Data Picking Linked Data: Enabling Users to create Faceted Browsers

Daniel A. Smith
IAM Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ
UK
ds@ecs.soton.ac.uk

Igor O. Popov
IAM Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ
UK
ip2g09@ecs.soton.ac.uk

mc schraefel
IAM Group,
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ
UK
mc@ecs.soton.ac.uk

ABSTRACT

Despite the massive amount of data on the Web in Linked Data format it remains, however, difficult to explore, aggregate and consume this data. The access barrier is particularly higher for users with little or no technical experience. End users, with a vested interest in data but little technical expertise, typically rely on simple tools, such as spreadsheets, to store and analyze data. On the other side, publishers cant always model or republish their data to appeal to every particular user group. In this paper we report on our attempt to lower this barrier. We suggest that both parties, publishers and users, can benefit from tools which allow them to quickly exchange data by (1) allowing the publisher to quickly mash up slices of different sources of data centered around a particular topic of user interest and (2) allow the user to manipulate facets of this data and export it in a familiar format. To facilitate this we employ the Data Picker, a tool for the mSpace faceted browser that allows publishers of Linked Data to quickly set up a faceted explorer from multiple data sources, which employ a SPARQL endpoint. One of main advantages of this approach is that it is easy to assemble a spreadsheet from several different sources thus utilizing the integrative properties of Linked Data while outputting it in a format familiar to the end user. Once the faceted browser is set up around a particular subject, the user is free to manipulate the fields by selecting the facets and subsequently generate the spreadsheet, allowing the user to carry on additional tasks. We tested the tool on our dataset of UK Public Sector Information (PSI) Linked Data using a number of test scenarios, which we set up as interesting questions requiring multiple sources of data to answer.

Keywords

Linked Data, Semantic Web, Faceted Browsing, Web Science

1. INTRODUCTION

Linked Data is about making data available on the Web in open, accessible and interoperable format. In the past there years there a significant number of datasets have been published in Linked Data format. In the last year in particular, there is an emphasis (particularly in the UK and US) of converting Public Information Sector (PSI) data into Linked Data. Opening PSI data has been embraced by many governments recently, as a way to boost transparency in government and allowing the public to reuse the data either by providing new application services for the public or make interesting insights from a pool of diverse government data. Recently we have seen a convergence of both the PSI data publishing community and Linked Data community on ways how PSI data can benefit from being converted into a Linked Data format.

While Linked Data provides self-describing data which has the property of discovering new and interesting data through following links, publishing it in a such a format still sets a barrier for average users, those with little or no technical skills, to access and consume the data. PSI data published as Linked Data gives a good example of such a barrier. Consumers of PSI Linked Data are usually either Web developers, with an interest of consuming interesting information by developing applications which use the data, or end users, people with no technical skills that want to be able to find interesting data and process them in already familiar tools. For the latter in particular, the problem is mostly due to a lack of tools which allow users to search and browse these data sets efficiently and in a way that abstracts the complexities of the underlying graph structure of the data. One of the Linked Data principles includes dereferencing *URIs* to provide useful information regarding data, however this information is nearly useless to the average user who tends to generally associate data with spreadsheet representations. Generic data browsers, on the other hand, offer one or at most a few ways to browse or view the data. Most of these interfaces, however, expose too much unnecessary knowledge to the user about the schema of the data . Most general consumers of data, particularly those with a interest of viewing and analyzing government data, typically rely on simple tools such as Excel to present and analyze data, a fact exemplified by the amount of excel and pdf spreadsheets on public government data repositories. On the other hand developing hand crafted applications and special purpose browsers for each dataset is too costly for data provides.

To address these challenges we created a prototype that allows publishers to describe the parts of their Linked Data

Web Science Conf. 2010, April 26-27, 2010, Raleigh, NC, USA.

that are suitable for direct browsing by humans, and generates an mSpace faceted browser using that description. Our tool creates a definition of the data, which is then itself published as Linked Data, so it can be shared and used as the basis for others to create their own faceted browsers based on the published definition. The mSpace faceted browser framework allows users to browse a data source by any facet, and by making selections in that facet, records are filtered. In addition, available selections in other facets are also filtered, aiding users to building knowledge about the domain. In order to address users answering statistical questions about Linked Data we have extended the mSpace framework to allow records to be exported as spreadsheets. Users can than load the spreadsheets using software such as Microsoft Excel, and leverage its powerful statistical aggregation methods, such as PivotTable. We demonstrate that through PivotTable, a aggregation method that allows row of spreadsheets to be grouped, users can then answer statistical queries of data. We motivate our technique through an example of a user wishing to find out which political party claimed the most in MPs expenses in the UK. While our example uses PSI data, our tool is generic in design and can be deployed over any type of Linked Data.

This paper is structured as follows. First we present related work by reviewing various data publishing and information exploration interfaces . Next we present the published PSI data [6] and its underlying ontology that was used in the case study, the mSpace data picker and exporting functionalities. We then describe a scenario using the data mentioned in Section 3. Finally we conclude and present future work.

2. RELATED WORK

One instance of data picking is Fresnel [7], a display vocabulary for RDF data. Fresnel allows publishers to specify *lenses* over the data, which are portions of the graph which the publisher would like to display. Then HTML/CSS is used to render *views* i.e. the presentation of the data. While Fresnel is good for quickly setting a presenting simple abstracted views of RDF data, providing any browsing capability is a task left to a developer to provide.

Data picking has recently been used to create more easy-to-use APIs[1] over RDF data. However these are targeted toward developers who are familiar with REST APIs rather the average users. The publishers can easily create view of the data similarly to Fresnel and provide an interface to the abstracted data to developers. The API allows several affordances such as getting the data in pagination, do filtering the data by a certain property discover additional resources etc. These tools help developers create applications over RDF data more easily, however creating a full browser still requires a lot of programming on part of the programmer using the API.

On the other hand exploratory interfaces are usefully adding value over data sets that cannot be explored, however there are various problems with these systems, and challenges that existing work has attempted to address. For example, Tabulator [1] is useful for visualizing knowledge that has been formalized, but if that knowledge is not codified in RDF, it cannot be displayed. Even if it has been codified in RDF, there may be barriers to easy visualization with Tabulator, for example a lot of information may be collected in a very large file; Tabulator, and, more specifically, the paradigm of client-side data analysis has scalability problems, particularly within a web browser, as is the framework that Tabulator utilizes. Orthogonally, knowledge may be split among so many smaller files, that loading them all in to Tabulator's knowledge base would take the user a long time to add.

Similarly, "Exhibit" [2] is a web-based UI widget designed as an easy-to-deploy column browser for use in websites. It accepts data in the form of rows, with each column of data in a row being an entry in each column in the browser. The benefit of the Exhibits approach is that concepts need not be defined semantically, and tabular exports are frequently available on many database systems. The downside is that the system does not scale to well past hundreds of concepts, that multi- linked columns (i.e. Composer / Composer Country) are not possible, given the flat nature of the data table form, and that provenance of data is not shown.

One of the challenges faced by interfaces developers [4], particularly on the web [5] is with dealing with heavily-distributed data, is optimizing the flow of data to the client, so that the user is not left waiting for data that they didn't want or need to explore, which is a double-headed problem. Firstly it involves determining what information the user wants, a question of how to develop semantic data discovery systems, and how to segment knowledge into chuck of relevance for users, either before-the-fact or through query interfaces, such as SPARQL[8]. The problem also involves how to maintain adequate performance when querying knowledge bases and engines for results, and how to push these to the user for the best possible user experience.

3. MSPACE DATA PICKER

3.1 mSpace Faceted Browser

The mSpace faceted browsing interface [3] provides a framework and paradigm for browsing over data spaces by exposing attributes of data as facets. Users can then select items of interest, and the interface updates the visible facets showing only those items that share the selection the user has made. For example, in an mSpace about Members of Parliament, facets of *MP*, *Political Party* and *Total Expenses* can be exposed. By making the selection of "Conservative Party" in *Political Party*, the list of people in the *MP* facet is updated to those which are in the Conservative party, and the list of expenses is limited to those made my members of the conservative party. A benefit of this type of interface is that the user does not need any prior knowledge of the domain to make choices, all of the possible options are exposed to them, and by making selections they can build knowledge of the area by seeing how the other facets change.

3.2 User-selected Picking Data

In order to allow users to explore data sources and create interfaces from them, we have designed a prototype *data picker interface*. The interface allows users to load data sources, and explore them for facets of interest. A breakdown of the backend architecture of the Data Picking Interface is shown in Figure 1.

Due to the potential complexity of the task of exploring large data sets, there is potential for an interface to become overwhelming if not carefully designed to prevent this. As such, when designing the interface for the data picker,

[1]http://code.google.com/p/linked-data-api/

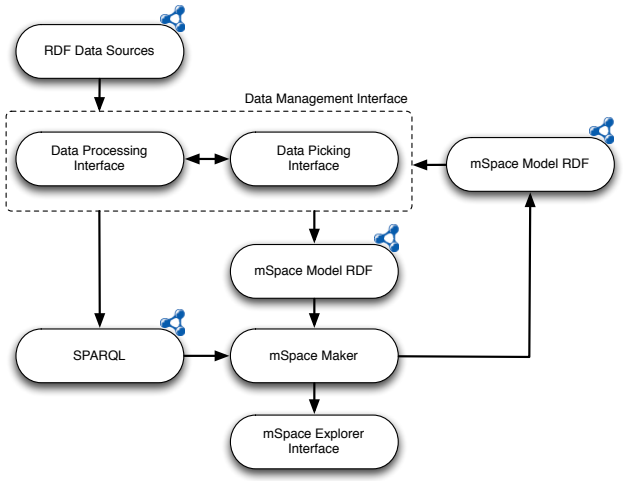

Figure 1: Architectural overview diagram of the data picking interface to support User Created Interfaces, showing the mSpace Model RDF being output by the Data Picker, which is used by the mSpace Maker to create a faceted interface.

we have been mindful to attempt to make the interface as lightweight as possible.

One of the ways in which we have done this, is to design the interface to default-include data, rather than require the user to specify that the data they are exploring should be shown in the final interface that they are creating.

Through the data picker interface it is possible to create an interface in very few steps. The shortest set of interactions is as follows:

1. Click, or enter the URI of a SPARQL endpoint, and click submit.
2. Select a Class from the list.
3. Click "make mSpace."

By following these actions, the user will be presented with an mSpace interface that contains data from the chosen SPARQL endpoint. A screenshot of a sample interaction where the user has selected the "People" class, is shown in Figure 2.

Through the use of the Data Picker, a Facet Ontology definition of the data source is created. This definition is used to create the mSpace faceted interface over the user's data. The availability of the Facet Ontology definition have benefits in itself, above the creation of the mSpace. Specifically, the definition can be loaded into the Data Picker by other users that wish to create a view on the same data source, but customize which facets they want to view, and build upon the work of the first user.

4. WALKTHROUGH USING PSI DATA

In this section we illustrate our approach using data about the expenses of Members of Parliament of the United Kingdom. In our scenario we assume that a user wants to find expenses done by each party in different geographic locations and is using a mSpace interface set by the publisher

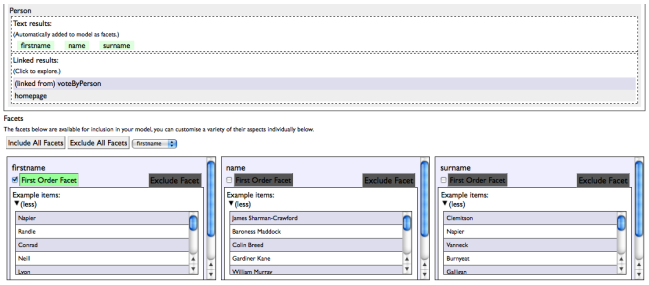

Figure 2: A screenshot of the Data Picker interface, showing the "People" class selected, which has extracted three facets: firstName, Name and surname.

using the Data Picker. Following the release of the data, publishers might want to showcase only certain parts of the data which might be interesting to be browsed and then exported by users for further analysis. For example, our datasets (depicted in Figure 3) contain data on various topics all linked through a hierarchically linked administrative geography. As the figure shows, data pertaining to mortality, hospital waiting times, crime as well as information on MPs all link to certain geographical entities, which in turn are linked among each other in a hierarchy. Publishers of such diverse data can use the Data Picker interface to pick data across the ontology to create a faceted browser. In our example, we are using a subset of the dataset by selecting a subgraph of the data holding information relating to MP expenses. In particular, we wish to quickly set up a faceted browser showing expenses made by Members of Parlament, which we want to be able to filter based on the hierarchy of geographic data.

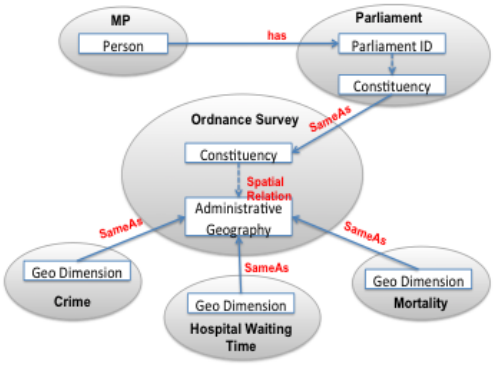

Figure 3: An overview of the PSI ontology showing the various datasets, the key classes and the relationships between them.

The mSpace Data Picker generates the mSpace interface to run over a SPARQL query. The Data Picker interface shows the user the various classes (or type) found in the dataset. The user starts by selecting a goal class. The goal class specifies the items which the mSpace will show and filter over in the facets. In our dataset, a particular MP is modeled using two RDF types - a *foaf:Person* and a *MPIdentity*. The person class holds personal information re-

garding the MP such as name, surname, homepage etc. An MPIdentity defines a role of a person and is used to denote a single term in office by a certain person. The class essentially holds all the information pertaining to a certain time period in office such as the duration of the term, the political party which the MP was representing for that term in office and the constituency he/she was representing. Thus, a person can have one or several terms in office, which might differ in the political party the MP was representing in each term or more commonly the constituency for which the MP served. In our case we want to be able to differentiate between terms in office so we want our mSpace to filter information over MPs and give also personal information as well as expenses data. We can pick the MPIdentity as the goal class but additionally we pick data from the fields spanning out of the MPIdentity class, so it also also show the persons personal information since (as Figure 4 shows) the expenses of a MP.

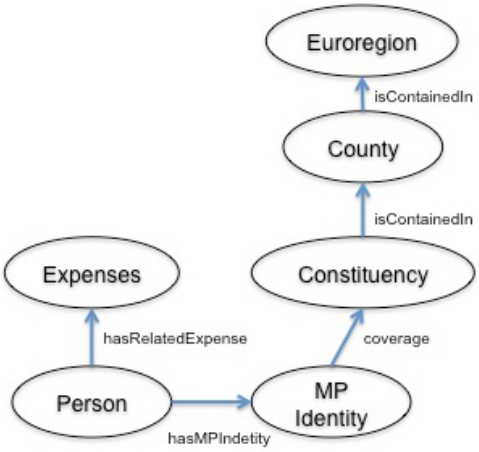

Figure 4: General overview of the ontology

Once the goal class is selected the user can proceed to select facet to be used to filter the data. When selecting a goal class in the Data Picker the user can see all of its literal and object properties. The user is presented with an initial selection of facet spanned by these properties. Clicking on any of the literal properties adds facet to the generated mSpace. Excluding any facets is also possible at any time. The user can additionally specify a first-order facet which is a facet in which every selection uniquely identifies with one goal item. The creator of the mSpace can either select facets move through different portions of the data through navigating through the object properties. For each of the explored objects, the user can select facets from the literal properties of the current facet.

In our example, from the MPIdentity class we can pick a party facet. We then navigate from the MPIdentity class to the constituency, then to county and finally a Euroregion. Each of these are objects, so in each of them we pick the label as to generate a facet representing that region. As the facets are being picked a live view of the current mSpace configuration is shown to the publisher. Once configuring all the facets is done, the user can simply export the newly created mSpace on to template site and can be additionally configured to show the data of the items in a certain way (Figure 5).

Figure 5: Resulting mSpace.

The created mSpace can now be used by users to view MPs, their expenses, and filter them by different geographic regions and by party. The exporting functionality of the browser allows to export any filtered view of the data. For example a user might want to create a spreadsheet only having MPs of a certain party or to have MPs from a certain constituency or counties. Continuing our example, we wish to create an aggregate of expenses done by each party. Exporting data on MPs filtered by party allows a user to easily create a pivot table to sum up the corresponding values in the spreadsheet as shown in Figure 6. Thus the user was able to quickly get an answer to his inquiry. The faceted browser allows to modify or get alternative views by choosing a different geographic regions or parties.

5. CONCLUSIONS AND FUTURE WORK

Our example has shown that advanced browsing capabilities can be set up quickly over Linked Data even by publishers of data who do not possess extensive knowledge in languages such as RDF or SPARQL. The mSpace Data Picker allows data publisher to create a quick faceted view over their data without any SPARQL manipulations - they just need to know and understand the schema of the data. Once the faceted browser is set up, end users can quickly browse through and export the data in spreadsheet format.

In the future we would like to extend the capabilities of the interface beyond performing only faceted browsing over data. For example one extension can be providing basic statistical functions which allow simple statistical insights to be preformed over the data. Additionally, interfaces can provide capabilities for end-user generated content such as deriving properties similarly to how a formula in Excel can create a new column by preforming computations/transformation on one or more columns. In such a way data consumers can become active participants in creating and maintaining Linked Data over the Web.

Acknowledgements The research leading to this paper was partially supported by the EnAKTing project, funded by EPSRC project number EI/G008493/1.

Sum of TOTAL ALLOWANCES CLAIMED, INC TRAVEL, 2007-2008	
Party	Total
Conservative	21175233
Conservative	985326
Consevative	141792
Democratic Unionist	158903
Democratic Unionist Party	393212
DUP	624606
http://spreadsheets.google.com/ccc?key=phNtm3LmDZEObQ2itmSqHIA	0
Independent	454041
Independent (Changed from Labour 17 Sep 2007)	163775
Independent (Changed party from Conservative 29/01/2008)	130947
Independent Labour	139210
Labour	41901067
Labour	1247271
Labour (Changed from Conservative 26/06/2007)	142857
Labour (until 2005)	
Labour and Co-operative	326145
Lib Dem	780786
Liberal Democrat	6845131
Liberal Democrat	885412
Plaid Cymru	292953
Plaid Cymru	165765
Scottish National Party	599917
SDLP	156902
Sinn Féin	681235
SNP	311371
UUP	134004
(blank)	74522
Grand Total	78912383

Figure 6: The Pivot Table output of the MP Expenses data, showing total expense claims, grouped by political party. Data is unedited, and highlights a "dirty data" problem with notes, URIs, duplicates and typos present in the "Party" field. Data is as at 8 March 2010.

6. REFERENCES

[1] T. Berners-lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *In Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006.

[2] D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: lightweight structured data publishing. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 737–746, New York, NY, USA, 2007. ACM.

[3] m.c. schraefel, M. Wilson, A. Russell, and D. A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006.

[4] J. Nielsen. *Usability engineering*. Morgan Kaufmann, 1993.

[5] J. Nielsen. The need for speed. *Alertbox*, 1997.

[6] T. Omitola, C. L. Koumenides, I. O. Popov, Y. Yang, M. Salvadores, M. Szomszor, T. Berners-Lee, N. Gibbins, W. Hall, mc schraefel, and N. Shadbolt. Put in your postcode, out comes the data: A case study. In *7th Extended Semantic Web Conference*, 2010.

[7] E. Pietriga, C. Bizer, D. Karger, and R. Lee. Fresnel - a browser-independent presentation vocabulary for rdf. In *In: Proceedings of the Second International Workshop on Interaction Design and the Semantic Web*, pages 158–171. Springer, 2006.

[8] E. Prudhommeaux, A. Seaborne, et al. SPARQL Query Language for RDF. *W3C Working Draft*, 23, 2005.