GEORDi: Supporting lightweight end-user authoring and exploration of Linked Data

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

Daniel Alexander Smith
School of Electonics and Computer Science
University of Southampton
SO17 1BJ, Southampton
UK
ds@ecs.soton.ac.uk

Max Van Kleek
School of Electonics and Computer Science
University of Southampton
SO17 1BJ, Southampton
UK
emax@ecs.soton.ac.uk

m.c. schraefel
School of Electonics and Computer Science
University of Southampton
SO17 1BJ, Southampton
UK
mc@ecs.soton.ac.uk

Gianluca Correndo
School of Electonics and Computer Science
University of Southampton
SO17 1BJ, Southampton
UK
gc3@ecs.soton.ac.uk

Nigel Shadbolt
School of Electonics and Computer Science
University of Southampton
SO17 1BJ, Southampton
UK
nrs@ecs.soton.ac.uk

ABSTRACT

The US and UK governments have recently made much of the data created by their various departments available as data sets (often as csv files) available on the web. Known as "open data" while these are valuable assets, much of this data remains useless because it is effectively inaccessible for citizens to access for the following reasons: (1) it is often a tedious, many step process for citizens simply to find data relevant to a query. Once the data candidate is located, it often must be downloaded and opened in a separate application simply to see if the data that may satisfy the query is contained in it. (2) It is difficult to join related data sets to create richer integrated information (3) it is particularly difficult to query either a single data set, and even harder to query across related data sets. (4) To date, one has had to be well versed in semantic web protocols like SPARQL, RDF and URI formation to integrate and query such sources as reusable linked data. Our goal has been to develop tools that will let regular, non-programmer web citizens make use of this Web of Data. To this end, we present GEORDi, a set of integrated tools and services that lets citizen users identify, explore, query and re-present these open data sources over the web via Linked Data mechanisms. In this paper we describe the GEORDi process of authoring new and translating existing open data in a linkable format, GEORDi's lens mechanism for rendering rich, plain language descriptions and views of resources, and the GEORDI link-sliding paradigm for data exploration. With these tools we demonstrate that it is possible to make the Web of open (and linked) data accessible for ordinary web citizen users.

Categories and Subject Descriptors

H5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*; H5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*User issues*

General Terms

Design, Human Factors.

Keywords

Linked Data, Publishing, User Interfaces, Semantic Web, Usability

1. INTRODUCTION

Over the past 18-24 months the US and UK governments have released copious amounts of data generated by almost all government departments on subjects from national defence budget spending to crime statistics to map coordinates. The expressed goal of this "open data" production in large part has been to foster greater governmental transparency by offering greater access to government department data.

While the ambition is noble, a critical problem is that there is a vast gap between citizens having access to this dumped data, and citizens being able to do anything meaningful with it. There have been several celebrated examples of what kinds of questions this data can enable to be explored. USAspending.gov is a compelling visualisation that lets people explore queries such as defence spending per region. Unfortunately, it does not allow a citizen either to change the questions asked of the data set or to augment the data with another source, say UK budget spending to compare what each country spends on similar matters. The demonstration, therefore, rather than being open, is just as closed a data system as any Web 2.0 mash up.

The current state of the art of open data means that, as with Web 2.0 mash ups, unless one is a programmer, one will not be able to readily access open data for one's own queries; unless someone else makes a mash up of open data that suits a person's interests, that person is largely out of luck. Resources that are supposed to be free and open to the public remain effectively inaccessible.

Over the past 12 months as part of the EnAKTing research project in the UK, we have been working to develop tools for citizens so that one no longer has to be a programmer geek to be able to discover, explore query and represent web data. For instance, someone looking for causes of high mortality rates[1] across regions in the UK may wish to pull together data on pollution, crime and hospital waiting times for related regions over time. The data where it exists is often in large spreadsheets. Any integration and representation either needs to be encoded into a specific program like USA spending or comparisons in sheets aligned and explored manually. Such operations may never be shared or may end up as another one off application.

Our approach takes the programming out of the equation, enabling citizens to focus on what they want to do rather than on building up tools to support that desire. To accomplish this engagement with the Web of Data, we have developed a Web of Data process that facilitates any stage of the publishing, discovery, exploration, query and representation process. First, we make it straightforward to convert raw open data from say spreadsheets into RDF. Converting data to a linkable format means that it is far easier for a person to associate data from one data set to another linked data set. Second, Linked Data itself is largely machine rather than human readable oriented. So we have developed a catalogue to enable plain language views of any availalbe-to-be-linked data set. Third, we have a Linked Data explorer that lets a person open and explore any of these data sets. Fourth, we have an Association tool that enables any relevant data to be connected with any data set being viewed.

While the approach described above sounds straight-forward, the solutions to enable these steps has been far from simple. Our approach in developing these tools, however, has been to focus on problems faced by citizens; in order to create a usable citizen experience, we have had to address some difficult challenges, using mainly semantic web technologies. The result has been the GEORDi framework, a set of interconnected data exploration widgets that users can use to interact and sculpt their own data with public data.

In the following paper, we discuss each of the GEORDi service components in turn. We describe the related work pertinent to each component, and identify the challenges and architectural solutions in each case. We conclude with a discussion of the successes to date, lessons learned and outstanding areas for future development.

2. GEORDI FRAMEWORK

GEORDi is both a framework and a set of tools to enable citizens to work with open and Linked Data in the Web of Data as readily as they might engage with spreadsheet data. The spreadsheet metaphor has been proposed before by Tim Berners-Lee [3] as an appropriate paradigm for working with semantic web data. We agree, and like Berners-Lee, we use this metaphor both for the degree of cognitive skills needed to make sense of and operate tools in the Semantic Web, and as a metaphor for interacting with the data. As we show below, we also extend the metaphor beyond Tabulator's use of spreadsheets only to represent results; we use a spreadsheet variant to represent queries and query interaction as well. Mainly however, the challenges we explore with GEORDi (described in the implementation sections as well) are how to develop tools for heterogeneous data linking and sensemaking that remove the requirement for a citizen to be a coder to be able to craft dynamic queries across arbitrary data sources.

We note that the following is not research on usability, but more on engineering driven by user-oriented requirements/challenges such as "cannot require *any* knowledge of RDF or URI's to run a query".

2.1 Making spreadsheet data linkable

The first challenge users face is converting the spreadsheet data into RDF. This entails developing an ontology and vocabulary based on the spreadsheet, and create URIs to represent the individual bits in the data. Most citizen users, however, have no knowledge of the notion of ontologies, schemas or graph models, nor have they ever used URIs to represent data. They therefor rely on end user tools to assist them with the desire that these tools make that process as transparent and automatic as possible.

Automatic and semi-automatic conversion of data from complex spreadsheets can be done in one of three ways: a user could make a custom made script to convert the data, use an existing convertor to RDF, or use advance capabilities in spreadsheet tools to create and export RDF files. Programming is clearly beyond consideration for any end user. Several tools for automatic and semi-automatic conversion, however, do exist. RDF123[6] provides a powerful tool to convert spreadsheets of various schemas in a two step process. The first part is a translating a spreadsheet by using a mapping that explains how it should be converted into RDF. The second part is a GUI interface that let users author these mappings. RDF123 allows users to create a mapping for any spreadsheet so it could potentially be used over several spreadsheets conforming to the same schema. While this allow flexability they require users to learn yet another new tool to just to author mappings. On the other hand tools such as Babel[2] and several other convertors[3] transform data without the requirement of mappings but only allow conversion from a single format.

Aside from convertors, a manual approach to conversion means using existing spreadsheet tools to aid the conversion to RDF. Recently, data authoring and cleaning tools such as Google Refine[4] [5] have demonstrated how using ad-

[1] `http://www.statistics.gov.uk/downloads/theme_population/Table_3_Deaths_Area_Local_Authority.xls`

[2] http://simile.mit.edu/babel/

[3] http://esw.w3.org/ConverterToRdf

[4] http://code.google.com/p/google-refine/?redir=1

[5] http://www.jenitennison.com/blog/node/145

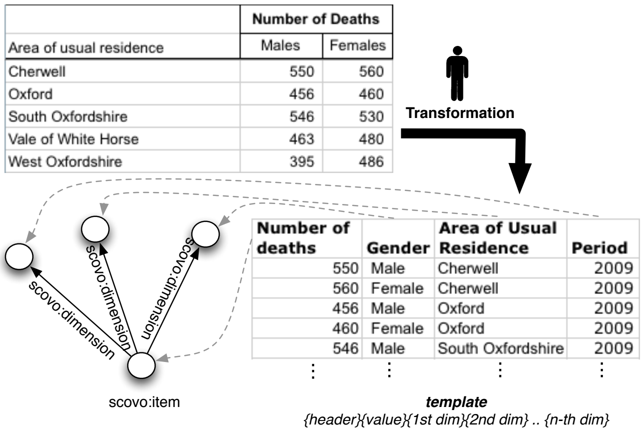
vance features of spreadsheet tools to assist in converting spreadsheet data into RDF. There are, however, two main problems for such techniques to be adopted by end users. First, the still expose the user to unnecessary details such as URI creation. Second, while they require less time to do than programming a script they still require some basic programming knowledge and are thus intended more for intermediary users, those that have some basic programming skills, rather than casual end user's with no programming skills.

A key feature of Exhibit [11] and the follow up Dido [12] has been the ease with which a person can translate a spreadsheet of data into a resource that can be parsed into a facet browsing interface for easy exploration and manipulation, super for single source data. The disadvantage of the approach is that it is limited to single (and thus isolated) data sets and is limited in terms of scale. We need Exhibit/Dido's ease of spreadsheet conversion, but that will facilitate large data sets and multiple, arbitrary source cross linking.

With GEORDi we allow user to upload spreadsheets that conform to a number of predefined templates. We thus rely on the knowledge of end users to use spreadsheet tools to author new or do simple transformations to existing spreadsheets, such as transposing data, to transform them into a template that GEORDi can understand and translate into RDF. Once the data is in the proper template, the user can simply upload the spreadsheet into GEORDi. The user also provides additional information such as the dataset name, description, publisher, type of template and a valid Web URL (all needed to make a entry for the data catalogue in GEORDi (discussed in the next section)). Additionally the Web URL is used to mint the URIs needed for the newly converted data. Overall, the responsibility of the user is finding which template is the most suitable and either author the data based on that template or make transformations to existing data to conform to that template. We thus alleviate the user from the burden of minting URIs or caring about any ontological models that need to be supplied.

Currently GEORDi supports three templates of spreadsheet data (inputed as CSV): simple spreadsheets, extended simple spreadsheets, and multidimensional data spreadsheets. A simple spreadsheet is a template requiring a spreadsheet to have a header row of which the first element is the type of thing the spreadsheet data talks about (e.g. schools) and all other elements are the properties (e.g. address, district, level). In RDF terms, the first item corresponds to the type or class of the instances in the spreadsheet, and all the other elements are properties of that type. Every other row corresponds to an RDF statement where the subject is the first element of the row and all other subsequent values are treated as literal values. In order to annotate specific values as resources rather than literals, the user can use the extended simple spreadsheet template, which requires an additional row above the header which contains values of "0" for which the user might want to find additional data. Finally, for complex statistical data we provide a multidimensional statistical template which translates the data to RDF using an extension of the SCOVO ontology [9]. The first column in the template represents the value of a `scovo:item` while all other columns are correspond to a subclass of a `scovo:dimension`. We are currently in the process of extending the number of templates to support more diverse forms of spreadsheets which will reduce the time users need to do transformations. Thus our current set of templates only servers as a demonstration.

Figure 1 illustrates a transformation of a portion of the statistical data transformed to conform with the multidimensional statistical data template that GEORDi can translate. It also shows the mappings between the template and the result using the SCOVO vocabulary.



	Number of Deaths	
Area of usual residence	Males	Females
Cherwell	550	560
Oxford	456	460
South Oxfordshire	546	530
Vale of White Horse	463	480
West Oxfordshire	395	486

Number of deaths	Gender	Area of Usual Residence	Period
550	Male	Cherwell	2009
560	Female	Cherwell	2009
456	Male	Oxford	2009
460	Female	Oxford	2009
546	Male	South Oxfordshire	2009
⋮	⋮	⋮	⋮

Figure 1: Converting statistical data from data.gov.uk to a GEORDi template.

2.2 Finding and browsing other linked data

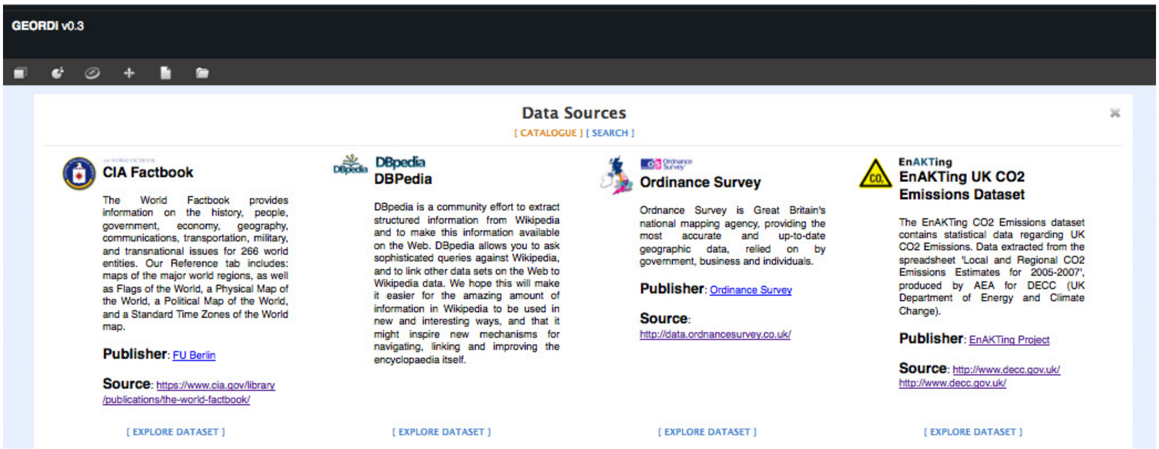
Once the data from the spreadsheet has been converted the user needs to find existing Linked Data sources to link the newly converted data. In order to this users need to be able to search for and discover potential data sources and if needed, interrogate or brows these data sources to see if the actual data inside is of any interest to them.

2.2.1 Discovering data

Data discovery over Linked Data is an active research topic. Most approaches to finding Linked Data over the Web are usually based on keyword search (e.g. VisiNav [7], Sig.ma [6]). These approaches however do not return actual data sources per say, but return actual instances of the data that match any of the given keywords. While these approaches might be useful for certain type of queries, they might not always provide accurate information to users searching for potential data sources. Consider a dataset of CO2 emissions for the UK. While all of the regions pertain to the UK, this information might not be explicitly supplied in the data, in which case a keyword search of *"UK CO2 emissions"* would not return any results or in the best case give the result a much lower ranking.

Recent initiatives such such as VoID[1], provide a way to describe datasets and other useful information e.g. links to other datasets. However the information they contain is tailored for machines rather than humans. Furthermore, while

[6] `http://sig.ma`



CIA Factbook

The World Factbook provides information on the history, people, government, economy, geography, communications, transportation, military, and transnational issues for 266 world entities. Our Reference tab includes: maps of the major world regions, as well as Flags of the World, a Physical Map of the World, a Political Map of the World, and a Standard Time Zones of the World map.

Publisher: FU Berlin

Source: https://www.cia.gov/library/publications/the-world-factbook/

DBPedia

DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data. We hope this will make it easier for the amazing amount of information in Wikipedia to be used in new and interesting ways, and that it might inspire new mechanisms for navigating, linking and improving the encyclopaedia itself.

Ordinance Survey

Ordnance Survey is Great Britain's national mapping agency, providing the most accurate and up-to-date geographic data, relied on by government, business and individuals.

Publisher: Ordinance Survey

Source: http://data.ordnancesurvey.co.uk/

EnAKTing

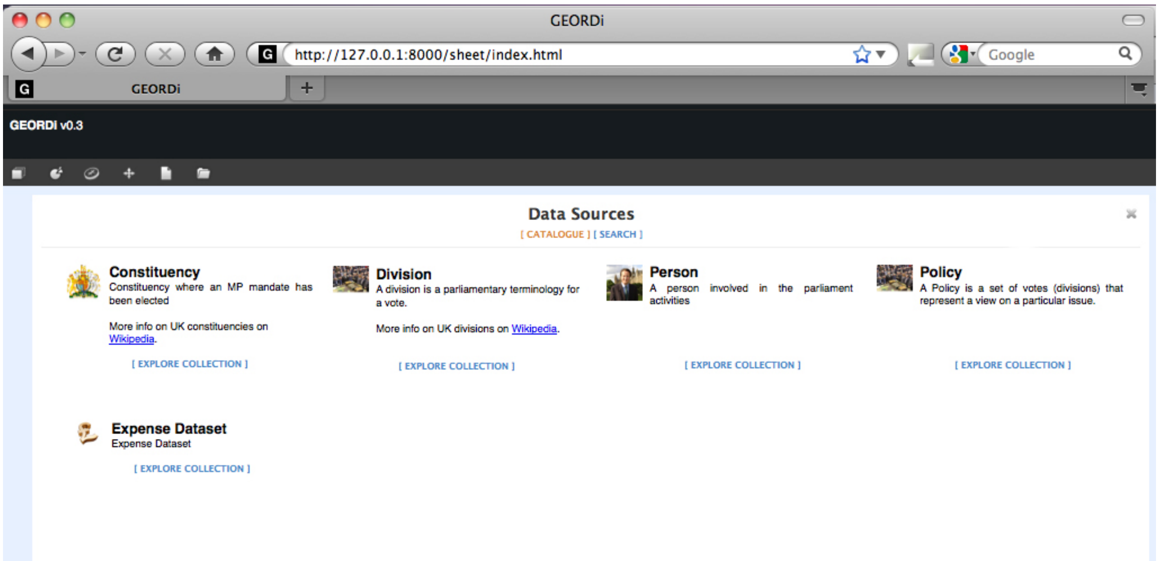
EnAKTing UK CO2 Emissions Dataset

The EnAKTing CO2 Emissions dataset contains statistical data regarding UK CO2 Emissions. Data extracted from the spreadsheet 'Local and Regional CO2 Emissions Estimates for 2005-2007', produced by AEA for DECC (UK Department of Energy and Climate Change).

Publisher: EnAKTing Project

Source: http://www.decc.gov.uk/ http://www.decc.gov.uk/

Figure 2: Starting exploration in GEORDi through a data catalogue showing the title, description, publisher and website of the data source.



Constituency
Constituency where an MP mandate has been elected

More info on UK constituencies on Wikipedia.

Division
A division is a parliamentary terminology for a vote.

More info on UK divisions on Wikipedia.

Person
A person involved in the parliament activities

Policy
A Policy is a set of votes (divisions) that represent a view on a particular issue.

Expense Dataset
Expense Dataset

Figure 3: Collections in GEORDi allow users to start exploring a chosen dataset from a particular set of items.

services such as VoID Store[7] and Talis VoID Store[8] provide a single stop place to search for linked datasets require the use of SPARQL to query an thus are not suited for end users.

In GEORDi users can discover potential data sources either through a catalogue of datasets that GEORDi knows about and can access or alternatively use keyword search.

Figure 2 depicts the catalogue in GEORDi. Each entry in the catalogue holds the name of the dataset, a short description of the data one might find in the dataset and some additional information, like the institution or person publishing the information and a Web site where the user can refer to for more information about the publisher. While the information is a few lines it serves as a preview or cue to users about what they might find in a particular dataset, so they can decide whether it is worth diving in a particular dataset to search for any useful data. Figure 2 shows several well know linked datasets such as DBPedia, CIA Factbook and more recent ones around public information sector data such as Ordinance Survey, and the datasets of data.gov.uk.

If users decides to explore a dataset, they simply can click the *"Explore Dataset"* button which yields *collections* of resources of a particular item. Collections serve as starting point from which the user can start exploring the actual data. Similar to the data catalogue each collection shows a brief description.

The catalogue and collection mode allow end users to browse through datasets whenever there search is of exploratory nature. GEORDi, however, allows users to also do a keyword search.

[7] `http://void.rkbexplorer.com/`
[8] `http://kwijibo.talis.com/voiD/`

Under the hood, each dataset actually corresponds to a single public SPARQL endpoint, and the collections in each dataset represent a predetermined set of types of classes from which the user can start exploring the graph. The catalogue is described in RDF which includes properties such as the description of the dataset, SPARQL endpoint URI and information about the publisher. The rich representation of these resources is achieved through the use of lenses which themselves are described in RDF (more on lenses check Section 2.2.3).

2.2.2 *Browsing through inter-linked data*

To enable end user browsing through graphs of RDF data, users require a data browsing tool that will both display RDF data and aid navigation through the graph.

Certain browsers such as IsaViz[9] and Fenfire [8] display RDF as a graph visualisation. Graph-based visualisations can be useful for showing certain aspects of data, e.g. showing density of connections between things in the data or relatively simple hierarchies. They, however, become unusable when displaying large amounts of data. Schrafel and Karger [14] have argued that graph visualizations of data do not offer any extra affordances for casual users but rather unnecessarily expose them the data model of the data they are exploring.

RDF browsers, on the other hand, allow users to browse RDF data by navigating through a graph of data displaying one resource at a time, rendering the RDF as a web page and displaying outgoing links from that resource which are used to navigate to the next resource. Example of such browsers are Disco[10], Marbles[11] and Zitgist[12]. The Tabulator [3] is a similar browser, but instead of showing one instance per page, Tabulator uses a nested tree metaphor to expose additional resources, thus allowing users to view their exploration trail in the same page. Additionally, Tabulator allows to select fields in explored areas and tabulate any results that contain the same pattern.

All of the above browser are, however, insufficient do explore large amounts of linked data, or unable to answer complicated queries. More powerful browsing techniques such as link-sliding (also referred to as *set-oriented* or *pivoting*) are needed to navigate through sets of Linked Data. Set-oriented browsing is a technique that allows refocusing a view on a particular set of items through navigation through a common property. In principle, set-oriented browsing is an generalization of the *one-to-one* browsing paradigm to a *many-to-many* browsing mode.

More recent implementations of data browsers use link-sliding as a way to navigate through a graph of data. Explorator [2] uses link-sliding as a metaphor for querying, however the user is required to select things such as subjects, objects and predicates as well as choose operations such as *unions* and *intersections* adding unnecessary load to casual point-and-click users. The Humboldt browser [13] provides a list of items and faceted filters from which the user can choose to *pivot* to the next set. Humboldt, however, shows only a trail of sucsessive linkslides, but no connections between the items of the refocused set and the set from which the linkslide occured. Parallax [10] shows the current items, a list of facets and a list of connections showing the available link-sliding operations. Additionally, Parallax provides a toolbar where users can trace their navigation choices and revert to a previous state. Every item which has been derived through link-sliding contains a header to specify what item it link-sled from. However after another link-slide operation you are unable to view the relationships between items in the first and third set.

In GEORDi we extend the link-sliding paradigm to provide maximum context for the entire trail of link-sliding operations. Once a user finds a dataset for exploration and choses a particular collection to start exploring from, selecting that particular collection instantiates a list showing all the items of that collection (Figure 4 (1)). The header of the list contains a property slider which if selected displays a collection of the properties about those items (Figure 4 (2)). This collection represents a *union* of all the properties of the items shown in that list. Selection of a particular property in the menu produces another column of items which is appended to the initial column. The items of the new column represent corresponding resources that which are linked from the items in the initial column with that property (Figure 4 (3,4)).

GEORDi allows the user to instantiate as many spreadsheets as they like, either by reopening the catalogue and select another dataset or collection. Additionally, users can create duplicates of the current spreadsheet and take then do link-sliding across different paths allowing them to see the results of both spreadsheets side-by-side.

As Figure 4 shows, the link-sliding paradigm in GEORDi is represented as an unfolding spreadsheet that the user generates through multiple link-slides. To have maximum clarity between the items in multiple column, we chose a nested table representation, so the height of a single item cell is equal to the maximum height of all the items which have been derived from that item through link-sliding. As with the initial column, the user has the ability to link-slide from any other column therefore slowly unpacking the graph by building up a custom spreadsheet. As mentioned previously, the users can view the entire context of their link-slides, thus allowing them view relationships between items beyond only successive link-slides. Additionally with this tabular view of the data the user is already using the familiar metaphor of a spreadsheet. In addition to link-sliding the user can filter results of any of the column as is shown in Figure 5.

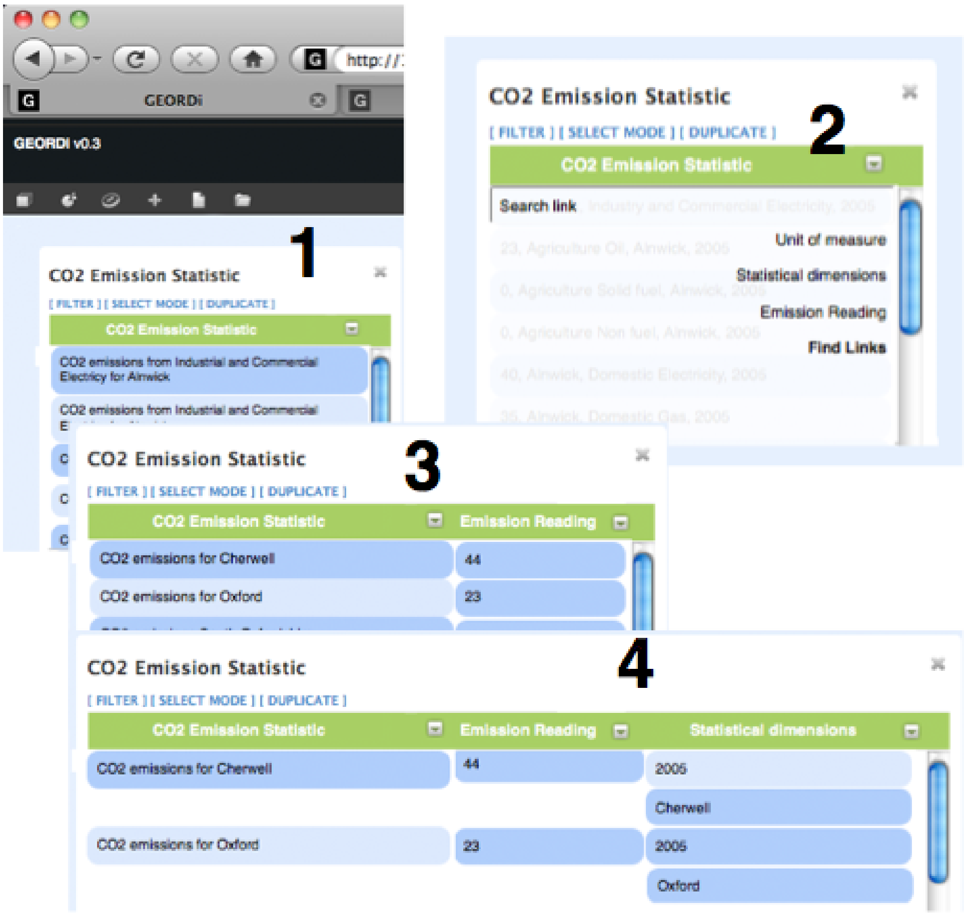
Coming back to our example, the user may choose to explore a dataset containing CO2 emissions to figure out if the geography is the same as in his mortality dataset therefore to examine whether links could be established. Figure 4 shows three link-sliding operations in order to explore a *"CO2 emission statistics"* collection from the CO2 emissions dataset (Figure 2). The once the user has instantiates a list of all the *CO2 emission statistics* they can link-slide to get the *Emission readings* and the *Statistical dimensions* for the

[9] http://www.w3.org/2001/11/IsaViz/

[10] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

[11] http://www5.wiwiss.fu-berlin.de/marbles/

[12] http://dataviewer.zitgist.com/


GEORDi
GEORDi v0.3
1
CO2 Emission Statistic
[FILTER] [SELECT MODE] [DUPLICATE]
CO2 Emission Statistic
CO2 emissions from Industrial and Commercial Electricty for Alnwick
CO2 emissions from Industrial and Commercial
2
CO2 Emission Statistic
[FILTER] [SELECT MODE] [DUPLICATE]
CO2 Emission Statistic
Search link
Unit of measure
Statistical dimensions
Emission Reading
Find Links
3
CO2 Emission Statistic
[FILTER] [SELECT MODE] [DUPLICATE]
CO2 Emission Statistic
Emission Reading
CO2 emissions for Cherwell
44
CO2 emissions for Oxford
23
4
CO2 Emission Statistic
[FILTER] [SELECT MODE] [DUPLICATE]

CO2 Emission Statistic	Emission Reading	Statistical dimensions
CO2 emissions for Cherwell	44	2005 Cherwell
CO2 emissions for Oxford	23	2005 Oxford

Figure 4: A example of a link-sliding operation in GEORDi. Users can open up new columns from any existing column.

statistics.

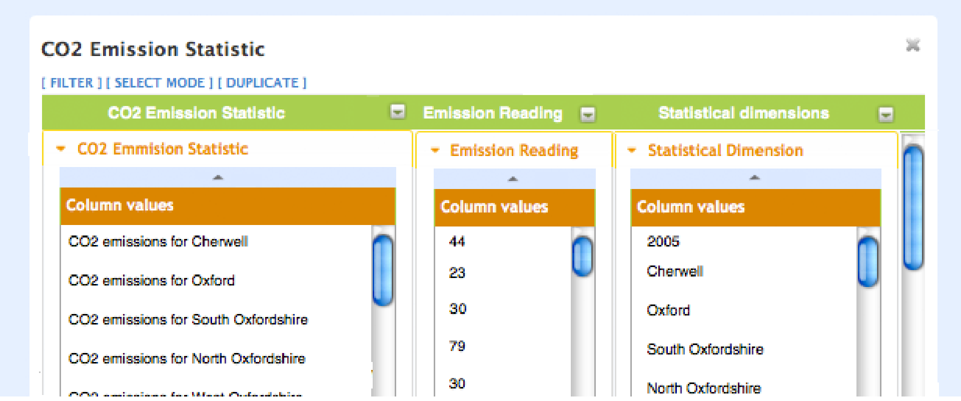
2.2.3 *Lenses*

In order to provide richer renderings of individual resources, there needs to be additional information which prescribes which portions of the graph in relation to the resource are show to the user, and how they should be displayed. Existing approaches such as Fresnel [15] allow publisher to describe which properties to display and how to display them. Most generic data browsers, however, do not employ advanced views of RDF but rather try to render RDF directly.

In GEORDi lenses are used for more than just prescribing how a individual resource is displayed. Selection of properties to be shown are done through *selectors* which in a GEORDi lens specify which properties should be displayed to the user using either simplified expressions specified as directory paths that are translated into SPARQL queries. For more expressive power one could use full SPARQL queries instead. Additionally, one can specify some common operations to the results of the selector query (e.g. summation, average etc.) Apart from specifying the view of a resource these annotation severe other purposes. First they directly impact the properties that will be shown as properties in the property selector (e.g. if rdf:type is not displayed in the lens it will not show up). Additionally a column might have multiple filters depending on the properties shown in the lens.

GEORDi as a platform does not mandate data publisher to supply lenses in order to display data, however if the publisher (or any other contributor) does supply them, this means that the user experience with that particular data would be better for end users. Lens in GEORDi can be authored either for individual items or for items of a particular type. Thus when GEORDi tries to display an individual item it first searches to see lenses are defined for that individual item. If it does not find it checks to see if there are any lenses for the type of which that item is. In case no lenses are found, GEORDi attempts to find the label (`rdfs:label`) to display to the user. If no labels are found GEORDi displays the URI. By having such mechanism GEORDi and can provide publishers with an instant view of their data and point out any mistakes, missing fields or where the data might not be understandable for browsing by end users.

As can be seen for Figures 2, and 3 the catalogue and collection resources all have lenses specified for them. On the other hand the items in spreadsheet in Figure 4 show only labels. This type of selection allows publishers to fully customise the way they want to display data to the users.



CO2 Emission Statistic

CO2 Emission Statistic	Emission Reading	Statistical dimensions
CO2 Emmision Statistic	Emission Reading	Statistical Dimension
Column values	Column values	Column values
CO2 emissions for Cherwell CO2 emissions for Oxford CO2 emissions for South Oxfordshire CO2 emissions for North Oxfordshire CO2 emissions for West Oxfordshire	44 23 30 79 30	2005 Cherwell Oxford South Oxfordshire North Oxfordshire

Figure 5: Filtering in GEORDi. The filters allow users to filter through the unique values of each of the columns.

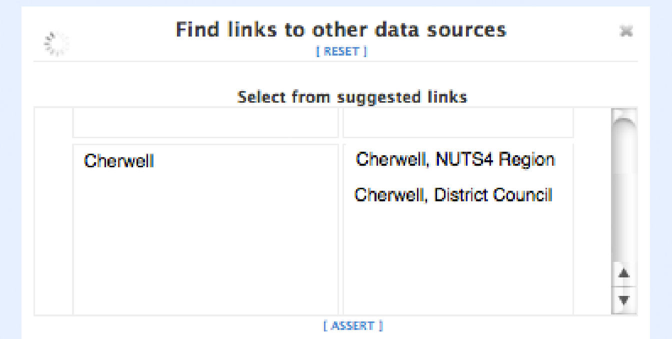
2.2.4 *Linking data*

In order for users to mingle converted data form spreadsheets with existing Linked Data they need to establish links between the two pieces of data. To do that, first potential data to link to must be found, second once found the next thing is finding the corresponding items to be linked and finally asserting those links so that they can be used.

Currently most link-finding tools are unusable for end users. Tools like Silk [4] require authoring similarity metrics to search for possible links between datasets. On the other hand users might resort to applications such as Sig.ma[13] or sameAs.org[14] however, this again exposes them to the world of URIs and RDF concepts. Furthermore, they will need to take the raw data and would have to do manual linking directly inside the RDF. At this stage all of these task are impossible for end users to do.

GEORDi contains a simple widget that allows users to select a running spreadsheet, and search for links between items of a particular column to other Linked Data on the Web (Figure 6). The user can either choose to search for equivalent concepts form a particular dataset or search the Linked Data as a whole to establish relationships with any datasource that returns any results. The user then needs to simply select things that they consider are equivalent.

In our example, the user can open up his converted data (found in the catalogue) and do the same process of link-sliding as described in Section 2.2.2. Once the columns are shown they can open the association widget, select the column (in this case *Statistical dimensions*) and select the CO2 dataset that was previously browsed to search for equivalent regions. The user then selects one of several available choices and simply makes the assertions by clicking the *Assert* button. This causes the spreadsheet with mortality data to refresh and a new property *"Find more in CO2 dataset"* is instantly available in the *Statistical dimensions* column for the user to use.



Find links to other data sources

Select from suggested links

Cherwell	Cherwell, NUTS4 Region Cherwell, District Council

Figure 6: The association tool in GEORDi allows end-users to find and confirm links between two data sources.

Currently GEORDi is limited to finding and establishing only equivalence links using the `owl:sameAs` property. Part of the future work in GEORDi is to generalise this capability to include other relationships.

2.2.5 *Visualising data and querying multiple datasources*

Once the data is linked, the user has the ability to traverse through the newly established links. After the user unpacks the data through iterations of link-sliding and filtering the user can then also do a number of visualizations.

GEORDi also supports visualising data through a number of widgets. Currently we have a chart visualisation, as well as a map visualisation which in addition to mapping geographic coordinates it also has a backend service that can search for areas given a URI as well as perform geographical reasoning (more in the Section 3). Parallax and Tabulator both have a number of widget for data visualizations, including charts and maps. None of them however can display regions nor can they reason over geographical containments.

In our example, the user can now use the established links to get the CO2 emissions about the regions for which they previously had only mortality data. As Figure 6 show data form both datasets appear in the same spreadsheet which was again generated through link-sliding. Then using the

[13] `http://Sig.ma`
[14] `http://sameAs.org`

geographic widget they can display regions that can be found from a column on a map. In order to display meaningful statistic the user can adjust the map widget to display the intensity of the colour based on numerical values in another column. Figure 6 shows two maps instantiated form the spreadsheet which show mortality and CO2 emissions for the same geographic region.

3. IMPLEMENTATION

GEORDi has been designed as an open platform for data interrogation (Figure 8). As such it has been implemented as a set of interacting widgets that use open protocols and online services to link together, render and discover data. GEORDi can be categorised in terms of the front-end user interface and the back-end data processing. In order to query the back end we use a model of lightweight state encapsulation, whereby the state of the user's view is represented as a JSON structure by the front end, and used by the back end to generate SPARQL queries to knowledge bases.

3.1 Back end

In order to query semantic and linked data, we utilise two key strategies, depending on the size and use of the data. Firstly, if data is large and hosted on a SPARQL endpoint that is fast enough to handle interactive queries, we query it directly using SPARQL. If the data is smaller, we import it into a new server-side knowledge base on a per-user basis. Using per-user ”buckets” affords us the ability to enable users to query across sources, while our browser needs only to query one knowledge base, at the cost of hosting that data for them. In order to enable large-scale ad-hoc data hosting, we use a multiple-server 4store cluster.

As mentioned above, one of the enabling technologies to enable communication between our back end and front end is the use of the state model, which encapsulates all of the elements of the uses's current state. Specifically it holds the links between the columns, and the state of the user's scroll, which determines what the SPARQL queries to run will be, and what the ranges of the LIMIT and OFFSET will be.

Our back-end also uses sameas.org and Sindice to process links between different data sets, by querying literals, and gathering matching datasets when users linkslide using the ”owl:sameAs” links.

3.2 Front end

As noted above, the state model is the method by which out front and back ends communicate. Thus, one of the key features of the front end is to allow the user to linkslide across predicates, and to encode that action in a state model, send that state model to the back end, and render the results.

The front end also employs a ”lazy loading” system, whereby only the first 100 entities in a column are loaded; additional entities are then loaded when the user scrolls to the bottom of the list. This behaviour continues as the user scrolls further. We enable this behaviour by encoding the state of the list in the GEORDi state model. One problem that frequently occurs when dealing with open data is that different data sets are collected using different geographic containments. For example, UK crime data is collected per police force region, CO2 emissions per NUTS4 area, and hospital waiting times per NHS Trust region. This makes comparing these overlapping and non-tessalating areas difficult. Thus, we use the enAKTing geographic containment service [5] to determine common containments for different region types, and use those containments as a source of comparison and for linking across datasets that use different regions.

We also enable users to visualise data in ways that are appropriate to their data source, for example using map widgets for geographic data, specifically a map widget built on the geograhic containment system that shows the polygonal region boundaries for the UK and Europe, (in addition to usual point-based latitude and longitude support). We also enable users to pick any data they have built up in a single sheet, and plot them on charts in order to visually compare and contrast statistical data.

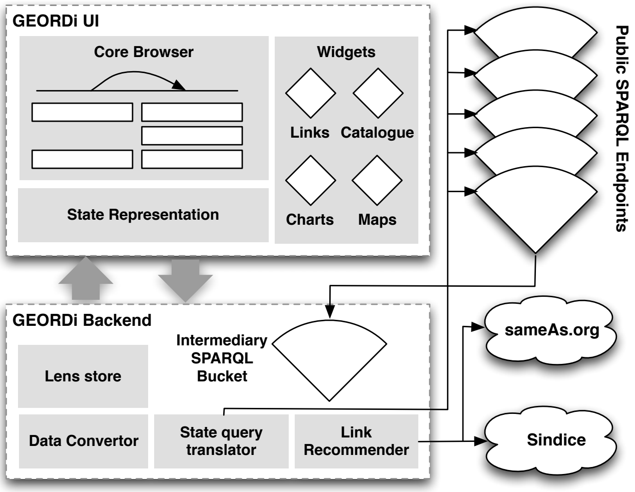
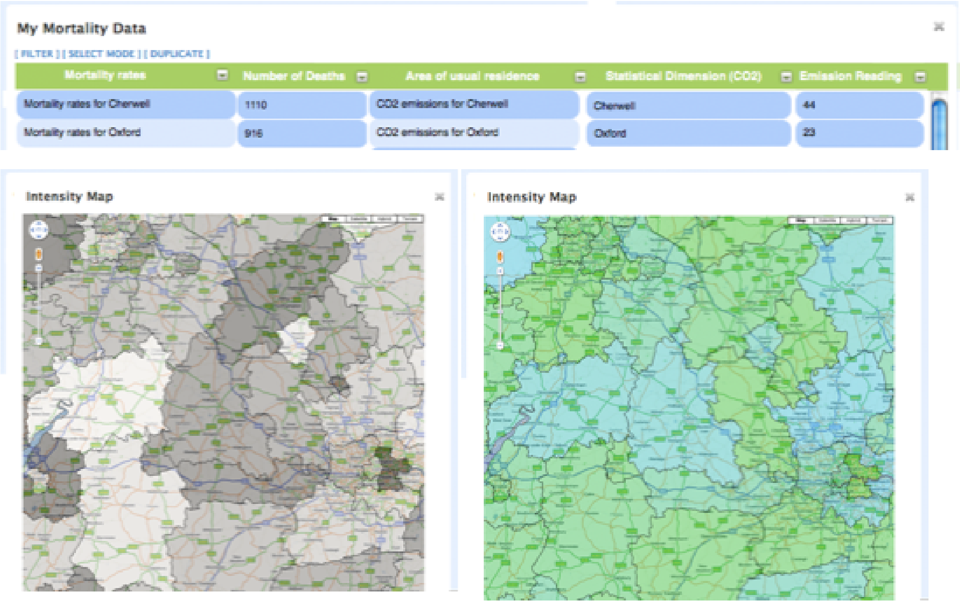


Figure 8: A high level view of the architecture of GEORDi showing services on both front end and back end.

4. DISCUSSION

Our aim in developing GEORDi is to investigate several components of making open/linked data accessible for citizens to use. First we sought to look at how we might make it as citizen friendly as possible to add data to the web of data so that it could be easily explored, but especially easily connected with other linkable web data sources for enriched querying. Second, we wanted to make linking up related data sources as simple as saying ”these two are connected.” Third, in making such linking possible, in particular we wanted to see if we could break the closed-box, one off, Web2.0 mash up approach to open/linked data embodied in USAspending so that any citizen might be able to produce on-demand queries and representations across arbitrary data sets about as effortlessly as it is simply to use a web2.0 mash up now. In other words, we have wanted to explore the cost and feasibility of taking the ”must be a geek” out of the requirements to make use of the current web of data, and make it as tractable as the current web of documents.



My Mortality Data

Mortality rates	Number of Deaths	Area of usual residence	Statistical Dimension (CO2)	Emission Reading
Mortality rates for Cherwell	1110	CO2 emissions for Cherwell	Cherwell	44
Mortality rates for Oxford	916	CO2 emissions for Oxford	Oxford	23

Figure 7: Showing two maps displaying different data about the same geographic regions.

Of both interest and necessity, to support the dynamic interlinking of data described above, we have used mainly semantic web/linked data protocols. Given the requirements to support interactions that will require no semantic web protocol expertise, the challenges of the project have afforded a considerable opportunity to discover some of the key limitations of semantic technologies, particularly when it comes to developing UIs that allow end user to engage and interact with the Web of Data. Our main technical problem has been one of scalability. While we are maximally using techniques such as *"lazy loading"* scalability problems do exist. To satisfy real time dynamic querying of data, we are performing live SPARQL queries to servers over the Web. The approach suffers from both the usual network delay problems as well as the reliability of the SPARQL endpoint.

SPARQL Optimization. Another problem has been the limitations of SPARQL itself. Data generated for each column of the link sliding UI, for instance, takes numerous SPARQL queries in order to return results. For a particular column we need to query for the URIs of the resources to be displayed in that column, after which we need to run a separate query for each one in order to get the properties and property values. This is due to the limitation of SPARQL which in it's current version does not allow subqueries – something we understand will be resolved in the upcoming version. Additionally if lenses are specified, a number of queries are executed to retrieve the properties needed to display that resource. Filters also need to be populated on demand. We are looking into ways of enriching existing RDF data as well as putting in place different caches in order to mitigate the need to run such high number of queries.

Usability Refinements. Finally, we have developed services to support a particular set of citizen requirements: enable interaction across the web of data without requiring semantic web protocol expertise to do queries across arbitrary heterogeneous linked data sources. Much work now remains to be done to tune the interaction of the interfaces themselves to ensure their clarity and effectiveness. For the most part we see that standard usability methods can be applied here for such design refinement.

5. FUTURE WORK

GEORDi is still in a early prototype and we are in the process of making the framework robust at which point we plan to make a large-scale public deployment. Several extensions to the system are also planned which we briefly discuss.

5.1 Exporting data as Linked Data

Currently GEORDi stores converted data in a internal triple store creating a graph for any data that has been uploaded. Thus while the user can link and consume data from other Linked Data sources the converted data is never exposed as Linked Data to the outside world i.e. the URIs are not derefrencable and the data can only be used inside the GEORDi framework. This limits other user to refer to the converted data. Part of our future work is to allow users to export this data to a Web server of their choice and expose this data as Linked Data so that other users can contribute and access it in a way they see fit.

5.2 Doing it all in GEORDi

We are encouraged to see more end user authoring in GEORDi which will include annotations as well as shaping data inside the user interface as well as contributing things such as lenses. To do this however requires certain policies to be in place that will guide how and where users can contribute. Additionally, with added functionalities our spreadsheets in GEORDi can

5.3 Fostering collaboration and crowd-sourcing

Currently GEORDi acts as a platform for single-end users but our future plans is to make it as much a collaborative space as is personal space for data. This includes people having a way of sharing their mashups and uploaded data with other people.

Additionally, we are intrigued by the idea of using crowdsourcing to speed up processes such as end user linking. Currently in GEORDi linking can be done by a single user which is fine for relatively small datasets. For linking far bigger datasets opening this process so other people can contribute is a possibility. We are also thinking of other possibilities such as using Amazon so people can recruit workers to do these manual tasks that require human intervention.

6. CONCLUSION

Perhaps the most significant contribution of GEORDi at this stage in its evolution is as a demonstration that we can take the "must be at least a hacker" for a person to do something useful with arbitrary data sources for dynamic queries. No other approaches to our knowledge have designed each stage of the create, discover, explore, integrate, query, represent, re-query, re-represent cycles for citizens/non-programmers to make use of the web of data as a web of data.

Within this demonstration as a whole, we have also presented several tools/services, some of which , like the geography service, are already in use in the linked data community. Likewise these tools offer solutions for known real problems for adding value to data web. The conversion tool beyond its citizen-friendly approach to converting data to rdf for reuse is simply fast: it is an expedient tool to generate RDF data whether it is to be used in GEORDi or elsewhere. The link-sliding approach in GEORDi itself facilitates rapid exploration and filtering of data sources. The linking service enables data sets to be connected and then again explored further. The catalogue and lens services facilitate easier discovery of data for multiple reasons, not the least of which is being able to search and then carry out one click loading of a data set. The geography service tackles a critical problem of helping to mesh geographical regions that are specified by different criteria.

There is still considerable work to be done to create a robust tool for citizens to explore the data web. GEORDi however offers at least a first step towards establishing that it is possible to address linked data challenges and deliver publishing, exploration and querying tools that citizens rather than semantic web geeks only can use to generate new knowledge.

7. ACKNOWLEDGMENTS

This work was supported by the EnAKTing project, funded by EPSRC project number EI/G008493/1.

8. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009.

[2] S. Araujo, D. Schwabe, and S. Barbosa. Experimenting with explorator: a direct manipulation generic rdf browser and querying tool. In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009)*, February 2009.

[3] 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 Procedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06*, page 06, 2006.

[4] C. Bizer, J. Volz, G. Kobilarov, and M. Gaedke. Silk - a link discovery framework for the web of data. In *18th International World Wide Web Conference*, April 2009.

[5] G. Correndo, M. Salvadores, Y. Yang, N. Gibbins, and N. Shadbolt. Geographical service: a compass for the web of data. In *Linked Data on the Web (LDOW2010)*, April 2010.

[6] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi. Rdf123: From spreadsheets to rdf. In *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 451–466. Springer Berlin / Heidelberg, 2008.

[7] A. Harth. VisiNav: Visual Web Data Search and Navigation. In *Database and Expert Systems Applications*, pages 214–228. Springer, 2009.

[8] T. Hastrup, R. Cyganiak, and U. Bojars. Browsing linked data with fenfire. In *Linked Data on the Web (LDOW2008)*, 2008.

[9] M. Hausenblas, W. Halb, Y. Raimond, L. Feigenbaum, and D. Ayers. Scovo: Using statistics on the web of data. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. P. B. Simperl, editors, *ESWC*, volume 5554 of *Lecture Notes in Computer Science*, pages 708–722. Springer, 2009.

[10] D. F. Huynh and D. R. Karger. Parallax and companion: Set-based browsing for the data web. In *WWW Conference*. ACM, 2009.

[11] 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.

[12] D. R. Karger, S. Ostler, and R. Lee. The web page as a wysiwyg end-user customizable database-backed information management application. In *UIST '09*, pages 257–260, New York, NY, USA, 2009. ACM.

[13] G. Kobilarov and I. Dickinson. Humboldt: Exploring linked data. 2008.

[14] m.c. schraefel and D. Karger. The pathetic fallacy of rdf. In *International Workshop on the Semantic Web and User Interaction (SWUI) 2006*, 2006.

[15] 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.