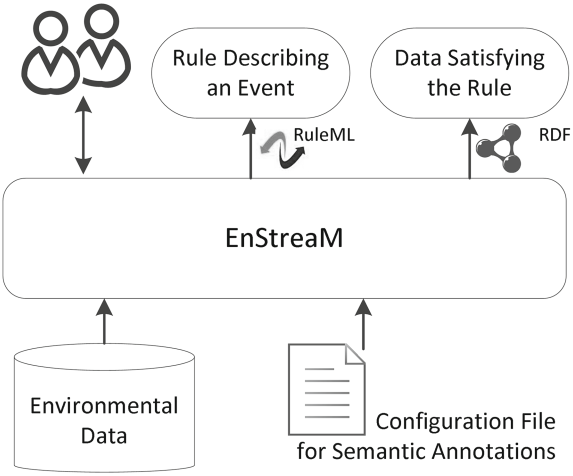


Fig. 1. EnStreaM – overall architecture

[1] http://ruleml.org/.

[2] http://www.opencyc.org.

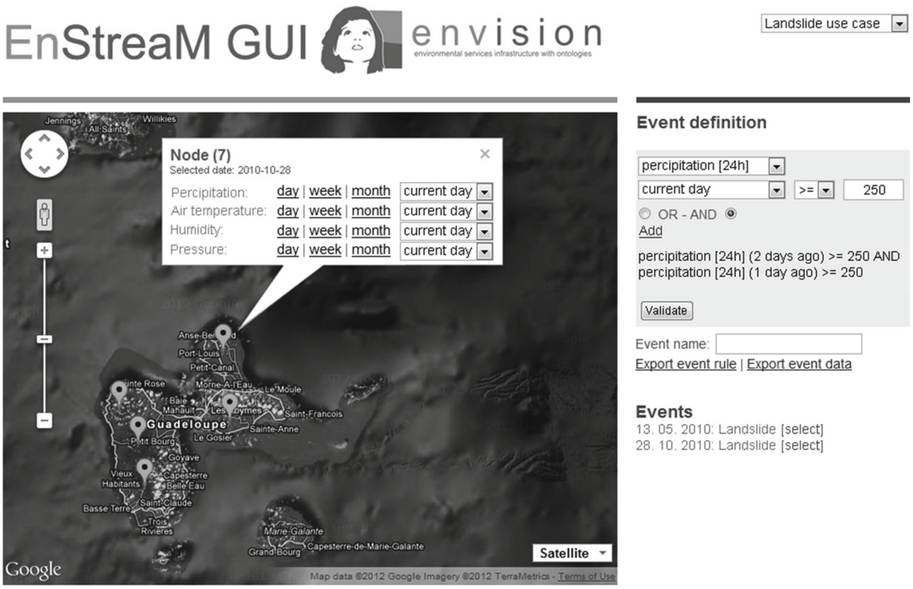
[3] http://www.envision-project.eu/.

related to landslides, oil spills and river floods. A high level architecture illustrating the inputs and outputs of EnStreaM used in our demonstration is presented in Fig. 1 and discussed in the flowing subsection.

2.1 System Overview

The input for EnStreaM consists of environmental datasets and configuration files used for annotating the datasets with ontology concepts. The environmental data is composed mostly of sensor measurements of the relevant properties from the area of interest (e.g. volume of rainfall for a given geographical location) and events that have occurred in the past. The metadata attached to the sensor measurements is providing the context needed for understanding these measurements. The sensor measurements have a dynamic nature, while the metadata associated is static. For efficient data management, these two types of data are stored internally in EnStreaM using specialized indexing methods. In order to provide domain experts the possibility to explore archived data in an ad hoc manner, we index data based on different aspects (e.g. location, date of measurement) and also provide numerous aggregates (sum, min, max, mean, standard variation, etc.). A unified view over different sources of sensor data is created through semantic annotations, based on a configuration file which maps the internal structure of EnStreaM stores to concepts from an ontology.

The abstraction layer provided by the semantic annotations and aggregation of data enables the domain experts to analyze historical data in order to find various patterns. These patterns are represented by rules which can be created and tested through the GUI of the system (see Fig. 2). The process of rule creation can be done in repetitive



Landslide use case ▾

Node (7)
Selected date: 2010-10-28

Percipitation:	day | week | month	current day ▾
Air temperature:	day | week | month	current day ▾
Humidity:	day | week | month	current day ▾
Pressure:	day | week | month	current day ▾

Event definition

percipitation [24h] ▾
current day ▾ >= ▾ 250
○ OR - AND ●
Add

percipitation [24h] (2 days ago) >= 250 AND
percipitation [24h] (1 day ago) >= 250

Validate

Event name:
Export event rule | Export event data

Events
13. 05. 2010: Landslide [select]
28. 10. 2010: Landslide [select]

Fig. 2. EnStreaM User Interface

steps in which the user can refine or add new parameters. For validating the rule, the user can test it on the historical data. Finally, the rule can be exported in RuleML format and the dataset complying with the rules can be exported in RDF format.

2.2 Use Case Scenario

To continue with our example from the introduction, let us consider that a landslide domain expert knows that some amount of raindrop can be an alarm for an eminent landslide. For illustration purposes we can consider that a pattern for this is represented by the following rule: *if the amount of rainfall exceeds 250 mm per day in 3 consecutive days then a landslide can occur.* Based on historical data gathered from rain gauge sensors, together with events when landslides have occurred in the past, the validity of such a rule can be verified.

2.2.1 Creation and Validation of Rules

The user can start by analyzing the events which have occurred in the past, listed in the bottom right corner of the interface. Next, the sensors related to the event selected are displayed on the map based on their geographical location. The sensor measurements can be visualized for different time periods as illustrated in Fig. 2. Next, the fields on the right-hand side of the GUI are used to specify the relations and operators to appear in the rules. For our example we have three relations in conjunction (the logic operators supported are “AND” and “OR”) which constitute the conditions of the rule. The result of such conditions being fulfilled represents a type of event, whose name is given by the user in the “Event name” field. The validation step is done by running the query with all the conditions specified over the historical data and comparing the events returned by the query with the list of entire events available for the specified location. The user should decide the importance and quality of the rule defined.

The rules created through the interface are exported in RuleML Datalog format, which provides a simple and clean syntax for expressing “if-then” rules. Each condition is represented by one or more atomic formulas (“Atom”). For example the condition

```
<And> <Atom> <op> <Rel iri="openCyc:sensorObservation"/> </op>
       <Var> sensor </Var>
       <Ind iri="openCyc:Raindrop"/> </Atom>
<Atom>n <op> <Rel iri="openCyc:doneBy"/> </op>
       <Var> sensor </Var>
       <Var> measurement </Var> </Atom>
<Atom> <op> <Rel iri="openCyc:measurementResult"/>  </op>
       <Var> measurement </Var>
       <Var> val1 </Var> </Atom>
<Atom> <op> <Rel iri="openCyc:duration"/> </op>
        <Var> measurement </Var>
       <Ind type="xs:time">24:00:00</Ind> </Atom>
<Atom> <op> <Rel iri="openCyc:greaterThanOrEqualTo"/>  </op>
       <Var> val1 </Var>
       <Ind type="xs:float">250</Ind> </Atom> </And>
```

Fig. 3. RuleML sample from a rule

that raindrop exceeds 250 mm per day is represented in our scenario as illustrated in Fig. 3. The export in the RuleML format is depended on the vocabulary used for the relation constants ("Rel"). Specialized domain ontologies can simplify the RuleML representation as they can have more specific relations and concepts.

2.2.2 Semantic Annotations

The RDF export of datasets corresponding to the rules created is using as model the OpenCyc ontology. We choose to use OpenCyc ontology as it is very large and contains concepts for many specific domains, however, any ontology can be used for annotation as the EnStreaM infrastructure is not tied to a specific ontology. Since our scenario is closely related to the domain of sensor networks, an alternative for OpenCyc could be the Semantic Sensor Network[4] ontology to which extension must be added for representing the landslides domain. For the semantic annotation of the datasets corresponding to a rule, the input configuration file is used.

3 Conclusions and Future Work

In this paper we have presented a system for supporting rule generation on environmental data based on EnStreaM infrastructure. The efficient implementation of data storing and indexing allows the user to interact with the system in timely fashion and makes the system appropriate for demonstration. The use case based on which we demonstrated our system is an environmental scenario using real live data related to landslides phenomena. We plan to extend EnStreaM for real-time monitoring of streaming data in order to detect the events described in the rules generated. Moreover, other future work includes integrating the rules discovered into knowledge bases used by specific reasoning engines. This will help in semi-automatic extension of knowledge bases, supporting advanced reasoning for problems such as complex events processing, anomaly detection or automatic monitoring.

Acknowledgements. This work was partially supported by the Slovenian Research Agency through the programme P2-0016 and project J2-4197, the competence center KC OPCOMM, and the ICT Programme of the EC under PASCAL2 (ICT-NoE-216886), ENVISION (ICT-2009-249120) and PlanetData (ICT-NoE-257641).

References

1. Cugola, G., Margara, A.: Processing flows of information: from data stream to complex event processing. ACM Comput. Surv. **44**(3), 62 p. (2012)
2. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Secaucus (2006)
3. Mitchell, H.B.: Multi-Sensor Data Fusion: An Introduction. Springer, Heidelberg (2007)

[4] http://purl.oclc.org/NET/ssnx/ssn.

4. Schaefer, M., Wanner, F., Mansmann, F., Scheible, C., Stennett, V., Hasselrot A.T., Keim, D.A.: Visual pattern discovery in timed event data. In: Proceedings of the SPIE 7868, 78680K (2011)
5. Kenda, K., Fortuna, C., Fortuna, B., Grobelnik, M.: Videk: A mash-up for environmental intelligence. In: AI Mashup Challange, ESWC (2011)
6. Škrjanc, M., Mladenić, D.: Stream mining on environmental data. In: Proceedings of Information Society Conference IS-2010, Ljubljana, Slovenia, vol. A, pp. 184–187 (2010)

Linked Open Data University of Münster – Infrastructure and Applications

Carsten Keßler(✉) and Tomi Kauppinen

Institute for Geoinformatics, University of Münster, Münster, Germany
{carsten.kessler,tomi.kauppinen}@uni-muenster.de

Abstract. The Linked Open Data University of Münster (LODUM) project establishes a university-wide infrastructure to publish university data as Linked Open Data. The main goals are to increase visibility and accessibility of data produced and collected at the university, and to facilitate effective reuse of these data. This includes the goal to ease the development of applications and mashups based on the data, so that the common user can benefit from the LODUM data. This demo shows the LODUM infrastructure that facilitates application development, and two applications that demonstrate the potential of the LODUM data API.

1 The LODUM Project

Today's research results are no longer limited to papers and books, but also include a variety of data, models, and software. Preserving and accessibility of these contents is fundamental to ensure the transparency and reproducibility of studies. The different scientific resources, be they publications, datasets, methods or tools should be annotated, interlinked and openly shared in order to make science more transparent and reproducible according to the Linked Science[1] approach [1]. Likewise, developers and endusers can benefit from administrative data published as Linked Oped Data.

The LODUM project tackles this long-term goal by implementing a strategy that aims to improve the transparency and visibility of the university, publishing any non-sensitive data online following the Linked Data principles. Data sources existing across the different sites of the university remain in place, leaving the control and responsibility in the hands of the original owner. Such data can be linked to and accessed from http://data.uni-muenster.de. This covers both scientific data and publications, as well as administrative data such as building databases and course schedules. LODUM is laid out as a long-term strategy that will open up and connect different data sources across the 15 faculties and departments step-by-step. With LODUM, the University of Münster is the first German university to implement such a program, following the early examples from the UK.[2] The growing number of publications on Linked Data in science and education shows that this approach is gaining momentum (see, e.g., [2–4]).

[1] See http://linkedscience.org.

[2] See http://linkeduniversities.org/lu/index.php/datasets-and-endpoints/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 447–451, 2015.
DOI: 10.1007/978-3-662-46641-4_43

This demo shows the technical infrastructure to manage data and make them available online. Moreover, we introduce two sample applications that have been built on the LODUM infrastructure: a 3D productivity map of the university, and the WWU App, a Web application optimized for mobile phones that provides information about the university for students, staff, and visitors.

2 Infrastructure

This section documents the LODUM infrastructure, i.e., the technical workflow and the vocabularies in use.

2.1 Technical Workflow

The LODUM infrastructure consists of three layers, as shown in Fig. 1. At the core of this setup, an OWLIM triple store hosts the LODUM data. The management tools support tasks such as vocabulary management and link discovery between the initially disconnected datasets imported into LODUM. The retrieval tools expose the LODUM data via type-specific HTML pages and as raw RDF data, which can also be queried via the SPARQL endpoint at http://data.uni-muenster.de/sparql. In order to get data from existing systems into this infrastructure, we have developed custom triple factories that fetch the data from the different existing systems, convert them to RDF, and push them to the triple store.

2.2 Vocabularies in Use

We reuse existing vocabularies as much as possible to provide semantic annotations of the data we offer in LODUM. In addition, we have also created new vocabularies to introduce classes and properties not defined in existing vocabularies.

The Bibliographic Ontology (BIBO) is used for all bibliographic resources such as books and articles. The Dublin Core Metadata Element Set is used for all occasions where basic metadata are required. The Friend of a Friend (FOAF) vocabulary is used for all information about people and organizations. The W3C Basic Geo Vocabulary is used for all things that are georeference via WGS84 lat/long coordinates. The Teaching Core Vocabulary (TEACH) was used to annotate course descriptions. Publishing Requirements for Industry Standard Metadata (PRISM) provided additional properties for annotating publishing content.[3]

3 Applications

This section introduces the two applications we would like to demonstrate, the LODUM productivity map and the WWU app.

[3] See the following URLs for these vocabularies: http://bibliontology.com/, http://dublincore.org/documents/dces/, http://www.foaf-project.org/, http://www.w3.org/2003/01/geo/, http://linkedscience.org/teach/ns/, and http://www.idealliance.org/specifications/prism/.

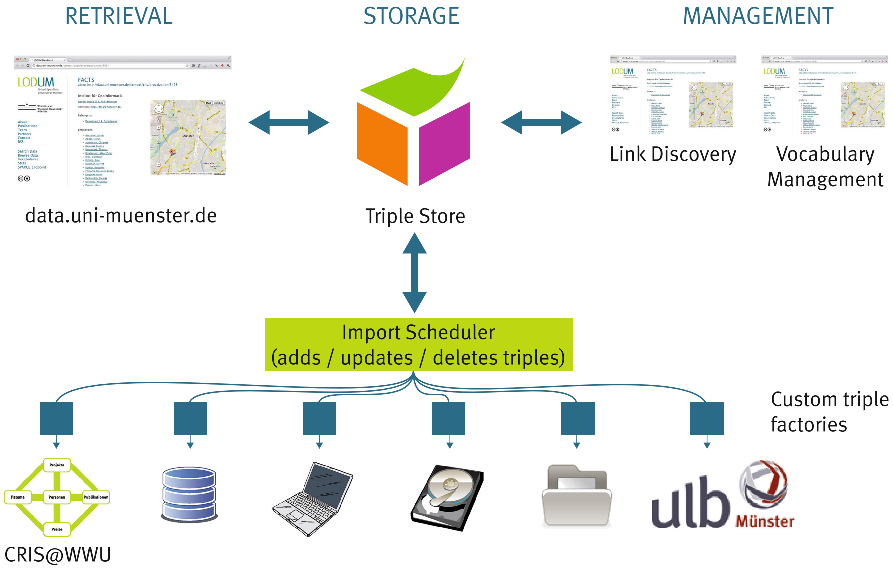


Fig. 1. High-level overview of the LODUM architecture.

3.1 Productivity Map of University Researchers

The productivity map for Google Earth[4] shown in Fig. 2 is an example of how LODUM facilitates data analysis. It renders the university buildings in 3D, where the building height indicates the number of publications written by researchers

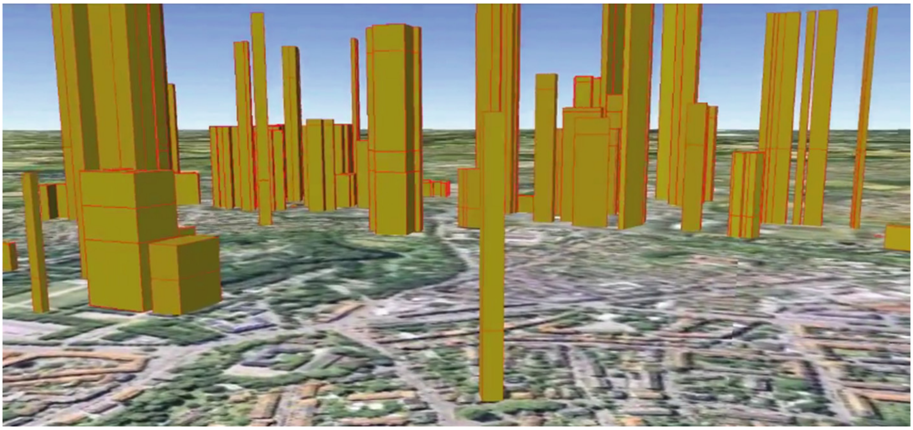


Fig. 2. Screen shot of the map in Google Earth.

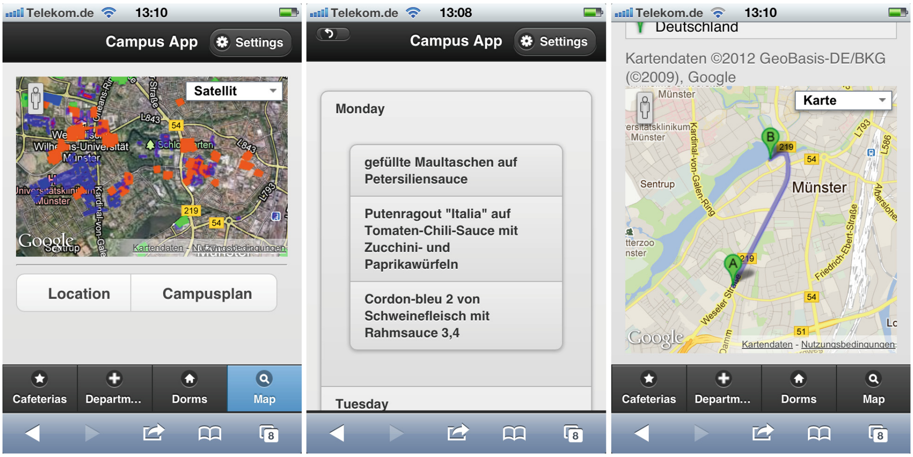
[4] See http://data.uni-muenster.de/context/static/lodum-productivity-map.kml for the KML file and http://youtu.be/QvQ2GF2Zz5g for a video tour.

working in the respective building. The geometries of the buildings are stored according to the emergent GeoSPARQL standard [5]. This interactive map can act as an entry point for the evaluation of and comparison between different departments.

For the building height, the absolute number of papers is normalized by the number of researchers working in the given building for a more balanced impression. The buildings are split in two parts: the lower part indicates the number of journal papers, whereas the upper part represents all other publications. Clicking either of these two parts opens a pop-up with the actual numbers. The distribution of publications between the different institutions in a building is visualized as a pie chart. The pop-ups also include links to the SPARQL queries to pull the data for the given building out of our store, so that interested developers can learn how we have built this map.

3.2 WWU App

The *Westfälische Wilhelms-Universität Münster* (WWU) wants to offer students, staff and visitors information about cafeteria menus, navigation instructions to university buildings, and an overview map of all university buildings on their mobile phone. The latest version of this *WWU App* is a platform-independent Web app based on the LODUM data store, as shown in Fig. 3. It has been developed to replace existing native apps that had the data *baked in*, thus requiring frequent updates, and that were difficult to maintain for the wide range of different platforms. The initial version of the new Web app has been developed in a seminar taught by the authors. It has been built around the PrimeFaces JSF framework and a Glassfish application server that pulls the data from the SPARQL endpoint.



Monday

gefüllte Maultaschen auf Petersiliensauce

Putenragout "Italia" auf Tomaten-Chili-Sauce mit Zucchini- und Paprikawürfeln

Cordon-bleu 2 von Schweinefleisch mit Rahmsauce 3,4

Tuesday

Kartendaten ©2012 GeoBasis-DE/BKG (©2009), Google

Fig. 3. Screen shots with overview of university buildings (left), cafeteria menus (middle), and routing to a cafeteria (right).

4 Conclusions

The University of Münster, Germany, is one of the first universities to commit to a institution-wide Linked Open Data program. The goal of the LODUM initiative is to increase transparency and comprehensibility both for research and for administrative matters. Beyond the technical infrastructure required to put this undertaking into practice—server facilities to run a triple store, SPARQL endpoint, data dumps, backups, and synchronization jobs—fostering a change in the mindset of the university's research community is key for the success of LODUM. In this paper we described two applications demonstrating the use of the LODUM infrastructure. With these demonstrations we want to show that the real potential of Linked Data is in the ease of developing applications on top of the data using a standardized API—i.e. the way to access data online.

Acknowledgements. This research has been funded by the *Linked Open Data University of Münster* project (http://lodum.de) and the International Research Training Group *Semantic Integration of Geospatial Information* (DFG GRK 1498). We would like to thank the LODUM team, as well as Steffan Voss and Christopher Stephan for developing the WWU App.

References

1. Kauppinen, T., Baglatzi, A., Keßler, C.: Linked science: interconnecting scientific assets. In: Critchlow, T., Kleese-Van Dam, K. (eds.) Data Intensive Science. CRC Press, Boca Raton (2013)
2. Shotton, D., Portwin, K., Klyne, G., Miles, A.: Adventures in semantic publishing: Exemplar semantic enhancements of a research article. PLoS Comput. Biol. **5**(4), 621–630 (2009)
3. Demartini, G., Enchev, I., Gapany, J., Cudré-Mauroux, P.: The bowlogna ontology: fostering open curricula and agile knowledge bases for Europe's higher education landscape. Semant. Web **4**(1), 53–63 (2013)
4. Zablith, F., d'Aquin, M., Brown, S., Green-Hughes, L.: Consuming linked data within a large educational organization. In: 2nd International Workshop on Consuming Linked Data at 10th International Semantic Web Conference, Bonn, Germany (2011)
5. Open Geospatial Consortium: OGC GeoSPARQL - A Geographic Query Language for RDF Data (2011). http://www.opengeospatial.org/standards/requests/80. Accessed 29 Dec 2011

OntoPartS: A Tool to Select Part-Whole Relations in OWL Ontologies

Annette Morales-González[1], Francis C. Fernández-Reyes[2], and C. Maria Keet[3](✉)

[1] Advanced Technologies Application Center, CENATAV, Havana, Cuba
amorales@cenatav.co.cu

[2] Instituto Superior Politécnico "José Antonio Echeverría" (CUJAE), Havana, Cuba
ffernandez@ceis.cujae.edu.cu

[3] School of Computer Science, University of KwaZulu-Natal, Durban, South Africa
keet@ukzn.ac.za

Abstract. Representing part-whole and mereotopological relations in an ontology is a well-known challenge. We have structured 23 types of part-whole relations and hidden the complexities of the underlying mereotopological theory behind a user-friendly tool: OntoPartS. It automates modelling guidelines using, mainly, the categories from DOLCE so as to take shortcuts in the selection process, and it includes examples and verbalizations to increase understandability. The modeller's domain ontology, represented in any of the OWL species, can be updated automatically with the selected relation with a simple one-click button.

1 Introduction

There are a plethora of part-whole relations used in ontology development (e.g., [1,2]), and for modellers who are not expert in this field, it is known to be difficult to obtain an overview of the options and subsequently to select the appropriate part-whole relation between entities. This is complicated further by the differences in expressiveness of the OWL species and that, when the appropriate relation is chosen, it can help overcome weaknesses in representation and reasoning over OWL 2-formalised ontologies [3]. To solve these issues, we have developed an ontology-inspired part-whole relation selection tool, OntoPartS, which can be used with the different OWL species, and covers part-whole relations, mereology, and mereotopology (parthood and location).

The theoretical analysis behind the design of OntoPartS is based on the extension of the taxonomy of part-whole relations of [1] with the addition of a taxonomy of formally defined mereotopological relations, which is driven by the KGEMT mereotopological theory [4], resulting in a taxonomy of 23 part-whole relations—mereological, mereotopological, and meronymic—ensuring a solid ontological and logic-based foundation; details can be found in [3]. Although some prior work in modelling guidance for part-whole relations exists in the context of the ORM conceptual modelling language [5] and for topological relations only

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 452–457, 2015.
DOI: 10.1007/978-3-662-46641-4_44

[6], to the best of our knowledge, there is no other software tool for automating and guiding the selection of part-whole relations in general or applied to OWL ontologies.

The demonstration consists of the presentation of OntoPartS and its capabilities to assist modellers in making decisions regarding the correct part-whole relation that may exist among the classes in their ontology. The tool, additional files, and demo videos can be consulted in the online supplementary material at http://www.meteck.org/files/ontopartssup/supindex.html.

In the remainder of the demo paper, we describe aspects related to design and implementation of the tool, and provide an illustration how the tool works.

2 Design and Implementation of OntoPartS

Design. The main requirement of the software is to hide the logic involved in the formal definitions of the 23 part-whole relations. Therefore, the user should be guided to make the decision through a series of steps in an intuitive and effortless way. Besides, the selection procedure should be made as short as possible, leading the user to a small subset of relationship suggestions relevant enough to make the selection of the most suitable one. Since users may be more familiar with the domain and range categories of the classes they are working with rather than with the relations' definitions, it is important to provide a set of (top-level) categories that will help discriminate among relations, which, in turn, streamlines the criteria for selecting the relations and eliminates the possibility of making errors in typing the relation. Usability is to be enhanced by providing simple examples for each relation and category, and pseudo-natural language sentences of the candidate axioms have to be generated. Last, the user should have the possibility to save the selected relation to the OWL ontology file from where the involved classes were taken.

Given the core requirements, several design decisions were made for OntoPartS. In order to quickly assess the contribution of the tool for the intended purpose, we chose to use a rapid way of prototyping to develop the software. A stand-alone tool that works with OWL files was developed, allowing the ontologist not to be bound to a single ontology editor. We chose to use the DOLCE top-level ontology categories to standardize the relationships' decision criteria (though another top-level ontology could have been chosen as well). Important for solid software design, we used activity diagrams to describe the steps to be executed to interact with OntoPartS and to select the appropriate relation, therewith making the selection principles technology-independent; Fig. 1 shows one such activity diagram.

Implementation. The tool has been implemented in C# using the .Net Framework. The Jena2 ontology API [7] has been used to handle OWL files and avail of its features to update an ontology. The updates are saved back into the OWL file using the standard RDF/XML syntax, including the axiom involving the selected part-whole relation between the classes that play the part- and whole-role, and the

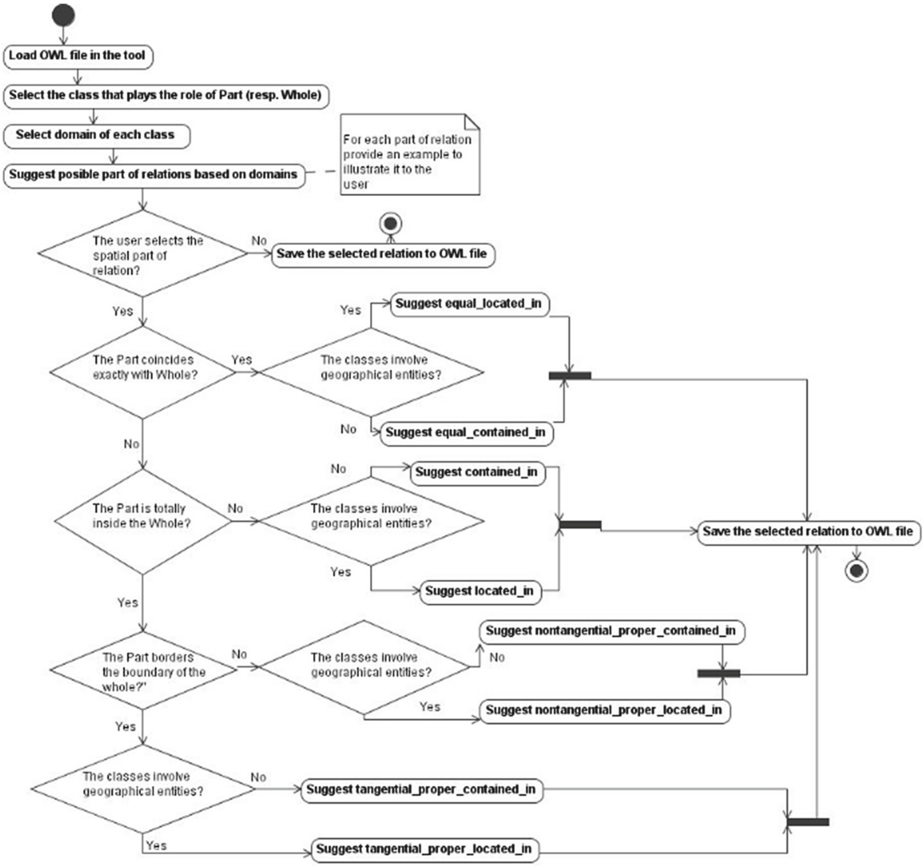

Load OWL file in the tool
Select the class that plays the role of Part (resp. Whole)
Select domain of each class
Suggest posible part of relations based on domains
For each part of relation provide an example to illustrate it to the user
The user selects the spatial part of relation?
No
Save the selected relation to OWL file
Yes
The Part coincides exactly with Whole?
Yes
The classes involve geographical entities?
Yes
Suggest equal_located_in
No
Suggest equal_contained_in
No
The Part is totally inside the Whole?
No
The classes involve geographical entities?
No
Suggest contained_in
Yes
Suggest located_in
Save the selected relation to OWL file
Yes
The Part borders the boundary of the whole?"
No
The classes involve geographical entities?
No
Suggest nontangential_proper_contained_in
Yes
Suggest nontangential_proper_located_in
Yes
The classes involve geographical entities?
No
Suggest tangential_proper_contained_in
Yes
Suggest tangential_proper_located_in

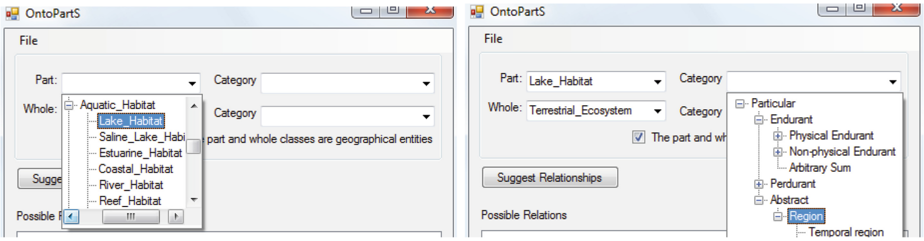
Fig. 1. One of the activity diagrams; this one shows the steps for using the tool and the decision making process to select the correct mereotopological relation.

whole taxonomy of 23 part-whole relations. The tool has been evaluated experimentally [3], and feedback was incorporated, so that OntoPartS is a fully working prototype.

3 Description of the Selection Procedure

OntoPartS guides the ontologist in the process of making the decision in an intuitive way. We illustrate the selection procedure through two examples.

Example 1. Suppose a modeler is developing an earth science ontology and has to figure out the relation between `Lake_Habitat` and `Terrestrial_Ecosystem`. The first step is to load the OWL file in OntoPartS, and then select the class that represents the part, Lake_Habitat, and the class that represents the whole, Terrestrial_Ecosystem (see Fig. 2a). OntoPartS will provide the set of possibly suitable part-whole relations (0–4) once the user has specified the domain and



Part: Lake_Habitat
Whole: Terrestrial_Ecosystem

a) Selecting the part and whole classes from the loaded OWL file

b) Selecting the categories for each entity

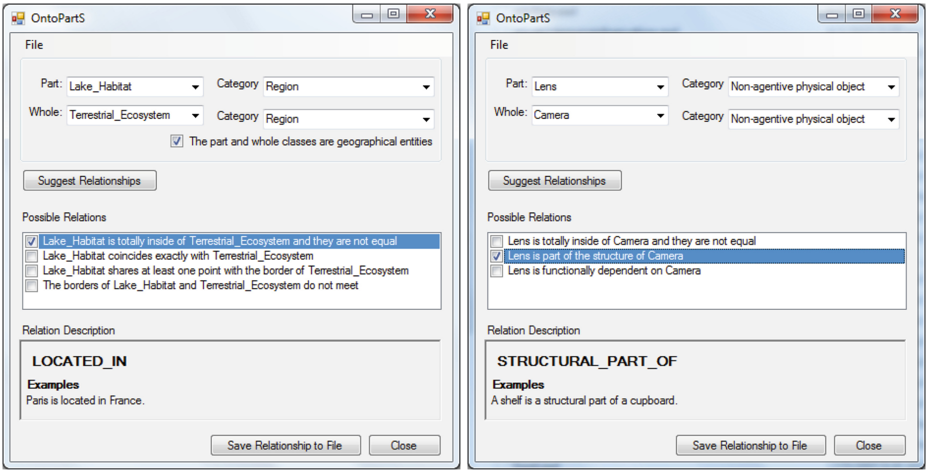
Fig. 2. OntoPartS interface: selection of classes and categories.

range categories (currently taken from DOLCE), as depicted in Fig. 2b. Examples for each DOLCE category are shown by hovering the pointer over the terms in the taxonomy; e.g., hovering over Achievement shows "Ex: A conference, an ascent, a performance". If the selected categories are regions (or any of its subclasses), we enter the branch of mereotopological relations and one must specify whether the regions correspond to geographical entities (2D, versus 3D containment). Lake_Habitat and Terrestrial_Ecosystem have the dual notion indicating a region and being a particular type of enduring entity, and in a map-making context, the 2D perspective is chosen. Subsequently, one clicks on the button "Suggest relationships". The amount of relations suggested depends on the chosen categories; which are four in this case (Fig. 3a); if the selected classes were, e.g., processes, then there is only one option (involved_in, as OntoPartS includes the taxonomy of [1]). Each proposed relation is verbalised to make the option more understandable, e.g., Lake_Habitat is totally inside of Terrestrial_Ecosystem and they are not equal (i.e., located in), and an illustrative example is shown as an additional guide (see Fig. 3a). Once the desired relation is selected by marking the corresponding checkbox, on can choose to add it to the OWL file by clicking the button "Save relationship to file" (Fig. 3a bottom) and continue with other classes and selection of a part-whole relation or with developing the ontology in the ontology development environment of choice. ♢

Example 2. Suppose a modeler or domain expert is developing a photography ontology and wants to choose a relation between Lens and Camera, where Lens is selected as the part class, and Camera as the whole. The categories for both classes correspond to Non-agentive physical objects in DOLCE. When the user clicks Suggest relationships, we obtain three possible relations between these classes. By selecting each of the suggested relationships, one can read its verbalization in context, thereby clarifying examples for each option, so as to help deciding which one is the correct one. This process is summarized in Fig. 3b. ♢

4 Discussion

With this demonstration we will show how OntoPartS can sensitize the modeller to the myriad of part-whole relations in a piecemeal fashion, how to make



OntoPartS

File

Part: Lake_Habitat Category Region

Whole: Terrestrial_Ecosystem Category Region

☑ The part and whole classes are geographical entities

Suggest Relationships

Possible Relations

☑ Lake_Habitat is totally inside of Terrestrial_Ecosystem and they are not equal
□ Lake_Habitat coincides exactly with Terrestrial_Ecosystem
□ Lake_Habitat shares at least one point with the border of Terrestrial_Ecosystem
□ The borders of Lake_Habitat and Terrestrial_Ecosystem do not meet

Relation Description

LOCATED_IN

Examples

Paris is located in France.

Save Relationship to File Close

a) Example of the mereotopological branch

OntoPartS

File

Part: Lens Category Non-agentive physical object

Whole: Camera Category Non-agentive physical object

Suggest Relationships

Possible Relations

□ Lens is totally inside of Camera and they are not equal
☑ Lens is part of the structure of Camera
□ Lens is functionally dependent on Camera

Relation Description

STRUCTURAL_PART_OF

Examples

A shelf is a structural part of a cupboard.

Save Relationship to File Close

b) Example of the mereological branch

Fig. 3. Relationships suggested by OntoPartS.

a selection of the appropriate part-whole relation for the identified classes, and achieve the corresponding ontology update effortlessly. Attendees also can bring their own ontology to augment it with part-whole relations or to check its quality with respect to what was already represented in their ontology and compare it with OntoPartS' suggestions.

OntoPartS was experimentally evaluated using two groups of students with different degrees of expertise and a smaller group of experienced researchers. Although we cannot claim that the tool leads to statistically significant less modelling errors, it does assist performing the selection process so that it is done more efficiently and quickly [3].

Current and future work pertains to adding more modelling guidance features, and developing a plugin for Protégé.

References

1. Keet, C.M., Artale, A.: Representing and reasoning over a taxonomy of part-whole relations. Appl. Ontology **3**(1–2), 91–110 (2008)
2. Mejino, J.L.V., et al.: Representing complexity in part-whole relationships within the foundational model of anatomy. In: Proceedings of the AMIA Fall Symposium, pp. 450–454 (2003)
3. Keet, C.M., Fernández-Reyes, F.C., Morales-González, A.: Representing mereotopological relations in OWL ontologies with OntoPartS. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 240–254. Springer, Heidelberg (2012)
4. Varzi, A.: Spatial reasoning and ontology: parts, wholes, and locations. In: Aiello, M., Pratt-Hartmann, I., Van Benthem, J. (eds.) Handbook of Spatial Logics, pp. 945–1038. Springer, Heidelberg (2007)

5. Keet, C.M.: Part-whole relations in object-role models. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4278, pp. 1118–1127. Springer, Heidelberg (2006)
6. Yang, W., Luo, Y., Guo, P., Tao, H.F., He, B.: A model for classification of topological relationships between two spatial objects. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3614, pp. 723–726. Springer, Heidelberg (2005)
7. Dickinson, I.: The Jena ontology API (2009). http://jena.sourceforge.net/ontology/index.html

Hubble: Linked Data Hub for Clinical Decision Support

Rinke Hoekstra[1,3](✉), Sara Magliacane[1], Laurens Rietveld[1],
Gerben de Vries[2], Adianto Wibisono[2], and Stefan Schlobach[1]

[1] Department of Computer Science, VU University Amsterdam,
Amsterdam, The Netherlands
{rinke.hoekstra,s.magliacane,laurens.rietveld,k.s.schlobach}@vu.nl
[2] Department of Computer Science, University of Amsterdam,
Amsterdam, The Netherlands
{g.k.d.devries,a.wibisono}@uva.nl
[3] Leibniz Center for Law, University of Amsterdam,
Amsterdam, The Netherlands

Abstract. The AERS datasets is one of the few remaining, large publicly available medical data sets that until now have not been published as Linked Data. It is uniquely positioned amidst other medical datasets. This paper describes the Hubble prototype system for clinical decision support that demonstrates the speed, ease and flexibility of producing and using a Linked Data version of the AERS dataset for clinical practice and research.

Keywords: Linked data · Adverse event · Clinical decision support · Health care

1 Introduction

This paper describes a prototype system for clinical decision support, Hubble, that demonstrates the ease with which (medical) legacy data can be turned into RDF, linked to other data sets, and used to support both clinical research and clinical practice. At the heart of the system lies a Linked Data version of the Adverse Event Reporting System (AERS) dataset of the Federal Drug Administration (FDA). To some extent, this exercise can be categorised under the heading 'yet another exposing of data as Linked Data'. However, the system convincingly demonstrates three important *sales pitches* of Linked Data: *interoperability*, *interlinking*, and *tool availability*. In particular, the system shows the huge difference in *usability* between the original 'dead' dataset and the 'live' Linked Data version, it shows how *quickly* this can be achieved using standard tools, and it *validates* the standard three-tier architecture that separates *data*, *application logic* and *presentation*. We validate the quality of the dataset by comparing it to results of a real use-scenario on the AERS dataset.

Clinical Decision Support. Clinical decision support (CDS) can be defined as "the use of the computer to bring *relevant* knowledge to bear on the health care

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 458–462, 2015.
DOI: 10.1007/978-3-662-46641-4_45

and well being of a patient" [2]. Clinical guidelines play a central role in CDS systems; they contain the consolidated knowledge on patient treatment. However, guidelines are *slow movers*, decided upon in periodic conferences where new evidence is weighed for updating a guideline document. The evidence itself, however, accumulates at a tremendous pace: there are numerous clinical trials and well over 10 thousand publications on breast cancer every year. Therefore, a CDS should bring together patient information, relevant guidelines and important new findings in clinical research. In this context, the key challenge is: how to ensure that the information presented by the CDS is relevant and trustworthy?

The Data. The fields of health care and life science (HCLS) have traditionally seen a lot of attention from the Semantic Web community, and vice versa: semantic web languages, and their predecessors have proven to be a convenient paradigm for representing biomedical knowledge. Vocabularies in the HCLS field are highly standardised; computer analysis, and computer-based information exchange are ubiquitous throughout the field (viz. the Humanities). As a result, many (bio)medical databases and terminologies are now published as linked data, taking up about a fourth of the Linked Data cloud. Examples are medical vocabularies such as SnomedCT, MeSH, MedDRA, and the NCI Thesaurus (all part of the Unified Medical Language System (UMLS)),[1] and datasets such as LinkedCT (clinical trials), Sider, Drugbank and RxNorm (drug information), Uniprot (protein sequences), to name but a few.

The AERS datasets is one of the few remaining, large publicly available medical data sets that until now have not been published as Linked Data. An adverse event (AE) is an adverse change in health or side effect while the patient is receiving treatment. A *serious* adverse event (SAE) is life-threatening and, amongst others, may result in death, requires hospitalisation or prolongation of existing hospitalisation and will result in persistent or significant disability or incapacity. Known chemotherapy-related SAEs in breast cancer (US only) were linked to 22 % of hospitalisations. Clearly, from a clinical perspective, serious adverse events are very important: this is where CDS can make a huge difference.

1.1 System Description

The architecture of Hubble follows a three tiered architecture: (a) a 4Store triple store,[2] containing the AERS dataset (AERS-LD), CTCAE,[3] a selection of DBPedia, Sider, and Drugbank.[4]; (b) a set of SPARQL 1.1 queries and some server-side code; and (c) a Java Smart GWT framework client interface.[5] This section briefly discusses the way we *convert* and *link* data, *annotate* documents and *present* the result to a user.

[1] See http://www.nlm.nih.gov/research/umls/.

[2] See http://4store.org.

[3] CTCAE, subset of MedDRA, lists AEs for cancer therapy: http://bit.ly/zOVPUt.

[4] See http://dbpedia.org, http://www4.wiwiss.fu-berlin.de/sider/ and http://www4.wiwiss.fu-berlin.de/drugbank/, respectively.

[5] See http://code.google.com/p/smartgwt/.

Data Conversion and Linking. The AERS data files are published on a quarterly basis, as zip files containing dollar separated tables. These zip files are roughly 20 MB in size, and available from the FDA website from two separate static webpages.[6] Converting this data is a five step process: (1) scrape the FDA website, download and unzip the data dump; (2) check integrity of the files, applying fixes if necessary;[7] (3) import the data into a MySQL database; (4) dump the data to RDF following a D2RQ mapping;[8] and (5) import the data into 4Store.[9] This conversion was implemented as a pipeline called through a Python provenance wrapper. This wrapper generates provenance information expressed in the PROV-O vocabulary.[10] Due to hardware limitations we had to restrict the dataset to the years 2011 and 2012 (first two quarters), resulting in a total size of 80 M triples.

The AERS dataset is uniquely positioned amidst other HCLS datasets, providing opportunities for linking to drug, location, patient and diagnosis related information. Furthermore, reports in AERS are filled in by hand. Linking out to other datasets could help in identity reconciliation (e.g. drug names, marketing names, and chemical substances) as well as detecting misspellings (e.g. in manufacturer names). We specified mappings between the UMLS, Sider, LinkedCT, Drugbank, DBPedia and CTCAE datasets using the SILK link specification language [4], resulting in over 60 K links based *only* on exact string matches.[11]

Annotations. Step two is the automatic annotation of scientific publications and clinical guidelines (available as PDF files) using the vocabularies in the repository. This process has three steps: *stripping* of PDF documents to plain text, *indexing* the plain text documents and *generating* annotations. We use the PDF-Box library[12] for conversion to plain text. Each document is then divided into separate paragraphs, dubbed '*chunks*'. For each chunk we store the coordinates of its bounding box in the PDF. The chunks are then indexed for terms (including synonyms) from the CTCAE ontology, using Lucene.[13]

The Annotation Ontology (AO) is a vocabulary for annotating scientific publications and documents on the Web. AO has a lightweight provenance model, which allows storing information about the authors, curation and different versions for each annotation. We use the AO format [1] to represent an annotation for every term found, using a *prefix-postfix selector* to identify its position inside a chunk. The advantage of using this method is that annotations will persist across different manifestations of the same document (pdf, html, xml, etc.).

[6] See http://1.usa.gov/uyoAI.

[7] For instance, some rows contain line breaks in the wrong places, do not properly escape the separator character or span fewer columns than expected.

[8] See https://github.com/cygri/d2rq.

[9] Unfortunately, exposing through D2R Server turned out to be too slow.

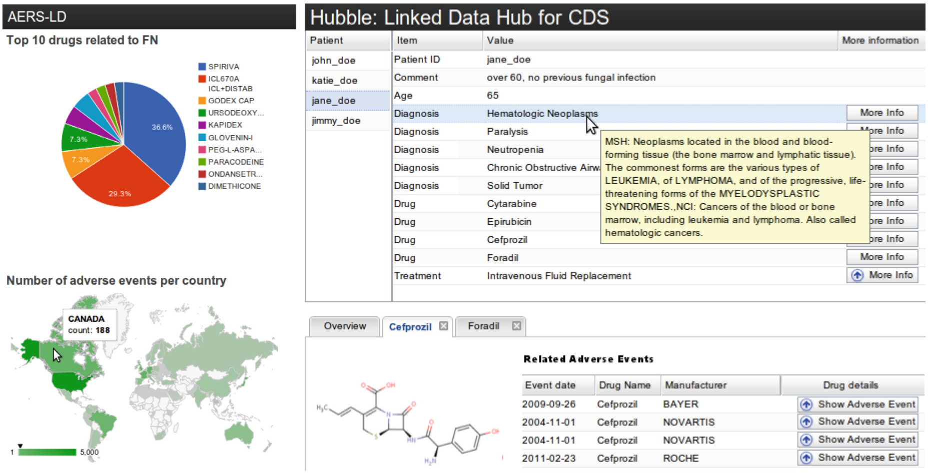
[10] PROV-O-Matic, see http://github.com/Data2Semantics/, currently in alpha stages of development. PROV-O: See http://www.w3.org/TR/prov-o/.

[11] Using less exact matching on drug names can have unwanted consequences.

[12] See http://pdfbox.apache.org/.

[13] See http://lucene.apache.org.

We store an *image selector* that uses the chunk bounding box: this image selector is then used to highlight part of the PDF document.



Hubble: Linked Data Hub for CDS

Patient
john_doe
katie_doe
jane_doe
jimmy_doe

Item	Value	More information
Patient ID	jane_doe	
Comment	over 60, no previous fungal infection	
Age	65	
Diagnosis	Hematologic Neoplasms	More Info
Diagnosis	Paralysis	More Info
Diagnosis	Neutropenia	ore Info
Diagnosis	Chronic Obstructive Airw	ore Info
Diagnosis	Solid Tumor	ore Info
Drug	Cytarabine	ore Info
Drug	Epirubicin	ore Info
Drug	Cefprozil	ore Info
Drug	Foradil	More Info
Treatment	Intravenous Fluid Replacement	More Info

MSH: Neoplasms located in the blood and blood-forming tissue (the bone marrow and lymphatic tissue). The commonest forms are the various types of LEUKEMIA, of LYMPHOMA, and of the progressive, life-threatening forms of the MYELODYSPLASTIC SYNDROMES.,NCI: Cancers of the blood or bone marrow, including leukemia and lymphoma. Also called hematologic cancers.

Overview | Cefprozil ☒ | Foradil ☒

Related Adverse Events

Event date	Drug Name	Manufacturer	Drug details
2009-09-26	Cefprozil	BAYER	Show Adverse Event
2004-11-01	Cefprozil	NOVARTIS	Show Adverse Event
2004-11-01	Cefprozil	NOVARTIS	Show Adverse Event
2011-02-23	Cefprozil	ROCHE	Show Adverse Event

(a) (b)

Fig. 1. Examples of possible visualisations (a) and Hubble interface (b)

User Interface. The Hubble interface is our prototype CDS system.[14] It lists patients, shows information about a selected patient, and presents more detailed information about specific parts of the patient information to the bottom (Fig. 1b). This is done in several steps, each consisting of a single SPARQL query. First, we retrieve a list of available patient records. Second, when the user selects a record we retrieve actual patient data: detailed patient information (diagnoses, drugs, age, etc.), enriched with information from the Linked Life Data (LLD) endpoint (e.g. a drug description tooltip).[15] At the same time, we retrieve annotations that match the patient description, and depict small snippets of clinical guidelines and relevant literature. We can drill-down from this detailed information to: more information about a *diagnosis* (taken from LLD), similar *cases* in AERS-LD, *drug information* such as its chemical structure (from LLD and Drugank) and common AEs related to the drug, *provenance* information about an annotation, and the underlying text. We have intentionally limited the amount and diversity of information presented through the interface, pending feedback from expert users.

The AERS-LD repository is publicly accessible through its SPARQL endpoint, and can be browsed through a customised Pubby browser interface.[16]

[14] See http://aers.data2semantics.org/prototypeInterface.

[15] See http://linkedlifedata.com.

[16] See http://aers.data2semantics.org for more information. Pubby: http://www4.wiwiss.fu-berlin.de/pubby/.

Arguably a more actionable presentation than dollar-separated files. The endpoint turned out to be very well suited for various visualisations of the underlying data (Fig. 1a).[17]

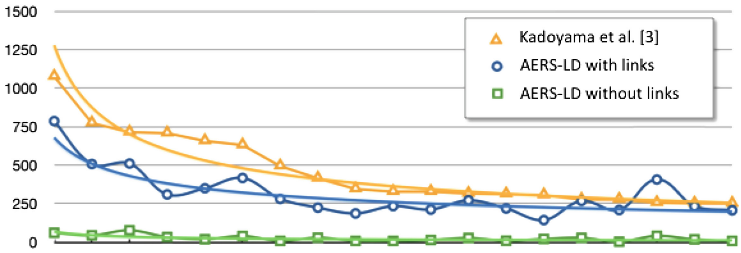


Fig. 2. Number of co-occurrences of Adverse Events and 5-FU (Y-axis). The X-axis represents the ranking of AEs based on the one in [3].

1.2 Evaluation and Discussion

The Hubble CDS prototype was built in a very short period (literally over Christmas), and is already showing real potential for clinical research. We validated the dataset by comparing results from AERS-LD with a study into the co-occurrence of AEs with two drugs (5-FU and Capecitabine) [3]. This study was preceded by a labor intensive effort to clean the dataset: consolidation of multiple names for drugs and removal of duplicate submissions and non-drug entries. We compared AE-drug co-occurrence on the same selection of AEs in [3] *with* and *without* taking advantage of the 60 K links (see above) in AERS-LD. We did not apply any other data cleaning or harmonisation and used only a limited dataset. The result, depicted in Fig. 2, although far from perfect, shows that the Linked Data cloud provides a huge bootstrap for improving the quality of results.

Future work includes publishing the full AERS-LD dataset (all 7 years), increasing both the breadth and depth of annotations through supervised annotation of guidelines, combined with large scale annotation of scientific publications, improving selection and ranking of query results in the Hubble interface based on annotations, citation indexes, and provenance related information.

References

1. Ciccarese, P., et al.: An open annotation ontology for science on web 3.0. J. Biomed. Semant. (2011)
2. Greenes, R.A.: Clinical Decision Support: The Road Ahead. AP/Elsevier Science and Technology, Burlington (2007)
3. Kadoyama, K., et al.: Adverse event profiles of 5-Fluorouracil and Capecitabine: data mining of the public version of the FDA adverse event reporting system, AERS, and reproducibility of clinical observations. Ing. J. Med. Sci. **9**, 33–39 (2012)
4. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - a link discovery framework for the web of data. In: 2nd Workshop about Linked Data on the Web (LDOW 2009) (2009)

[17] Built directly from the endpoint using Sgvizler, http://sgvizler.googlecode.com/.

Confomaton: A Conference Enhancer with Social Media from the Cloud

Houda Khrouf[1], Ghislain Atemezing[1], Thomas Steiner[2],
Giuseppe Rizzo[1](✉), and Raphaël Troncy[1]

[1] EURECOM, Sophia Antipolis, France
{houda.khrouf,ghislain.atemezing,
giuseppe.rizzo,raphael.troncy}@eurecom.fr
[2] Universitat Politècnica de Catalunya, Barcelona, Spain
tsteiner@lsi.upc.edu

Abstract. A scientific conference is a type of event for which the structured program is generally known in advance. The Semantic Web community has setup a so-called Semantic Web dog food server that exposes structured data about the detailed program of more and more conferences and their sub-events (e.g. sessions). Conferences are also events that trigger a tremendous activity on social media. Participants tweet or post longer status messages, engage in discussion with comments, share slides and other media captured during the conference. This information is spread over multiple platforms forcing the user to monitor many different channels at the same time to fully benefit of the event. In this paper, we present Confomaton, a semantic web application that aggregates and reconciles information such as tweets, slides, photos and videos shared on social media that could potentially be attached to a scientific conference.

Keywords: Data aggregation · Linked Data consumption · Data reconciliation · Data visualization

1 Introduction

Just like any other popular event, scientific conferences trigger an ever-growing amount of activities on social media. But in contrast to events like concerts or sport matches, a conference is highly structured, consisting generally of workshops and tutorials, parallel sessions composed of talks, keynotes, panels, posters and demos that all have planned schedules, topics, and allocated rooms. The Semantic Web community is used to model this structured data using RDF and to publish it following the Linked Data principles using a so-called Semantic Web dog food server[1] [3]. The social media activities that are shared around the conference consist of slides, photos and videos posted by authors and participants but also status messages published on social networks such as Twitter, Google+ or Facebook. The problem is that these activities are unstructured data, spread

[1] http://data.semanticweb.org/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 463–467, 2015.
DOI: 10.1007/978-3-662-46641-4_46

over multiple platforms that are just weakly associated to a conference event as a whole as opposed to its fine grained sub-events. Overall, the physical participants or the ones who try to follow the event online are forced to monitor multiple channels to full benefit from a scientific conference.

Exploring this intrinsic connection between structured events and media shared on the web has been the focus of several studies [1,2,5]. They propose different techniques in the area of media classification, data interlinking and event detection, trying to leverage the wealth of user generated knowledge. However, most of these works have mainly targeted a specific social service such as Twitter or Flickr, without any guarantee that they can be valid for others services. We believe that exploiting the diversification of user generated content from different social services inside one application is a challenging task. In this work, we aim at creating a rich environment to enable users navigating events as well as their various representative media such as pictures, slides and tweets. A typical usage is to gather data about a scientific conference and investigate the added value of collecting scientific-related media. A non trivial task in such application is to connect structured data with extremely noisy content, especially in the case of a major conference. In this paper, we present Confomaton, a semantic web application that collects social media activities, reconciles the data and attempts to align it to the various sub-events that compose a conference. We will showcase Confomaton live during the ESWC 2012 conference.

2 Confomaton

The name *Confomaton* is a word play on the French term *Photomaton* (English photo booth) and *conference*. Just like a Photomaton illustrates the scene inside of the booth, the Confomaton illustrates an event such as a conference enriched with social media. Confomaton is a semantic web application that produces and consumes linked data and is composed of four main components: (i) an Event Collector which extracts events descriptions such as the ones available in the Semantic Web Dog Food corpus; (ii) a Media Collector which collects social media content and represents it in RDF using various vocabularies; (iii) a Reconciliation Module playing the role of associating social media with sub-events and external knowledge; (iv) a User Interface powered by an instance of the Linked Data API as a logical layer connecting all the data in the triple store with the front-end visualizations.

Event Collector: it takes as input the Dog Food corpus described using the SWC ontology and converts all events into the LODE ontology[2], a minimal model that encapsulates the most useful properties for describing events. We use the Room ontology[3] for describing the rooms contained in a conference center. An explicit relationship between an event and its representative media (photo, slide, tweet, etc.) is realized through the `lode:illustrate` property. For describing

[2] http://linkedevents.org/ontology/.

[3] http://vocab.deri.ie/rooms.

those media, we re-use two popular vocabularies: the W3C Ontology for Media Resources[4] for photos and videos, and SIOC[5] for tweets, status, posts and slides.

Media Collector: it has the purpose to search from various social networks and media platforms for event-related media items such as photos, videos, and slides. We currently support 4 social networks (Google+, MySpace, Facebook, and Twitter) and 7 media platforms (Instagram, YouTube, Flickr, MobyPicture, img.ly, yfrog and TwitPic). Our approach being agnostic of media providers, we offer a common alignment schema for all of them containing information such as the deep link of the media, the media type, the story URL, the story content, the author profile URL, the timestamp, etc. In order to retrieve data from media providers, we use the particular media provider's search Application Programming Interfaces (API) where they are available, and fall back to screen scraping the media provider's website if not.

Reconciliation Module: it aims to align the incoming stream of social media with their appropriate events and to interlink some descriptions with general knowledge available in the LOD cloud (e.g. people and institutions descriptions). Attaching social media to fine-grained event is a challenging problem. We tackle it by pre-processing the data with two successive filters in order to reduce the noise: one of them relies on keyword search applied to some fields such as title and tag, while the other one filters data based on temporal clues. The reconciliation is then ensured through a pre-configured mapping between a set of keywords and hashtags and their associated events. Furthermore, we extract named entities from the microposts using the NERD framework [4] and we develop a specific heuristic for aligning tweets with sub-events.

User Interface: it is built around four perspectives (tabs in the UI) characterizing an event: (i) *"Where does the event take place?"*, (ii) *"What is the event about?"*, (iii) *"When does the event take place?"*, and finally (iv) *"Who are the participants of the event?"*. In addition, the UI offers full text search for these four dimensions. On the left side of the main view, the user can select the main conference event or one of the sub-events as provided by the Dog Food metadata corpus. On the center, the default view is a map centered on where the event took place and the user is also encouraged to explore potential other type of events (concerts, exhibitions, sports, etc.) happening nearby, this data being provided by EventMedia [5]. The *What* tab is media-centered and allows to quickly see what illustrates a selected event (tweets, photos, slides). Zooming in an event triggers a popup window that contains the title and timetable of the event, the precise room location and a slideshow gallery of all the medias collected for this event. For the *When* tab, a timeline is provided in order to filter events according to a day time period. Finally, the *Who* tab aims at showing all the participants of the conference. This is intrinsically bound to a social component, aiming not only to present relevant information about a participant (his

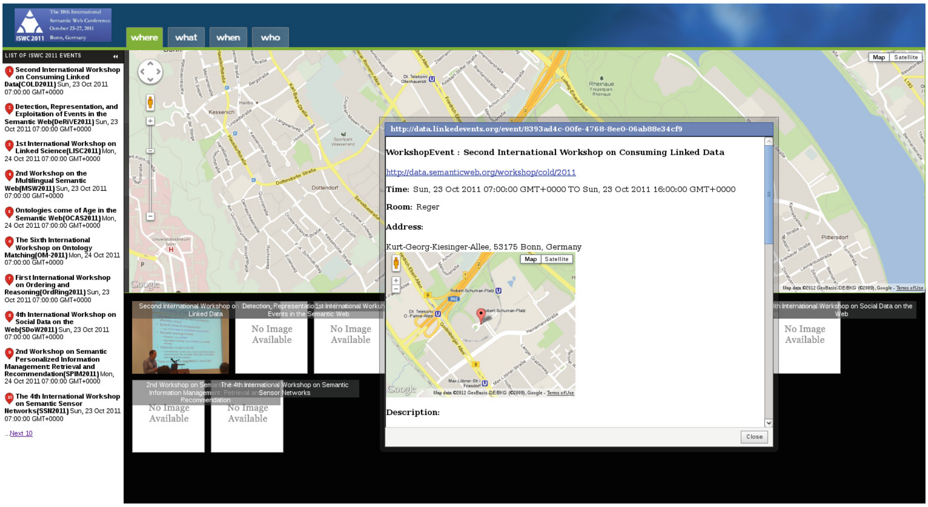
[4] http://www.w3.org/TR/mediaont-10/.

[5] http://rdfs.org/sioc/spec/.

affiliation, homepage, or role at the conference) but also the relationships between the participants between themselves and with the events.

The UI is powered by the Linked Data API[6] which provides a configurable way to access RDF data using simple RESTful URIs that are translated into queries to a SPARQL endpoint. More precisely, we use the Elda[7] implementation developed by Epimorphics. Elda comes with some pre-built samples and documentation which allow to build specification to leverage the connection between the back-end (data in the triple store) and the front-end (visualizations for the user). The API layer helps to associate URIs with processing logic that extract data from the SPARQL endpoint using one or more SPARQL queries and then serialize the results using the format requested by the client. A URI is used to identify a single resource whose properties are to be retrieved or to identify a set of resources, either through structure of the URI or through query parameters.

We have deployed the application using the data describing the ISWC 2011 conference (Fig. 1). The application is available at http://eventmedia.eurecom.fr/iswc2011/.


The 10th International Semantic Web Conference October 23-27, 2011 Bonn, Germany
ISWC 2011
where
what
when
who
Map
Satellite
Close

LIST OF ISWC 2011 EVENTS

1 **Second International Workshop on Consuming Linked Data(COLD2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

2 **Detection, Representation, and Exploitation of Events in the Semantic Web(DeRiVE2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

3 **1st International Workshop on Linked Science(LISC2011)** Mon, 24 Oct 2011 07:00:00 GMT+0000

4 **2nd Workshop on the Multilingual Semantic Web(MSW2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

5 **Ontologies come of Age in the Semantic Web(OCAS2011)** Mon, 24 Oct 2011 07:00:00 GMT+0000

6 **The Sixth International Workshop on Ontology Matching(OM-2011)** Mon, 24 Oct 2011 07:00:00 GMT+0000

7 **First International Workshop on Ordering and Reasoning(OrdRing2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

8 **4th International Workshop on Social Data on the Web(SDoW2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

9 **2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation(SPIM2011)** Mon, 24 Oct 2011 07:00:00 GMT+0000

10 **The 4th International Workshop on Semantic Sensor Networks(SSN2011)** Sun, 23 Oct 2011 07:00:00 GMT+0000

...Next 10

http://data.linkedevents.org/event/8393ad4c-00fe-4768-8ee0-06ab88e34cf9

WorkshopEvent : Second International Workshop on Consuming Linked Data

http://data.semanticweb.org/workshop/cold/2011

Time: Sun, 23 Oct 2011 07:00:00 GMT+0000 TO Sun, 23 Oct 2011 16:00:00 GMT+0000

Room: Reger

Address:

Kurt-Georg-Kiesinger-Allee, 53175 Bonn, Germany

Description:

Fig. 1. A showcase of Confomaton with the ISWC 2011 data

3 Conclusion

In this paper, we have presented Confomaton, a semantic web application using the diversity of media resources generated by users, that can potentially be linked with more structure metadata such as a detailed program of a scientific conference as exposed by the Dog Food corpus. We show that collecting

[6] http://code.google.com/p/linked-data-api/wiki/Specification.
[7] http://code.google.com/p/elda.

and reconciliating media items from many services (Twitter, SlideShare, Flickr, Google+, etc.) enables to provide a better conference experience including visual conference summarization or explorative search during and after an event.

Confomaton is an ambitious project that shows well the difficulty to use Linked Data technologies in a real setting. The solution we propose makes use of many services starting from scraping, aggregating data and their reconciliation. It unifies them and exposes the aggregated information as linked data. However, the more services are handled and the more issues one has to deal with, generally with the API provided by those services (e.g. the number of requests of some APIs). Concerning the Linked Data API, at the time we developed the paper, it was not possible to handle queries using selectors with *DISTINCT* and *GROUP BY* queries, although there are means to go through this limitation using *UNION*. We also face the problem of the objective criteria to select a particular vocabulary for modeling the data. Finally, Confomaton is a flexible solution to encompass the data fragmentation due to the proliferation of services used by the conference guests. Confomaton will be deployed with the data describing the ESWC 2012 conference and we will invite all physical and remote participants to provide suggestions for a better user experience.

Acknowledgments. This work is partially supported by the project AAL-2009-2-049 “Adaptable Ambient Living Assistant” (ALIAS) and by the European Commission under Grant No. 248296 FP7 I-SEARCH project.

References

1. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010), New York, USA, pp. 291–300 (2010)
2. Liu, X., Troncy, R., Huet, B.: Using social media to identify events. In: 3rd Workshop on Social Media (WSM 2011), Scottsdale, Arizona, USA (2011)
3. Möller, K., Heath, T., Handschuh, S., Domingue, J.: Recipes for semantic web dog food — the ESWC and ISWC metadata projects. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 802–815. Springer, Heidelberg (2007)
4. Rizzo, G., Troncy, R.: NERD: a framework for unifying named entity recognition and disambiguation extraction tools. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Avignon, France, pp. 73–76 (2012)
5. Troncy, R., Malocha, B., Fialho, A.: Linking events with media. In: 6th International Conference on Semantic Systems (I-SEMANTICS 2010), Graz, Austria (2010)

OBA: Supporting Ontology-Based Annotation of Natural Language Resources

Nadeschda Nikitina(✉)

Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany
nikitina@kit.edu

Abstract. In this paper, we introduce OBA – an application for NLP-based annotation of natural language texts with ontology classes and relations. OBA provides support for different tasks required for semi-automatic semantic annotation. Among other things, it supports creating manual semantic annotations in order to enrich the set of lexical patterns, automatically annotating large corpora based on specified lexical patterns, and evaluating the results of semantic annotation.

1 Introduction

In the last decade, semantic annotation of unstructured data has significantly gained in popularity due to its usefulness for semantic search. A well-known example is Semantic Media Wiki, an extension of Media Wiki allowing for relation annotations, i.e., annotations specifying the meaning of relations between wiki articles, and attribute annotations, i.e. annotations specifying the meaning of a particular text within a wiki article. These annotations enable the editors to make certain facts within wiki articles accessible to programmes, which in turn makes it easier for users to find or reuse the information. However, since human resources are usually sparse, an extensive manual annotation of large information resources is not feasible. In most such cases, semi-automatic annotation is a more efficient alternative potentially allowing to significantly increase the number of semantic annotations that can be obtained for a given information resource with the same effort.

The ontology-based annotation tool (OBA) has been developed as a plugin for GATE [1] – a popular open source framework for analysis and processing of natural language texts – and, therefore, provides access to a wide range of features integrated in the aforementioned framework. OBA is one of the results of the project NanOn[1] aiming at ontology-supported literature search. Within the NanOn project, a hand-crafted ontology specified in the Web Ontology Language OWL 2 DL [3] modeling the scientific domain of nano technology has been used as the core resource to automatically analyze scientific documents for the occurrence of ontology classes and relations based on a set of lexical patterns. In this project, OBA was the main tool, on the one hand, supporting the acquisition

[1] http://www.aifb.kit.edu/web/NanOn.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 468–473, 2015.
DOI: 10.1007/978-3-662-46641-4_47

of lexical patterns and extension of the ontology structure, and, on the other hand, supporting an automatic annotation of a large text corpus based on the obtained lexical patterns. In general, OBA can be used to support the following tasks relevant to semantic annotation:

- Manual annotation of text with classes and relations of an ontology in order to enrich the amount of lexical patterns or to extend the ontology itself with new classes and relations;
- Automatic annotation of a text or a corpus according to a given set of NLP-based patterns and the corresponding domain and range restrictions resulting in a population of the ontology with instances and relations represented in various formats, for instance, as RDF triples or OWL 2 DL instances and relations;
- Evaluation of automatically or manually created annotations (for instance, in order to estimate precision and recall).

The complete GATE package with OBA plugin including the source code is available at http://people.aifb.kit.edu/nni/GATE.zip.

2 Annotation Properties

OBA comes with a set of pre-defined general annotation types and general settings determining the scope, in which the user can define different types of annotations called *annotation properties* (for instance, those representing types of lexical patterns). First of all, we distinguish between patterns for classes and patterns for relations due to the possibility of additional information about the domain and range annotations of the corresponding relation. In case of class patterns, in addition to the possibility of a part-of-speech restriction, the possible settings include different ways to deal with plural forms, case deviations and non-matching word boundaries. In case of relation patterns, we further distinguish between

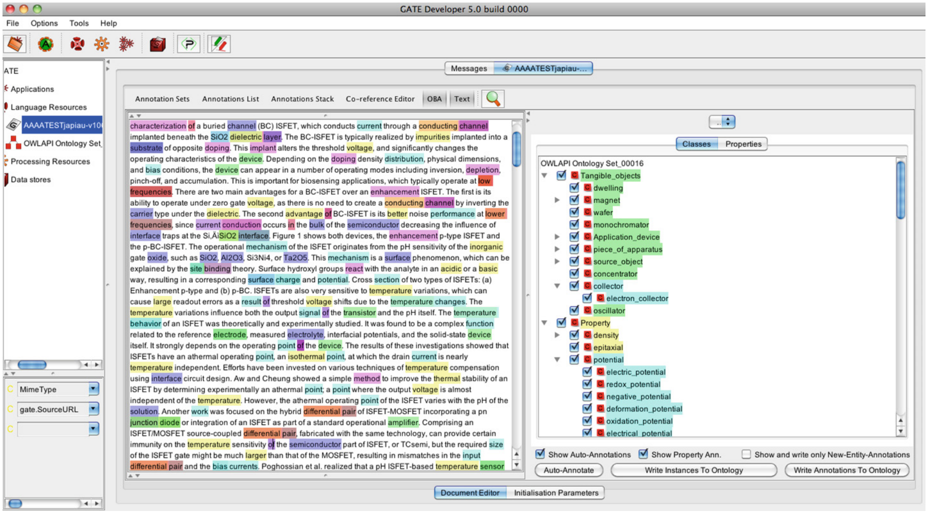
- string-based patterns, i.e., patterns containing an expression that must appear in the text between domain and range annotations (for instance, a preposition such as *by, of, as*), and
- patterns based only on the information about the domain and range annotations (for instance, two nouns within a single noun group such as *surface defects* or *ITO nanoparticles*).

In both cases, the settings include the allowed annotation properties for the domain and range annotations and a distance in characters between the subject and object annotations.

Example 1. *Within the project NanOn, we used, among others, the following annotation properties:*

- *patternNomen (class annotation, plural forms, no case-sensitivity, word boundaries, POS:NN)*

- *patternAcronym (class annotation, no plural forms, case-sensitivity, word boundaries, no POS)*
- *patternNomenNomen (relation annotation, POS domain: NN, POS range: NN, no string match required)*
- *patternsModifierNomen (relation annotation, POS domain: JJ—VBG—VBN, POS range: NN, no string match required)*
- *patternNomenVerbP (relation annotation, POS domain: NN, POS range: VBG, no string match required)*



characterization of a buried channel (BC) ISFET, which conducts current through a conducting channel implanted beneath the SiO2 dielectric layer. The BC-ISFET is typically realized by impurities implanted into a substrate of opposite doping. This implant alters the threshold voltage, and significantly changes the operating characteristics of the device. Depending on the doping density distribution, physical dimensions, and bias conditions, the device can appear in a number of operating modes including inversion, depletion, pinch-off, and accumulation. This is important for biosensing applications, which typically operate at low frequencies. There are two main advantages for a BC-ISFET over an enhancement ISFET. The first is its ability to operate under zero gate voltage, as there is no need to create a conducting channel by inverting the carrier type under the dielectric. The second advantage of BC-ISFET is its better noise performance at lower frequencies, since current conduction occurs in the bulk of the semiconductor decreasing the influence of interface traps at the Si,ÄìSiO2 interface. Figure 1 shows both devices, the enhancement p-type ISFET and the p-BC-ISFET. The operational mechanism of the ISFET originates from the pH sensitivity of the inorganic gate oxide, such as SiO2, Al2O3, Si3Ni4, or Ta2O5. This mechanism is a surface phenomenon, which can be explained by the site binding theory. Surface hydroxyl groups react with the analyte in an acidic or a basic way, resulting in a corresponding surface charge and potential. Cross section of two types of ISFETs: (a) Enhancement p-type and (b) p-BC. ISFETs are also very sensitive to temperature variations, which can cause large readout errors as a result of threshold voltage shifts due to the temperature changes. The temperature variations influence both the output signal of the transistor and the pH itself. The temperature behavior of an ISFET was theoretically and experimentally studied. It was found to be a complex function related to the reference electrode, measured electrolyte, interfacial potentials, and the solid-state device itself. It strongly depends on the operating point of the device. The results of these investigations showed that ISFETs have an athermal operating point, an isothermal point, at which the drain current is nearly temperature independent. Efforts have been invested on various techniques of temperature compensation using interface circuit design. Aw and Cheung showed a simple method to improve the thermal stability of an ISFET by determining experimentally an athermal point; a point where the output voltage is almost independent of the temperature. However, the athermal operating point of the ISFET varies with the pH of the solution. Another work was focused on the hybrid differential pair of ISFET-MOSFET incorporating a pn junction diode or integration of an ISFET as part of a standard operational amplifier. Comprising an ISFET/MOSFET source-coupled differential pair, fabricated with the same technology, can provide certain immunity on the temperature sensitivity of the semiconductor part of ISFET, or TCsemi, but the required size of the ISFET gate might be much larger than that of the MOSFET, resulting in mismatches in the input differential pair and the bias currents. Poghossian et al. realized that a pH ISFET-based temperature sensor

☑ Show Auto-Annotations ☑ Show Property Ann. □ Show and write only New-Entity-Annotations

Fig. 1. General user interface of OBA

3 User Interface and Selected Features

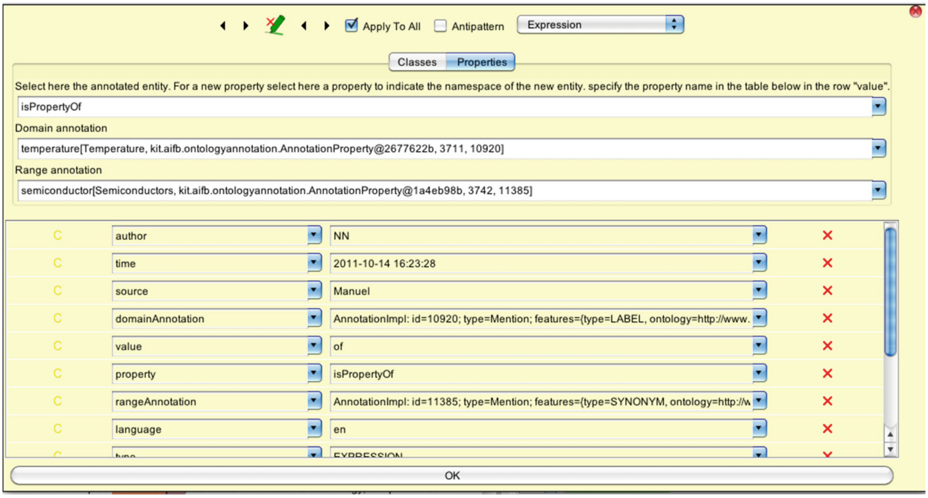
Figure 1 shows the general UI of OBA. The plugin extends OCAT[2], which provides basic functionality for ontology-based annotation. Based on the OWL API [2], OBA can load an ontology in all supported formats[3]. The annotations (in this case created mostly automatically) are shown in the color of the corresponding ontology entities.[4] In order to simplify the evaluation of annotations, several general filters (for instance, hiding all automatically generated annotations or all relation annotations) were defined in addition to the check-boxes in front of each ontology entity. In addition to the possibility to save the state of an annotated document as XML, OBA allows the user to convert annotations in the following two ways:

[2] http://gate.ac.uk/sale/tao/splitch14.html#sec:ontologies:OCAT.

[3] http://owlapi.sourceforge.net/index.html.

[4] Due to the large number of classes, by default colors are only distinguished for direct subclasses of owl:Thing. However, the color of each entity can also be selected manually using the context menu.

1. Annotations can be converted into new ontology entities and lexical patterns. The latter are stored within the ontology itself as values of OWL 2 annotation properties (rdfs:label is one of such properties). It is up to the user to define OWL 2 annotation properties representing a particular type of a lexical pattern. Given a mapping of pattern types to OWL 2 annotation properties specified by the user, the annotation values are stored in a particular format also representing the metadata of each annotation. Using the context menu, the user can explore for each ontology entity the values of OWL 2 annotation properties stored in the ontology including the values of rdfs:label or any user-defined annotation property such as those representing lexical patterns. The corresponding annotation property viewer also shows the available metadata of lexical patterns.
2. Annotations can be converted into actual *semantic annotations* in form of class and relation instances that can be stored either as OWL 2 instances, RDF triples or quadruples, additionally containing a reference to the source document and a small excerpt.



☑ Apply To All ☐ Antipattern Expression

Select here the annotated entity. For a new property select here a property to indicate the namespace of the new entity. specify the property name in the table below in the row "value".

isPropertyOf

Domain annotation

temperature[Temperature, kit.aifb.ontologyannotation.AnnotationProperty@2677622b, 3711, 10920]

Range annotation

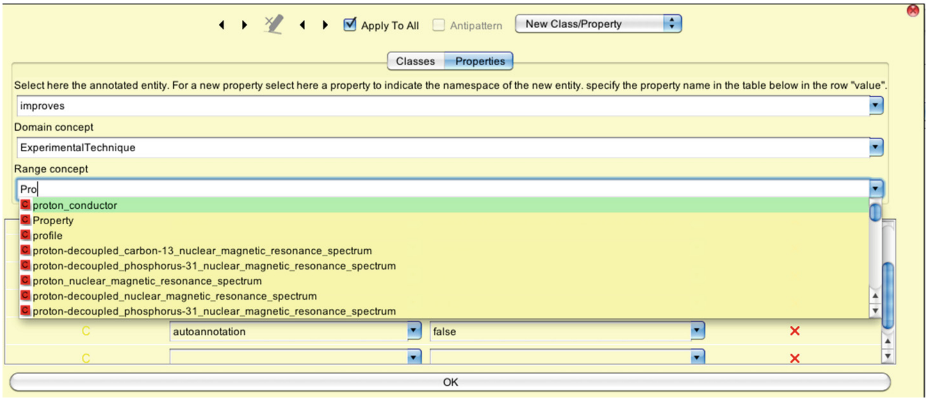
semiconductor[Semiconductors, kit.aifb.ontologyannotation.AnnotationProperty@1a4eb98b, 3742, 11385]

C	author	NN	✕
C	time	2011-10-14 16:23:28	✕
C	source	Manuel	✕
C	domainAnnotation	AnnotationImpl: id=10920; type=Mention; features={type=LABEL, ontology=http://www.	✕
C	value	of	✕
C	property	isPropertyOf	✕
C	rangeAnnotation	AnnotationImpl: id=11385; type=Mention; features={type=SYNONYM, ontology=http://w	✕
C	language	en	✕
C	type	EXPRESSION	✕

Fig. 2. Creating new pattern annotations with OBA

Figure 2 shows the annotation editor, which is activated by selecting a part of the text in order to create a new annotation. While, in case of class annotations, the selection of the corresponding annotation property and a class is sufficient to create the annotation, in case of relation annotations, the user additionally has to specify the domain and range annotations from the list of existing class annotations.

In addition to the required information such as the corresponding ontology entity, offset within the document and the annotation property, the annotation editor allows the user to edit various metadata entries determined by default every time a pattern annotation is created. Among other things, they include the author of the pattern, document in which the pattern has been annotated, time and date as well as the summary of the corresponding domain and range annotations in case of relation annotations.



☑ Apply To All □ Antipattern New Class/Property

Select here the annotated entity. For a new property select here a property to indicate the namespace of the new entity. specify the property name in the table below in the row "value".

improves

Domain concept

ExperimentalTechnique

Range concept

Pro

autoannotation false

Classes
Properties
proton_conductor
Property
profile
proton-decoupled_carbon-13_nuclear_magnetic_resonance_spectrum
proton-decoupled_phosphorus-31_nuclear_magnetic_resonance_spectrum
proton_nuclear_magnetic_resonance_spectrum
proton-decoupled_nuclear_magnetic_resonance_spectrum
proton-decoupled_phosphorus-31_nuclear_magnetic_resonance_spectrum
OK

Fig. 3. Creating new classes and properties with OBA

In order to create a new class or relation (Fig. 3), the pre-defined annotation property “New Class/Property” must be selected. Subsequently, the corresponding superclass or superrelation must be selected along with the domain and range classes.

4 Demonstration

The goal of the demonstration is to show how, given an ontology and a set of texts, the various features of OBA are employed to define a set of lexical patterns and automatically annotate a corpus with the pattern set. The demonstration will be split in four main scenarios:

- Definition of annotation properties based on the requirements of the application scenario: Here, we explain the pre-defined settings and the usage of NLP information in order to fine-tune the performance of the annotation.
- Manual annotation using defined annotation properties and classes/relations: We show how new lexical patterns in form of annotations can be created, edited or deleted and explain the available annotation metadata.
- Extension of the ontology with new classes and relations: Here, we demonstrate, how, based on the information occurring in a text, the ontology can be extended on the fly during the annotation.
- Automatic annotation of corpora: Finally, we show how the created lexical patterns can be used to automatically annotate documents.

The visitors will also be shown how to download and install the tool for their own use.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011). http://tinyurl.com/gatebook
2. Horridge, M., Bechhofer, S.: The OWL API: a java API for working with OWL 2 ontologies. In: OWLED (2009)
3. Motik, B., Patel-Schneider, P.F., Cuenca Grau, B.: OWL 2 web ontology language: direct semantics. W3C Recommendation, October 2009. http://www.w3.org/TR/owl2-direct-semantics/

HadoopSPARQL: A Hadoop-Based Engine for Multiple SPARQL Query Answering

Chang Liu[1](✉), Jun Qu[1], Guilin Qi[2], Haofen Wang[1], and Yong Yu[1]

[1] Shanghai Jiaotong University, Shanghai, China
{liuchang,qujun51319,whfcarter,whfcarter,yyu}@apex.sjtu.edu.cn
[2] Southeast University, Nanjing, China
gqi@seu.edu.cn

1 Introduction

An increasing amount of data represented using Resource Description Framework (RDF) have appeared on the Semantic Web. By September 2011, datasets from Linked Open Data [4] had grown to 31 billion RDF triples, interlinked by around 504 million RDF links. As a consequence, it is extremely challenging to deal with the scalability issue of handling such large amount of semantic data.

SPARQL [7] is a standard query language for RDF datasets. There has been a lot of work [2,5] to handle SPARQL queries. However, most of them only treat SPARQL as a transaction-based query language, and consider low latency query answering time as an important design requirement. Furthermore, the query engine processes one query at a time and concentrates on single query optimizations. Nevertheless, users can also use SPARQL in very different scenarios. For example, two users may submit queries to a dataset about publications at the same time. The first user wants to get a list containing all authors who publish at least one proceeding and at least one article while the second user wants to get a list containing all authors who publish at least one article but not necessarily publish a proceeding. Then, Query 1 and Query 2 in Fig. 1 are submitted at the same time.

In this scenario, the system is highly desired to process these queries in parallel. [3] discussed a multi-query optimization algorithm for SPARQL. However, their method is based on revision from a set of basic graph queries into a new set of queries which may include OPTIONAL restriction. Introducing OPTIONAL restriction into queries, however, potentially increases the computational complexity. We find that the essential optimization opportunity for multiple queries lies in identifying common subqueries which may cause duplicate calculations. The engine should avoid such redundant calculations. At last, since the datasets are growing larger and larger, the scalability problem becomes more and more severe. MapReduce [1] has proved as a scalable framework to handle high latency and highly parallel tasks over large-size datasets. A natural idea is to leverage MapReduce techniques to overcome the above challenges. To this end, we built a system, called HadoopSPARQL, based on Hadoop[1].

[1] http://hadoop.apache.org/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 474–479, 2015.
DOI: 10.1007/978-3-662-46641-4_48

Table 1. Sample SPARQL queries

Query 1

```
SELECT DISTINCT ?person ?name
WHERE { ?article rdf:type bench:Article.
?article dc:creator ?person.
?inproc rdf:type bench:Inproceedings.
?inproc dc:creator ?person.
?person foaf:name ?name. }
```

Query 2

```
SELECT DISTINCT ?person ?name
WHERE { ?article rdf:type bench:Article.
?article dc:creator ?person.
?person foaf:name ?name }
```

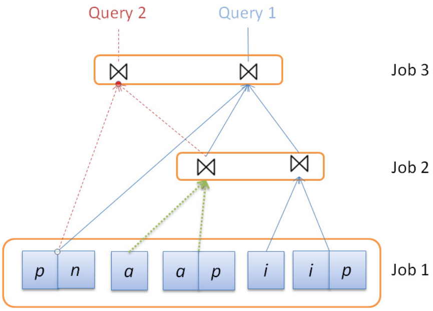
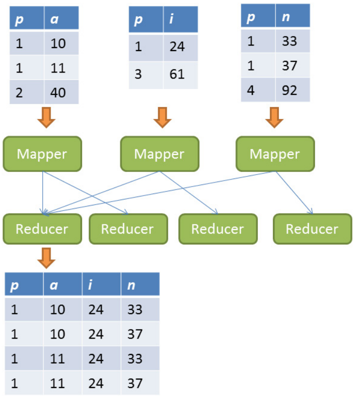


Fig. 1. An execution plan for Query 1 and Query 2



p	a
1	10
1	11
2	40

p	i
1	24
3	61

p	n
1	33
1	37
4	92

p	a	i	n
1	10	24	33
1	10	24	37
1	11	24	33
1	11	24	37

Fig. 2. Execution of a join operator

The major feature of HadoopSPARQL is that it allows the users to submit multiple queries at the same time. To handle multiple queries, we propose an algorithm to detect the common subqueries. To leverage the MapReduce framework, we use multi-way join operator instead of the traditional two-way join operator. Therefore we propose our new optimization algorithm to calculate the best join order. Furthermore, HadoopSPARQL provides a Web interface to allow accessing the underlying system using Web browsers.

2 Query Engine

The query engine consists of an optimizer to handle multiple SPARQL queries simultaneously and a Hadoop-based evaluator. In this section, we explain the key problems and design choices of the optimizer and the evaluator.

2.1 Operators

There are two kinds of operators, *data-loading operator* and join operator. Both of these two operators will produce a binding set as the result. A data-loading operator always corresponds to a triple pattern in the query. To evaluate a data-loading operator, we only need to load the data from HDFS files. Thus a data-loading operator has no input operator.

A join operator takes $k \geq 2$ inputs and produces a binding set. Formally speaking, a join operator is defined by a set of binding sets B_1, B_2, ..., B_k ($k \geq 2$) and a set of *key variables* K so that $K = S(B_1) \cap S(B_2) \cap ... \cap S(B_k)$ and $(S(B_i) - K) \cap (S(B_j) - K) = \emptyset$ for every $1 \leq i < j \leq k$. Here we use $S(B_i)$ to denote the schema of the binding set B_i. We can see that the definition of our join operator differs from the traditional multiple join operator which is widely used in RDBMS in the following aspect: We do not allow the schemas of any two input binding sets to share the same variable except the keys. For example, we do not allow to join three binding sets with schemas $\{x, y, a\}$, $\{x, y, b\}$ and $\{x, c\}$ on key x. This restriction will bring us two benefits: 1) the optimizer can leverage this information to accelerate the enumeration of candidate plans; and 2) the evaluator can efficiently calculate this kind of join.

2.2 Optimizer

The optimizer will translate a batch of SPARQL queries into one execution plan. An execution plan is a Directed Acyclic Graph (DAG) where each node in the graph represents an operator. Those nodes with 0 indegree are data-loading operators and those internal operators are join operators. The major task of an optimizer is to find the optimal execution plan. Since our optimizer is designed for multiple queries, it is composed of two parts: the first part detects the common sub-queries, and the second part employs a cost-based algorithm to generate the optimal execution plan.

The main task of the first part is to detect those duplicate sub-queries. Considering the example queries in Table 1, the following triple patterns appear in both of the two queries: $\langle$?article, rdf : type, bench : Article$\rangle$, $\langle$?article, dc:creator, ?person$\rangle$, and $\langle$?person, foaf : name, ?name$\rangle$.

Since the results of join operators will be stored in HDFS, we can reuse these results. If the two queries listed in Table 1 are evaluated together, we only need to calculate the result of the above sub-query once. By exploiting such duplications, a lot of redundant operations can be saved so that the performance can be improved.

Given a set of operators, one important problem is how to find the optimal execution plan, e.g. the best join orders. Here we employ a cost-based optimization algorithm to achieve this goal. We generate all potential execution plans, and estimate the cost of each of them. We choose the execution plan with minimal cost as our optimal plan and submit it to the query evaluator. However, there are always too many potential execution plans so that enumeration of all plans wastes a large amount of time. Several pruning techniques are applied to the optimizer to achieve an acceptable performance. In our use case study, the optimal plan of each query can be found within one second.

Figure 1 illustrates the execution plan for the example queries in Table 1. We use a, i, n and p to represent the variable ?article, ?inproc, ?name and ?person respectively. The bottom layer contains five data-loading operators corresponding to the five triple patterns. For example, a data-loading operator with schema p, n corresponds to a triple pattern $\langle$?person, foaf : name, ?name$\rangle$.

All nodes in the middle layer and top layer are join operators. We use solid and dashed lines to illustrate the query execution path of Query 1 and Query 2 respectively. We use dotted lines to illustrate the common sub-query. Notice the common sub-query contains only two triple patterns $\langle ?\texttt{article}, \texttt{rdf} : \texttt{type}, \texttt{bench:Article}\rangle$ and $\langle ?\texttt{article}, \texttt{dc} : \texttt{creator}, ?\texttt{person}\rangle$ instead of the three ones listed above. The reason is that in such a way, the engine only executes three Hadoop jobs instead of four, such that the execution time is reduced.

2.3 Evaluator

The evaluator will translate the execution plan (a DAG) into Hadoop jobs, and submit it to the Hadoop cluster for evaluation. The evaluator iteratively generates the Hadoop jobs. In the first round, all nodes in DAG with 0 indegree will be grouped into one job and removed from the DAG. Since all these nodes correspond to data-loading operators, the first Hadoop job will load the data from index files. Then in each iteration, all nodes with 0 indegree will be grouped into one job and removed. Since all internal nodes in the original DAG correspond to join operators, the second and later Hadoop jobs will perform joins.

For example, considering the execution plan given in Fig. 1, all the operators are grouped into three jobs which are illustrated by orange boxes. Job 1 performs all the data-loading operators, while Job 2 and Job 3 do the joins.

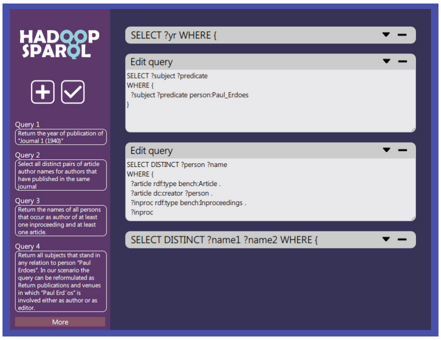
The implementation of data-loading operators is straightforward, thus we only discuss the implementation of join operators. A set of join operators is translated into a Hadoop job. Each join will be assigned a unique ID. If there is a join with ID id defined by input binding sets $B_1, ..., B_k$ and variable key set K, the mappers will scan all binding sets $B_1, ..., B_k$. When a mapper reads a binding $b_i \in B_i$, it will emit $(id, b_i(K))$ as map output key, and $(b_i(S(B_i) - K))$ as map output value. Therefore for each join id, every $b_i \in B_i$ with the same $b_i(K)$ will be grouped into the same reducer. Since we restrict that $(S(B_i)-K)\cap(S(B_j)-K) = \emptyset$ for every $1 \leq i < j \leq k$, the join result $B_1 \bowtie B_2 \bowtie ... \bowtie B_k$ equals to $\bigcup_{key} B_1^{key} \times ... \times B_k^{key}$ where $B_1^{key} = \{b : b \in B_1 \wedge b(K) = key\}$ and × represents the Cartesian product. Therefore, to calculate the join, we only need to calculate the Cartesian product. Figure 2 gives an example to illustrate this calculation.

3 Demonstration Scenario

The goal of the demonstration is to illustrate how to use HadoopSPARQL to execute SPARQL queries. HadoopSPARQL provides a Web interface which can be used to submit queries and view the results. The dataset and queries will be described in Sect. 3.1 while Sect. 3.2 will describe the functionalities of Web interface.

3.1 Dataset and Queries

The demonstration uses a synthesis dataset called SP^{2}Bench [6] which is designed to test the performance of a SPARQL query engine. SP^{2}Bench generates data



Query 1
Return the year of publication of "Journal 1 (1940)"

Query 2
Select all distinct pairs of article author names for authors that have published in the same journal

Query 3
Return the names of all persons that occur as author of at least one inproceeding and at least one article.

Query 4
Return all subjects that stand in any relation to person "Paul Erdoes". In our scenario the query can be reformulated as Return publications and venues in which "Paul Erd¨os" is involved either as author or as editor.

```
SELECT ?yr WHERE {
```

Edit query

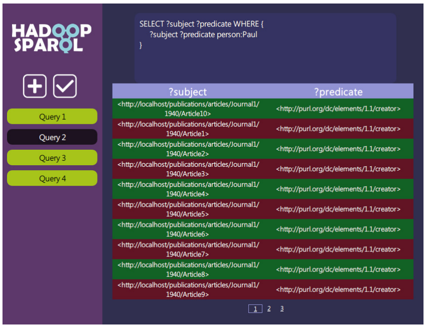
```
SELECT ?subject ?predicate
WHERE {
  ?subject ?predicate person:Paul_Erdoes
}
```

Edit query

```
SELECT DISTINCT ?person ?name
WHERE {
  ?article rdf:type bench:Article .
  ?article dc:creator ?person .
  ?inproc rdf:type bench:Inproceedings .
  ?inproc
```

```
SELECT DISTINCT ?name1 ?name2 WHERE {
```

Fig. 3. HadoopSPARQL Query UI



```
SELECT ?subject ?predicate WHERE {
    ?subject ?predicate person:Paul
}
```

?subject	?predicate
<http://localhost/publications/articles/Journal1/1940/Article10>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article1>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article2>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article3>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article4>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article5>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article6>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article7>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article8>	<http://purl.org/dc/elements/1.1/creator>
<http://localhost/publications/articles/Journal1/1940/Article9>	<http://purl.org/dc/elements/1.1/creator>

Fig. 4. HadoopSPARQL Result UI

based on DBLP database which includes authors, articles, conferences, journals and so on. One can use SP^{2}Bench to generate the dataset by specifying the data volume. SP^{2}Bench provides 12 testing queries which include the two queries in Table 1. Most of the benchmark queries are BGP queries and only two containing union and optional constructions. We use all queries as samples except the two unsupported by our system.

3.2 Web Interface

HadoopSPARQL provides a Web interface for users to submit queries through Web browser. The interface is written in JavaScript. Figure 3 shows the query UI. Users can add a new query by clicking the plus button, and remove a query by clicking the minus button. All queries are listed in the left of the screen. A submit button with tick symbol is used to submit these queries to the HadoopSPARQL system for evaluation. On the side bar, several sample queries from SP^{2}Bench are listed. Users can click on each of them to add it to their query set. Once the queries are submitted, the system will provide a link which can be used to view the results.

Figure 4 shows the result page. Queries submitted by the user are listed in the side bar. By clicking one of the queries, the content of the selected query is showed in the top while the results are listed below. The number of results usually exceeds one page's limitation, so there is a list of buttons allowing the user to view the whole result set.

References

1. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: Proceedings of OSDI 2004, pp. 137–147 (2004)
2. Huang, J., Abadi, D.J., Ren, K.: Scalable SPARQL querying of large RDF graphs. In: Proceedings of VLDB 2011, pp. 1123–1134 (2011)
3. Le, W., Kementsietsidis, A., Duan, S., Li, F.: Scalable multi-query optimization for SPARQL. In: Proceedings of ICDE 2012 (2012)

4. LOD. http://linkeddata.org/
5. Neumann, T., Weikum, G.: The RDF-3X engine for scalable management of RDF data. VLDB J. **19**(1), 91–113 (2010)
6. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP2Bench: a SPARQL performance benchmark. CoRR (2008)
7. SPARQL. http://www.w3.org/TR/rdf-sparql-query/

DEFENDER: A DEcomposer for quEries agaiNst feDERations of Endpoints

Gabriela Montoya[1](✉), Maria-Esther Vidal[1](✉), and Maribel Acosta[1,2]

[1] Universidad Simón Bolívar, Caracas, Venezuela
{gmontoya,mvidal,macosta}@ldc.usb.ve
[2] Institute AIFB, Karlsruhe Institute of Technology, Karlsruhe, Germany
maribel.acosta@kit.edu

Abstract. We present DEFENDER and illustrate the benefits of identifying promising query decompositions and efficient plans that combine results from federations of SPARQL endpoints. DEFENDER is a query decomposer that implements a two-fold approach. First, triple patterns in a SPARQL query are decomposed into simple sub-queries that can be completely executed on one endpoint. Second, sub-queries are combined into a feasible bushy tree plan where the number of joins is maximized and the height of tree is minimized. We demonstrate DEFENDER and compare its performance with respect to state-of-the-art RDF engines for queries of diverse complexity, networks with different delays, and dataset differently distributed among a variety of endpoints.

1 Introduction

During the last years, the number of datasets in the Linked Open Data cloud has exploded as well as the number of SPARQL endpoints that provide access to these datasets[1]. Although existing endpoints should be able to execute any SPARQL query, some endpoints reject the execution of queries whose estimated execution time or cardinality is greater than a certain number, while others simply time out without producing any answer. With the appropriate endpoint technology not ready, there is a need to develop techniques to decompose complex queries into queries that can be executed as well as strategies to integrate retrieved data. We present DEFENDER, a decomposer for queries against federations of endpoints that stores information about the available endpoints and the ontologies used to describe the data accessible through the endpoints, and decomposes queries into sub-queries that can be executed by the selected endpoints. Additionally, DEFENDER combines sub-queries into an execution plan where the number of joins is maximized and the height is minimized. The former condition implies that the number of Cartesian products is minimized, while the latter benefits the generation of plans where leaves can be independently executed. DEFENDER was implemented on top of ANAPSID [1], an adaptive query engine for the SPARQL 1.1 federation extension[2] that adapts query execution

[1] http://labs.mondeca.com/sparqlEndpointsStatus/.

[2] http://www.w3.org/TR/rdf-sparql1-query/.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 480–484, 2015.
DOI: 10.1007/978-3-662-46641-4_49

schedulers to data availability and runtime conditions. We demonstrate the performance of the plans identified by DEFENDER, and show that these plans are competitive with the plans generated by existing RDF engines. A portal that publishes results presented at the demo section can be found at http://code.google.com/p/defender-portal/.

2 The DEFENDER Architecture

DEFENDER comprises a *Query Planner*, an *Adaptive Query Engine* and a *Catalog of Endpoint Descriptions*. The DEFENDER *Query Planner* is composed of two main components: the *Query Decomposer* and the *Heuristic-Based Query Optimizer*. The former divides sets of triple patterns in SPARQL 1.0 queries into sub-sets of triple patterns (TPs) that can be executed by the same endpoint and: *(i)* share exactly one variable, or *(ii)* share one variable with at least one of the TPs in the sub-query. The query decomposer begins creating single sub-queries with TPs, then it merges the sub-queries that share exactly one variable, and repeats this process until a fixed-point is reached in the process of creating the sub-queries. Then, TPs that share one variable with any TPs are added (Fig. 1).

Once the SPARQL 1.1 query is created, heuristic-based optimization techniques are followed to generate a bushy tree plan, where the leaves correspond to the sub-queries of TPs previously identified. Optimization techniques do not rely on statistics recollected from the endpoints, just information about predicates in the datasets accessible through the endpoints. A greedy heuristic-based algorithm is implemented; it traverses the space of bushy plans in iterations and outputs a bushy tree plan of the SPARQL 1.1 query with the service clause where the number of joins is maximized and the height of tree is minimized. Thus, the size of intermediate results and the number of HTTP requests are reduced.

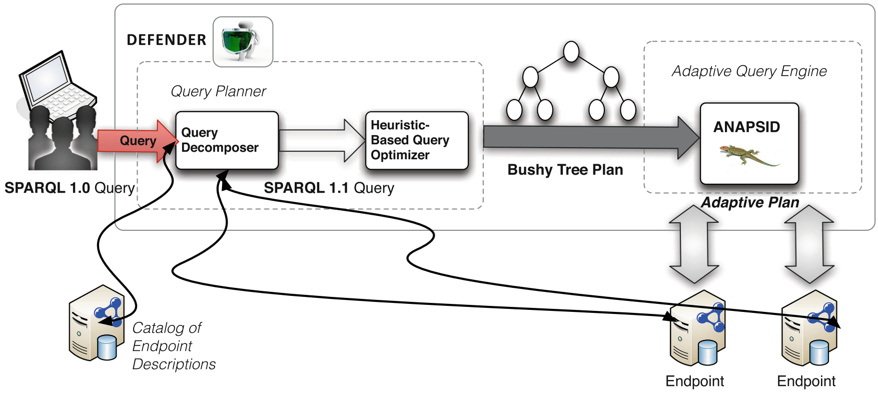


Fig. 1. The DEFENDER architecture

3 Demonstration of Use Cases

Consider the following SPARQL 1.0 query: *Retrieve diseases and genes associated with drugs tested in clinical trials where Prostate Cancer was studied.*

```
SELECT DISTINCT ?II ?D ?GN2
WHERE {
   ?CT1<http://data.linkedct.org/resource/linkedct/condition> ?C1 .
   ?CT1 <http://data.linkedct.org/resource/linkedct/intervention> ?I .
   ?I <http://data.linkedct.org/resource/linkedct/intervention_type> ”Drug” .
   ?C1 <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?D.
   ?I <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?II.
   ?C <http://data.linkedct.org/resource/linkedct/condition_name> ”Prostate Cancer” .
   ?CT <http://data.linkedct.org/resource/linkedct/intervention> ?I .
   ?CT <http://data.linkedct.org/resource/linkedct/condition> ?C .
   ?D <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/associatedGene> ?GN2 .
   ?D <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/possibleDrug> ?II}
```

The answer is composed of 192 tuples when data from Diseasome and LinkedCT are retrieved. However, if the query is run against existing endpoints, Diseasome[3] or LinkedCT[4], the answer is empty. This problem is caused by the need to traverse links between these datasets to answer the query. Still, the majority of existing endpoints have been created for lightweight use and they are not able to dereference data from other datasets. Existing approaches [2,4] are able to decompose this query into sub-queries; although these approaches can be very efficient and effective, if queries are comprised of a large number of triple patterns that can be executed by different endpoints, they may time out without producing any answer. To overcome these limitations, DEFENDER decomposes the former query into the following SPARQL 1.1 query with the service clause, which is comprised of four *star-shaped* sub-queries. These sub-queries are composed of TPs that can be executed by the same endpoint, and that share exactly one variable or share one variable with at least one of the TPs in the sub-query.

```
SELECT ?II ?D ?GN2
WHERE {
   { SERVICE <http://www4.wiwiss.fu-berlin.de/diseasome/sparql> {
      ?D <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/associatedGene> ?GN2 .
      ?D <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/possibleDrug> ?II. } }.
   { SERVICE <http://linkedct.org/sparql> {
      ?I <http://data.linkedct.org/resource/linkedct/intervention_type> ”Drug” .
      ?CT1 <http://data.linkedct.org/resource/linkedct/condition> ?C1 .
      ?CT1 <http://data.linkedct.org/resource/linkedct/intervention> ?I .
      ?C1 <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?D. }} .
   { SERVICE <http://linkedct.org/sparql> {
      ?C <http://data.linkedct.org/resource/linkedct/condition_name> ”Prostate Cancer” .
      ?CT <http://data.linkedct.org/resource/linkedct/condition> ?C. }}.
   { SERVICE <http://linkedct.org/sparql> {
      ?I <http://data.linkedct.org/resource/linkedct/intervention_type> ”Drug” .
      ?I <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?II .
      ?CT <http://data.linkedct.org/resource/linkedct/intervention> ?I }}. }
```

Once the query is decomposed, DEFENDER builds a plan that combines the sub-queries; the generated plan minimizes intermediate results and the number

[3] http://www4.wiwiss.fu-berlin.de/diseasome/sparql.

[4] http://data.linkedct.org/sparql/.

of HTTP requests. The DEFENDER portal[5] presents the behavior of 36 queries against the FedBench collections: Cross-Domain, Linked Data and Life Science [3]. These queries include 25 FedBench queries and 11 complex queries[6]; complex queries are comprised of between 6 and 48 triple patterns and can be decomposed into up to 8 sub-queries. FedBench collections were accessed through 9 Virtuoso[7] endpoints which time out at 240 s. or 71,000 tuples. Endpoint simulators were used to configured network latency, endpoint availability and network packet size; simulators are comprised of servers and proxies. Servers correspond to real endpoints that are contacted by the proxies, which send data between servers and RDF engines following a particular transfer delay and respecting a given network packet size. Different types of delays are illustrated; all of them follow a Gamma distribution with different average latency to simulate perfect, fast, and medium-fast, and set up the network packet size. Additionally, we produced SPARQL 1.1 queries for different decompositions and executed these queries in ARQ 2.8.8. BSD-style[8] that supports the Federation extension of SPARQL 1.1. The behavior of ARQ, DEFENDER, and FedX [4] is demonstrated in these network configurations. Also, it can be observed the impact of different decompositions on performance and answer completeness.

Effects of network delays on query execution performance. In an ideal network without delays, it can be seen that ARQ may time out without producing any answer, while DEFENDER may be able to finalize the query processing task before reaching a timeout of 1,800 s. On the other hand, delays may considerably affect the performance of DEFENDER and ARQ depending on the type of decomposition. For example, the majority of queries may either time out or produce empty answers when unitary sub-queries are executed, i.e., when sub-queries are comprised of only one triple pattern. In contrast, plans comprised of non-unitary sub-queries, i.e., the ones identified by DEFENDER, are not equality affected by network delays. Although DEFENDER performance can be deteriorated, execution time of around half of the queries remain in the same order of magnitude with respect to these queries executed in a perfect network. ARQ is also able to execute some of these plans in the delayed networks without decreasing performance significantly. The observed behavior of the plans comprised of DEFENDER sub-queries is caused by a reduced number of HTTP requests as well as the size of intermediate results which usually can be delivered from the endpoints in a small number of network packets. Thus, even in presence of delayed networks, the performance of these plans is acceptable.

Answer completeness when different decompositions are executed. In a perfect network, DEFENDER and ARQ produce all the answers for the majority of the queries before timing out. But, when delays are considered the quality is decreased, mainly when plans are comprised of unitary

[5] http://code.google.com/p/defender-portal/.

[6] http://www.ldc.usb.ve/~mvidal/FedBench/queries/ComplexQueries.

[7] http://virtuoso.openlinksw.com/.

[8] http://sourceforge.net/projects/jena/.

sub-queries are executed. These results are consequence of the poor performance exhibited by both engines when unitary sub-queries are run. However, if intermediate results remain small, quality is not equally affected in DEFENDER plans even in presence of network latency. We also show the scenario where the same predicate is accessible through different endpoints and demonstrate how dataset distributions impact on answer completeness.

Effects of the plan shape on execution time and answer completeness. Optimal bushy trees and left-linear plans are reported for each query and executed in DEFENDER. These plans may reduce execution time by up to one order of magnitude when optimal bushy trees are executed. FedX also exhibits good performance when FedBench queries are executed, being able to produce most of the answers. In contrast, DEFENDER plans may outperform the ones generated by FedX when the queries are comprised of a large number of triple patterns. Bushy trees are able to scale up to complex queries, and are competitive with other execution strategies when simple queries are processed. Finally, the execution time of plans comprised of DEFENDER sub-queries is low; these plans may reduce by up to two orders of magnitude the time consumed by plans comprised of unitary sub-queries.

4 Conclusions and Future Work

We present DEFENDER and illustrate results that suggest that our proposed techniques may reduce execution times by up to two orders of magnitude, and are able to produce answers when other engines fail. Also, depending on data distributions among different endpoints and transfer delays, DEFENDER query plans overcome plans generated by existing RDF engines if size of intermediate results and the number of HTTP requests are reduced. In the future we plan to provide DEFENDER endpoint for real-world applications.

References

1. Acosta, M., Vidal, M.-E., Lampo, T., Castillo, J., Ruckhaus, E.: ANAPSID: an adaptive query processing engine for SPARQL endpoints. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 18–34. Springer, Heidelberg (2011)
2. Görlitz, O., Staab, S.: SPLENDID: SPARQL endpoint federation exploiting VOID descriptions. In: Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany (2011)
3. Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: FedBench: a benchmark suite for federated semantic data query processing. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 585–600. Springer, Heidelberg (2011)
4. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: optimization techniques for federated query processing on linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 601–616. Springer, Heidelberg (2011)

Poster Session

A LarKC Approach to Development of Service-Oriented Semantic Reasoning Applications

Alexey Cheptsov(✉)

High-Performance Computing Center Stuttgart,
Nobelstr. 19, 70569 Stuttgart, Germany
cheptsov@hlrs.de

Abstract. Reasoning is one of the essential application areas of the modern Semantic Web. At present, the semantic reasoning algorithms are facing significant challenges when dealing with the emergence of the Internet-scale knowledge bases, comprising extremely large amounts of data. The traditional reasoning approaches have only been approved for small, closed, trustworthy, consistent, coherent and static data domains. As such, they are not well-suited to be applied in data-intensive applications aiming on the Internet scale. We introduce the Large Knowledge Collider as a platform solution that leverages the service-oriented approach to implement a new reasoning technique, capable of dealing with exploding volumes of the rapidly growing data universe, in order to be able to take advantages of the large-scale and on-demand elastic infrastructures such as high performance computing or cloud technology.

Keywords: Semantic web · Incomplete reasoning · LarKC · Service architecture

1 Introduction

The large- and internet-scale data applications are the primary challenger for the modern Semantic Web, and in particular for reasoning algorithms, used for processing exploding volumes of data exposed currently on the Web. Reasoning is the process of making implicit logical inferences from the explicit set of facts or statements, which constitute the core of any knowledge base. The key problem for most of the modern reasoning engines such as Jena or Pellet is that they can not efficiently be applied for the real-life data sets that consist of tens, sometimes of hundreds of billions of triples (a unit of the semantically annotated information), which can correspond to several petabytes of digital information on the disc. Whereas modern advances in the Supercomputing domain allow this limitation to be overcome, the reasoning algorithms and logic need to be adapted to the demands of the rapidly growing data universe, in order to be able to take advantages of the large-scale and on-demand computing infrastructures. On the other hand, the algorithmic principals of the reasoning engines need to be reconsidered as well in order to allow for the specific of very large volumes of data (e.g. inconsistent and noisy data). Service-oriented architectures (SOA) can greatly contribute to this goal,

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 487–489, 2015.
DOI: 10.1007/978-3-662-46641-4

acting as the main enabler of the newly proposed reasoning techniques such as incomplete reasoning.

2 The Large Knowledge Collider

One of the most prominent efforts to facilitate the development of trend-new applications for large-scale reasoning has been the EU-funded project of the Large Knowledge Collider (LarKC) [1]. The mission of the project was to set up a distributed reasoning infrastructure for the Semantic Web community, which should enable the application of reasoning to scale far beyond the currently recognized limitations, by implementing the incomplete reasoning approach. The current and future Web applications that deal with "big data" are in focus of LarKC.

To realize this mission, LarKC has created an infrastructure that allows construction of plug-in-based reasoning applications, following the incomplete reasoning approach, facilitated by incorporating interdisciplinary techniques such as inductive, deductive, interleaved reasoning, in combination with the methods from other knowledge representation domains such as information retrieval, machine learning, cognitive and social psychology, etc. The core of the infrastructure is a platform – a software framework that facilitates design, testing, and exploitation of new reasoning techniques for development of large-scale applications. The platform does this by providing solutions for creating very lightweight, portable and unified services for data sharing, accessing, transformation, aggregation, and inferencing, as well as building Semantic Web applications on top of those services. The efficiency of the LarKC services is ensured by providing a transparent access to the underlying resource layer, served by the platform, involving elastic high performance computing, storage, and cloud resources, and in the other way around, providing performance analysis and monitoring information back to the user. The platform is built in a distributed, modular, and open source fashion. Moreover, the platform offers means for building and running applications across the plug-ins, provide them a persistent data layer for storing data, facilitate parallel execution of large-scale data operations by leveraging the distributed and high-performance resources.

Guided by the preliminary goal to facilitate the incomplete reasoning, LarKC has evolved in a unique platform, which can be used for the development of robust, flexible, and efficient semantic web applications, also leveraging the modern grid and cloud resources. Our poster presents the LarKC approach to developing the service-oriented reasoning applications. We demonstrate some of the most successful applications developed on the LarKC foundation such as Bottari – the Semantic Challenge winner in 2011, or WebPIE – the Billion Triple Challenge winner in 2010. We show that LarKC can be applied to solve many of the current Semantic Web challenges, and would like to serve a discussion forum on its adoption in the external development communities.

Reference

1. Assel, M., Cheptsov, A., Gallizo, G., Celino, I., Dell'Aglio, D., Bradeško, l., Witbrock, M., Della Valle, E.: Large knowledge collider: a service-oriented platform for large-scale semantic reasoning. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'2011) (2011)

Named Entity Disambiguation Using Linked Data

Danica Damljanovic and Kalina Bontcheva(✉)

Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, UK
{d.damljanovic,k.bontcheva}@dcs.shef.ac.uk

1 Introduction

Identification of Named Entities (NE) such as people, organisations and locations is fundamental to semantic annotation and is the starting point of more advanced text mining algorithms. For instance, sentiment analysis is widely used in finance to extract the latest signals and events from news that could affect stock prices. However, before extracting company-related sentiment, it is necessary to identify the documents containing the corresponding and *unambiguous* company entities. Humans usually resolve ambiguities based on context. We argue that Linked Data can be a valuable source for extending the already available context. We combine a state-of-the-art named entity tool with novel Linked Data-based similarity measures and show that our algorithm can improve disambiguation accuracy on a subset of Wikipedia user profiles.

2 Entity Linking and Disambiguation Algorithm

The goal of the algorithm is to identify named entities in text and attach the correct DBpedia URI to each one of them. For the former, we use the ANNIE Information Extraction system from GATE [1]. It combines some small lists of names (e.g. days of the week, months) and rule-based grammars, to processes text and produce NE types such as *Organization*, *Location* and *Person*. ANNIE also resolves coreference so that entities with the same meaning are linked. For example, *General Motors* and *GM* would be identified as referring to the same entity.

GATE's ontology-based gazetteer, namely the Large Knowledge Gazetteer (LKB), is used for entity linking. LKB performs lookup and assigns URIs to words/phrases in the text. For the purpose of our application, we match only against the values of the *rdf:label* and *foaf:name* properties, for all instances of the *dbpedia-ont:Person*, *dbpedia-ont:Organisation* and *dbpedia-ont:Place* classes.

Both ANNIE and LKB can be used independently, however, while NE types generated by ANNIE miss the URI which is necessary to disambiguate them, LKB does not use any context, which results in generating many spurious entities. For example, each letter *B* is annotated as a possible mention of *dbpedia:B_%28Los_Angeles_Railway%29*, which refers to a line called *B* operated by Los

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 490–492, 2015.
DOI: 10.1007/978-3-662-46641-4

Angeles Railway. We next describe the algorithm which filters out such noise, by consolidating the output of ANNIE and LKB, followed by a disambiguation step. A high-level pseudo code looks as follows:

```
1. Identify NEs (Location, Organisation and Person) using ANNIE
2. For each NE add URIs of matching instances from DBpedia
3. For each ambiguous NE calculate disambiguation scores
4. Remove all matches except the highest scoring one
```

The disambiguation algorithm uses context in which the particular entity appears and a weighted sum of the following three similarity metrics:

- *String similarity*: refers to the Levenshtein distance between the text string (such as *Paris*), and the labels describing the entity URIs (for example, *Paris Hilton, Paris* and *Paris, Ontario*).
- *Structural similarity* is calculated based on whether the ambiguous NE has a relation with any other NE from the same sentence or document. For example, if the document mentions both *Paris* and *France*, then structural similarity indicates that *Paris* refers to the capital of France. All other entity URIs can be disregarded, based on the existing relationship between *dbpedia:Paris* and *dbpedia:France*.
- *Contextual similarity* is calculated based on the probability that two words have a similar meaning as in a large corpus (DBpedia abstracts in our case) they appear with a similar set of other words. To implement that we use the Random Indexing method [2] and calculate similarity using the cosine function.

3 Experiments

We manually labelled the corpus with 100 Wikipedia user profiles to create a gold standard against which we can evaluate performance, using precision, recall and f-measure. Table 1 summarises the results. The addition of ANNIE to the DBpedia lookup using LKB improved precision at a slight cost in recall. Adding the disambiguation layer results in further improved precision with slightly lower recall. However this results in an overall improvement in f-measure, which rises to 0.82. This shows that our disambiguation algorithm which exploits Linked Data as an additional knowledge source, eliminated a large number of incorrect annotations.

Table 1. Precision, recall and f-measure of the different algorithms

	Precision	Recall	F-measure
LKB	0.03	0.86	0.05
LKB+ANNIE	0.14	0.81	0.24
LKB+ANNIE+Disambiguation	0.84	0.80	0.82

Acknowledgements. This research has been supported by the EU-funded FP7 TrendMiner[1] project.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011). http://tinyurl.com/gatebook
2. Sahlgren, M.: An introduction to random indexing. In: Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005. Copenhagen, Denmark, August 2005. http://www.sics.se/∼mange/papers/RI_intro.pdf

[1] http://www.trendminer-project.eu/

Sharing and Analyzing Remote Sensing Observation Data for Linked Science

Tomi Kauppinen[1](✉),
Benedikt Gräler[1], and Giovana Mira de Espindola[2]

[1] Institute for Geoinformatics, University of Münster, Münster, Germany
[2] Earth System Science Center, National Institute for Space Research (INPE), São José dos Campos, Brazil

Abstract. In order to make research settings transparent and reproducible there is a need for publishing both data and methods behind the research. In this paper our contribution is to show how large amounts of remote sensing observation data about the Brazilian Amazon Rainforest have been published as Linked Spatiotemporal Data. Moreover, we show how this data can be further accessed and analyzed using R statistical computing environment by openly available methods. This all is a contribution towards Linked Science, where not just publications, but data, methods, tools, and other scientific assets are interconnected and shared online.

1 Opening and Linking Science

Open Science needs Open Data to maximize the transparency, reproducibility and reuse of scientific efforts. An example of a high demand for data is the research about climate change, for example about the role of deforestation in it.

Deforestation and its related phenomena such as market prices of agricultural products form together a complex system. There is an urgent need to share and publish research data about it, as it would enable other researchers to interconnect their data to the published ones. The benefit is that these explicit interconnections allow for the analysis of all of the resulting linked data in a transdisciplinary manner. Thus the whole complex socio-economic and environmental system could be modelled and not just subsets of it.

Opening up of scientific assets like data and methods behind scientific settings, and interconnecting them is called Linked Science [1]. One crucial aspect of Linked Science is how to access and analyse data, and especially how to get only that part of data which is of interest for a given research question. Linked Data solves the access part, and SPARQL allows to query only a subset of the data. For statistical computing there are tools like R[1], and a separate package[2] for it supports querying Linked Data.

[1] http://www.r-project.org

[2] http://linkedscience.org/tools/sparql-package-for-r/

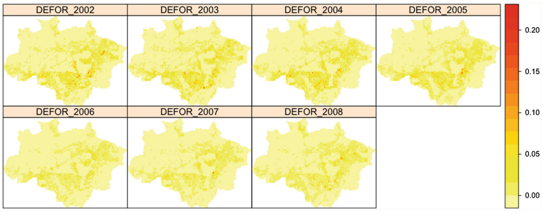
E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 493–494, 2015.
DOI: 10.1007/978-3-662-46641-4

2 Case for Linked Science: Sharing and Analyzing Observation Data

For analyzing processes and operations of complex systems such as environmental and societal systems there is a need to have (1) well interconnected data about a system and (2) techniques for statistical computing and for other types of reasoning to find new information, and means to (3) explore and visualize this information.

Our contribution is the Linked Brazilian Amazon Rainforest Data [2] which is openly available for anyone to use for non-commercial research. The data can be accessed in a Linked Data- fashion via a SPARQL-endpoint, and via dereferenciable URIs. The data consists of 8250 cells—each of size of 25 km × 25 km—capturing the observations of deforestation in the Brazilian Amazon Rainforest and a number of related and relevant socio-economic indicators. In summary, our contribution was to show

- how to access Linked Spatiotemporal Data about the complex environmental system of the Brazilian Amazon Rainforest, including deforestation statistics, and a variety of socio-economic and environmental indicators
- how to store the spatiotemporal data in a meaningful way as Linked Data
- how to plot[3] (see below an example) and handle the data within R.



Acknowledgements. This research has been partially funded by the International Research Training Group Semantic Integration of Geospatial Information (DFG GRK 1498).

References

1. Kauppinen, T., de Espindola, G.M.: Linked open science—communicating, sharing and evaluating data, methods and results for executable papers. In: Proceedings of the International Conference on Computational Science (ICCS 2011) (2011)
2. Kauppinen, T., de Espindola, G.M., Jones, J., Sanchez, A., Gräler, B., Bartoschek, T.: Linked Brazilian Amazon Rainforest Data. Semant. Web J. **5**(2) (2014)

[3] http://linkedscience.org/tools/sparql-package-for-r/tutorial-on-sparql-package-for-r/

ANISE: An ANatomIc SEmantic Visualizer

Luis Landaeta(✉), Alexander Baranya, Alexandra La Cruz(✉),
and Maria-Esther Vidal(✉)

Universidad Simón Bolívar, Caracas, Venezuela
{llandaeta, abaranya, alacruz, mvidal}@ldc.usb.ve

Abstract. ANISE exploits knowledge encoded in controlled vocabularies to precisely visualize 3D Medical images. ANISE receives 3D images annotated with existing medical ontologies and performs reasoning tasks to improve effectiveness of organs and tissues visualization. Data of a Computed Tomography Head is used to show the benefits of considering semantic annotations and the precision achieved by a visualizer when these annotations are used during volume rendering.

1 Our Approach

A transfer function (TF) maps density values from a voxel in a volumetric data into optical properties, e.g., opacity and color. 2D images are generated from TFs and these images are particular for the executed volume rendering technique. During rendering, the opacity property hides or visualizes voxels behind, while using color and opacity properties together, may help to distinguish different tissues belonging to different anatomic organs into the body. Specifying a TF is not an easy task on medical volumetric images, and normally a segmentation technique needs to be applied in order to separate different anatomical organs in an image. Traditionally, TFs are based on existing characterizations of the organs that relate medical image modality, e.g., Computed Tomography, Ultra Sound, a tissue in a organ, and a density range [2]. However, some tissues belonging to different organs may have overlapped densities. Thus, considering only density values is not enough to produce a precise tissue classification, and segmentation processes are required. The problem of semantically annotating volumetric data has gained attention in the literature and applications of the annotations have been illustrated [1]; however, nothing is said about the benefits of using annotation semantics during TF definition or data visualization.

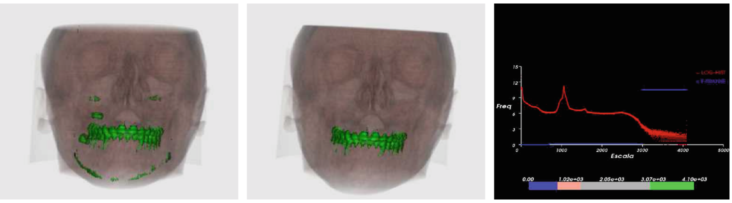
We present ANISE, a semantic visualizer which relies on a new strategy for specifying TFs. ANISE TFs are based on pre-elaborated semantic annotations of volumetric data, which are validated against existing medical ontologies, e.g., RadLex[1], FMA [3]. ANISE is comprised of a reasoner which infers the bounding box that contains the organs or tissues of a given organ or sub-volume area as well as the organ main properties, e.g., density and opacity Also, knowledge encoded in the

Part of this work has been funded by FONACIT.

[1] http://www.rsna.org/radlex/

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 495–497, 2015.
DOI: 10.1007/978-3-662-46641-4

ontologies is used to infer the location of an organ and the organs that should be around its; thus, voxels that are not part of the organ of interest can be eliminated during the classification process. Further, semantic annotations can be used for locating different tissues and providing a precise visualization of medical data. We study the quality of our proposed approach in a volume data sample of a Computed Tomography (CT) Head, which consists of 3D data of 184 × 256 × 170 voxels; Fig. 1(a) illustrates the rendering of the image when only densities are used by the TF; note that different tissues are colored with green. However, when semantic annotations are used in conjunction with knowledge encoded in the FMA and RadLex ontologies, ANISE can determine that only the teeth should be colored green; this is done just using the same TF (Fig. 1(c)) but performing a reasoning task that allows to detect the voxels that semantically do not correspond to the tooth tissue and that should not be included in the final volume rendering (see Fig. 1(b)).



(a) CT Head Rendering without Annotations.

(b) CT Head Rendering with Semantic Annotations.

(c) A Transfer Function.

Fig. 1. CT head images and transfer function

2 Conclusion

We present ANISE which is able to exploit knowledge encoded in ontologies used to annotate 3D medical images, and enhance the rendering process of the images. Quality of ANISE renderings have been studied in different images, and we have observed that they can allow accurate location of organs that comprise a medical image. In the future we plan to define specialized visualizers able to identify anomalies in the images to be rendered.

References

1. Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in CT volumes. In: MICCAI Workshop on Probabilistic Models for Medical Image Analysis (MICCAI-PMMIA) (2009)

2. Prei, B., Bartz, D.: Visualization in Medicine: Theory, Algorithms, and Applications. The Morgan Kaufmann Series in Computer Graphics, San Francisco (2007)
3. Rosse, C., Mejino, J.: The foundational model of anatomy ontology. In: Burger, A., Davidson, D., Baldock, R. (eds.) Anatomy Ontologies for Bioinformatics, vol. 6, pp. 59–117. Springer, Heidelberg (2008)

RESTdesc—A Functionality-Centered Approach to Semantic Service Description and Composition

Ruben Verborgh[1](✉), Thomas Steiner[2], Davy Van Deursen[1],
Sam Coppens[1], Erik Mannens[1],
Rik Van de Walle[1], and Joaquim Gabarro[2]

[1] Ghent University – IBBT, ELIS – Multimedia Lab,
Gaston Crommenlaan 8 bus 201, 9050 Ledeberg-Ghent, Belgium
{ruben.verborgh, rik.vandewalle}@ugent.be
[2] Department LSI, Universitat Politécnica de Catalunya,
08034 Barcelona, Spain
{tsteiner, gabarro}@lsi.upc.edu

Abstract. If we want automated agents to consume the Web, they need to understand what a certain service does and how it relates to other services and data. The shortcoming of existing service description paradigms is their focus on technical aspects instead of the functional aspect—what task does a service perform, and is this a match for my needs? This paper summarizes our recent work on RESTdesc, a semantic service description approach that centers on functionality. It has a solid foundation in logics, which enables advanced service matching and composition, while providing elegant and concise descriptions, responding to the demands of automated clients on the future Web of Agents.

1 Where are the Agents?

When asked as researchers to explain what the Semantic Web is about, how do we respond? Many would refer to the initial vision of automated agents browsing the Web and executing tasks for us. But exactly how far are we today? Ten years have passed since the famous *Scientific American* article [3], yet co-author James Hendler questioned at ESWC 2011 why, while all infrastructure is in place nowadays, the agents are still missing.

What makes the Web so difficult for machines? So far, we have only seen successful clients for *specific* purposes, mostly tailored to the API of a *certain* site. This contrasts with human behavior: we surf for several *different* purposes on a *variety* of websites. The discrepancy originates in two related aspects: semantics and hyperlinks. The Resource Description Framework (RDF) and the Linked Data effort help to overcome the problem of *data* semantics by providing machine-interpretable data with linked concepts. On the other hand, *services* tend not to provide semantics or links, although these are vital for the Web.

This paper summarizes our ongoing research on RESTdesc [4], a semantic service description method based on hyperlinks. We believe that the connection of service

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 498–500, 2015.
DOI: 10.1007/978-3-662-46641-4

descriptions and hyperlinks can play a key role in a solution towards making the Web more accessible to automated agents.

2 RESTdesc Explains a Service's Functionality to Agents

RESTdesc is both a description and a discovery method targeting RESTful Web services, with an explicit focus on functionality. It consists of well-established technologies such as HTTP [3] and RDF/Notation3 [1] and is built upon the concepts of hyperlinks and Linked Data. Its goal is to complement the Linked Data vision, which focuses on *static* data, with an extension towards Web services that focus on *dynamic* data. All RESTdesc descriptions are:

- *self-describing*: using Notation3 semantics;
- *functional*: explaining exactly what the operation does;
- *simple*: descriptions are expressed directly using domain vocabularies.

Since RESTdesc entails the operational semantics of Notation3, it allows for versatile discovery methods. We can indeed use the power of Notation3 reasoners to determine whether a service satisfies at set of conditions. Even more advanced reasoning is possible to decide on service matching, and to create complex compositions of different services [4]. We see this as an important prerequisite for services in order for them to contribute to the future Web of Agents, since *new* functionality can only be obtained by on-demand compositions tailored to a specific problem.

3 Conclusion and Future Work

We firmly believe that RESTdesc has a strong potential in the field of service description, automatic discovery, and consumption. Based on RESTful principles, it targets modern, resource-oriented websites and focuses on the resources and their functional relationships instead of technical properties.

Future work includes the application of RESTdesc technologies to different fields and applications, and implementing generic, automated RESTdesc agents. We plan to provide a public implementation of the reasoning framework for use as a black box, so intelligent agents can employ RESTdesc composition techniques transparently. Another interesting area is the collaboration and integration of different services, for example using ontology matching.

Visit the project website http://restdesc.org/ for updates on our work.

References

1. Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax. W3C Team Submission (Mar 2011). http://www.w3.org/TeamSubmission/n3/
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Sci. Am. **284**(5), 34–43 (2001)
3. Fielding, R.T., Taylor, R.N.: Principled design of the modern Web architecture. ACM Trans. Internet Technol. **2**(2), 115–150 (2002)
4. Verborgh, R., Steiner, T., Van Deursen, D., De Roo, J., Van de Walle, R., Gabarró Vallés, J.: Capturing the functionality of Web services with functional descriptions. Multimedia Tools and Appl. **64**(2), 365–387 (2013). http://dx.doi.org/10.1007/s11042-012-1004-55

Bringing the Web of Data to Developing Countries: Linked Market Data in the Sahel

Victor de Boer[1](✉), Nana Baah Gyan[1], Pieter De Leenheer[1], Anna Bon[2], Chris van Aart[1], Christophe Guéret[1], W. Tuyp[2], Stephane Boyera[3], Mary Allen[4], and Hans Akkermans[1]

[1] Computer Science Department, The Network Institute, VU University Amsterdam, Amsterdam, The Netherlands
{v.de.boer,n.b.gyan,pieter.de.leenheer,c.j.van.aart, c.d.m.gueret,j.m.akkermans}@vu.nl

[2] Centre for International Cooperation, VU University Amsterdam, Amsterdam, The Netherlands
a.bon@vu.nl, wam.tuyp@cis.vu.nl

[3] World Wide Web Foundation, London, UK
stephane@boyera.net

[4] Sahel Eco, Bamako, Mali
mary.saheleco@afribonemali.net

1 Linked Market Data and the RadioMarché System

Although the Web is a great success, around 4.5 billion people -mainly in developing countries- are still unable to access its information. Currently, a number of efforts are being undertaken to bridge this so-called ‘digital divide’ in the Web of Documents. At the same time, as engineers of the Web of Data, we have the opportunity to not let the “Digital Linked Data Divide” grow too large. Like it does in the developed world, sharing and re-use of locally produced and consumed data can also increase its value in developing regions. We here describe our ongoing efforts to implement Linked Data-backed solutions for the rural Sahel regions.

These efforts center around, RadioMarché, a web-based market information system aimed at stimulating agricultural trade in the Sahel region. Its market data is accessible for local farmers through voice-based interface in local languages using first generation mobile phones. The data from regionally distributed instances of RadioMarché, can be aggregated and exposed using Linked Data approaches, so that new opportunities for product and service innovation in agriculture and other domains can be unleashed.

An instance of RadioMarché has one data store with market information such as product offerings (including product type, quality, quantity, location and logistical issues) and contact details from sellers and buyers. To maximize the reusability across different domains and regions and allow for automatic machine processing, we adopt Linked Data standards to represent the data. Linked Data fits our purposes well since it provides a particularly light-weight way to share, re-use and integrate various data sets and does not require the definition of a specific database schema for a dataset. Our implementation methodology assumes that we start from a legacy Market Information System and Linked Data provides us with a way of integrating the data across multiple

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 501–503, 2015.
DOI: 10.1007/978-3-662-46641-4

regional instances of RadioMarché or reuse the data for completely new services, both within a region and across regions. Additionally, Linked Data is well-suited to deal with multiple languages as its core concepts are resources rather than textual terms.

Other than Linked Data-access to the data (through RDF request or SPARQL), Radiomarché provides multiple user interfaces to the data. Through the traditional Web channels or via e-mail, users can get weekly digests of the latest offerings or add their own information. The innovative *voice-based* interface allows non-intrusive market information access for all users having a first-generation mobile phone. It allows local farmers to navigate a voice-based menu and enter product offerings using a call-in service at a local telephone number. The voice service is available in the local languages relevant to the specific region. For the voice-based interface, we use prerecorded phrases in local languages and dialects for a slot-and-filler text-to-speech system.

2 Current Status

An instance of Radiomarché has been deployed in the Tominian area in Mali, Africa. Here, it augments a legacy MIS focused on non-timber forest products (NTFP's) such as honey and nuts set up by the project partner Non-governmental organisation (NGO), Sahel Eco. The system has been running since November 2011 and many product offerings have been added to the data store. A voice-interface is defined for the local French dialect as well as the Bambara language spoken in the region. Currently, the voice- and web-interfaces are designed for use by local radio station operators, who serve as middle men in delivering the market data to people in the region. We are currently elaborating the interface to allow access for the individual farmers themselves. We are gathering user feedback to validate the system and inform a next iteration of the system design.

The data is exposed as Linked Data using an instance of the Cliopatria semantic server[1], accessible at http://eculture.cs.vu.nl:1979/radiomarche/[2]. Links to external sources such as DBPedia and GeoNames are established. We are investigating opportunities to host this data on low-powered hardware in the region itself.

3 Linked Data Use Cases

We are defining use cases and building applications that benefit from the sharing and re-use of the Linked Market Data. One additional service being developed now is a meeting scheduling system which provides local NGOs with a more effective way to transfer agricultural knowledge about NTFP's to their farmer communities. By integrating this information with the market information, personal profiles can be

[1] http://cliopatria.swi-prolog.org

[2] An example URI for a single offering is http://purl.org/collections/w4ra/radiomarche/offering_54

enriched with information about the type of products that specific farmers have been producing.

A second use case that is currently under development is a voice-based journalism platform (named Furoba Blon), which allows both professional and citizen journalists to send voice-recorded news items to local community radios. The target region for this use case consists of agricultural communities providing opportunities for re-use of both technical infrastructure as well as data.

We are currently developing services that benefit from aggregated market information across regions. By linking the market information to DBPedia, agricultural vocabularies and geographical thesauri, local and national governments as well as NGOs can exploit the aggregated market information for analytic purposes, monitoring the trade in NTFPs within and across regions.

Leveraging the Emergent Semantics in Twitter Lists for Ontology Development

Andrés García-Silva(✉) and Oscar Corcho

Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain
{hgarcia, ocorcho}@fi.upm.es

Abstract. In this poster we present an approach to obtain automatically domain ontologies relying on domain vocabularies elicited from Twitter lists and on the reuse of conceptualizations in existing knowledge bases. We tap into relations established between list names, under which prominent users in the domain of study as well as in the microblogging platform have been classified, to harvest related concepts. The relations between concepts are identified from interlinked general-purpose knowledge bases.

1 Introduction

Twitter, the microblogging platform, enables users to organize others into lists. Other users can benefit of existing lists by subscribing to them so that they can receive updates of the people classified under these lists. Given the size of the social network which nowadays reaches 100 million active users, and the bottom-up classification structure emerging from the connection between list names, curators, subscribers, and members, these lists potentially constitutes a valuable resource for knowledge acquisition. In Table 1 we presents the terms found in list names under which a journalist have been listed. Note that most of the terms are semantically related around the news domain. In fact, in [1] we have shown that list names are semantically related according to co-occurrence patterns which have been defined in terms of the use given by curators, members and subscribers.

Table 1. Terms found in list names with the corresponding frequency of appearance.

news	297	politics	208	media	58	new_politics	36	celebrities	34
celebs	28	political	26	journalists	25	twibes	18	national	17

Therefore, we aim at collecting a vocabulary, relevant in the domain of study, from the classification structure emerging from Twitter lists. We reuse conceptualizations in existing knowledge bases so that we can define the semantics of the relations between the terms in the vocabulary. With the lists of terms and relations we create an ontology schema which we may also populate with instances from the knowledge bases.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 504–506, 2015.
DOI: 10.1007/978-3-662-46641-4

2 Approach

2.1 Preprocessing

During this activity we extract, normalize, and transform the lists. For data *extraction* we rely on the REST services provided by the platform. Next, during the *normalization* task we obtain a standardized version, according to a lexical resource, of the terms contained in list names. Finally, we *transform* the Twitter list data into a one-mode graph where nodes are the members of the lists, and there exists a weighted edge between two users if they were classified under a list containing a shared term. The weight of each edge corresponds to the number of shared lists.

2.2 Collecting the Domain Vocabulary

We traverse the graph starting from some initial users which are prominent in the domain. We propose to use external resources and Twitter information to identify prominent users. External resources can be domain experts or for instance bibliographic resources. Prominent users in real life may not be important in Twitter, so we measure their influence index (e.g., using the klout.com service). The intuition is that around prominent users is more feasible that a vocabulary has emerged since more people are interested in what they are saying. Then we traverse the graph starting from each prominent user and compare the users, that we are reaching while traversing, using the terms under which they have been listed with the terms related to the starting user. The terms of the most similar users are added to the domain vocabulary.

2.3 Eliciting the Vocabulary Semantics

Finally, we use knowledge bases to elicit the semantics of the terms found in the previous activity. We propose to use linked data sets so that we benefit from the interlinked conceptualizations. We associate terms with semantic entities which in turn are used to obtain classes. Next we search for relations between the classes using SPARQL queries. We include relations set up through intermediate semantic entities which can be also included in the final ontology. In addition the ontology can be populated with existing instances of the classes reused in the process. Challenges in this activity include ambiguity of terms and knowledge base heterogeneity. The output is an ontology consisting of classes, relations and instances.

Reference

1. García-Silva, A., Kang, J.H., Lerman, K., Corcho, O.: Characterising emergent semantics in Twitter lists. In: 9th extended Semantic Web Conference, Crete (2012)

Full-Text Support for Publish/Subscribe Ontology Systems

Lefteris Zervakis[1], Christos Tryfonopoulos[1(✉)],
Antonios Papadakis-Pesaresi[2], Manolis Koubarakis[2],
and Spiros Skiadopoulos[1]

[1] Department of Computer Science and Technology,
University of Peloponnese, Sparti, Greece
trifon@uop.gr
[2] Department of Informatics and Telecommunications,
University of Athens, Athens, Greece

Abstract. We envision a publish/subscribe ontology system that is able to index millions of user subscriptions and filter them against ontology data that arrive in a streaming fashion. In this work, we propose a *SPARQL extension* appropriate for a publish/subscribe setting; our extension builds on the natural semantic graph matching of the language and supports the creation of *full-text subscriptions*. Subsequently, we propose a main-memory *subscription indexing algorithm* which performs both semantic and full-text matching at low complexity and minimal filtering time. Thus, when ontology data are published matching subscriptions are identified and notifications are forwarded to users.

System Overview

Resource Description Framework (RDF) constitutes a conceptual model and a formal language for representing resources in the Semantic Web. It is also the data format of choice for modern *publish-subscribe* ontology systems, which demand sophisticated data representation and efficient filtering mechanisms to match massive ontology data against millions of *user subscriptions* (also referred to as *continuous queries*). The SPARQL query language is currently the W3C recommendation for querying RDF data and the Semantic Web. The graph model over which it operates naturally joins data together and represents a fully-fledged language; however, it lacks the support of a complete *full-text retrieval* mechanism, beyond existing regular expression support, with sophisticated algorithms and data structures to minimise processing and memory requirements.

In this work, we focus on full-text filtering of ontology data that contain RDF literals in their property elements. To preserve the expressivity of SPARQL, we view the full text operations as an additional *filter* of the subscription variables. In this context, we define a new binary operator *ftcontains* that takes a variable of the subscription and a full-text expression that operates on the values of this variable as parameters. An example of a SPARQL subscription with full-text support is shown below.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 507–509, 2015.
DOI: 10.1007/978-3-662-46641-4

```
SELECT ?article
WHERE  {?publisher rdf : type      Publisher.
        ?publisher publishes       ?article.
        ?article   articleText     ?articleText.
FILTER  ftcontains (?articleText, "economic" ftand "crisis")}
```

We focus on RDF triples where the *subject* is always a node element and the *predicate* denotes the subject's relation to the *object*, which is a literal expressed as a typed or untyped string. A full text expression is evaluated only against a literal; thus the variable of the subscription can only be the object of a triple pattern. The expressions supported involve the usual *Boolean operators* (denoted by *ftand*, *ftor*, etc.), as well as *proximity* and *phrase* matching. Below we present an example of a full-text SPARQL subscription that will match all *rdf:type Article* node elements, with a property named *title* containing a string literal with the keywords *"economic"* and *"crisis"*.

To perform the semantic matching, we define a Semantic Match Table in the spirit of [1], where a two-level hash table is used to represent the series of joins in a SPARQL subscription as a connected chain. We extend this idea to provide a hashing scheme that is able to accommodate all possible types of triple patterns in SPARQL subscriptions. Additionally, to support the full-text features introduced in the SPARQL subscriptions, we utilise a *property hash table* that uses as key the constant part of the triple pattern in the SPARQL subscription. This hash table provides access to a data structure, which comprises of (i) *tries* storing the keywords contained in the full-text

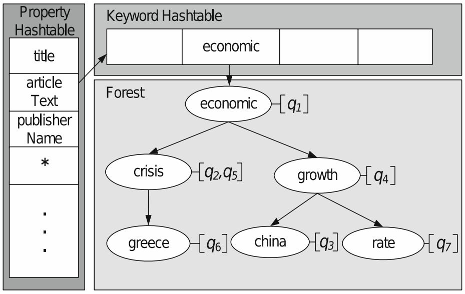


Fig. 1. Subscription indexing scheme

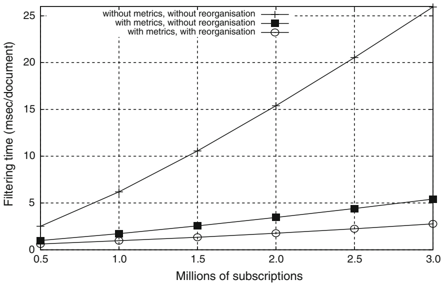


Fig. 2. Filtering time/document (msecs)

part of subscriptions and (ii) a *keyword hash table* that allows fast access to the trie roots. Figure 1 shows these data structures for a set of seven user subscriptions.

User subscriptions are organised into tries extending the approach of [2] to rely on *common subsets* of subscriptions. The main idea behind the indexing algorithm is to use tries to capture common elements of subscriptions. To do so, we utilise *metrics* to locate the best possible indexing position in the forest of tries. Since our algorithm is influenced by the order of insertion of subscriptions (due to greedy subscription indexing), a statistics-based *subscription reorganisation* is employed. In the reorganisation phase of the algorithm, a scoring mechanism is utilised to modify the order of subscription indexing for all subscriptions inserted since the last reorganisation of the forest. In our evaluation we used 3.1 M extended abstracts downloaded from DBpedia as incoming RDF documents and artificially generated subscription databases of varying sizes. Figure 2 shows the filtering time when (i) no metrics for the best indexing position in the forest are employed (deterministic subscription indexing), (ii) metrics are employed, but no re-organisation is used, and (iii) both metrics and reorganisation are employed.

References

1. Park, M.J., Chung, C.W.: iBroker: An intelligent broker for ontology based publish/subscribe systems. In: ICDE (2009)
2. Tryfonopoulos, C., Koubarakis, M., Drougas, Y.: Information filtering and query indexing for an information retrieval model. In: ACM TOIS (2009)

Distributed Stream Reasoning

Rehab Albeladi(✉), Kirk Martinez, and Nicholas Gibbins

Electronics and Computer Science,
University of Southampton, Southampton, UK
{raab1g09, km, nmg}@ecs.soton.ac.uk

Abstract. Stream Reasoning is the combination of reasoning techniques with data streams. In this paper, we present our approach to enable rule-based reasoning on semantic data streams in a distributed manner.

Data streams are being continually generated in diverse application domains such as traffic monitoring, smart buildings, and so on. Continuous processing of such data has been intensively investigated in the database community, where a special class of management systems [1,2] has been introduced to perform on-the-fly processing of data streams. However, these data streams lack standard formats, which make interoperability a real challenge. On the other hand, Semantic Web data has well-defined meanings and a number of semantic formats have been standardised. Semantic reasoners have been developed that can perform complex reasoning tasks on this data. Nevertheless, reasoning upon streaming data has received far less attention than reasoning upon static data. Stream Reasoning is the area that aims to combine reasoning techniques with data streams [3].

Research in this area mainly focuses on extending SPARQL to process RDF streams. We focus more on the infrastructure of the reasoning process. Our approach enables reasoning in a continuous manner using low level operators; this approach differs from that in [5], in which each query is split into a static and a dynamic part and the dynamic part is passed to a DSMS system. In the implementation of our stream reasoner, we have focussed on the two main issues of reasoning and distribution.

Stream Reasoning. To enable the rule-based reasoning process, we use the RETE algorithm [5] for pattern matching. Rules are translated into RETE networks of nodes. The nodes represent different operators and the data flows between these nodes. The tree-like network divides the matching process into multiple steps that perform different checks, so if a data element does not match the first node, it is simply discarded and does not complete its way through the network. A typical RETE network has two types of node: filter (or alpha) nodes, and join (or beta) nodes. A filter node is similar to the select operation in query languages; it only propagates statements that match its condition. On the other hand, a join node is responsible for joining some data elements of its two input streams depending on a specified condition. Each join node manages a time-based or tuple-based sliding window on each stream input.

A prototype RDFS reasoner for RDF data streams has been fully implemented, combining features from both reasoning techniques and stream processing techniques.

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 510–511, 2015.
DOI: 10.1007/978-3-662-46641-4

It performs the inference task as a rule engine using a Rete network, while the implemented Rete network performs some DSMS operations, such as converting streams into relations by using the sliding window technique. The system is fed by RDF streams, it matches them against the RDFS entailment rules, and produces new sets of data in a continuous manner. In our initial evaluation, we have been able to demonstrate the trade-off between completeness and execution time by varying window sizes.

Distribution. For efficient processing of large volume data, scalability is a major concern. Distributed processing of data streams enables more scalable and fault-tolerant systems, so we distribute our reasoning networks using the eXtensible Messaging and Presence Protocol (XMPP). We chose XMPP for its push based distribution style which satisfies the real-time requirement of streaming applications with minimal latency, and for its ease of integration with Web technologies.

In order to minimise network traffic, we use a graph-based definition for the RDF stream data type. Instead of triples, we define the RDF stream as an ordered sequence of RDF graphs associated with a time element. This brings other advantages besides being more efficient in terms of transportation; some queries can be evaluated by viewing the graph as a single record, and data provenance can be tracked more easily at a graph level.

We have built a prototype system that can process RDF data streams using distributed RETE networks, in which nodes are distributed in multiple machines and can communicate with each other using the XMPP in a publish/subscribe pattern, and are now working on combining this system with our previous reasoner to perform continuous reasoning on streaming RDF data in a distributed manner.

References

1. Abadi, D., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S. Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: A new model and architecture for data stream management. VLDB J., **12**(2), 120–139 (2003)
2. Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Motwani, R., Nishizawa, I., Srivastava, U., Thomas, D., Varma, R., Widom, J.: STREAM: the Stanford stream data manager. IEEE Data Eng. Bull. **26**(1), 19–26 (2003)
3. Della Valle, E., Ceri, S., Barbieri, D.F., Braga, D., Campi, A.: A first step towards stream reasoning. In: Domingue, J., Fensel, D., Traverso, P. (eds.) FIS 2008. LNCS, vol. 5468, pp. 72–81. Springer, Heidelberg (2009)
4. Barbieri, D., Braga, D., Ceri, S., Grossniklaus, M.: An execution environment for C-SPARQL queries. In: 13th International Conference on Extending Database Technology. ACM, Lausanne (2010)
5. Forgy, C.: Rete: a fast algorithm for the many pattern/many object pattern match problem. Artif. Intell. **19**, 17–37 (1982)

ParkJam: Crowdsourcing Parking Availability Information with Linked Data (Poster)

Jacek Kopecký(✉) and John Domingue

Knowledge Media Institute,
The Open University, Milton Keynes, Buckinghamshire, UK
{j.kopecky,j.b.domingue}@open.ac.uk

This poster shows a mobile Android app that uses openly available geographic data and crowdsources parking availability information, in order to let its users conveniently find parking when coming to work or driving into town. The application builds on Linked Data, and publishes the crowdsourced parking availability data openly as well. Further, it integrates additional related data sources, such as events and services, to provide rich value-adding features that will act as an incentive for users to adopt the app.

Motivation

Managing parking in congested areas is a well-recognized problem. In the modern car-oriented world, many will experience difficulties finding parking places when driving to work or into a congested city. In [1], Shoup discusses the effects of free parking, and suggests that parking spaces should be dynamically priced at a level that would result in about 85% utilization, with many benefits beside the improved availability. He acknowledges, however, that there is strong resistance to charging for previously free parking.

Another approach to managing parking is improving the efficiency of the use of existing spaces, by informing drivers about available spaces, and by guiding them to alternate car parks, for example through electronic systems and display boards. Still, only a minority of car parks are monitored by electronic systems.

This poster presents an app that leverages the growing popularity and affordability of internet-enabled smartphones, and the wealth of data available online, to crowdsource parking availability information from drivers.

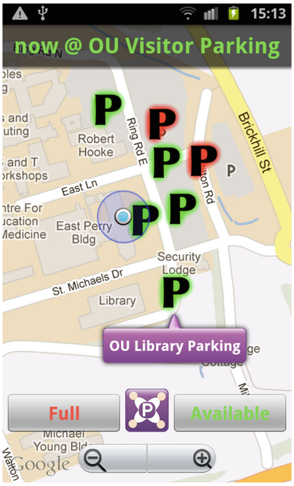
The name of the app, "ParkJam", may change when the app is released publicly, which is expected to happen well before the conference. More information can be found at http://parking.kmi.open.ac.uk/

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 512–514, 2015.
DOI: 10.1007/978-3-662-46641-4

Application Description

As shown in the screenshot on the next page, the app is built around a map view that shows car parks located in the zoomed-in area, which by default follows the user's location. The app can show the availability status of the car parks, and notify the user if the availability of a watched car park changes. Users can submit car park availability information; all submissions are aggregated to provide an up-to-date availability estimate for each car park.

In a separate view, the app will show any available detailed information about the car park, such as its opening hours and pricing. This information is initially taken from the LinkedGeoData project.[1] Where information about car parks is missing, ParkJam users will be able to add it, and the system will feed it back to LinkedGeoData. In effect, the app crowd sources the creation and maintenance of parking location and availability data. The aggregate results are published as linked open data, to enable other third-party mashups and applications.



Further, ParkJam integrates additional related data sources, as incentives for users to adopt the app. For example, the car park detail view can show nearby events (which may affect parking situation in the area), and dynamically discovered services associated with the car park, such as advance booking, that can be invoked directly from the app.

The research focus of the ParkJam project is mainly (1) on crowdsourcing near-real-time data, and (2) on publishing such near-real-time data as Linked Open Data. For crowdsourcing, we especially investigate how semantic data formats and the parking use case bear on the challenges listed in [2]: *How to recruit and retain users? What contributions can users make? How to combine user contributions to solve the target problem? How to evaluate users and their contributions?* Finally, publishing near-real-time semantic data is closely related to work on semantic sensors [3], and we look into how the app, or its users, can be seen as sensors.

There are many mobile apps that help with parking. At best, they show the up-to-date availability of a limited number of car parks; for instance SFpark[2] tracks selected on-street parking and parking garages in San Francisco. In contrast, ParkJam focuses on crowdsourcing of parking availability data from its users, which can be applied globally, at the cost of a somewhat lower data quality.

This poster will complement the demo of the same app, if accepted.

[1] http://linkedgeodata.org/

[2] http://sfpark.org/

Acknowledgements. This work is funded by the project PlanetData, a European Network of Excellence on Large-Scale Data Management, http://planet-data.eu.

References

1. Shoup, D.: The High Cost of Free Parking. University of Chicago Press (2005)
2. Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing systems on the world wide web. Commun. ACM, **54**, 86–96 (2011)
3. Sheth, A., Henson, C., Sahoo, S.: Semantic sensor web. IEEE Internet Comput. **12**(4), 78–83 (2008)

APA VIVO: A Semantic Framework for Scholarly Identity and Trusted Attribute Exchange

Hal Warren and Eva Winer(✉)

American Psychological Association, Washington, DC USA
{hwarren, ewiner}@apa.org

Abstract. In this paper, we describe a semantic approach to scholarly identity and scientific attribution based on a trust extension of VIVO, an open-source semantic social network platform for scientists. The Publish Trust pilot demonstrates how researchers can extend and manage verified claims of authorship in a semantic framework using VIVO instances and open identity technologies.

1 Introduction

This paper describes a pilot that examines the feasibility of adding trust values to online identities for authors of scholarly publications. Existing online networks for scientists (BiomedExperts, iamResearcher, etc.) do not include mechanisms for verifying self-asserted authorship claims. The pilot, developed by the American Psychological Association (APA), uses open source VIVO[1] semantic technology and the Open Identity Exchange (OIX) Trust Framework Provider[2] to deploy an author identity platform and track scientific attribution. VIVO is an ontology-driven Java application designed to facilitate the discovery of research expertise and enable scientific collaboration across disciplines. There are currently 50 VIVO instances at the leading research universities in the US, and about 20 international projects. VIVO is built on the Jena semantic web framework, and models researcher profiles based on core Vitro ontology, originally developed at Cornell University [1]. VIVO harvests data from verified sources like publication databases (PubMed, APA PsycNET), institutional human resources databases, grants, and data repositories, and ingests them into the matching researcher profiles. This data is represented as RDF and published as Linked Data.

[1] http://vivo.sourceforge.net/

[2] http://openidentityexchange.org/

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 515–517, 2015.
DOI: 10.1007/978-3-662-46641-4

2 APA VIVO Framework and Trusted Attribute Exchange

We developed the Publish Trust Framework (PTF) pilot to deploy and test reliable methods for trusted attribute exchange as an extension of VIVO semantic framework, where URIs identify people, groups, publications, events, equipment, etc. The APA pilot is centered on authors of articles published in APA's scholarly peer-reviewed journals, and produces publisher-validated trusted assertions of authorship, enabling scientists to reliably aggregate their works and connect with other experts in their field. We proof author accounts using APA intake forms at apa.publishtrust.org. Authors set conditions for trusted authorship attribute extension and retraction after the account holder identity is verified through a surface mail-back method.

Once the account is activated, authors are presented with a list of works they can claim. As attributes of authorship are extended from author.publishtrust.org to the APA VIVO profile, the status of the individual's VIVO page changes from "unverified" to "confirmed". Data for a community of over 3,000 authors is now available on APA VIVO instance https://vivo.apa.org.

Two-factor trusted claims of authorship are managed via the US government approved Open Identity Exchange (OIX) and Attribute Exchange Network (AXN), a secure closed network of Identity Providers, Attribute Providers, and Relying Parties that supports trust assertion payload delivery and consumption.

Exchanges result as authorized RDF-XML payloads that can be included within other VIVO instances as Relying Parties. This is indicated by a trustmark, which links back to RDF-based metadata supporting the claim. Attribute provider credentials contain a description of the assertion including a description of the source, the relationship between the source and the account holder, and a definition of the assertion made transparent at the base URL for the attribute. Authors remain in control of the privacy constraints for the attributes they have extended, and can retract those claims at their discretion. Claims can be challenged by other known identities in the framework, or anonymously.

The framework also provides an ability to anonymously assert expertise in a certain field. A clinical psychologist, for example, can offer advice on Facebook as a specialist on eating disorders. Anonymous ability is especially important in animal research and other sensitive research areas.

Publish Trust will allow researchers to validate many of the attributes that they assert on their profile. New APA-backed attributes, including trustmarked reviewer contributions, scientific Board and Task Force membership, and service to the profession will be harvested and added to the framework in the next phase of the pilot.

Cornell University is participating in the pilot as the first consumer of APA author attributes. Trust assertions move within a closed network at different levels of assurance. Using InCommon Federation SAML-based authentication as the single-sign-on mechanism allows attribute exchange and linking to APA attribute servers and resources from the account holder's Cornell VIVO profile. Each attribute provider and consumer server is authorized by InCommon protocols and SSL Certificates and meets National Institute of Standards and Technology Levels of Assurance-2 (NIST LOA-2) requirements [2].

For future work, we also aim to integrate PTF attributes via VIVO with ORCID (Open Researcher and Contributor ID) API and other author identity and disambiguation initiatives, engineer attribute binding using OpenID Connect protocol, and develop reputation and trust assessment algorithms for PTF assertions.

In conclusion, trusted attribute exchange creates a fundamental new opportunity for the advancement of science by improving scientific contributions through increased velocity in scholarly communication.

References

1. Krafft, D., Cappadona, N., Caruso, B., Corson-Rikert, J., Devare, M., Lowe, B., et al.: VIVO: Enabling national networking of scientists. In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, April 2010
2. Burr, W., Dodson, D., Newton, E., Perlner, R., Polk, W., Gupta, S., Nabbus, E.: Electronic Authentication Guideline: NIST Special Publication 800-63-1. National Institute of Standards and Technology (2011)

SciNet: Augmenting Access to Scientific Information

Tuukka Ruotsalo[1(✉)], Kumaripaba Athukorala[2], Antti Oulasvirta[1], Matti Nelimarkka[1], Samuli Hemminki[2], Petteri Nurmi[2], Patrik Floréen[2], Dorota Glowacka[2], Giulio Jacucci[2], Petri Myllymäki[2], and Samuel Kaski[1,2]

[1] Helsinki Institute for Information Technology HIIT,
Aalto University, Espoo, Finland
{Tuukka.ruotsalo,antti.oulasvirta,matti.nelimarkka,
samuel.kaski}@hiit.fi

[2] Helsinki Institute for Information Technology HIIT,
University of Helsinki, Helsinki, Finland
{Kumaripaba.athukorala,samuli.hemminki,petteri.nurmi,
patrik.floren,dorota.glowacka,petri.myllymaki,samuel.
kaski}@hiit.fi

Abstract. The information needs of researchers are increasingly personalized, tailored to the state of knowledge about different topics of the user, dependent on the work context, and part of an interactive process, where users are engaged with the scientific information space. We aim at revolutionizing the way scientific information can be accessed. This vision is realized as the SciNet, a framework that enables interactive scientific information access through personalized search and user profiling through monitoring the user's behavior and allowing the user to interact with the underlying user and data models.

The amount of scientific product is estimated to be millions of publications worldwide per year; the growth rate of PubMed alone is now 1.8 paper per minute[1], and Google Scholar indexes 2.93 million papers for the year 2011[2]. This is an indication that the established system of communicating the results of scientific research is already being challenged by the existence of the electronic publishing and distribution of the content through the Web. As predicted already a decade ago, we are only in the early days of a such digital revolution, one which will have a deeper and more disruptive impact on scientific publishing [1]. The problem of communication that the scientific community is facing is shifting from publishing and sharing the information to finding and filtering the suitable materials to support every day work of researchers; to augment scientific work practices. Specifically, we focus on the following technical challenges:

1. Epistemic Search. Taking different level of knowledge onof the user into account in the search process is a key factor affecting the satisfaction of the user. We model an

[1] http://www.ncbi.nlm.nih.gov/pubmed/

[2] http://scholar.google.fi/

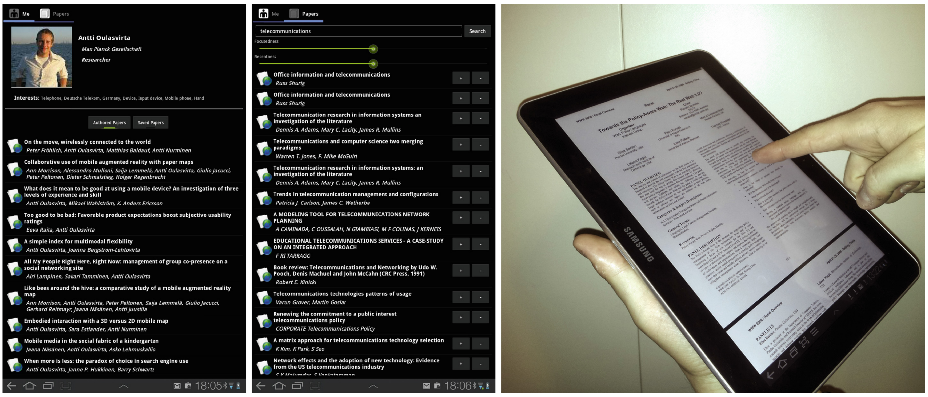
E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 518–520, 2015.
DOI: 10.1007/978-3-662-46641-4

epistemic state of a user using the age of the document, citation graph, and topicality.

2. Context Modeling. We model the context on two dimensions: topicality of the selections made by the user, including the inserted query and reading behaviour, and mobile context, all affecting the selection of a relevant part of the user profile.
3. Visualization and Interactive Feedback. The user can give feedback on the article level, but also directly manipulate the model parameters, such as preference for the age of the document or topical relevance.

To facilitate the scientific process and to address the above-mentioned challenges, we are in the process of constructing a set of applications to increase users' ability to receive sufficient information. Our current prototype, the SciNet, is an application framework consisting of three separate applications: search engine that uses topic modeling to build semantic models of the user's interests and ranks the results accordingly, context modeler that stores sensor input and other context data to enable better user profiling, and a custom data hubs that index and crawl content. Two interfaces are offered for the end users (shown in Fig. 1): an Android application that enables fetching initial profile information from the Web, searching and browsing the articles, and a custom PDF reader that tracks the text visible on the user's screen and stores the information in the user's profile. The current prototype is being deployed to be used by test users and we are investigating how it changes the way researchers search for scientific information.

The system currently indexes over 20 million resources from the following data sources. Certain data included herein are derived from the Web of Science prepared by THOMSON REUTERS, Inc., Philadelphia, Pennsylvania, USA: Copyright THOMSON REUTERS, 2011. All rights reserved, the Digital Library of the Association of Computing



Antti Oulasvirta
Max Planck Gesellschaft
Researcher

Interests: Telephone, Deutsche Telekom, Germany, Device, Input device, Mobile phone, Hand

On the move, wirelessly connected to the world
Peter Fröhlich, Antti Oulasvirta, Matthias Baldauf, Antti Nurminen

Collaborative use of mobile augmented reality with paper maps
Ann Morrison, Alessandro Mulloni, Saija Lemmelä, Antti Oulasvirta, Giulio Jacucci, Peter Peltonen, Dieter Schmalstieg, Holger Regenbrecht

What does it mean to be good at using a mobile device? An investigation of three levels of experience and skill
Antti Oulasvirta, Mikael Wahlström, K. Anders Ericsson

Too good to be bad: Favorable product expectations boost subjective usability ratings
Eeva Raita, Antti Oulasvirta

A simple index for multimodal flexibility
Antti Oulasvirta, Joanna Bergstrom-Lehtovirta

All My People Right Here, Right Now: management of group co-presence on a social networking site
Airi Lampinen, Sakari Tamminen, Antti Oulasvirta

Like bees around the hive: a comparative study of a mobile augmented reality map
Ann Morrison, Antti Oulasvirta, Peter Peltonen, Saija Lemmelä, Giulio Jacucci, Gerhard Reitmayr, Jaana Näsänen, Antti Juustila

Embodied interaction with a 3D versus 2D mobile map
Antti Oulasvirta, Sara Estlander, Antti Nurminen

Mobile media in the social fabric of a kindergarten
Jaana Näsänen, Antti Oulasvirta, Asko Lehmuskallio

When more is less: the paradox of choice in search engine use
Antti Oulasvirta, Janne P. Hukkinen, Barry Schwartz

telecommunications

Office information and telecommunications
Russ Shurig

Office information and telecommunications
Russ Shurig

Telecommunication research in information systems an investigation of the literature
Dennis A. Adams, Mary C. Lacity, James R. Mullins

Telecommunications and computer science two merging paradigms
Warren T. Jones, F. Mike McGuirt

Telecommunication research in information systems: an investigation of the literature
Dennis A. Adams, Mary C. Lacity, James R. Mullins

Trends in telecommunication management and configurations
Patricia J. Carlson, James C. Wetherbe

A MODELING TOOL FOR TELECOMMUNICATIONS NETWORK PLANNING
A CAMINADA, C OUSSALAH, N GIAMBIASI, M F COLINAS, J KERNEIS

EDUCATIONAL TELECOMMUNICATIONS SERVICES - A CASE-STUDY ON AN INTEGRATED APPROACH
F RI TARRAGO

Book review: Telecommunications and Networking by Udo W. Pooch, Denis Machuel and John McCahn (CRC Press, 1991)
Robert E. Kinicki

Telecommunications technologies patterns of usage
Varun Grover, Martin Goslar

Renewing the commitment to a public interest telecommunications policy
CORPORATE Telecommunications Policy

A matrix approach for telecommunications technology selection
K Kim, K Park, S Seo

Network effects and the adoption of new technology: Evidence from the US telecommunications industry
S K Majumdar, S Venkataraman

(a) Users can initiate their profiles based on existing Web sources.

(b) User sare provided with a personalized seam-less access to the article collections.

(c) A custom PDF reader is used to track the users' reading behaviour.

Fig. 1. The SciNet Android application (left) and the custom made PDF reader (right).

Machinery (ACM), the Digital Library of Institute of Electrical and Electronics Engineers (IEEE), and the Digital Library of Springer.

Reference

1. Berners-Lee, T., Hendler, J.: Scientific publishing on the ‘semantic web’. Nature, **410**, 1023–1025 (2001)

Managing Contextualized Knowledge with the CKR

Loris Bozzato[1(✉)], Francesco Corcoglioniti[1,2], Martin Homola[3], Mathew Joseph[1,2], and Luciano Serafini[1]

[1] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento Italy
{bozzato,corcoglio,mathew,serafini}@fbk.eu
[2] DISI, University of Trento, Via Sommarive 14, 38123 Trento, Italy
[3] FMFI, Comenius University, Mlynská dolina, 84248 Bratislava, Slovakia
homola@fmph.uniba.sk

As large amounts of Linked Data are published on the Web, it is becoming apparent that the validity of published knowledge is not absolute, but often depends on time, location, topic, and other contextual attributes. Therefore, an increasingly perceived need for Semantic Web (SW) applications is the representation of the *context* of such knowledge and its formalization for using it in reasoning and querying.

Recognizing this problem, several extensions of RDF and OWL to support contextual qualification of knowledge have been proposed [1,3,5,7]. Among these, we recently presented the Contextualized Knowledge Repository (CKR) [6], a framework with a well-founded semantics based on established AI principia [2,4] for contextual representation and reasoning.

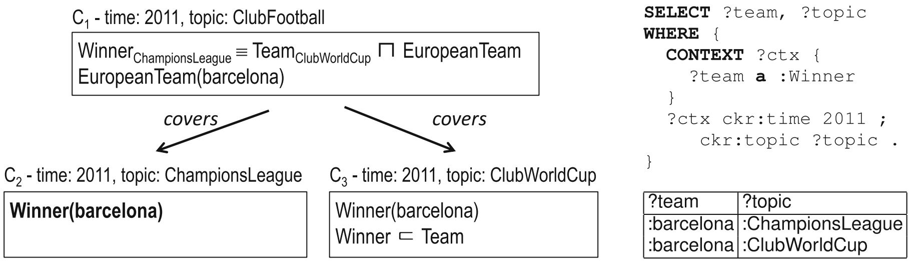
A distinguishing feature of the CKR is that contextual organization and knowledge propagation among contexts are largely derived from the qualification of knowledge along contextual dimensions: thus, users are not asked to manually express complex bridging axioms to define context relations.

While our previous work has mainly focused on the formal definitions and implementation of the CKR framework, the proposed poster illustrates the practical applicability of CKR features in real-world SW applications. The poster presents a concrete example of CKR use under the point of view of the tasks of *modelling, reasoning* over and *querying* contextualized knowledge.

In the poster, a modelling example in the domain of football is used to illustrate the use of CKR for managing contextual knowledge. Using the example of Fig. 1, we provide in the following an overview of the features presented in the poster.

Modelling. The CKR organizes *knowledge* in a set of OWL2 knowledge bases called *contexts* (e.g. C_1, C_2, C_3 in Fig. 1a) each one annotated with a set of dimension-value pairs that describe the circumstances in which statements inside the context hold. Context annotations are stored in the *meta-knowledge*: through OWL2 RL reasoning, they are used to identify relevant *compatibility relations* holding between contexts, such as the cover relation connecting broader with narrower-scoped contexts (other relations are under investigation). Inside a context, regular OWL2 classes and properties (e.g. Winner and Team) have a *local meaning* (i.e. context-dependent), while *qualified symbols* (such as $\mathsf{Winner}_{\mathsf{ChampionsLeague}}$, $\mathsf{Team}_{\mathsf{ClubWorldCup}}$) are introduced to

E. Simperl et al. (Eds.): ESWC 2012 Satellite Events, LNCS 7540, pp. 521–523, 2015.
DOI: 10.1007/978-3-662-46641-4


C1 - time: 2011, topic: ClubFootball
Winner_ChampionsLeague ≡ Team_ClubWorldCup ⊓ EuropeanTeam
EuropeanTeam(barcelona)
covers
covers
C2 - time: 2011, topic: ChampionsLeague
Winner(barcelona)
C3 - time: 2011, topic: ClubWorldCup
Winner(barcelona)
Winner ⊑ Team

(a) Contexts

```
SELECT ?team, ?topic
WHERE {
  CONTEXT ?ctx {
    ?team a :Winner
  }
  ?ctx ckr:time 2011 ;
       ckr:topic ?topic .
}
```

?team	?topic
:barcelona	:ChampionsLeague
:barcelona	:ClubWorldCup

(b) Query

Fig. 1. CKR example.

refer to the meaning of a class or property in a particular context, thus enabling the reference from a context to the meaning of a symbol in another context.

Reasoning. Reasoning in CKR is the mixing of two processes: *local reasoning* inside contexts and *knowledge propagation*} among contexts. The first is performed using regular OWL2 RL reasoning on the local contents of contexts; the latter is based on knowledge propagation rules that exploit compatibility relations and qualified symbols. For example, the statement Winner(barcelona) in C_2 (shown in bold in Fig. 1a can be derived by applying local reasoning in C_3 to infer Team(barcelona), which is then shifted up to C_1 obtaining $\mathsf{Team}_{\mathsf{ClubWorldCup}}$(barcelona); by local reasoning in C_1, $\mathsf{Winner}_{\mathsf{ChampionsLeague}}$(barcelona) is derived and then shifted down into C_2 obtaining Winner(barcelona). Through a repeated application of local reasoning and knowledge propagation, the *CKR closure* operation permits to materialize all inferrable statements.

Querying. *Contextual queries* in CKR are an extension of SPARQL where the keyword CONTEXT constrains the queried context. For instance, Fig. 1b presents a contextual query to extract all the winners of 2011 football competitions. Query answering is performed after the CKR closure operation is applied.

In conclusions, the poster presents how CKR features can be used for managing contextualized knowledge. Differently from other context frameworks, the CKR has both a well-founded semantics and is standard-friendly, as it is implemented by associating contexts to named graphs and CKR meta-knowledge to graph metadata. Therefore, the CKR represents an easily adoptable and workable solution for real-world SW applications needing to deal with contextualized knowledge.

References

1. Bao, J., Tao, J., McGuinness, D.L., Smart, P.: Context representation for the Semantic Web. In: Web Science Conference (2010)
2. Benerecetti, M., Bouquet, P., Ghidini, C.: Contextual reasoning distilled. JETAI, **12**(3), 279–305 (2000)

3. Schueler, B., Sizov, S., Staab, S., Tran, D.T.: Querying for meta knowledge. In: WWW-08. pp. 625–634. ACM (2008)
4. Lenat, D.: The Dimensions of Context Space. Technical report, CYCorp (1998)
5. Klarman, S., Gutiérrez-Basulto, V.: $\mathcal{ALC}_{\mathcal{ALC}}$: a context description logic. In: JELIA (2010)
6. Serafini, L., Homola, M.: Contextualized knowledge repositories for the Semantic Web. Web Semant. Sci. Serv. Agents World Wide Web, **12-13**, 64–87 (2012)
7. Straccia, U., Lopes, N., Lukacsy, G., Polleres, A.: A general framework for representing and reasoning with annotated semantic web data. In: AAAI-10. pp. 1437–1442 (2010)

Author Index