Valentina Ivanova Patrick Lambrix
Steffen Lohmann Catia Pesquita (Eds.)

VOILA! 2015

Proceedings of the International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data

Co-located with ISWC 2015
Bethlehem, Pennsylvania, USA, October 11, 2015

Title: Proceedings of the International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data (VOILA! 2015)

Editors: Valentina Ivanova, Patrick Lambrix, Steffen Lohmann, Catia Pesquita

ISSN: 1613-0073

CEUR Workshop Proceedings
(CEUR-WS.org)

Organizing Committee

Valentina Ivanova, Linköping University, Sweden
Patrick Lambrix, Linköping University, Sweden
Steffen Lohmann, University of Stuttgart, Germany
Catia Pesquita, University of Lisbon, Portugal

Program Committee

Benjamin Bach, Microsoft Research-Inria Joint Centre, France
Marius Brade, TU Dresden, Germany
Isabel F. Cruz, University of Illinois at Chicago, USA
Aba-Sah Dadzie, KMi, The Open University, UK
Aidan Delaney, University of Brighton, UK
Michael Granitzer, University of Passau, Germany
Anika Gross, University of Leipzig, Germany
Willem Robert van Hage, SynerScope B.V., Netherlands
Eero Hyvönen, Aalto University & University of Helsinki, Finland
Tomi Kauppinen, Aalto University, Finland
Suvodeep Mazumdar, University of Sheffield, UK
Enrico Motta, KMi, The Open University, UK
Paul Mulholland, KMi, The Open University, UK
Stefan Negru, MSD IT Global Innovation Center, Czech Republic
Silvio Peroni, University of Bologna & CNR-ISTC, Italy
Emmanuel Pietriga, INRIA Saclay, France
Mariano Rico, Technical University of Madrid, Spain
Harald Sack, HPI, University of Potsdam, Germany
Gem Stapleton, University of Brighton, UK
Vojtěch Svátek, University of Economics, Prague, Czech Republic

Preface

A picture is worth a thousand words, we often say, yet many areas are in demand of sophisticated visualization techniques, and the Semantic Web is not an exception. The size and complexity of ontologies and Linked Data in the Semantic Web constantly grow and the diverse backgrounds of the users and application areas multiply at the same time. Providing users with visual representations and intuitive user interfaces can significantly aid the understanding of the domains and knowledge represented by ontologies and Linked Data. There is no *one size fits all* solution but different use cases demand different visualization and interaction techniques. Ultimately, providing better user interfaces and visual representations will foster user engagement and likely lead to higher quality results in different applications employing ontologies and to the proliferation of Linked Data usage.

User interfaces are essential to easily provide access to the increasing diversity of knowledge modeled in ontologies. As ontologies grow in size and complexity, the demand for comprehensive visualization and sophisticated interaction also rises. In particular, user interfaces are an integral part of ontology engineering, to help bridge the gap between domain experts and ontology engineers. Ontology visualization is not a new topic and a number of approaches have become available in recent years, with some being already well-established, particularly in the field of ontology modeling. In other areas of ontology engineering, such as ontology alignment and debugging, although several tools have recently been developed, few provide a graphical user interface, not to mention navigational aids or comprehensive visualization techniques.

While ontology users usually possess domain and/or knowledge representation expertise, this is not necessarily the case with potential Linked Data consumers who can come from very different backgrounds and have varying levels of expertise. Currently, the main Linked Data consumers are technology experienced users, one of the reasons being the lack of appropriate user interfaces and visualizations to support other user groups. Visual approaches are needed to assist various kinds of users, who pursue diverse goals and pose individual requirements.

In the presence of a huge network of interconnected resources, one of the challenges faced by the Linked Data community is the visualization of the multidimensional datasets to provide for efficient overview, exploration and querying tasks, to mention just a few. With the focus shifting from a Web of Documents to a Web of Data, changes in the interaction paradigms are in demand as well. Novel approaches also need to take into consideration the technological challenges and opportunities given by new interaction contexts, ranging from mobile and touch interaction to visualizations on large displays, and encompassing highly responsive web applications.

The VOILA! workshop addressed these and related issues in its call for papers and attracted 18 submissions in different paper categories. Three reviewers were assigned to each submission. Based on their reviews we selected 12 contributions for presentation at

the workshop in the following categories: full research papers (5), position papers (2) and short papers (5).

The first edition of VOILA! is co-located with the 14th International Semantic Web Conference (ISWC 2015) and will take place as a full day event on October 11, 2015 in Bethlehem, Pennsylvania, USA. It will be organized around paper presentations and discussions and will be accompanied by interactive software demonstrations, giving developers a chance to gather feedback from the community.

We thank all authors for their submissions and all members of the VOILA! program committee for their useful reviews and comments. We are grateful to Miriam Fernandez and Krzysztof Janowicz, the ISWC workshop chairs, for their continuous support during the workshop organization. The workshop would not be possible without all of you!

September 2015

Valentina Ivanova,
Patrick Lambrix,
Steffen Lohmann,
Catia Pesquita

Contents

Visual Exploration of Formal Requirements for Data Science Demand Analysis

Aba-Sah Dadzie and John Domingue

KMi, The Open University
Milton Keynes, UK
{aba-sah.dadzie,john.domingue}@open.ac.uk

Abstract. The era of Big Data brings with it the need to develop new skills for managing this heterogenous, complex, large scale knowledge source, to extract its content for effective task completion and informed decision-making. Defining these skills and mapping them to demand is a first step in meeting this challenge. We discuss the outcomes of visual exploratory analysis of demand for Data Scientists in the EU, examining skill distribution across key industrial sectors and geolocation for two snapshots in time. Our aim is to translate the picture of skill capacity into a formal specification of user, task and data requirements for demand analysis. The knowledge thus obtained will be fed into the development of context-sensitive learning resources to fill the skill gaps recognised.

Keywords: big data, visual exploration, visual analytics, demand analysis, RtD, data-driven decision-making, ontology-guided design

1 Introduction

We are in the middle of a technological, economic, and social revolution. How we communicate, socialise, occupy our leisure time, learn and run a business has slowly moved online. In turn, the Web has entered our phones, our newspapers and notebooks, our homes and cities, and the industries that power the (digital) economy. The resulting explosion of data is transforming enterprise, government and society.

These developments have been associated with a number of trends of which the most prominent is *Big Data*. As noted recently by Google [1], what is important about data is not volume, but its contribution to innovation and, thereby, value creation. We agree with this assessment of the situation today: we are in the midst of a Data Driven Innovation (DDI) revolution. The benefits for DDI will be significant. Studies suggest that companies that adopt data driven decision-making have an output and productivity 5-6% higher than would be expected given their IT investments alone [5]. This assessment is backed up by Cisco, who report [4] that the Internet of Everything (the confluence of people, processes, data and things) will create $14.4 trillion in value globally through the combination of increased revenue and cost savings. McKinsey [15] makes similar predictions. Big Data has an estimated value of $610 billion across four sectors in the US (retail, manufacturing, healthcare and government services), with open data alone raising more than $3 trillion per year in seven key business sectors worldwide – education, transport, retail, electricity, oil & gas, healthcare, consumer finance [17].

According to a recent OECD report EU governments could reduce administrative costs by 15-20% by exploiting public data, equivalent to savings of €150-300 billion [5].

A major blocker for these promising prospects is the lack of *Data Science* skills within the workforce, be that technical specialists, managers or public servants. A well-known McKinsey study [16] estimated in 2011 that the US would soon require 60% more graduates able to handle large amounts of data effectively as part of daily work. With an economy of comparable size (by GDP) and similar growth prospects, Europe will most likely be confronted with a talent shortage of hundreds of thousands of qualified data scientists, and an even greater need for executives and support staff with basic data literacy. The number of job descriptions and increasing demand in higher-education programmes and professional training confirm this trend,[1] with some EU countries forecasting an increase of over 100% in demand for data science positions in less than a decade.[2] For example, a recent report by e-Skills UK/SAS [7] notes a tenfold rise over the past five years in demand for Big Data staff in the UK, and a 41% increase in the number of such jobs posted during the 12-month period from Oct 2013–Oct 2014, with over 21,000 vacancies in 2013. The study also estimated that 77% of Big Data roles were "hard-to-fill" and forecast a 160% increase in demand for Big Data specialists from 2013–2020, to 346,000 new jobs. Similar trends may be extrapolated to other EU countries. The European Commission's (EC) communication on '*Towards a thriving data-driven economy*' highlights an "*adequate skills base is a necessary condition of a successful data driven economy*"[3], while the Strategic Research and Innovation Agenda (SRIA) for the Big Data Value contractual Public-Private Partnership (cPPP) lists Big Data skills development as their top non-technical priority.[4]

The *European Data Science Academy* (EDSA)[5] is a new EU-funded project which will deliver the learning tools that are crucially needed to close this problematic skills gap. EDSA will implement cross-platform, multilingual data science curricula which will play a major role in the development of the next generation of European data practitioners. To meet this ambitious goal, the project will constantly monitor trends and developments in the European industrial landscape and worldwide, and deliver learning resources and professional training that meets the present and future demands of data value chain actors across countries and vertical sectors. Thus, a core part of our work is focused on demand analysis. We need to ensure that the data science curricula and associated learning resources that we create meet the needs of industry across Europe, recognising that this will vary by sector, job role and geographical region. In this paper we describe some of the visual tools we are developing to support our *Demand Analysis*. Through our visual analysis, we aim ultimately to make visible to a wider audience data we are collecting through a combination of interviews with key stakeholders, online surveys and data mining of job websites.

We continue in section 2 with a discussion of related work. We then describe in section 3 the methodology we are following, through data exploration (detailed in sec-

[1] Government calls for more data scientists in the UK: `http://bit.ly/1RLztP8`
[2] Demand for big data IT workers to double by 2017 ... `http://bit.ly/1Ntwcm8`
[3] Communication on data-driven economy: `http://bit.ly/1pNmxQq`
[4] SRIA on Big Data Value for Europe: `http://bit.ly/1LDSR1C`
[5] European Data Science Academy: `http://edsa-project.eu`

tion 4), to uncover design and task requirements. We discuss our findings in section 4.1 and feed these into the definition, in section 5, of target users, typical user tasks and the data necessary for users to meet their goals. We conclude in section 6 with pointers to the next stage in our study. We envisage, through this process in which we use dynamic, living data to guide our investigation, to identify intuitive, expressive approaches that will aid our analysis and serve as pointers to our targets, mainly Data Scientists, to a picture of capability and demand for their skills in today's data driven economy.

2 Related Work

Ontologies provide a useful framework for capturing, sharing and guiding (re)use of knowledge about an object, a domain or a situation [9,13,18,22]. Devedzic [6] in an early paper forecast the utility of ontologies for the Semantic Web and to improve competition in industry. The survey ([6]) illustrates how knowledge modeling and acquisition using ontologies aids collaboration and interoperation within and across disciplines, by providing standardised references to, and, therefore, interpretation of knowledge extracted from independent sources. The ESCO (European Skills, Competences, Qualifications and Occupations) vocabulary [8], for instance, was built to reduce recognised mismatch between demand in employment sectors across the EU and expertise in the current and future workforce. ESCO aims to help reduce unemployment by matching also to up-to-date training for each market. A final version is to be released in 2017 as Linked Open Data (LOD) to increase reusability in, e.g., statistical and demand analysis.

Ontologies may also be used to directly influence design, development and use of technology [13,22]. Grimm *et al.,* [9] describe their use to guide design choices during software development, to generate metadata about design and other intermediate artefacts created during the software development process and, therefore, improve communication between developers. Paulheim *et al.,* [18] survey work carried out to enhance capability in employing user interfaces (UIs) for specified tasks using ontologies, e.g., for filtering, clustering, visualising and navigating through information, as well as customising the UI itself for a task, user type or the user's environment. In our case, we aim to employ ontologies to guide: (i) knowledge capture – about demand for Data Science skills and capability to meet this demand, (ii) (re)use of this knowledge for context-driven, analytical and decision-making support, (iii) through an interface that supports context- and user-centred exploration and extraction of information about skill gaps and training resources for plugging them.

Visual analytics provides an intuitive, interactive approach for extracting task-based knowledge from complex data such as in our use case [12]. Both visualisation technique and how it is applied influence where visual and cognitive attention are directed and how data content is perceived and interpreted [10,11]. Especially for abstract, dynamic, large scale, multi-dimensional data, therefore, it is useful to provide alternative perspectives on the same dataset. These, used in isolation or in concert, allow different patterns and relationships to be revealed, triggering insight and resulting in more comprehensive exploration. Further, integrating (highly advanced) human perception into the analytics loop, to guide data processing, analysis and visualisation widens the

scope for exploration and increases confidence in decision-making based on the results obtained [10,12].

Reusable, extensible libraries and APIs for already proven visualisation and analytics techniques are particularly useful in such cases, as they ease development of and interaction with visual analytics tools [3,12], allowing a focus on research into novel solutions and their application and evaluation. However, identifying the optimal technique(s) for a task is influenced by a variety of factors, including user skill, data and domain, the task itself and whether and what subsequent use will be made of the results [10,11,12]. Taxonomies and ontologies play a useful role here by providing a formalism for specifying requirements and translating them into design. They may be used to document best practice for specific use cases, thereby providing context-based design guidelines [13,14,18]. We aim to harness ontologies to drive and document our design activities, to guide the development of an intuitive, reusable, extensible solution that serves the user's particular and evolving needs and context.

3 Methodology

Keim *et al.,* [12] extend the "visual information seeking mantra" [20] by placing first analysis of the data and/or situation, before an overview that highlights salient information, followed by exploration and further, detailed analysis of regions of interest (ROIs). In line with user-centred design (UCD) principles and recommended practice in visual analytics [10,11,12], we must ensure an intuitive UI and interaction methods that allow a focus on user tasks rather than the tool or its interface. We follow the principles of research through design (RtD) [2,19], using the process of exploring the knowledge and design space, during iterative data exploration, to probe initial and reveal additional requirements – see Fig. 1.

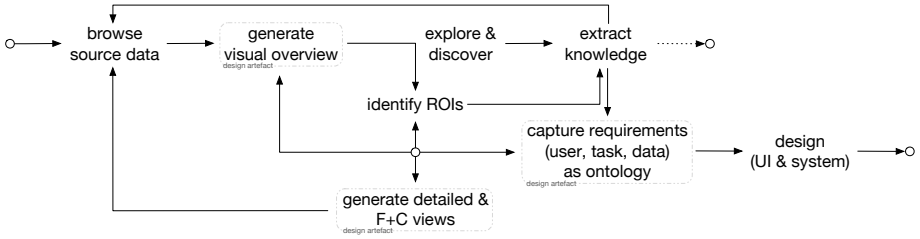


Fig. 1. Methodology followed in exploratory study, highlighting (with a faint border) where design artefacts are generated.

This study bears some similarity to [11]; however, we do not seek to formalise design patterns for composite visualisations. We aim, from the knowledge exploration exercise detailed in sections 4 and 5, to identify a range of intuitive visual analytics options, and as a result design that will guide customisation for use, individually and in concert, to meet the needs of different user types and tasks within the scope of demand analysis as described in sections 1 and 2.

4 Initial, Visual Exploration of Demand Data

We present a case study in demand analysis: investigating capacity against capability in key industrial sectors within the EU for expertise in Data Science. The dataset used comprises summary data crawled from LinkedIn, the first of a number of target domain expert networks and web services advertising job postings. Search terms are core and specialised skills grouped into seven skillsets (e.g., as listed in Fig. 2), identified through interviews with policy and decision-makers in industry across the EU and a focus group in the UK. Each skillset is translated into the official or dominant working language (37 in all) in each of 47 European countries. Due to restrictions to data extraction on LinkedIn only term frequency in job adverts is currently extracted, collected daily across three top-level dimensions: (i) industrial sector (captured as skill), (ii) geolocation and corresponding language and (iii) time. (Term selection and the data collection process is detailed in ([21].) While the dataset size is currently relatively small, complexity is introduced by large differences in scale and cross-linking within it. A key requirement is therefore scalability, to handle growth in size with time, and also complexity as additional context is mined from LinkedIn and other relevant sources.

We focus in this study on the temporal element in the data, treating the spatial attribute (geolocation and, by derivation, language) as an additional lens (filter) through which we examine the non-spatial data (skillsets). The aim here is two-fold: to identify effective methods for revealing, first, temporal patterns in the data, and then relating these to our target audience, using techniques that speak to them [10,12]. Another key requirement is therefore learnability, and to a point, customisability. The second goal is, in the process of exploring the data, to clarify our target end user characteristics and identify key tasks users would expect to be able to perform. We expect as a result, also, to identify additional data requirements (structure and content) for completing these tasks.

This necessitates data exploration from different perspectives, to identify where and how insight is triggered and which views reveal ROIs and answer key questions. While our overall goal includes the exploration of novel (visual) analytics approaches, this exploratory exercise focused on obtaining an initial, broad picture of demand and the identification of ROIs – anomalies, peaks and islands – within the data. We therefore made use of web-based solutions able to support quick prototyping of simple, yet informative overviews. A number of research prototypes and working visualisation tools, graphics libraries and APIs exist, implementing one or more of a range of visual analysis techniques (examples can be found in [10,11,12,14,20]). These include (not considering 3D for practical reasons): for high-dimensional data – parallel coordinates and small multiples (e.g., scatterplot matrices); techniques useful for temporal or dynamic data such as timelines and theme rivers; cartographic or geographical plots; statistical charts (e.g., line and scatter plots, bar and pie charts); and finally, techniques typically applied to non-spatial data such as word maps, tree and node-link graphs, and space-filling techniques such as tree maps and sunbursts. Freeware and open source tools such as *p5.js*, *Cytoscape.js*, *Raphaël*, *D3.js* and *Leaflet DVF* vary with regard to scalability, stability, author support, user community and compliance with web standards. Tools backed by commercial organisations, such as *Visual.ly*, *Tableau Public*, *IBM Watson Analytics* and *Google Charts* typically make available a limited set of features as free to

use and/or open source, often with restricted licenses. Such services may also require data upload to company servers.

For this exercise we use *D3.js*, a relatively well-established JavaScript library developed for "data-driven", interactive visualisation [3]. D3.js was built to overcome challenges encountered by its authors using existing web-based libraries, due to, among others, reliance on custom features with inconsistent compliance with web standards and browsers, or with high complexity. (Server-side) data input and initial, basic parsing was carried out with PHP, reading from the demand data written to CSV. The visualisations have been tested in Firefox, Chrome and Safari. It should be noted that not all events are triggered in all browsers, e.g., onChange in drop-down lists. We report our findings, in section 4.1, from the first three visual analytics techniques we employed:[6]

(i) ***line*** and ***dot timeline plots***, to obtain an overview of trends and variation in demand (patterns) over time;
(ii) ***small multiples***, employing a ***matrix plot***, to compare variation in patterns in attributes of interest for each data point and the whole dataset;
(iii) ***aster plots***, to examine skill demand by location.

We then discuss, in section 5, formal requirements specification and the implications for design for intuitive, interactive demand analysis and decision-making.

4.1 Findings

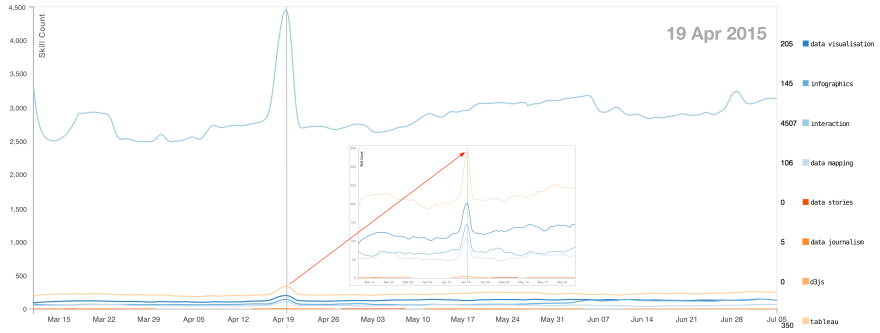
Summary statistical analysis showed some skew[7] with counts several times higher across all skillsets for *language == 'gb'* (English) and also for the UK only. However, further investigation showed more uniform relative distribution across countries. We therefore normalise the data or exclude the UK or all 'gb' countries where necessary to reveal trends suppressed in the remaining data.

Fig. 2 shows the trend in three *timeline* plots from the start of the collection period, 11 Mar 2015, to 05 Jul 2015 for demand for the skillset *visualisation* for 'gb' countries. There is a small peak early in the plot, and a sharp rise from 17th Apr to the 20th, peaking on the 19th. Beyond this there is in general a gradual rise for the rest of the period, with a few small dips. Trends are similar across all skills but *D3.js*, which records no counts till 11 Jun, rising to five on the 30th.

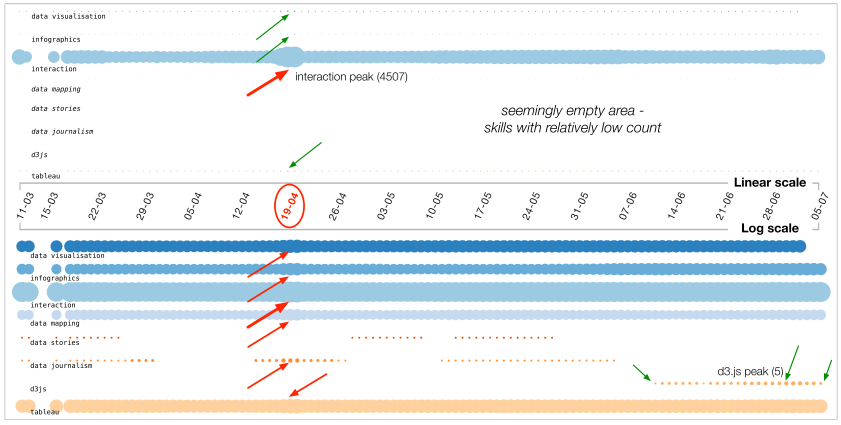
One skill, *interaction*, records counts more than 20 times greater than all others, suppressing, as a result, trends in the latter (see Fig. 2a). We looked at two options for revealing this detail. In the line chart hiding the outlier and modifying, correspondingly, the range of the y-axis, provides more space for the remaining attributes (see inset, Fig. 2a). We show also in Fig. 2b the second option: the bottom plot contrasts a $\log_n$ scale with the linear scales used in the other two plots. By stacking the journal plot with the linear scale on top of the normalised plot we obtain two gains. We are able to examine relative trend for each skill, still within the context of overall demand patterns, but with little increase in cognitive load.

[6] Additional, high-resolution snapshots (including video) at: `http://bit.ly/1FLevZE`

[7] The skew may be due in part to differences in terminology usage and interpretation across regions and/or the translator used. Further, the data collected to this point comes from a single (web) source. While we take this into consideration for further analyses, a full investigation of the cause of the skew is out of scope for this paper.



(a) A multi-line chart showing the demand trend for a selected skillset



(b) Normalisation using a $\log_n$ scale to reveal relative trend for each skill

Fig. 2. Daily demand for *'gb' visualisation*; large variation across skillsets for the four month period requires filtering (2a) and/or normalisation (lower plot, 2b) to reveal suppressed patterns.

We looked next at a data snapshot, taken in Apr 2014,[8] aggregating for five countries (excluding the UK), demand categorised under eleven key topics in Data Science (see [21] and bottom, right, Fig. 3). We use *small multiples* to examine multiple attributes simultaneously. Fig. 3 stacks, from left-right and top-bottom, mini plots showing in descending order counts for demand for each skill for each country, followed by skill and country percentage. Dynamic sliders are used to examine each skill as a percentage of the total demand for all countries (in the dataset, including the UK – olive inner border), and skill distribution within each country (faint blue outer border). The snapshot highlights values for skill and country percentage greater than 30 and 25% respectively. Blue borders are found predominantly at the top, showing top-heavy demand for selected skills. This is mirrored in the colour-coded line plot for the full dataset overlaid

[8] While the picture of demand continues to change the variables to be examined remain the same. Comparing earlier findings with current demand allows us to revisit the initial project requirements defined with respect to data structure and content.

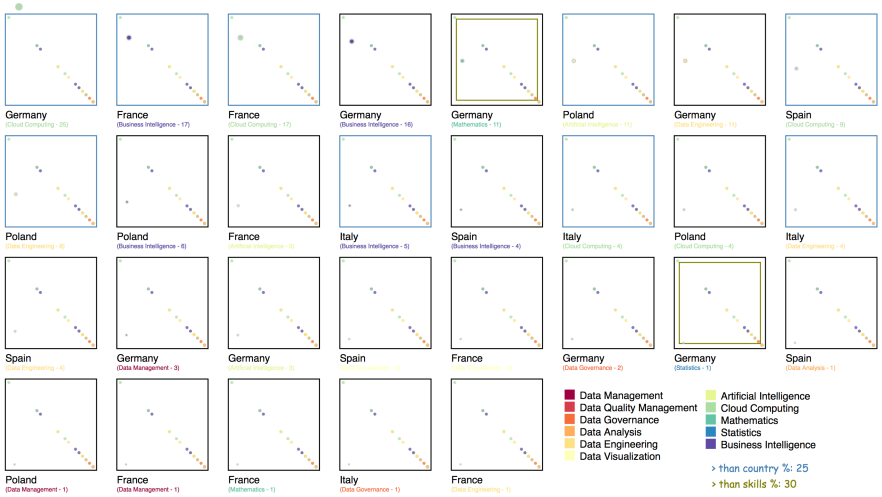


Fig. 3. Small multiples used with dynamic filters to investigate trends across three attributes – skill count and distribution (%) by country and skill percentage per country. Data for the UK, which is ∼70% of the total (see Fig. 4) is filtered out.

on each mini plot, which shows a long tail with very small counts per skill and country. Olive borders are more randomly distributed – e.g., Germany sees 50% of all *Statistics*, with a count of one, near the tail end of the chart (the other 50% (one) is in the UK).

We used, next, a space-filling technique, *aster plots*, a variant of a nested pie chart, to examine further skill distribution within each country, including the UK, using again the small multiples technique. Fig. 4 shows on the far left an overview of total demand for each country, then distribution by skill. Area maps to count for each slice. For the first aster, height maps to skill count per country (up to 11), and for all others skill percentage (over all countries). While the UK dominates all others, the individual country plots reveal a degree of similarity in skill distribution. *Business Intelligence*, *Data Engineering* and *Cloud Computing* are in demand across all, followed by *Artificial Intelligence*, which is highest in Poland. One skill, *Data Quality Management*, records one count in only one country, the UK – so slim that only by thickening the borders of each inner slice, to provide an additional visual cue, is it recognised. Here, we see the

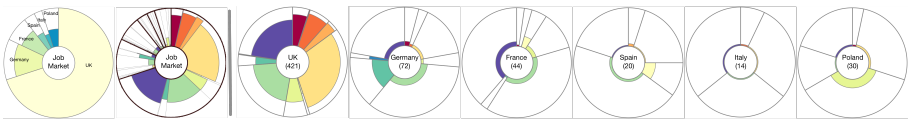


Fig. 4. The overview (first two from left) shows total demand per country and distribution of skills, respectively, for the six countries that follow, clockwise, from 12:00. As in Fig. 2, juxtaposing the two sets of charts enables *focus* on the detail for each country within the *context* of the overview (F+C). Colour coding is as in Fig. 3.

power in multiple perspectives – this is highlighted in the matrix plot (e.g., in Fig. 3) for the full dataset with the skills slider set to the maximum (100%).

5 Core Requirements for Demand Analysis

A key point reiterated throughout this exercise is the importance of letting the data drive our exploration of the knowledge and design space. By examining the questions raised as we explored this initial dataset, we have identified four areas that we must address if we are to meet our goals for demand analysis: (i) definition of target users and (ii) the tasks relevant to each; (iii) data, and therefore, knowledge requirements for effective task performance and completion, all of which lead to: (iv) effective support for decision-making. Fig. 1 shows three points at which we expect to generate design artefacts; we focus in this section on requirements specification, a living artefact that will evolve with the project. To allow use also as a communication tool with end users and within the project team, and to feed into the design, development and evaluation cycle, we aim to specify our requirements formally using an ontology. At this early stage we use simple structure diagrams, as in Fig. 5, to map this knowledge space. We document in the process, also, existing standards that we may reuse.

5.1 Target User Types

In line with our aim to match skill gaps in Data Science with context-sensitive training, at the start of the study we had two key targets identified: *Data Scientists*, practising

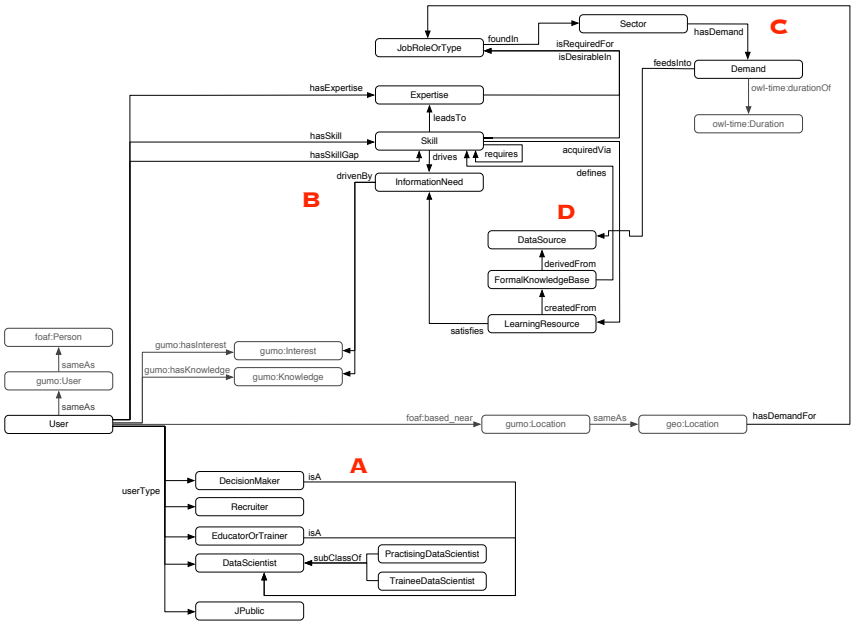


Fig. 5. High level definition of knowledge structure for design space

or new, and the institutions and personnel (*Educators* or *Trainers*) who will develop learning resources. In exploring the data we recognised that the picture of demand as extracted from recruitment sites such as LinkedIn may not adequately define the resources required to fill the gaps identified. A third user is critical in ensuring that training resources meet the needs of the market: the *Decision Maker* ultimately responsible for the definition of new roles and essential skills for them, who will also influence training of new and existing employees. Decision makers may be data scientists or other technology or domain experts. Finally, we recognise two additional user types. Recruiters may influence the language in and the interpretation of job adverts. In line with requirements for communicating the results of research to the (interested) public, we include the non-domain end user, who may or may not be a technology expert.

5.2 Task Identification and Requirements Analysis

Sections B and D of Fig. 5 focus on task requirements for data scientists, looking at what factors drive the acquisition of new skills and what is required to identify which skills are in demand and where. For clarity, Fig. 5 omits the detailed breakdown of skills as shown in, e.g., Fig. 2; the demand data will however be mapped along these categories (classes) to related information captured from, among others, LOD and OERs (Open Educational Resources). Our exploration raised three questions which impact the development of training resources: (1) which skills are typically required together? (2) how are skills ranked in and (3) how transferable are skills across sectors and regions? We must provide means for users to answer these and other questions, and also filter the data on different criteria, with a baseline for filtering on *sector*, *skill*, *location* and *time*, before matching to resources recognised to meet their needs. For technology experts (such as data scientists) functionality for complex query formulation should be useful for more complex analysis and knowledge retrieval.

5.3 Input Data Requirements

The summary data gives a broad overview of demand. However, our analysis is limited by the lack of context; while distinct patterns can be seen, what led to them cannot be ascertained. To complete the picture of demand we need to fill in the gaps resulting from content access restrictions on web and other services. We require detail that answers questions such as raised in section 5.2, and that may be used to enrich information extracted from other target sources. Data requirements include: (i) ideally, full text for role descriptions, containing, among others: (ii) term frequency per advert, (iii) weighting of terms, i.e. required vs. desired; and (iv) other detailed views on the sectors and skills of interest, from, e.g., related ontologies and vocabularies, LD and OERs.

With this additional detail we can begin to build a knowledge base (KB) that our target may draw from to make informed decisions, based on current market data and context-specific skills training. In section D of Fig. 5 we use a catch-all – *Data Source* – to represent both the data mined specifically for the project and other third party KBs. In the next stage of our study we will define more fully target KBs and how we will link to and reuse their content.

6 Discussion & Conclusions

Big Data presents a challenge for industry, due not just to its scale and the rate at which it continues to grow, but because its inherent heterogeneity and complexity present additional challenges for mining and reusing its valued content, value which contributes to gaining competitive advantage. Making effective use of Big Data starts with the specification of the skills required of the *Data Scientist*, for roles specific to and that span industrial sector and local context.

Our exploratory study has provided an initial picture of the demand for Data Science skills in key sectors across the EU. We have, in the process, uncovered questions with implications for the design of effective, intuitive knowledge exploration and analytical support for our target users. Demand is in turn sparse and large, within each and across skillsets. Relative distribution, conversely, is fairly uniform across location, but with instances where a specific skill is isolated to a small pocket. We must bear in mind, however, a key limitation in our study – the loss of context in the summary data. A second is that our current picture of demand comes from a single source, albeit extracted using search terms specified by a range of technology and domain experts. These impact the depth of analysis and the determination of optimal techniques for doing so, both for our requirements and those of our target users. We must therefore design for scalability, to manage continued increase in size and complexity and the potential for even greater variability. This demands alternative perspectives from which to examine data content and structure. Following the methodology used in this study, we will investigate additional approaches to ensure the generation of intuitive, informative overviews, with lenses for detailed analysis tailored to the user's particular context.

The knowledge structure summarised in Fig. 5 is a living document that will evolve as we obtain a more balanced and complete picture of demand and corresponding skill gaps. We have started to map the concepts and relationships defined so far to other related knowledge sources such as the ESCO vocabulary and new information obtained from further interviews with industry experts. This is to enable more detailed examination of the relationships between skills within and across skillsets and industry sectors. The aim is to refine our current skill definitions and map these to role descriptions. We will then be able to return to our target users, to review the formal, updated requirements and discuss further design for the analytical and decision-making tools they require.

The ultimate aim is to map the picture of demand to the user's specific requirements and feed the knowledge thus obtained into developing effective learning resources. This is to aid data scientists and decision-makers in industry and academia to identify optimal paths to acquiring and updating skills that meet the requirements for managing Big Data in the modern digital economy.

Acknowledgments. The work reported in this paper was funded by the EU project EDSA (EC no. 643937).

References

1. Andrade, P.L., Hemerly, J., Recalde, G., Ryan, P.: From Big Data to Big Social and Economic Opportunities: Which policies will lead to leveraging data-driven innovation's potential. In:

GITR 2014: The Global Information Technology Report 2014: Rewards and Risks of Big Data, pp. 81–86 (2014)
2. Bardzell, J., Bardzell, S., Koefoed Hansen, L.: Immodest proposals: Research through design and knowledge. In: CHI '15: 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 2093–2102 (2015)
3. Bostock, M., Ogievetsky, V., Heer, J.: D^3: Data-driven documents. IEEE Transactions on Visualization and Computer Graphics 17(12), 2301–2309 (2011)
4. Bradley, J., Barbier, J., Handler, D.: Embracing the Internet of everything to capture your share of $14.4 trillion. Tech. rep., Cisco (2013)
5. Brynjolfsson, E., Hitt, L.M., Kim, H.H.: Strength in numbers: How does data-driven decision-making affect firm performance? Social Science Research Network (2011)
6. Devedzic, V.: Knowledge modeling – state of the art. Integrated Computer-Aided Engineering 8(3), 257–281 (2001)
7. Big data analytics: Assessment of demand for labour and skills 2013–2020. Tech. rep., e-Skills UK/SAS (2014)
8. ESCO: European classification of skills/competences, qualifications and occupations: The first public release – a Europe 2020 initiative. Tech. rep., Luxembourg: Publications Office of the European Union (2013)
9. Grimm, S., Abecker, A., Vlker, J., Studer, R.: Ontologies and the Semantic Web. In: Handbook of Semantic Web Technologies, pp. 507–579. Springer (2011)
10. Heer, J., Bostock, M., Ogievetsky, V.: A tour through the visualization zoo. Communications of the ACM 53(6), 59–67 (2010)
11. Javed, W., Elmqvist, N.: Exploring the design space of composite visualization. In: PacificVis: 2012 IEEE Pacific Visualization Symposium. pp. 1–8 (2012)
12. Keim, D., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: Definition, process, and challenges. In: Information Visualization: Human-Centered Issues and Perspectives, pp. 154–175. Springer (2008)
13. Kitamura, Y.: Roles of ontologies of engineering artifacts for design knowledge modeling. In: Design Methods for Practice. The Design Society (2006)
14. Lohse, G.L., Biolsi, K., Walker, N., Rueter, H.H.: A classification of visual representations. Communications of the ACM 37(12), 36–49 (1994)
15. Lund, S., Manyika, J., Nyquist, S., Mendonca, L., Ramaswamy, S.: Game changers: Five opportunities for US growth and renewal. Tech. rep., McKinsey Global Institute (2013)
16. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. Tech. rep., McKinsey Global Institute (2011)
17. Manyika, J., Chui, M., Farrell, D., Kuiken, S.V., Groves, P., Doshi, E.A.: Open data: Unlocking innovation and performance with liquid information. Tech. rep., McKinsey Global Institute (2013)
18. Paulheim, H., Probst, F.: Ontology-enhanced user interfaces: A survey. International Journal on Semantic Web and Information Systems 6(2), 36–59 (2010)
19. Pierce, J.: On the presentation and production of design research artifacts in HCI. In: DIS '14: the Designing Interactive Systems Conference. pp. 735–744 (2014)
20. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: 1996 IEEE Symposium on Visual Languages. pp. 336–343 (1996)
21. Tarrant, D., Bullmore, S., Costello, M.: Deliverable D1.1: Study design document. Tech. rep., European Data Science Academy (EDSA), EC. project 643937 (2015)
22. Zlot, F., de Oliveira, K.M., Rocha, A.R.: Modeling task knowledge to support software development. In: SEKE '02: Proc., 14th International Conference on Software Engineering and Knowledge Engineering. pp. 35–42 (2002)

ProvenanceMatrix: A Visualization Tool for Multi-Taxonomy Alignments

Tuan Dang[1], Nico Franz[2], Bertram Ludäscher[3], and Angus Graeme Forbes[1]

[1]University of Illinois at Chicago, Chicago, IL, USA
[2]Arizona State University, Tempe, AZ, USA
[3]University of Illinois at Urbana-Champaign, IL, USA

Abstract. Visualizing and analyzing the relationships between taxonomic entities represented in multiple input classifications is both challenging and required due to recurrent new discoveries and inferences of taxa and their phylogenetic relationships. Despite the availability of numerous visualization techniques, the large size of hierarchical classifications and complex relations between taxonomic entities generated during a multi-taxonomy alignment process requires new visualizations. This paper introduces *ProvenanceMatrix*, a novel tool allowing end users (taxonomists, ecologists, phylogeneticists) to explore and comprehend the outcomes of taxonomic alignments. We illustrate the use of *ProvenanceMatrix* through examples using taxonomic classifications of various sizes, from a few to hundreds of taxonomic entities and hundreds of thousands of relationships.

Keywords: Taxonomic classification, multi-taxonomy alignment, phylogenetic relationship, matrix representation, glyph-based visualization

1 Introduction

Visualization tools developed for the field of biological taxonomy (herein broadly defined to include phylogenetics) may focus on representing the information content of one comprehensive classification, or provide visual information on the relationships between taxonomic entities represented in multiple, alternative classifications [9, 11]. The latter visualization services are useful in particular for illustrating important similarities and differences in taxonomic perspective, which may be empirically rooted in the discovery of new taxonomic entities, new evidence of phylogenetic relationship, or in the differential sampling and weighting of phylogenetic evidence [10]. Such multi-taxonomy comparisons can be viewed as a solution to the challenge of representing *taxonomic provenance* [11], i.e., linking a taxonomy T_1 to another (pre- or postceding) taxonomy T_2. To achieve this, taxonomic concepts endorsed by each alternative classification are individuated using taxonomic concept labels with the syntax: *taxonomic name sec.* (according to) *taxonomic source* [8]. Linkage of same-sourced concepts via parent-child (*is-a*) relationships permits the assembly of multiple independent classifications, and therefore presents new opportunities for inferring and visualizing taxonomic provenance across multiple classifications.

Here were describe *ProvenanceMatrix*, a novel tool for visualizing some of the knowledge products of EULER/X, a multi-taxonomy alignment toolkit [1]. EULER/X is

a logic-based reasoning software capable of aligning (or "merging") two or more taxonomic concept hierarchies, using different underlying inference mechanisms, in particular, answer sets [12] and qualitative reasoning using RCC-5 constraints [15]. The reasoning process models taxonomies T_1 and T_2 as sets of *is-a* constraints, together with a set A of expert-asserted input *articulations* that relate concepts in T_1 with those in T_2, typically at the leaf level. Using RCC-5 (Region Connection Calculus) relations, the expert can express through the articulations in A which relation holds between a concept $T_1.X$ and a concept $T_2.Y$, i.e., *equals*, *includes*, *is_included_in*, *overlaps*, or *disjoint*. If the precise relationship is not known, then one of the non-elementary $2^5 = 32$ disjunctive combinations of the 5 base relations can be used to express this uncertainty [17].

Fig. 1(a) shows two input taxonomies and the expert articulations. The alignment (or merge) result is depicted in Fig. 1(b). Further below (e.g., see Fig. 4), we propose to replace this view with a more dynamic *ProvenanceMatrix* view, juxtaposing concepts of T_1 (as rows) and of T_2 (as columns). In principle, we could also do this for the input data in Fig. 1(a), but it is primarily the alignment result in Fig. 1(b) that a user will want to visualize and explore.

The toolkit workflow iteratively guides the expert user towards identifying sets of input articulations that are both logically consistent and sufficiently specific to yield only a limited number of consistent alignments [2]. An important product of the alignment process is the set of *Maximally Informative Relations* (MIR): for any pair (C_1, C_2) of concepts from T_1, T_2, the MIR of (C_1, C_2) is the unique relation in the powerset lattice R_{32} over the RCC-5 base relations which implies all other relations that hold between C_1 and C_2, given T_1, T_2 and A.

The MIR play a critical role in generating the set of consistent alignments ("possible worlds"), in diagnosing undesired ambiguities in the input or output articulations, and generally in understanding the toolkit reasoning outcomes. Visualization tools are important in this context because the number of MIR for two taxonomies with m and n concepts, respectively, is $m \times n$. For instance, the alignment use case of *Primates sec. Groves (1993; T_1)* and *Primates sec. Groves (2005; T_2)* contains 317×483 taxonomic concepts and hence 153,111 MIR relations [11]. Displaying the MIR in list format is not an effective method for exploration. Instead users need dynamically rendered, scalable visualization solutions to navigate the large and semantically complex reasoning outcomes and adjust the input accordingly to achieve the desired alignments.

Key visualization challenges for multi-taxonomy alignment outcomes include the following scenarios. Frequently the alternative taxonomies have unequal sets of leaf-level children. For instance, recently published taxonomies may include new species-level concepts for which there are no corresponding entities in preceding classifications [9]. Additionally, the visualization must display large numbers of data points ($>$ 150,000 in the medium-sized Primate use case), where each point can be constituted by any subset of RCC-5 articulations in the R_{32} lattice. In order to empirically assess the reasoner-inferred articulations, users may also need to access taxonomic provenance information such as feature-based diagnoses, illustrations, and other taxonomic information.

Using *ProvenanceMatrix*, we can visualize alignments of large taxonomies with up to hundreds of concepts. Our technique uses matrix representation and glyphs in each cell to highlight RCC-5 articulation sets and alignments. In Section 4, we demonstrate how our technique effectively facilitates the exploration of multi-taxonomy alignments with varying sizes and levels of alignment ambiguity.

2 Related Work

An overview of the EULER/X multi-taxonomy alignment approach is provided in [9]. Fig. 1 shows the current visualizations of two related concept taxonomies, plus articulations among the respectively entailed taxonomic concepts. The aim is to visualize the input taxonomies T_1 and T_2 and the resulting merged visualizations (rendered with GraphViz [6]). In the figure, "==" means equals, "<" means is_included_in, ">" means includes, "><" means *overlaps*, and "|" means disjoint. The final product is a merged taxonomy (as depicted in Fig. 1(b)) that represents the concept-level similarities and differences among the aligned input trees. However, current GraphViz visualizations are not interactive and do not facilitate efficient exploration of ambiguous (under-specified) articulations which generate multiple possible world solutions. Resolving ambiguity is a critical aspect of the alignment process.

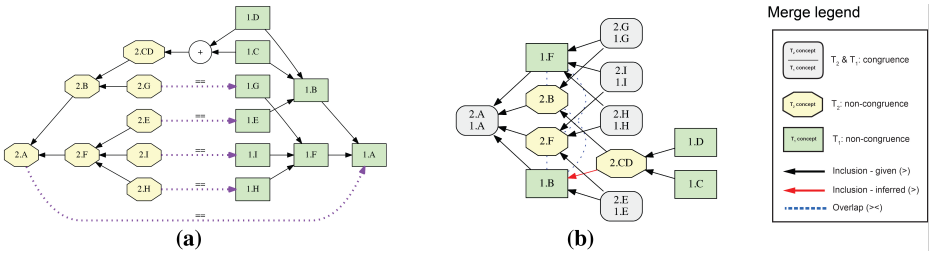


Fig. 1. Abstract toolkit input and output example rendered with GraphViz: (a) Representation of two input taxonomies T_1 (left) and T_2 (right) and articulations in the toolkit input file. (b) Single, consistent alignment of the input shown as a containment with overlap graph.

Tanglegrams are widely used in biology, for instance to represent the inferred evolutionary histories of rooted phylogenetic networks [16] and to highlight common structures as well as differences in multiple DNA sequences [18]. A tanglegram draws two rooted trees with the leaves opposing each other and uses auxiliary lines to connect matching taxonomic entities at the leaf-level. These auxiliary lines can be rendered in different styles or colors to encode different types of relationships (e.g., host-parasite associations).

The *Concept Relationship Editor* [3] extends the alignment process to support assertions of relationships between taxonomic classifications at all levels of each aligned hierarchy. *Concept Relationship Editor* adopts a space-filling adjacency layout which allows users to expand multiple lists of taxonomic concepts with common parents. The

lens mode and scroll mode are two different ways to navigate across the hierarchy of either classification while ensuring that the text strings in focus remain legible. Lines are used to connect the related taxa with symbols at either end to indicate the relationship type. Similar to tanglegrams, this technique can introduce visual clutter due to edge crossings as the number of taxa increases.

An alternative visualization approach utilizes icicle tree representations [14]. The RCC-5 relationships are colored bands to connect pairs of taxonomic concepts. Neighboring bands of the same color are bundles that reduce cognitive load. Spaces between concepts of one taxonomy may be used to better align the two trees and reduce crossed bans. In addition, nodes may be color-coded to indicate what percentage of a node's descendants are congruent or not. Figure 2 shows an example of the icicle tree representation. In the diagram shown, purple means *equals* or congruent (==), black means *is_included_in* or subset (<), blue means *overlaps* (><). However, this technique is only suitable for smaller numbers of concepts or aggregate views of large classifications. When we try this technique on a large number of taxonomic concepts, and especially when multiple articulations between paired concepts must be displayed, the visualization becomes cluttered due to band crossings.

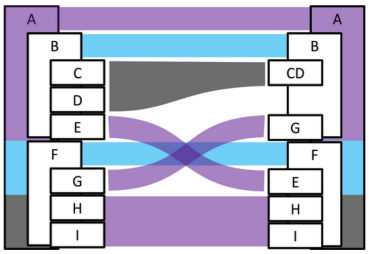


Fig. 2. An example of the icicle tree representation and colored bands to highlight articulations between pairs of taxonomic concepts. (credit Michael McGuffin)

3 Design Motivation

In this section, we review the primary challenges inherent in displaying RCC-5-based, multi-taxonomy alignments. Addressing these challenges has motivated us to create a new visualization technique that better supports the visual exploration tasks relevant to such taxonomic reasoning products.

The EULER/X input (constraint) and output (alignment) visualizations as depicted in Figure 1 present slightly different sets of challenges. They are currently produced by toolkit-native stylesheets that translate the user input and reasoner output into GraphViz-compatible data files. While there is some limited flexibility in tweaking the GraphViz output using EULER/X stylesheet options[1], the ranked graph layout computed by GraphViz may not reflect the user's intuitions regarding the spatial arrangement of concepts and relationships. For example, the ordering of children of a parent concept computed by GraphViz is different from the order in which they appear in the source publications for that taxonomy. This can create unintuitive experiences for the user.

Smaller scale visualization enhancement goals is improving usability for annotating/editing the GraphViz output data files. Larger scale goals entail acquiring the ability

[1] The stylesheet options result in different GraphViz attribute settings, e.g., "`constraint=false`" ignores certain edges for layout purposes.

to export/edit EULER/X visualizations in other (phylo-visualizing) platforms (however, a related challenge is that the most popular programs may not support EULER/X semantics which mandate the use of taxonomic concept labels, parent/child relationship [same taxonomy], RCC-5 relationships [across taxonomies], and merge concepts labels [AB, Ab, aB]).

Here, we provide an overview of some of the main visualization tasks for visualizing related concept taxonomies (hierarchies), as well as the relations between the concepts in the input taxonomies. Given two or more taxonomies:

T1. Focus on specific articulations.

T2. Provide different ways to organize hierarchies. This helps to compare structure of the input taxonomies.

T3. Highlight taxonomic concepts in one classification that stand in various specific and incongruent relations to concepts in the other classification.

T4. Find related concepts and subtrees of one taxonomic classification to the other taxonomic classification.

T5. Display details on demand. In particular, users want to be able to overlay distinctions between user-provided and toolkit-inferred articulations (i.e., articulation source), and display additional domain-relevant information (such as characters, images) when mousing over a concept label.

T6. Collapse and expand a subtree to simplify or fully explore a branch. This feature is particularly useful when dealing with large taxonomies.

4 Methods

Matrix representation is a useful tool for visualizing networks in many application domains, such as protein-protein biological interactions [5] and social networks [7]. This technique is superior to using node-link diagrams when the networks are dense, given that edge-crossings are the main limitation of node-link diagrams in visualizing these networks. A drawback of matrix representation is the inability to represent the flow of the networks [4]. However, since network flow is irrelevant in multi-taxonomy alignments, we found matrix representation to be best suited for visualizing the data products discussed above. Moreover, matrix representation enables the display of multiple (disjunct) relationships that may exist between a pair of elements from both dimensions in a matrix [5].

Figure 3 shows an example of *ProvenanceMatrix* for the *Perelleschus* classifications [9]. Each side of the matrix displays an input taxonomy. The arcs are used to indicate hierarchical information, directed from parent to subordinate child concepts. MatLink [13] also uses arcs to indicate relationships but only considers undirected networks. Moreover, the taxonomic concept labels are also indented appropriately to highlight hierarchical arrangement of each input classification. In each cell of the matrix we use circular sectors, divided similarly into a pie-chart, to indicate the articulations that hold true between two taxonomic concepts, where each sector (pie-slice) in the circle is given a color to consistently indicate the articulation type. The more pie-slices are shown, the less we know about the pair of concepts. Thus, a "full circle" (with all 5 pie-slices) means we know nothing about a relation. These "full circle" can act as "alerts"

to the user that the alignment is problematic (too ambiguous). Conversely, a single slice is the best case, specifying a unique (fully specified) relationship between two concepts. Color legend is depicted on the right of Figure 3. For example, green represents *equals* and blue represents *includes*. We use the same color coding for articulations in the the rest of this paper. Users can enable or disable an articulation type as desired (**T1**).

ProvenanceMatrix supports three ways of ordering taxonomic concepts, designed to highlight different aspects of the input hierarchies as well as their RCC-5 articulations. (1) Ordering the matrix with respect to the structure of the input trees. Figure 4 shows *ProvenanceMatrix* with different orderings of taxonomic concepts (**T2**). (1.1.) Breadth-first ordering in Figure 4(a) lists all sibling together before diving into their respective child-level concepts. (1.2.) Depth-first ordering in Figure 4(b) lists the children right after each taxonomic concept. The hierarchy is more readable in this ordering since there are no crossing arcs in the same taxonomic classification. To avoid the overlapping between arcs and glyphs in the matrix, we can replace arcs by straight lines connecting parent to child concepts. (2) In Figure 4(c), we order the taxonomic concepts based on the similarity of their articulation sets. (The details of how we compute similarity and the ordering algorithm are described in [5].) This ordering brings concepts with multiple alignments to the top left corner of the matrix; these multiple alignments generate the 160 possible worlds in the taxonomy alignment of Gymnospermae sec. Weakley (2010) versus RAB (Radford, Ahles and Bell) (1968) [9]. The example shows ambiguities in the multi-taxonomy alignment which our visualization software can readily identify and isolate to facilitate user-mediated diagnosis and resolution of such ambiguities. In addition, congruent relations (in green) are pushed further to the bottom right of the matrix.

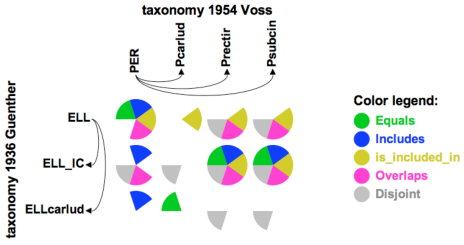


Fig. 3. An example showing the use of *ProvenanceMatrix* for the *Perelleschus* classifications.

Due to the discovery and/or inclusion of new taxonomic entities in the later (2010) classification, the alternative taxonomies have unequal sets of leaf-level children. In other words, recently published taxonomies may include new species-level concepts for which there are no corresponding entities in preceding classifications. Accordingly, in *ProvenanceMatrix*, we classify taxonomic concepts into three different categories (**T3**):

- Neither the concept nor any of its children of one taxonomy have congruent relationships with entities in the other taxonomy. In other words, a (set of) concept(s) has no match whatsoever ("bad apples"). Such concepts are highlighted in red in Figure 5.
- A parent-level concept is incongruent but entails one or more congruent child-level concepts. In other words, the higher-level concepts is a unique conglomerate of

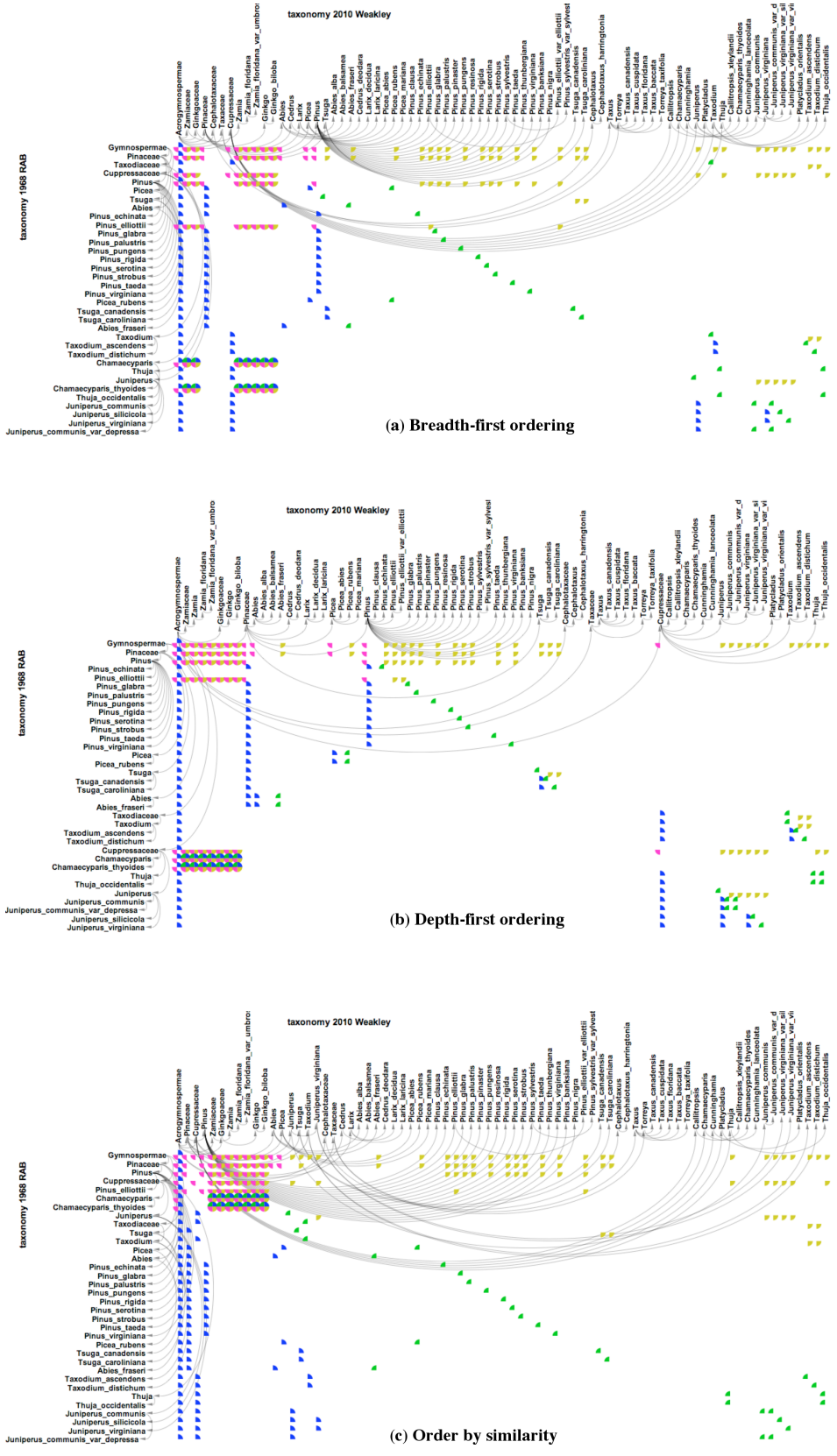


Fig. 4. Visualizing the alignment of Gymnospermae sec. Weakley (2010) vs. RAB (1968) [9]: (a) Breadth-first ordering (b) Depth-first ordering (c) Order by similarity.

variously congruent subentities, some of which have matching entities in the other taxonomy. Such parent-level concepts are the dark green entities in Figure 5.

– A concept has at least one congruent relationship with a concept in the other taxonomy. Such concepts are highlighted in green.

In Figure 5, we also show brushing and linking to highlight the corresponding subtrees of the aligned taxonomic classifications (**T4**). An associated subtree is discovered based on the presence of congruent relationships which are connected by green lines. In this example, the associated subtree (on the left) of *Pinus* sec. 2010/1968 (in the box) is discovered in light of its aligned children, not the selected (higher-level) taxonomic concept itself. Notice that half of the children (in red) of *Pinus* sec. Weakley (2010) have no congruent match in the RAB (1968) classification.

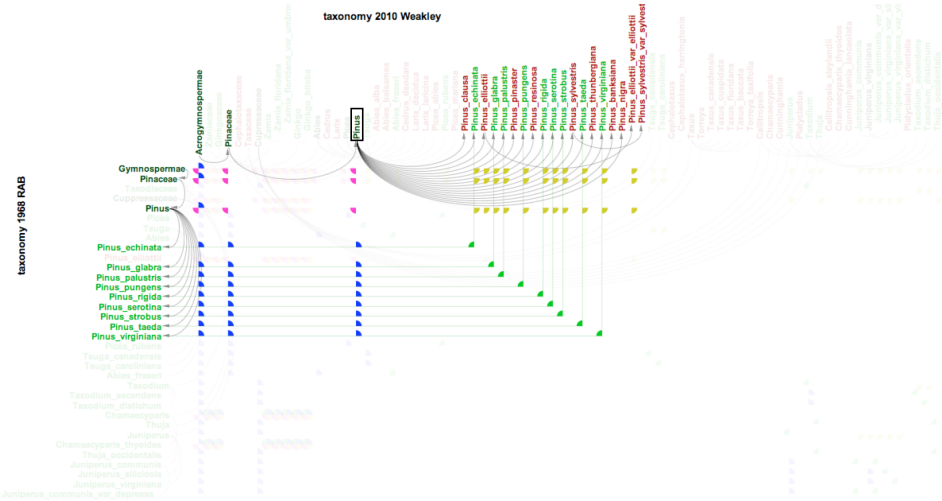


Fig. 5. Brushing *Pinus* sec. 2010/1968 in the Gymnospermae use case [9]. Red are incongruent concepts, dark green are incongruent but (some of) the children are congruent, green are congruent.

Additional information and sample images (e.g., from Wikipedia pages of which may entail taxonomic concept information) can be displayed on demand when mousing over a taxonomic concept label (**T5**). Moreover, users can request to overlay the source of articulations (i.e., user input, reasoner inference). Figure 6 shows an example of overlaying such articulation sources in a non-domain demonstration alignment of U.S. regional classifications from National the Diversity Council and Big Data Hubs, respectively. In particular, black cells indicate user input whereas light blue and pink cells are deduced and inferred articulations. Notice that articulation types (circular sectors) are still visible in each cell.

ProvenanceMatrix offers two ways navigate and comprehend larger classifications of hundreds of taxonomic concepts: (a) lensing and (b) collapsing a subtree of the input

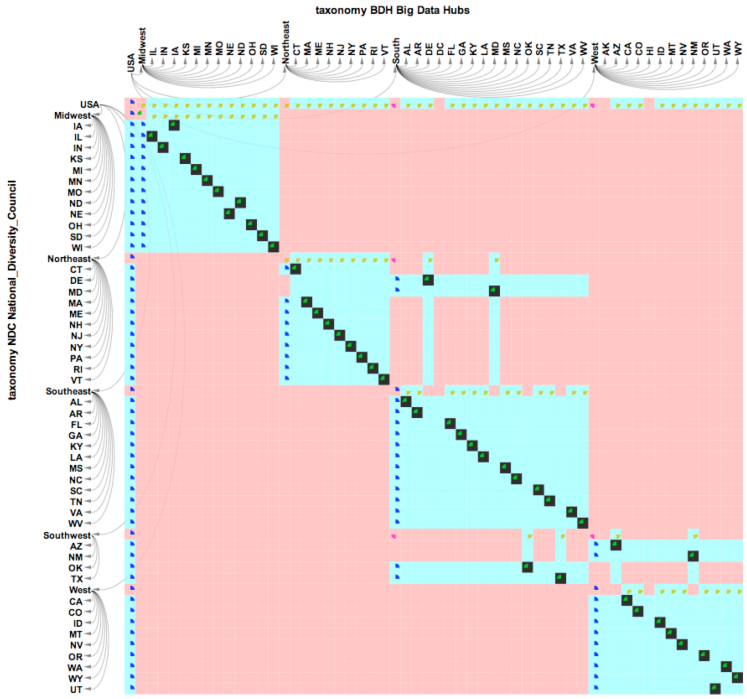


Fig. 6. Visualizing a non-domain, demonstration alignment of U.S. regional classifications. Black cells are user input whereas light blue and pink cells are deduced and inferred articulations.

hierarchies (**T6**). Figure 7 shows a use case of aligning the Primates sec. Groves (1993) and sec. Groves (2005) that contains 317*483 taxonomic concepts and hence 153,111 MIR [11]. Figure 7(a) shows lensing on a sub-section of the matrix, where only the concept labels (about 20 labels) in the lensing area are printed out. Figure 7(b) shows collapsing of a section of the input hierarchies. A plus sign appears in front of those taxonomic concept labels which are collapsed. *ProvenanceMatrix* also provides searching capability. When users input a concept name into a textbox, *ProveanceMatrix* only expands the subtree of the search concept and collapses other irrelevant subtrees. At the same time, only related concepts in the other taxonomic classification are expanded.

The *ProvenanceMatrix* application, source code, and an accompanying video tutorial are available online via our project repository.[2]

5 Expert User Feedback

The herein provided use cases were provided by EULER/X user and co-author NMF, whose feedback has driven the optimization of the new visualizations. *ProvenanceMatrix* confers two immediate and new visualization services:

[2] `https://github.com/CreativeCodingLab/ProvenanceMatrix`.

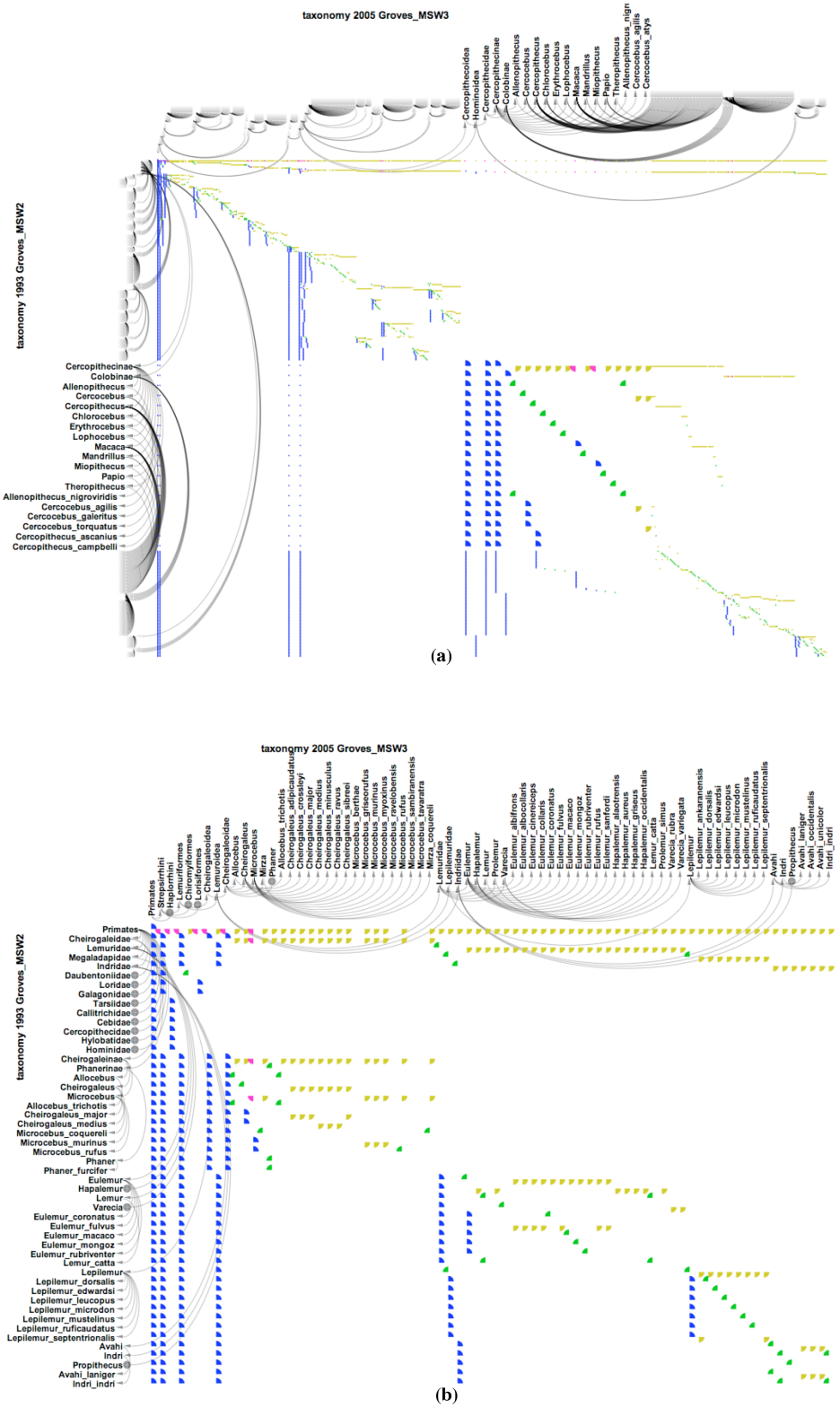


Fig. 7. Visualizing the alignment of two Primates classifications containing 317*483 taxonomic concepts and 153,111 MIR [11]: (a) lensing on area of interest in the matrix (b) collapsing sub-hierarchies.

(1) In cases where certain concept-to-concept articulations are ambiguous (RCC-5 disjunctions) in the output, the corresponding concepts can be spatially aggregated and thus identified very easily by the user. This can lead to an accelerated understanding and subsequent removal of the ambiguity issues. Without the visualization, one has to instead "comb through" a spreadsheet that may contain many thousands of rows of data. We have succeeded in scaling *ProvenanceMatrix* to this level, even with 153,111 articulations in the Primates sec. 2005/1993 use case [11].

(2) We can show "information expression" that is newly acquired through the EULER/X toolkit reasoning process. For instance, in the Primates use case the expert user provided 402 pairwise input articulations. The reasoning process produces from this 153,111 pairwise MIR relations, i.e., about 380 times as many articulations are logically implied by the input but were not explicitly stated therein. The differential levels of information expression before and after the reasoning process are correspondingly visualized with *ProvenanceMatrix* through two matrix versions, and thus show the powers of the reasoning approach.

In summary, *ProvenanceMatrix* provides speedy and interactive identification of ambiguous input and newly inferred output information, presenting a major improvement over existing visualizations.

6 Conclusion and Future Work

This paper introduces a novel technique, *ProvenanceMatrix*, for visualizing the products of a multi-taxonomy alignments generated with the reasoning toolkit EULER/X. Using *ProvenanceMatrix*, users (taxonomists, ecologists, phylogeneticists) can visualize alignments of large taxonomies with up to hundreds of input concepts. Glyphs in each cell highlight RCC-5 articulations for a pair of taxonomic concepts. *ProvenanceMatrix* supports a range of desirable user interactions, such as filtering the matrix by articulations, ordering taxonomic entities with respect to the structure of the input hierarchies, brushing and linking concepts, and collapsing/expanding sub-hierarchies. We have demonstrated how our application effectively facilitates the exploration of multi-taxonomy alignments with different levels of alignment ambiguity and varying sizes, from a few to hundreds of taxonomic entities (and hundreds of thousands of relationships). This technique can be extended to visualize more than two taxonomic classifications – a feature in development for the corresponding reasoning toolkit. In particular, we can have multiple input classifications aligned by rows and columns, where each pair of taxonomic classifications forms a new *ProvenanceMatrix*. In other words, we can create a matrix of *ProvenanceMatrix* matrices, where each cell contains a matrix (similar to the idea of a scatterplot matrix). Future work will investigate this strategy to enable multi-dimensional alignments.

Acknowledgements

This work was funded by the DARPA *Big Mechanism* Program under ARO contract WF911NF-14-1-0395, and in part by the National Science Foundation through NSF DEB-1155984, DBI-1342595, NSF IIS-118088, and DBI-1147273.

References

1. M. Chen, S. Yu, N. Franz, S. Bowers, and B. Ludäscher. Euler/x: A toolkit for logic-based taxonomy integration. *CoRR*, abs/1402.1992, 2014.
2. M. Chen, S. Yu, N. Franz, S. Bowers, and B. Ludäscher. A hybrid diagnosis approach combining black-box and white-box reasoning. In A. Bikakis, P. Fodor, and D. Roman, editors, *Rules on the Web. From Theory to Applications*, volume 8620 of *Lecture Notes in Computer Science*, pages 127–141. Springer International Publishing, 2014.
3. P. Craig and J. Kennedy. Concept relationship editor: a visual interface to support the assertion of synonymy relationships between taxonomic classifications, 2008.
4. T. N. Dang, P. Murray, J. Aurisano, and A. G. Forbes. ReactionFlow: Visualizing relationships between proteins and complexes in biological pathways. *BMC Proceedings*, 9(6):S6, August 2015.
5. T. N. Dang, P. Murray, and A. G. Forbes. PathwayMatrix: Visualizing binary relationships between proteins in biological pathways. *BMC Proceedings*, 9(6):S3, August 2015.
6. J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz - Open Source Graph Drawing Tools. *Graph Drawing*, pages 483–484, 2001.
7. N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J. Fekete. Zame: Interactive large-scale graph visualization. In *Visualization Symposium, 2008. PacificVIS '08. IEEE Pacific*, pages 215–222, March 2008.
8. N. Franz and R. Peet. Perspectives: towards a language for mapping relationships among taxonomic concepts. *Systematics and Biodiversity*, 7(1):5–20, 2009.
9. N. M. Franz, M. Chen, S. Yu, P. Kianmajd, S. Bowers, and B. Ludäscher. Reasoning over taxonomic change: Exploring alignments for the *Perelleschus* use case. *PLoS ONE*, 10(2):e0118247, 02 2015.
10. N. M. Franz, R. K. Peet, and A. S. Weakley. On the use of taxonomic concepts in support of biodiversity research and taxonomy. *Systematics Association Special Volume*, 76:63, 2008.
11. N. M. Franz, N. M. Pier, D. M. Reeder, M. Chen, S. Yu, P. Kianmajd, S. Bowers, and B. Ludäscher. Taxonomic Provenance: Two Influential Primate Classifications Logically Aligned. *ArXiv e-prints*, Dec. 2014.
12. M. Gelfond. *In Handbook of Knowledge Representation*, chapter Answer Sets. Elsevier Science, 2007.
13. N. Henry and J.-D. Fekete. Matlink: Enhanced matrix visualization for analyzing social networks. In C. Baranauskas, P. Palanque, J. Abascal, and S. Barbosa, editors, *Human-Computer Interaction INTERACT 2007*, volume 4663 of *Lecture Notes in Computer Science*, pages 288–302. Springer Berlin Heidelberg, 2007.
14. J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.
15. D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd international conference on Knowledge Representation and Reasoning*, 1992.
16. C. Scornavacca, F. Zickmann, and D. H. Huson. Tanglegrams for rooted phylogenetic trees and networks. *Bioinformatics*, 27(13):i248–i256, 2011.
17. D. Thau. Reasoning about taxonomies and articulations. In *Proceedings of the 2008 EDBT Ph. D. workshop*, pages 11–19. ACM, 2008.
18. W. N. W. Zainon and P. Calder. Visualising phylogenetic trees. In *Proceedings of the 7th Australasian User Interface Conference - Volume 50*, AUIC '06, pages 145–152, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

Visual Analytics for Ontology Matching Using Multi-Linked Views

Jillian Aurisano[1], Amruta Nanavaty[2], and Isabel F. Cruz[2]

[1] EVL Lab, Department of Computer Science, Univ. of Illinois at Chicago, USA
jillian.aurisano@gmail.com

[2] ADVIS Lab, Department of Computer Science, Univ. of Illinois at Chicago, USA
aybgr8@gmail.com, isabelcfcruz@gmail.com

Abstract. Ontology matching is the key to data integration on the Semantic Web. Advanced ontology matching systems incorporate a variety of algorithms. However, they do not always guarantee a complete and correct alignment (set of mappings). Hence, user involvement in the matching process is essential for complex ontologies. In this paper, we explore the power of multi-linked views, where actions in one view affect the display of the other views, thereby extending significantly the state of the art in ontology matching visualization in general and that of visual analytics for ontology matching in particular. A preliminary assessment of our approach that uses the ontologies of the OAEI Conference Track points to the effectiveness of our approach.

1 Introduction

Data integration provides the ability to manipulate data transparently across multiple data sources. At the heart of data integration are ontologies and the ability to establish semantic mappings among them using ontology matching [10].

Semi-automatic approaches to ontology matching allow for experts to intervene by validating or eliminating results that were automatically determined and then iteratively incorporating that feedback into the matching process [7, 3, 4]. To perform this determination, analytical reasoning is needed, which, when supported by an interactive visual interface, is called *visual analytics* [1]. In this paper, we propose the AlignmentVis visualization tool, which uses the AgreementMaker ontology matching system [5], but can be easily adapted to other advanced matching systems with a comparable architecture. We describe next the terminology associated with ontology matching systems and describe the architectural components of AgreementMaker.

The process of ontology matching finds semantic mappings between different entities (classes and properties) of a source and target ontologies, by using a wide range of lexical, syntactic, and structural automatic matching algorithms called *matchers*. A matcher produces a similarity matrix where each row represents a source entity, each column represents a target entity, and each cell contains the *confidence score* for the source-target pair. In AgreementMaker, matchers include the Base Similarity Matcher (BSM), the Parametric String based Matcher (PSM), the Vector-based Multi-word Matcher (VMM), the Lexical Synonym

Matcher (LSM), and the Descendant Similarity Inheritance (DSI) matcher [8, 6]. The Linear Weighted Combination (LWC) matcher combines similarity matrices as produced by the automatic matchers using weights determined by a local quality measure [6]. For each mapping, the combined confidence score is stored in the corresponding element of the LWC matcher similarity matrix. Finally, a set of mappings, called an *alignment*, is selected from this matrix according to an optimization criteria [6]. The performance of an ontology matching system is evaluated by comparing the obtained alignment against a gold standard, also called *reference alignment*, created by domain experts.

We interviewed ontology matching experts to identify the analytic tasks that need to be supported by an advanced visualization tool, as summarized next:

Matcher's performance evaluation Expert users need to evaluate the performance of individual matchers and the quality of the final alignment with respect to the reference alignment. Users also want to characterize the mappings into true positives (correct mappings), false positives (incorrect mappings), and false negatives (missed mappings). When no reference alignment is available, the techniques outlined below may be necessary.

Mapping clusters In addition to a high-level evaluation of the performance of each matcher, expert users may take advantage of clusters of mappings that are grouped according to different statistics and then analyze each cluster in order to assess the performance of an individual matcher.

Exploration and comparison The evaluation of the performance of a matcher makes use of exploration and comparison tasks. Views of entity details, through meaningfully designed explorative interactions and through comparative views of the results across different matchers, should help in identifying potential sources of error.

Diagnosis Once errors are identified by using exploration and comparison, this complex task will help to identify the cause of the errors. It is not an individual task, but rather a combination of the previously outlined tasks as users will iterate through them to arrive to a determination.

For these analytic tasks, in this paper we explore the power of multi-linked views, where actions in one view affect the display of the other views [20, 2], therefore extending significantly the state of the art in ontology matching visualization in general and that of visual analytics for ontology matching in particular.

This paper is organized as follows. In Section 2, we outline the most relevant approaches to ontology matching visualization with a focus on visual analytics. In Section 3, we describe in detail all the views we have created, the tasks they fulfill, and how they are linked to one another. In Section 4, we describe the dataset on which we tested AlignmentVis and the environment in which it was developed. In Section 5, we point to a few examples that demonstrate the kind of anomalies that the interface can help detect. Finally, in Section 6, we draw brief conclusions and point to future work that will quantify the benefits of a visual analytics tool like AlignmentVis.

2 Related Work

A recent survey on user involvement for large ontology matching covers several visualization tools [16]. However, those tools do not support fully the necessary requirements laid out by the authors. For those domain expert users that rely on visualization tools for ontology matching, much more functionality is needed including debugging the obtained alignment (set of mappings), observing similar characteristics in a group of mappings, and assessing the contribution of individual matching algorithms to the final alignment. Essentially, those users need a tool that allows them to detect those mappings that are incorrect and confirm the mappings that are correct. In spite of their limitations, we cover next some of the visualization tools in the aforementioned survey and add to them a couple more, which are especially relevant given their focus on visual analytics.

A representation that is cluster based shows both detailed and general information of the matching results and provides in addition a JTree-like visualization [18]. Users can select the level at which they want to cluster the results. For the visualization of each ontology this approach uses a spring-embedded graph drawing algorithm [11, 9]. A drawback of this approach is that only the results of a single matching algorithm can be visualized. Another approach based on a spring-embedded technique was developed for the AgreementMakerLight system [15], which extends AgreementMaker [5] to very large ontologies; it provides a single visualization where both ontologies and the mappings between classes are displayed. However, it is not intended to display more than a few mappings at a time [21]. This technique also does not allow for the users to compare the results of more than one matching algorithm at once.

Prompt+CogZ is an advanced visualization tool that supports multiple visualizations, including one based on TreeMaps and another one that displays pie charts [14]. TreeMaps have the advantage that they can be used to visualize large amounts of data, but fit in a small area. However, this tool does not seem to be able to show concurrent displays of more than one matching algorithm and also does not provide analytical details about the mappings or about the contribution of an individual matcher to the alignment process. A recent highly interactive visualization based solely on pie charts has two important features: it scales to very large ontologies and can compare different matching algorithms [19]. Its focus on scalability makes it a possible complement to the multi-linked visualization approach of this paper.

A matrix visualization where the classes of both ontologies are placed along the X and Y axes provides a more comprehensive view of the matching process as compared with other methods because it allows for the whole mapping space to be visualized with equal detail. We know of two such visualizations: the one provided by iMERGE [12] and the visual analytics panel provided by AgreementMaker [7]. Both systems support multiple visualizations, including a traditional JTree-like visualization for each ontology with connections between the two ontologies showing the mappings. AgreementMaker has the distinct capability of allowing for the comparison of different matching algorithms side by side and simultaneous navigation across the various similarity matrices. In AlignmentVis,

we want to preserve the unique characteristic of AgreementMaker to display the matching results across several algorithms and its applicability to visual analytics for ontology matching [7]. However, we also want to support multiple views in the same panel, including a matrix view.

3 AlignmentVis Design

AlignmentVis addresses the cognitive support requirements for ontology alignment systems, which are meant to facilitate user involvement, by presenting the mapping results in four linked views. First we describe the three views that are related to the same individual matcher, then the fourth view compares all the matchers. The views display: (1) an overview of the mappings obtained between all the entities in the source ontology and in the target ontology, as presented in the *Matcher Output Grid View*; (2) the behavior of the entities of the source and target ontologies with respect to various statistics, as provided by the *Entity Mapping Characteristics Scatter Plot View*; (3) the mappings between entities in the source and target ontologies, which uses the interactive *Ontology Tree View*; (4) the results for all the matchers alongside the reference alignment (when available) for comparative analysis, as enabled by the *Parallel Coordinate View*. The interface of the AlignmentVis tool is shown in Figure 1.

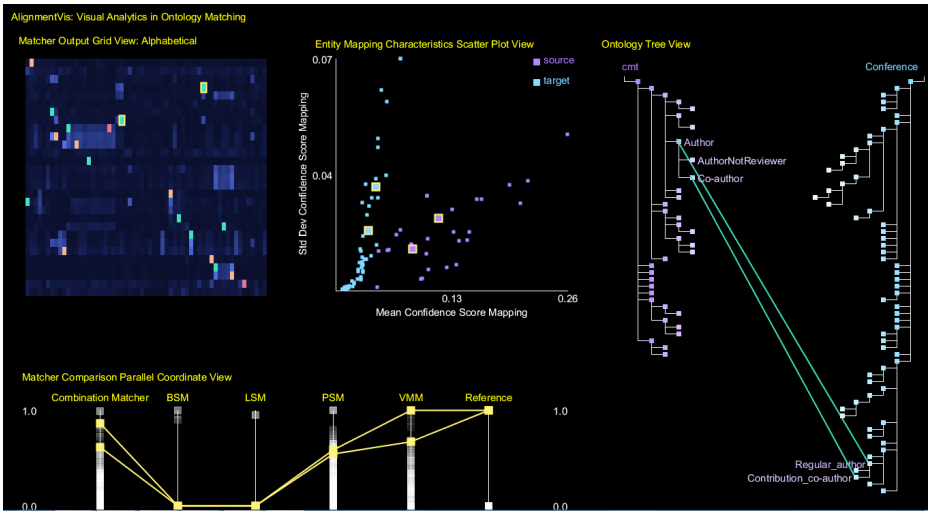


Fig. 1: AlignmentVis user interface.

3.1 Matcher Output Grid View

The Matcher Output Grid View displays a two dimensional matrix where each row represents a source entity, each column represents a target entity, and each

cell value represents the confidence score of the selected matcher for a source-target pair. That score ranges from 0 to 1 where values close to 1 indicate high similarity between the source and target entities and values close to 0 indicate high dissimilarity. The confidence score of a mapping sets a color gradient from black for a score of 0 to bright blue for a score of 1. If a cell is colored green then it is a correct mapping. It means that the corresponding mapping is present both in the alignment that is computed by the algorithm and in the reference alignment. If a cell is colored red it is a false negative or missed mapping, which means that the mapping is present in the reference alignment but not in the final alignment. If a cell is colored orange, it is a false positive, which indicates that the mapping is present in the final alignment but not in the reference alignment. The color scheme aims to make the overall performance of the selected matcher immediately evident.

Users can hover over the view to see the confidence score and the labels of the participating source and target entities of the selected mapping. Moreover, as the view is linked to other views, the cell representing the corresponding mapping in the matrix is highlighted by a yellow box whenever a corresponding mapping or participating source and/or target entities are selected in other views.

If an individual source (or target) entity is selected in the other views of AlignmentVis, then its corresponding row or column in the Matcher Output Grid View is highlighted. The Grid View helps users to rapidly explore individual mappings and to observe how each entity from the source ontology is related to the entities of the target ontology.

Several reordering features are available for the source and target entities to facilitate the recognition of patterns associated with the detected or missed mappings:

Alphabetical order The labels of the source and target ontology class entities are ordered alphabetically in ascending order. The corresponding rows and columns in the Grid View are rearranged accordingly as shown in Figure 2a.

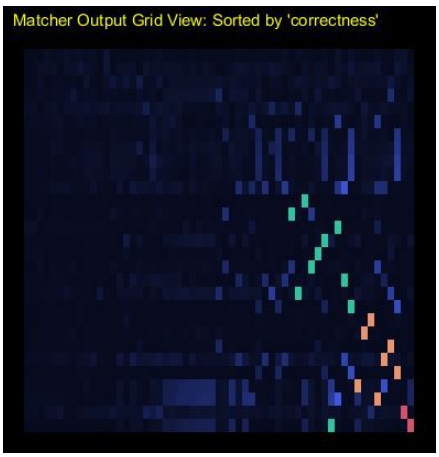
Ascending order of the mean value of the confidence scores of the corresponding class entity As mentioned earlier, each row represents a source entity and its relation to the target entities. The mean value is computed for each row and then the rows are reordered in ascending order of their mean value. Similar computation and reordering can be performed for each column.

Ascending order of the standard deviation value of the confidence scores of the corresponding class entity The procedure for reordering is as in the previous case, but instead of the mean, the standard deviation is calculated.

Mapping categorization The entities are reordered by first displaying the source entities that are not related to any of the target entities followed by those that are present in the reference alignment. Thereafter, the source entities that are present in the false positive mappings are displayed and lastly the source entities that are involved in the missed mappings are displayed. The same reordering is available for the target entities. This kind

of reordering displays distinct mapping clusters with similar characteristics. Users can then explore these entities and associated mappings and look for similar characteristics in the other views. The mapping categorization view is shown in Figure 2b.

(a) Reordered view in ascending alphabetical order.



(b) Reordered view in ascending correctness order.

Fig. 2: Matcher Output Grid View.

3.2 Entity Mapping Characteristics Scatter Plot View

An entity can be described by a vector, where each element indicates a confidence score of the mapping between the entity and all the entities in the other ontology. Various statistics like mean, standard deviation, and correctness can be computed from that vector. These statistics can give an insight into the potential mappings associated with that individual entity. In the Scatter Plot View, which is displayed in Figure 3, entities of the source and target ontology are displayed as nodes in a scatter plot with respect to any of these two statistics, where one of them is displayed in the X axis and the other one in the Y axis. Users can switch between the chosen statistics and exchange the X and the Y axes.

A node is colored depending on whether the representative entity belongs to the source or to the target ontology. The Scatter Plot View helps to identify different characteristics of the source and target ontologies. Users can interact with this view by hovering over the nodes, which become highlighted in the other views. In addition, when users select nodes in another view, they are highlighted in the Scatter Plot View. This view also allows for comparing the performance of an individual matcher with that of other matchers with respect to the computed statistics.

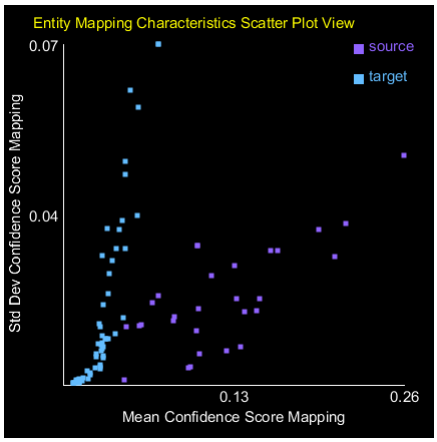


Fig. 3: Entity mapping characteristics using the Scatter Plot View.

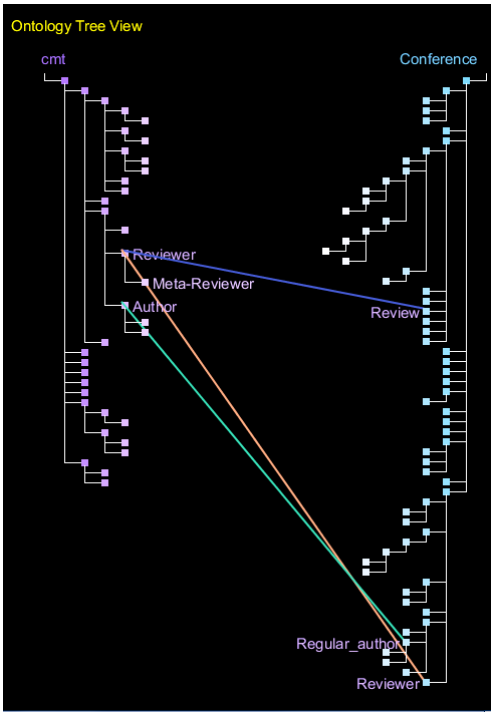


Fig. 4: Ontology Tree View displaying the source and the target ontologies.

3.3 Ontology Tree View

In the Ontology Tree View, which is shown in Figure 4, the hierarchical structure of the source and of the target ontologies are displayed using trees. Users can

hover over a section of the tree in order to view the mappings involving the entities under the selected section. Only those mappings that have a confidence score above a predefined threshold for the selected matcher are displayed by a colored line between the source and target trees. The color scheme is the same as in the Matcher Output Grid View. Mappings are available on demand to facilitate the users' focus on entities of interest and to avoid information overload. The related information about the displayed mapping can be viewed in other views due to the multi-linked view feature of AlignmentVis.

3.4 Comparative Analysis of Matchers Using a Parallel Coordinate View

The Parallel Coordinate View, which is shown in Figure 5, is at the heart of the AlignmentVis interface. Each vertical axis represents a matcher on which rectangles associated with the mappings are positioned relative to their confidence score. This allows for users to quickly compare the confidence score associated with a mapping across all the matchers. The minimum value on each axis is 0 and the maximum value is 1. When hovering over any of the vertical axes, the mappings in that area are highlighted and lines are drawn connecting the highlighted mappings across the rest of the vertical axes. The confidence score related to the current position of the mouse on the selected vertical axis is also displayed. The Parallel Coordinate View also helps users identify which matcher plays a dominant role in identifying the mapping. This identification is possible because one of the vertical axes represents the combination matcher. In turn, it is easy to compare the result produced by the combination matcher with the reference alignment. The related information to the highlighted mapping is displayed in the other linked views. Hovering over the Matcher Output Grid View or the Ontology Tree View produces yellow colored lines drawn across all the vertical axes for the selected mappings. In addition, by linking this view to other views, users can analyze whether mappings having similar confidence scores across various matchers tend to have distinct characteristics in the other views or not.

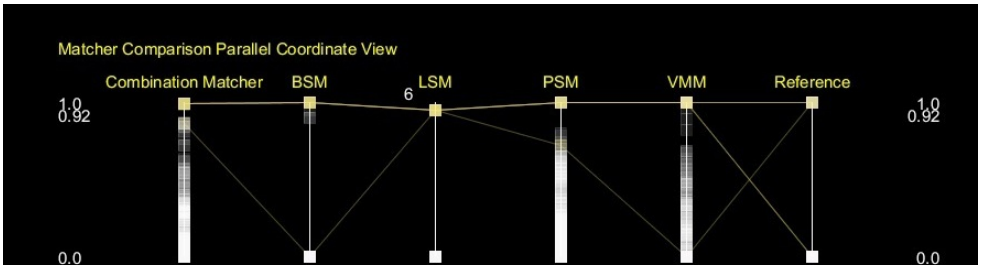


Fig. 5: Parallel Coordinate View.

4 Dataset and Implementation Language

The datasets used for testing and evaluating this interface are from the Conference Track of the Ontology Alignment Evaluation Initiative (OAEI), which is an annual international campaign for the systematic evaluation of ontology matching systems.[3] The Conference Track uses 16 ontologies from the conference organization domain from three types of underlying resources:

1. Actual conferences and their web pages. For example, the SIGKDD ontology is based on the organization of the ACM conference with the same name.
2. Actual software tools for conference organization. For example, the OpenConf ontology is designed using high level concepts from the tool with the same name that was developed for peer-review, abstract, and conference management.
3. People's experience based on their participation in the organization of an actual conference.

These ontologies are suitable for the ontology matching task because of the homogeneity of their domain of interest and of the heterogeneity of their organization, given their very different origins. Each ontology contains less than 200 concepts.

We have used AgreementMaker to perform the ontology matching task for these ontologies and used the similarity matrix and alignment that was produced by AgreementMaker for each of the matchers. We note that AgreementMaker has been the winner for this track, therefore it produces high quality mappings on this dataset [13]. Thus, user interaction and visual analytics can play an important role even when the automatically obtained results are of high quality.

AlignmentVis is implemented in Processing. Processing is an open source programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks.[4] Processing is built on the Java language, but it uses a simplified syntax and graphics programming model. It allows for quick prototyping and is easy to learn.

5 Evaluation

We tested AlignmentVis with the ontologies of the Conference Track of the OAEI. Each ontology contains less than 200 entities. Till now, most of the ontology matching systems have focused on different ways of visualizing the alignment and very few have made an effort to apply visual analytics to support the involvement of users in the ontology alignment task, therefore is not a standard way to evaluate the benefits provided by tools such as ours. In the absence of an established evaluation methodology, we tested extensively our user interface to evaluate the benefits provided by the multi-linked views to analyze the performance of single matchers and of their combination to produce a final alignment

[3] http://oaei.ontologymatching.org/
[4] https://processing.org/

for the Conference Track. We describe a couple of interesting examples and observations.

In the Ontology Tree View of Figure 4, there is an incorrect mapping highlighted in orange between the source entity *Reviewer* and the target entity *Reviewer* and a correct mapping between the source entity *Author* and the target entity *Regular_author*. Another mapping, in blue, shows a potential mapping between *Reviewer* and *Review*, the only mapping whose value is above a set threshold. Here the domain expert analyzes first the tree view, to see that the distance between *Reviewer* and *Author* in the source ontology is much smaller (they are siblings) than the distance between *Review* and *Regular_author* in the target ontology, a possible indication of an incorrect mapping [17]. In comparison, the green and orange mappings (even if not preserving the sibling relationship), appear acceptable. The expert then analyzes the corresponding Parallel Coordinate View of Figure 6, to discover that all the matchers show high confidence for the mapping between *Reviewer* and *Reviewer*, only contradicted by the reference alignment. This example indicates a possible error in the reference alignment of the Conference Track, which is, in fact, currently undergoing a revision.

For another example that shows how two views can provide complementary information, we focus on Figures 5 (Parallel Coordinate) and 2a (Grid). The former shows that the LSM matcher produces heavily split confidence scores (that is, either 1 or 0). The latter shows the six mappings detected by LSM, of which the majority (four) are true positive mappings. Further interaction will allow for the detailed analysis of each of these mappings in comparison with the results provided by the other matchers.

The Scatter Plot View of Figure 3 shows that the source and target entities display distinct mean and standard deviation statistics. It would be valuable to see whether a similar difference exists between the source and target ontologies of the other OAEI tracks, or whether it is unique to the Conference Track. The Scatter Plot View can contribute to the determination of the intrinsic quality of mapping, given that a high standard deviation may point to the existence of a target entity for which the matcher has a clear preference over the other target entities [6]. This indication can be cross-investigated by the multiple perspectives that are made possible by the unique multi-linked functionality of AlignmentVis.

6 Conclusions

Ontology matching is a key component of data integration. Various lexical, syntactic, and structural automatic matching algorithms contribute to the set of mappings between two ontologies. However, as these algorithms do not guarantee 100 percent accuracy, user involvement is required. Expert users can make real-time decisions for a set of candidate mappings during the ontology matching process, so as to validate or eliminate those mappings. To make such decisions, they benefit from the visualization of the mappings and of the results produced by the various matchers by focusing on the performance of each of them, allowing for statistics to be displayed, mapping clusters to be visualized, and enabling exploration and comparison, so as to diagnose any anomalies in the ontology matching process or to confirm mappings.

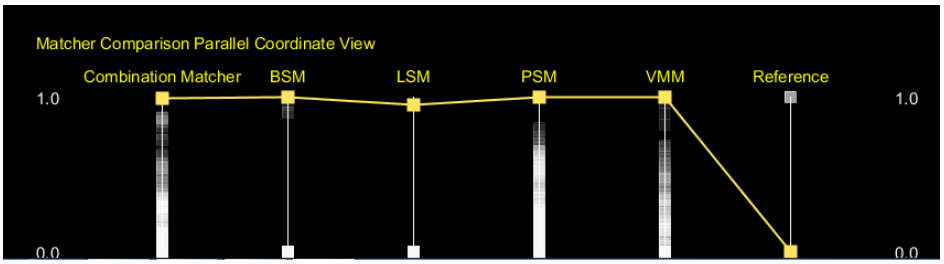


Fig. 6: Parallel Coordinate View for the mapping between the source entity *Reviewer* and the target entity *Reviewer*, which corresponds to the Ontology Tree View of Figure 4.

AlignmentVis provides users with an interactive visual interface, allowing them to conduct analytical reasoning, the two key components of a visual analytics process. In our interactive visual interface, we explore the use of multi-linked views, a known technique in the field of information visualization, yet till now seldom used in the realm of Ontology Matching. Our initial evaluation indicates that the multi-linked views of the interface satisfy important cognitive and interactive user requirements necessary for the ontology matching task. Future work will attempt to quantify the improvement in performance that is obtained from using AlignmentVis.

Acknowledgments

This research was partially supported by NSF Awards IIS-1143926, IIS-1213013, and CCF-1331800.

References

1. Bertini, E., Lalanne, D.: Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery. SIGKDD Explorations Newsletter 11(2), 9–18 (May 2010), http://doi.acm.org/10.1145/1809400.1809404
2. Cruz, I.F., Huang, Y.F.: A Layered Architecture for the Exploration of Heterogeneous Information Using Coordinated Views. In: Symposium on Visual Languages and Human-Centric Computing (VL/HCC). pp. 11–18. IEEE Computer Society (2004)
3. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW), pp. 80–96. Springer (2014)
4. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Quality-Based Model for Effective and Robust Multi-User Pay-As-You-Go Ontology Matching. In: Semantic Web Journal. IOS Press (2015)
5. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB 2(2), 1586–1589 (2009)
6. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC Interna-

tional Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 551, pp. 49–60 (2009)
7. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE). pp. 1321–1324 (2012)
8. Cruz, I.F., Sunna, W.: Structural Alignment Methods with Applications to Geospatial Ontologies. Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications 12(6), 683–711 (2008)
9. Cruz, I.F., Tamassia, R.: How to Visualize a Graph: Specification and Algorithms. In: IEEE Symposium on Visual Languages (VL) (1994), `http://www.cs.brown.edu/people/rt/gd-tutorial.html`
10. Cruz, I.F., Xiao, H.: The Role of Ontologies in Data Integration. Journal of Engineering Intelligent Systems 13(4), 245–252 (December 2005)
11. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for Geometric Representation of Graphs. Prentice Hall (1999)
12. El Jerroudi, Z., Ziegler, J.: iMERGE: Interactive Ontology Merging (poster). In: International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW) (2008)
13. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., dos Santos, C.T., Vouros, G.A., Wang, S.: Results of the Ontology Alignment Evaluation Initiative 2009. In: ISWC International Workshop on Ontology Matching (OM). CEUR Workshop Proceedings, vol. 551, pp. 73–126 (2009)
14. Falconer, S.M., Storey, M.A.D.: A Cognitive Support Framework for Ontology Mapping. In: International Semantic Web Conference/Asian Semantic Web Conference (ISWC/ASWC). Lecture Notes in Computer Science, vol. 4825, pp. 114–127. Springer (2007)
15. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight Ontology Matching System. In: International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE). pp. 527–541. Springer (2013)
16. Ivanova, V., Lambrix, P.: User Involvement for Large-Scale Ontology Alignment. In: International Workshop on Visualizations and User Interfaces for Knowledge Engineering and Linked Data Analytics (2014)
17. Joslyn, C., Donaldson, A., Paulson, P.: Evaluating the Structural Quality of Semantic Hierarchy Alignments. In: International Semantic Web Conference (Posters & Demos) (2008)
18. Lanzenberger, M., Sampson, J.: AlViz - A Tool for Visual Ontology Alignment. In: Conference on Information Visualization (IV). pp. 430–440. IEEE Computer Society (2006)
19. Li, Y., Stroe, C., Cruz, I.F.: Interactive Visualization of Large Ontology Matching Results. In: ISWC International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data (Voila!) (2015)
20. North, C., Shneiderman, B.: Snap-together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata. In: Advanced Visual Interfaces (AVI). pp. 128–135 (2000)
21. Pesquita, C., Faria, D., Santos, E., Neefs, J.M., Couto, F.M.: Towards Visualizing the Alignment of Large Biomedical Ontologies. In: Data Integration in the Life Sciences (DILS). pp. 104–111 (2014)

Interactive Visualization of Large Ontology Matching Results

Yiting Li, Cosmin Stroe, and Isabel F. Cruz

ADVIS Lab, Department of Computer Science, University of Illinois at Chicago, USA
yiting.star@gmail.com, cstroe@gmail.com, isabelcfcruz@gmail.com

Abstract. We add to the widely used AgreementMaker system the capability to visualize the results of matching large ontologies with a user interface that supports navigation and search of the ontologies. The interface also supports user intervention when using a feedback loop strategy where users validate candidate mappings that have been computed automatically by matching algorithms. The interface further displays properties of the concepts to facilitate the decision process.

1 Introduction

An ontology provides a vocabulary describing a domain of interest and a specification of the meaning of terms in that vocabulary. An increasing number of organizations are using ontologies to organize their knowledge. However, different ontologies exist for the same knowledge domain. To address this issue, *ontology matching* is needed, which is the process of finding the relationships, called *mappings*, between concepts (classes or properties) of two different ontologies, the *source* and the *target* ontologies [1]. Ontology matching can be performed automatically, manually, or semi-automatically.

A variety of algorithms, which we call *matchers*, have been developed for matching. For example algorithms that are based on string similarity of the class labels or on the structure of the ontologies. Advanced ontology matching systems, such as AgreementMaker, use combinations of a large variety of algorithms [2, 3]. In this paper, we do not focus on any particular matching algorithm, but rather on visualizing the results of the ontology matching process so as to enable the involvement of users with the objective of obtaining better results. The quality of a matching algorithm or of a combination of matching algorithms is measured in terms of precision, recall, and F-measure, by comparing the obtained mappings with the mappings that belong to the *gold standard* or *reference alignment*. The OAEI (Ontology Alignment Evaluation Initiative)[1] makes reference alignments available for a variety of their tracks, which is a great asset for the ontology matching community.

The purpose of our work is twofold. First, we want an interactive visualization method for large ontologies. Second, we want to support visual analytics in a semi-automatic ontology matching process. We define some of these terms next. *Semi-automatic ontology matching* integrates automatic and manual methods. Those mappings that are believed to be incorrect are presented to users for

[1] http://oaei.ontologymatching.org/

validation [4, 5]. The workflow consists of a loop where the outcome of the validation step is fed back into the ontology matching process. *Visual analytics* is the science of analytical reasoning supported by interactive visual interfaces [6]. In ontology matching, visual analytics can help users validate the mappings [4].

Our focus is on ontology matching visualization, not on ontology visualization, that is, we want to support the visualization of complex relationships between source and target ontology structures. Ontology matching visualization is further complicated when matching large ontologies. For example, the display of an ontology as a tree structure using the JTree class can be very helpful for small and medium size ontologies, but is less helpful for large ontologies because of the amount of scrolling needed to locate the different mappings. To better compare and analyze the matching results, a visual representation that can scale to large or very large ontologies is needed. In this paper, we investigate interactive visualizations that use pie charts, which naturally scale to any ontology size.

Our paper is organized as follows. In Section 2, we cover related ontology matching visualization approaches and in particular those that are intended for large ontologies, be they based on graphs, treemaps, or pie charts. We also cover interactive approaches based on matrices that support visual analytics. In Section 3, we describe our visualization technique, starting with the design criteria. We then describe the pie chart visualization and the comparative visualization of matching algorithms, as well as the user interface organization. In Section 4, we describe the use of our interactive interface that supports the validation of mappings in a feedback loop setting and its integration with AgreementMaker. Finally, in Section 5 we draw conclusions and point to future work.

2 Related Work

A recent survey of visualization methods for ontology matching [7] establishes a list of requirements for those systems to support user involvement. However, the functionality of the systems that are covered fall short of those requirements. Therefore, there is the urgent need to develop ontology matching visualization approaches that scale to large and very large ontologies. In what follows, we describe briefly relevant interactive visualization methods.

2.1 Cluster Visualization

The cluster representation [8] shows both detailed and general information of matching results and provides in addition a JTree visualization. Users can select the level at which they want to cluster the results. For the visualization of each ontology, this approach uses a spring-embedded graph drawing algorithm. This method is constrained by its computation complexity, which is $O(n^2 \times s)$ where n is the number of concepts in the ontology and s the number of iterations. Other drawbacks of the approach are that only the results of a single matching algorithm can be visualized. The concepts of each ontology are color coded so as to show whether they have been mapped and the level of similarity found with classes of the other ontology.

Another graph drawing representation that was developed for the AgreementMakerLight system [9], which extends AgreementMaker [2] to very large ontolo-

gies, provides a single visualization that also uses a spring-embedded technique where both ontologies and the mappings between classes are displayed. However, it displays few mappings at a time [10]. This technique does not allow to compare the results of more than one matching algorithm at once.

2.2 Treemap and Pie Chart Visualizations

The PROMPT+COGZ tool supports multiple visualizations, including one based on TreeMaps and another one that displays pie charts [11]. TreeMaps have the advantage that they can be used to visualize large amounts of data, but fit in a small area. Forcefully, details cannot be provided for large ontologies. Some details are provided by a pie chart view with information for each branch of the ontology, such as the number of candidate mappings, mapped classes, and classes that are not mapped. We note, however, that for the display of candidate mappings, the tool falls back on a JTree-like visualization, which uses a fish-eye view lens to allow for the display of larger ontologies. Clearly this is overall an advanced visualization tool. However, it does not seem to be able to show concurrent displays of more than one matching algorithm at a time. Together with the approach we present in this paper, this is the only other tool that supports pie charts with the difference that our pie charts drive the navigation across all levels of the ontologies, while their navigation appears instead to be driven by the TreeMap visualization.

2.3 Matrix Visualization

A matrix visualization where the classes of both ontologies are placed along the X and Y axes provides a more comprehensive view of the matching process as compared with the aforementioned methods because it allows for the whole mapping space to be visualized with equal detail. We know of two such visualizations: the one provided by iMerge [12] and the one provided by AgreementMaker [4]. Both systems support multiple visualizations, including a traditional JTree-like visualization for each ontology with connections between the two ontologies showing the mappings. Like the systems already mentioned, these two systems do not scale to very large ontologies, however AgreementMaker has the distinct capability of allowing for the comparison of different matching algorithms side by side and simultaneous navigation across the various similarity matrices. The color intensity supported by AgreementMaker, which depicts the matching confidence score for each mapping, adds an extra dimension to the visualization without adding extra space. Figure 1 shows the AgreementMaker visual interface for matrix visualization, which is called *Visual Analytics Panel* because it is used to support the visual analytics process. The top toolbar controls the matching process. The overall panel highlights a vector of points for the same mapping (the signature vector). Each matrix is associated with a matching algorithm [4]. AlignmentVis is an interactive user interface that supports matrices among other visualizations, using a multi-linked view paradigm [13]. However, it is not currently targeted to very large ontologies.

In conclusion, none of the above approaches provides both for scalability and for an interactive meaningful display that supports visual analytics for large or very large ontologies.

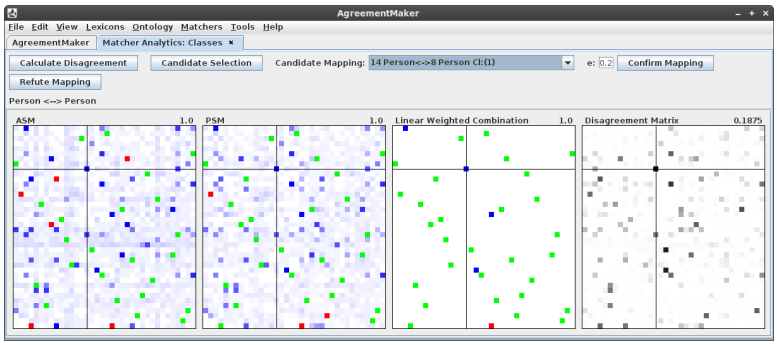


Fig. 1: The Visual Analytics Panel of AgreementMaker [4].

.

3 Visualization

3.1 Design Criteria

We have made the following choices and present their rationale:

- We allow for ontology navigation, exploration, and searching.
- We do not display all the mappings at once. If we did so, it would be difficult to find a visualization whose size does not depend on the number of mappings or on the size of the ontologies involved.
- We want to focus on the mappings one level at a time and aggregate the results for the children of the nodes at that level. As long as navigation and searching functions are available, users can easily locate any single ontology node in the whole structure and see the mappings they need.
- We choose a visualization based on pie charts. The reason of this choice is that no matter how large the data size is, the pie chart requires always the same modest area. Given this, we can display the results of more than one matching algorithm at a time.
- When matching two classes, the most valuable information is the confidence score found by the matching algorithm (between 0% and 100%). The visualization can give priority to those mappings that maximize the confidence score between two nodes.
- We want to make apparent the differences between matching algorithms.
- We enable user feedback acquisition.

3.2 Pie Chart Visualization

We start by describing how we visualize the information in terms of pie charts. For each visualization there are two pie charts, one that corresponds to a concept or node in the source ontology graph, S, called the *current node*, and a pie chart corresponding to a concept or node in the target ontology graph, T. We show the percentage of their children whose confidence score falls in a particular range. Figure 2 shows those ranges, namely 81%-100%, 61%-80%, 41%-60%, and $\leq$40%. For example, 41% of the children nodes of S have confidence scores in the range

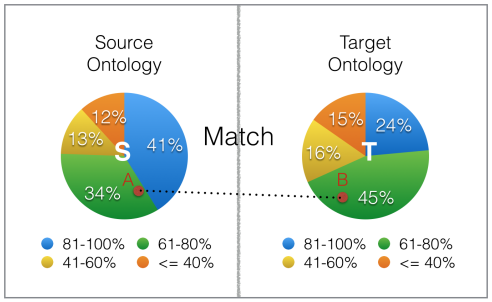


Fig. 2: User interface design that displays matching results.

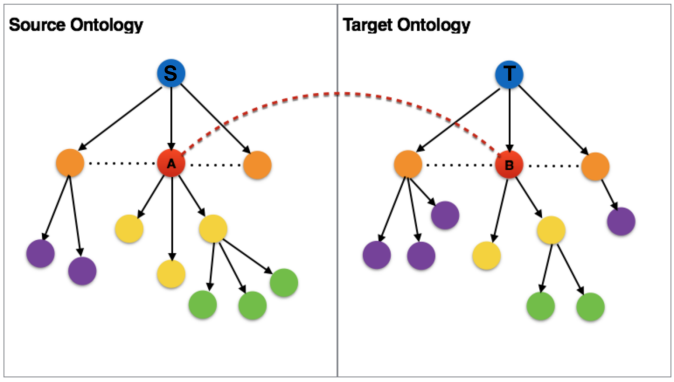


Fig. 3: Two ontology subgraphs showing a mapping between nodes A and B.

81%-100%. Figure 2 also highlights a mapping between two nodes, A that is a child of S and B that is a child of T.

Figure 3 shows schematically subgraphs of both ontologies with roots S and T, their children, among which there are subclasses A and B, the siblings of A and B, and their children (and grandchildren). To enable navigation along the ontologies, users should be able to traverse the ontology *vertically* from a node to its children but also *horizontally* from a node to its siblings.

The next question we address from the interface viewpoint is how to combine both types of navigation. We provide a list that represents the children of a node and a tree view that represents the siblings of the current node. The main panel of the user interface for the initial version of the prototype is shown in Figure 4. In the center area there are the two pie charts previously discussed.

Immediately left of the pie charts there is a list. When users click on a pie chart slice, the list contains the ontology nodes with confidence score within the corresponding range, sorted by the confidence score. Clicking on a node in the list leads to an update of the pie charts, as that node becomes the current node.

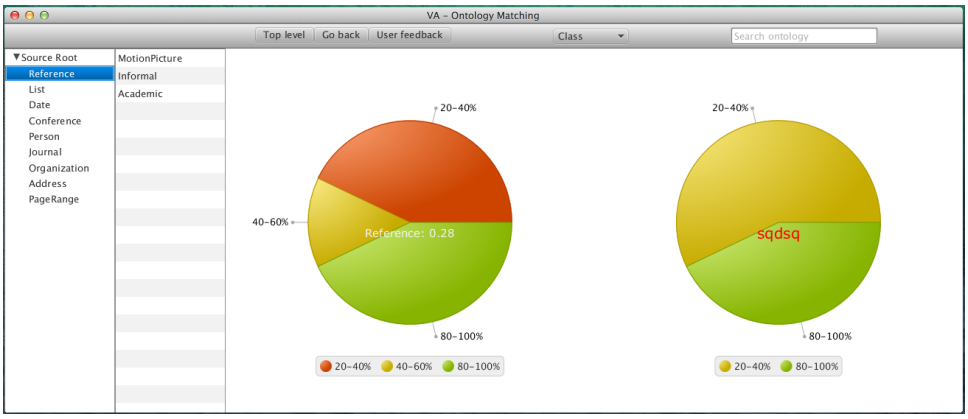


Fig. 4: Main panel, where *Reference* and *sqdsq* are concepts in the OAEI Benchmark Track.

The leftmost part of the interface contains the tree view. It shows all the siblings of the current node. When clicking on a node in the tree view, both the pie charts and the lists are updated, reflecting the change of the current node. On the top right there is a search box. Upon entering the name of the node in the search box, the left pie chart displays that source node and the right pie chart displays the target node that matches the source node. Accordingly, the tree and list view on the left are updated as well. In addition, for an easier navigation we provide additional functions such as "go to the top level", "go to the previous level", and "switch between class and property".

3.3 Algorithm Comparison

To compare the results of two matching algorithms, we need to visualize their results at the same time. We have therefore upgraded our user interface panel to load multiple results as shown in Figure 5. We are making full use of the containers in JavaFX to manage the visual elements within the available space. We use two tile panels to display two ontology pairs. The upper panel is the main panel (colored green) and the lower one is the sub panel (colored yellow). The tree view shows the siblings of the source node in the main panel and the list view shows the children of that source node. Figure 6 displays the schematic representation of the user interface, including the flow panel. The flow panel shows the algorithms we have loaded into the application. In this example, the main panel shows the results of the algorithm we loaded using the second button of the flow panel. When selected, the second button is colored green, so as to provide a color match with the main panel.

Users can choose any algorithm for the main panel or for the sub panel. The difference between the two panels is that all lists get updated according to the changes to the main panel.

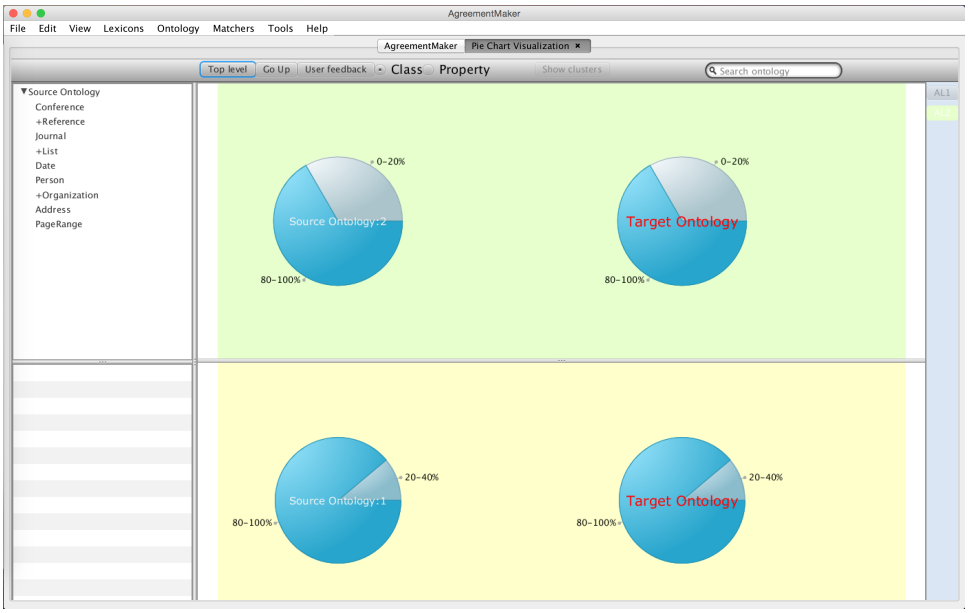


Fig. 5: Upgraded user interface to display multiple algorithm results.

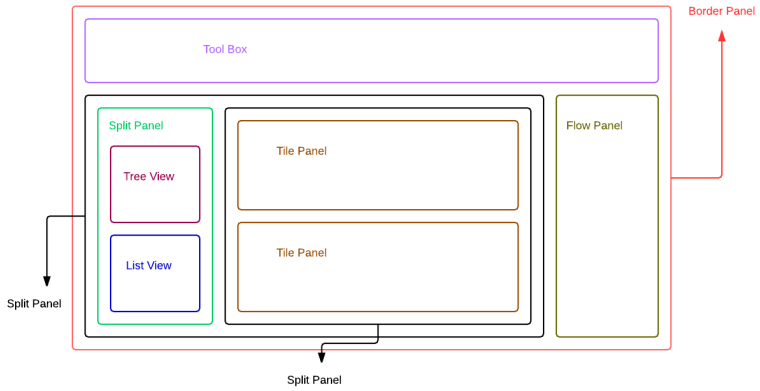


Fig. 6: Schematic representation of the upgraded user interface.

When the users click on a slice of the main pie chart, all features including all pie charts and lists are updated. To change the matching algorithm for the main panel, users only have to select another algorithm from the flow panel.

4 Interactive Ontology Matching

We list here our objectives for an interactive mechanism for matching ontologies that can assist users in a semi-automatic ontology matching process, where users provide feedback:

- Show candidate mappings for validation to the users; candidate mappings are determined automatically using quality measures [4, 14, 15].
- Register the validation choices made by the users.
- Allow for class and property navigation to assist users in their validation decisions.
- Allow users to search for a specific class and navigate through the classes.
- Support the creation of new mappings that are missed by the automatic process.

4.1 Interactive Workflow

Because automatic matching methods do not always provide complete or correct mappings, the combination of user validation with the automatic methods can lead to better results than the automatic methods alone.

The interactive workflow is shown in Figure 7. It shows a "user feedback loop" (UFL) strategy [14, 15] integrated with the visual analytics (VA) approach, in that the results provided by the user are fed back into the matching process and the user is helped by the interactive user interface [4]. In the figure, the visual

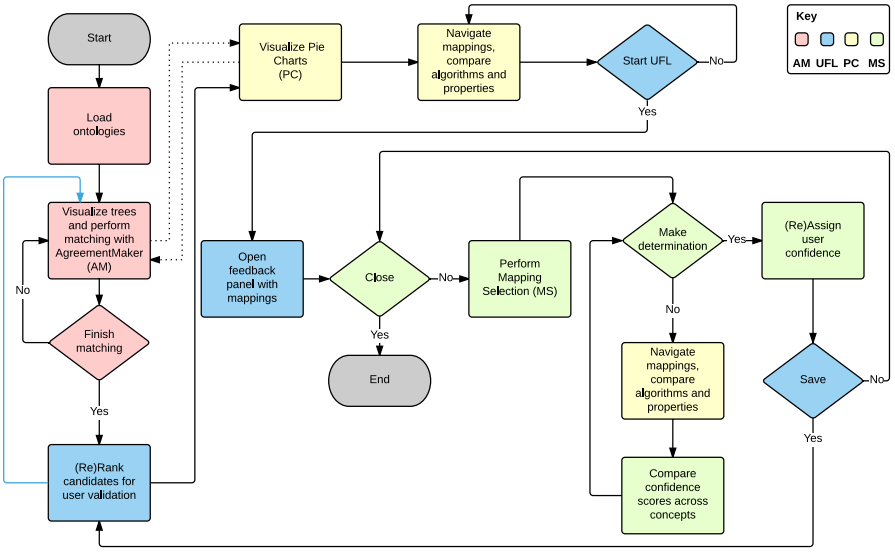
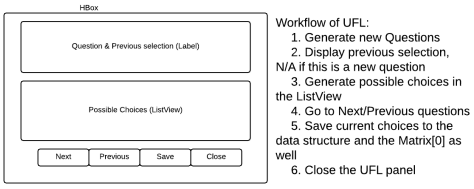


Fig. 7: Workflow of the interactive process.

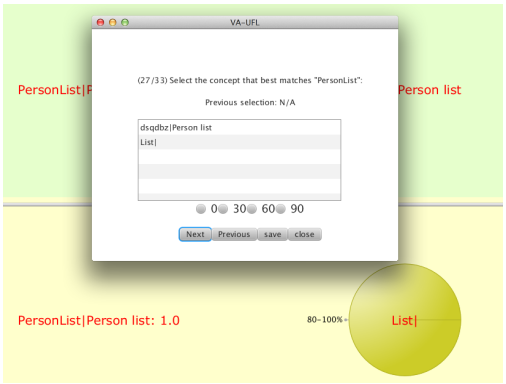
interfaces provided by AgreementMaker and the new interface communicate, so as to allow for complementary views. The user interface that is used to perform the mapping selection is an important component, which we describe next.

4.2 User Interface for Mapping Selection

The user interface for the mapping selection must display one or more mappings to be validated by the users. When it shows more than one mapping, users are asked to choose among them. Navigation using the main user interface (Figure 5) provides the confidence scores to assist users in making their selection. Figure 8a shows the schematic representation of the interface and Figure 8b shows a snapshot of its implementation.



(a) Schematic representation.



(b) Mapping selection, where users input their level of confidence (0-30-60-90).

Fig. 8: User interface.

4.3 Property Comparison

Automatic algorithms match classes according to a variety of lexical, syntactic, and structural criteria [2]. In addition, they may use other criteria, which can be incorporated into the automatic algorithms or visualized and presented to the users. For example, the properties associated with the classes can be considered [16]. In our user feedback loop strategy when allowing users to choose among mappings, we present the properties associated with the classes. Our interface displays both the confidence scores and floating panels that display the properties of each of the classes, as shown in Figure 9. We note that we only display once the panel associated with the source concept, *ConferenceEvent*.

4.4 Integration with AgreementMaker

Using the AgreementMaker system, the source and target ontologies are visualized side by side using a tree paradigm as shown in Figure 10. The control panel

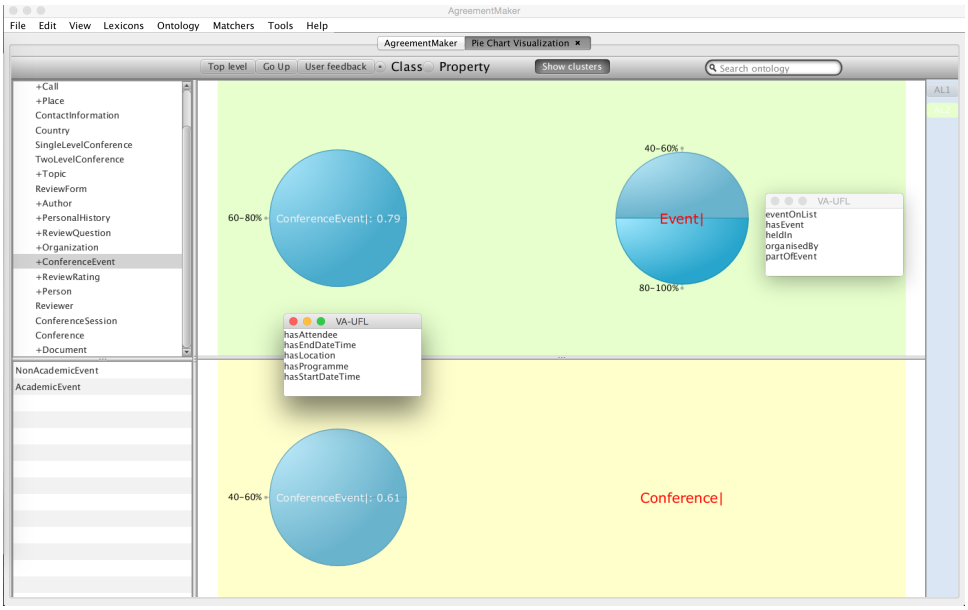


Fig. 9: Property clustering.

(at the bottom of the interface) allows for users to run a variety of matchers. In this example, two automatic matchers have been activated (the Parametric String Matcher and the Vector-based Multi-word Matcher) in addition to manual matching. The picture depicts the display of the two ontologies and the mappings obtained in this way.

Every set of mappings in AgreementMaker is represented by the MatchingTask class. The MatchingTask class contains the following elements: a matcher, its associated parameters (e.g., the confidence score threshold), and the mappings produced by the execution of the matcher. To open our visualization system, users can select the Pie Chart Visualization tab of the drop down menu, as shown in Figure 10. The key point in the integration of the pie chart visualization with AgreementMaker is that we pass all the MatchingTask instances from AgreementMaker to the pie chart visualization. After selecting the Pie Chart Visualization tab, another tab shows up and the pie charts will be initialized automatically (see Figure 5). Users can easily switch between the tree and pie chart visualizations by clicking on the available tabs at the top of the display.

5 Conclusions and Future Work

We devised a visualization tool for large ontology matching that integrates seamlessly with the widely used AgreementMaker system. It supports advanced navigation, interaction, and analysis and decision making features. In particular, for navigation, our tool supports the visual representation of the source and target ontologies and mappings between classes in those ontologies. It allows for the

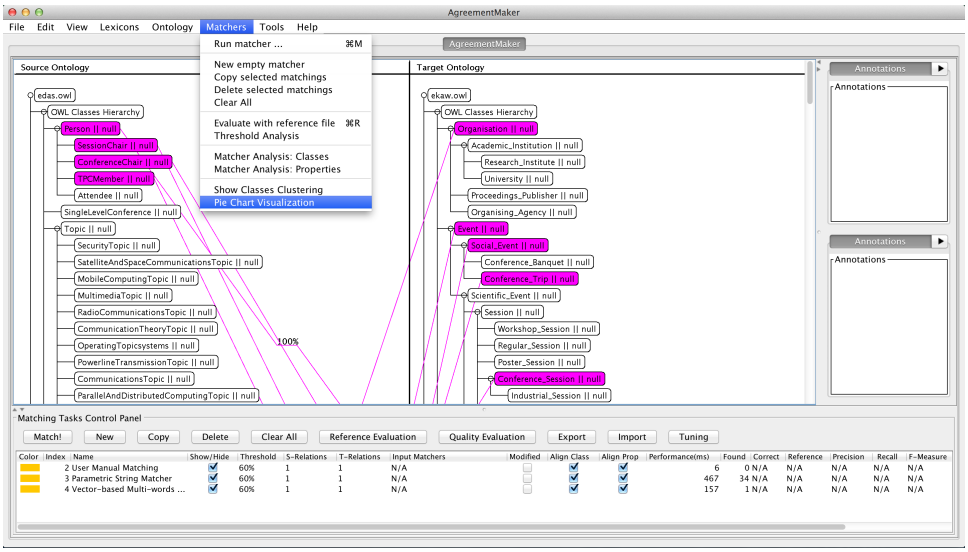


Fig. 10: Integration with AgreementMaker.

detailed access to pairs of classes to match, while providing ready access to other parts of the ontologies. Our tool displays confidence scores between classes and provides an overview of the confidence scores for the children of those classes.

For interaction, our tool supports user-driven navigation of classes and properties, the ability to search for a specific class, and to traverse the ontologies vertically (children of a class) and horizontally (siblings of a class).

Finally, for analysis and decision making, our tool displays several possible mappings to the users, so that they can choose among them, as part of a user feedback loop strategy that combines automatic with manual matching methods.

Clearly, there are several directions for future work. The first one is that we would like to extend the comparison of matching algorithms to more than two at a time. It may be the case that no single visualization strategy works separately, especially for very large ontologies. AgreementMaker already provides several different strategies [4, 2, 13]. Experiments would be needed to determine the usability and effectiveness of the different strategies when used separately or in coordination with one another.

Acknowledgments

This research was partially supported by NSF Awards IIS-1143926, IIS-1213013, and CCF-1331800. We would like to thank one of the anonymous reviewers, whose questions helped improve the final version of the paper.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (DE) (2007)
2. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB **2**(2) (2009) 1586–1589
3. Cruz, I.F., Palandri Antonelli, F., Stroe, C.: Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In: ISWC International Workshop on Ontology Matching (OM). Volume 551 of CEUR Workshop Proceedings. (2009) 49–60
4. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive User Feedback in Ontology Matching Using Signature Vectors. In: IEEE International Conference on Data Engineering (ICDE). (2012) 1321–1324
5. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively Learning Ontology Matching via User Interaction. In: International Semantic Web Conference (ISWC). Volume 5823 of Lecture Notes in Computer Science., Springer (2009) 585–600
6. Bertini, E., Lalanne, D.: Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery. SIGKDD Explorations Newsletter **11**(2) (May 2010) 9–18
7. Ivanova, V., Lambrix, P.: User Involvement for Large-Scale Ontology Alignment. In: International Workshop on Visualizations and User Interfaces for Knowledge Engineering and Linked Data Analytics. (2014)
8. Lanzenberger, M., Sampson, J.: AlViz - A Tool for Visual Ontology Alignment. In: Conference on Information Visualization (IV), IEEE Computer Society (2006) 430–440
9. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The AgreementMakerLight ontology matching system. In: International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), Springer (2013) 527–541
10. Pesquita, C., Faria, D., Santos, E., Neefs, J.M., Couto, F.M.: Towards Visualizing the Alignment of Large Biomedical Ontologies. In: Data Integration in the Life Sciences (DILS). (2014)
11. Falconer, S.M., Storey, M.A.D.: A Cognitive Support Framework for Ontology Mapping. In: International Semantic Web Conference/Asian Semantic Web Conference (ISWC/ASWC). Volume 4825 of Lecture Notes in Computer Science., Springer (2007) 114–127
12. El Jerroudi, Z., Ziegler, J.: iMERGE: Interactive Ontology Merging (poster). In: International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns (EKAW). (2008)
13. Aurisano, J., Nanavaty, A., Cruz, I.F.: Visual Analytics for Ontology Matching Using Multi-Linked Views. In: ISWC International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data (Voila!). (2015)
14. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go Multi-User Feedback Model for Ontology Matching. In: International Conference on Knowledge Engineering and Knowledge Management (EKAW). Springer (2014) 80–96
15. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Quality-Based Model for Effective and Robust Multi-User Pay-As-You-Go Ontology Matching. In: Semantic Web Journal. IOS Press (2015)
16. Hu, W., Qu, Y.: Falcon-AO: A Practical Ontology Matching System. Web Semantics: Science, Services and Agents on the World Wide Web **6**(3) (2008) 237–239

FedViz: A Visual Interface for SPARQL Queries Formulation and Execution

Syeda Sana e Zainab[1], Muhammad Saleem[2], Qaiser Mehmood[1], Durre Zehra[1], Stefan Decker[1], and Ali Hasnain[1]

[1] Insight Centre for Data Analytics, National University of Ireland, Galway
firstname.lastname@insight-centre.org
[2] Universität Leipzig, IFI/AKSW, PO 100920, D-04009 Leipzig
{lastname}@informatik.uni-leipzig.de

Abstract. Health care and life sciences research heavily relies on the ability to search, discover, formulate and correlate data from distinct sources. Over the last decade the deluge of health care life science data and the standardisation of linked data technologies resulted in publishing datasets of great importance. This emerged as an opportunity to explore new ways of bio-medical discovery through standardised interfaces. Although the Semantic Web and Linked Data technologies help in dealing with data integration problem there remains a barrier adopting these for non-technical research audiences. In this paper we present FedViz, a visual interface for SPARQL query formulation and execution. FedViz is explicitly designed to increase intuitive data interaction from distributed sources and facilitates federated as well as non-federated SPARQL queries formulation. FedViz uses FedX for query execution and results retrieval. We also evaluate the usability of our system by using the standard system usability scale as well as a custom questionnaire, particularly designed to test the usability of the FedViz interface. Our overall usability score of **74.16**% suggests that FedViz interface is easy to learn, consistent, and adequate for frequent use.

Keywords: SPARQL, Life Sciences (LS), Query Federation, Visual Query Formulation

1 Introduction

The researchers in health care, life sciences and biomedical (also known as domain users) adopted Semantic Web and Linked Data technologies due to the data integration challenges faced as a result of excessive data produced [6,16]. Different researchers recommended the use of SPARQL services for publishing biomedical resources [2,20,19]. The use of these technologies facilitate the domain users for issuing structured SPARQL queries over highly heterogeneous data spread over diverse data sources [5,1]. Such structured queries are vital, not only in order to query relevant data regarding different entities e.g. Drugs, Molecules and Pathways but also to drive meaningful biomedical correlations such as Drug Drug Interactions and Protein Protein Interactions etc. Such retrieved information can subsequently be applied to various bioinformatics tasks such as functional analysis, protein modelling or image analysis. As pointed out earlier that

in the most of cases, the required information to draw any biological correlation or to answer a biological question involve querying multiple data source, provided by different providers, sometimes available in different format with different accessing mechanism. Meaningful biological query such as *"Find out the Diseases that causes due to the deficiency of Iodine"* can only be answered by querying and aggregating data from multiple reliable data sources. The use of Semantic Web and Linked Data technologies are commonly exploited by computer scientists, who can formulate structured SPARQL queries to access data from different SPARQL endpoints, the ultimate end-users and the domain experts either biologists or clinical researchers, remain unable to assemble complex queries in order to access such data [8]. Making complex SPARQL queries to drive necessary information to support clinical experiments and observations poses a barrier in health care and life sciences domain that confront the adoption and acceptance of such technologies. Moreover, even for computer scientists, assembling a federated SPARQL query is time-consuming and technical process since it requires the knowledge of underlying datasets schema and the connectivity between the datasets [9,10]. An alternative to this is an intuitive and interactive platform that can facilitate domain users to assemble complex but meaningful SPARQL query through visual interface. To this end, we introduce FedViz which enables a user to formulate and execute complex federated SPARQL queries using intuitive visual query interface. FedViz allows user to select concepts and properties from multiple datasets using nodes and edges, assemble SPARQL query in a background independent of user involvement and allow users to edit the resultant SPARQL query before sending it to the SPARQL query federated engine. Assembled query is executed through FedX- a state of the art engine [22], that federates the query to relevant data sources and retrieves the results. The choice of FedX was due to the fact it can execute both federated (both SPARQL 1.0 and SPARQL 1.1) and non federated queries. At present, six real time biomedical data sources, i.e., Kegg, Drugbank, DailyMed, Medicare, Sider, and Diseasome are selected to visually construct the SPARQL query. However, FedViz can be generalise to any set of datasets.

The remaining part of this paper is organised as follows: we highlight the related work in section 2. Later we present the motivational use case in section 3. We introduce our methodology and FedViz salient features in section 4. Subsequently, we present a thorough evaluation of FedViz in section 5. We finally conclude the paper with an overview of future work.

2 Related work

Several approaches have been proposed for Visual query formulation over Linked data. *Form-based querying* is one of the famous paradigm, where *Form elements* (i.e. filters, variables, identifiers) are used for query formulation. Example of this approach is SPARQLViz [3]. However it is less flexible and allows only those users with some knowledge of RDF and SPARQL language. In *Graph-based querying* paradigm query is formulated using node-link diagrams and this approach is more flexible as compared to *Form-based paradigm* and requires the RDF notations of subject-predicate-object cause barrier for users with limited semantic web knowledge. Examples for such approaches include NITELIGHT [15], iSPARQL[1], RDF-GL [11] and ReVeaLD [13]. QueryVOWL[7] uses

[1] http://oat.openlinksw.com/isparql/

Listing 1.1: Find all the drugs and their interactions for curing thyroid disease.

```
PREFIX drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX diseasome:<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/>
Select Distinct ?interactionDrug1 ?interactionDrug2 ?text ?name
WHERE
{
?Drugbank0 a drugbank:drug_interactions;
drugbank:interactionDrug1 ?interactionDrug1;
drugbank:interactionDrug2 ?interactionDrug2;
drugbank:text ?text.
?interactionDrug1 drugbank:possibleDiseaseTarget ?possibleDiseaseTarget.
?possibleDiseaseTarget diseasome:name ?name.
FILTER (regex(?name, "thyroid" , "i" ) )
}
LIMIT 100
```

specific language and graph database. Most of aforementioned available systems focused on query formulation using specific graphs, available predicate links and user may need sufficient SPARQL knowledge using such system. FedViz is a step towards interactively and intuitively formulating federated SPARQL queries using class and property links visually presented per dataset.

3 Motivation

We believe FedViz enables a variety of use cases, of which one is explained as follows: **Drug-Drug Interaction for Medication of Certain Disease:** When patients are diagnosed with certain disease, a large number of drugs are associated with that depending upon its stage and condition. It is imperative that physician are thoroughly educated about drug-drug interaction before prescription for certain disease. Take `hypothyroidism` for example. It is a disease which results from an under-active thyroid, leading to the necessity of taking extrinsic thyroxine hormone to maintain normal bodily functions. One treatment option for hypothyroidism is using `Levothyroxine`, which is a synthetic thyroid hormone similar to T4 hormone, which is intrinsically produced by the thyroid gland, deficiency of which leads to the disease in the first place. `Levothyroxine` has many drug interactions, especially with the warfarin family and similar drugs, including `Acenocoumarol`. It is an anticoagulant that functions as a Vitamin K antagonist, and so controls clot formation in the body. Simultaneous use of `Levothyroxine` with `Acenocoumarol` can sensitise the body to the latter, which may put the patient at an increased risk of bleeding. This is just an example how FedViz can be used to monitor interactions of a drug, in this particular case `Levothyroxine`, by creating a visual query, making it easier for the physician to have a comprehensive look at the potential contraindications to using the drug in particular patients (Listing 1.1).

4 FedViz

FedViz is an online application that provides Biologist a flexible visual interface to formulate and execute both federated and non-federated SPARQL queries. It translates the visually assembled queries into SPARQL equivalent and execute using query engine.

At present, FedViz visualises Life Sciences datasets and facilitates complex query formulation and execution in order to draw meaningful biological co-relations including drug-drug interaction, drug-disease interaction and drug-side effect correlations. Through FedViz Biologist can formulate simple queries that typically involve single or multiple concepts from one dataset as well as complex federated queries that might involve more than one datasets with multiple constraints.

4.1 Methodology

Our methodology consists of two steps namely: 1) building visual interface and 2) result retrieval using query engine (Figure 1).

Building visual interface A concise graphical representation is needed to display datasets to facilitate biologist in order to formulate query. We chose the concept map approach [12] for building the visual interface, which is a graphical method representing the relationship between nodes and links, and has been used in various domains for organising knowledge [24]. Using this approach in FedViz, we represent concepts as big circular nodes (drugs, disease etc) and properties as small circular nodes (protein sequence, possible disease target etc). As mentioned earlier, currently FedViz contains six datasets and their concepts with associated properties are visualised for query formulation also known as catalogue (Fig 1). Each dataset represented in catalogue is marked with unique colour. The nodes are modelled as objects in a two-dimensional system using a force-directed layout[23]. In force-directed layout nodes repel each other based on their sizes that prevents overlapping and increases concept-property visibility to end-user.

Result Retrieval Using Query Engine To process the FedViz query request, FedX the state of the art efficient SPARQL query federation engine [18] is chosen to execute both federated (SPARQL 1.1 and SPARQL 1.0) and non-federated queries. FedViz provides the set of required SPARQL endpoints (i.e., data sources) URLs in order to enable FedX's query execution. Overall, the query execution works as follow: (1) FedViz formulate SPARQL query and sends to FedX, (2) FedX executes the query and sends back the results to FedViz, (3) FedViz presents the results to end user.

Technologies FedViz is browser-based client application that provides biologist a flexible front-end. To build this application variety of web technologies are used including HTML5, CSS, JavaScript, JQuery[2], Java Servlet, SVG[3], AJAX[4] and JSON[5]. The datasets visualisation is based on SVG (Scaler Vector Graphics) with Javascript usage. In catalogue, datasets are represented in JSON format and displayed as nodes (Concept and Properties). The communication between the client query and federated query engine(FedX) has done by AJAX calls through middle layer. Open source Javascript library D3.js[4] is used to implement force-directed layout for datasets visualisation.

[2] https://jquery.com/
[3] www.w3schools.com/svg/
[4] http://api.jquery.com/jquery.ajax/
[5] http://json.org/

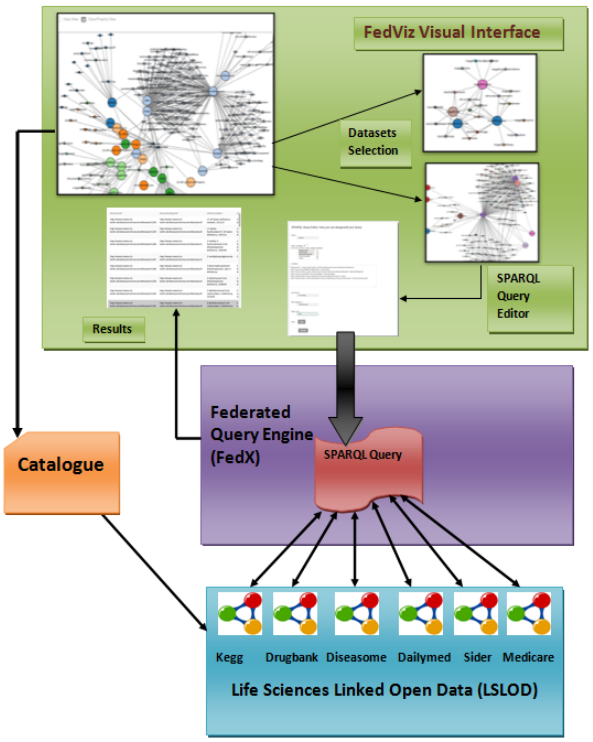


Fig. 1: FedViz Architecture Diagram

Availability The FedViz application can be accessed at http://srvgal86.deri.ie/FedViz/index.html. Example queries both simple (include single dataset) and complex (include more than single dataset) are provided at https://goo.gl/AOJGpu.

4.2 Datasets

Current version of FedViz supports a total of 6 real-world datasets. All the datasets were collected from Life Sciences domains. We began by selecting two real world datasets from Fedbench [21] namely Drugbank[6] a knowledge base containing information of drugs, their composition and their interactions with other drugs and Kegg Kyoto Encyclopedia of Genes and Genomes (KEGG)[7] which contains further information about chemical compounds and reactions with a focus on information relevant for geneticists. Apart from aforementioned selected datasets four other datasets were chosen that had connectivity with the existing ones that enabled us to include real federated queries. These datasets include Sider[8]- that contains information on marketed drugs and their

[6] http://www.drugbank.ca/
[7] http://www.genome.jp/kegg/
[8] http://wifo5-03.informatik.uni-mannheim.de/sider/

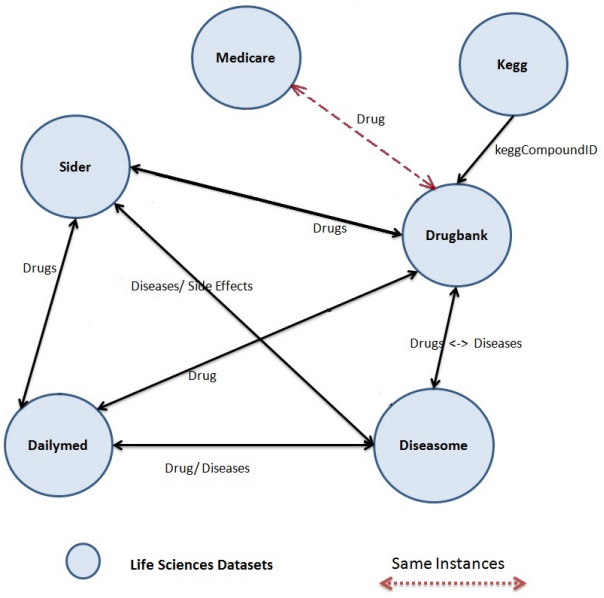


Fig. 2: Datasets Connectivity.

adverse effects, Diseasome[9]- that publishes a network of 4,300 disorders and disease genes linked by known disorder-gene associations for exploring all known phenotype and disease gene associations, indicating the common genetic origin of many diseases., Dailymed[10]- provides information about marketed drugs including the chemical structure of the compound, its therapeutic purpose, its clinical pharmacology, warnings, precautions, adverse reactions, over dosage etc., and Medicare[11]. Figure 2, shows the topology of all 6 datasets while some other basic statistics like the total number of triples, the number of resources, predicates and objects, as well as the number of classes and the number of links can be found in table 1.

4.3 Query Formulation

In this section, an example scenario is discussed to demonstrate our visual query formulation process.

Drug-Disease and Drug-Compound interaction: *Drugs with their compound mass for curing disease Anemia.* This query requires data integration from Drugbank (containing drugs information), Diseasome (containing disease information) and Kegg(containing compound mass information) and can be formulated by using the following step-by-step approach (ref., Fig. 3):

[9] http://wifo5-03.informatik.uni-mannheim.de/diseasome/
[10] http://dailymed.nlm.nih.gov/dailymed/index.cfm
[11] http://wifo5-03.informatik.uni-mannheim.de/medicare/

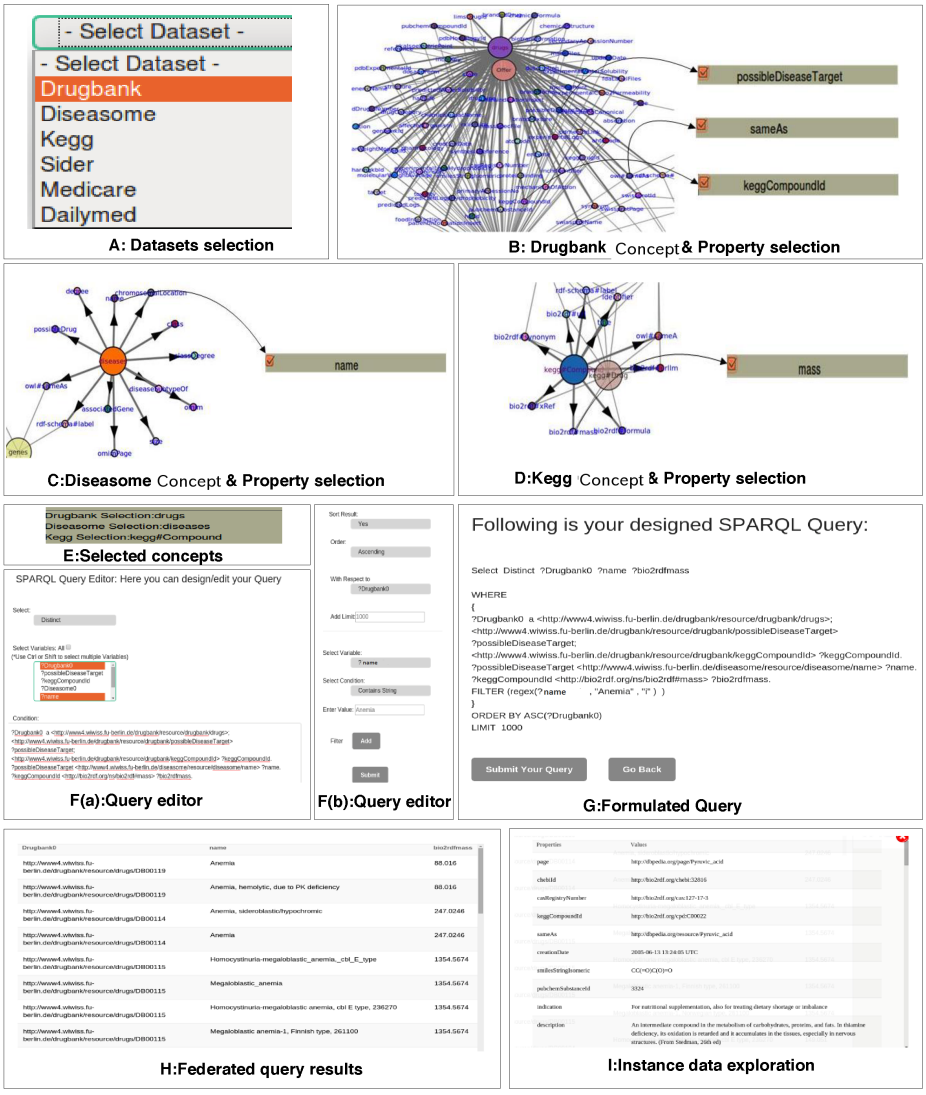


Fig. 3: Federated query formulation using FedViz

Dataset	Triples	Subjects	Predicates	Objects	Classes
DrugBank	517023	19693	119	276142	8
Kegg	1090830	34260	21	939258	4
Dailymed	162972	10015	28	67782	6
Diseasome	72445	8152	19	27704	4
Sider	101542	2674	11	29410	4
Medicare	44500	6825	6	23308	3
Total	1989312	81619	204	1363604	29

Table 1: Dataset Statistics

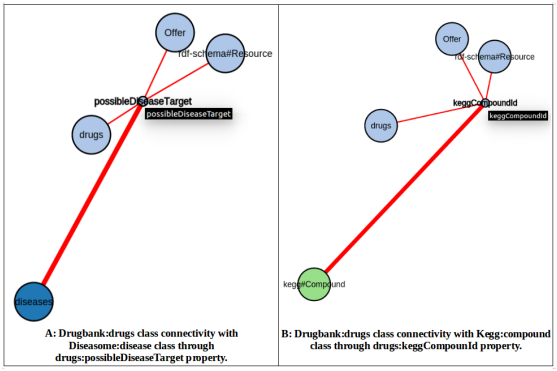


Fig. 4: Datasets Class visualisation view assign each dataset with unique colour. Light Blue: Drugbank, Dark Blue: Diseasome and Light Green: Kegg. Connectivity between Drugbank:drugs with Diseasome:disease through drugs:possibleDiseaseTarget property (Fig 4-A). Connectivity between Drugbank:drugs with Kegg:compounds through drugs:keggCompoundId property (Fig 4-B).

1. The first step is to identify how Drugbank, Diseasome and Kegg datasets are connected to each other? This connectivity (i.e., via classes `drugbank:drug`, `diseasome:disease` and `kegg:compound` can be found by using the Class visualisation view of FedViz that shows all classes of datasets along with there connectivity (ref., Fig. 4).
2. User selects Drugbank from the Datasets Selection box (window A).
3. The visualisation for Drugbank dataset can be seen in window B where he selects `drugbank:drug` class and its properties(i.e., `drugs:possibleDiseaseTarget` and `drugs:keggCompoundId`).
4. Step 2 and 3 are now followed for Diseasome dataset, i.e., select `diseasome:disease` class and it's `name` property (window C) and for Kegg dataset, i.e., select `kegg:compound` class and it's `mass` property (window D).
5. Selected Concepts are shown in status bar (window E).
6. Next, FedViz SPARQL Query Editor allows user to add constraints to the formulated federated query such as select projection variables, apply SPARQL `LIMIT`,

`FILTER`(in this scenario disease name Anemia), `ORDERY BY` clauses, and can further edit the query according to his choice (window Fa, Fb).
7. The final query can be seen on submission (window G).
8. Query is executed over FedX and the retrieved results are displayed by FedViz (Result window H).
9. Finally, by selecting any URI from the retrieved result, FedViz can provide detailed information regarding that instance (Data Exploration window I).

4.4 Query Execution

On dispatching from FedViz, SPARQL query is received and handled by an intermediate layer (IL) built on top of FedX [22]. The IL acts as an adopter, which allows the FedX to communicate with outer world (i.e, Web). FedX requires the set of endpoints URLs as input to query execution engine. The FedViz request incorporates the set of endpoints required by the query. The IL forwards the endpoints to FedX query engine by selecting endpoints from request. FedX executes a SPARQL *ASK* requests on set of endpoints. Furthermore, FedX optimise the query by splitting it into sub-queries. The selected endpoints are requested to run these sub-queries to generate the results. Finally, all the retrieved results from various sub-queries are integrated and displayed through FedViz interface.

5 Evaluation

The goal of our evaluation is to quantify the usability and usefulness of FedViz graphical interface. We evaluate the usability of the interface by using the standard *System Usability Scale* (SUS) [14] as well as a customised questionnaire designed for the users of our system. In the following, we explain the survey outcomes.

5.1 System Usability Scale Survey

In this section, we explain the SUS questionnaire[12] results. This survey is more general and applicable to any system to measure the usability. The SUS is a simple, low-cost, reliable 10 item scale that can be used for global assessments of systems usability[14,17]. As of 10th July 2015, 15 users[13] including researchers and engineers in Semantic Web were participated in survey. According to SUS, we achieved a mean usability score of **74.16**% indicating a high level of usability according to the SUS score. The average scores (out of 5) for each survey question along with standard deviation is shown in Figure 5.

The responses to question 1 (average score to question 1 = 3.8 ± 0.86) suggests that FedViz is adequate for frequent use. The responses to question 3 indicates that FedViz is easy to use (average score 4 ± 0.84) and the responses to question 7 (average score 4.06

[12] SUS survey can found at: http://goo.gl/forms/bhReuNgd6O

[13] Users from AKSW, University of Leipzig and INSIGHT Centre, National University of Ireland, Galway. Summary of the responses can be found at: https://goo.gl/ZOrJx9

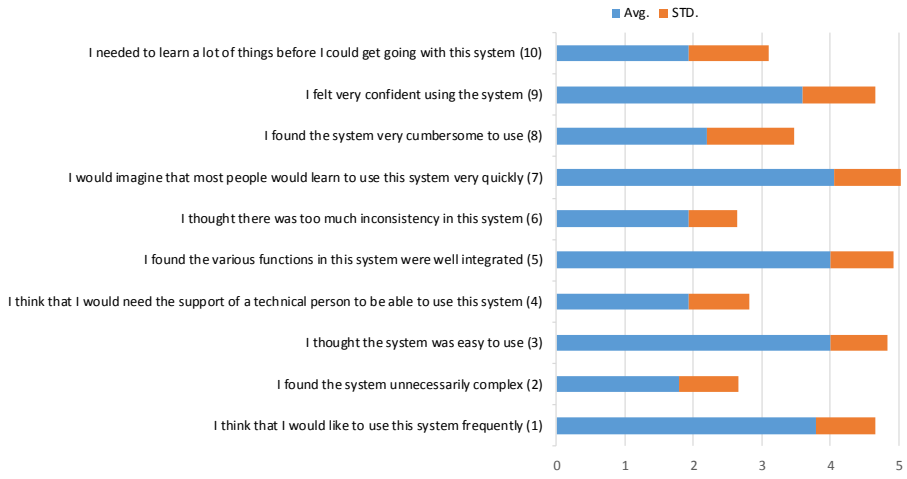


Fig. 5: Result of usability evaluation using SUS questionnaire.

$\pm$ 0.96) suggests that most people would learn to use this system very quickly. However, the slightly higher standard deviation to question 9 (standard deviation = $\pm$ 1.05) and question 10 (standard deviation = $\pm$ 1.16) suggest that we may need a user manual to explain the different functionalists provided by the FedViz interface.

5.2 Custom survey

This survey[14] was particularly designed to measure the usability and usefulness of the different functionalists provided by FedViz. In particular, we asked users to formulate both federated and non-federated SPARQL queries and share their experience through question 10 and question 11. As of 10th July 2015, 10 researchers including Computer Scientist[15] and Bioinformaticians were participated in survey. The average scores (out of 5 with 1 means strongly disagree and 5 means strongly agree) for each survey question along with standard deviation is shown in Figure 6. The average scores to question 10 (i.e., 4.2 $\pm$ 0.91) and question 11 (i.e., 3.9 $\pm$ 0.73) show that most of the user feel confident in formulating simple and federated queries, respectively. The responses to question 2 (average score = 4.4 $\pm$ 0.69) suggests that navigating on different datasets are much easy by using FedViz "Selection Box". A slightly lower scores to question 7 (average score = 3.5 $\pm$ 0.70) suggests that we need to further improve the datasets visualisation component of the FedViz.

As an overall usability evaluation, our SUS and custom surveys outcome suggest that FedViz interface is easy to use, consistent, adequate for frequent use, easy to learn, and the various functions in the system are well integrated.

[14] Custom survey can be found at: http://goo.gl/forms/2DWvK2qYsV
[15] Summary of the responses can be found at: https://goo.gl/tT8TXF

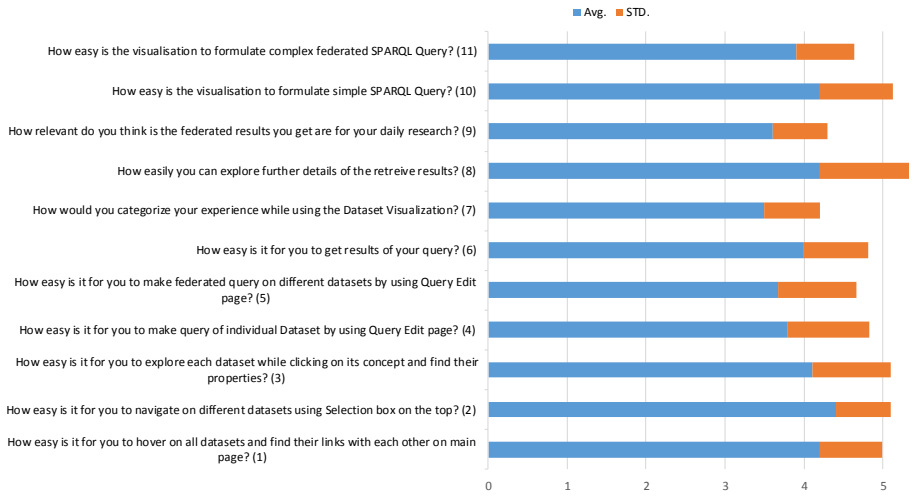


Fig. 6: Result of usefulness evaluation using our custom questionnaire.

6 Conclusion and Future Work

In this paper we introduce FedViz as a online interface for SPARQL query formulation and execution. We evaluate our approach and usability of our system using the standard system usability scale as well as through domain experts. Our preliminary analysis and evaluation revels the overall usability score of 74.16%, concluding FedViz an interface, easy to learn and help users formulating complex SPARQL queries intuitively. As a future work we aim to extend FedViz with Faceted browsing and also provide visualization at entity level e.g, Genes and Molecules where user can see the Gene sequences and 3D structure for Molecules.

7 Acknowledgement

The work presented in this paper has been partly funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2).

References

1. J. Almeida, H. Deus, and W. Maass. Development of integrative bioinformatics applications using cloud computing resources and knowledge organization systems (kos). *Nature proceedings*, 2011.
2. F. Belleau, M.-A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.
3. J. Borsje and H. Embregts. Graphical query composition and natural language processing in an rdf visualization interface. *Erasmus School of Economics and Business Economics, Vol. Bachelor. Erasmus University, Rotterdam*, 2006.

4. M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.
5. B. Chen, D. J. Wild, Q. Zhu, Y. Ding, X. Dong, M. Sankaranarayanan, H. Wang, and Y. Sun. Chem2bio2rdf: A linked open data portal for chemical biology. *arXiv preprint arXiv:1012.4759*, 2010.
6. H. Chen, T. Yu, and J. Y. Chen. Semantic web meets integrative biology: a survey. *Briefings in bioinformatics*, 14(1):109–125, 2013.
7. F. Haag, S. Lohmann, S. Siek, and T. Ertl. Visual querying of linked data with QueryVOWL. In *Joint Proceedings of SumPre 2015 and HSWI 2014-15*. CEUR-WS, to appear.
8. A. Hasnain, R. Fox, S. Decker, and H. F. Deus. Cataloguing and linking life sciences LOD Cloud. In *EKAW*, 2012.
9. A. Hasnain, M. R. Kamdar, P. Hasapis, D. Zeginis, C. N. Warren Jr, et al. Linked Biomedical Dataspace: Lessons Learned integrating Data for Drug Discovery. In *International Semantic Web Conference (In-Use Track), October 2014*, 2014.
10. A. Hasnain, S. S. E. Zainab, M. R. Kamdar, Q. Mehmood, C. Warren Jr, et al. A roadmap for navigating the life scinces linked open data cloud. In *International Semantic Technology (JIST2014) conference*, 2014.
11. F. Hogenboom, V. Milea, F. Frasincar, and U. Kaymak. Rdf-gl: a sparql-based graphical query language for rdf. In *Emergent Web Intelligence: Advanced Information Retrieval*, pages 87–116. Springer, 2010.
12. D. H. Jonassen, K. Beissner, and M. Yacci. *Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge*. Psychology Press, 1993.
13. M. R. Kamdar, D. Zeginis, A. Hasnain, S. Decker, and H. F. Deus. Reveald: A user-driven domain-specific interactive search platform for biomedical research. *Journal of biomedical informatics*, 47:112–130, 2014.
14. J. R. Lewis and J. Sauro. The factor structure of the system usability scale. In *HCD*. 2009.
15. A. Russell and P. Smart. Nitelight: A graphical editor for sparql queries. 2008.
16. A. Ruttenberg, T. Clark, W. Bug, M. Samwald, O. Bodenreider, H. Chen, D. Doherty, K. Forsberg, Y. Gao, V. Kashyap, et al. Advancing translational research with the semantic web. *BMC bioinformatics*, 8(Suppl 3):S2, 2007.
17. M. Saleem, M. R. Kamdar, A. Iqbal, S. Sampath, H. F. Deus, and A.-C. N. Ngomo. Big linked cancer data: Integrating linked tcga and pubmed. *Web Semantics: Science, Services and Agents on the World Wide Web*, 27:34–41, 2014.
18. M. Saleem, Y. Khan, A. Hasnain, I. Ermilov, and A.-C. N. Ngomo. A fine-grained evaluation of sparql endpoint federation systems. *Semantic Web Journal*, 2014.
19. M. Saleem, S. S. Padmanabhuni, A.-C. N. Ngomo, A. Iqbal, J. S. Almeida, S. Decker, and H. F. Deus. Topfed: Tcga tailored federated query processing and linking to lod. *Journal of Biomedical Semantics*, 2014.
20. M. Samwald, A. Jentzsch, C. Bouton, C. S. Kallesøe, E. Willighagen, J. Hajagos, M. S. Marshall, E. Prud'hommeaux, O. Hassanzadeh, E. Pichler, et al. Linked open drug data for pharmaceutical research and development. *Journal of cheminformatics*, 3(1):19, 2011.
21. M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte, and T. Tran. Fedbench: A benchmark suite for federated semantic data query processing. In *The Semantic Web–ISWC 2011*, pages 585–600. Springer, 2011.
22. A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. Fedx: Optimization techniques for federated query processing on linked data. In *The Semantic Web, ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 601–616. 2011.
23. R. Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013.
24. J. D. Wallace and J. J. Mintzes. The concept map as a research tool: Exploring conceptual change in biology. *Journal of research in science teaching*, 27(10):1033–1052, 1990.

Cognitive-based Visualization of Semantically Structured Cultural Heritage Data

Kalliopi Kontiza, Antonis Bikakis, Rob Miller

Department of Information Studies, University College London, UK
{k.kontiza.12,a.bikakis,rsm}@ucl.ac.uk

Abstract. We present preliminary findings regarding the increasing use of InfoVis tools and semantically structured data by cultural heritage institutions. This sector faces a number of challenges in developing best practices for publishing Linked Data, including the presentation of their digital cultural heritage collections and the visualization of their multidimensional hidden histories. We suggest that, as these institutions' interest in Semantic Web technologies grows and associated applications are more widely adopted, the need to provide InfoVis tools for efficient overview and exploration of cultural data increases. We postulate that changes in the paradigms for interaction with cultural datasets are also needed, with more focus on users' needs and cognitive processes. We suggest that by taking into account human information processes, better cognitive support can be introduced via InfoVis tools for Linked Data, thus reducing the cognitive load experienced by users.

1 Introduction

The aim of this position paper is to give a brief summary of the current state-of-the-art as regards the use of Information Visualisation (InfoVis) for semantically structured cultural heritage (CH) data, and to suggest some directions for research and development in this area. The Semantic Web (SW), also known as Web 3.0, is the new environment in which cultural digital resources will be exploited. The Resource Description Framework (RDF), which integrates a variety of applications using XML or other machine-readable formats for syntax and URIs for naming, has become the standard data model for describing semantically structured data using statements in the form of triples, which describe resources and can be considered as metadata. Linked Data (LD) is a way of publishing structured data that allows metadata to be connected and enriched, so that different representations of the same content can be identified, and links between related resources can be made. Although the term LD is often used as if it is a specific, well defined technology, it could be better understood as a set of best practices [8]. These aim to get data published on the web in a way that is readable, interpretable and usable by machines, by ensuring that their meaning is explicitly defined by a string of words and markers [6].

LD practices are starting to be introduced as novel and promising approaches to address the specific challenges that the CH sector encounters when publishing "collection" data on the Web [7]. Museums as memory organizations are key players in

preserving CH Tangible[1] objects by storing them with attached metadata. Despite efforts amongst the CH community to analyze the needs of their online 'audiences', no clear understanding about these users and their expectations has yet been gained [8]. Of course, publishing CH contents on the Web cannot replace the physical experience of visiting a museum or an exhibition in reality. But the key question is whether online content, when combined with appropriate Visualization tools, can act as a complementary alternative for access and exploration of cultural data. The hope is that the structure and semantics of RDF representations and standard ontologies, combined with InfoVis techniques, can lead to new ways of thinking about aspects of cognition as emergent properties of the interaction of people with cultural artifacts [16].

We take as a starting point the definition of InfoVis as "the use of computer supported interactive visual representation of data to amplify cognition" [3]. Our long term aim is to investigate why and how InfoVis can assist people in understanding CH information, in particular by exploiting the structure and the semantics of RDF representations and relevant ontologies. More specifically we aim to develop a cognitively based InfoVis model that is geared towards revealing the various temporal, spatial, contextual, conceptual links between different cultural artifacts, their creators and associated events, building upon standard ontologies and vocabularies. From this we hope to develop a set of principles for the design of interactive visual interfaces for exploring and understanding the semantically encoded data used in museum collections.

The rest of the paper is structured as follows. Section 2 provides an overview of some relevant SW terminology and then describes a number of current initiatives that use InfoVis Tools to present semantically structured CH data. Section 3 provides arguments in favor of taking greater account of theories of human cognition when designing CH InfoVis Tools, in order to provide efficient exploration of semantically encoded cultural datasets. Finally, Section 4 summarizes the preceding discussion, and sets out an agenda for future research.

2 Semantic Web and Linked Cultural Data

2.1 Background

Tim Berners-Lee has re-introduced the SW as "a new layer of metadata being build inside the Web" [7]. One of the main principles behind its architecture is that it offers users the ability to share knowledge by constructing meaningful representations on the Web [1]. RDF was published as the first SW standard by the World Wide Web Consortium (W3C) in 1999[2]. But it was only around 2005 when the ideas of SW and LD started to gain momentum and the SW and LD research communities focused on enhancing existing large datasets using simple RDF and lightweight ontologies.

The use of lightweight ontologies based on RDF is one of the reasons for the success of LD in the CH sector. The Dublin Core vocabulary[3] for example, which became popular for expressing metadata as RDF, provides the basis for the Europeana Data Model

[1] Tangible cultural heritage consists of concrete cultural objects, such as artifacts, works of art, buildings and books

[2] http://www.w3.org/RDF/

[3] http://dublincore.org/documents/dcmi-terms/

(EDM)[4] used by Europeana[5] for gathering and publishing metadata for thousands of digitized cultural collections [10]. But even more sophisticated models are provided as simple RDF Schema ontologies. One example is the CIDOC Conceptual Reference Model (CIDOC-CRM)[6], a core ontology for describing the semantics of schema and data structure elements in CH documentation [7]. This was developed as an ambitious attempt to link museum objects across collections ontologically [4]. The CIDOC-CRM aims to facilitate "the integration, mediation and interchange of heterogeneous cultural information, by capturing their richness" [4].

Heath [7] enumerates some of the content related challenges that CH institutions are faced with in order to publish their data collections: their content can be multi-format (text, images, audio, video, collection items, learning objects), multi-topical (art, history, artifacts, traditions, etc.), multi-lingual, multi-cultural and multi-targeted (targeted to both laymen and experts, various ages, etc.). SW standards and LD as best practice could provide a shared basis with which to facilitate content-related and cross-domain semantic interoperability.

Progressive steps have been taken towards LD online publishing. Following the LD principles[7] Tim Berners-Lee also introduced a five-star rating scheme for LD at increasing levels of openness and linkage[8]. For a linked dataset to be accepted by the Open Linked Data Project[9] it must not be subject to commercial licenses or use restrictions.

2.2 Current Initiatives on Linked Data

Figure 1 (larger version at lod-cloud.net) gives an overview of the datasets that have been published online following the Linked Data format and principles. It shows the growing number of CH institutions, web services and museums that have adopted LD standards as a way to turn the rich descriptions of their digital collections into a web of related data. Examples of such institutions include The Getty Research Institute[10] and the Yale Center for British Art[11]. Examples of web services are the Swedish Open

[4] http://www.europeana.eu/schemas/edm/
[5] http://europeana.eu
[6] http://www.cidoc-crm.org
[7] The LD principles formulated by Tim Berners-Lee are: 1) use URIs as names for things, 2) use HTTP URIs, 3) when someone looks up a URI, provide useful information using the standards (RDF, SPARQL), 4) include links to other URIs, so that they can discover more things.
[8] http://www.w3.org/DesignIssues/LinkedData.html
[9] http://linkeddata.org/
[10] http://www.getty.edu/research/tools/vocabularies/lod/
[11] http://britishart.yale.edu/collections/using-collections/technology/linked-open-data

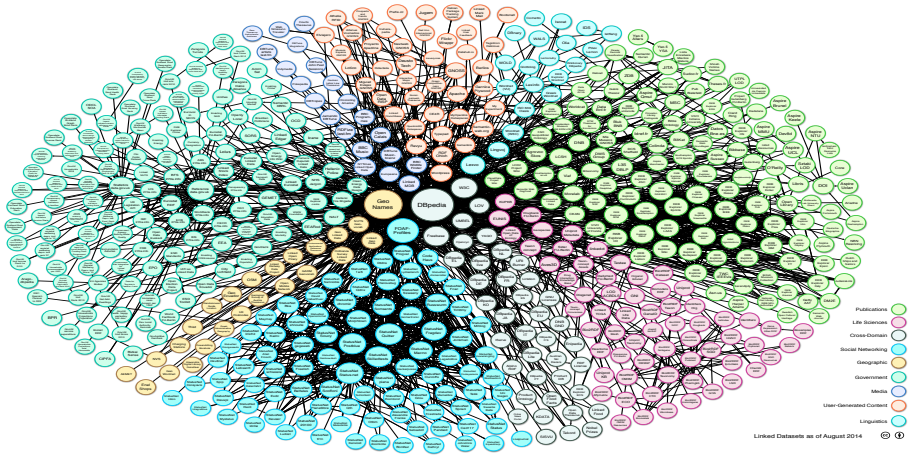


Fig. 1. The Linked Open Data cloud (August 2014) consists of datasets (shown as bubbles) in different domains (colour identified) and mappings between similar sources in the dataset (shown as arcs between bubbles). *Source: Linking Open Data cloud diagram 2014, Schmachtenberg, Bizer, Jentzsch and Cyganiak.* `http://lod-cloud.net/`.

Cultural Heritage (SOCH)[12], Europeana[13], and Museos de España[14]. Specific museums include various Italian Museums[15], The Smithsonian American Art Museum[16], The British Museum[17], The Victoria and Albert Museum[18], and The Rijksmuseum[19].

The integration of semantics-based interaction paradigms showcases the benefits of semantically marked up information, delivered to the end user through a *semantic browsing facility*, and highlights an important feature of the SW for the CH domain; the ability to explore and navigate relationships. The SCULPTEUR project focused initially on the semantic interoperability of LD and further developed the mSpace interaction framework[20], resulting in an interface that provides the users with the ability to organize their information space. The interactive interface aimed to allow users unfamiliar with

[12] A web service used to search and retrieve data from any organization holding information or media relating to Swedish cultural heritage, `http://www.ksamsok.se/in-english/`

[13] Aims to support research of the digitised content of Europe's galleries, museums, libraries and archives by addressing issues such as licencing, interoperability and access, `http://labs.europeana.eu/api/linked-open-data/introduction/`

[14] a directory that allows facet-based searches on a collection with more than 1,500 dynamic pages of public and private Spanish museums, `http://museos.gnoss.com/comunidad/mismuseos/`

[15] `http://www.linkedopendata.it/datasets/musei`

[16] `http://americanart.si.edu/collections/search/lod/about/`

[17] `http://collection.britishmuseum.org/`

[18] `http://www.vam.ac.uk/api/`

[19] `https://www.rijksmuseum.nl/en/api`

[20] `http://www.w3.org/2001/sw/wiki/MSpace`

museum collections to navigate, visualize and explore rich sources of cultural heritage information [19]. In particular, the application of InfoVis tools in mSpace helps people to develop knowledge by exploring relationships in data, and to customise access to the content to suit their individual interests by "slicing, sorting, swaping, information views and multimedia preview cues" [18].

MUSEUMFINLAND (Finnish Museums on the Semantic Web)[21] is a semantic portal for publishing heterogeneous museum collections on the Semantic Web in order to provide the museum visitors with intelligent content-based search and browsing services [9]. The view-based multi-facet search engine exploits the semantic link recommendation system to reveal the underlying semantic context of the collection items and their mutual relation [9]. CultureSampo[22] is a continuation of this work. The CultureSampo interface has been enhanced with InfoVis tools that allow the user to explore cultural heritage through nine semantic perspectives/thematic views: Maps and historical places, relational search, faceted domain-centric browsing, collections, Finnish history, cultural processes and skills, biographies, semantic Kalevala, and Carelia[23]. The main concept behind the interface is "to let users create virtual exhibitions that mimic the way real museums are organized, containing themed exhibition rooms of items and displays that together, through the objects, tell the story of a particular subject" [14].

Through the CHIP (Cultural Heritage Information Personalization) project[24], the Rijksmuseum focused on delivering novel personalization functions for the visitors on the museum's website. The CHIP demonstrator included three innovative components; the Art Recommender, the Tour Wizard and the Mobile Tour Guide, where the SW was deployed to support "the presentation of recommendations by combining different views like a historical timeline, a museum map and a faceted browser"[20]. The evaluations of the CHIP demonstrator provided critical insight in how to further adapt the user interaction facilities and interface to suit user's needs and preferences.

The ResearchSpace project[25] is a contextual search system that allows searching against objects, people, places, events, periods and concepts, providing context by making use of semantically enriched cultural data. The new design of the ResearchSpace search interface is a radical departure from traditional keyword and advanced searching, and can be customized to suit different online audiences.

Finally, the Russian Linked Culture Cloud[26] project is a collaboration between The Russian Museum and ITMO University, and is based on open data related to Russian Culture heritage. SW technologies, enabling enrichment of initial data with other facts of Russian culture, provide an advanced user experience with the help of visualization and navigation through enriched text, interactive timelines and an interactive influence graph [15].

[21] http://www.museosuomi.fi/
[22] http://www.kulttuurisampo.fi/
[23] http://www.kulttuurisampo.fi/about.shtml?lang=en
[24] http://www.chip-project.org
[25] http://www.researchspace.org/home/project-information/design
[26] http://culturecloud.ru/

3 Cognition-based InfoVis Tools for Linked Datasets

As industry's and society's interest in SW technologies grows and the number of widely adopted SW applications increases, there is an opportunity to focus on the properties and characteristics of semantically encoded data that can most readily be used to enhance InfoVis tools. In the case of online museum collections, SW technologies can have a beneficial impact on (a) exploring and navigating *relationships*, as richer semantics highlight the conceptual relationships between artifacts, and (b) *presentation-interaction*, as they offer richer presentation possibilities in terms of browsing and navigation.

Linked data interoperability in the semantic web has recently received much research attention. However, the emphasis has largely been in automating the mapping process to standards, even though the creation of mappings often involves the user. The main Linked Data users are technology experienced, and one reason for this is the lack of appropriate user interfaces and visualizations for non-expert users. Visual approaches are needed to assist various kinds of users, who pursue diverse goals and have individual requirements. InfoVis tools developed using a human-cognition-based model will cetainly improve users' engagement with and understanding of semantically encoded cultural data.

Case studies of museums that have implemented personalization facilities to their Web sites show that understanding is stimulated when the systems use concepts familiar to the user [2]. Visuals help understanding by acting as a frame of reference or as a temporary storage area for human cognitive processes. By providing a larger working set for thinking and analysis they become external cognition aids [12]. Card, Mackinlay and Shneiderman [3] list some key ways in which visuals can amplify cognition:

(i) Increasing memory and processing resources available
(ii) Reducing search for information
(iii) Enhancing the recognition of patterns
(iv) Enabling perceptual inference operations
(v) Using perceptual attention mechanisms for monitoring
(vi) Encoding information in a manipulable medium

One of the main aims of InfoVis is to amplify and augment the cognition of users. But a key challenge of the field is to measure its effectiveness in this respect. The absence of a cognitive-based framework for the evaluation of InfoVis systems makes the significance of achievements in this area difficult to describe, validate and defend. According to [13] even though InfoVis research has matured technically in recent years, an important problem for the field remains the lack of an underlying theory or even a systematic framework for guiding design and investigation. InfoVis classifications have focused on the process of synthesising and displaying data, and how to standardise this from a computational point of view. Considerations of how users interact with the resulting interfaces have only recently been integrated into this area of research. A greater emphasis on human cognition has the potential to uncover new ways of presenting, searching, exploring and visualizing the available semantic data in order to enhance human understanding. For example, the very recent work of Patterson et al [16] links

InfoVis with high-level cognitive processes such as reasoning and thinking, and Liu et al [13] argue that the use of "distributed cognition ... has the potential to serve as a theoretical framework for InfoVis". Other approaches that have revealed potential research opportunities in this area include Greene and Petre's Cognitive Dimensions [5], Johnson-Laird's work on model theory and reasoning using visual notations [11] and Peirce's systems of diagrammatic logic[17].

In the particular context of cultural heritage, a cognitive-based framework for the design of InfoVis interactive systems can improve visitors understanding of collections and their ability to explore the cultural information according to their needs. This will help visitors to discover the interconnectedness of digital cultural collections, enable information to be presented attuned to their interests and background, and therefore increase users interest and engagement with both digital and physical collections.

4 Summary and Conclusions

This paper presented some preliminary findings from an ongoing investigation into the use of InfoVis for semantically structured cultural heritage data. We gave a summary of the current state-of-the-art in this area, followed by an argument for more user-focused research and development, which will draw on models and theories of human cognition. Since the ultimate purpose of InfoVis in this context is to enhance users' ability to explore and understand (cultural) data, its effectiveness must be measured in these terms, and attention must therefore be paid to ordinary users' cognitive processes. Our future research plan is to use data from observational studies together with ideas from cognitive psychology in order to develop a theoretical framework for guiding the design and evaluation of cultural heritage InfoVis tools. Such tools will allow users to explore and interact with cultural data with only minimal additional cognitive load. An additional longer term aim is to apply this framework to produce an actual exemplar for a new generation of online InfoVis tools for exploration of museum data collections.

References

1. Alesso, H.P., Smith, C.F.: Thinking on the Web: Berners-Lee, Goedel, and Turing. Wiley, Hoboken, N.J. (2008)
2. Bowen, J., Fantoni Filippini, S.: Personalization and the Web from a Museum Perspective. In: Museums and the Web 2004: Proceedings. Archives & Museum Informatics, Washington DC / Arlington VA, USA (2004)
3. Card Stuart K. : Readings in information visualization : using vision to think / written and edited by Stuart K. Card, Jock D. Mackinlay, Ben Shneiderman. Morgan Kaufmann series in interactive technologies Y, Morgan Kaufmann Publishers (1999)
4. Eide, O., Felicetti, A., Ore, C.E., DAndrea, A., Holmen, J.: Encoding cultural heritage information for the semantic web. procedures for data integration through cidoc-crm mapping. In: Open Digital Cultural Heritage Systems Conference. p. 47 (2008)
5. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a cognitive dimensions framework. Journal of Visual Languages & Computing 7(2), 131–174 (1996)
6. Guerrini, M., Possemato, T.: Linked data: a new alphabet for the semantic web. JLIS. it 4(1), 67 (2013)

7. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, vol. 1 (2011)
8. Hooland, S.v.: Linked data for libraries, archives and museums: how to clean, link and publish your metadata. Facet Publishing, London (2014)
9. Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: MuseumFinlandFinnish museums on the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3), 224–241 (2005)
10. Isaac, A., Baker, T.: Linked data practice at different levels of semantic precision: The perspective of libraries, archives and museums. Bulletin of the American Society for Information Science and Technology 41(4), 34–39 (2015)
11. Johnson-Laird, P.: 8 the history of mental models. Psychology of reasoning: Theoretical and historical perspectives p. 179 (2004)
12. Kerren, A.: Information visualization human-centered Issues and perspectives. Springer (2008)
13. Liu, Z., Stasko, J.T.: Mental models, visual reasoning and interaction in information visualization: A top-down perspective 16(6), 999–1008 (2010)
14. Makela, E., Hyvonen, E., Ruotsalo, T.: How to deal with massively heterogeneous cultural heritage data lessons learned in culturesampo. Semantic Web Interoperability, Usability, Applicability 3(1) (2012)
15. Mouromtsev, D., Haase, P., Cherny, E., Pavlov, D., Andreev, A., Spiridonova, A.: Towards the Russian Linked Culture Cloud: Data Enrichment and Publishing. In: The Semantic Web. Latest Advances and New Domains, pp. 637–651. Springer (2015)
16. Patterson, R.E., Blaha, L.M., Grinstein, G.G., Liggett, K.K., Kaveney, D.E., Sheldon, K.C., Havig, P.R., Moore, J.A.: A human cognition framework for information visualization. Computers & Graphics 42, 42–58 (2014)
17. Pietarinen, A.V.: Peirce and the logic of image. Semiotica 2012(192) (2012)
18. Pitzalis, D., Lahanier, C., Pillay, R., Aitken, G., Russell, A., Smith, D.A., Sinclair, P.A.S., Addis, M.J., Lowe, R., Hafeez, S., others: Semantically exposing existing knowledge repositories: a case study in cultural heritage (2006)
19. Sinclair, P.A.S., Goodall, S., Lewis, P.H., Martinez, K., Addis, M.J.: Concept browsing for multimedia retrieval in the SCULPTEUR project (2005)
20. Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.: Recommendations based on semantically enriched museum collections. Web Semantics: Science, Services and Agents on the World Wide Web 6(4), 283–290 (2008)

Visualizing the Evolution of Ontologies: A Dynamic Graph Perspective

Michael Burch[1] and Steffen Lohmann[2]

[1] VISUS, University of Stuttgart, Germany
[2] VIS, University of Stuttgart, Germany

Abstract. Ontologies can be represented as graphs, since they essentially comprise a set of interconnected concepts describing a certain field of knowledge. Consequently, ontologies are often visualized as graphs, using different visual notations and common graph drawing techniques. The visualization of static graphs has been researched a lot, but when it comes to time-varying graphs, researchers face much more challenges in order to design useful, readable, and intuitive visualizations. If we have to deal with dynamic, i.e., evolving and time-dependent ontologies, we have to adapt existing visualization techniques to this challenging problem or develop new ones. In this position paper, we take a look at the visual representation of time-varying ontologies, and provide a discussion from a dynamic graph visualization perspective.

Keywords: time-varying ontologies, graph-based ontology visualization

1 Introduction

Visualization can be a powerful tool in the exploration and analysis of ontologies. Advanced visual designs are demanded to effectively and efficiently manage, browse, and navigate ontologies, and to finally gain insights and conclusions. The growing number and size of ontologies, on the other hand, require sophisticated visualization techniques that are capable to handle algorithmic, perceptual, and visual scalability problems. Consequently, developers of ontology visualizations need to enhance their visual designs by developing faster and better layouts, by providing more visual features, and by improving existing approaches based on user studies and other evaluation methods.

Although various techniques for ontology visualization have flourished over the past years [10,26,28,31], we are still facing the challenging problem of visually handling time-varying ontologies, i.e., ontologies that change over time. Ontologies are often not static but have an inherent dynamic behavior, making them an evolving data structure that is worth researching.

Since time-varying ontologies can be regarded as some kind of dynamic graph, we discuss the applicability of existing visualization techniques for dynamic graphs surveyed by Beck et al. [6], while we basically distinguish between time-to-time mappings (animations) and time-to-space mappings (static displays).

2 Related Work

Numerous approaches for ontology visualization have been presented in the last couple of years [10,26,28,31]. Most of them represent ontologies as graphs, while the graphs are typically rendered as node-link diagrams in a force-directed, hierarchical, or radial layout [31].

Examples for force-directed graph visualizations of ontologies are provided by TGViz [1], NavigOWL [25], and VOWL [31]. Hierarchical graph layouts depicting the inheritance tree of ontologies are used in OWLViz [23] and OntoTrack [29], among others. There are also approaches that represent the inheritance tree with treemaps [35], nested circles [40], or other visualization techniques for hierarchical tree structures, such as hyperbolic trees [11].

Fu et al. [14] conducted a user study where they compared graph visualizations of ontologies with indented tree representations. They found that the graph visualizations are perceived as "more controllable and intuitive without visual redundancy, especially for ontologies with multiple inheritance". They are considered "more suitable for overviews" and "held [the] attention" of the study participants better than trees [14].

Some approaches combine different techniques to visualize ontologies, such as node-link diagrams and adjacency matrices [4], or provide various graph layouts that the users can choose from depending on their task [13,22]. Others apply techniques such as hierarchical edge bundling [21] to increase the readability of the graph visualization, while yet others propose 3D graph visualizations for ontologies [7,15].

Furthermore, there are diagrammatic approaches that use UML [5] or similar graph-based notations [12,38] to visualize ontologies. For instance, COE [18] adopts the popular idea of Concept Maps [32] and applies it to the visualization of OWL ontologies, while a similar attempt has been made with Concept Diagrams [24] that particularly consider the logic of OWL.

However, the visualization of time-varying ontologies has not received any attention in all these approaches. Although the evolution of ontologies has been subject to research [20,33,36], we are not aware of any approach that visualizes evolving ontologies over time. There are methods to compute and analyze differences between two or more versions of an ontology [16,17,34], and tools that display such differences using rudimentary visual properties [17,27,35]. However, we do not know of any sophisticated visualization that supports the detailed analysis of ontologies at different points in time and assists users in the detection of dynamic patterns and trends in time-varying ontologies.

3 Visualization of Time-Varying Ontologies

Visualizing time-varying data is challenging due to the fact that users need to obtain an overview of a longer subsequence of individual time steps. This aspect must also be considered for dynamic, i.e., evolving ontologies. Usually, the

users have to solve comparison tasks in order to reliably derive trends and countertrends or to find outliers and anomalies. In general, two paradigms for time-oriented visualizations are distinguished: 1) time-to-time mappings (animated diagrams), and 2) time-to-space mappings (static displays, often enhanced by interaction techniques). In the following, we discuss these two alternatives and address some of their benefits and drawbacks.

3.1 Representation of Vertices and Edges

If ontologies are visualized as graphs in the form of node-link diagrams, the vertices and edges can have various visual appearances. The vertices usually carry different semantic information as well as additional attributes. If the attributes are of a rather categorical nature, color coding and shape can be used as visual features to support the viewer to efficiently distinguish vertices of different types. Quantities may be visually indicated by varying the sizes of the graphical primitives (circles, triangles, rectangles and the like, see Figure 1a).

Using too many visual features at once, however, can make it troublesome to visually analyze the ontology for certain aspects. This can be seen as a conjunction search that does not allow for preattentive processing [19], i.e., elements have to be explicitly searched, which is more time-consuming. Consequently, our suggestion is to use as little visual features as possible—less is more in this case.

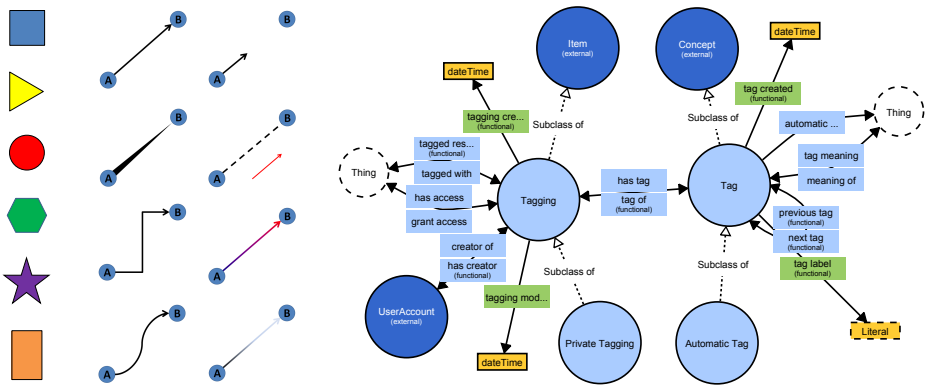


Fig. 1: Different representations for a) vertices and b) directed edges; c) node-link visualization of a small ontology.

Edges can be directed, weighted, attributed, and they can occur multiple times between the same pair of vertices, turning the ontology representation into a multigraph. Directions are typically indicated by arrowheads, color gradients, or with tapered representation styles. Weights can be expressed by color coding

or by using differently thick link representations, which produce additional visual clutter on the negative side. Figure 1b illustrates several link representations for directed edges (namely, arrows, partial, tapered, animated, orthogonal, colored, curved, and dark-to-light links).

Figure 1c shows a node-link representation of the small MUTO ontology [30], using the VOWL notation [31]. Classes are represented by circles and property labels by rectangles. Datatypes are also depicted as rectangles but with a border and in a different color. Dashed and dotted lines indicate special types of classes and properties. The visual graph elements are not weighted in this example.

3.2 Topology, Structure, and Hierarchy

The topology of the graph representation is important, since it conveys useful information on the structure of subgraphs, clusters, or cliques. This aspect is also crucial in ontology visualization, as it can provide useful insights into the ontology structure.

In graph visualization (which is mainly applying node-link visual metaphors), layout algorithms are typically following aesthetic graph drawing criteria that produce diagrams with reduced visual clutter, which is "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task" [37]. In the example of Figure 1c, a force-directed algorithm has been used to generate the initial layout, which was then manually optimized.

In many graph datasets, a hierarchical organization among the vertices is of special interest. Either it is inherently present in the data or it can be generated, for example, by a hierarchical clustering algorithm. Ontologies often contain concept hierarchies that have to be visualized in order to visually explore the ontology on different levels of hierarchical granularity or to use the hierarchy as a means to interact, filter, aggregate, and navigate in the ontology.

In Figure 1c, there are only two relatively flat hierarchies, each consisting of three classes, which can be easily spotted due to the small number of well arranged vertices. For ontologies with large inheritance trees, other graph layouts, or even tree visualizations, might be more appropriate to clearly depict the different hierarchy levels. However, since ontologies allow for multiple inheritance, simple tree visualizations can be confusing [14]. Also, most approaches have their limitations when it comes to the visualization of very large ontologies, at least when a node-link representation of the graph is used.

3.3 Visual Encoding of Time

The representation of time-varying ontologies demands for sophisticated visualization techniques that consider all the aforementioned features in order to sufficiently support the visual exploration of ontologies for dynamic patterns. Consequently, vertices, edges, the topology and structure, as well as any existing hierarchical organization among the vertices are of special interest.

One major type of tasks that users typically want to answer when inspecting time-varying data are comparison tasks. Several time steps are visually compared

to derive insights by detecting changes or stabilities over time. To answer such tasks, the human visual system has to rely on its short term memory allowing to briefly remember visual patterns which are then compared at different spatial positions. Only by this internal cognitive process we are able to come up with the detection of trends, countertrends, or anomaly patterns over time [8].

If animated diagrams were used for the exploration of dynamic ontologies, we soon reach a point where the cognitive load becomes high. The human viewer may have difficulties to visually analyze the time-varying ontology for dynamic patterns. Advanced and time-complex layout algorithms have to be used to guarantee a high degree of dynamic stability [9], with the goal to preserve a viewer's mental map while inspecting the graph [3]. However, in the end, the detection of trends can be challenging anyway, even if the animation is replayed several times. Moreover, interaction techniques cannot be integrated in a traditional way, since the graphical elements are moving around in the worst case, which demands to stop the animation to meaningfully interact with the dynamic ontology.

Another option for displaying dynamic ontologies is by means of static diagrams [2] that map the time dimension to display space, which are known as time-to-space mappings. Such diagrams can, for example, use a vertex-aligned representation that allows to attach a hierarchical organization in a static way. Other than in ontology animation, the users can decide where to look at in the display in order to search for static or dynamic visual patterns. They can perform comparison tasks visually, not mentally as in animated diagrams that demand for higher cognitive efforts and are usually performing worse for time-oriented tasks.

To illustrate this second approach, we created different versions of the MUTO ontology [30] that was already shown in Figure 1c. We visualized these ontology versions with VOWL [31] in small display regions next to each other, which is known as a *small multiples* visualization [39]. If the vertices are roughly aligned and the visual features are not encoded differently over time (apart from changing variables), this approach leads to a good means to derive time-dependent patterns.

In Figure 2, we see the VOWL representations of six of the MUTO versions, starting with version 0.1 in the upper left and ending with version 1.0 in the lower right. We can observe how the ontology has changed over time. At first, the key concepts and links are defined, which are gradually extended by further concepts and links. At some point, alignments to existing ontologies are added, followed by the definition of datatype properties describing attributes for the key concepts. Finally, subclasses are introduced providing specializations of the key concepts.

3.4 Interaction Techniques

It must be noted that small multiples visualizations usually serve as an overview representation for the time dimension. If users detect a dynamic pattern of interest, they can apply interaction techniques to zoom and filter, and finally get details on demand also for individual ontology versions. The individual ontology

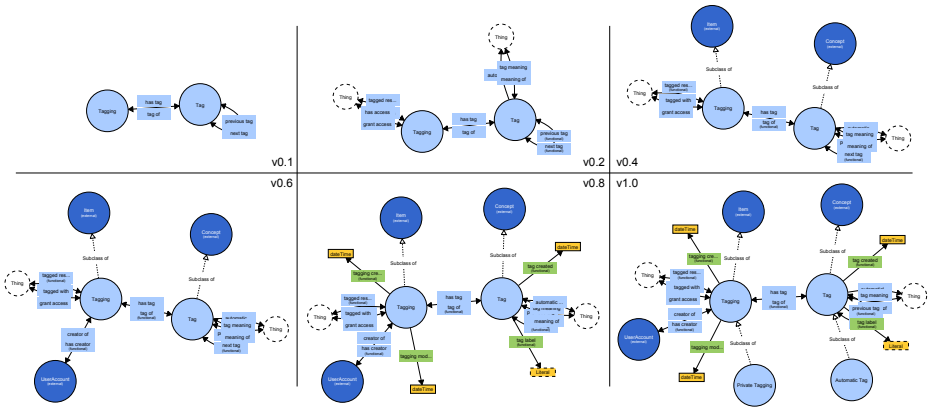


Fig. 2: Different versions of an ontology visualized as small multiples.

versions might be represented with existing ontology visualization techniques, as described in Section 2 and applied in Figure 2.

However, interacting with visual representations of time-varying data can be challenging, in particular, when the visualization is animated, i.e., when the single snapshots are changing over time. Users have to stop the animation when they like to interact with the visualization, otherwise a detected pattern might have already been disappeared until the users realize, for example, that they would like to select it for a more detailed exploration. Consequently, time-to-space mappings usually provide better means to visualize time-varying ontologies from the perspective of applying interaction techniques, since the ontology sequence is shown in a static fashion. Different ontology versions can be individually explored and visually connected by linking and brushing, even if this form of graphical representation usually does not scale to very long time sequences.

4 Conclusion and Future Work

In this position paper, we discussed the challenge of visualizing time-varying ontologies. We described benefits and drawbacks of common visualization techniques for dynamic graphs, since ontologies can be considered as a special type of graphs with additional attributes attached to the vertices and edges.

The visualization of ontologies has mainly been researched from the perspective of static graphs so far, which is – at least in our opinion – only half of the truth. Ontologies – as many other data structures – are typically not staying static but evolve over time. Consequently, the visualization of ontology evolution is a topic of its own and worth discussing.

For future work, we plan to apply dynamic graph visualization techniques to time-varying ontologies. Typically, we first need an overview representation which shows dynamic patterns in a static view, allowing the user to effectively and efficiently dig deeper and analyze the time-varying ontology.

References

1. H. Alani. TGVizTab: An ontology visualisation extension for Protégé. In *2nd Workshop on Visualizing Information in Knowledge Engineering*, 2003.
2. D. Archambault and H. C. Purchase. The mental map and memorability in dynamic graphs. In *IEEE Pacific Visualization Symposium*, pages 89–96, 2012.
3. D. Archambault and H. C. Purchase. The "map" in the mental map: Experimental results in dynamic graph drawing. *International Journal on Human-Computer Studies*, 71(11):1044–1055, 2013.
4. B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev. OntoTrix: A hybrid visualization for populated ontologies. In *20th International Conference on World Wide Web, Companion Volume*, pages 177–180. ACM, 2011.
5. J. Bārzdiņš, G. Bārzdiņš, K. Čerāns, R. Liepiņš, and A. Sproģis. OWLGrEd: A UML style graphical notation and editor for OWL 2. In *International Workshop on OWL: Experiences and Directions*, CEUR-WS, vol. 614, 2010.
6. F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. In *EuroVis – STARs*, pages 83–103. EA, 2014.
7. A. Bosca, D. Bonino, and P. Pellegrino. OntoSphere: More than a 3D ontology visualization tool. In *2nd Italian Semantic Web Workshop*, CEUR-WS, vol. 166.
8. M. Burch and D. Weiskopf. Visualizing dynamic quantitative data in hierarchies – TimeEdgeTrees: Attaching dynamic weights to tree edges. In *International Conference on Information Visualization Theory and Applications*, pages 177–186, 2011.
9. S. Diehl and C. Görg. Graphs, they are changing. In *International Symposium on Graph Drawing*, pages 23–30, 2002.
10. M. Dudáš, O. Zamazal, and V. Svátek. Roadmapping and navigating in the ontology visualization landscape. In *19th International Conference on Knowledge Engineering and Knowledge Management*, pages 137–152. Springer, 2014.
11. P. Eklund, N. Roberts, and S. Green. OntoRama: Browsing RDF ontologies using a hyperbolic-style browser. In *1st International Symposium on Cyber Worlds*, pages 405–411. IEEE, 2002.
12. R. Falco, A. Gangemi, S. Peroni, D. Shotton, and F. Vitali. Modelling OWL ontologies with graffoo. In *ESWC 2014 Satellite Events*, pages 320–325. Springer, 2014.
13. S. Falconer. OntoGraf. http://protegewiki.stanford.edu/wiki/OntoGraf, 2010.
14. B. Fu, N. F. Noy, and M.-A. Storey. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In *12th International Semantic Web Conference*, pages 117–134. Springer, 2013.
15. S. S. Guo and C. W. Chan. A tool for ontology visualization in 3D graphics: Onto3DViz. In *23rd Canadian Conference on Electrical and Computer Engineering*, pages 1–4. IEEE, 2010.
16. M. Hartung, A. Groí, and E. Rahm. COnto-Diff: Generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics*, 46(1):15–32, 2013.
17. M. Hartung, T. Kirsten, A. Groß, and E. Rahm. OnEX: Exploring changes in life science ontologies. *BMC Bioinformatics*, 10(250), 2009.
18. P. Hayes, T. C. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff. Collaborative knowledge capture in ontologies. In *International Conference on Knowledge Capture*, pages 99–106. ACM, 2005.
19. C. Healey and J. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2012.

20. J. Heflin and J. A. Hendler. Dynamic ontologies on the web. In *17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, pages 443–449. AAAI, 2000.
21. D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.
22. W. Hop, S. de Ridder, F. Frasincar, and F. Hogenboom. Using hierarchical edge bundles to visualize complex ontologies in GLOW. In *27th Annual ACM Symposium on Applied Computing*, pages 304–311. ACM, 2012.
23. M. Horridge. OWLViz. http://protegewiki.stanford.edu/wiki/OWLViz, 2010.
24. J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: A case study. In *10th International Semantic Web Conference*, pages 257–272. Springer, 2011.
25. A. Hussain, K. Latif, A. Rextin, A. Hayat, and M. Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355–367, 2014.
26. A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods – a survey. *ACM Computer Surveys*, 39(4), 2007.
27. M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In *13th International Conference on Knowledge Engineering and Knowledge Management*, pages 197–212. Springer, 2002.
28. M. Lanzenberger, J. Sampson, and M. Rester. Visualization in ontology tools. In *International Conference on Complex, Intelligent and Software Intensive Systems*, pages 705–711. IEEE, 2009.
29. T. Liebig and O. Noppens. OntoTrack: A semantic approach for ontology authoring. *Web Semantics*, 3(2-3):116–131, 2005.
30. S. Lohmann, P. Díaz, and I. Aedo. MUTO: The modular unified tagging ontology. In *7th International Conference on Semantic Systems (I-SEMANTICS '11)*, pages 95–104. ACM, 2011.
31. S. Lohmann, S. Negru, F. Haag, and T. Ertl. Visualizing ontologies with VOWL. *Semantic Web Journal*, to appear.
32. J. D. Novak and A. J. Cañas. The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition, 2006.
33. N. F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004.
34. N. F. Noy and M. A. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *18th National Conference on Artificial Intelligence*, pages 744–750. AAAI, 2002.
35. D. Perrin. PROMPT-Viz: ontology version comparison visualizations with treemaps. Master thesis, University of Victoria, 2004.
36. P. Plessers, O. De Troyer, and S. Casteleyn. Understanding ontology evolution: A change detection approach. *Web Semantics*, 5(1):39–49, 2007.
37. R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: A measure of display clutter. In *Conference on Human Factors in Computing Systems*, pages 761–770, 2005.
38. M. Sintek. OntoViz. http://protegewiki.stanford.edu/wiki/OntoViz, 2007.
39. E. R. Tufte. *The visual display of quantitative information.* Graphics Press, 1992.
40. T. D. Wang and B. Parsia. CropCircles: Topology sensitive visualization of OWL class hierarchies. In *5th International Semantic Web Conference*, pages 695–708. Springer, 2006.

Discovering Issues in Datasets Using LODSight Visual Summaries

Marek Dudáš and Vojtěch Svátek

Department of Information and Knowledge Engineering,
University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic,
{marek.dudas|svatek}@vse.cz

Abstract. Quality-checking of linked data is a hot topic nowadays. As complement to fully automated quality analysis we propose issue discovery via manual exploration of dataset summary graphs. Our LODSight summary visualizer has been extended with new features, ontology/predicate filtering, instance picking and multi-dataset summarization, so as to better support this task. Three scenarios of dataset issue discovery have been investigated with the help of the extended tool.

1 Introduction

Quality checking of RDF datasets is a widely researched topic with diverse approaches being applied [7]. Automated error detection might be test-driven, where a set of tests implemented, e.g., as SPARQL queries might be run to search for incorrect predicate usage or typing. However, the tests have to be prepared in advance. This may work well for checking the usage of entities from a single ontology; datasets however often refer to several ontologies, and preparing and maintaining a set of tests for every possible combination of ontologies that might be used in the dataset does not seem feasible. Using reasoning and checking for inconsistencies is an obvious option, but requires the ontologies to be systematically equiped with axioms, including, e.g., the class disjointness ones, which is not always the case (e.g., in the DBpedia ontology). Manual, user-driven evaluation of facts in the dataset has also been proposed, but its scalability is obviously limited.

As a novel approach we propose to first *summarize* the dataset graph/s and then to apply specifically tailored *visualization* over the summary, allowing for manual discovery of issues. Depending on context, the visual exploration of summaries may either precede the automated quality analysis (indicating, e.g., on which predicates the tests are to be run), or, conversely, focus on parts of the dataset already indicated as problematic by automated analysis; for smaller datasets the analysis in visualizer might even be sufficient. In the paper we present several possibilities how a previously developed dataset summary visualization tool, *LODSight* [2], enriched with several new features, can be used as a complement to existing error detection systems.

Related research The visualization in LODSight is similar to *maps of ontology usage* [4] and Explod [3]. Both tools could also be used for error detection in a similar way as presented with LODSight. We are unaware of any research focused on exploiting visualization for dataset error detection. However, many non-visual approaches to error detection exist. Atencia et al. [1] proposes finding pseudo-keys in the dataset and using them to detect errors such as a person with the same death and birth date. Detection of such errors cannot be performed nor supported with LODSight type of visualization – it is too general to allow comparison of values linked to specific instance. Outlier detection implemented by Paulheim [6] could be supported by LODSight: combinations of classes and properties detected as outliers could be e.g. highlighted in the visualization. A simpler form of outlier detection can be even performed in LODSight by looking at type-property combinations with lower frequency in the dataset represented by link thickness. Error detection done by Péron et al. [8] uses domain/range axioms from ontologies. Kontokostas et al. [5] implemented versatile error detection based on SPARQL queries automatically created from patterns. Complex approaches like the last two mentioned obviously can reveal errors that cannot be seen in the simplified visualization. However, the general overview of the dataset contents provided by the visualization might still help to determine which approaches to error detection should be used for the specific dataset. Datasets can be also checked for errors manually, as shown by Zaveri et al. [9].

2 LODSight

LODSight[1] is a dataset summary visualization tool. It uses SPARQL to find all type-property and datatype-property paths in the dataset. Type-property path is a sequence `type1 - property - type2`. `Type1` and `type2` are the types of instances from the dataset that are connected by the property. We use the term *path frequency* to denote the number of triples `?s ?property ?o` in the dataset where `?s` is an instance of `type1` and `?o` of `type2`. Datatype-property paths are analogous sequences of `type - datatype property - datatype`. All paths are merged into one graph and visualized in one view allowing the user to see generalized structure of the dataset and usage of ontologies in it. The visualization is interactive and the user can also filter the displayed paths to show only those with lower or higher frequency. The summarization is run offline as it might be prohibitively time-consuming in case of larger datasets. The results of the summarization are stored in a database. A list of previously summarized datasets is offered to view in the LODSight web application. To support error detection, we implemented several new features.

Ontology Filter Whenever dataset visualization is loaded, a list of ontology IRIs used in the dataset is shown. Users can select any subset of the IRIs to limit the visualization to entities from the selected ontologies and entities linked directly to them. This way users can analyze usage of selected ontology in the context of the dataset.

[1] Available at `http://lod2-dev.vse.cz/lodsight-v2`

Predicate Filter Similarly to the ontology list, a list of all properties used in the dataset is displayed. When a subset of the properties is selected, only the entities linked with them are shown. Users can thus analyze their usage without other links cluttering the view.

Analyzing Example Instances Users can select a subset of class nodes in the graph and retrieve their example instances that are linked with the properties shown in the generalized graph. The labels or URIs of the instances are displayed above the class nodes. The user can click on any of them to open a new browser tab where the resource description is retrieved. This makes manual checking of the facts related to the instances easier.

Merging Summarizations of Several Datasets Any number of the available dataset summarizations can be selected in the list and then visualized in one view. The paths from all the selected summarizations are simply merged into one graph and displayed. This feature can be useful in conjunction with ontology filter – see Section 3.3 for more details.

3 Preliminary Tests in Example Usage Scenarios

3.1 Analyzing Large Dataset with Predicate Filter

As an example of a large dataset, we used Greek DBpedia. Visualizing the whole summarization is simply impossible in this case as it contains thousands of paths and the resulting visualization is too cluttered and thus unreadable. A way to get an overview of possibly erroneous parts of the structure would be to limit the maximum path frequency to a very low number. In this case, that still leads to too many results and unreadable visualization. So does filtering the visualization by ontology. A feasible option is to filter by predicate. We can go through the predicates one by one, or select those suggested by some other error detection method or by an expert. Consider the latter case, where, e.g., the property *dbo:child* from DBpedia ontology was identified as possibly incorrectly used and thus selected in the predicate filter. In the resulting visualization (Fig. 1) we can immediately see classes like *dbo:WrittenWork*, whose instances clearly should not be linked with *dbo:child* property. We manually adjust the visualization to focus on one of them and see that *dbo:WrittenWork* is linked to *dbo:Person* with *dbo:child*. We select the two class nodes and retrieve their example instances. Their labels are shown above the class nodes (Fig. 2). They are in Greek, so perhaps not yet helpful by themselves, but we can click on them and their description is opened in a new browser window. There we can see that both instances are actually persons, but one of them was incorrectly typed as book.

3.2 Showing the Whole Structure To See Missing Links

Smaller summarizations (approx. up to hundred paths) of less complex datasets can be visualized as whole in a single view. This may allow to see another type of error: missing links. Consider the visualization of the RISM Authorities dataset

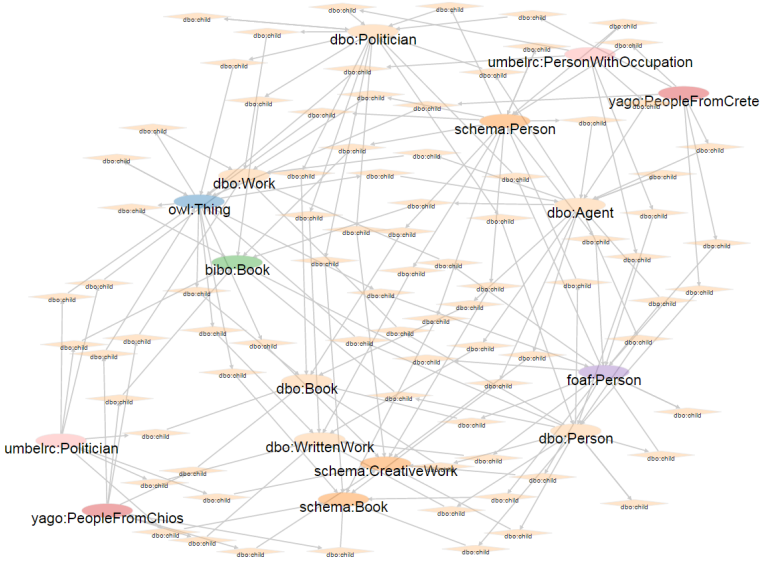


Fig. 1. Usage of dbo:child property in Greek DBpedia visualized in LODSight.

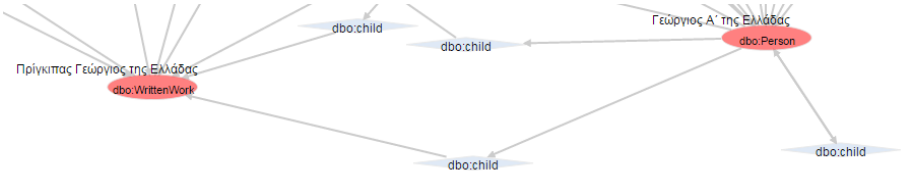


Fig. 2. Example instances of dbo:WrittenWork and dbo:Person linked with dbo:child.

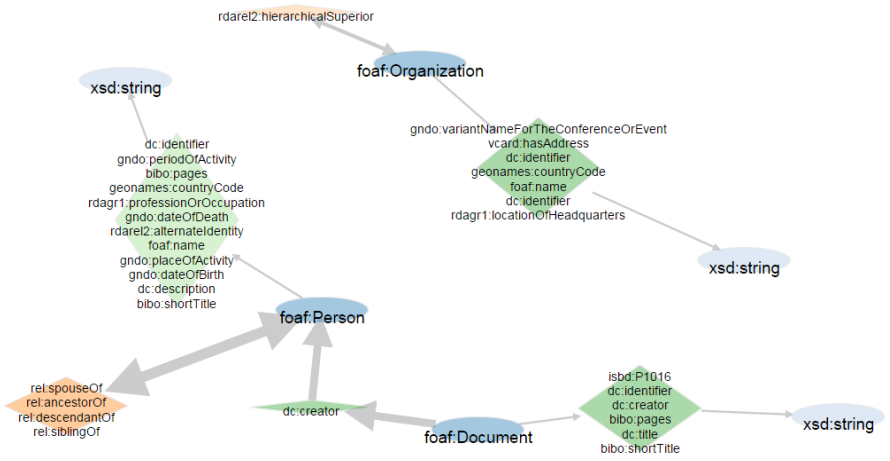


Fig. 3. RISM Authorities dataset summarization in LODSight.

summarization in Fig. 3. It shows that the dataset contains interlinked persons and documents. There are also organizations that are not linked to any person nor document. That indicates that some facts may be missing in the dataset: e.g., that persons are members of organizations or organizations are owners of documents.

3.3 Detecting Errors in Ontology Usage Across Several Datasets

Filtering the visualization to entities of a selected ontology might be used in conjunction with merging summarizations of several datasets into one graph. This way an expert on the given ontology might check its usage in several datasets at once, instead of looking at each dataset separately. Consider an expert on FOAF who wants to check if the ontology is used correctly in the RISM Authorities and ESWC2015 datasets.[2] The expert selects both datasets and sets the ontology filter to FOAF. The result is in Fig. 4. The expert might spot, e.g., the possibly

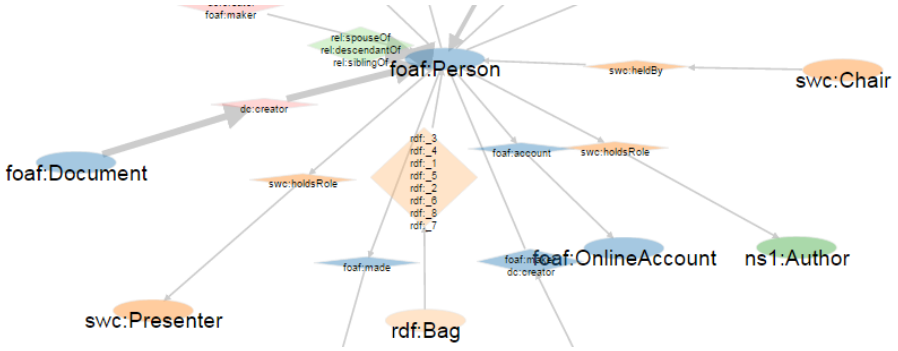


Fig. 4. A part of the visualization of FOAF usage merged from two datasets.

suboptimal usage of *dc:creator* to link *foaf:Document* to *foaf:Person* (*foaf:maker* is recommended by FOAF documentation to link document and person instances instead of *dc:creator*). The expert could find out which datasets contain such triples and inform their maintainers about the proper usage. The functionality for showing which part of the graph comes from which dataset in such merged visualization is however yet to be implemented.

4 Conclusions and Future Work

We proposed that dataset structure visualization might be helpful for detecting errors in a dataset. We enriched existing dataset summary visualization tool, LODSight, with several new features and showed their usage in several scenarios

[2] Randomly chosen out of the datasets we summarized with LODSight so far.

of error detection. Preliminary results suggest that in case of large datasets[3] the capabilities of the visualization are somewhat limited – the same results can be achieved using some existing automated error detection method more easily. For smaller datasets, whose whole summarizations can be viewed on one screen, we so far identified two possible use cases when the visualization might be useful: finding missing links and checking ontology usage across several datasets. Although the former might be done automatically without the visualization, the visualization may allow an expert user to more easily decide whether disconnected subgraphs in the summarization are a result of an error or just a coincidence. The latter cannot be easily replaced by automated tests, since it would be hard to prepare tests for every possible combination of properties and classes from different ontologies; in contrast, an expert can spot the incorrect usage immediately in the visualization. Future work will include investigating other error detection scenarios, thorough evaluation, and reliability enhancement of the tool.

The research is supported by UEP IGA F4/90/2015 and by long-term institutional support of research activities by Faculty of Informatics and Statistics, Univ. of Economics, Prague.

References

1. Atencia, M., David, J., Scharffe, F.: Keys and pseudo-keys detection for web datasets cleansing and interlinking. In: Knowledge Engineering and Knowledge Management, pp. 144–153. Springer (2012)
2. Dudáš, M., Svátek, V., Mynarz, J.: Dataset summary visualization with LODSight. In: The 12th Extented Semantic Web Conference (ESWC2015). `http://lod2-dev.vse.cz/lodsight/lodsight-eswc2015-demopaper.pdf`
3. Khatchadourian, S., Consens, M.: Explod: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. The Semantic Web: Research and Applications pp. 272–287 (2010)
4. Kinsella, S., Bojārs, U., Harth, A., Breslin, J.G., Decker, S.: An interactive map of semantic web ontology usage. In: Information Visualisation, 2008. IV'08. 12th International Conference. pp. 179–184. IEEE (2008)
5. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: Proceedings of the 23rd international conference on World Wide Web. pp. 747–758. ACM (2014)
6. Paulheim, H.: Identifying wrong links between datasets by multi-dimensional outlier detection. In: 3rd International Workshop on Debugging Ontologies and Ontology Mappings, WoDOOM (2014)
7. Paulheim, H.: Automatic knowledge graph refinement: A survey of approaches and evaluation methods (2015), `http://www.semantic-web-journal.net/system/files/swj1083.pdf`
8. Péron, Y., Raimbault, F., Ménier, G., Marteau, P.F.: On the detection of inconsistencies in RDF data sets and their correction at ontological level (2011)
9. Zaveri, A., Kontokostas, D., Sherif, M.A., Bühmann, L., Morsey, M., Auer, S., Lehmann, J.: User-driven quality evaluation of DBpedia. In: Proceedings of the 9th International Conference on Semantic Systems. pp. 97–104. ACM (2013)

[3] In terms of the number of combinations of classes and properties used in the dataset.

Representing and Visualizing Text as Ontologies: A Case from the Patent Domain

Stamatia Dasiopoulou[1], Steffen Lohmann[2], Joan Codina[1], and Leo Wanner[1,3]

[1] Department of Information and Communication Technologies, Pompeu Fabra University, Barcelona, Spain
[2] Institute for Visualization and Interactive Systems, University of Stuttgart, Stuttgart, Germany
[3] Catalan Institute for Research and Advanced Studies, Barcelona, Spain

Abstract. This paper presents preliminary results on a framework for the representation and visualization of text as OWL ontologies under an open-domain paradigm, where no a priori schema for the facts to be extracted is available. The extracted ontology is visually represented as a specifically tailored node-link diagram. The applicability of the approach is demonstrated on a use case from the patent domain.

1 Introduction

Extracting ontologies from text can significantly facilitate knowledge integration and querying, through semantic alignment and mediation [6]. Only recently though, under the Linking Open Data (LOD) paradigm of publishing and linking structured information on the Web, has research shifted towards open-domain approaches, where no a priori schema for the facts to be extracted is available and the textual input is considered in its entirety [1,20].

Within such context, two main challenges emerge: 1) to ensure the translation of textual input into well-formed ontologies that facilitate knowledge integration and querying in a schema-agnostic fashion; and 2) to provide the means for comprehensive visualizations that foster the understanding of the extracted knowledge, particularly at the factual level, in an intuitive manner that appeals adequately to users of diverse backgrounds and with varying levels of expertise.

These challenges are sharply manifested in the patent domain. The highly specialized and cross-domain terminology used in patent documents makes it very difficult, if not impractical, to rely on the availability of predefined schemata for the extraction of knowledge relevant to the task at hand. Moreover, the inherent complexity of patent documents render effective visualizations key tools for assisting experts in quickly grasping the main elements and their interactions.

In this paper, we present preliminary results on a framework for the representation and visualization of text as an OWL ontology under an open-domain paradigm, and illustrate its application with a use case from the patent domain. Abstracting from the specifics of the various semantic parsing methodologies, we describe an entity-relation-centric model for OWL-based text representation together with a graphical notation for its visualization as node-link diagram.

2 Related Work

In accordance with the twofold goal of the proposed framework, related approaches to ontology extraction and visualization are discussed in the following.

2.1 Extracting Ontologies from Text

Although ontology learning and population from text have been the subject of arduous research [4,21], investigations into the conceptualization of text in its entirety have commenced only recently with LODifier [1] and FRED [20]. Both use Boxer [7] to extract Discourse Representation Structures (DRSs), namely *discourse referents* (entities) and *conditions* (unary and binary relations), and respective rules to translate them into ontological representations. LODifier keeps modeling commitments minimal, by introducing a blank node for each discourse referent and by using reification to capture embedded DRSs. FRED [20] implements a more earnest mapping of DRSs to OWL constructs, utilizing frame semantics [2], links to the DOLCE+DnS foundantional ontology and heuristic rules that aim to maximize conformance to Semantic Web best practices.

Both result in representations that explicitly cater for *n-ary* relations, which represent a critical share of relations for effectively capturing the richness of textual contents. However, LODifier compromises ontology design with choices such as blank nodes, whereas FRED ensures high compliance with best practices, but the presented translations and heuristic rules are specifically tailored to DRSs. Instead, our goal is to provide a model for the generation of OWL representations from text that avoids commitments to specifics of the predicate-argument structures.

2.2 Visualizing Fact-based Ontologies

Many approaches to graphically represent ontologies have been proposed in the last couple of years [8,14]. However, they are not tailored to the visualization of ontologies that are extracted from text, and have limitations in this regard. While some approaches (e.g., OWLViz [13] and KC-Viz [18]) merely visualize the class hierarchy of ontologies, others (e.g., OntoGraf [10] and FlexViz [11]) are able to represent different types of properties. All these attempts are related to the visualizations generated by FRED in that they focus on terminological knowledge (aka TBox) and not on assertional knowledge (aka ABox), which we aim to visualize in our work. The same holds for ontology visualizations that provide more elaborated notations (e.g., Graffoo [9] and VOWL [15]), i.e., they also mainly address the ontology schema and are therefore less appropriate for the representation of fact-based ontologies extracted from text.

This is different in visualizations of RDF and Linked Data that are typically more oriented towards the ABox. Examples include RDF Gravity [12], Welkin [17], and LodLive [5]. Such visualizations depict the triple structure of RDF but they are usually not capable to represent n-ary relations. In addition, they use plain node-link diagrams with only little variation in the visual elements.

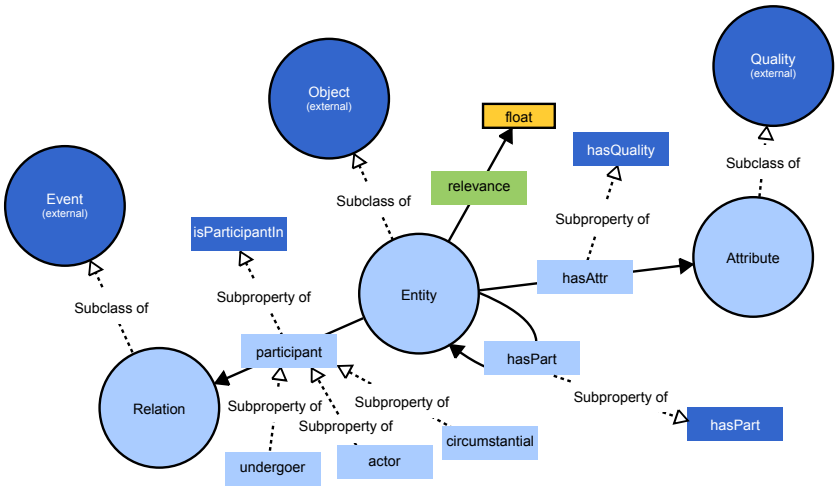


Fig. 1. Classes and properties of the core vocabulary (visualized with VOWL).

3 Ontological Text Representation

Aiming to abstract from predicate-argument specifics while assuring maximal interoperability within the Semantic Web and LOD context, we developed a minimal reference model for generating ontological text representations at a factual level.[4] Hence, our goal is to provide core classes and properties for capturing the ways in which the extracted entities are interrelated, and that can be applied across domains, serving as anchors for attaching application-tailored class and property hierarchies.

A key design decision has been to model the extracted relations as classes rather than properties. This is motivated by the saliency of n-ary relations in textual resources, and the incurred loss of semantics when, instead of preserving the n-ary dependencies, they are broken down into binary relations [19]. Furthermore, direct mappings to well-established foundational ontologies, such as DOLCE+DnS Ultralite[5] and SUMO[6], are promoted to enhance the interoperability and compliance with ontology design practices.

In accordance with the aforementioned principles, the model comprises the following core classes: `Entity` subsumes the set of physical objects, processes, and substances; `Relation` captures n-ary interrelations between entities; `Attribute` encompasses characteristic aspects of an entity that cannot exist without it. Alongside, a minimal set of upper-level object properties connect individuals of the three classes: `participant` allows to link entities to the relations in which

[4] Modalities, such as belief, causality, and entailment, are not considered as they can be covered through specialized ontologies and knowledge patterns.
[5] `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl`
[6] `http://www.adampease.org/OP/SUMO.owl`

they participate; `actor` and `undergoer` specialize `participant` in order to discriminate between direct participants ("who?", "what?", etc.) and complementary ones (e.g., the $[\text{pump}]_{actor}$ pumps $[\text{water}]_{undergoer}$), while `circumstantial` is a specialization used as a catch-all property for other types of participation; `hasAttr` is used to associate entities to their attributes; `hasPart` is used to capture mereological relations between entities; lastly, the datatype property `relevance` allows to capture the relevance of the extracted entities to the matter being considered. Figure 1 visualizes the core vocabulary using VOWL [15]. The vocabulary is aligned with classes and properties from the DOLCE+DnS Ultralite ontology, which have a white font on a dark background in Figure 1.

The extracted predicate-argument structures can then be translated into OWL representations, according to the following rules:

- For each extracted entity, attribute, or relation, a named individual is generated; for co-referential entities, i.e., entities referring to the same real-world object, a single individual is introduced.
- For each added named individual, respective `rdf:type` statements are added based on the extracted vocabulary of entities, attributes, and relations.
- Respective `rdfs:subClassOf` axioms are added for each introduced entity, relation, and attribute class.
- Instigative and passive participation links between entities and relations are translated into respective `actor` and `undergoer` property assertions; likewise for circumstantial participation, where additionally the prepositions lexicalizing the participation are defined as subproperties. For example, given the excerpt "...connected along...", `along` is added as a subproperty of `circumstantial`.
- Links between entities and attributes as well as entities and their parts are captured as `hasAttr` and `hasPart` property assertions, respectively.

The result is an OWL ontology consisting primarily of assertional knowledge, i.e., class and object property assertions, and to a lesser extent of terminological knowledge, as it could be derived from links to LOD resources, such as DBpedia and WordNet. Further specializations and schema enrichments, according to the given application needs, can be acquired through ontology learning.

4 Visualization of the Extracted Ontology

Our visual notation for the graphical representation of the extracted ontology is inspired by VOWL [15], which provides user-oriented visualizations for OWL ontologies. VOWL has, for instance, been used to create the visualization of Figure 1. However, whereas VOWL focuses on the visualization of the ontology schema, we are interested in the visualization of facts extracted from text. Therefore, we could not simply reuse VOWL but developed a related ABox visualization that combines the strengths of VOWL with the peculiarities of visualizing fact-based ontologies extracted from text.

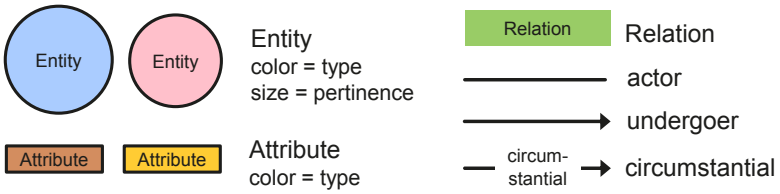


Fig. 2. Notation for the graphical representation of the extracted ontology.

Figure 2 summarizes the current visual notation. We adopted the basic visual elements of VOWL, consisting of circles which represent the extracted entities and rectangles representing the relations. The colors of the circles and attributes can be varied depending on their type. In contrast to VOWL, relations can be n-ary, which requires that they are rendered as nodes. This is in line with our design decision to model relations as classes rather than properties in the extracted OWL ontologies. Furthermore, we introduced a labeled link element to depict prepositions that qualify circumstantial participations.

We also adopted the idea of scaling the size of the circles, which, in VOWL, reflects the number of individuals that are members of a class. In our case, the circle size indicates the relevance values computed for the terms: Entities with a higher relevance value are shown in a larger size in the visualization. This helps to easily spot those entities that are most relevant to the matter being considered.

Finally, we decided to attach the attributes directly to the entity nodes instead of adding another link, as for the datatype properties in VOWL, in order to emphasize their strong connection and visually indicate that attributes cannot exist without the corresponding entities.

5 Use Case from the Patent Domain

Patent documents are highly idiosyncratic, verbose texts that describe elaborate inventions and make heavy use of specialized terminology. These characteristics, in combination with the continuously growing rate at which patents are filed worldwide, incur extensive labor and time costs for carrying out typical patent portfolio analysis tasks. In this context, structured representations that can assist experts in identifying and contrasting patents relevant to the task in question, by rendering semantics explicit, and visualizations that effectively summarize the key elements of an invention and foster understanding, can entail immediate competitive advantages.

In the investigated use case, we address constructive patents, i.e., patents that describe the constituent parts of machine inventions and the ways in which they interact. In this context, it is important to specialize the described entities into components (e.g., *coil*, *battery*), substances, processes, and other entities (e.g., *temperature*); likewise, for spatial and quantity attributes, such as *inner* charger

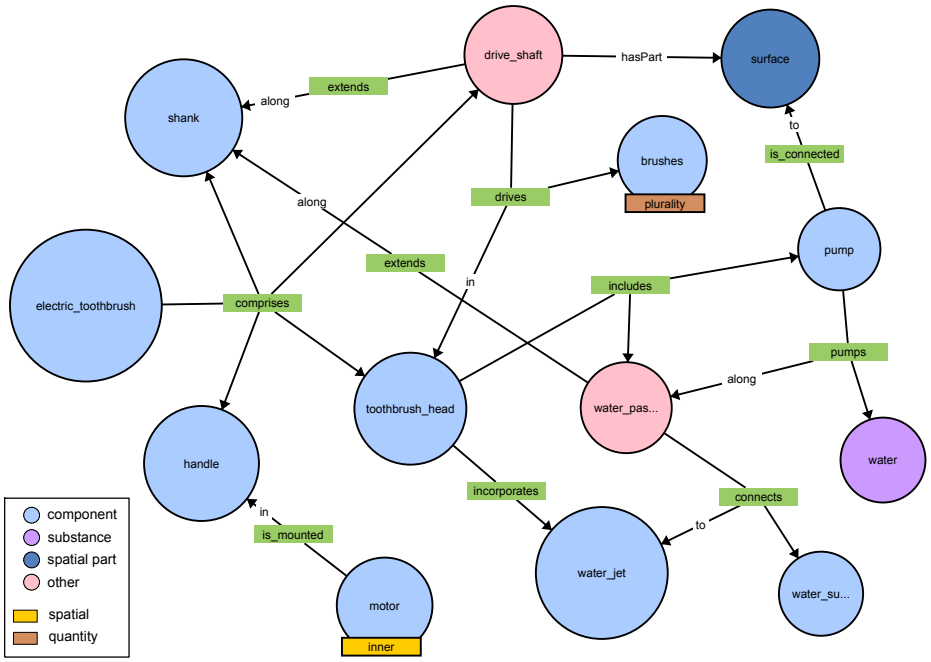


Fig. 3. Visualization of an ontology extracted from the claim text of a patent.

and *plurality* of brushes, as well as spatial parts (e.g., *surface*, *bottom*). To this end, the upper-level model definitions have been extended accordingly through the introduction of respective subclasses to the classes `Entity` and `Attribute`.

Using the *mate tools* [3], predicate-argument structures are extracted and subsequently their relevance is computed following a methodology similar to one used for identifying relevant sentences in extractive summarization tasks [16]. Then, OWL representations in compliance with the extended core model are generated, based on the transformation rules described in Section 3.

Figure 3 shows the visualization resulting from the below patent claim, where the extracted entity, relation, and attribute individuals are outlined in respective fonts. The initial layout of the diagram has been generated with a force-directed algorithm and has then been manually adapted to increase its readability.

An ***electric toothbrush*** *with a* ***water jet****, the* ***toothbrush*** COMPRISING *a* ***handle****, a* ***shank****, and a* ***toothbrush head*** *that* INCORPORATES *the* ***water jet****, in which an inner* ***motor*** IS MOUNTED *in the* ***handle*** *and the* ***toothbrush*** INCLUDES *a* ***reciprocating drive shaft*** EXTENDING *along the* ***shank*** *to* DRIVE *a plurality of* ***brushes*** *in the* ***brush head****,* INCLUDING *a* ***water passage*** EXTENDING *along the* ***shank*** *to* CONNECT *a* ***water supply*** *to the* ***water jet****, and a* ***pump*** *in the* ***shank*** *for* PUMPING ***water*** *along the* ***passage*** *that* IS *mechanically* CONNECTED *directly to the surface of the* ***drive shaft****.*

In the given example, there are four types of extracted entities (components, substances, spatial parts, and other) and two types of attributes (spatial and quantity), as indicated by the different colors assigned to the entity and attribute

nodes. As mentioned before, coreferential entities are captured by a single individual, upon which the respective participation links are projected. For example, the mentions of passage in "...a `water passage` EXTENDING along the `shank`..." and "...PUMPING `water` along the `passage`..." refer to the same passage entity; accordingly, there is a single "water passage" node to which the participation links in the EXTENDING and PUMPING relations have been projected.

All in all, the visualization provides an adequate representation of the patent claim that could be used to support analysts in understanding the elements and interrelations of the described invention.

6 Conclusions and Future Work

In this paper, we have presented an upper-level model for extracting ontological text representations under an open-domain paradigm that allows abstracting from the specifics of predicate-argument structures, and a visual notation for its graphical representation that focuses on the visualization of facts rather than the ontological schema. The applicability of the proposed representation and visualization framework has been demonstrated through a use case from the patent domain.

Future work includes further validation and fine-tuning of the representation model, through extensive evaluation in cooperation with experts from the patent domain, as well as in an application-wise manner, where it will be used as the basis for assessing semantic similarity between patents. Furthermore, future research will have to address enhanced visualization paradigms that are more tailored to the patent domain.

General challenges with regard to the notation are improved scalability and readability of the visualization. A scalable visualization must be capable to represent larger ontologies extracted from several paragraphs of a text. In the patent use case, the individual claims could, for instance, form different subgraphs that are connected with each other according to specified dependencies.

Generalizing from the patent domain, the presented representation and visualization framework may serve as a valuable starting point for related cases of ontology extraction and visualization. The open-domain character of the ontology extraction and representation approach enables its wide application, along with the visual notation that combines the clarity of VOWL with an ABox-oriented view and capabilities to explicitly represent n-ary relations.

Acknowledgments

This work has been supported by the EU FP7-SME-606163 project iPatDoc.

References

1. Augenstein, I., Padó, S., Rudolph, S.: LODifier: Generating linked data from unstructured text. In: 9th Extended Semantic Web Conference (ESWC '12). pp. 210–224. Springer (2012)

2. Baker, C., Fillmore, C., Lowe, J.: The Berkeley FrameNet project. In: 36th Annual Meeting of the Association for Computational Linguistics and 17th Int. Conference on Computational Linguistics (COLING-ACL '98). pp. 86–90. ACL (1998)
3. Bohnet, B., Nivre, J., Boguslavsky, I., Farkas, R., Ginter, F., Hajic, J.: Joint morphological and syntactic analysis for richly inflected languages. Transactions of the Association for Computational Linguistics 1, 415–428 (2013)
4. Buitelaar, P., Cimiano, P. (eds.): Ontology Learning and Population: Bridging the Gap Between Text and Knowledge. IOS Press (2008)
5. Camarda, D.V., Mazzini, S., Antonuccio, A.: LodLive, exploring the web of data. In: 8th International Conference on Semantic Systems (I-SEMANTICS '12). pp. 197–200. ACM (2012)
6. Cimiano, P.: Ontology learning and population from text - algorithms, evaluation and applications. Springer (2006)
7. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale NLP with c&c and boxer. In: 45th Annual Meeting of the Association for Computational Linguistics (ACL '07). ACL (2007)
8. Dudáš, M., Zamazal, O., Svátek, V.: Roadmapping and navigating in the ontology visualization landscape. In: 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW '14). pp. 137–152. Springer (2014)
9. Falco, R., Gangemi, A., Peroni, S., Shotton, D., Vitali, F.: Modelling OWL ontologies with graffoo. In: ESWC 2014 Satellite Events. pp. 320–325. Springer (2014)
10. Falconer, S.: OntoGraf. http://protegewiki.stanford.edu/wiki/OntoGraf (2010)
11. Falconer, S., Callendar, C., Storey, M.A.: A visualization service for the semantic web. In: 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW '10). pp. 554–564. Springer (2010)
12. Goyal, S., Westenthaler, R.: RDF Gravity. http://semweb.salzburgresearch.at/apps/rdf-gravity/ (2004)
13. Horridge, M.: OWLViz. http://protegewiki.stanford.edu/wiki/OWLViz (2010)
14. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods – a survey. ACM Computer Surveys 39(4) (2007)
15. Lohmann, S., Negru, S., Haag, F., Ertl, T.: VOWL 2: User-oriented visualization of ontologies. In: 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW '14). pp. 266–281. Springer (2014)
16. Mani, I.: Automatic summarization. John Benjamins Publishing (2001)
17. Mazzocchi, S., Ciccarese, P.: Welkin. http://simile.mit.edu/welkin/
18. Motta, E., Mulholland, P., Peroni, S., d'Aquin, M., Gomez-Perez, J.M., Mendez, V., Zablith, F.: A novel approach to visualizing and navigating ontologies. In: 10th International Semantic Web Conference (ISWC '11), Part I. pp. 470–486. Springer (2011)
19. Noy, N., Rector, A., Hayes, P., Welty, C.: Defining n-ary relations on the semantic web. http://www.w3.org/TR/swbp-n-aryRelations/ (2006)
20. Presutti, V., Draicchio, F., Gangemi, A.: Knowledge extraction based on discourse representation theory and linguistic frames. In: 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW '12). pp. 114–129. Springer (2012)
21. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future. ACM Computer Surveys 44(4) (2012)

OptiqueVQS: Ontology-based Visual Querying

Ahmet Soylu[1,2], Evgeny Kharlamov[3], Dmitriy Zheleznyakov[3],
Ernesto Jimenez-Ruiz[3], Martin Giese[1], and Ian Horrocks[3]

[1] Department of Informatics, University of Oslo, Norway
{ahmets, martingi}@ifi.uio.no
[2] Faculty of Informatics and Media Technology, Gjøvik University College, Norway
ahmet.soylu@hig.no
[3] Department of Computer Science, University of Oxford, United Kingdom
{name.surname}@cs.ox.ac.uk

Abstract. Visual methods for query formulation undertake the challenge of making querying independent of users' technical skills and the knowledge of the underlying textual query language and the structure of data. In this paper, we demonstrate an ontology-based visual query system, namely OptiqueVQS, which we have been developing for end users within a large industrial project.

Keywords: Visual Query Formulation, Ontology, Usability, SPARQL.

1 Introduction

Query interfaces play an essential role by enabling end users to express their ad hoc information needs. In this respect, visual query systems (VQSs) primarily undertake the challenge of making querying independent of users' technical skills and the knowledge of the underlying textual query language and the structure of data. To this end, we have been developing an ontology-based visual query system for end users, namely OptiqueVQS [1], within a large industrial project called Optique [2]. OptiqueVQS does not use a formal notation and syntax for query representation, but still conforms to the underlying formalism. It employs a formal approach projecting the underlying ontology into a graph for navigation, which constitutes the backbone of the query formulation process.

In this paper, we first demonstrate OptiqueVQS from an end-user perspective, and then present the ontology to graph projection approach.

2 OptiqueVQS

OptiqueVQS is meant for end users who have no or very limited technical skills and knowledge, such as on programming, databases, query languages, and have low/no tolerance, intention, nor time to use and learn formal textual query languages. It is not our concern to reflect the underlying formality (i.e., query language and ontology) per se; however, user behaviour is constrained so as to enforce the generation of valid queries. Secondly, we are not interested in providing full expressivity in order to reach a usability-expressivity balance.

2.1 User Interface

The OptiqueVQS interface is designed as a widget-based user-interface mashup (UI mashup). Apart from flexibility and extensibility, such a modular approach provides us with the ability to combine multiple representations, interaction, and query formulation paradigms, and distribute functionality appropriately.

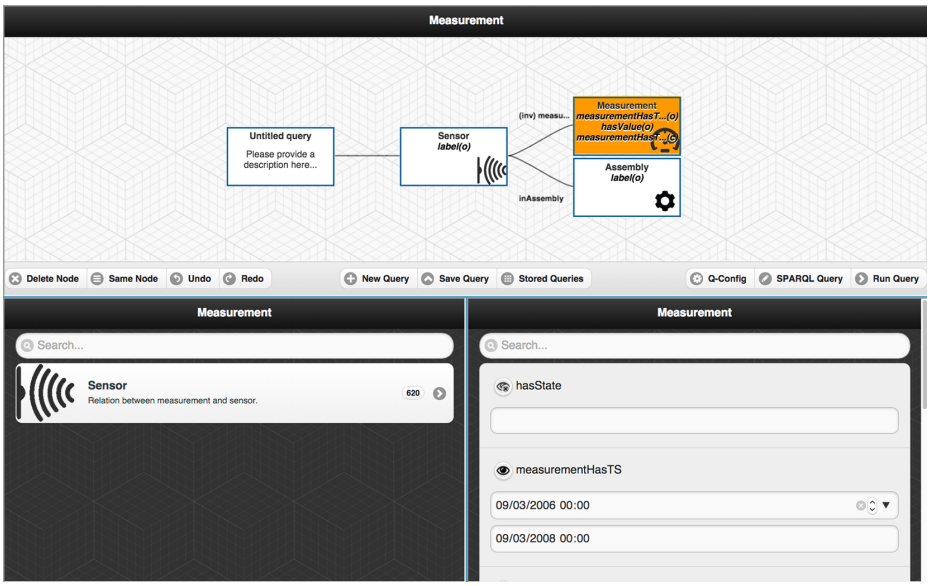


Fig. 1. An example query in visual mode is depicted.

Initially, three widgets appear in OptiqueVQS as depicted in Figure 1. The first widget (W1 – see the bottom-left part of Figure 1) is menu-based and allows users to navigate concepts by pursuing relationships between them. The second widget (W2 – see the bottom-right part of Figure 1) is form-based and presents the attributes of a selected concept for selection and projection operations. W1 and W2 provide view by focusing user to the active concept and provide means for gradual and on-demand exploration and construction. The third widget (W3 – see the top part of Figure 1) is diagram-based and provides an overview of the constructed query and functionality for manipulation.

Typically, a user first selects a kernel concept, i.e., the starting concept, from W1, which initially lists all domain concepts. The selected concept appears on the graph (i.e., W3) as a variable-node and becomes the pivot/active/focus node (i.e., the node coloured in orange or highlighted). W2 displays its attributes in the form of text fields, range sliders, etc. The user can select attributes to be included in the result list (i.e., using the "eye" button) and/or impose constraints on them through form elements in W2. Currently, the attributes selected for output appear on the corresponding variable-node with a letter "o", while constrained attributes appear with letter "c". The user can further refine the type of variable-node from

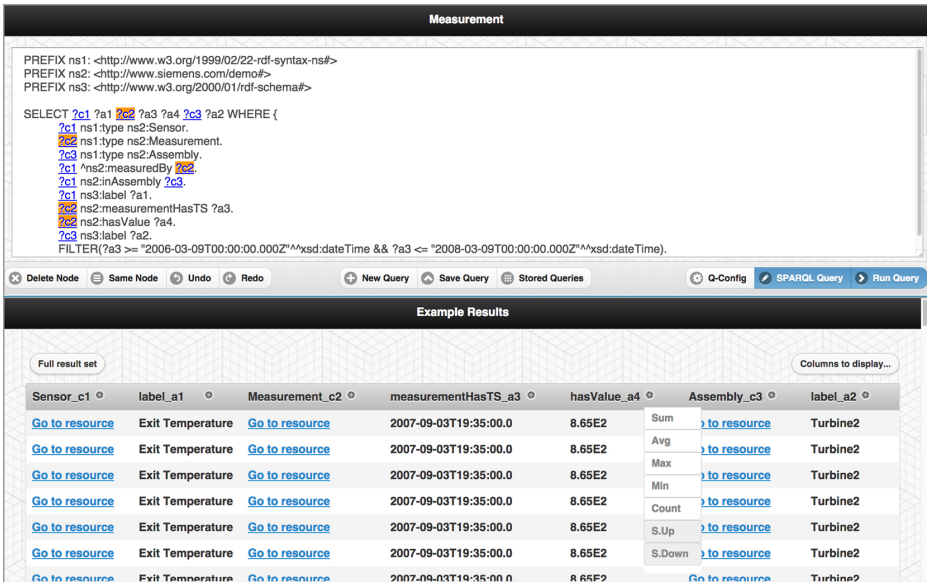


Fig. 2. An example query in textual mode and result view are depicted.

W2, by selecting appropriate subclasses, which are treated as a special attribute (named "Type") and presented as a multi-selection combo-box form element. Note that once there is a pivot node, W1 does not purely list concepts anymore but a set of (sub)paths. Each item/path in W1 represents a combination of a possible relationship with its range concept pertaining to the pivot (i.e., indeed a path of length one). The user can select any available item from the list; this results in a new path with a new variable-node of type specified by the selected item, a join between the pivot and the new variable-node over the specified relationship, and a move in focus to the new variable-node (i.e., pivoting). The user has to follow the same steps to involve new concepts in the query and can always jump to a specific part of the query by clicking on the corresponding variable-node in W3. The arcs that connect variable-nodes do not have any direction, but it is implicitly left to right. In W3, a tree-shaped query representation is employed to avoid a graph representation for simplicity.

The user can delete nodes, access the query catalogue, save/load queries, and undo/redo actions by using the buttons at the bottom part of W3. The user can also switch to editable textual SPARQL mode by clicking on "SPARQL Query" button at the bottom-right part of the W3 as depicted in Figure 2. The textual mode enables collaboration between end users and technology experienced users.

Finally, we recently extended OptiqueVQS with two new widgets, which provide an evidence on how a widget-based architecture allows us to hide complex functionality behind layers and combine different paradigms. The first widget is tabular result widget (W4 – see Figure 2). It provides an example result list from the current query and also means for aggregation and sequencing operations. The second widget is a map widget (W5 – see Figure 3). It allows end users to

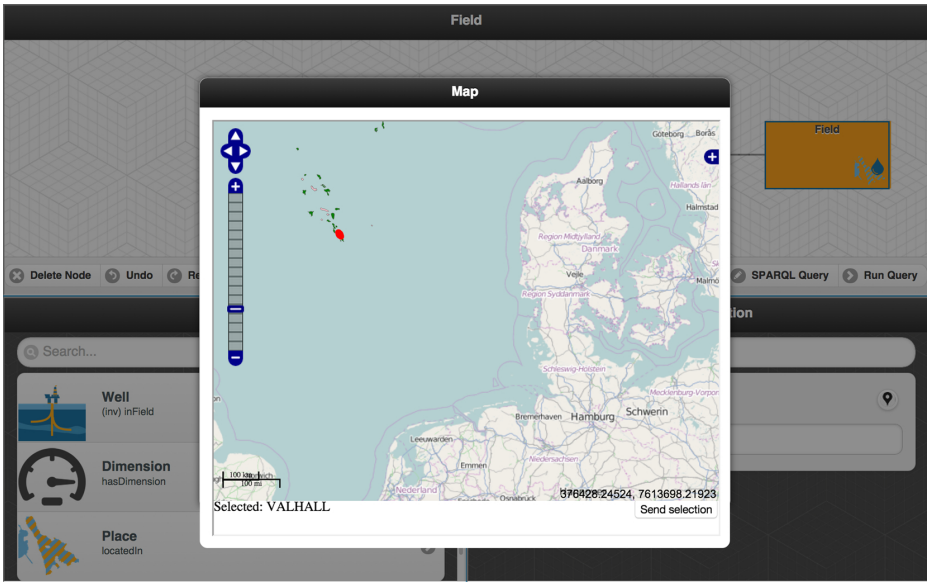


Fig. 3. An example query with the map widget is depicted.

constrain geospatial attributes by selecting an input value from the map. For this purpose, a button with a pin icon is placed next to every appropriate attribute.

2.2 Navigation Graph

Intuitively, OptiqueVQS allows users to construct tree-shaped conjunctive queries where each path is of the form: $Person(x), livesIn(x, y), City(y), ...$ Each such path essentially 'connects' classes like *Person* and *City* via properties like *livesIn*. At each query construction step OptiqueVQS suggests the user classes and properties that are semantically relevant to the already constructed partial query. We determine this relevance by exploiting the input OWL 2 ontology: we project the input ontology onto a graph structure that is called *navigation graph* [3] and use this graph at query construction time. More precisely, for each class in the partial query OptiqueVQS suggests only those properties and classes which are reachable in the navigation graph in one step. Note that OWL 2 ontologies are essentially sets of first-order logic axioms and thus there is no immediate relationship between them and a graph. This makes projection of OWL 2 ontologies onto a navigation graph a non-trivial task.

In the remaining part of this section we will formally introduce navigation graph, define when a query is meaningful with respect to it, and finally we define the grammar of queries that users can construct with the help of OptiqueVQS.

The nodes of a navigation graph are unary predicates and constants, and edges are labelled with possible relations between such elements, that is, binary predicates or a special symbol type. The key property of a navigation graph is that every X-labelled edge (v, w) is justified by a rule or fact entailed by $\mathcal{O} \cup D$

which "semantically relates" v to w via X. We distinguish three kinds of semantic relations: *(i) existential*, where X is a binary predicate and (each element of) v must be X-related to (an element of) w in the models of $\mathcal{O} \cup D$; *(ii) universal*, where (each instance of) v is X-related only to (instances of) w in the models of $\mathcal{O} \cup D$; and *(iii) typing*, where $X = \mathsf{type}$, and (the constant) v is entailed to be an instance of (the unary predicate) w. Formally:

Definition 1. *Let $\mathcal{O}$ be an OWL 2 ontology and D a knowledge graph. A* navigation graph *for $\mathcal{O}$ and D is a directed labelled multigraph G having as nodes unary predicates or constants from $\mathcal{O}$ and D and s.t. each edge is labelled with a binary predicate from $\mathcal{O}$ or* type*. Each edge e is justified by a fact or rule α_e s.t. $\mathcal{O} \cup \mathcal{C} \models \alpha_e$ and α_e is of the form given next, where c, d are constants, A, B unary predicates, and R a binary predicate:*

(i) if e is $c \xrightarrow{R} d$, then α_e is of the form $R(c, d)$ or $\forall y.[R(c, y) \rightarrow y \approx d]$;

(ii) if e is $c \xrightarrow{R} A$, then α_e is a rule of the form $\top(c) \rightarrow \exists y.[R(c, y) \wedge A(y)]$ or $\forall y.[R(c, y) \rightarrow A(y)]$;

(iii) if e is $A \xrightarrow{R} B$, then α_e is a rule of the form $\forall x.[A(x) \rightarrow \exists y.[R(x, y) \wedge B(y)]]$ or $\forall x, y.[A(x) \wedge R(x, y) \rightarrow B(y)]$;

(iv) if e is $A \xrightarrow{R} c$, then α_e is a rule of the form $\forall x.[A(x) \rightarrow R(x, c)]$ or $\top(c) \rightarrow \exists y.[R(y, c) \wedge A(y)]$ or $\forall x, y.[A(x) \wedge R(x, y) \rightarrow y \approx c]$;

(v) if e is $c \xrightarrow{\mathsf{type}} A$, then $\alpha_e = A(c)$.

The first (resp., second) option for each α_e in *(i)-(iii)* encodes the existential (resp., universal) R-relation between nodes in e; the first and second (resp., third) options for each α_e in *(iv)* encode the existential (resp., universal) R-relation between nodes in e; and *(v)* encodes typing. A graph may not contain all justifiable edges, but rather those that are deemed relevant to the given application.

To realise the idea of ontology and data guided navigation, we require that interfaces *conform to* the navigation graph. We assume that all the following definitions are parametrised with a fixed ontology $\mathcal{O}$ and a knowledge graph D.

Definition 2. *Let Q be a conjunctive query. The* graph of *Q is the smallest multi-labelled directed graph G_Q with a node for each term in Q and a directed edge (x, y) for each atom $R(x, y)$ occurring in Q, where R is different from $\approx$. We say that Q is* tree-shaped *if G_Q is a tree. Moreover, a variable node x is labelled with a unary predicate A if the atom $A(x)$ occurs in Q, and an edge (t_1, t_2) is labelled with a binary predicate R if the atom $R(t_1, t_2)$ occurs in O.*

Finally, we are ready to define the notion of conformation.

Definition 3. *Let Q be a conjunctive query and G a navigation graph. We say that Q* conforms to *G if for each edge (t_1, t_2) in the graph G_Q of Q the following holds:*

- *If t_1 and t_2 are variables, then for each label B of t_2 there is a label A of t_1 and a label R of (t_1, t_2) such that $A \xrightarrow{R} B$ is an edge in G.*

- *If t_1 is a variable and t_2 is a constant, then there is a label A of t_1 and a label R of (t_1, t_2) such that $A \xrightarrow{R} t_1$ is an edge in G.*
- *If t_1 is a constant and t_2 is a variable, then for each label B of t_2 there is a label R of (t_1, t_2) such that $t_1 \xrightarrow{R} t_2$ is an edge in G.*
- *If t_1 and t_2 are constants, then a label R of (t_1, t_2) such that $t_1 \xrightarrow{R} t_2$ is an edge in G.*

OptiqueVQS allows to construct conjunctive tree-shaped queries. The generation is done via reasoning over the navigation graph which contain edges of types (iii)-(v) (see Definition 1).

Now we describe the class of queries that can be generated using OptiqueVQS and show that they conform to the navigation graph underlying the system. First, observe that the OptiqueVQS queries follow the following grammar:

$$\begin{aligned} \texttt{query} &::= A(x)(\wedge\ \texttt{constr}(x))^*(\wedge\ \texttt{expr}(x))^* \\ \texttt{expr}(x) &::= \texttt{sug}(x,y)(\wedge\ \texttt{constr}(x))^*(\wedge\ \texttt{expr}(y))^* \\ \texttt{constr}(x) &::= \exists y R(x,y) \mid R(x,y) \mid R(x,c) \\ \texttt{sug}(x,y) &::= Q(x,y) \wedge A(y) \end{aligned}$$

where A is an atomic class, R is an atomic data property, Q is an object property, and c is a data value. The expression of the form $A(\wedge\ B)^*$ designates that B-expressions can appear in the formula 0, 1, and so on, times. An OptiqueVQS `query` is constructed using suggestions `sug` and constraints `constr`, that are combined in expressions `expr`. Such queries are conjunctive and tree-shaped. All the variables that occur in classes and object properties are output variables and some variables occurring in data properties can also be output variables.

3 Conclusion

OptiqueVQS enables non-experienced users to formulate comparatively complex queries at a conceptual level. The future work includes implementation of more features without compromising the usability, such as optionals.

Acknowledgements. This research is funded by "Optique" (EC FP7 318338), as well as the EPSRC projects Score!, DBOnto, and MaSI^3.

References

1. Soylu, A., et al.: Experiencing OptiqueVQS: a multi-paradigm and ontology-based visual query system for end users. Universal Access in the Information Society (in press)
2. Giese, M., et al.: Optique: Zooming in on Big Data. IEEE Computer Magazine **48**(3) (2015)
3. Arenas, M., et al.: Faceted Search over Ontology-Enhanced RDF Data. In: CIKM'14. (2014)

An Autocomplete Input Box for Semantic Annotation on the Web

Tuan-Dat Trinh, Peter Wetz, Ba-Lam Do,
Peb Ruswono Aryan, Elmar Kiesling, and A Min Tjoa

TU Wien, Vienna, Austria
{tuan.trinh,peter.wetz,ba.do,
peb.aryan,elmar.kiesling,a.tjoa}@tuwien.ac.at

Abstract. A large share of websites today allow users to contribute and manage user-generated content. This content is often in textual form and involves names, terms, and keywords that can be ambiguous and difficult to interpret for other users. Semantic annotation can be used to tackle such issues, but this technique has been adopted by only a few websites. This may be attributed to a lack of a standard web input component that allows users to simply and efficiently annotate text. In this paper, we introduce an autocomplete-enabled annotation box that supports users in associating their text with DBpedia resources as they type. This web component can replace existing input fields and does not require particular user skills. Furthermore, it can be used by semantic web developers as a user interface for advanced semantic search and data processing back-ends. Finally, we validate the approach with a preliminary user study.

1 Introduction

The interaction paradigm on the world wide web has changed dramatically in recent years. Users are no longer limited to receiving information passively; rather, they often create and manage their own textual content. A lot of such user-generated content is in raw text format, which is typically entered via input fields (e.g., via the html *textarea* element). In various contexts, users find it difficult to understand each others' terms and expressions due to language ambiguities and inconsistent terminology. Adding annotations such as names, attributes, comments, descriptions, etc. to a selected part of the text allows users to more explicitly express semantics and increase the precision of the statements made.

TripAdvisor[1], for instance, provides more than 200 million traveler-created reviews, opinions, and photos of tourist attractions and accommodations. Many travelers use TripAdvisor to obtain information on points of interest such as parks, museums, historic buildings, streets, etc. To make decisions about what places to visit and to plan a trip, travelers spend a lot of time in search for additional information on interesting places. This search for context information

[1] http://www.tripadvisor.com/ (accessed 20 July 2015)

could be much more efficient if concepts in travel-related text were associated with the respective DBpedia content. We will illustrate some travel-related use cases in Section 3.

Although annotations can clarify meaning, provide a context for textual descriptions, and hence facilitate search and automatic data processing, most websites currently do not use them. This can at least partly be explained by the lack of a suitable input component that can easily be integrated into existing web sites. In this paper, we introduce a simple to use autocomplete-enabled annotation box as a replacement for traditional HTML textarea. Using this box, users can quickly and easily enrich selected text with semantic annotations (i.e., DBpedia resources). The component is written in Java Script and can easily be integrated by developers into their websites.

The remainder of this paper is organized as follows. We introduce the annotation box and its implementation and workflow in Section 2. Section 3 analyzes how users and developers can benefit from the features provided by the box, and Section 4 presents our user study to validate the approach. Finally, we discuss related work and draw conclusions in section 5.

2 Autocomplete Text Annotation

Our annotation input box looks and behaves similar to a standard HTML text input. Behind the user interface, however, there is a small processor that allows users to quickly annotate text. To create an annotation, users invoke the processor by pressing *Ctrl + Space*. Next, the processor uses the selected words to generate a query that returns a list of relevant resources (cf. Fig. 1). Users can then select an item from the results to associate their word with the respective resource. To annotate a compound word, users select and highlight related words and press *Ctrl + Space*.

Fig. 1: Autocomplete-enabled annotation box

To provide suitable suggestions, we need a common dataset to look up resources from the given text. Because the expected user content and accompanying contexts are arbitrary, the dataset used should cover various domains. To query for related resources, we therefore use DBpedia [1], which can be considered the central hub of the LOD cloud.

From a Wikipedia page we can obtain the corresponding DBpedia resource and vice versa. For instance, the matching DBpedia resource for *https://en.wikipedia.org/wiki/Swimming_pool* is *http://dbpedia.org/resource/Swimming_pool*. As a consequence, we can use either DBpedia or Wikipedia APIs to look up and suggest resources from the given text to users. Available options are (i) DBpedia

Lookup[2], (ii) DBpedia Faceted Search[3], (iii) Wikidata API[4], (iv) and Wikipedia API[5]. We performed an informal evaluation in order to decide which of these APIs supports us in finding relevant LOD resources. As a result of our experiments, we decided to use the Wikipedia API. We found that compared to the others, this API responds faster and returns more relevant resources.

The annotation box is an HTML5 *div* area whose *editableContent* attribute is set to *true*. To enforce compatibility with various browsers, we had to deal with a number of technical challenges. For example, when we edit the text and press enter to go from one line to the next, Firefox will add a *br* element to the *div*, while Internet Explorer and Chrome use *p* and a sub *div*, respectively. We also needed to standardize the final content of the box so that it can be saved consistently into a data base. It consists of *div* elements where each represents a line in the box; each *div* element contains only *span* and *a* elements for unannotated and annotated text, respectively. This structure allows developers to easily extract DBpedia resources from the input content to perform further processing in the back-end. Other implementation challenges were: (i) The box processor needs to calculate the correct location of the text cursor in pixels to present the resource list at a reasonable position (i.e., right below the first character of the selected word). (ii) Allowing to replace selected text by the selected link leading to the desired LOD resource.

3 Benefits of the Annotation Box

The annotation box supports two major use cases: (i) to replace HTML text controls in a data input form, and (ii) to be used as a search box – an element often required on web pages. In both cases, it reduces the ambiguity of natural language text and helps the processor (i.e., the back-end of websites) or other users to correctly understand a term.

Annotations associated with text lay the foundation for automatic data processing in the back-end. Suppose that TripAdvisor makes use of our annotation box; it could then easily extract annotated geographic places, historic buildings, parks, or museums from users' collected comments and reviews. This would, for instance, make it possible to perform evaluations on the popularity of these entities in different seasons of the year. The knowledge gained in the process could be used to automatically suggest users the best time period to travel to a given city, or list the most popular places that travelers should visit.

Moreover, the box facilitates search. It enhances precision of the result, because besides text matching, it allows for LOD resource mappings in the back-end by leveraging *owl:sameAs* properties.

A website equipped with annotation boxes is capable of performing queries on top of LOD resources. For example, if every geographic place listed in the

[2] `https://github.com/dbpedia/lookup` (accessed 20 July 2015)
[3] `http://dbpedia.org/fct/` (accessed 20 July 2015)
[4] `https://www.wikidata.org/w/api.php` (accessed 20 July 2015)
[5] `https://en.wikipedia.org/w/api.php` (accessed 20 July 2015)

travel guides of a city would be associated with DBpedia resources, we can access these resources and get the construction year of each place. From that, we can calculate the average age of every city spot mentioned in a travel guide. Assume that a traveler visits the TripAdvisor website to find a city in a country to visit. We can then ask users for their preference (i.e., modern or antique city) to infer and recommend the most appropriate place for them.

Finally, the annotation box is simple to use. It requires no additional technical expertise, and takes end users only a few seconds to enrich their words with DBpedia resources. Developers can replace the traditional HTML input with the annotation box and empower the back-end with intelligent search and automatic data processing. The box is written in JavaScript and published on GitHub[6].

4 User Study

We conducted a small-scale user study to evaluate the annotation box. We chose eight subjects aged between 25 and 35; all generally spend at least one hour per day on the internet and can be characterized as being experienced in working with computers. We explained to the subjects how words can be annotated by using the box. After that, we asked the subjects to perform three tasks of increasing complexity as follows. (i) Simple annotation. The subjects annotate predefined text, that is: "*Vienna is the capital of Austria*". As they type, they are asked to annotate Vienna and Austria with the respective Wikipedia resources. (ii) Compound word annotation. We present the subjects a predefined sentence, that is "*Do not confuse it with another Vienna, which is a town in Virginia, United States*". They are asked to annotate two single words (i.e., Vienna and Virginia) and a compound word (i.e., United States). (iii) Single and compound word annotation. The subjects are asked to annotate all places mentioned in the following sentence: "*When travelling to Vienna, Austria, we visited Rathaus, Graben, St. Stephen's Cathedral, Vienna Ring Road, Hundertwasserhaus, Hofburg Palace, Schönbrunn Palace, Belvedere, Naschmarkt*". If some place is not available in the list containing suggested annotations, the subjects should manually search and tie the Wikipedia link to the text.

Fig. 2 and 3 show the results of the experiment. While the first two tasks are designed to help the subjects getting used to the box, we mainly use the third to evaluate the usability of the box. It contains eleven terms needed to annotate. The subjects needed 138 seconds on average to complete this task, which means they spent 12.5 seconds per term. Most of the spent time is used for locating the relevant resources. The task completion time plot leads us to the conclusion that adding annotations to text can be done with little effort even for a high number of annotations. Moreover, the time needed for completing the tasks does not vary significantly between the subjects, indicating that the process of creating annotations is efficient and straightforward. The subjects typically made one mistake in the second task as they confused *Vienna, Austria* with

[6] `https://github.com/datsat/Annotating-Box` (accessed 20 July 2015)

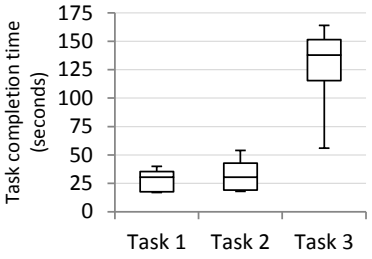


Fig. 2: Task completion time

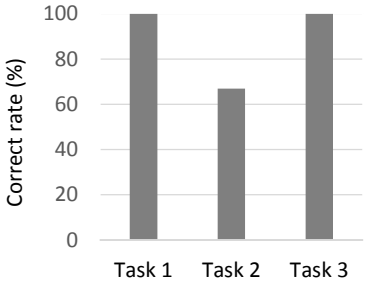


Fig. 3: Median correct rate

Vienna, USA. This can be seen in the bar chart for Task 2 where the median of correct annotations is at 67%. For the other tasks the median is at 100%.

At the end of the experiment, we asked the subjects to fill out a questionnaire in order to get feedback on the usability experience. They agreed that the box is relatively easy to use, but it would be easier if they could initiate the annotation process by right clicking or by pressing a single key. They also suggested to improve the process of locating resources which are not in the suggestion list. For instance, they would prefer not having to manually search and add the link to the text. The subjects enjoyed reading the annotations as it provided them additional information. However, they would only add annotations to their posts if it is really necessary, because it requires time and effort. All in all the results of this preliminary user study validate the efficiency and usability of the annotation box. Based on the feedback, we plan to further improve its usability and functionality.

5 Related Work and Conclusion

In the semantic web community *semantic annotation* has been an active field of research for several years. Semantic annotation describes a more granular approach than the tagging or the use of folksonomies [3] (i.e., a system of classification derived from collaborative or social tagging). The latter approaches assign keywords or terms to a whole document, speeding up search and helps us to find relevant and precise information. In contrast, semantic annotation enriches a word or a part of a document with context that is further linked to structured knowledge (e.g., a DBpedia page that provides information on a resource). Annotations are more informative than tags and allow to show results that are not explicitly related to the original search.

RDFace[7] is an online RDFa content editor that uses existing semantic web APIs to help users manage and embed RDFa contents to a web article. Users can manually add a triple, or simply select one or more NLP APIs to perform automatic named entity extraction [2]. Compared to enriching text with LOD resources using our annotation box, RDFa annotation requires specialized knowledge, meaning that users need to be familiar with semantic web concepts.

[7] `http://wiki.aksw.org/Projects/RDFaCE` (accessed 20 July 2015)

RDFace is appropriate for skilled users in a web content management system; However, it is not relevant for general users to quickly create simple web content (e.g., comments or posts). WYMeditor[8] is another WYSIWYM (What-You-See-Is-What-You-Mean) tool that allows for editing RDFa content; the functionality, however, seems not to be complete and development is already discontinued.

PoolParty thesaurus [9] is an example for a similar product that is already commercially available. It is a WordPress plugin that allows users to import a controlled vocabulary or retrieve a thesaurus from a SPARQL endpoint. It automatically analyzes a post to find words and phrases that match labels of a concept in the thesaurus. When hoovering annotated texts it displays respective tooltips. PoolParty also developed similar plugins for SharePoint and Drupal. The differences between PoolParty and our annotation box are: (i) PoolParty employs a high-level thesaurus whereas our box makes use of DBpedia resources. (ii) PoolParty only annotates text that matches limited concepts of the thesaurus whereas we annotate arbitrary text to DBpedia resources. (iii) We focus on real-time annotating while PoolParty annotates the whole text content after it is created. (iv) Because of the large number of DBpedia resources, we support manual annotating to enhance precision; the autocomplete feature will compensate the annotating time. Meanwhile, PoolParty implements an automatic approach where text is annotated without users interaction.

The idea of an autocomplete annotation box can also be found in services such as Facebook[10], ChatGrape[11] or Slack[12]. Users, when inputting text for a post or a chat message, can link text to resources such as friends, cloud documents, or calendar entries. However, these applications and their input boxes do not make use of semantic annotation; the added context is limited to their own resources.

To conclude, in this paper, we introduce an autocomplete-enabled annotation web component that can replace html text input areas. It enriches user-generated content with annotations of DBpedia resources to facilitate use cases such as semantic search and automatic data processing in the back-end.

References

1. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web 6(2), 167–195 (2015)
2. Mihalcea, R., Csomai, A.: Wikify!: Linking Documents to Encyclopedic Knowledge. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management. pp. 233–242. CIKM '07, ACM, New York, NY, USA (2007)
3. Peters, I., Becker, P.: Folksonomies: Indexing and Retrieval in Web 2.0. Knowledge & information : studies in information science, De Gruyter/Saur (2009)

[8] `http://wymeditor.github.io/wymeditor/` (accessed 20 July 2015)
[9] `https://wordpress.org/plugins/poolparty-thesaurus/` (accessed 20 July 2015)
[10] `https://facebook.com/` (accessed 20 July 2015)
[11] `https://chatgrape.com/` (accessed 20 July 2015)
[12] `https://slack.com/` (accessed 20 July 2015)

SPARQL Playground: a Block Programming Tool to Experiment with SPARQL

Paolo Bottoni and Miguel Ceriani

Sapienza, University of Rome, Italy
bottoni@di.uniroma1.it, ceriani@di.uniroma1.it

Abstract. SPARQL is a powerful query language for Semantic Web data sources but one which is quite complex to master. As the block programming paradigm has been succesfully used to teach programming skills, we propose a tool that allows users to build and run SPARQL queries on an endpoint without previous knowledge of the syntax of SPARQL and the model of the data in the endpoint (vocabularies and semantics). This user interface attempts to close the gap between tools for the lay user that do not allow to express complex queries and overtly complex technical tools.

1 Introduction

While the available Linked Data sources are increasing in quantity and diversity, their usage is still limited. One of the barriers for the adoption of Semantic Web standards, even by technology-savvy users, is their perceived complexity.

Whether someone wants to explore an RDF dataset or in general the Linked Data cloud, the options are usually either to use a Linked Data browser for a purely resource-centric view or to switch to writing queries using the SPARQL language [9], the standard query language for RDF. Writing SPARQL requires knowledge of the syntax and also a basic knowledge of the model underlying the dataset (vocabularies that are used, semantics that are implemented). As community of developers and consumers of Semantic Web technologies, we should challenge us to close this gap. There is a need for tools that may be used in a modular and progressive way to guide the users from the design of simple queries to complex ones.

Block programming languages, in which coding is done by dragging and connecting fragments shaped like jigsaw puzzle pieces, have been successfully used to introduce programming to non-experts. Recently, tens of milions of users have been exposed to the basics of programming using Blockly [5] as part of code.org's *Hour of Code* [1]. The same metaphor was used in Scratch [11] to create animations and games and in MIT App Inventor [16] to build Android Apps.

In [3] we proposed to use the block programming paradigm to design queries on Linked Data sources. Apart from the goals stated above, we decided, as in the philosophy of Block Programming, to design our tool as a way to gradually experiment the structure of the "real" underlying language and in the end to be able to switch to directly programming in that language. For these reasons the visual language mimics the

[1] https://code.org/about

structure of the syntax of SPARQL, while at the same time trying to avoid excessive verbosity that would lead to cognitive overload for the user. Compared to previous uses of block programming languages, this proposal addresses novel challenges due to two main specific properties: 1) the heterogeneous nature of Linked Data, that requires the ability to explore graph datasets even without any *a priori* knowledge; 2) the structural difference between procedural imperative languages for which this paradigm was previously used and a functional query language like SPARQL. To deal with these challenges we proposed a novel paradigm, favouring direct reuse of query results through the integration of the visual space used for query design and results visualization.

In the present paper we present a live demo of the tool. Through this online demo we want to promote discussion on the topic of visual interfaces for SPARQL and specifically evaluate the usability and reception of our proposal. Moreover, the user interface has been enhanced to permit the execution of queries on multiple SPARQL endpoints.

In the rest of the paper, Section 2 reports on related work while Section 3 presents the tool. Section 4 gives details on the implementation and the presented demo and Section 5 summarizes the proposal and draws some conclusions.

2 Related Work

Several interactive tools have been proposed to support the structured querying of RDF data sources, at various levels of abstraction and using different paradigms. A basic distinction can be made between: 1) tools that require writing and reading SPARQL syntax and 2) tools that provide other metaphors (usually visual) aimed at lowering the learning curve and providing more intuitive interaction. The first kind of UIs include advanced editors as YASGUI [12] or integrated environments as Twinkle[2], but to design the query the user has still to know SPARQL and the vocabularies used.

UIs of the second kind provide interaction with another representation of the query –textual or visual– that is then transformed to SPARQL to be executed. The text based UIs use forms –such as SPARQLViz [2]– or controlled construction of natural language statements –such as SPARKLIS [4]. These systems do not scale well when the query complexity increases and do not easily permit code reuse. As for the visual tools, most of them use a graph-based paradigm (NITELIGHT [13], QueryVOWL [8]), others use a dataflow-based paradigm (SparqlFilterFlow [7]), and at least one uses a combination of both (VQS [6]). Graph-based interfaces fit the RDF graph pattern matching model very well, while dataflow-based interfaces are effective in representing SPARQL functional operators (e.g., `UNION`). Nevertheless, both types of interfaces are highly inefficient in terms of space on user screen and often present problems with interaction.

A previous important proposal for using block programming for SPARQL queries is the SPARQL/CQELS Visual Editor designed for the Super Stream Collider framework [10]. In that case the blocks strictly follow the language structure and syntax and the tool requires at least basic knowledge of SPARQL to be used. Conversely, the user interface we propose is designed to provide blocks that should be mostly self describing and usable without knowing the SPARQL syntax in advance. Finally, for most of

[2] `http://www.ldodds.com/projects/twinkle/`

the existing tools the visualization of the result set is passive and often presented in an indipendent panel/window (e.g., in many Web-based interfaces the result page replaces the query page). In our proposal, on the contrary, results and query share there same workspace to allow for an exploratory pattern of interaction.

3 Proposed User Interface

The following were the basic requirements around which the user interface was defined:

1. users should not care about the syntax – hence visual clues and constraints should prevent syntax errors;
2. the need to input text by users should be minimized;
3. there should be direct ways to build commonly used structures;
4. users should be able to use the tool as a step to learn the SPARQL (textual) syntax – hence the used blocks should follow the structure of the language;
5. users should be able to work even without prior knowledge of the dataset – hence exploratory queries should be explicitly supported.

The queries are designed composing the set of available blocks. For example, Figure 1 represents a *select* query –against LinkedGeoData data set [15]– to get the names of the first three regions (the first administrative subdivision) of Italy by alphabetical order. The query is represented by the *select all* block and its sub-blocks. Among them, the sub-block connected to the *where* connection is a graph pattern and corresponds to the `where` clause of the query. Inside graph patterns and expressions, different types of graph terms can be used: *IRIs* (represented in brown and using the prefixed notation), *variables* (using Blockly appearance of variables for consistency), and *literals* (represented in different colours according to their type, numeric, string or boolean). The SPARQL query corresponding to Figure 1 is:

```
SELECT DISTINCT * WHERE {
  lgdt:relation365331 lgdo:members [?p ?member].
  ?member
    lgdo:role 'subarea';
    lgdo:ref [rdfs:label ?subareaName].
  FILTER(LANGMATCHES(LANG(?subareaName), 'en'))
}
ORDER BY (?subareaName) LIMIT 5
```

However, in order to design such a query some knowledge about the specific dataset (that there exists a resource `lgdt:relation365331` representing Italy) and the used vocabularies (that the property `lgdo:members` associates an area with a container of items, of which the ones with `gdo:role` equal to "subarea" are administrative subdivisions of the area) is still required.

If the dataset is unknown this information is usually gathered through preliminary, explorative queries. We thus designed the user interface especially to favour the reuse of query results in the same or new queries. The *execution* block is used to execute a query and to show the result set as soon as it is available. The produced result set is shown again in the form of blocks, which can be dragged to other parts of the workspace and connected to other blocks. Results from a query can thus easily be used as parts of another query. For the previous query the tabular results can be seen in Figure 1 as well.

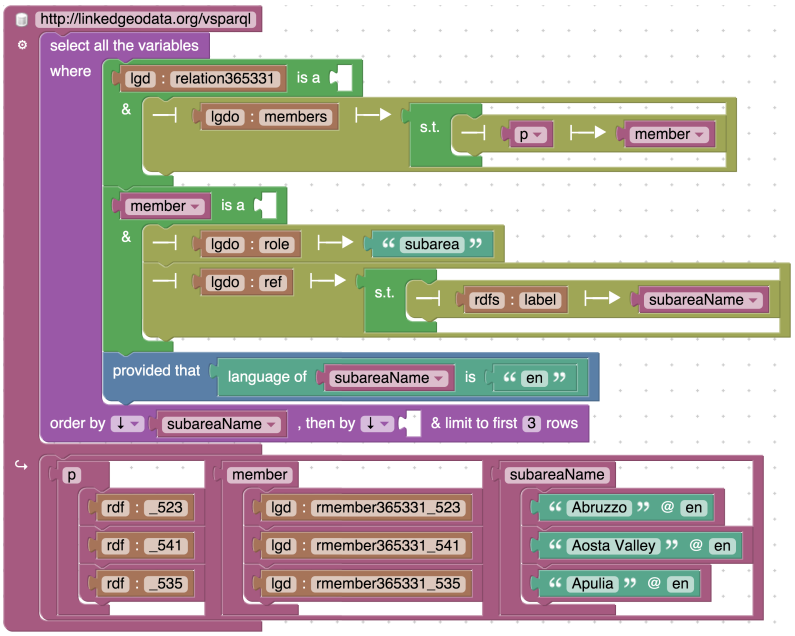


Fig. 1. Execution of a query to get the first three English names of regions of Italy.

The execution block provides also a field to set the remote SPARQL endpoint to which the query is sent –in this case the public SPARQL endpoint for LinkedGeoData[3]. There is no formal limit to the number of execution blocks that can be used, new queries may thus be built while keeping aside previously built queries and their results.

Ideally, the results of execution would arrive within a small time. While this may be true for not-too-complex queries against SPARQL endpoints that perform well, it cannot be guaranteed for the general case. For that reason query executions are non-blocking, i.e. the user interface stays reactive while waiting for a result from the server. The user is thus able to keep working at the same query or other queries while the query is being executed. If a query is modified during execution, the execution is aborted and restarted with the updated query.

When exploring a new dataset, knowing the used vocabularies and looking for specific resources are a common need. For this reason the toolbox already contains some pre-built queries that can be used to look for resources, classes and properties used in the dataset. These pre-built queries are just sets of pre-connected blocks that can be freely rearranged and decomposed on the workspace. Figure 2 shows the query prepared to look for specific resources of a certain type, modified to look for a resource labelled "Italy" and of type `lgdm:Relation` (that is the class used in LinkedGeoData for geographical composite elements). The result of the query, together with the result of similar explorative queries, can be reused to write a query as the one in Figure 1.

[3] `http://linkedgeodata.org/vsparql`

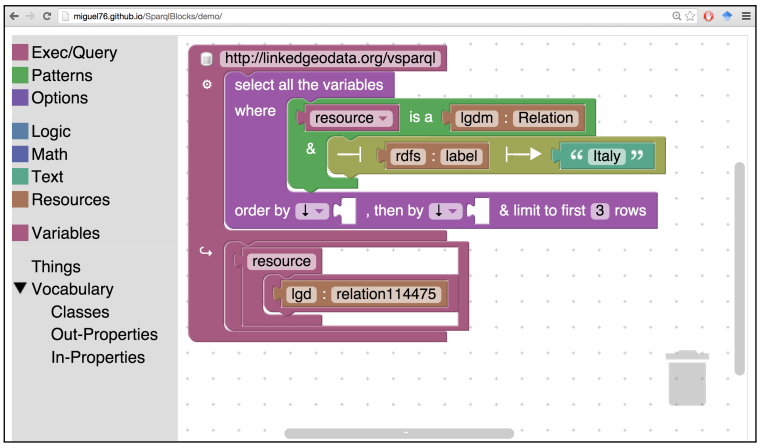


Fig. 2. View of the UI after getting a resource corresponding to Italy in LinkedGeoData.

Figure 2 also shows the user interface (enlarged for readability). The toolbox on the left –from which new blocks may be dragged on the workspace– has categories corresponding to the different types of blocks. The last two categories –Things and Vocabulary, with its subcategories– contain the aforementioned pre-built queries.

4 Implementation and Demo

The tool is based on an extension of the Blockly JavaScript library, working entirely on the client side. We extended the library to supply the specific blocks needed for SPARQL queries and execution. We also added the necessary code to generate SPARQL fragments from the blocks. The SPARQL execution block listens for changes in its query connection; each time the query changes, the corresponding SPARQL query is generated and sent to a SPARQL endpoint. The SPARQL endpoint used is set as a field of the execution block. The results are used to dynamically generate the result block and its sub-blocks. The standard prefix definitions from *prefix.cc*[4] are used to add prefix declarations in the query sent to the endpoint and to convert the IRIs in the result to the prefixed notation.

The online demo[5] is an instance of the tool provided to everyone willing to experiment with this new user interface. By default the queries are executed against the SPARQL endpoint of DBPedia[6] [1], but –as shown in the examples in Section 3– any other public SPARQL endpoint can be accessed[7]. Using the context menu on the blocks, queries and fragments of queries may be exported as SPARQL. Query results may also

[4] `http://prefix.cc/`
[5] `http://miguel76.github.io/SparqlBlocks/demo`
[6] `http://live.dbpedia.org/sparql`
[7] As the tool runs on the browser, the endpoints have to be CORS-enabled; non CORS-enabled endpoints may be reached through a proxy.

be exported in JSON format[14]. Apart from using the online demo, the code itself may be also freely forked or donwloaded from GitHub[8].

5 Conclusions

We developed a new visual user interface to allow non-experts to build SPARQL queries. The tool does not require prior knowledge of the used dataset and vocabularies, favouring an exploratory and constructive way of building queries. An online demo –from which any public SPARQL endpoint can be queried– has been setup to showcase the user interface to the communities of Semantic Web developers and reasearchers, in order to have feedback from a wider audience and foster discussion in this field.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proc. of ISWC 2007. pp. 722–735. Springer (2007)
2. Borsje, J., Embregts, H.: Graphical query composition and natural language processing in an RDF visualization interface. B.S. Thesis, Erasmus School of Economics and Business Economics, Erasmus University, Rotterdam (2006)
3. Bottoni, P., Ceriani, M.: Linked Data Queries as Jigsaw Puzzles: a Visual Interface for SPARQL Based on Blockly Library. In: Proc. of CHItaly 2015. p. [To Appear]. ACM (2015)
4. Ferré, S.: Sparklis: a SPARQL Endpoint Explorer for Expressive Question Answering. In: Proc. of ISWC 2014 Posters & Demonstrations Track. vol. 1272. CEUR-WS (2014)
5. Fraser, N., et al.: Blockly: a visual programming editor (2013)
6. Groppe, J., Groppe, S., Schleifer, A.: Visual Query System for Analyzing Social Semantic Web. In: Proc. of the WWW '11. pp. 217–220. ACM (2011)
7. Haag, F., Lohmann, S., Bold, S., Ertl, T.: Visual SPARQL Querying based on Extended Filter/Flow Graphs. In: Proc. of AVI 2014. pp. 305–312. ACM (2014)
8. Haag, F., Lohmann, S., Siek, S., Ertl, T.: QueryVOWL: Visual Composition of SPARQL Queries. In: Proc. of ESWC 2015 Satellite Events. Springer (2015)
9. Harris, S., et al.: SPARQL 1.1 Query Language. W3C REC 21 March 2013
10. Quoc, H.N.M., Serrano, M., Le-Phuoc, D., Hauswirth, M.: Super Stream Collider-Linked Stream Mashups for Everyone. In: Proc. of the Semantic Web Challenge at ISWC 2012 (2012)
11. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al.: Scratch: programming for all. Communications of the ACM 52(11), 60–67 (2009)
12. Rietveld, L., Hoekstra, R.: YASGUI: Not Just Another SPARQL Client. In: Proc. of ESWC 2013 Satellite Events. pp. 78–86. Springer (2013)
13. Russell, A., Smart, P.R., Braines, D., Shadbolt, N.R.: NITELIGHT: A Graphical Tool for Semantic Query Construction. In: Proc. of SWUI '08. vol. 543. CEUR-WS (2008)
14. Seaborne, A.: SPARQL 1.1 Query Results JSON Format. W3C REC 21 March 2013
15. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: LinkedGeoData: A Core for a Web of Spatial Open Data. Semantic Web Journal 3(4), 333–354 (2012)
16. Wolber, D., Abelson, H., Spertus, E., Looney, L.: App Inventor. O'Reilly Media, Inc. (2011)

[8] `https://github.com/miguel76/SparqlBlocks`