

Interoperation in Complex Information Ecosystems

Edited by

Andreas Harth¹, Craig A. Knoblock², Kai-Uwe Sattler³, and Rudi Studer⁴

1 KIT – Karlsruhe Institute of Technology, DE, harth@kit.edu

2 University of Southern California – Marina del Rey, US, knoblock@isi.edu

3 TU Ilmenau, DE, kus@tu-ilmenau.de

4 KIT – Karlsruhe Institute of Technology, DE, studer@kit.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13252 “Interoperation in Complex Information Ecosystems”.

Seminar 16.–19. June, 2013 – www.dagstuhl.de/13252

1998 ACM Subject Classification D.3.1 Formal Definitions and Theory, H.2.3 Languages, H.2.4 Systems, H.2.8 Database Applications, I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases Information integration, System interoperation, Complex information ecosystems, Dataspaces, Linked Data, Semantic Web, Sensor networks, Restful design, Web architecture.

Digital Object Identifier 10.4230/DagRep.3.6.83


1 Executive Summary

Andreas Harth

Craig A. Knoblock

Kai-Uwe Sattler

Rudi Studer

License  Creative Commons BY 3.0 Unported license
© Andreas Harth, Craig A. Knoblock, Kai-Uwe Sattler, and Rudi Studer

Individuals, enterprises and policy makers increasingly rely on data to stay informed and make decisions. The amount of available digital data grows at a tremendous pace. At the same time, the number of systems providing and processing data increases, leading to complex information ecosystems with large amounts of data, a multitude of stakeholders, and a plethora of data sources and systems. Thus, there is an increasing need for integration of information and interoperation between systems.

Due to the ubiquitous need for integration and interoperation, many research communities have tackled the problem. Recent developments have established a pay-as-you-go integration model, where integration is seen as a process starting out with enabling only basic query functionality over data and iteratively spending targeted integration effort as the need for more complex queries arises. Such an ad-hoc model is in contrast to previous integration models which required the construction of a mediated schema and the integration of schema and data before any queries – even simple ones – could be answered. The move towards less rigid integration systems can be traced back to many communities: the database community established Dataspaces as a new abstraction for information integration; the Semantic Web



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Interoperation in Complex Information Ecosystems, *Dagstuhl Reports*, Vol. 3, Issue 6, pp. 83–134

Editors: Andreas Harth, Craig A. Knoblock, Kai-Uwe Sattler, and Rudi Studer



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

community provided ontologies and logic-based modelling in a web context; finally, the Web community established the Hypermedia principle which enables decentralized discovery and ad-hoc unidirectional interlinking in very large information systems.

Current systems for data integration focus on query-related aspects. However, to enable real interoperation, updates and invocation of functionality are required. Mobile applications, for example, require both access to information and functionality. We want to broaden the scope of research on data integration towards a vision of interoperation between systems (i.e., systems that not only exchange and integrate their data but also link functionality) and investigate how an iterative model can be established for the interoperation of systems.

The seminar has multiple objectives:

- to bring together researchers from these diverse communities to identify common themes and to exploit synergies,
- to develop the theoretical foundations and an understanding of architectures and methods,
- to develop a research agenda and road-map towards a vision of web-scale integration and interoperation.

Acknowledgement

The seminar has been supported by ONR Global, grant N62909-13-1-C161.

2 Table of Contents

Executive Summary

Andreas Harth, Craig A. Knoblock, Kai-Uwe Sattler, and Rudi Studer 83

Overview of Talks

Subspace Methods for Data Integration
Felix Bießmann 87

Data Integration in Global Data Spaces
Christian Bizer 88

A Case for Prototypes: Knowledge Representation on the Web
Stefan Decker 89

Navigational Languages for the Web of Linked Data
Valeria Fionda 91

Improving Scientific Practice for Data Integration and Analysis
Yolanda Gil 92

Supporting the Linked Data Lifecycle
Armin Haller 95

On-the-fly Integration of Static and Dynamic Linked Data
Andreas Harth 97

Specifying, Executing, and Refining Complex Data Analytics Processes over Massive
 Amounts of Data
Melanie Herschel 99

Towards Vertebrate Linked Datasets
Aidan Hogan 101

Interoperability for Linked Open Data and Beyond
Katja Hose 104

A Process-Oriented View of Website Mediated Functionalities
Martin Junghans 105

Merits of Hypermedia Systems
Kjetil Kjernsmo 107

Next Generation Data Integration for the Life Sciences
Ulf Leser 109

Service- and Quality-aware LOD; the Yin and Yang of Complex Information
 Ecosystems
Andrea Maurino 109

A Purposeful View of Data: Getting the Data When You Need and How You Like
 It
Sheila McIlraith 111

Position Statement
Bernhard Mitschang 112

Next Generation Data Profiling
Felix Naumann 112

| | |
|---|-----|
| Bridging Real World and Data Spaces in Real-time: Data Stream Management and Complex Event Processing | |
| <i>Daniela Nicklas</i> | 113 |
| Cartography on the Web | |
| <i>Giuseppe Pirrò</i> | 114 |
| Completeness of RDF Data Sources | |
| <i>Giuseppe Pirrò</i> | 115 |
| Building Blocks for a Linked Data Ecosystem | |
| <i>Axel Polleres</i> | 116 |
| Towards Future Web-based Aerospace Enterprises | |
| <i>René Schubotz</i> | 118 |
| LITEQ: Language Integrated Types, Extensions and Queries for RDF Graphs | |
| <i>Steffen Staab</i> | 120 |
| Connecting Web APIs and Linked Data through Semantic, Functional Descriptions | |
| <i>Thomas Steiner</i> | 121 |
| Heterogeneous Information Integration and Mining | |
| <i>Raju Vatsavai</i> | 123 |
| Semantics and Hypermedia Are Each Other's Solution | |
| <i>Ruben Verborgh</i> | 124 |
| Working Groups | |
| Future of Actions | |
| <i>Réne Schubotz</i> | 126 |
| Data Profiling | |
| <i>Felix Naumann</i> | 126 |
| Enterprise Semantic Web | |
| <i>Melanie Herschel</i> | 127 |
| Planning and Workflows | |
| <i>Ulf Leser</i> | 128 |
| Linked Data Querying | |
| <i>Aidan Hogan</i> | 128 |
| Semantics of Data and Services | |
| <i>Sheila McIlraith</i> | 130 |
| Streams and REST | |
| <i>Daniela Nicklas</i> | 132 |
| Clean Slate Semantic Web | |
| <i>Giovanni Tummarello</i> | 132 |
| Participants | 134 |

3 Overview of Talks

3.1 Subspace Methods for Data Integration

Felix Bießmann (TU Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Felix Bießmann

Main reference F. Bießmann, F.C. Meinecke, A. Gretton, A. Rauch, G. Rainer, N.K. Logothetis, K.-R. Müller, “Temporal kernel CCA and its application in multimodal neuronal data analysis,” *Machine Learning Journal*, 79(1–2):5–27, 2010.

URL <http://dx.doi.org/10.1007/s10994-009-5153-3>

A lot of machine learning research focuses on methods for integration of multiple data streams that are coupled via complex time-dependent and potentially non-linear dependencies. More recently we started applying these methods to web data, such as online social networks or online publishing media, in order to study aspects of temporal dynamics between web sites and users in online social networks. Our results suggest that statistical learning methods can be useful for data integration in complex information ecosystems.

Many data sets in complex information ecosystems are characterized by multiple views on data. An example could be online social networks such as Twitter: one view on each data point – a short text message called *tweet* – is the actual text of the message. Another view on this data point is the geographical information coupled to this message.

The generative model underlying our analysis approach assumes there is a hidden variable, called Z , that gives rise to all possible views on this data point. But we usually can only observe one of multiple views X, Y, \dots on this hidden variable. In the Twitter example, the hidden variable would be all the semantic content of a message, and the views one can investigate are a textual realization and the geographical information. Both of these views carry complementary and shared information. The data integration problem is then to reconstruct the hidden variable from these multiple views. The reconstruction should ideally combine the textual and the geographical information. Combining this information can be useful for understanding geographical trends, trends in the text domain (i.e. topics) and how these two are related. For instance some topics could be connected to the geographical location of the users, while others are not.

An important complication that arises in the context of data integration is that different views on the hidden variables might not be coupled in a linear way. But most data integration algorithms rest on the assumption of linear couplings. Another important aspect is: Most modern data sets contain rapidly evolving high-dimensional time series. Especially when dealing with web data streams with high temporal resolution, the different views might be coupled via complex temporal dynamics.

To tackle these difficulties we developed a machine learning method, temporal kernel canonical correlation analysis (tkCCA) [1], for optimal data integration in the context of non-linearly and non-instantaneously coupled multimodal data streams. Given two (or more) multivariate time series $x \in \mathbb{R}^U$ and $y \in \mathbb{R}^V$ the method finds a projection ϕ_x and ϕ_y for each data view that projects the data into a *canonical subspace* in which x and y are maximally correlated

$$\operatorname{argmax}_{\phi_x, \phi_y} \operatorname{Corr}(\phi_x(y), \phi_y(y)).$$

The underlying assumption is that we can approximate the hidden variable by extracting a subspace that information in x and y that is maximally correlated. Working with the data in the *canonical subspace* can dramatically reduce computational complexity while preserving

all relevant information. In our future work we will extend this approach to different types of data from complex information ecosystems.

References

- 1 Felix Bießmann, Frank C Meinecke, Arthur Gretton, Alexander Rauch, Gregor Rainer, Nikos K Logothetis, and Klaus-Robert Müller. *Temporal kernel CCA and its application in multimodal neuronal data analysis*. Machine Learning Journal, 79(1-2):5–27, 2010.

3.2 Data Integration in Global Data Spaces

Christian Bizer (Universität Mannheim, DE)

License  Creative Commons BY 3.0 Unported license
© Christian Bizer

The idea of Data Spaces [1] as a new abstraction for large-scale data management has been around for quite some time but the development of concrete methods for integrating data from such spaces was still hampered by the lack of good testbeds that enable researchers to employ their methods in large-scale, real-world settings. This situation has changed in the last years with the emergence of two global public data spaces: The Web of HTML-embedded Data and the Web of Linked Data.

1. **Web of HTML-embedded Data** [2, 3, 4]: This data space consists of all web sites that embed structured data into their HTML pages using markup formats such as RDFa, Microdata or Microformats. A recent analysis¹ of the Common Crawl, a large web corpus consisting of around 3 billion pages, showed that 12.3% of all web pages embed structured data. These pages originate from 2.29m websites (PLDs) among the 40.5m websites contained in the corpus (5.65%). The main topical areas of the data are people and organizations, blog- and CMS-related metadata, navigational metadata, product data, and review data, and event data. The data is structurally rather shallow but plentiful. For instance, there are 35,000 websites providing product descriptions and 16,000 websites containing business listings, both challenging test cases for identity resolution as well as data fusion methods.
2. **Web of Linked Data** [5]: A public global data space in which data is more structured and in which data providers ease data integration by providing integration hints in the form of RDF links is the Web of Linked Data. The Web of Linked Data contains data from a wide range of domains including geographic information, people, companies, online communities, films, music, e-government data, library data and scientific research data².

I think that both data spaces provide challenging requirements and are nice testbeds for developing data space integration methods. I'm interested in discussing at the workshop the research challenges that arise from global public data spaces including questions such as:

1. **Data Space Profiling:** How to describing and summarize global data spaces as well as individual data sources within these spaces? Which metrics are suitable? Which sampling strategies make sense?

¹ <http://webdatacommons.org/>

² <http://lod-cloud.net/state/>


2. **Schema Matching:** How to adjust schema matching methods to the specifics of global data spaces? How to make them scale to 1000s of data sources? How to enable them to take advantage of integration hints provided by the community in the form of RDF links?
3. **Identity Resolution:** How to determine the potentially large set of data sources that describe a specific real-world entity? How to make identity resolution methods scale to situations involving 1000s of data sources?
4. **Data Quality Assessment:** How to assess the quality of data within global public data spaces? Which quality dimensions matter? Which metrics can be used to assess these dimensions? Which quality-related meta-information is available? How to fuse data from large sets of data sources based on the assessment results?

References

- 1 M. Franklin, A. Halevy, and D. Maier. *From Databases to Dataspaces: A new Abstraction for Information Management*. SIGMOD Record 34, 4, pages 27–33, 2005.
- 2 H. Mühleisen and C. Bizer. *Web Data Commons – Extracting Structured Data from two Large Web Corpora*. Proceedings of LDOW 2012: Linked Data on the Web, CEUR Workshop Proceedings. CEUR-ws.org, 2012.
- 3 P. Mika. *Microformats and RDFa Deployment across the Web*. <http://triple-talk.wordpress.com/2011/01/25/rdfa-deployment-across-the-web/>, 2011.
- 4 P. Mika and T. Potter. *Metadata Statistics for a large Web Corpus*. Proceedings of LDOW 2012: Linked Data on the Web, CEUR Workshop Proceedings. CEUR-ws.org, 2012.
- 5 T. Heath and C. Bizer. *Linked Data – Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan Claypool, 2011.

3.3 A Case for Prototypes: Knowledge Representation on the Web

Stefan Decker (National University of Ireland – Galway, IE)

License  Creative Commons BY 3.0 Unported license
© Stefan Decker

Motivation

Languages like OWL (Web Ontology Language) are providing means for knowledge representation on the Web using Description Logics [1], especially for the representation of Ontologies. They in particular provide means for the definition Classes. The notion of Classes and Instances has been evolved from Frame based representation languages and was first formalised by Pat Hayes in [4]. The same formalisation then has been used in the development of the theory of Description Logics. The notion of classes and instances has some properties which seem contrary to the notion of Knowledge Sharing on the Web and the development of reusable Web ontologies:

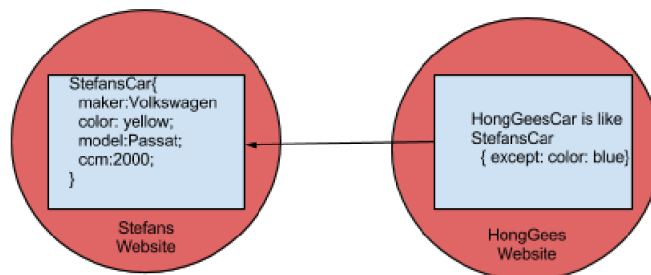
- During modeling of a particular domain the choice of what should be modeled as instances or a class sometimes seems arbitrary. Consider the creation of a knowledge bases about biological species. Should the species be modeled as classes or instances?
- Classes and Instances limit the way information can be shared. It enables only “vertical” information sharing by means of a central ontology, but provides no means of “horizontal” information sharing between peers, which seems to be a central purpose of a decentralised information system like the Web.

Prototypes

Early Frame Representation systems also deployed alternative approaches to classes and instances. [5] mentions “Prototypes: KRL, RLL, and JOSIE employ prototype frames to represent information about a typical instance of a class as opposed to the class itself and as opposed to actual instances of the class.” To our knowledge, a theory of prototype based knowledge representation has not been developed. However, prototypes have been investigated in programming languages. Early examples include Self [6, 7] and more recently languages like ECMAScript [3] and its various variants (e.g., JavaScript). The basic process of reuse and sharing in prototype based programming languages is called Cloning, whereby a new object is constructed by copying the properties of an existing object (its prototype). In addition the cloned object can be modified. In some systems the resulting child object maintains an explicit link to its prototype, and changes in the prototype cause corresponding changes to be apparent in its clone. Those ideas can be adapted to enable “horizontal” Knowledge Representation on the Web.

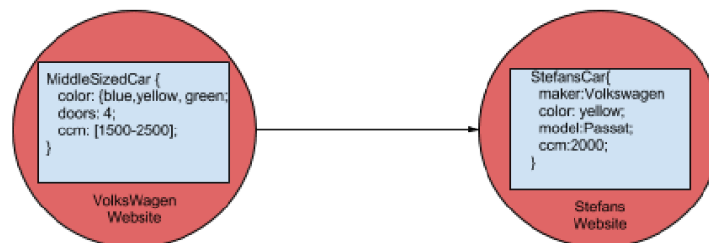
Using Prototypes on the Web – an Example

We illustrate the usage of prototypes for Knowledge Representation on the Web with two examples.



■ **Figure 1** Horizontal information sharing on the Web.

As an example consider Figure 1: a Website (or data site) publishes a object description called “StefansCar”. Another Website (“HongGees Website”) is referencing StefansCar and states that HongGeesCar is like StefansCar with the change of having a different color. This enables the development of a network of horizontally shared information and the emergence of “popular” objects, which could be widely used (and therefore of “Ontologies”).



■ **Figure 2** Emulation of Class-based Ontologies using Prototypes.

However, prototypes can also emulate the current use of class and instances based knowledge representation languages. As Figure 2 illustrates, prototypes can be used similar

to the notion of classes and can be specialised instead of just being cloned and changed. Figure 2 shows as an examples a prototype provided on a central side (a car maker), which is being used similar to to a class by being specialised. In consequence prototypes can still provide reusable vocabularies such as FOAF or SIOC.

Prototypes – an Attempt of a Research Agenda

Using prototypes for Knowledge Representation has been fallen by the wayside and consequently to our knowledge no theory has been developed in the last years. There is a need to at least investigate the following topics:

- Formalisation of Prototypes
- Inheritance Properties
- Learning new classifications (Data Mining)
- Representation
- Large Scale Storage and Querying of Prototypes

References

- 1 S. Decker, D. Fensel, F. Van Harmelen, I. Horrocks, S. Melnik, M. Klein, J. Broekstra. *Knowledge Representation on the Web*. Proceedings of the 2000 Description Logic Workshop, pages 89–97, 2000.
- 2 C. Dony, J. Malenfant, and D. Bardou. *Classifying prototype-based programming languages*. Ivan Moore, James Noble, and Antero Taivalsaari, editors, *Prototype-Based Programming*, pages 17–45. Springer Verlag, 1999.
- 3 *Standard ECMA262 ECMAScript Language Specification*. Edition 5.1, June 2011. <http://www.ecmascriptinternational.org/publications/standards/Ecma262.htm>
- 4 P. J. Hayes *The logic of frames*. D. Metzging, ed., *Frame Conceptions and Text Understanding*, Walter de Gruyter & Co., Berlin, 1979. Reprinted in Brachman & Levesque, pages 287–295, 1985.
- 5 P. D. Karp. *The Design Space of Frame Knowledge Representation Systems*. SRI International, 1993.
- 6 D. Ungar and R. B. Smith. *Self: The power of simplicity*, ACM Sigplan Notices, vol. 22. no. 12, pages 222–242, 1987.
- 7 R. B. Smith and D. Ungar. *Programming as an experience: The inspiration for Self*. Proceedings of ECOOP’95 – ObjectOriented Programming, 9th European Conference, Aarhus, Denmark. Lecture Notes in Computer Science Volume 952, pp 303–330, 1995.

3.4 Navigational Languages for the Web of Linked Data

Valeria Fionda (*Free University of Bozen-Bolzano, IT*)

License © Creative Commons BY 3.0 Unported license
© Valeria Fionda

Joint work of Consens, Mariano; Fionda, Valeria; Gutierrez, Claudio; Pirrò, Giuseppe

The increasing availability of structured data on the Web stimulated a renewed interest in its graph nature. The classical Web of interlinked documents is transforming into a Web of interlinked Data. In particular, the Linked Open Data project sets some informal principles for the publishing and interlinking of open data on the Web by using well-established technologies (e.g., RDF and URIs). The Web of linked data can be seen as a semantic graph where nodes represent resources and edges are labeled by RDF predicates.

Research in designing languages for accessing data in graphs and the intrinsic semantic nature of the Web of linked data suggest that graph navigation is a useful tool to address portion of this huge semantic graph. Navigation is the process of going from one point (node) in the graph to another by traversing edges. With semantic graphs, the traversal can go beyond the classical crawling that consists in traversing all the edges toward other nodes. It is possible to drive the traversal by some high-level semantic specification that encodes a reachability test, that is, the checking if from a given seed node there exists a path (defined by considering the semantics of edges) toward other nodes.

However, traditional techniques are not suitable for the Web of linked data graph. In fact the whole graph is not known a priori (as assumed by existing graph languages) but its structure has to be discovered on the fly. Hence, the notion of graph navigation has to be rethought to be useful in this new setting.

Some navigational languages have been proposed to work on *discoverable* graphs such as the Web of linked data (e.g. NAUTLOD [2] and GENTLE [1]). However, most of the current navigational languages enable to specify *relevant* resources on the Web of linked data (i.e., sets of nodes in the Web of linked data) connected by a sequence of edges that match an expression but they do not provide information about the structure of the fragment where these nodes have been found. Such piece of information is crucial in some contexts such for instance citation or social networks. Hence, there is the need to augment current navigational languages with capabilities to extract *fragments* (i.e., subgraphs) of the graph being navigated besides of sets of nodes [1, 3].

References

- 1 M. Consens, V. Fionda, G. Pirró. GENTLE: Traversing Discoverable Graphs. Short paper at the 7th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW) , 2013.
- 2 V. Fionda, C. Gutierrez, G. Pirró. Semantic Navigation on the Web of Data: Specification of Routes, Web Fragments and Actions. Proc. of the 21st World Wide Web Conference (WWW), pp. 281–290, 2012.
- 3 V. Fionda, C. Gutierrez, G. Pirró. Extracting Relevant Subgraphs from Graph Navigation. Proc. of the 11th International Semantic Web Conference (ISWC) (Posters & Demos) , 2012.

3.5 Improving Scientific Practice for Data Integration and Analysis

Yolanda Gil (University of Southern California – Marina del Rey, US)

License  Creative Commons BY 3.0 Unported license
© Yolanda Gil

Working with a variety of science data to support scientific discovery [8, 1], we face several major challenges.

Reducing the Cost of Data Preparation and Integration

It is estimated that 60-80% of the effort in a science project is devoted to data preparation and integration, before any science can be done. Reducing this effort is essential to accelerate the pace of discoveries. Our research focuses on several areas:

1. Reuse of data preparation software: We are looking at embedding open software practices within science communities, so that data preparation software is freely shared. We have developed semantic workflow techniques that can reason about constraints and metadata of both data and analytic steps [6], and learning approaches to extract workflow fragments that are common across workflows [3].
2. Crowdsourcing data preparation and integration through organic data sharing: We are investigating organic data sharing as a new approach to opening scientific processes so that the purpose of the data is transparent and encourages volunteer contributions from scientists [8]. An important aspect of this framework is that contributors get credit for their work, whether it is for sharing data they collect, describing data shared by others, or normalizing or integrating datasets so they can be analyzed together.
3. Leveraging the Web of Data for scientific data curation: Scientific data curation involves including metadata descriptions about world objects that may have been described elsewhere (e.g., limnology datasets could refer to lake characteristics, paleoclimatology datasets could refer to coral reef locations and species etc). We are developing user interfaces to Linked Open Data [12], and integrating them with scientific data curation systems.

Improving Provenance Practices

Scientists keep detailed provenance of their data, but in very rudimentary ways. Most of the tools that they currently use are oblivious to provenance and other metadata, which is typically managed by hand. This makes data very hard to understand by other people and by machines, and therefore its integration and reuse continues to require manual effort. The W3C PROV standard for Web provenance [9], finalized in April 2013, provides a basis for improving provenance practices, but many challenges remain:

1. Provenance-aware software: We must design data analysis software so that it uses the metadata that is available to automate data preparation and analysis, and so that it automatically creates appropriate metadata for any outputs generated [7]. Provenance-aware software can enable intelligent assistance and automation.
2. Reconstructing provenance: Provenance is typically incomplete and often incorrect, so it is important to develop approaches to make informed hypothesis about the provenance of data [11].
3. Incorporating provenance and metadata capture in science practice: Capturing provenance should be embedded in the practice of science rather than being an aside or afterthought. We are working on improving scientific publications with additional provenance, such as capturing methods as workflows [4] and improving data citations [10].

Data Science Education


A major challenge we face is the need to educate practitioners in all aspects of data science. Data science curricula are beginning to emerge that focus mostly on statistical aspects of data analytics, scale aspects concerning distributed execution, and database management. Existing curricula omit important topics such as data citation, provenance generation, and metadata tracking. Lack of awareness of these important aspects of data science is problematic, as practitioners need to address them and unfortunately end up doing so using primitive means that will not scale in the era of big data.

References

- 1 Deelman, E.; Duffy, C.; Gil, Y.; Marru, S.; Pierce, M.; and Wiener, G. *EarthCube Report on a Workflows Roadmap for the Geosciences*. National Science Foundation, Arlington, VA., 2012. <https://sites.google.com/site/earthcubeworkflow/earthcube-workflows-roadmap>
- 2 D. Garijo and Y. Gil. *A New Approach for Publishing Workflows: Abstractions, Standards, and Linked Data*. Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science (WORKS-11), held in conjunction with SC-11, Seattle, WA. 2011.
- 3 D. Garijo, O. Corcho and Y. Gil. *Detecting common scientific workflow fragments using execution provenance*. Proceedings of the International Conference on Knowledge Capture (K-CAP), Banff, Alberta, 2013.
- 4 D. Garijo, S. Kinnings, L. Xie, L. Xie, Y. Zhang, P. E. Bourne, and Yolanda Gil. *Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome*. To appear, 2013.
- 5 Gil, Y. and H. Hirsh (Eds). *Final Report of the NSF Workshop on Discovery Informatics*. National Science Foundation project report, August, 2012. <http://www.discoveryinformaticsinitiative.org/diw2012>
- 6 Gil, Y.; Gonzalez-Calero, P. A.; Kim, J.; Moody, J.; and Ratnakar, V. *A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogs*. Journal of Experimental and Theoretical Artificial Intelligence, 23(4), 2011.
- 7 Gil, Y.; Szekely, P.; Villamizar, S.; Harmon, T.; Ratnakar, V.; Gupta, S.; Muslea, M.; Silva, F.; and Knoblock, C. *Mind Your Metadata: Exploiting Semantics for Configuration, Adaptation, and Provenance in Scientific Workflows*. Proceedings of the Tenth International Semantic Web Conference (ISWC), Bonn, Germany. 2011.
- 8 Gil, Y.; Ratnakar, V.; and Hanson, P. *Organic Data Sharing: A Novel Approach to Scientific Data Sharing*. Proceedings Second International Workshop on Linked Science: Tackling Big Data (LISC), In conjunction with the International Semantic Web Conference (ISWC), Boston, MA. 2012.
- 9 Gil, Y.; Miles, S.; Belhajjame, K.; Deus, H.; Garijo, D.; Klyne, G.; Missier, P.; Soiland-Reyes, S.; and Zednik, S. *A Primer for the PROV Provenance Model*. World Wide Web Consortium (W3C) Technical Report, 2013.
- 10 A. Goodman, C. L. Borgman, K. Cranmer, Y. Gil, P. Groth, D. W. Hogg, V. Kashyap, A. Mahabal, X. Meng, A. Pepe, A. Siemiginowska, A. Slavkovic, R. Di Stefano. *Ten Simple Rules for the Care and Feeding of Scientific Data*. Submitted for publication, May 2013. <https://www.authorea.com/users/23/articles/1394/>
- 11 Groth, P.; Gil, Y.; and Magliacane, S. *Automatic Metadata Annotation through Reconstructing Provenance*. Proceedings of the Third International Workshop on the Role of Semantic Web in Provenance Management (SWPM), Heraklion, Greece, 2012.
- 12 Denny Vrandecic, Varun Ratnakar, Markus Krötzsch, Yolanda Gil. *Shortipedia: Aggregating and Curating Semantic Web Data*. Journal of Web Semantics, 9(3). 2011.

3.6 Supporting the Linked Data Lifecycle

Armin Haller (Australian National University, AU)

License  Creative Commons BY 3.0 Unported license
© Armin Haller

Introduction

The continuous growth of the Linked Data Web brings us closer to the original vision of the semantic Web – as an interconnected network of machine-readable resources. One of the reasons for the growth of Linked Data has been the significant progress on developing ontologies that can be used to define data in a variety of domains. The tools of choice for creating and maintaining quality-assured ontology instances (the so-called *ABox*) are still ontology editors such as Protégé. However, creating the *ABox* in an ontology editor requires some degree of understanding of RDF(s) and OWL since the user has to define to which class an individual belongs to and what are the permissible relationships between individuals. Further, as ontology editors do not separate the schema editing from the data editing, users can, for example, inadvertently make changes to the classes and relations in the ontology (the so-called *TBox*) while creating data. Addressing this issue, some Web publishing tools on top of Wikis, Microblogs or Content Management systems have been developed (e.g. the work discussed in [2], [6] and [3]) that allow a user to exclusively create ontology instances. However, they are mostly developed for a specific domain (i.e. specific ontologies) and often do not strictly follow OWL semantics and consequently allow the creation of an unsatisfiable *ABox*. Consequently, manually created, quality-assured, crowd-sourced semantic Web datasets are still largely missing. Drawing a parallel to data creation on the traditional Web, most of which happens through Web forms, an analogous method to create data is needed on the semantic Web. An abundance of tools exist to support developers in creating such Web forms operating on a relational database scheme. Many of them also support the Model-View-Controller (MVC) pattern where a developer can generate scaffolding code (Web forms) that can be used to create, read, update and delete database entries based on the initial schema. To create such a Web form-based tool that operates based on an ontological schema, a number of challenges have to be addressed:

- Web form data is encoded in a key/value pair model (application/x-www-form-urlencoded) that is not directly compatible to the triple model of RDF. Therefore, a data binding mechanism is needed that binds the user input in Web form elements to an RDF model.
- Whereas a Web form based on a relational table has a fixed set of input fields based on the number of table columns, the RDF model is graph based with potential cycles. Further, RDF(s) properties are propagated from multiple superclasses (including inheritance cycles) and the types of properties for a class are not constrained by the definition (Open World assumption). Consequently, methods are required to decide on the properties to be displayed in a Web form for a given RDF node.
- In contrast to the relational model where tuples are bound to a relation (table), class membership for individuals in RDF(s) is not constrained for a class. Thus individuals that have been created as a type of a specific class need to be made available for reuse within a different class instance creation process.
- Beyond the standard datatypes in the relational model that can be easily mapped to different form input elements (e.g. String/Integer to text boxes, Boolean to radio buttons, etc.), the OWL model supports object properties that link individuals to other individuals via URIs. Object properties can also span multiple nodes in an RDF graph, forming a

property chain, i.e. they can refer to a class that is linked to another class through more than one property. To aid users in the creation of object properties methods have to be established to identify and link to existing individuals and to enable the creation of new individuals in the process of creating the object property.

- Once the Linked Data is created, access methods have to be defined that allow to retrieve the data with the Web form they have been created with, but also with any other Web form that supports the editing of the same types of relations that are defined in the RDF instance data. Further, these Web forms need to allow the user to add new properties to easily extend the RDF instance with new relations.
- Data creation in Web forms is often dependent on computational analysis functionality that cannot be expressed in RDF/OWL directly. Thus, these computations are implemented as services that are called after the completion of one Web form, and potentially lead to the creation of further Web forms based on the input data of the preceding Web form. This workflow of connecting Web forms for the creation and maintenance of Linked Data need to be captured in a model that can be accessed together with the Web form and RDF data model at each step in the process.

Some work [5, 4, 1, 7] exist that address some of the challenges above, but a model and tool that addresses the whole lifecycle of Linked Data creation and maintenance, including the ability to execute an explicit process model that controls the Linked Data lifecycle is still missing. Such a RESTful Linked Data workflow engine will support ordinary Web users in the process of creating Linked Data, eventually fulfilling one of the vision of the Semantic Web to provide a platform in which the same data is not created many times over again, but reused many times in different contexts.

References

- 1 S. Battle, D. Wood, J. Leigh, and L. Ruth. The Callimachus Project: RDFa as a Web Template Language, 2012.
- 2 J. Baumeister, J. Reutelshoefer, and F. Puppe. KnowWE: a Semantic Wiki for knowledge engineering. *Applied Intelligence*, 35:323–344, 2011.
- 3 S. Corlosquet, R. Delbru, T. Clark, A. Polleres, and S. Decker. Produce and consume linked data with drupal! In *Proceedings of the ISWC 2009*, pages 763–778. 2009.
- 4 A. Haller, T. Groza, and F. Rosenberg. Interacting with Linked Data via Semantically Annotated Widgets. In *Proceedings of JIST2011*, pages 300–317, 2011.
- 5 A. Haller, J. Umbrich, and M. Hausenblas. RaUL: RDFa User Interface Language – A data processing model for Web applications. In *Proceedings of WISE2010*, 2010.
- 6 A. Passant, J. G. Breslin, and S. Decker. Open, distributed and semantic microblogging with smob. In *Proceedings of the ICWE 2010*, pages 494–497. 2010.
- 7 S. Stadtmüller, S. Speiser, A. Harth, and R. Studer. Data-Fu: A Language and an Interpreter for Interaction with Read/Write Linked Data. In *Proceedings of WWW2013*, Rio de Janeiro, Brasil, 2013.

3.7 On-the-fly Integration of Static and Dynamic Linked Data

Andreas Harth (KIT – Karlsruhe Institute of Technology, DE)

License © Creative Commons BY 3.0 Unported license
© Andreas Harth

Joint work of Andreas Harth, Craig Knoblock, Kai-Uwe Sattler, Rudi Studer

The relevance of many types of data perishes or degrades over time. Consider the scenario where a user wants to catch a bus to go to a sporting event: the current location of the bus is more relevant than yesterday's or last week's position. Having access to such real-time information facilitates fast decision-making. However, the manual alignment of the information with additional sources to gain a deeper understanding of the tracked objects and the observed area is a labor-intensive process. The expending of up-front effort to access data in real-time may out-weigh the advantage of having real-time information available.

To facilitate the integration of real-time sources, we propose to use a uniform approach to describe, access, and integrate real-time dynamic data, which in turn supports the integration with static geographical data and background knowledge. The types of supported data sources include:

- Static sources such as 2D maps, 3D models and point-of-interest (POI) data from XML files, Linked Data, and web APIs from the open web.
- Dynamic sources producing state updates (e.g. of moving objects), event information, etc. continuously or periodically, possibly with additional spatial/temporal properties, through web APIs.

A uniform approach enables the use of the integrated data in a variety of decision-supporting applications. For example, a real-time visualization that is annotated with background information can help a human user to make a well-informed and timely judgment of a situation. Additionally, software can automatically detect complex events based on the data and trigger appropriate notifications or actions. For example, a trigger could notify the user when to head for the bus stop based on the location of the approaching bus. Furthermore, having a uniform approach to describing, accessing, and integrating data sources enables the rapid discovery and addition of new relevant data sources. For example, the use of an existing bus application in a new country could require identifying and integrating new data sources.

Special value can be gained by integrating relevant data openly available on the web that has a quality and level of detail that is not achievable by a single, proprietary entity. Examples include data from OpenStreetMap or Foursquare that contain extensive geographical and meta-information about streets, buildings, businesses and other general points-of-interest (POIs). These data sets are maintained by huge numbers of contributing users.

To realize our goal, we need technologies that enable flexible, dynamic and scalable interactions of computing services and data sources.

As a key enabler for building such technologies we employ a uniform abstraction for components (resources) in large information systems termed Linked APIs. Components following the Linked API abstraction provide a standardized small set of supported operations and a uniform interface for both their data payload and their fault handling. Such a common abstraction allowing the manipulation of states as primitives enables us to specify the interactions between components declaratively both on the operational and data levels. We envision supporting the interoperability between web resources on an operational level similar to how semantic web technologies, such as RDF and OWL, support interoperability on a data level.

Given the minimal and unified interfaces of Linked APIs, we can use declarative means to specify interactions between different components in complex information systems [1]. The state manipulation abstraction and the high-level specifications describing the interplay between resources bring the following major benefits:

- Scalable execution: declarative specifications can be automatically parallelized more easily than imperative programs.
- Uniform and consistent error handling: instead of being confronted with source-specific error messages, universal error handlers can be realized.
- Substitution of resources: replacing a resource only requires adapting to the new vocabulary, while both the data model and the supported operations stay the same. Such flexibility is required as in large distributed information systems the underlying base resources may become unavailable and thus may put the entire networked application at risk.
- More flexible and cleaner specifications of interactions: the specifications can concentrate on the business logic of the intended interaction, while the operational interaction between components can be automated due to their standardized interfaces.

The overall benefits of such a method and apparatus are as follows:

- We may achieve real-time access to data integrated from several sources, some of them static and some of them dynamic.
- We can quickly integrate new data sources, as we use standard software interfaces to poll the current state of resources at specified time intervals or receiving updates and reacting on them, easing the transition from static to dynamic sources.
- We can quickly integrate new data sources, as the relation to the existing sources is specified declaratively, which allows for a high-level description of the interplay between resources that can be operationalized and optimized.

We believe that current web architecture offers the right abstraction and allows for the cost-effective implementation of such systems. Linked Data already allows for the integration of static data and the same mechanism can be used to achieve real-time functionality. In order to support interactive access to data, it will be necessary to execute extract/transform/load pipelines for performing integration at query time within seconds. We will also need tools to support end-users [2] in modeling real-time data sources to reduce the time needed to include a new live source into a constellation of systems that interoperate.

References

- 1 S. Stadtmüller, S. Speiser, A. Harth, and R. Studer. *Data-Fu: A Language and an Interpreter for Interaction with Read/Write Linked Data*. Proceedings of the 22nd International Conference on World Wide Web, WWW, pages 1225–1236, 2013.
- 2 M. Taheriyani, C. A. Knoblock, P. Szekely, and J. L. Ambite. *Rapidly Integrating Services into the Linked Data Cloud*. Proceedings of the 11th International Semantic Web Conference, ISWC, 2012.

3.8 Specifying, Executing, and Refining Complex Data Analytics Processes over Massive Amounts of Data

Melanie Herschel (University of Paris South XI, FR)

License  Creative Commons BY 3.0 Unported license
© Melanie Herschel

Overview

We are currently experiencing an unprecedented data deluge, as data in various formats is produced in vast amounts at an unprecedented rate. These properties of *variety*, *volume*, and *velocity* are at the core of the recent *Big Data* trend. One of the main issues of Big Data is how to make sense of the data that can no longer be handled or interpreted by a human alone. This commonly requires *integrating data* (to get a broader view of a subject) and then performing *analytical processes* on top of the data (to extract key figures of interest). In this context, we are particularly interested in how to specify and execute such Big Data Analytics processes. Section 3.8 presents our current efforts in this domain in the context of the Datalyse project.

Another observation is that in the dynamic context of Big Data, complex data transformation processes can no longer be designed and deployed once and left as-is afterwards, as has been the assumption for instance for data integration processes that require the design of a fixed global schema for the integrated result and the definition of a complex data integration workflow leading to the desired result. This observation has recently led to the *pay-as-you-go* paradigm, for instance proposed for data integration in *dataspaces* [5]. The main idea is to get first results fast by a rapidly developed solution that is good enough to get started, to then refine the process subsequently. Section 3.8 describes our vision of supporting the evolution of data transformation processes throughout their complete lifecycle.

Big Data Analytics

The *Datalyse* project³ on Big Data Analytics in the Cloud, started on May 1st 2013 and is a collaboration of several French industrial and academic partners (including Eolas, Business & Decision, INRIA, LIFL, LIG, and LIRMM). One objective is to provide a platform to facilitate developers to specify data analytical tasks over massive amounts of heterogeneous data provided by multiple data sources. As such, the platform provides both data integration and data analysis primitives that a developer can leverage when specifying the complete process in a declarative language. This language is compiled to be executed efficiently on a cloud based platform. In this context, we are particularly interested in the following aspects:

Data model. To be capable to provide processing primitives for a large variety of data sources, we first aim at defining a data model that captures this high variability. In particular, it should encompass NoSQL data models (such as JSON, XML, RDF) as well as associated schema specifications (e.g., XML Schema, RDFS). This data model, which extends previous work [4], will be used by processing primitives to access and manipulate data. These primitives include data access primitives, data transformation primitives (e.g., join, linking, aggregation) and data analytics primitives (e.g., data mining, clustering).

³ See <http://www.datalyse.fr/> for more details. The project is funded by the *Programme d'Etat des Investissements d'Avenir, Développement de l'Economie Numérique, Appel à projets "Cloud Computing" no. 3 - Big Data*.

Big Join and Linking. We will contribute to the definition, optimization and execution of both join and linking primitives. More precisely, we will first focus on *Big Join* (BJ), extending work of [8] to further take into account multi-source, heterogeneous, and continuous data. After developing an algebra for BJ manipulating data in the previously defined data model, we plan to develop and optimize BJ algorithms running on a Map/Reduce platform. We will follow a similar methodology for the *linking* primitive. The task of linking is very similar to a join, however, the goal is not only to join equal entries, but entries that refer to the same real-world entity, albeit being represented differently. Scalable linking has mainly focused on relational and hierarchical data [3] and we plan on further investigating scalable linking for complex data, for which few approaches have been proposed so far [1, 7].

Storing and indexing. The primitives described above are first defined at the logical level, before they are compiled into a physical plan that will be executed on a Map/Reduce platform. One essential aspect of physical plan execution is how to efficiently store and retrieve data, as demonstrated in [2]. To this end, we will investigate cloud storage and indexing strategies for data represented in our proposed data model and how to perform automatic storage optimizations.

Declarative Language. We will design a declarative language that allows developers to specify their analytical processes. This language then compiles into a logical plan using the primitives we consider. During compilation, we will perform optimizations targeted towards reducing the overall runtime of the process.

The developed techniques will be deployed in three real-world testbeds from different domains, i.e., monitoring, Open Government Data, and retail. In the first domain, we consider two use cases to ensure traceability, reporting, optimization, and analysis of irregular behavior w.r.t. energetic efficiency and IP network security, respectively. Concerning Open Data, we plan two use cases, i.e., one for data dissemination and one for data valorization. Finally, one retail use-case will focus on in-store and real-time business intelligence, whereas a second will concentrate on enriching catalogue data with semantic annotations.

Transformation Lifecycle Management

When developing complex data transformations, e.g., in the context of pay-as-you-go data integration, the data integration process (i.e., the data transformation leading to the integrated result) is gradually adapted and refined. The goal of transformation lifecycle management (TLM) is to semantically guide this refinement. The three main phases of TLM are (i) the *analysis* phase, where a developer verifies and analyzes the semantics of the data transformation (e.g., for debugging or what-if analysis), (ii) the *adapt* phase, where the transformation is changed (e.g., to fix a bug or to adapt to changing user requirements), and (iii) the *test* phase that helps in monitoring the impact of performed changes (e.g., to validate that the bug fix was indeed effective and no further error appeared).

Within the *Nautilus* project⁴, we are devising algorithms and tools to semi-automatically support all phases of TLM. Currently, we are leveraging data provenance techniques for the analysis phase [6] of data transformations specified in a subset of SQL. We have also recently started investigating what query modifications can reasonably be suggested for SQL queries and how to compute a set of reasonable query modifications. All proposed solutions still

⁴ <http://nautilus-system.org/>

lack efficient and scalable implementations, one avenue for future research. We also plan to support other data models and query languages in the future.

In the context of the *OakSaD* collaboration between the Inria Oak team and the database group at UC San Diego (<https://team.inria.fr/oak/oaksad/>), we plan to address the issue of analysis of complex business processes specified as data-centric workflows.

References

- 1 C. Böhm, G. de Melo, F. Naumann, and G. Weikum. Linda: distributed web-of-data-scale entity matching. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2104–2108, 2012.
- 2 J. Camacho-Rodríguez, D. Colazzo, and I. Manolescu. Web Data Indexing in the Cloud: Efficiency and Cost Reductions. In *EDBT – International Conference on Extending Database Technology*, Genoa, Italy, Mar. 2013.
- 3 P. Christen. *Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-centric systems and applications. Springer, 2012.
- 4 F. Goasdoué, K. Karanasos, Y. Katsis, J. Leblay, I. Manolescu, and S. Zampetakis. Growing Triples on Trees: an XML-RDF Hybrid Model for Annotated Documents. *VLDB Journal*, 2013.
- 5 A. Y. Halevy, M. J. Franklin, and D. Maier. Principles of dataspace systems. In *Symposium on Principles of Database Systems (PODS)*, pages 1–9, 2006.
- 6 M. Herschel and H. Eichelberger. The Nautilus Analyzer: understanding and debugging data transformations. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2731–2733, 2012.
- 7 M. Herschel, F. Naumann, S. Szott, and M. Taubert. Scalable iterative graph duplicate detection. *IEEE Transactions on Knowledge and Data Engineering*, 24(11):2094–2108, 2012.
- 8 A. Okcan and M. Riedewald. Processing theta-joins using mapreduce. In *International Conference on the Management of Data (SIGMOD)*, pages 949–960, 2011.

3.9 Towards Vertebrate Linked Datasets

Aidan Hogan (National University of Ireland – Galway, IE)

License  Creative Commons BY 3.0 Unported license
© Aidan Hogan

The Problem. . .

Compared to the closed, cold, metallic, top-down and decidedly inorganic structure of a typical relational database, when I think of Linked Datasets, I think of evolving, seething, organic creatures with an emergent and complex structure. These organic creatures come in different shapes and sizes and can sometimes work together for various complementary purposes: a loosely interlinked ecology of sorts.

When I think of a Linked Dataset *like* DBpedia⁵, I picture a gelatinous invertebrate entity—something like the creature from the 1958 movie “The Blob”: organic, vaguely formed, turbid, expansive, expanding, wandering around in an Open World and occasionally assimilating some of the more interesting scenery. There is a certain unique beauty to this complex creature, no doubt, but it is a beauty that few can appreciate. If one is versed

⁵ . . . here selecting a prominent example purely for argument’s sake.

in SPARQL, one can strike up a conversation with the creature and ask it a variety of questions about the things it has assimilated. If you phrase your question right (use the correct predicates and so forth) and the creature is in the mood (the endpoint is available), and it has assimilated the things you're interested in (has the data), you can sometimes get a response.

But it is an enigmatic creature: oftentimes the response will not be what you expected. For example, if you ask DBpedia something simple like how many countries are there:

```
SELECT (COUNT(?c) as ?count) WHERE { ?c a dbpedia-owl:Country }
```

it tells you 2,710. Sounds a bit high, no? Should be just shy of 200? Why did DBpedia say 2,710?

The Position...

Let's leave the admittedly belaboured and entirely unfair analogy of The Blob aside.⁶ Linked Datasets are often analogous to black boxes. All we typically know about these black boxes are the the shape of content they house (triples), the size of that content (number of triples), the category of that content (domain of data), and some features of that content (classes and properties used).⁷ Figuring out the rest is left as an exercise for the consumer. Oftentimes this is simply not enough and the consumer loses patience. So the question then is: how best to describe the contents of these black boxes.

The main aim of such a description should be to inform a consumer as to what expectations of freshness, accuracy and coverage of results they can expect for various types of queries over various chunks of the data (and let's assume that the endpoint performs perfectly now; an assumption that does not nearly hold in practice). The description should summarise the content of the black-box in such a way that it can be searched and browsed as a catalogue: it should allow a consumer to, hopefully at a few glances, *make sense* of the dataset they are querying, or at least a *core* of the dataset. My position in a few words: *we need intuitive mechanisms for humans to make sense of Linked Datasets (in whole and in part)*.

How can this be done? Relying on the semantics of the used terminology, as given in a vocabulary/ontology, says nothing about the data itself. We can look at instances of classes and common relationships between instances of various classes, which is quite a useful exercise, but class- centric descriptions do not tell much of the story. Again, in DBpedia, we have 2,710 instances of **Country**. So what definition of **Country** permits 2,710 instances? We can play around a bit and determine that a lot of listed countries no longer exist, or are not independent states, or are aliases or historical names, etc. But which are the current countries? And how many countries have flags or populations defined? How recent are those population measures?

It's great that DBpedia and other such datasets have lots of organic tidbits of information like historical countries and aliases and so forth. However, the result is a highly "non-normalised" soup of data that is difficult to describe and difficult to query over. We need a little *ordo ab chaos*. One solution would be to trim the fat from datasets like DBpedia so they fit into a highly normalised database-like schema that best fits the most common needs and that gives consumers a smoother experience. But this is not only unnecessary, it is inorganic, inflexible and, dare I say it, not very Semantic-Webby.

⁶ I have nothing but cautious affection for DBpedia and Linked Data.

⁷ If you're very lucky, you might find all of these details encapsulated in a VOID description.

One Proposal...

Instead of imposing a rigid inorganic structure on Linked Datasets like DBpedia so that they fit neatly into familiar rectangular frames of conceptualisation, perhaps we can just try find a natural shape to the dataset: a “spine”. We can begin by clustering instances in an extensional sense: looking at clusters of instances defined using the same predicates and the same types. For example, a cluster might be a group of instances that all have type `Country`, at least one `capital`, at least one `gdp`, and at least one `city`. We can call these clusters “abstract classes” or “prototypes” or “templates” or “least common factors” or “instance cores” or ... well in fact, such clusters are not even an entirely new idea (and are similar to, e.g., “characteristic sets” [2]) but no matter.

More important is what we can use these clusters for. In this model, clusters naturally form a subsumption hierarchy where instances within a cluster are also contained within less specific sub-clusters. The number of clusters and the level of detail they capture can be parameterised for their computation [1]. A person can browse the hierarchy of clusters—the “spine”—of a dataset to see at a glance what it contains and to find the cluster he/she can target with a query. One might start exploring the super-cluster containing instances with type `Country` and see it branch into one sub-cluster with 910 instances with `dbprop:dateEnd` (dissolved countries) and a disjoint sub-cluster of 191 instances with the subject `category:Member_states_of_the_United_Nations` (current countries recognised by the UN; 2 are missing). Browsing the hierarchy thus helps with understanding the scope and breadth of data in increasing detail, helps with disambiguation, and helps with formulating a query that targets only the instances of interest.

Furthermore, the richer and more complete the clusters, the higher the degree of homogeneity in those instances. Child-clusters in the hierarchy with similar cardinalities may indicate incomplete data: e.g., taking the UN member cluster of 191 instances, we find a sub-cluster with of 188 instances with defined capitals, where we could conclude that capitals are missing in 3 instances. Filling in the blanks will merge one step of the hierarchy and increase the homogeneity of the country descriptions. Merging highly-overlapping sub-clusters in the hierarchy then becomes a quantifiable bottom-up goal for local normalisation.

Subsequently, clusters can be annotated for the instances they encapsulate; e.g., in this cluster, capitals rarely change, populations are all as recent as 2008, GDP values are 87% accurate, etc. A directed (subject- to-object), labelled (predicate), weighted (count) graph can be constructed between clusters as the aggregation of the most common links between their instances. Furthermore, the integration of two or more Linked Datasets can then be coordinated through these clusters, with the goal of identifying and consolidating conceptually overlapping clusters to a high (and easily quantifiable) degree.


The resulting spine of the dataset then gives a core and a shape to the dataset; an entry point to follow; a way of distinguishing normalised and complete data from non-normalised and incomplete data; a basis for coordinating integration; an emergent structure from which all the other organic matter can extend.

References

- 1 J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes. *J. R. Stat. Soc.*, 66:815–849, 2004.
- 2 T. Neumann and G. Moerkotte. Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins. In *ICDE*, pages 984–994, 2011.

3.10 Interoperability for Linked Open Data and Beyond

Katja Hose (Aalborg University, DK)

License  Creative Commons BY 3.0 Unported license
© Katja Hose

Querying Linked Open Data

During the past couple of years, query processing as a foundation to achieve interoperability has been a very active field of research in the Semantic Web community. Especially, efficiently processing SPARQL queries over RDF data in general and Linked Open Data in particular attracted attention [9, 7, 5, 1, 2] with some approaches being conceptually closer to data integration in database systems than others.

Assuming a number of independent Linked Data sources (SPARQL endpoints), we have designed a framework and optimized distributed query processing of SPARQL queries over RDF data [10]. In addition to generating efficient query execution plans involving efficient implementations of operators, another problem in distributed systems is source selection, i.e., the decision whether a source should be considered during query processing or not. Apart from basic approaches such as budgets or time-to-live constraints, indexes are a key ingredient to enable source selection. On the one hand, we have developed approaches that estimate how many results a source might contribute to (part of) the query [4]. On the other hand, we have also considered benefit-based source selection [6] that also considers the overlap in the data of available sources and estimates the benefit of querying a particular source in terms of the unique new query results its data will produce. To make local processing at a single source with large amounts of RDF data more efficient, we have also been working on providing efficient solutions to SPARQL query processing in parallel scale-out systems by splitting the data into partitions and assigning the partitions to cluster nodes in a certain way that we can exploit during query processing [3].

Collaborative Knowledge Networks

The works mentioned so far focus on efficient processing of structured queries (SPARQL) over RDF sources, which is only one problem of many to achieve interoperability. Therefore, we have recently proposed a framework, Colledge [8], with the vision to support a higher degree of interoperability by allowing for interaction and combining aspects of P2P systems, social networks, and the Semantic Web.

In Colledge, functionalities such as query processing, reasoning, caching, data sources, wrappers, crowd sourcing, information extraction, etc. are modeled as services offered by nodes participating in the network. Queries are processed in a P2P-like manner and typically involve chains of nodes that are automatically selected and not known a priori. The user provides feedback on the correctness and relevance of the results, which is propagated based on collected provenance information to the nodes involved. This process helps detecting inconsistencies and errors, improves the query result over time, and eventually leads to updates of the original data, hence allowing for collaborative knowledge that is developing over time.

Challenges

In general, the problem of overcoming heterogeneity naturally arises whenever we want to enable interoperability between multiple data sources. This problem is not new and has

played a prominent role in research for quite some time now. Regarding interoperability for RDF and Linked Open Data sources alone, there are still a number of open issues, especially taking efficiency, updates, and reasoning into account. But there is also a great potential to learn from the advances and mistakes of other communities.

However, what we are mostly doing is developing solutions designed for a particular (sub)community, a bit of mapping and wrapping here and there. The question that remains to be answered is: Is this enough to handle the huge degree of heterogeneity that is coming along with accessing data on the Web (relational data, SQL, XML, XQuery, HTML, RDF, Linked Open Data, SPARQL, reasoning, information extraction, ontology matching, updates, Web services,...)?

Another questions that arises naturally is whether such a complex system is really what we need and want in the future. And if we want it, what are the main challenges? And what would be the best way to achieve interoperability then, (i) building upon the “old” ways and extending them to support the new problems that are introduced or (ii) is it possible to approach the whole problem in a different, novel way?

References

- 1 Carlos Buil Aranda, Marcelo Arenas, and Óscar Corcho. Semantics and optimization of the SPARQL 1.1 federation extension. In *ESWC (2)*, pages 1–15, 2011.
- 2 Valeria Fionda, Claudio Gutierrez, and Giuseppe Pirrò. Semantic navigation on the web of data: specification of routes, web fragments and actions. In *WWW*, pages 281–290, 2012.
- 3 Luis Galarraga, Katja Hose, and Ralf Schenkel. Partout: A distributed engine for efficient rdf processing. *CoRR*, abs/1212.5636, 2012.
- 4 A. Harth, K. Hose, M. Karnstedt, A. Polleres, K., and J. Umbrich. Data summaries for on-demand queries over linked data. In *WWW*, pages 411–420, 2010.
- 5 Olaf Hartig. Zero-knowledge query planning for an iterator implementation of link traversal based query execution. In *ESWC (1)*, pages 154–169, 2011.
- 6 Katja Hose and Ralf Schenkel. Towards benefit-based rdf source selection for sparql queries. In *SWIM*, pages 2:1–2:8, 2012.
- 7 Andreas Langegger, Wolfram Wöß, and Martin Blöchl. A Semantic Web middleware for virtual data integration on the Web. In *ESWC*, pages 493–507, 2008.
- 8 Steffen Metzger, Katja Hose, and Ralf Schenkel. Colledge – a vision of collaborative knowledge networks. In *2nd International Workshop on Semantic Search over the Web (SSW 2012)*, page . ACM, 2012.
- 9 Bastian Quilitz and Ulf Leser. Querying distributed RDF data sources with SPARQL. In *ESWC*, pages 524–538, 2008.
- 10 A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. FedX: Optimization techniques for federated query processing on linked data. In *ISWC*, pages 601–616, 2011.

3.11 A Process-Oriented View of Website Mediated Functionalities

Martin Junghans (KIT – Karlsruhe Institute of Technology, DE)

License  Creative Commons BY 3.0 Unported license
© Martin Junghans

Functionalities that are offered in form of interactive websites are omnipresent. It requires a substantial manual and tedious effort to use the functionalities within increasingly sophisticated use cases. Structured descriptions enable the development of methods that support users in dealing with everyday tasks. We present how a process-oriented view of the Web is

gained and helps end users, for instance, in finding information scattered across multiple websites.

The Web is not only an interlinked collection of documents. Functionalities provided in form of interactive Web pages and Web applications are offered and consumed on a regular basis by most of the end users connected to the Internet. Web pages serve as a front-end to access services and information of the Deep Web. In contrast to the Semantic Web with its aim to allow providers to annotate their services such that automatic discovery and composition are enabled, website mediated functionalities target primarily at human use.

By observing end users and their **browsing behavior**, the Web is perceived as a pool of functionalities solving simple tasks. Users select functionalities and use them in a certain sequence in order to achieve a goal. For example, in order to arrange a travel or to buy a high rated and cheap product, several individual functionalities are composed in the users' minds. Logic dependencies between inputs and outputs have to be managed manually and often the same inputs are provided at multiple websites repeatedly.

A large portion of the Web can be seen as a set of distributed and networked processes. They can provide access to information and cause effects during their execution. Also, they can require multiple interactions (such as form submissions and link selections) with the user. Unfortunately, these processes are currently not explicit. Users have to compose them every time again on their own. However, a **process-oriented view** on the Web requires an explicit and structured description of the functionalities and processes. In our approach, we model end user browsing processes that describe how users have to interact with which website at which time in order to consume functionalities and reach the desired outcomes. The descriptions allow to support users in combining, sharing, and reusing the processes, which capture previous efforts to achieve a goal [1].

We learned from the lack of public Semantic Web Services that we cannot rely on providers to create semantic annotations and to adopt a top-down formal semantics of Web services. So, we let users capture their browsing processes by existing **Web automation scripting** tools, which monitor, describe, and partially automate the process execution. In a bottom-up approach [2], semantic annotation can be added by users when needed, e.g., to describe elements of interactions and Web pages. Semantic annotations of websites can also be derived from the scripts [3].

Information Search Based on a Process-Oriented View

For many practical purposes, end users need information that is scattered across multiple websites. Consider for example an end user who is interested in knowing the names of the chairs of a particular track at the previous WWW conferences. As of today, Web search engines do not deliver satisfactory results for queries similar to “track chairs of all WWW conferences”. In order to obtain the required information the end user has to pose multiple queries to a search engine, browse through the hits, and aggregate the required information fragments outside of the found Web pages.

End users need help in selecting the pages that are relevant for obtaining the scattered information. Such a help must contain at least the set of the pages that the end user should visit, and support for invoking all the pages of the set easily. More advanced help could contain the complete end user browsing process including support for data flow between the user and the pages as well as among the pages, and control flow if there are data dependencies among inputs and outputs of Web pages in the set. We aim at providing the end users with a list of browsing processes that are relevant for a given information need instead of a list of links to Web pages. Each browsing process in the list of hits will lead the end user to the required information.

In order to search for existing browsing processes, e.g., from a repository shared with other users, we developed efficient **discovery** techniques. We proposed an offline classification of processes [4], which is based on formally defined classes, and the use of offline and online index structures [5] to efficiently locate desired browsing processes from large repositories. The discovery allows to reuse existing browsing efforts. The composition of browsing processes promises to create solutions to the information need that have not been executed before.

Acknowledgments

This paper presents results of joint efforts of a group of colleagues including Sudhir Agarwal (Stanford University), Charles J. Petrie (Stanford University), and the author.

References

- 1 Agarwal, S., Junghans, M.: *Swapping out coordination of web processes to the web browser*. Brogi, A., Pautasso, C., Papadopoulos, G. (eds.), ECOWS, IEEE pages 115–122, 2010.
- 2 Agarwal, S., Petrie, C.J.: *An alternative to the top-down semantic web of services*. IEEE Internet Computing 16(5), pages 94–97, 2012.
- 3 Agarwal, S., Junghans, M.: *Towards simulation-based similarity of end user browsing processes*. Daniel, F., Dolog, P., Li, Q. (eds.), ICWE, pages 216–223, Springer, 2013.
- 4 Junghans, M., Agarwal, S., Studer, R.: *Behavior classes for specification and search of complex services and processes*. Goble, C., Chen, P., Zhang, J. (eds.), ICWS, pages 343–350, IEEE, 2012.
- 5 Junghans, M., Agarwal, S.: *Efficient search for web browsing recipes*. ICWS, IEEE (to appear), 2013.

3.12 Merits of Hypermedia Systems

Kjetil Kjernsmo (University of Oslo, NO)

License © Creative Commons BY 3.0 Unported license
© Kjetil Kjernsmo

Research Interests

My primary research interest is the optimisation of SPARQL queries in a federated regime, as we have noted that this is not practical because the federation engine has insufficient information to optimise, or the information is so large that it defeats the purpose of optimisations to begin with. I plan to help remedy this problem by computing very compact digests and expose them in the service description. I have not yet published any articles on this topic, but the research is in the immediate extension of SPLENDID[3]. My secondary research interest is using statistical design of experiment in software performance evaluation.

However, coming from an industry background in software development, experience suggests that the above research interests does not adequately address many immediate needs when developing information systems to process the rapidly increasing amount of available data. I believe that large SPARQL-driven systems would be the “right tool for the job” in only a limited, and currently unclear, set of cases. Further exposure to new ideas in the developer community lead me to develop a third interest, namely hypermedia RDF.

Problems with SPARQL interfaces are many: They require extensive training of developers; it is not immediately clear what data are available and what may done with the data; it is easy to formulate queries that will cause the endpoint to become overloaded and hard to

protect against them without also rejecting legitimate queries; it places heavier systems in the execution path of an application, etc.

Hypermedia RDF

In [4], I examined some practical implications of the HATEOAS constraint of the REST architectural style, see [2] Chapter 5, and in that light argued why hypermedia RDF is a practical necessity.

Mike Amundsen defines hypermedia types[1] as

Hypermedia Types are MIME media types that contain native hyper-linking semantics that induce application flow. For example, HTML is a hypermedia type; XML is not.

We continued to derive a powerful hypermedia type based on RDF within a classification suggested by Mike Amundsen. Since this publication, I have also noted that the “embedded links” factor can be achieved by using data URIs, thus satisfying all but one of the factors proposed by Mike Amundsen.

Further, we noted that other interesting factors is the self-description, which is a important characteristic of the RDF model, and other minor concerns.

To bring forward a concrete example of how to make a serialised RDF graph into a hypermedia type, we suggest adding some triples to every resource (where prefixes are omitted for brevity):

```
<> hm:canBe hm:mergedInto, hm:replaced, hm:deleted ;
    hm:inCollection <../> ;
    void:inDataset [void:sparqlEndpoint </sparql> .] .
```

Possible Uses

Over time, I believe that interfaces that require developers to read external documentation will loose to interfaces where “View Source” is sufficient to learn everything that is needed. This is the essence of hypermedia systems, in the RDF case, everything needed to program is available in the RDF. It tells application what it may do next.

In the above, we have included mostly create, add and delete primitives, but these could be refined for application scenarios if needed.

For example a pizza baker publishes Linked Data about pizzas they sell, including data sufficient to create a sophisticated search and sales application. The Linked Data will then include triples that state explicitly how the order should be placed, i.e. what resources need updating. Moreover, the may not only want to sell pizzas, but also drinks. Thus, the Linked Data presented to the application should not only be a Symmetric Concise Bounded Description, as common today, but careful designed to provide exactly the data needed to optimise sales.


References

- 1 Mike Amundsen. Hypermedia Types. <http://amundsen.com/hypermedia/>, 2010.
- 2 Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- 3 Olaf Görlitz and Steffen Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In *Proceedings of the 2nd International Workshop on Consuming Linked Data*, Bonn, Germany, 2011.

- 4 Kjetil Kjernsmo. The necessity of hypermedia rdf and an approach to achieve it. In *Proceedings of the First Linked APIs workshop at the Ninth Extended Semantic Web Conference*, May 2012.

3.13 Next Generation Data Integration for the Life Sciences

Ulf Leser (*HU Berlin, DE*)

License  Creative Commons BY 3.0 Unported license


© Ulf Leser

Joint work of Sarah Cohen-Boulakia, Ulf Leser

Ever since the advent of high-throughput biology (e.g., the Human Genome Project), integrating the large number of diverse biological data sets has been considered as one of the most important tasks for advancement in the biological sciences. Whereas the early days of research in this area were dominated by virtual integration systems (such as multi-/federated databases), the current predominantly used architecture uses materialization. Systems are built using ad-hoc techniques and a large amount of scripting. However, recent years have seen a shift in the understanding of what a “data integration system” actually should do, revitalizing research in this direction. We review the past and current state of data integration for the Life Sciences and discuss recent trends in detail, which all pose challenges for the database community.

3.14 Service- and Quality-aware LOD; the Yin and Yang of Complex Information Ecosystems

Andrea Maurino (*University of Milan-Bicocca, IT*)

License  Creative Commons BY 3.0 Unported license

© Andrea Maurino

Introduction

In Chinese philosophy, the concept of yin and yang is used to describe how seemingly opposite or contrary forces are interconnected and interdependent in the natural world; and, how they give rise to each other as they interrelate to one another⁸. From a user perspective data and services are the yin and yang principle of complex information ecosystems. Data search and service invocation can be individually carried out, and both these operations provide value to users in the context of complex interactions. However, typically, users need to perform aggregated searches able to identify not only relevant data, but also services able to operate on them. The research on data integration and service discovery has involved from the beginning different (not always overlapping) communities. As a consequence, data and services are described with different models, and different techniques to retrieve data and services have been developed. Nevertheless, from a user perspective, the border between data and services is often not so definite. According to Chinese philosophy Yin and yang are actually complementary, not opposing, forces, interacting to form a whole greater than either separate part[1] and data and services provide a complementary view of the same ecosystems.

⁸ http://en.wikipedia.org/wiki/Yin_and_yang

Data provide detailed information about specific needs, while services execute processes involving data and returning an informative result. The advent of the Web of data and in particular linked open data (LOD) that is open data semantically interconnected, could help the definition of new approaches to effectively access data and services in a unified manner due to the fact that LOD and semantic web services speak the same language. Strictly related to the previous issue, quality in LOD is another new, challenging and relevant issue. While there is broad literature on data quality (in particular on both relational or structural data [1] there are very few works that consider the quality of linked open data [?]. Quality in LOD is a crucial problems for the effective reuse of published data and recently it has increased its relevance in the context of the research communities due to the availability of existing data. At the ITIS lab of University of Milano Bicocca both problems has been considered and in this position paper I report the most important results and the main issues to be still solved.

Quality in Linked Open Data

In my vision the multiple (unconsciously) marriage of database, sematic web and web communities brought the birth of linked open data. By considered LOD principles and its applications, we quickly recognize that parents of LOD are the three communities; because data must to be modeled, queried, indexed and so on (classic database topics), data must to be semantically linked and it is published on the Web by publishing their URIs. As a consequence, LOD brings some old issues coming from existing parents, but there are also new ones coming from the marriage. Quality in LOD is a typical example. In database community data quality is very well studied and a lot of methodologies and dimensions has been defined [3], but in LOD new issues arise and older ones are different. For example let consider the time related dimensions (such as currency) it is an important quality dimension that can easily managed in relational database by means of log file and temporal db. In LOD domain time, related information are very important because they can be used as proxy for the evaluation of validity of a rdf triple, but as recently shown [4] current practices on LOD publishing does not include such metadata. Moreover the completeness dimension is easily defined and assessed in the database community thanks to the “closed world assumption”, while at the level of Web this assumption is not correct and this make more and more difficult to measure the completeness of LOD. The study of quality in LOD is made harder due to the fact quality is in data consumer’s eyes not data producer’s ones and so techniques for assessing and improving LOD are strongly related to the need of data producers that are not known when data is published.

Service and Linked Open Data Integration

In [5] with other colleagues, I proposed a solution for aggregated search of data and services. We started by the assumption to have different and heterogeneous datasources and a list of semantic web services. In the proposed solution we first build the data ontology by means of the momis approach [6] then we create a service ontology by considering the semantic description of available web services. A set of mappings between the data and service ontology allow the possibility to search both data and services. A framework architecture comprising the data ontology (DO) manager, which supports the common knowledge extracted from heterogeneous sources, and XIRE, an information retrieval-based Web Service engine able to provide a list of ranked services according to a set of weighted keywords are designed and developed to evaluate the effectiveness of the overall approach. Evaluations based on

state-of-the-art benchmarks for semantic Web Service discovery shown that our information retrieval-based approach provides good results in terms of recall and precision. With the advent of quality enhanced LOD the proposed approach in [5] is still valid due to the fact that the data ontology can be considered as the LOD cloud that can be easily queried by SPARQL endpoints. Thanks to the fact that both data and services speaks the same (ontological) language the integration process can be easily considered as an ontology instance matching problems. The main issue in the integration of service and linked open data is the availability of semantically described services that it is still an open problem.

References

- 1 Sadiq, S., ed. *Handbook of Data Quality*. Springer, 2013.
- 2 Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S. *Quality assessment in linked open data*. Submitted for publication, 2013.
- 3 Batini, C., Cappiello, C., Francalanci, C., Maurino, A. *Methodologies for data quality assessment and improvement*. ACM Comput. Surv. 41(3), 2009.
- 4 Rula, A., Palmonari, M., Harth, A., Stadtmüller, S., Maurino, A. *On the diversity and availability of temporal information in linked open data*. Proc. International Semantic Web Conference, pages 492–507, Springer, 2012.
- 5 Palmonari, M., Sala, A., Maurino, A., Guerra, F., Pasi, G., Frisoni, G. *Aggregated search of data and services*. Inf. Syst. 36(2) pages 134–150, 2011.
- 6 Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D. *Semantic integration of heterogeneous information sources*. Data Knowl. Eng. 36(3), pages 215–249, 2001.

3.15 A Purposeful View of Data: Getting the Data When You Need and How You Like It


Sheila McIlraith (University of Toronto, CA)

License © Creative Commons BY 3.0 Unported license
© Sheila McIlraith

Most data, whether structured, semi-structured or unstructured, is acquired for a purpose, and that purpose often necessitates its composition with other data, and sometimes its transformation or aggregation via distinct sub-processes. Such composition and processing may be as simple as a set of join operations, typical in relational query processing, or as complex as a workflow of data analytics computations, selected on the fly, based on intermediate outcomes. In this talk, we look at the complex, evolving ecosystem of data and programs on the web and in the cloud, through the lens of (business) processes. We argue that (business) processes provide a vehicle for the customized and optimized selection, acquisition, integration, transformation, and display of data. Informally, the purpose for which the data is being collected can be described by a (business) process, mandating constraints on what data is collected, when it is collected and how it is transformed. I will report on our ongoing work in developing such optimized data-aware processes. This will be followed by a brief discussion of the relationship of this endeavor to the vision of Semantic Web Services, and a reflection on what, if anything, Semantic Web Services and the lessons learn from that effort have to contribute to the general task of developing complex interoperating data and sub-process ecosystems.

3.16 Position Statement

Bernhard Mitschang (Universität Stuttgart, DE)

License  Creative Commons BY 3.0 Unported license
© Bernhard Mitschang

The topic of the seminar is getting more and more important, since we do not only produce more and more data, we even do not erase data (we got somehow) anymore. Furthermore, the need to relate different portions of data is also getting more important due to informedness and actuality needs as well as for decision making needs. So the question naturally arises: What can/should we do from a technical perspective as well as from an economic perspective, and who pays the bill?

The technical view: all is there, somehow, see below. Money: well, if someone benefits from it, then, of course, this one pays, or it is free of charge, because of crowdsourcing, Google and the like, or governments pay for it.

I assume that to this seminar the first topic is the important one. So let me detail a bit on this: I can see that there is a bunch of ready-to-use and well understood technologies and systems that are perfectly suitable. For example:

1. ETL: transform and integrate starting from data portions taken from various data sources.
2. Data Analytics: aggregate, condense
3. Stream processing: “on the fly” processing of incoming data
4. Mashup technology, like Yahoo Pipes: ad hoc

I would like to discuss, whether these techniques are sufficient or not, and, if there is a need for new technology, what are the properties and characteristics.

3.17 Next Generation Data Profiling

Felix Naumann (Hasso-Plattner-Institut – Potsdam, DE)

License  Creative Commons BY 3.0 Unported license
© Felix Naumann


Profiling data is an important and frequent activity of any IT professional and researcher. We can safely assume that any reader has engaged in the activity of data profiling, at least by eye-balling spreadsheets, database tables, XML files, etc. Possibly more advanced techniques were used, such as key-word-searching in data sets, sorting, writing structured queries, or even using dedicated data profiling tools. While the importance of data profiling is undoubtedly high, and while efficiently and effectively profiling is an enormously difficult challenge, it has yet to be established as a research area in its own right.

References

- 1 Felix Naumann. *Data Profiling Revisited*. In SIGMOD Record, 2013.

3.18 Bridging Real World and Data Spaces in Real-time: Data Stream Management and Complex Event Processing

Daniela Nicklas (*Universität Oldenburg, DE*)

License  Creative Commons BY 3.0 Unported license
© Daniela Nicklas

Motivation

With the upcoming widespread availability of sensors, more and more applications depend on physical phenomena. Up-to-date real-world information is embedded in business processes, in production environments, or in mobile applications, where context-aware applications can adapt their behavior to the current situation of their user or environment. However, while more and more sensors observe the world, the ratio of data which is actually used is decreasing – we are drowning in a sea of raw data. Big data is often characterized in the dimension of volume, variety, and velocity. Dealing with this upcoming and ever-increasing stream of sensor data is not easy: it is often lowlevel (just raw sensor readings with no interpretation yet), distributed, noisy, bursty, and comes from heterogeneous sources, ranging from simple stationary single-value sensors (e.g., a thermometer) over mobile measurements to high-volume sensors like cameras or laser scanners. And archiving the data leads to ever-increasing storage needs. So, we face all three challenges of big data.

Background

Our approach to deal with these challenges is to develop generic data management systems for streaming data. The goal of data stream management is to provide the same flexibility and data independence for data streams as for stored data. For sensor data representing real-world situations, we need additional operators, e.g., to deal with different semantic layers (from raw data over features to objects/entities) or sensor data quality. Thus, we combine techniques from database management, probabilistic databases, sensor data fusion, and context-aware computing to create new base technology for these smart applications of the future. Our current application scenarios are highly dynamic world models for autonomous vehicles [5], safe offshore operations [6], ambient assisted living [4], and smart cities [2], and we implement our concept in the open source data stream framework Odysseus [1].

Open Questions

From this motivation (and from my background) we might discuss the following topics / challenges:

- Ecosystems of stream and non-stream data: how would system architectures look like that combine streaming and event-based data with large amounts of stored data? Could federated architectures be a solution [3]?
- Knowledge management: how can we manage supervised and un-supervised data mining techniques with online observation of relevant situations in streaming environments?
- Challenges from application domains: smart cities and/or smart factories (and the new German keyword industrie 4.0)


References

- 1 H.-Juergen Appelpath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. *Odysseus: a highly customizable framework for creating efficient event stream*

- management systems*. Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, DEBS '12, pages 367–368, 2012.
- 2 Marcus Behrendt, Mischa Böhm, Mustafa Caylak, Lena Eylert, Robert Friedrichs, Dennes Höting, Kamil Knefel, Timo Lottmann, Andreas Rehfeldt, Jens Runge, Sabrina-Cynthia Schnabel, Stephan Janssen, Daniela Nicklas und Michael Wurst. *Simulation einer Stadt zur Erzeugung virtueller Sensordaten für Smart City Anwendungen (Demo)*. 15. GI-Fachtagung Datenbanksysteme für Business, Technologie und Web, Magdeburg, 2013.
 - 3 Andreas Behrend, Daniela Nicklas und Dieter Gawlick. *DBMS meets DSMS: Towards a Federated Solution*. Proceedings of 1st International Conference on Data Technologies and Applications (DATA'12). SciTePress, 2012.
 - 4 Dennis Geesen, Melina Brell, Marco Grawunder, Daniela Nicklas und H.-Juergen Appelpath. *Data Stream Management in the AAL – Universal and Flexible Preprocessing of Continuous Sensor Data*. Reiner Wichert und Birgid Eberhardt (eds). Ambient Assisted Living, pages 213–228. Springer Verlag, 2012.
 - 5 Christian Kuka, Andre Bolles, Alexander Funk, Sönke Eilers, Sören Schweigert, Sebastian Gerwinn und Daniela Nicklas. *SaLSa Streams: Dynamic Context Models for Autonomous Transport Vehicles based on Multi-Sensor Fusion*. Proceedings of IEEE MDM 2013, 14th International Conference on Mobile Data Management, Milan, Italy, 2013.
 - 6 Nils Koppaetzky und Daniela Nicklas. *Towards a model-based approach for context-aware assistance systems in offshore operations*. Workshop proceedings of 11th Annual IEEE International Conference on Pervasive Computing and Communications, 2013.

3.19 Cartography on the Web

Giuseppe Pirrò (Free University of Bozen-Bolzano, IT)

License  Creative Commons BY 3.0 Unported license
© Giuseppe Pirrò

Joint work of Valeria Fionda, Claudio Gutierrez, Giuseppe Pirrò

Cartography is the art of map making. Abstractly, it can be seen a set of information transformations aimed at reducing the characteristics of a large selected area and putting them in a visual image, that is, a map. A map should meet the property of *abstraction*, that is, it has to be always smaller than the region it portraits. Besides, its visual representation plays a fundamental role for facilitating its interpretation by a map (human) user.

When mapping physical landscapes there are well-consolidated cartographic principles and techniques. However, we live in the Web era and thus applying cartographic principles also to digital landscapes becomes intriguing. Similarly to the Earth, the Web is simply too large and its interrelations too complex for anyone to grasp much only by direct observation. The availability of maps of Web regions is fundamental for helping users to cope with the complexity of the Web. Users via Web maps can track, record and identify conceptual regions of information on the Web, for their own use, for sharing/exchanging with other users and/or for further processing (e.g., combination with other maps). Indeed, Web maps are useful to find routes toward destinations of interest, navigate within (new) complex domains, and discover previously unknown connections between knowledge items.

Toward the development of Web maps, there are some challenging research issues. First, the Web is huge, distributed and continuously changing; therefore, techniques to specify, access and retrieve parts of it, that is, regions of interest are needed. Second, given a region of the Web, what is a reasonable definition of map? Third, is it feasible to efficiently and automatically build Web maps? Besides these points that focus on how to project

the principles of traditional cartography to the Web, additional research challenges emerge. One is the possibility to go beyond maps for only human users. This sets the need for having a mathematical model of regions and maps of the Web so that their properties and mutual relationships can be rigorously defined and understood. Hence, maps can be given a machine-readable format, which will foster their exchange and reuse. Moreover an algebra for maps can be defined, which will enable their combination via well-defined operations like union and intersection.

Nowadays, tools like bookmarks and navigational histories touch the problem of building maps of the Web. However, they do not comply with the notion of map that we envision for several reasons. First, they do not meet the property of abstractions: the Web region itself, that is the set of pages visited or bookmarked, is the map. Second, the map is a set of (disconnected) points; relations among them are lost. Third, these approaches rely on the manual activity of the Web user and thus are not suitable for the automation of the process of building maps. Last but not least, they are designed only for final human consumption. Initiative like Topic Maps face the problem of standardizing information management and interchange. The focus here is to manually create visual representations; Topic Maps cannot be automatically constructed (from the Web) and do not include an abstraction phase.

We investigate the applicability of cartographic principles to the Web. We model the Web space as a graph where nodes are information sources representing objects (e.g., people) and edges links between them (e.g., friendship). With this reasoning, a region becomes a (connected) subgraph of the Web and a map a (connected) subgraph of a region. Hence, we formalize the general problem of obtaining different kinds of maps from a graph. Our formalization makes maps suitable to be manipulated via an algebra also defined. To automate the construction of maps, we define a general navigational language that differently from existing languages, returning sets of (disconnected) nodes, returns regions. Then, we devise algorithms to efficiently generate maps from these regions. We instantiate and implement our map framework over the Web of Linked Open Data where thousands of RDF data sources are interlinked together. The implementation along with examples is available online⁹.

3.20 Completeness of RDF Data Sources

Giuseppe Pirrò (Free University of Bozen-Bolzano, IT)

License © Creative Commons BY 3.0 Unported license
© Giuseppe Pirrò

Joint work of Fariz Darari, Werner Nutt, Giuseppe Pirrò, Simon Razniewski

Main reference F. Darari, W. Nutt, G. Pirrò, S. Razniewski, “Completeness Statements about RDF Data Sources and Their Use for Query Answering,” in Proc. of the 12th International Semantic Web Conference, LNCS, Vol. 8218, pp. 66–83, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-41335-3_5

With thousands of RDF data sources today available on the Web, covering disparate and possibly overlapping knowledge domains, the problem of providing high-level descriptions (in the form of metadata) of their content becomes crucial. Such descriptions will connect data publishers and consumers; publishers will advertise “what” there is inside a data source so that specialized applications can be created for data source discovering, cataloging, selection and so forth. Proposals like the VoID vocabulary touched this aspect. However, VoID mainly

⁹ <http://mapsforweb.wordpress.com/>

focuses on providing *quantitative* information about a data source. We claim that toward comprehensive descriptions of data sources qualitative information is crucial. We introduce a theoretical framework for describing RDF data sources in terms of their completeness. We show how existing data sources can be described with completeness statements expressed in RDF. We then focus on the problem of the completeness of query answering over plain and RDFS data sources augmented with completeness statements. Finally, we present an extension of the completeness framework for federated data sources. The completeness reasoning framework is implemented in a tool available online¹⁰.

3.21 Building Blocks for a Linked Data Ecosystem

Axel Polleres (Siemens AG – Wien, AT)

License  Creative Commons BY 3.0 Unported license
© Axel Polleres

Linked Data has gained a lot of attention as a kind of “Silver Bullet” within the Semantic Web community over the past years. Still, some indications point at adoption not progressing as one might have expected optimistically around two years ago, when the standards and technologies around Linked Data seemed to be on the edge to “mainstream”. In this short position statement argue for critical reflection on the state of Linked Data and discuss missing building blocks for an effective Linked Data Ecosystem.

The promise of Linked Data as a common data platform is the provision of a rather lightweight, standardised mechanism to publish and consume data online. By (i) using RDF as a universal, schema-less data format and (ii) “linking” to different datasets via re-using URIs global as identifiers,¹¹ Linked Data bears the potential of building a true “Web” of data. Apart from RDF [7], the accompanying RDFS&OWL [2, 6] standards enable the description and “linkage” of schema information in a loosely coupled manner, plus finally SPARQL [5] provides standard means to access and query Linked data. As such, Linked Data provides the basis for enabling Web “dataspaces” [9].

The so-called Linking Open Data (LOD) cloud diagram¹² documents the development of openly accessible Linked Data datasets between May 2007 and September 2011 (comprising 295 datasets). Now, almost two years after the last incarnation of the LOD cloud diagram has been published, the community behind Linked Data is faced with high expectations to proof added value and one might ask her/himself what has happened since then. While there has not been a new diagram published since September 2011, we may take the number of currently 339 LOD datasets¹³ as an indication of developments since then, which may be viewed as at least a flattened growth rate in LOD. More worrying than the rate of growth (which may be hidden) though is actually the status of these LOD datasets. For instance, it seems that popular datasets that have been announced in 2010, such as the NYT dataset¹⁴ have not been updated since then: the most recent value for the RDF

¹⁰ <http://rdcorner.wordpress.com/>

¹¹ 1 paraphrasing Linked Data principles, cf. <http://www.w3.org/DesignIssues/LinkedData.html>, retrieved June 2013.

¹² <http://lod-cloud.net/>, retrieved June 2013

¹³ <http://datahub.io/group/lodcloud>

¹⁴ <http://data.nytimes.com/>

property http://data.nytimes.com/elements/latest__use pointing to the latest article about an organization is “2010-06-14”.

It is thus probably a good moment to take a step back to critically reflect on which puzzle pieces might be missing to achieve (even more) widespread adoption of Linked Data. Particularly, it seems that more than a few publishing principles and community enthusiasm is necessary to keep the idea of a Web-scale data ecosystem afloat. In the following, we will outline some challenges and missing building blocks to complement the available standards for Linked Data towards a fully functioning ecosystem.

Not all Linked Data is open. Particularly from an industry perspective, not all Linked Data will be open and available under open licenses: on the one hand, consumers will want to combine their own closed datasets with publicly available Linked Data; on the other hand, Linked Data may be published in different, non-compatible, possibly commercial licenses, which may impose restrictions on the use, re-use and re-publication of available data. In this context, it seems clear that the Linked Data community will need to provide standardised mechanisms to deal with access restrictions and different licenses, especially to make Linked Data interesting for industry. For starting points, cf. for instance Denny Vrandečić’s recent writeup¹⁵ or preliminary works on license composability via extending Semantic Web mechanisms [10].

Linked Data needs Mechanisms to deal with Dynamicity & Evolution. As the example in the introduction showed already, Linked Data may likely become outdated if not maintained properly. In this context, we note that there is a lack of standard technologies to both annotate temporal validity of evolving linked data as well as to process dynamic linked data as it evolves. We argue that standard technologies and best practices might help publishers to keep their data up-to-date and easy maintenance. Cf. for instance [8, 11].

Linked Data Quality needs Provenance & Trust. In order to determine trustworthiness and quality of Linked Data and combine data from different sources, it will be necessary to track provenance and trust, and to take these factors into account for query evaluation. Whereas the recent W3C PROV standard recommendation [4] provides a good starting point for describing and tracking provenance, integration with the remaining Linked Data standards and devising bespoke methods for query processing probably still needs more work.

Linked Data needs more (and less) than OWL. While not all features of OWL and particularly OWL2 seem to be equally adopted within published Linked Data [3], we note that a lot of published structured data is of numerical nature (e.g. public statistics). For integrating such data, different machinery than schema alignment supported via current ontology languages like RDFS and OWL is needed; rather, standard mechanisms for unit conversion or other mathematically expressible dependencies among properties are needed, cf. [1] for possible starting point.

The author is looking forward to discuss how these missing building blocks can be built and combined into a working ecosystem for Linked Data in the Dagstuhl seminar on “Interoperation in Complex Information Ecosystem”. Starting points mentioned in the present position paper do not mean to be exhaustive and we shall be discussing further inputs. The author’s expectation on the seminar is a road-map outlining

1. how these building blocks can implemented in terms of industry strength standards and best practices that can interplay and scale on the Web and
2. where further fundamental research is necessary.

¹⁵ <https://plus.google.com/104177144420404771615/posts/cvGay9eDSSK>

References

- 1 Bischof, S., Polleres, A. *RDFS with attribute equations via SPARQL rewriting*. Proceedings of the 10th ESWC. vol. 7882, pp. 335–350. Montpellier, France, 2013.
- 2 Brickley, D., Guha, R. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, 2004. <http://www.w3.org/TR/rdf-schema/>
- 3 Glimm, B., Hogan, A., Krötzsch, M., Polleres, A. *OWL: Yet to arrive on the web of data?* Proceedings of WWW2012 Workshop on Linked Data on the Web (LDOW 2012), 2012.
- 4 Groth, P., Moreau, L. *An overview of the PROV family of documents* W3C Recommendation, 2013. <http://www.w3.org/TR/prov-overview/>
- 5 Harris, S., Seaborne, A. *SPARQL 1.1 query language*. W3C Recommendation, 2013. <http://www.w3.org/TR/sparql11-query/>
- 6 Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. *OWL 2 Web Ontology Language Primer*. W3C Recommendation, 2009. <http://www.w3.org/TR/owl2-primer/>
- 7 Manola, F., Miller, E., McBride, B. *RDF Primer*. W3C Recommendation, 2004. <http://www.w3.org/TR/rdf-primer/>
- 8 Umbrich, J., Karnstedt, M., Hogan, A., Parreira, J.X. *Hybrid SPARQL queries: Fresh vs. fast results* Proceedings of ISWC 2012, pages 608–624, 2012.
- 9 Umbrich, J., Karnstedt, M., Parreira, J.X., Polleres, A., Hauswirth, M. *Linked Data and Live Querying for Enabling Support Platforms for Web Dataspaces*. Proceedings of the 3rd International Workshop on Data Engineering Meets the Semantic Web (DESWEB). Washington DC, USA, 2012.
- 10 Villata, S., Gandon, F. *Towards licenses compatibility and composition in the web of data*. Proceedings of ISWC2012 Posters & Demos, 2012.
- 11 Zimmermann, A., Lopes, N., Polleres, A., Straccia, U. *A general framework for representing, reasoning and querying with annotated semantic web data* JWS 12, 72–95, 2012.

3.22 Towards Future Web-based Aerospace Enterprises

René Schubotz (EADS Innovation Works – Munich, DE)

License  Creative Commons BY 3.0 Unported license
© René Schubotz

Aerospace industry is in the midst of a deep evolution of its industrial organization model. Refocusing on architect-integrator activities, product life-cycle modularization and outsourcing policies are major critical success factors and stimulate the emergence of virtual extended enterprises. This necessitates adequate integration and interoperation strategies between the architect-integrator and its risk sharing partners participating in the design and manufacturing of aerospace products.

Facing the abundance of stakeholders, schemata and systems in hitherto “non-information industry” enterprises, the Linked Data promise of provisioning a *single* common data platform is tantalizing and nurtures the long-term vision of consolidated and trustworthy (engineering) data spaces that integrate data in all phases of the product lifecycle, such as shape and geometry, various facets of its physical and functional description, structural and material properties, models of analyses, engineering trade-off decisions, manufacturing information, etc.

Yet, industrial uptake of Linked Data standards and technologies is timid. In the following, the author tries to indicate some impediments at different organizational levels.

- **On the inter-enterprise level**, pressing questions concerning data security, confidentiality, trust and provenance need to be addressed. Previous works [4, 3] investigate potential vectors of attack, however, an industrial-strength Web of Data requires substantially more hardening. Moreover, industrial businesses require a notion of Linked Closed Data [2] which is published with access and license restrictions and therefore demands standardizing access, authentication and payment protocols.
- **On the enterprise level**, the challenge is to integrate data and workflows of product lifecycle tools in support of end-to-end lifecycle processes. This requires the exploration of suitable integration techniques with minimalistic specification effort. First steps towards this direction have been taken in the form of a Linked Data Basic Profile [5], sparking considerable community interest [1].
- **On the specialty department level**, highly elaborate engineering design pipelines need to be captured. By exploiting the workflow paradigm for capturing the design of engineering workflows, and RDF to interlink the workflow, its specialty domain engineering tools as well as static and dynamic data sources, increased efficiency of design, engineering and evaluation activities becomes possible. To realize this goal, we need languages and execution environments [8] that enable scalable and flexible utilization and manipulation of computing and data resources.
- **On the level of the individual, working engineer**, user-centered data perspectives on the engineering data space should enable a wide range of interactions with respect to any engineering task. However, the working engineer is *not* a working ontologist. This necessitates adequate navigational languages in order to explore the data relevant for the task at hand, as well as intuitive user-interfaces [6, 7] making the consumption and publication of Linked Data light-weight, easy-to-use and easy-to-understand.

The author firmly believes in the applicability of the current web architecture in the extended industrial enterprise, and is looking forward to discuss and exchange views on the current trends, challenges, and state of the art solutions.


References

- 1 Open services for lifecycle collaboration. <http://open-services.net/>.
- 2 M. Cobden, J. Black, N. Gibbins, L. Carr, and N. Shadbolt. A research agenda for linked closed data. In *Second International Workshop on Consuming Linked Data (COLD2011)*, 2011.
- 3 A. Hasnain, M. Al-Bakri, L. Costabello, Z. Cong, I. Davis, T. Heath, et al. Spamming in linked data. In *Third International Workshop on Consuming Linked Data (COLD2012)*, 2012.
- 4 A. Hogan, A. Harth, and A. Polleres. Scalable authoritative owl reasoning for the web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(2):49–90, 2009.
- 5 A. J. Le Hors, M. Nally, and S. K. Speicher. Using read/write linked data for application integration—towards a linked data basic profile. *Fifth International Workshop on Linked Data on the Web (LDOW2012)*, 2012.
- 6 M. Luczak-Rösch and R. Heese. Linked data authoring for non-experts. In *Second International Workshop on Linked Data on the Web (LDOW2009)*, 2009.
- 7 R. Schubotz and A. Harth. Towards networked linked data-driven web3d applications. In *First International Workshop on Declarative 3D for the Web Architecture (Dec3D2012)*, 2012.

- 8 S. Stadtmüller, S. Speiser, A. Harth, and R. Studer. Data-fu: A Language and an Interpreter for Interaction with read/write Linked Data. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1225–1236, 2013.

3.23 LITEQ: Language Integrated Types, Extensions and Queries for RDF Graphs

Steffen Staab (Universität Koblenz-Landau, DE)

License  Creative Commons BY 3.0 Unported license
© Steffen Staab

Joint work of Steffen Staab, Scheglmann Steffen, Gerd Gröner, Eveyne Viegas

Motivation

RDF data representations are flexible and extensible. Even the schema of a data source can be changed at any time by adding, modifying or removing classes and relationships between classes at any time. While this flexibility facilitates the design and publication of linked data on the Web, it is rather difficult to access and integrate RDF data in programming languages and environments because current programming paradigms expect programmers to know at least structure and content of the data source. Therefore, a programmer who targets the access of linked data from a host programming language must overcome several challenges. (i) Accessing an external data source requires knowledge about the structure of the data source and its vocabulary. As linked data sources may be extremely large and the data tend to change frequently, it is almost impossible for programmers to know the structures at the time before they develop their programs. Therefore, approaches to simplify access to RDF sources should include a mechanism for exploring and understanding the RDF data source. (ii) There is an impedance mismatch between the way classes (types) are used in programming languages compared to how classes structure linked data. (iii) A query and integration language must be readable and easily usable for an incremental exploration of data sources. (iv) When code in a host language describes how RDF data is to be processed by the resulting program, the RDF data should be typed and type safety should be ensured in order to avoid run time errors and exceptions. To address these challenges, we present LITEQ, a paradigm for querying RDF data, mapping it for use in a host language, and strongly typing it for taking full advantage of advanced compiler technology.

In particular, LITEQ comprises:

- The node path query language (NPQL), which has an intuitive syntax with operators for the navigation and exploration of RDF graphs. In particular, NPQL offers a variable free notation, which allows for incremental writing of queries and incremental exploration of the RDF data source by the programmer.
- An extensional semantics for NPQL, which clearly defines the retrieval of RDF resources and allows for their usage at development time and run time.
- An intensional semantics for NPQL, which clearly defines the retrieval of RDF schema information and allows for its usage in the programming environment and host programming language at development time, compile time and run time. Our integration of NPQL into the host language allows for static typing – using already available schema information from the RDF data source – making it unnecessary for the programmer to manually re-create type structures in the host language.

Discussion

LITEQ has been partially implemented as part of F#, full implementation is underway. LITEQ benefits from type providers in F# that support the integration of information sources into F# [2] such that external data sources are directly available in programs. Type provider use F# LINQ queries [1] to retrieve schema information from (Web) data sources in order to build the corresponding types at run-time. Several Type Provider demonstrate the integration of large data sources on the Web, like the Freebase Type Provider that allows for the navigation within the graph-structure of Freebase¹⁶. The novel contribution of LITEQ is the full exploration of properties and the distinction of extensional and intensional use of the LITEQ query language. The core advantage of LITEQ compared to other integration approaches that map RDF into a host language is its integration of the different phases of exploration, compile and run time – benefitting both from the type definitions and the extensional queries.

Acknowledgements

This work has been supported by Microsoft.

References

- 1 Erik Meijer, Brian Beckman, and Gavin Bierman. *LINQ: Reconciling Object, Relations and XML in the .NET Framework*. In Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, page 706, Chicago, Illinois, June 2006.
- 2 D. Syme, K. Battocchi, K. Takeda, D. Malayeri, J. Fisher, J. Hu, T. Liu, B. Mc-Namaa, D. Quirk, M. Tavecchia, W. Chae, U. Matsveyeu, and T. Petricek. *F# 3.0 — Strongly Typed Language Support for Internet-Scale Information Sources*. Technical Report MSR-TR-2012-101, Microsoft Research, 2012.

3.24 Connecting Web APIs and Linked Data through Semantic, Functional Descriptions

Thomas Steiner (Google – Hamburg, DE)

License  Creative Commons BY 3.0 Unported license
© Thomas Steiner

Joint work of Verborgh, Ruben; Steiner, Thomas; Mannens, Erik; Van de Walle, Rik

In the beginning days of the Semantic Web, developers considered “services” as something separate from the “data” Web. There was a clear distinction between the weather offered by the webpage <http://example.org/weather/saarbruecken/today> and the weather offered through the endpoint found at <http://example.org/services>, which required you to construct a POST request with a body of `method=getWeather&city=Saarbrücken`. The former could be described as data, the latter had to be described as a service, while both were actually doing the exact same thing. Describing data could be done with simple RDF, whereas services needed to be described quite verbosely with OWL-S or WSMO.

Services built in the REST architectural style [3] do not exhibit this complexity, because the unit of a REST API is a *resource*, which is *identical* to the RDF concept of a resource. In

¹⁶The Freebase Wiki about the Schema: <http://wiki.freebase.com/wiki/Schema>

fact, “REST service” is a *contradictio in terminis*, because the purpose of a REST API is to *not* stand out as a service, but rather to expose application concepts as resources. It is a matter of hiding internal implementation details (which are subject to change anyway) in order to guarantee the evolvability of the exposed API. In a REST API, it doesn’t matter if the weather data is generated by a service in the back or if it is simply a pre-generated document. These details are something only the server should care about. Therefore, exposing application concepts as resources is the responsible thing to do, and it should not come as a surprise that this works well in combination with RDF, which is after all an acronym for *Resource Description Framework*.

The Semantics of Change

With services and data both being resources, there’s still a gap that needs to be bridged: what about state-changing operations? While data-providing services might be elegantly exposed as resources, data-modifying services might seem more difficult. However, that shouldn’t be that case. The HTTP uniform interface foresees different methods for manipulation [1], including PUT, POST, DELETE, and recently also PATCH. While the semantics of almost all methods are strongly constrained, POST involves a degree of freedom, as “[t]he actual function performed by the POST method is determined by the server” [1]. This essentially means that the protocol does not allow one to predict what the result of the action will be. While this is not a problem for humans, who can interpret out-of-band information, it is a complex task for machines, who need to somehow *understand* what it means to perform a POST request on a certain resource. This made us wonder how we could describe the semantics of change in a machine-interpretable way.

To achieve this, we created the description format RESTdesc [5, 7], the goals of which are two-fold:

Capturing functionality RESTdesc descriptions capture the functionality of an HTTP request by connecting the pre-conditions and post-conditions of a given action in a functional way. The key to this connection are variables and quantification over these variables, functionality which is not supported natively by RDF. Therefore, RESTdesc descriptions are expressed in Notation3 (N3), a superset of RDF. The benefit here is twofold. First, the semantics are *integrated* into the language, as opposed to the use of expression strings in RDF, which are not supported natively. Second, when RESTdesc descriptions are instantiated, they become regular RDF triples, which can be handled as usual.

Describing the request Additionally, RESTdesc aims to explain the request that needs to be made to achieve the action, *without* harming the hypermedia constraint [2]. RESTdesc descriptions are merely a guidance, an expectation, but the interaction is fully driven by hypermedia, inheriting all the benefits of the REST architectural style (such as independent evolution).

Design for easy discovery and composition

The fact that RESTdesc descriptions are native N3 citizens makes them interpretable by any N3 reasoner. This means that any reasoner is able to solve the problem of discovery (which of the descriptions match a given need) and composition (combining different descriptions to match a need). Composition experiments conducted with RESTdesc and the EYE reasoner [4] show that even compositions with complex dependency chains can be created in a few hundred milliseconds. This number is largely unaffected by the total number of present descriptions.

RESTdesc adds the missing piece of the puzzle to have a seamless integration between data and services. While the REST principles make a universal treatment in the form of resources possible (all of which can be described by regular semantic technologies such as RDF), RESTdesc bridges the gap by describing the functionality of state-changing operations, which are an important aspect of Web APIs. Examples of RESTdesc usage can be found on <http://restdesc.org/>, together with an explanation of reasoner-based composition. A recent use case is distributed affordance [4], RESTdesc-based generation of hypermedia controls.

References

- 1 R. T. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. IETF Standards Track, June 1999.
- 2 Roy T. Fielding. REST APIs must be hypertext-driven. *Untangled – Musings of Roy T. Fielding*, October 2008.
- 3 Roy T. Fielding and Richard N. Taylor. Principled design of the modern Web architecture. *Transactions on Internet Technology*, 2(2):115–150, May 2002.
- 4 Ruben Verborgh, Vincent Haerinck, Thomas Steiner, Davy Van Deursen, Sofie Van Hoecke, Jos De Roo, Rik Van de Walle, and Joaquim Gabarró Vallés. Functional composition of sensor Web APIs. In *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, November 2012.
- 5 Ruben Verborgh, Michael Hausenblas, Thomas Steiner, Erik Mannens, and Rik Van de Walle. Distributed affordance: An open-world assumption for hypermedia. In *Proceedings of the 4th International Workshop on RESTful Design*, May 2013.
- 6 Ruben Verborgh, Thomas Steiner, Davy Van Deursen, Sam Coppens, Joaquim Gabarró Vallés, and Rik Van de Walle. Functional descriptions as the bridge between hypermedia APIs and the Semantic Web. In *Proceedings of the 3rd International Workshop on RESTful Design*, pages 33–40. ACM, April 2012.
- 7 Ruben Verborgh, Thomas Steiner, Davy Van Deursen, Jos De Roo, Rik Van de Walle, and Joaquim Gabarró Vallés. Capturing the functionality of Web services with functional descriptions. *Multimedia Tools and Applications*, 64(2):365–387, May 2013.

3.25 Heterogeneous Information Integration and Mining

Raju Vatsavai (Oak Ridge National Laboratory, US)

License © Creative Commons BY 3.0 Unported license
© Raju Vatsavai

With the entry of private satellite corporations, the number of satellites operating in sun synchronous orbits has increased in recent years. As a result, today we are dealing with big heterogeneous data that is multi-resolution, multi-spectral, multi-sensor, and multi-temporal in nature. Multitude of these heterogeneous data products allows us to overcome information gaps arising due to environmental conditions (e.g., clouds during inclement weather conditions) and multi-temporal imagery allows us to monitor both natural and man-made critical infrastructure. However, analyzing these big heterogeneous data products poses several challenges. First, there are no good statistical models for heterogeneous data that allows accurate classification and change detection. Existing models are primarily designed for similar attributes (e.g., Gaussian Mixture Models). Second, derived data products (e.g., land-use/land-cover maps) do not follow any standard classification scheme. Though some of these products can be integrated using ontologies (at attribute level), spatial union of these data products is still an open research problem. Specific research problems are listed below.

- Statistical classification/clustering models for heterogeneous data (e.g., optical and synthetic aperture data; or continuous random variables and discrete/multinomial attributes)
- Fusion of thematic maps: Ontology drive spatial integration (both attributes and spatial joins)
- Model fusion:
 - Distributed data sources: How to construct a global modal from local models (derived independently)?
 - Heterogeneous data: How to fuse models generated independently on each sensor product?
 - Multi-temporal data: How to fuse models generated on each temporal instance?
- Data fusion and reduction methods that preserve object boundaries (e.g., liner relationships in feature space)

Acknowledgements

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

3.26 Semantics and Hypermedia Are Each Other's Solution

Ruben Verborgh (Ghent University – iMinds, BE)

License  Creative Commons BY 3.0 Unported license
© Ruben Verborgh

Joint work of Verborgh, Ruben; Steiner, Thomas; Mannens, Erik; Van de Walle, Rik

The Linked Data principles versus the rest constraints

Linked Data, oftentimes referred to as “*the Semantic Web done right*”, starts from four simple principles, as stated by Tim-Berners Lee [1]:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards.
4. Include links to other URIs, so that they can discover more things.

While these principles are starting to get known in the REST community as well, the main principles behind the REST architectural style [3] are mostly unknown to the semantic community. If you allow us to be creative with their phrasing and order, they are:

1. Any concept that might be the target of a hypertext reference has a resource identifier.
2. Use a generic interface (such as HTTP) for access and manipulation.
3. Resources are accessed through various representations, consisting of data and metadata.
4. Any hypermedia representation should contain controls that lead to next steps.

Since REST's flexible representation mechanism gives the freedom to publish information using standards, wouldn't you say that both sets of principles are actually pretty close? And considering the fact that the fourth Linked Data principle is the determining condition for the “Linked” adjective, and the fact that the corresponding fourth principle of REST indicates hypermedia controls as essential for REST APIs [2], doesn't it come to mind that both communities might actually be striving towards the same? Because it are exactly the *links* that give the data semantics and thus make it useful, and they're the same links that

drive REST APIs. After all, in a true REST API, you can perform all actions you need by following links, just like you do on the Web. This is called “*hypermedia as the engine of application state*”.

This insight leads us to conclude that the Linked Data principles are largely equivalent to the REST principles. In particular, we can see REST’s necessity of links as an operational variant of the fourth Linked Data principle. For data, links are used to create meaning; for applications, links are used to perform actions. In both cases, they are essential to discover the whole world starting from a single piece.

The tragedy of the missing link

The difference between both approaches is the impact of missing links. Within Linked Data, the condition is that you should link your data to *some* URIs. If you forget to link to a certain set of (meta-)data, it doesn’t matter all that much: the concept you are linking to might in turn link to the concept you forgot. Indeed, the linking concept is transitive, so the meaning a client is looking for can still be discovered. From the operational, REST point of view, things are different: if you are viewing a piece of information and you want to go to a certain place that is not linked, well... hypermedia gives up on you. It is then impossible to perform the desired action directly through the hypermedia document and you must do something else (like opening Google). This might only sound like a minor inconvenience, but it’s more: why do we have hypermedia if we can’t use its controls anyway, since they don’t bring us to the place we want?

This is exactly the problem we are trying to tackle in our latest research. The omission of the links you need as a user is only natural, because how can the server possibly know what next steps you want to take? So if the server does not provide the *affordance* [2] to go to the place you need, the client must add it. In our architecture and implementation for *distributed affordance* [4], we automatically generate the hyperlinks the user needs in a personalized way. These links are constructed based on semantic annotations in the page (*thus, Linked Data*), which are matched at runtime [5] to a user-selected set of services (REST). For instance, if you are reading a book review page, your browser can automatically generate links to borrow this book from your local library or to download it to your iPad. These links form personalized affordance for you, based on the content but selected according to your preferences.

So what happens here is that we connect two loose ends—the information on the one hand and the service on the other—solely based on semantics. This allows a loose coupling at design time (the information publisher does not need to know about the services you prefer) while having a strong coupling at runtime (the links directly let you use the information with the service). You could say that semantics complete the hypermedia engine: if the link is missing or does not exist, semantics can generate it. But let’s be fair and also say it the other way: hypermedia completes semantics, by offering the services that allow you to do something you need with the data you like.

References

- 1 Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data – The story so far. *International Journal On Semantic Web and Information Systems*, 5(3):1–22, 2009.
- 2 Roy T. Fielding. 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>.
- 3 Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, California, 2000.

- 4 Ruben Verborgh, Michael Hausenblas, Thomas Steiner, Erik Mannens, and Rik Van de Walle. Distributed affordance: An open-world assumption for hypermedia. In *Proceedings of the 4th International Workshop on RESTful Design*, May 2013. <http://distributedaffordance.org/publications/ws-rest2013.pdf>.
- 5 Ruben Verborgh, Thomas Steiner, Davy Van Deursen, Sam Coppens, Joaquim Gabarró Vallés, and Rik Van de Walle. Functional descriptions as the bridge between hypermedia APIs and the Semantic Web. In *Proceedings of the 3rd International Workshop on RESTful Design*, pages 33–40. ACM, April 2012. <http://www.ws-rest.org/2012/proc/a5-9-verborgh.pdf>.

4 Working Groups

4.1 Future of Actions

Réne Schubotz

License © Creative Commons BY 3.0 Unported license
© Réne Schubotz

Joint work of Valeria Fionda, Claudio Gutierrez, Armin Haller, Andreas Harth, Axel Polleres, Réne Schubotz, Ruben Verborgh

What if actions have their own URIs on the Web, are they getting executed every time you access the URI? Currently you can invoke actions via resource-centric web technologies and standards, however, you want to know beforehand what the resource is doing for you. Our assumption is that a standard solution for actions on the Web should include the following: Execution should not be possible by GET, you need to get a representation of the action, maybe get a redirect, have typed URIs (for results, execution) and content negotiation.

Our proposal is that we can use a standard protocol inspired by SPARQL endpoints, but to allow additionally to send the results to some other place, which is specified in the protocol and send with the query. This protocol can have a process model included that can delegate actions. Consequently, when you have a lot of data, you can define a URI where the data is stored, or some description of the data, rather than the direct request/response style that you have currently in the SPARQL protocol. As such resources are tasks, and the actions defined in the protocol can be part of a workflow. For example, if I order a flower, you post a job/action and a response URI and you can check the new resource that was created as part of your action to check if the flower was delivered to whomever you specified. We still need a process model for this protocol and the issue of notification is open. We also discussed that a publish/subscribe mechanism could potentially be implemented over this protocol.

4.2 Data Profiling

Felix Naumann

License © Creative Commons BY 3.0 Unported license
© Felix Naumann

Joint work of Felix Bießmann, Christian Bizer, Kjetil Kjernsmo, Andrea Maurino, Felix Naumann

Data profiling is the process of extracting metadata from a given dataset. Application areas for data profiling include data mining, visualization, schema reengineering and query

formulation, query optimization, data integration and cleansing, data quality assessment, and data source discovery and description. In summary, profiling is an important preparation step for any other data-intensive task.

While traditional data profiling considers single relational databases or even only single tables, data profiling for data ecosystems broadens the profiling scope and tasks significantly. Data profiling can and should be applied at every level: Entire data spaces (many and diverse sources), individual data domains (many sources from single topic), data sources (many tables and classes), individual data classes (such as persons, places, etc.), and finally data properties (such as name, address, size, etc.).

The main challenges of data profiling for large data ecosystems are the definition and specification of which metadata to extract at which level, the aggregation of metadata across levels, and of course the actual computation of the metadata by analyzing the sometimes very large datasets.

4.3 Enterprise Semantic Web

Melanie Herschel

License  Creative Commons BY 3.0 Unported license
© Melanie Herschel

Joint work of Stefan Desb loch, Melanie Herschel, Bernhard Mitschang, Giovanni Tummarello

In a wide variety of applications, ranging from reporting to complex business analytics, enterprises traditionally rely on systems processing relational data efficiently and at scale. However, the highly structured nature of such data makes them difficult to link and integrate. As Semantic Web technologies, such as linked RDF data and means to efficiently query and process these become available, an interesting question is how these technologies may impact on Enterprise Data Management.

In studying the question, the first observation is that RDF data by itself is interesting in the Enterprise context, as such data bears information that may not be available in other sources. So, similarly to other types of data sources (XML, Excel, CSV), RDF data can be considered as one among many interesting data sources that will however be processed in a similar way as other types of data. Therefore, the remainder of the discussion focused on where in the process of integrating, manipulating, and analyzing data Semantic Web technologies may apply in the Enterprise context.

We identified two domains of particular interest to an Enterprise that may benefit from the Semantic Web, i.e., (1) Knowledge Representation for documentation or traceability purposes and (2) gaining Knowledge from (Big) Data. We decided to further elaborate on this second aspect, where the main advantage of Semantic Web is the fact that it may render the time-consuming design and maintenance of a global schema and complex schema mappings unnecessary. One avenue for future research is however to study the price of this simplification, susceptible to be paid at a different point in the process. For instance, relational query engines take advantage of the rigid structure of data to efficiently process large volumes of data, so we raise the question whether we can leverage the maturity of these systems while benefitting from the advantages of RDF data. More specifically, is there a hybrid data model taking the best of both worlds and can we design query-languages (and execution engines) that can seamlessly and efficiently deal with both types of data?

In moving from a highly structured type of data to (Linked) RDF data, we also observe the need to shift from the classical extract-transform-load paradigm used in data warehouses

to an extract-explore-analyze paradigm. Here, more mature tools for data profiling, browsing, visualization, etc. need to be developed before Enterprises switch to this paradigm. Also, Enterprises will only be willing to move to this new paradigm if they can be convinced of technological benefits in terms of business relevant key performance indicators such as return on investment or total cost of ownership. Therefore, we believe that scientific evaluation should take these into account in the Enterprise context.

4.4 Planning and Workflows

Ulf Leser

License © Creative Commons BY 3.0 Unported license
© Ulf Leser

Joint work of Stefan Decker, Yolanda Gil, Melanie Herschel, Ulf Leser

The group discusses the impact of techniques from planning from first principles for scientific workflows. There was a general concern that such methods require a formal representation of properties of tasks, starting points and the targeted data product, which often are not available and also hard to maintain because tools and methods change very rapidly. The group then identified workflow repositories and workflow similarity search as another way to aid developers in designing workflows. Instead of generating workflows de-novo from abstract specifications, the idea here is to use similarity searches in workflow repositories to identify existing, proven workflows that solve the task at hand. If no perfect match is found, functionally similar workflows might help as starting point for adaptations. Workflow similarity search is a field with many existing and timely research questions, such as proper (semantic) similarity measures, methods for similarity-based workflow exploration and auto-completion, algorithms for searching across different workflow models, or workflow mining.

4.5 Linked Data Querying

Aidan Hogan

License © Creative Commons BY 3.0 Unported license
© Aidan Hogan

Joint work of Valeria Fionda, Katja Hose, Ulf Leser, Giuseppe Pirrò, Kai-Uwe Sattler, Steffen Staab

The ultimate destination of Linked Data is a decentralised eco-system of information spread over the Web. Following some Web standards (RDF, SPARQL, OWL, HTTP, URIs) and generic guidelines (dereference able URIs, provide links), publishers act independently when contributing to this information space. Aside from choosing a standard data-model, only loose co-ordination exists between publishers with respect to how data should be described, linked and made available. Although hundreds of Linked Datasets are now officially available on the Web – together comprising of billions of triples – it is still unclear what infrastructure is needed to query the data.

Current data-access mechanisms/problems include:

Dereferenceable documents: As per the definition of Linked Data, the URIs that name things should return useful information about those things when they are looked up over HTTP. Typically, all local URIs (i.e., URIs under the authority of the Linked Data site) should be dereferenceable and should return all RDF triples where the URI is in the

subject position. Some datasets also provide all RDF triples where the URI is in the object position. However, HTTP lookups are costly and typical queries can require large numbers of such lookups. Politeness policies (artificial delays) must be implemented by the client to avoid DoS attacks on the data provider. Furthermore, data that are not dereferenceable are difficult to access (short of a full site crawl).

Data Dumps: Many sites make data dumps available for download. Such dumps are often “all or nothing” in that the client can choose to download all of the data or none. Such data access mechanisms are of use to clients such as warehouses or analytical applications that wish to process/index the entire dataset locally. No standard protocols exist for accessing site dumps, other than sparsely-available VoID descriptions. Minor updates to data often require re-loading the full dump.

SPARQL Endpoints: Many Linked Data sites make SPARQL endpoints publicly available on the Web, which allows arbitrary clients to issue complex queries over RDF data hosted by the server. However, such endpoints suffer from performance and reliability problems. SPARQL is a complex query language (evaluation of which is PSpace-complete) and queries can be expensive to compute. Thus endpoints often time-out, return partial results, or fail under heavy loads.

During discussions at the Dagstuhl Seminar, we observed the following issues with respect to querying Linked Data:

- Protocols to find the overall “schema” of the data provided in a Linked Dataset are not available. Class and property URIs can be dereferenced individually, but the result is a collection of vocabulary definitions, not a structural overview of the data (cf. data profiling discussion).
- Creating structured queries is difficult since the client may not be familiar with the contents/vocabulary/structure of the dataset.
- Although Linked Data-providers adopt a common data structure, provide links and share some vocabulary definitions, this only enables a coarse form of client-side data integration: the integration problem is far from “solved”.
- RDF stores are no longer so naive with respect to query optimisation. Various works using database optimisation techniques have been published in the (database) literature. Various benchmarks, competitions and commercial engines have also emerged. Some engines rely heavily on compression/in-memory techniques. Schemes for sharding/partitioning RDF datasets also exist, where indexes over billions, tens of billions and hundreds of billions of triples have been claimed. In summary: problems of querying Linked Data not solely attributable to naive/poorly designed RDF stores.
- Making SPARQL endpoints publicly available breaks new ground. In the database world, back-end SQL engines are rarely/never made open. Rather services and limited query interfaces (e.g., RESTful interfaces) are built on top.

Looking to the future of Linked Data Querying, we identified the following directions:

- SPARQL endpoints require excellent cost-model predictions to know a priori if they can service a query adequately or not, and to do the required load-balancing of requests from multitudinous/arbitrary Web clients. Without this, Quality-of-Service guarantees and load scheduling become very difficult.
- There is a clear trade-off between SPARQL endpoints, which push all of the processing on to the server, versus Dereferenceable Documents/Site Dumps, which push all of the processing on to the client. There is nothing “in the middle”. A simplified RDF query

- language, perhaps expressed as a RESTful service, would perhaps strike a happy medium (working title: “GoldiLODs”). One idea was a query language that disabled join processing, only allowing atomic triple lookups and perhaps basic filtering and pagination mechanisms. This would allow the client to get a focused set (or sets) of data for local processing in a much more “controlled” fashion than traditional dereferencing. Furthermore, cost models, load balancing, hosting, etc. would be greatly simplified for the server, enabling a more reliable service than a SPARQL endpoint.
- Many expensive queries issued to SPARQL endpoints are analytical queries that require processing a large subset (or all) of the indexed data. It is not sustainable for the server to incur the costs of such queries. Languages for expressing analytical needs over RDF are still missing. One possibility is to create a procedural language designed for RDF analytics (perhaps allowing declarative SPARQL queries to be embedded).
 - Guides to help clients create queries are also of importance. The RDFS and OWL vocabulary descriptions are only superficially helpful in the task of query generation. Other methods to explore the structure of the data on a high-level are needed to understand how queries are best formulated (i.e., with respect to which vocabulary elements to use, how joins are expressed, what sub-queries will lead to empty results, etc.).
 - There is still no standard mechanism for full-text search over the textual content contained within RDF literals (SPARQL has REGEX but this is applied as a post-filtering operator, not a lookup operator).

4.6 Semantics of Data and Services

Sheila McIlraith

License © Creative Commons BY 3.0 Unported license
© Sheila McIlraith

Joint work of John Domingue, Craig Knoblock, Sheila McIlraith, Giuseppe Pirrò, Raju Vatsavai

The objective of this breakout session was to address issues related to the semantics of data and services. Of particular concern was a means of defining the semantics of linked data. The primary motivation for addressing this topic is to facilitate the composability and interoperation of data with data, and data with services. We believe that some degree of semantic description for data and services is necessary to successful integration and interoperation of data and services. A second motivation for semantic descriptions of data was so that data sets could be suitably annotated, archived and found. This was viewed as being increasingly important for scientific work.

There has been substantial previous work on the topic of semantics for services in support of Semantic Web Services (SWS), including W3C proposals and recommendations such as OWL-S, WSMO, SWSO, SAWSDL, and most recently the Linked Data based WSMO-Lite, MicroWSMO, the Minimal Service Model and Linked-USDL based on SAP’s Unified Service Description Language, as well as other efforts. The first three efforts, while differing slightly in their vocabulary and ontology formalism (OWL-S is in OWL, WSMO in WSML and SWSO is in first-order logic), each of these ontologies for services provides what we believe to be an adequate, or close to adequate, description of services. However, these early efforts results in complicated descriptions that are rarely created in full form and more recent work (e.g., iServe & Karma) has focused on simpler more streamlined source descriptions that are not as rich, but can be created automatically. One deficit we discussed was with respect to the description of the diverse forms of data that we may wish to integrate or interoperate

with. This includes a diversity of structured, semi-structured, unstructured and streaming data, and possibly the query engines that are employed to query that data. It was felt that further consideration needs to be given to a suitable description of such data.

There has also been significant previous work on tools to assist in the creation of specific service and data semantic annotations. These include SOWER for WSDL based services, SWEET for annotating Web APIs, and most recently Karma, which uses machine learning techniques to semi-automatically build source descriptions. There has also been significant previous work, too numerous to list, on the development of tools that exploit semantics of services and data. Some highlights include the SWS brokers WSMX and IRS-III, the Linked Data based semantic repository iServe (which is now part of the Linked Open Data Cloud), and the many OWL-S based tools. Finally, it was noted that Semantic Web Services has had broad influence in the development of tools by major corporations, including but not limited to IBM, SAP, and Apple through Siri.

Just as was the case with Semantic Web Services, the scope of the semantics is best circumscribed by what is needed to enable various applications. The tasks that informed the development of both OWL-S and WSMO included automated service discovery, invocation, composition, simulation, verification, mediation and execution monitoring. We felt these were also (at least) the tasks to be considered in any further effort.

Despite several SWS upper ontologies, few services and data contain annotations that describe their semantics. (That may be inaccurate/an overstatement.) As such, after a decade of SWS effort, it remains hard to find data and services. A number of service discovery tools have been developed in the past. The status of these tools needs to be explored, but it was predicted that they suffer from a lack of suitably annotated services and that the technology itself is still appropriate. We also felt there was a need for a data discovery engine. It was observed that it would be useful not only to search with respect to the topic of the data, but also with respect to its past use, and its provenance.

Challenges: The group identified the following challenges: 1. Defining a vocabulary/ontology for describing services/data. 2. Automatic methods for generating semantic descriptions of services. 3. How do you use Crowdsourcing/gamification to aid in the above. 4. How do we know if we're complete? Task dependent? 5. Calculating the domain and range of a service? 6. Need the relationship between the inputs and outputs. Easy to capture type information e.g. this service takes inputs of type A and then has outputs of type B. 7. Having descriptions which incorporate authentication is important as 80% of services require this e.g. on ProgrammableWeb. 8. Which representation language manage tradeoff between expressiveness/tractability. Existing Web standards. Do we need a KR at all? 9. Covering different types of data e.g. Linked Data and describing the semantics of streaming data. 10. How to link interpretations of data (e.g. satellite date) especially privacy preserving models. 11. Overcoming sampling problems if one can't access all the data.

4.7 Streams and REST

Daniela Nicklas

License © Creative Commons BY 3.0 Unported license
© Daniela Nicklas

Joint work of Stefan Deßloch, Bernhard Mitschang, Daniela Nicklas, Kai-Uwe Sattler, Réne Schubotz, Thomas Steiner

How can streams of data, coming from active data sources, be integrated in the REST architecture pattern? There are many work-arounds for this problem, ranging from repeating polls to “negotiations” over REST interfaces and streaming over other protocols. We came up with an approach that implements the common pub/sub mechanism for continuous queries, uses the well-known REST term (although with slightly adapted semantics), works with any stream query language, and can even integrate prosumers (combined producers and consumers of data, e.g., a mobile application that sends its location updates and gets back continuous query results that depend on its location). We believe that this approach nicely fits to the REST paradigm, and future work would be to implement it in a running prototype.

4.8 Clean Slate Semantic Web

Giovanni Tummarello

License © Creative Commons BY 3.0 Unported license
© Giovanni Tummarello

Joint work of Aidan Hogan, Katja Hose, Steffen Staab, Giovanni Tummarello

Linked (Open) Data has been a movement launched circa around 2007 which put emphasis on the publishing of data instances on the web, stressing the importance of a universal, agreed data retrieval mechanism as an enabler for the overall “web of data” or “semantic web vision”. While the objective of this initiative and the focus on a pragmatic connection between “data” and “existing web mechanism” is recognized as very positive, it is a fact that the initiative did not so far have much success, with the data growth apparently stopped, and well known issues in data availability, quality of the existing data – as well as lack of notable applications making use of it. The idea of the working group was to question the “Linked Data principles”, proposed originally and to ask ourself what could be other principles that could lead to in higher incentive for quality data publication and reuse. The group did by no mean have the time to investigate the issues thoroughly but made observed that, with some respect, Linked Data principles (basically dereferenceable URIs) is broken:

For “simple lookups” – questions and answer cases:

- It’s not useful for a client who doesn’t know where to start. Before asking information about anything one should know the URI but no mechanism is provided to ask for it. E.g. how to enter a URI about “Dagstuhl”?
- It was never defined what exactly a response should be in terms of completeness of a response: sometimes due to data modelling “the triples attached to something” is a very insufficient representation.

For use cases that would require a complete knowledge of the datasets e.g. “what is the biggest city you know of” or “give me all your movies” – crawl all/warehouse cases:

- There is no mechanism for this e.g. no connection to a SPARQL endpoint (and SPARQL would itself be a bad idea given how easy it is to ask questions which are too intense on the server).
- Even worse, there is no support for the ability to at least crawl the content of the site with a guarantee that one could get the whole dataset and then be able to ask the above questions. There is no mandatory sitemap in Linked Data and even if there was there would be no mechanism to retrieve data about URIs which are not on the same domain name. E.g. if a web site had this triple in its database `http://dbpedia/resource/Berlin isa city:NiceCity` it is unclear how would anyone be able to get this triple using the “Linked Data principles”.

We also questioned if the emphasis on URIs and the requirement for a “Linked Data” publisher to put links to other dataset is justified. Looking at reality would suggest that people – but most of all enterprises which are those who own the most important and often data rich web sites – would very seldom do anything that doesn’t give them immediate benefits and in general on the web one should seek decoupling and a link is a strong coupling. An alternative to suggesting to link, and even suggesting that anything should have a stable URI, could be that of stressing the importance of a good, comprehensive entity description, in a way that to humans and machine linking algorithms alike it would be easy to make a connection – a connection which could be created dynamically and according to the specifications of a task at hand (e.g. certain tasks might consider certain kind of equality vs other tasks which would require other kind, e.g. looser or more strict.). We called this “pointing based referencing approach vs model-theory approach to referencing”. We agreed however than when possible stable URIs are of course great and links are good for a client – given that for a lone client downloading a full site is out of the question.

We also mentioned that given big data and the power of search engines and web infrastructure one could easily see clients having some sort of back-end support to overcome these limitations, with great potential for useful functionalities (e.g. enter a website and immediately be suggested the most interesting pages on it by an external service who had previously crawled it all).

In the discussion, we hinted at use cases browsing, content management, data integration, enterprise data management, and stakeholders that might have a say in the determination of a new protocol to publish data (if found to be required): lay persons, scientists, developers, web site managers. For each of these one should consider the costs associated with any data publishing and data consumption methodologies and the rewards associated.

We discussed organizational issues and if it made sense to determine preferred centralized vocabulary but the consensus was these would emerge, or be de facto coordinated by important players, e.g. as in `schema.org`. We finally concluded with a very rough idea on how any new proposal should be evaluated: cost for stakeholders, fulfillment of use cases, compliance and natural match with existing web standards.

Participants

- Felix Bießmann
TU Berlin, DE
- Christian Bizer
Universität Mannheim, DE
- Stefan Decker
National University of Ireland – Galway, IE
- Stefan Dessloch
TU Kaiserslautern, DE
- John Domingue
The Open University – Milton Keynes, GB
- Valeria Fionda
Free Univ. of Bozen-Bolzano, IT
- Yolanda Gil
University of Southern California – Marina del Rey, US
- Claudio Gutierrez
University of Chile, CL
- Armin Haller
Australian National Univ., AU
- Andreas Harth
KIT – Karlsruhe Institute of Technology, DE
- Melanie Herschel
University of Paris South XI, FR
- Aidan Hogan
National University of Ireland – Galway, IE
- Katja Hose
Aalborg University, DK
- Martin Junghans
KIT – Karlsruhe Institute of Technology, DE
- Kjetil Kjernsmo
University of Oslo, NO
- Craig A. Knoblock
University of Southern California – Marina del Rey, US
- Ulf Leser
HU Berlin, DE
- Andrea Maurino
University of Milan-Bicocca, IT
- Sheila McIlraith
University of Toronto, CA
- Bernhard Mitschang
Universität Stuttgart, DE
- Felix Naumann
Hasso-Plattner-Institut – Potsdam, DE
- Daniela Nicklas
Universität Oldenburg, DE
- Giuseppe Pirrò
Free Univ. of Bozen-Bolzano, IT
- Axel Polleres
Siemens AG – Wien, AT
- Kai-Uwe Sattler
TU Ilmenau, DE
- Rene Schubotz
EADS – Ottobrunn, DE
- Steffen Staab
Universität Koblenz-Landau, DE
- Thomas Steiner
Google – Hamburg, DE
- Rudi Studer
KIT – Karlsruhe Institute of Technology, DE
- Giovanni Tummarello
National University of Ireland – Galway, IE
- Raju Vatsavai
Oak Ridge National Lab., US
- Ruben Verborgh
Ghent University, BE

