Undefined 1 (2010) 1–2
IOS Press

Approaches to Visualising Linked Data: A Survey

Aba-Sah Dadzie [a,*] and Matthew Rowe [b]
[a] *OAK Group, Department of Computer Science, Regent Court, University of Sheffield, Sheffield, United Kingdom*
E-mail: a.dadzie@dcs.shef.ac.uk
[b] *Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom*
E-mail: m.c.rowe@open.ac.uk

Abstract. The uptake and consumption of Linked Data is currently restricted almost entirely to the Semantic Web community. While the utility of Linked Data to non-tech savvy web users is evident, the lack of technical knowledge and an understanding of the intricacies of the semantic technology stack limit such users in their ability to interpret and make use of the Web of Data. A key solution in overcoming this hurdle is to visualise Linked Data in a coherent and legible manner, allowing non-domain and non-technical audiences also to obtain a good understanding of its structure, and therefore implicitly compose queries, identify links between resources and intuitively discover new pieces of information. In this paper we describe key requirements which the visualisation of linked data must fulfil in order to lower the technical barrier and make the Web of Data accessible for all. We provide an extensive survey of current efforts in the Semantic Web community with respect to our requirements, and identify the potential for visual support to lead to more effective, intuitive interaction of the end user with Linked Data. We conclude with the conclusions drawn from our survey and analysis, and present proposals for advancing current Linked Data visualisation efforts.

Keywords: Linked Data, Information Visualisation, Visual Analytics, User-centred Design, Users, Consumption

1. Introduction

The Web of Linked Data provides a large, distributed and interlinked network of information fragments contained within disparate datasets and provided by unique data publishers. The interest in this Web of Data has led to a *data race* where organisations – both non-commercial and commercial – have begun publishing their data in a format which is machine-readable – i.e., in RDF (Resource Description Framework[1]) – and linking this data to other external data. The utility of such linkage has enabled, for the first time, complex queries to be answered and traversals to be made through a diverse and semantically rich information network.

Such is the popularity of the Web of Linked Data, that it has now grown to a staggering size, with 38.5 billion triples now residing in the central connected cloud[2]. Making sense of such data presents a huge challenge to the research community, a challenge which is compounded further by the drive to produce data from both government and public bodies for end-user consumption [27,35]. It is evident that the ubiquity of mobile of devices and the reduction in cost to produce sensors will escalate these challenges further, as more and more data is published through a variety of means, e.g., information about events, foot fall in buildings and city centres.

The size and scale of the Web of Data presents challenges when trying to make sense of information contained within it. A basic visualisation of the Web of Data focussing on a resource which has a high out-degree of relationships to other data will present the viewer with a mass of edges linking into the resource,

*Corresponding author. E-mail: a.dadzie@dcs.shef.ac.uk
[1] `http://www.w3.org/RDF`
[2] `http://www4.wiwiss.fu-berlin.de/lodcloud`

resulting in information overload. How does an end user *make sense* of the response? How do they understand and interpret the data in a meaningful way?

The third principle of linked data states that "*When someone looks up a URI, provide useful information, using the standards*" [6]; therefore when a URI (Uniform Resource Identifier) is dereferenced the response is represented using RDF together with a given serialisation format (e.g., n3[3], Turtle[4], XML[5]). Knowledge of how to use this format and interpret information provided using it is restricted to tech-savvy end users, and even in certain cases, only those who have knowledge of Semantic Web (SW) technologies. It is clear that regular web users, so called *lay users*, who have no knowledge of RDF, nor ontologies, are inhibited in their ability to understand data returned when looking up a URI ([31], among others).

Automatic production of linked data and its interweaving into the Web of Data is regularly done at a large scale. One of the central issues with large-scale linked data production is the accuracy and completeness of links with other datasets. Identifying such links using the solitary RDF format of a dataset limits the reader's ability to notice any errors and incorrect links. Visualisation of linked data enables the identification of such errors more easily, using, for instance, a graph visualisation. This will show clearly links between resources where those links should not exist, and allow the viewer to identify instances where links *should* exist but are omitted.

Clear and coherent visualisation of linked data is essential if the Web of Data is to be used outside of the SW community. This therefore requires linked data to become *usable* by *lay users*, by providing interfaces and browsers of the Web of Data to support sense making and information exploration and discovery. Furthermore, query composition in languages such as SPARQL[6], although useful, requires understanding of a given query language's syntax and at least a basic knowledge of data content and structure. End users should be able to implicitly compose such queries without being aware of the underlying query mechanism that is used to pose the required questions.

In this paper we present a survey of existing approaches currently used to visualise linked data. To enable a consistent analysis of each approach we identify challenges to visualising linked data and the requirements which visualisation approaches must fulfil. By analysing such approaches, our survey provides, for the first time, a comprehensive discussion of linked data browsers and the inherent limitations of such tools.

We have structured this paper as follows: section 2 describes the background and context in which this survey is set. It describes the current state of the Web of Data and the end users which we expect to utilise this information network. Section 3 defines the requirements which approaches to visualising linked data must fulfil. Section 4 describes current approaches to browsing and visualising linked data, where each approach is analysed based on the requirements laid out in section 3. Section 5 discusses the findings from our survey, what functions were common across current linked data browsing approaches, and what limitations were also consistent. Section 6 presents the conclusions we draw from this survey.

2. Background

2.1. A Basic Scenario: the LOD Cloud

The scale of the Web of Data and sheer mass of information now available hinder the ability of end users to make sense of the data. To illustrate the challenge involved, let us stop for a moment and take a high-level look at the Linking Open Data (LOD) cloud, and the popular diagram used to visualise it [9], generated from metadata extracted from the Comprehensive Knowledge archive Network[7] (CKAN). The September 2010 version, shown in Fig. 1, contains 203 datasets, with 25 billion RDF triples and 395 million links. Fig. 1b overlays colour coding on the base graph in Fig. 1a. Both graphs use relative node size and edge thickness to represent the number of triples in each dataset and the links to other connected datasets respectively. The graph is centred on the *DBPedia*[8] [7] knowledge base created by extracting structured infor-

[3] http://www.w3.org/DesignIssues/Notation3.html

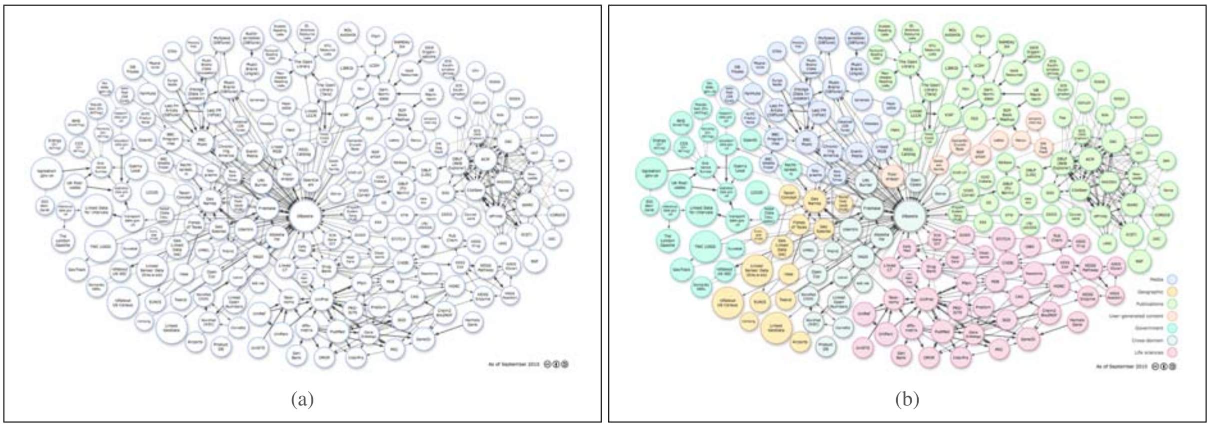
[4] http://www.w3.org/2010/01/Turtle

[5] http://www.w3.org/XML

[6] SPARQL query language for RDF: http://www.w3.org/TR/rdf-sparql-query/

[7] http://ckan.net

[8] http://wiki.dbpedia.org



(a) (b)

Fig. 1. The LOD cloud diagram; the graph on the right uses colour coding to distinguish different themes in the dataset.

mation from the online encyclopaedia Wikipedia[9], as part of the LOD effort.

An alternative visualisation of the CKAN dataset by Ed Summers[10] is shown in Fig. 2, using the *Protovis* API[11]. Colour is used here to encode CKAN rating. To reduce occlusion only those nodes corresponding to more than 5 million triples are labeled (hovering over a node will reveal hidden labels). The visualisation shows a central cluster, surrounded by six main clusters of varying size. Five outliers can be seen at the edges of the plot – these represent nodes for which links to others in the central cluster are not found, information about the data structure which is hidden by the very dense cloud in Fig. 1.

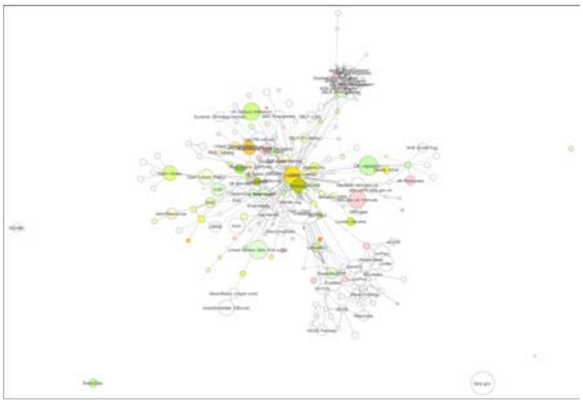


Fig. 2. The LOD cloud visualised using *Protovis*. (Credit – Ed Summers[10], update Oct 2010)

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

[10]`http://inkdroid.org/lod-graph`;
See also: `http://inkdroid.org/journal`

[11]`http://vis.stanford.edu/protovis`

Rinke Hoekstra[12] generates an alternative view on the LOD cloud (see Fig. 3) using the *Gephi Open Graph Viz Platform* [2]. Colour and relative node size are used to encode data attributes, as is done in Figs. 1b and 2). This visualisation also displays a central cluster, with DBPedia as the focus. Link length is also used to encode information about the data structure; two smaller clusters are drawn at a significant physical (and semantic) distance from the centre.

2.2. *Challenges to Consuming Linked Data*

(Node-link) graph representations such as those in section 2.1 are a technique often employed for visualising inherently hierarchical and inter-linked data. The density of the underlying data is obvious, even with clustering of related data points. These high-level examples highlight one of the challenges in visualising linked data, as ever increasing amounts of complex data are added to the Web of Data.

Additional challenges exist: links on the Web of Data are often constructed using an automated approach. Due to their huge size false links are often created which are hard to identify – this is reflected in Hogan et al. [16] who found that applying basic reasoning over null valued inverse functional properties for people in a dataset containing 54,836 valid RDF documents and 12,534,481 statements would infer 972,000 false `owl:sameAs` links. Manual browsing seldom reveals such links as they are buried away in the data. Effective, meaningful visualisation of the graph structure is necessary if these and other errors

[12]`http://twitpic.com/17qj1h`;
See also `http://www.leibnizcenter.org/~hoekstra`

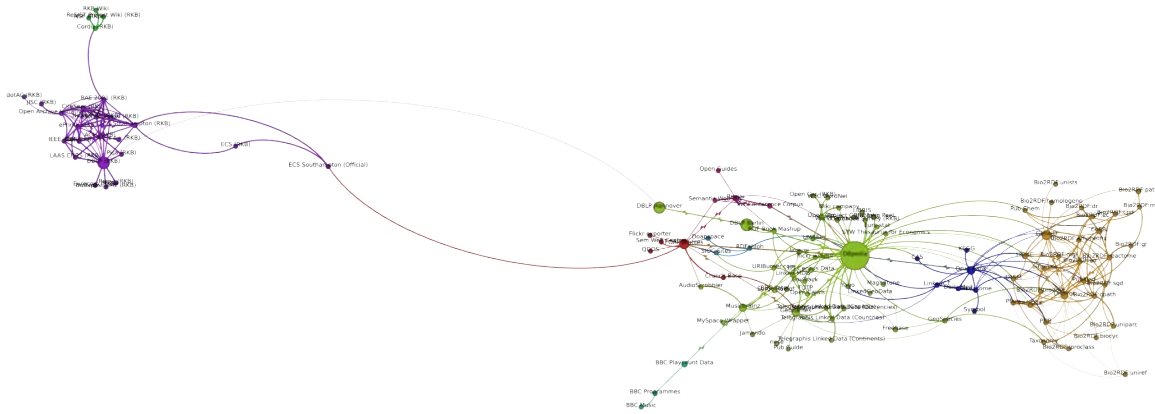


Fig. 3. Visual analysis of the LOD cloud using Gephi (Credit – Rinke Hoekstra[12], posted Mar 2010)

and noise in linked data are to be revealed [26]. The outliers in Fig. 2, for instance, highlight the existence of "broken links" in the LOD cloud - these may be due to missing or incorrect information in the underlying data set. Alternatively they may be legitimate islands or anomalies in the LOD. Identifying such structures through manual browsing of the text is very difficult at best.

Because LOD is by definition a community effort, data heterogeneity, beyond differences in data type and content, presents yet another challenge. While encoding the actual data in RDF imposes a degree of uniformity, this does not necessarily translate to homogeneity in the quality of data capture. Fidelity and reliability of data cannot always be guaranteed or verified, especially where data provenance is not clearly specified. Differences in data type, content and granularity require a degree of translation in order to link distinct data sets correctly. A significant challenge is where data representing the same or similar concepts is encoded using different language or terminology [1,8,27,35,37]. SW technology provides a simple solution to this, if standard ontologies are used to translate and encode data, such as encoding e-mail addresses using `foaf:mbox` rather than a custom label. Where new concepts must be defined, extending existing standards will generate a link to previously validated information, increasing the ability to reuse data.

2.2.1. *Defining End Users*

The overall aims of linked data require it to be usable by both tech-savvy and mainstream end users. A number of European Union (EU) governments (e.g., *data.gov.uk*[13], *Open Government Data Austria*[14]), the United States (US) government (*Data.gov*[15]), public bodies and the media (e.g., the British Broadcasting Corporation's (BBC) *Wildlife Finder*[16] and *Music* pages[17] [24], the Guardian's *World Government data store*[18], and Link TV's *ViewChange.org*[19]) are using LOD as a means for making distributed information publicly available ([27], among others). A significant challenge is found here: the (formalised) encoding of linked data limits its use to those who can read and interpret RDF in the raw – a format created for machine consumption [13,29,31]. Questions posed by mainstream users would include "What is RDF?", "What is it about?", "What can I do with it?" In this paper we define two types of users, which the visualisation of linked data must function for:

Lay-users (mainstream) users who do not understand the intricacies of RDF and other SW technologies. Such users are computer literate and are able to find information through online resources such as Wikipedia or search engines. Examples of their information seeking tasks include studying nationwide house price indices prior to buying a new home, comparing albums while buying chart music.

Tech-users expert users who understand SW and other advanced technologies, have experience in using

[13] `http://data.gov.uk`
[14] `http://gov.opendata.at`
[15] `http://www.data.gov`
[16] `http://www.bbc.co.uk/wildlifefinder`
[17] `http://www.bbc.co.uk/music/beta`
[18] `http://www.guardian.co.uk/world-government-data`
[19] `http://www.viewchange.org`

RDF as a data format, and are able to interpret an ontological model.

The sheer volume of data created on a daily basis due to advances in technology [15,21] means that even tech-users are severely limited in their ability to obtain a good understanding of the structure and knowledge content of these very large datasets. Another challenge is identifying suitable methods for presenting this data to all potential end users – both tech-users and lay-users – so that they are able to obtain a good overview of its content and retrieve the knowledge contained with its context of use. Equally important is support for intuitive navigation through the data, in order to exploit what is touted to be one of the most valuable assets of LOD – the links between diverse, heterogeneous, distributed data – to discover new, valuable and relevant information by extending existing knowledge to other related data [13].

Berners-Lee et al. [4], Karger and schraefel [20] further posit that SW data, of which LOD is a subset, enables information discovery from the user's perspective, when presented in a format suitable for the user and their task, whether this is text or visualisation-based. For example, while a software developer may need raw data in RDF or JSON[20] (JavaScript Object Notation) to feed into applications that analyse selected attributes of a data set, a formatted table is more useful to a policy analyst, and a visual representation to the lay-user browsing the results of the analysis of public health data in their community, for instance. Section 2.2.2 looks at why linked data presents advantages over traditional methods for collecting, encoding and sharing data. In order to justify the effort required to develop more effective means for consuming linked data we must first look at the merits the Web of Data provides over other existing large-scale data sets.

2.2.2. *Why Linked Data? A Public Data Consumption Scenario*

Data.gov is an example of a repository that provides access to heterogeneous data distributed over several areas of interest, with an aim to "empower" the US population. Data.gov currently contains more than 6.4 billion RDF triples, as part of the process of converting existing public data from traditional formats such as CSV (comma separated values) to a SW representation[21]. The value of linked data is expressed very simply by Ryan McKeel of the US National Renewable Energy Laboratory, in comments about a *mashup*[22] of two different energy data sets[23]:

> " *. . . you can create new stories from datasets that were completely independent but they were accessible – two databases coming together in a way that wasn't expected. . . . What's so cool about linked open data is that you can pull from all these different repositories. It's a common format that you can use to easily create stories. . . .* "
> *"This mashup could be expanded by anybody in the public. Anybody could take the source code and add another city or add other data sources. There's no licensing fee because it's open data and it's free software.*"

This is in line with the aims of the Linked Data community[24], who state:

> " *Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods.*"

Wikipedia[25] further states:

> " *. . . the term Linked Data is used to describe a method of exposing, sharing, and connecting data via dereferenceable URIs on the Web.*"

We illustrate the value of linked data to both tech- and lay-users with a subset of Data.gov that reports community health indicators visualised by county and state in the US[26], by linking to cartographic data collected by the US Census Bureau[27]. A demonstration of a set of simple methods for consuming linked data can be seen in a website created by Joe Meyer[28]. This site provides a good example of the differences in require-

[20] http://www.json.org

[21] See Data.gov - Semantic Web: http://www.data.gov/semantic/index

[22] http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)

[23] See 'Data.gov Mashathon 2010: an Energy Data Mashup': http://www.data.gov/semantic/energy-data-mashup#mashup

[24] http://linkeddata.org

[25] See: http://en.wikipedia.org/wiki/Linked_Data

[26] See: United States Department of Health and Human Services: http://www.hhs.gov; Community Health Status Indicators Report: http://www.communityhealth.hhs.gov/homepage.aspx

[27] See: Descriptions and Metadata – U.S. Census Bureau: (Cartographic Boundary Files) http://www.census.gov/geo/www/cob/metadata.html

[28] http://health.jameyer.com/health.py/home

ments of different types of end users in the consumption of linked data. To allow developers to extract the underlying data for use in other web-based applications, e.g., a JSON front end is provided. We use this to extract a sub-set of the data (see Fig. 4) with the properties:

```
county,
percent age 19 to 64,
life expectancy,
percent smokers,
percent high blood pressure,
percent obese,
percent no exercise,
@state='New York'
```

Two visualisations options: *scatter plots* (Fig. 5) and *blot maps* (Fig. 6), are provided for browsing the data, in addition to (a text representation in) a sortable table (Fig. 7).

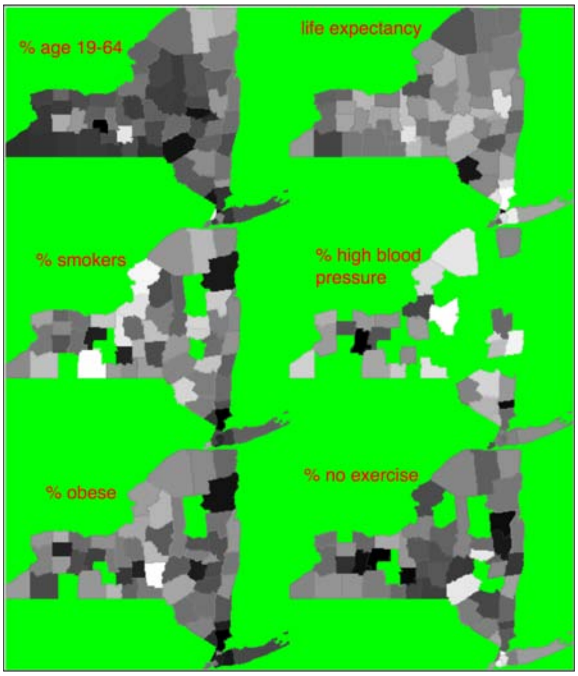


Fig. 6. Blot maps used to visualise data distribution, by county, of age and five health indicators (see Fig. 4) for New York state.

Each representation is suitable for different tasks and different end users. Linking the health data to the cartographic data set allows visualisation on a map (Fig. 6), highlighting data distribution from the perspective of geographical location. This application displays only one additional attribute per map (compared

county	percent age 19 to 64	life expectancy	percent smokers	percent high blood pressure	percent obese	percent no exercise
Bronx	59.1	75.0	19.6	24.5	24.4	35.5
Sullivan	61.9	75.4	30.9	24.5	22.6	25.2
Cattaraugus	60.2	76.2	28.9		19.9	26.9
Oswego	63.0	76.5	32.1	20.3	26.6	25.7
St. Lawrence	62.8	76.5	25.2	32.1	25.0	25.8
Greene	62.2	76.5	29.5		20.3	
Kings	60.5	76.7	18.7	23.6	23.0	34.6
Niagara	60.8	76.7	26.5	27.8	25.4	24.4
Orleans	62.4	76.8	21.3		27.3	
Rensselaer	62.7	76.8	25.1	33.2	23.0	20.4
Erie	60.4	76.9	22.7	26.3	23.9	23.9
Chenango	61.2	77.0	19.3		34.0	22.5
Clinton	65.4	77.0	24.4	24.8	22.0	26.6
Richmond	63.0	77.0	27.3	20.9	22.6	28.3
Franklin	66.0	77.0	28.3		24.5	23.0
Cortland	62.8	77.1	27.8		17.3	22.1
Allegany	60.4	77.1				26.8
Steuben	60.5	77.1	35.1	27.1	25.5	25.2
Yates	58.1	77.1				
Chemung	60.4	77.1	25.9	29.8	27.9	23.8
Schuyler	61.3	77.1				
Hamilton	61.6	77.2				

Fig. 7. Sorted by life expectancy, the data set for selected health indicators in Fig. 4 is displayed in a table.

against county); the use of colour, saturation and/or patterns, and height in 2.*D (dimensions) stacked layouts, are options available for overlaying multiple attributes on a single map. The scatter plot view allows focus on two attributes at a time. While this is useful for directly comparing two related properties it is still difficult to compare multiple attributes simultaneously. We align the plots to allow the five indicators to be compared on life expectancy (see Fig. 5). An interactive visualisation technique that supports multi-attribute visual analysis is *parallel co-ordinates* [18] such as illustrated in [19], which are sometimes seen as an extension of a multi-attribute scatter plot. Jern et al. [19], as is done here, use multiple linked views, allowing the end user to visualise the same data from multiple perspectives, highlighting different aspects of the data and allowing a more complete overview and understanding of its content.

The table (Fig. 7) provides a quick way to inspect the underlying data. Options for sorting allow simple statistical calculations such as minimum, maximum and median values to be estimated. For the lay-user who wishes to obtain an overview of the health statistics in their community this is a simple method for obtaining hard facts for selected regions of interest (ROIs), in addition to the more general picture enabled by the visual representations of the overall data set. Importing the table into a spreadsheet or statistical package for more detailed analysis is easily done.

While this is a very simple set of visual and text-based representations of a specific data set, the poten-

```
json_callback(
{
"pyql" : "county,percent age 19 to 64,life expectancy,percent smokers,percent high blood pressure,percent
obese,percent no exercise@state='New York'",
"headers": ['county', 'percent age 19 to 64', 'life expectancy', 'percent smokers', 'percent high blood
 pressure', 'percent obese', 'percent no exercise'],
"results" : [
["Erie","Cortland","Columbia","Dutchess","Delaware","Chenango","Chautauqua","Cayuga","Oswego","Putnam",
"Allegany","Bronx","Broome","Cattaraugus","Albany","Steuben","Sullivan","Tioga","Westchester","Kings", ...
 ... "Livingston","Warren","Ulster","Wayne","Greene","Essex","Franklin","Washington","Fulton","Genesee"],
[60.400,60.4,62.800,62.8,61.300,61.3,62.900,62.9,58.900,58.9,61.200,61.2,60.100,60.1,62.100,62.1, ... ,60.6],
[76.900,76.9,77.100,77.1,77.400,77.4,78.000,78.0,77.900,77.9,77.000,77.0,77.700,77.7,78.400,78.4, ... ,77.8],
[22.700,22.7,27.800,27.8,20.800,20.8,21.700,21.7,23.800,23.8,19.300,19.3,24.300,24.3,32.700,32.7, ... ,24.2],
[26.300,26.3,null,null,26.000,26.0,null,null,30.400,30.4,null,20.300,20.3,16.800,16.8,null,24.500, ... ,21.3],
[23.900,23.9,17.300,17.3,25.000,25.0,20.800,20.8,26.600,26.6,34.000,34.0,25.800,25.8,23.600,23.6, ... ,17.3],
[23.900,23.9,22.100,22.1,27.800,27.8,23.600,23.6,33.500,33.5,22.500,22.5,25.800,25.8,27.100,27.1, ... ,26.7]
]});
```


Fig. 4. An extract of a dataset from Data.gov to JSON, to examine selected health indicators in New York state

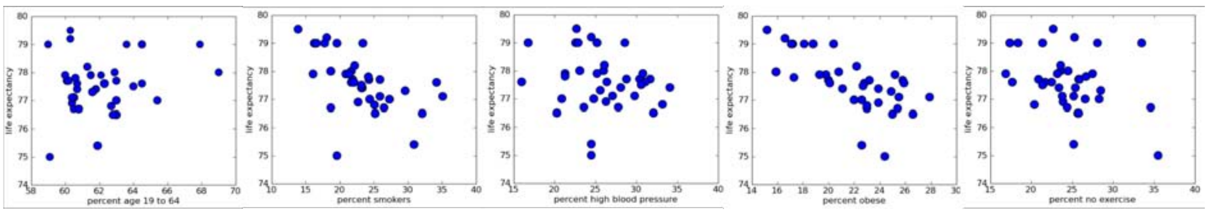


Fig. 5. The scatter plot view allows any two data attributes to be compared directly. One plot for each of the other five health indicators is compared against life expectancy for the data set in Fig. 4.

tial to extend these to more powerful analysis can be seen. The common representation for linked data, using a method that supports the capture also of metadata describing the data itself (as for the US Census Bureau data in the example in Fig. 4) provides not just rich data, but simplifies reuse in ways not easily achieved for data encoded in traditional formats.

2.3. *Summary*

Having made the case both for the value of linked data as a means of encoding and sharing distributed data and the need to develop effective, user- and task-oriented systems for consuming this data, we must identify suitable techniques for achieving this aim. The LD initiative is still in its infancy (see the design document written in 2006 by Berners-Lee [6]); applications to date for browsing linked data and inspecting its structure and content are therefore, not surprisingly, predominantly targeted at tech-users [26]. Considering the requirements of such users the majority of these are text-based, often simply displaying the raw RDF data. Where available descriptive labels in the RDF (e.g., from `rdfs:label` or `rdfs:comment`) may be used to provide a more human-friendly representation of the underlying data. For tech-users such representations are useful, and often preferred [4,20]. Being domain experts, stripping away the additional layers necessary for lay-users to interpret and consume effectively the complex, interlinked data, conversely helps tech-users to make optimal use of its content.

Where visualisation is available in such tools they are often no-frills, node-link graphs that strictly follow the structure of the corresponding RDF graph. For especially large data sets obtaining a good mental model of the data places a large cognitive burden on even technical users and domain experts [26,38]. Such visual representations are often adequate for the tech-user [20,28], whose focus is on inspecting the structure of a known domain and/or data set, to validate data that is manually or automatically generated, checking for and correcting errors introduced during its generation and/or conversion [26], and/or to feed data into specialised applications for further analysis [28]. Information retrieval and filtering tasks, for such users, are typically carried out using formal query syntax, which also requires knowledge of the underlying data structure.

However, linked data, encoded in RDF, albeit being self-describing, with its bent toward machine interpretation, restricts consumption by lay-users, whether this is simply browsing the data to obtain an overview, deeper exploration toward knowledge discovery, delving into selected ROIs to retrieve detail or carry out analysis toward a specific end goal, or publishing new data. Large scale and high inter-linking within and across linked data sets further increases the complexity of the underlying data. Anecdotal evidence and research indicate that visual representations of complex, heterogeneous, multi-dimensional, linked data should harness advanced human perception to increase intuition and capability while reducing users' cognitive load in carrying out such knowledge-intensive activity [10,11,21,22]. With effective design to meet the requirements of end users and their overall goals and simpler sub-tasks (see, among others, [32,36]), visualisation of linked data has the potential to augment human analytic and information retrieval ability [1].

A large body of work exists on the value visualisation brings to data exploration, analysis and the presentation of the results of analysis and the structure of the raw data itself to different audiences. Keim [21], Keller and Tergan [22], for instance, argues that visualisation increases human capability for data exploration and analysis, both when supported by automated means and when the latter fail. He found that this resulted in increased confidence in the results of analysis and the conclusions drawn from them. Shneiderman [36] recognises the cognitive challenges encountered in the exploration of large amounts of data, and the potential of information visualisation to support associated tasks, especially with advances in technology. Among the advantages in information visualisation is its ability to provide a spatial representation for abstract data, aiding the recognition of patterns within data, and highlighting related information as well as outliers, by relying on advanced human perceptual ability, both during directed analysis and especially in data exploration and knowledge discovery [10,22,32,36].

Making linked data more accessible, through the use of intuitive visualisation solutions, among others, should promote mainstream uptake and demonstrate large-scale utility. This would in turn encourage the generation of new data such that the links to existing data are highlighted and maintained, resulting in an increasingly richer, more useful and usable web of linked data.

3. Requirements for Visualisation

Section 2.2.2 illustrates a not atypical case for the consumption of linked data in the public domain. Meyer's application links two distinct data sets in order to reveal the distribution of data for health indicators of interest to the end user using simple but informative views. User skills and preferences often influence tool adoption and use; providing multiple options and perspectives on the data, as Meyer does, increases the usability of the overall application, as different tasks are best carried out with different tools. Further, this highlights different aspects of the data, allowing the end user to compare the output of each view. A simple method for querying the data is provided, in order to update each view based on the user's current focus. A drawback here, however, is the need to know the data attributes available to query on, dependent on prior knowledge of the data content or structure.

Such applications satisfy a sub-set of the requirements for the consumption of linked data. In addition to these often very specific solutions there is the need also for more generic browsers, that are able to take as input linked data of any type and support at least basic information retrieval, filtering and analysis tasks. These may then be supplemented with specialised tools for more advanced analysis and knowledge discovery.

Having made the case for the value of visual solutions for consuming linked data, we examine next the requirements for the design of such tools, to overcome the challenges identified for tech- and lay-users. Visualisation tools are most effective when designed to suit particular tasks and users [4,31]; this explains the myriad tools available even within the same application domain. However, a set of basic design features is common to the visualisation of large amounts of complex data. Further, an advantage in the common format used to encode linked data is that SW-based applications only need to be able to read RDF data as a starting point for tool development.

Working from the challenges we identified for effective consumption of linked data in section 2.2) we derive a list of requirements for the design of visualisation tools that should help to overcome these. We list first high level requirements for advanced user interfaces (UIs) for visualisation tools (highlighting in bold key requirements where first specified), as defined in [36]:

1. (a) the ability to generate an **overview** of the underlying data;

(b) support for **filtering** out less important data in order to focus on selected ROIs;
(c) support for visualising the **detail in ROIs**.

In addition to these Shneiderman [36] further defines different tasks that information visualisation is often used to support, and the requirements for each of these. We include here those relevant to the data and tasks carried out in the consumption of linked data, namely, support for:

2. (a) handling multi-dimensional data;
   (b) hierarchical data (or tree structures) – linked data is often generated as instances of (rooted, hierarchically structured) ontologies;
   (c) network data (graph structures) – high interlinking within linked data may result in a graph rather than a tree; networks may not define a root and may be acyclic;
   (d) identifying **relationships** within data;
   (e) data extraction (to a format that allows reuse by third parties (users and/or applications).

We define, also at a high level, specific requirements for tech-users for consuming linked data (we highlight commonality across sets of requirements with an asterisk):

3. (a) * intuitive **navigation** through linked data structures;
   (b) * data **exploration**, to obtain an understanding of its structure and content;
   (c) * data exploration, to identify the links within and across data sets;
   (d) data exploration, to identify errors, noise and other anomalies in content and syntax;
   (e) validation of both manually and automatically generated linked data sets;
   (f) * **advanced querying**, using formal query syntax, in order to obtain flexible, effective information retrieval
   (g) * **syndication**;
   (h) * data extraction for reuse in other applications (without the need for a change in format for SW applications – see point 2e).

Finally, general requirements for lay-users also result in more effective visualisation design for even tech-users:

4. (a) * intuitive navigation through the large amounts of complex, multi-dimensional data;
   (b) (exploratory) knowledge discovery;
   (c) * support for basic to advanced querying, to support filtering and information retrieval. This must accommodate both the end user who may have little to no knowledge of formal query syntax and more tech-savvy end users; however the lay-user is unlikely to have advanced knowledge of underlying data content or structure (see point 3f);
   (d) detailed analysis of ROIs;
   (e) * syndication;
   (f) * data extraction (to a format that allows reuse by third parties – see point 3h);
   (g) **presentation** of the results of analysis to different audiences.

We will use these requirements to guide the survey of existing applications for linked data (in section 4), in order to allow a clear, comparative and consistent analysis of the techniques currently in use for visualising and exploring linked data, and the advantages and limitations of each. We will also use the requirements as a benchmark in the discussion of our findings (see section 5) that follows.

3.1. Data 'Visualisation' vs. 'Presentation'

It is necessary to distinguish between the different methods of presentation of linked data generally lumped under the label of (data) *visualisation*. Text-based options are commonly used, including list and tabular aggregation and presentation of data, such as found in *Sig.ma*[29] [37] and *Disco*[30] respectively. *Piggy Bank*[31] [17] combines faceted browsing with a textual presentation. Visual representations include pictorial, map and graph-based representations such as in *DBPedia Mobile* [3], which employs *Fresnel lenses* [29] via the *Marbles Linked Data Engine*. The *OpenLink Data Explorer*[32] provides multiple options for browsing linked data, including both text- and visualisation-based presentations. The UK Parliament checker[33] plots information about and relevant to serving Members of Parliament (MPs) in a selected geographical region on a timeline overview, and provides detail for each MP using table and list views [27]. Additional information on the geographical region of interest, e.g.,

[29] http://sig.ma
[30] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco
[31] http://simile.mit.edu/wiki/Piggy_Bank
[32] http://lod.openlinksw.com/ode
[33] http://psiusecase.enakting.org

crime and mortality rates, is displayed using bar charts; the underlying datasets are linked via their common property – geographical location.

We aim to carry out a survey to review existing support for visualisation of linked data, and identify work still necessary to support effective, intuitive visual exploration and analysis for the different end users of this data. In order to do so we must define the term '*visualisation*'. *Infovis.org*[34], the original repository of the *IEEE Information Visualization conferences*[35], defines '***visualisation***' as:

> " *...a process of transforming information into a visual form enabling the viewer to observe, browse, make sense, and understand the information. It typically employs computers to process the information and computer screens to view it using methods of interactive graphics, imaging, and visual design. It relies on the visual system to perceive and process the information.*"

This is in line with the definition of '***visualise***' in the online Oxford English Dictionary[36]:

> *"To form a mental vision, image, or picture of (something not visible or present to the sight, or of an abstraction); to make visible to the mind or imagination.*"

This survey focusses on the requirements for interactive visualisation of linked data that will lead to intuitive knowledge discovery for all users, both technical and mainstream. As discussed in section 2.3 the field has to date focussed on tools to support tech-users, whose tasks are often best served by text-based representations. As a result these represent a disproportionate percentage of existing LD browsers; we will therefore also review some of the more well-known text-based tools.

For the purposes of this survey we will refer to any application that provides interactive support for navigating through or exploring linked data as an LD *browser*. We will distinguish those that use a graphical, pictorial or image representation as LD *visualisations* or *visual browsers*, for static snapshots and interactive applications for exploring linked data respectively. As a measure of usability we will look at the ability of each representation to support the construction of a good cognitive model of the underlying data and the relationships within the data, and support the end user in carrying out their knowledge-intensive tasks.

[34]`http://infovis.org`

[35]`http://ieeexplore.ieee.org/servlet/opac?punumber=1000370`

[36]`http://dictionary.oed.com`

4. Linked Data Browsers

We group browsers into those that provide only text-based functionality (section 4.1) and those that include visual analysis and/or presentation options (section 4.2). We will also discuss any limitations, by comparing them against the requirements we list for the visualisation of linked data (in section 3).

The BBC's Semantic Music Project[37] has led to the development of *BBC Music Beta*[17], as part of an initiative to identify and link semantically related information owned by the BBC, to increase reusability and improve information retrieval, both within the corporation and by its audience and third party developers [24]. The project also aims to link BBC's legacy information to other related resources in the Web of Data. In order to compare the use of different LD browsers, we attempt to resolve and retrieve information related to the URIs that point to the output of this project (note than when dereferenced the former is redirected to the latter):

```
http://www.bbc.co.uk/music/beta
http://www.bbc.co.uk/music
```

For those browsers that do not support browsing RDF data as unbounded sources (on the Web) or that are unable to fetch the resources pointed to by the test URIs we examine use based on the results of the closest example provided by the application developers to the topic of interest.

4.1. Text-Based Presentation

We present first oft-cited LD browsers that use a textual representation of SW resources and the relationships between them. This includes tools that use presentation templates that resolve literals as part of a more human-readable, text-based layout, to display images, e.g., in a FOAF profile (see, for instance, the result of employing templates in the *URI Burner* – Fig. 11a in section 4.1.6).

[37]`http://www.readwriteweb.com/archives/bbcs_semantic_music_project.php`

4.1.1. Dipper

Dipper provides a public entry point to retrieve and browse LD resources from a set of repositories. Where available (in the RDF data) human-readable labels are used to format the results (retrieved as URIs). Fig. 8 shows the output of a request for a sample resource in the *musicbrainz* store.

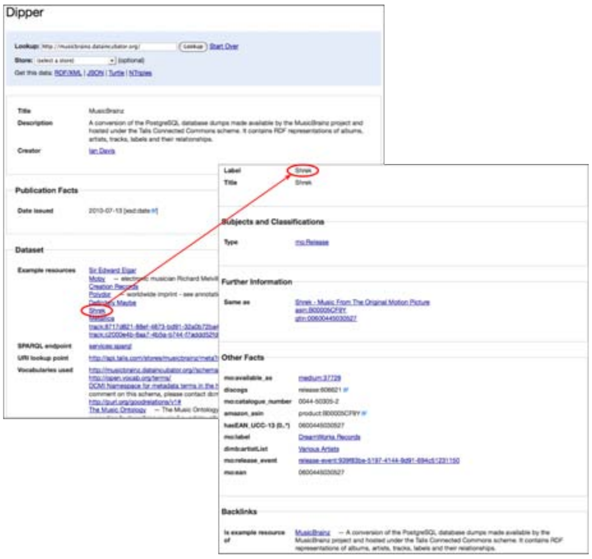


Fig. 8. The results of a request for a sample resource in *musicbrainz* is used as an entry point to browsing albums in the store. Labels in the RDF data are used with a basic text categorisation template to help provide more human-readable output.

4.1.2. Disco

The *Disco - Hyperdata Browser* was developed as a web browser-independent tool that browses unbounded RDF resources. Disco searches for information about a resource of interest on the Semantic Web. All processing occurs on the server, and results are presented in the client browser. The output is displayed as a table of property-value pairs, specifying the source of each resource returned. As for Dipper, where available, human-readable labels are used to format the results. The simple interface aims to support the debugging of SW sites.

The Disco public entry point was not retrievable as at the time of carrying out the review.

4.1.3. Marbles

Marbles retrieves information about resources of interest by querying the SW index, Sindice[38] (which is also used for Sig.ma), the SW search engine, Falcons[39], and the SW-based review site, Revyu[40] [14]. Marbles improves the user experience by performing (resource-intensive) processing on the server. *Fresnel lenses* are used to format SW resources for presentation to end users, presented as property-value pairs in a table. Different coloured "marbles" are used to distinguish the sources of the information retrieved, which are presented as a list of URIs. Fig. 9 illustrates the output of a request for the BBC Music Beta URI, from which the user browses to other interesting resources.

Marbles is also provided as a SPARQL endpoint. It may be installed locally, allowing information to be saved to and retrieved from the local data stores.

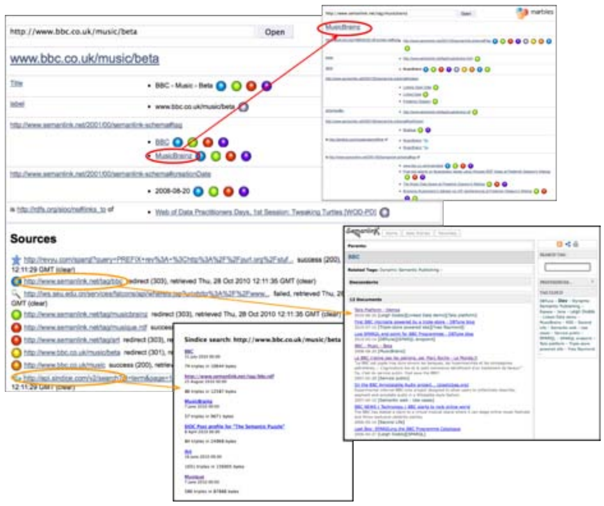


Fig. 9. The results of a request for the URI `http://www.bbc.co.uk/music/beta` are shown on the left. The user browses to the resource *MusicBrainz*, shown top, right. Two sources of information on the resource of interest (*BBC Music Beta*) are also retrieved – from *Sindice* and *Semanlink*[41] – see bottom, right.

4.1.4. Piggy Bank

Piggy Bank, part of the *SIMILE*[42] project, is a Firefox[43] web browser plug-in that enables the extraction (and conversion where required) of information from web sites as RDF to a semantic repository [17]. Users may tag resources extracted and share the information they collect with other Piggy Bank users. This allows

---

[38] `http://sindice.com`

[39] `http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp`

[40] `http://revyu.com`

[42] `http://simile.mit.edu`

[43] `http://www.mozilla-europe.org/en/firefox`

faceted search and browse across (the heterogeneous) information collected by multiple users from different data sources, based on user tags and other data properties such as date, document or resource topic. Results are presented using a list-like interface, and detail for information of interest using a table structure; Huynh et al. [17] reported that presenting the information as a collection of items was more suitable for the information seeking tasks they support than would a graph representation.

Providing the service via a web browser allows Piggy Bank to be used as part of end users' normal information retrieval activity. The local (and shared) semantic stores associated with Piggy Bank allow users' past actions to be recorded and reused, e.g., as suggestions while creating new tags.

4.1.5. Sig.ma

Sig.ma – **S**emantic **I**nformation **MA***shup* – views the "...Web of Data as an integrated information space" [37]. Sig.ma retrieves and integrates (often heterogeneous) linked data, starting from a single URI, by querying the Web of Data and applying machine learning and rules to the data found. Results are presented as a reorderable list of verified sources and links to potentially relevant information on the query subject; end users may confirm or reject relevance. Attributes of validated resources may also be used to filter the result set. To allow reasonable response time resource retrieval is staggered; repeat requests for additional information may be made – these will also take any filters in place into account. Support for basic editing, using a simple form, of the presentation templates is provided. Fig. 10 shows a short browsing chain which starts with a search for the phrase "BBC Music". Support is also provided for advanced querying to selected attributes of a resource.

4.1.6. URI Burner

OpenLink's[44] *URI Burner*[45] service retrieves information about resources of interest by (internally) traversing the RDF graph for a resource, by matching to standard ontologies and other relevant knowledge on the Web about the resource. The results are presented as a list of property-value pairs. Dereferenceable URIs that point to images and web pages are embedded into the HTML pages used to serve the output, in addition to displaying human-readable descriptions stored in RDF data.

[44]http://www.openlinksw.com
[45]http://linkeddata.uriburner.com

Fig. 10. The *Sig.ma* LD browser is shown top, left (a), with query subject "BBC Music". The results of dereferencing URIs for potentially relevant information (in the right-hand pane) are shown top, right (b), and lower, left (d). Bottom, right (c), shows the results of a query for "BBC Music + most played artists", which corresponds to the dereferenced URI in (b).

URI Burner provides support for requests input as free text strings[46] and for looking up URIs from a text label (see Fig.11b), in addition to the norm, querying from an absolute URI (illustrated in Fig.11a).

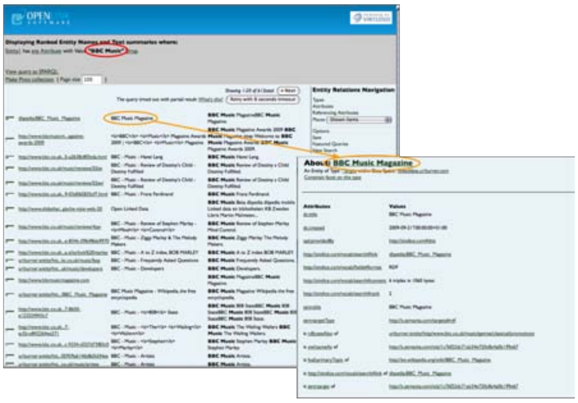
A SPARQL query endpoint is available. *OpenLink ISPARQL*[47], a graphical web-based UI that demonstrates query by example, is provided as a separate tool. Additional support is also provided for application developers to make advanced use of services for extracting linked data to raw RDF (as XML and other versions including n3 and Turtle) and other formats including HTML and JSON.

4.1.7. Zitgist DataViewer

The *Zitgist DataViewer*[48] was developed to help end users manage information overload. Zitgist collates information relevant to an entity of interest and presents it to the user (as a list of property-value pairs). Large result sets are split across multiple pages, improving response time and helping to manage information load. A set of templates enable adaptation of data presentation to suit the underlying data type and presentation medium (e.g., web browser or small screen). These include *FOAF*[49] (Friend of a Friend), *DOAP*[50] (Description Of A Project), and MO, the *Music On-*

[46]'URI Burner' Precision Search & Find: http://linkeddata.uriburner.com/fct
[47]http://linkeddata.uriburner.com/isparql
[48]http://dataviewer.zitgist.com
[49]http://www.foaf-project.org
[50]http://trac.usefulinc.com/doap

(a) Browsing from the resource URI for *BBC Music Beta*. On the right-hand side a template is used to display the result that includes the dereferenced URI embedded in the output.



(b) Browsing the Web of data from the text query "BBC Music"; URIBurner employs *Precision Find* to find the closest matching URI and returns a ranked result set as a table of entity names with corresponding URIs.

Fig. 11. Browsing the Web of Data in *URI Burner*

tology[51]. This allows information about a person, a project or a music album, e.g., to be displayed in a format that is more easily interpreted by lay-users. Support is provided to edit the layout of results, by selectively embedding the resources pointed to by dereferenceable URIs in the HTML page served. Simple filtering of sources retrieved (and displayed as a list in a navigator pane) is also available. Filtering based on data attributes also provides a simple form of faceted browse/filter.

A mini viewer allows the use of Zitgist on mobile devices[52].

4.1.8. Summary

Our review found that all but one of the text-based browsers reviewed target the tech-user. As previously discussed this is not unexpected at this stage in the lifetime of the LOD initiative.

None of the tools reviewed provides a high level overview. However, support for detailed analysis of the current focus is provided to at least some degree in all the browsers. Marbles additionally provides very limited highlighting of links across data, by adopting a visual paradigm – the coloured "marbles" used to distinguish data types in the view. This is to be expected: text does not support very well the generation of overviews,

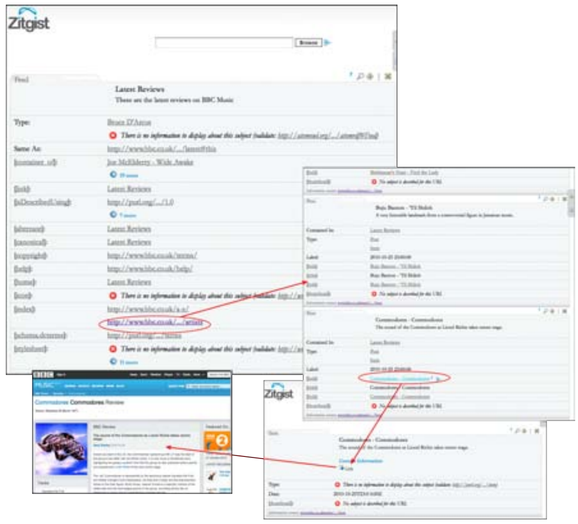


Fig. 12. The results of a request for the URI `http://www.bbc.co.uk/music/beta` is shown in the *Zitgist DataViewer*. Additional snapshots show browsing the resources retrieved to retrieve a list of artist reviews and link finally to the *Commmodores* page on the *BBC Music Beta* website.

nor the identification of relationships within data on scale dealt with in this field.

Some level of formatting and categorisation provides more human-readable representations of the RDF data output. Varying levels of (reusable) template support, such as enabled by Fresnel, result in more fine-tuned presentation of results; targeting the requirements of tech-users however means that this is fairly basic for the generic LD browsers. Their use for

[51] `http://musicontology.com`

[52] Zitgist mini-DataViewer test on iPhone emulator: `http://www.testiphone.com/?url=http://minidv.zitgist.com`

[53] `http://zlinks.zitgist.com`

Fig. 13. A query from `http://www.bbc.co.uk/music` in the *Zitgist mini-DataViewer* compares the browsing experience to that in a web browser (see Fig. 12). This time we follow the path to the review of the solo artist *Joe McElderry* and extract additional information about the entities retrieved using the *zLinks*[53] tool. This allows the MO template to be selected to format the information for the data of type `musicartist` (sub-figures b & c).

detailed analysis therefore still presents a challenge for the lay-user.

Genericity dominates, as expected, for the tools developed to browse the unbounded Web; with the only restriction due to the need to load and specify data repositories to search on in Dipper. Some level of support for querying via SPARQL endpoints is available, in addition to the default input via URI request. The former is the most effective way to retrieve data for reuse by third parties; for tech-users this meets expectation. Support for lay users is much lower, with only two of the browsers providing functionality for text search.

Tech-users are likely to generate large scale data automatically (as in e.g., [30]), with little need for explicit support from LD browsers, beyond error identification, noise reduction and validation. Very little support is provided for the publication and validation of new data, with only Piggy Bank, which targets lay-users, supporting intuitive user-centred publication of linked data to shared repositories.

The review found that the tech-user is fairly well supported by the text-based tools that enable simple, unbounded stepwise browsing from specified ROIs through the LOD cloud and the readable Web, with access to powerful, flexible functionality for formal querying across data sets. A significant gap however exists in support for the lay-user, especially for exploratory browsing and knowledge discovery. We examine next a selection of visualisation-based tools, both developed for linked data and generic RDF browsers able to handle linked data with little or not customisation, to determine the level of support available for both tech- and lay-users.

4.2. Browsers with Visualisation Options

The visualisation-based approaches to exploring linked data we review include custom applications developed specifically for linked data, such as *DBPedia Mobile*, *LESS*, *RelFinder* and *Tabulator*, and browsers developed for RDF data able o support also the particular requirements of linked data, such as *IsaViz* and *RDF Gravity*. The discussion will highlight the application to complex information spaces (i.e., densely linked graph data), and the potential for more intuitive support for visual information seeking and knowledge discovery tasks.

4.2.1. DBPedia Mobile

DBPedia Mobile uses geographical location as the connector between data points of interest, in order to support end users navigating through the real world, actively seeking or exploring resources near their physical location, to discover information about objects and events [3,7]. DBPedia Mobile uses GPS to identify the user's position on a map and physically close objects that correspond to DBPedia resources.

A point and click interface is used to navigate to and select points of interest, and the *Marbles Linked Data Engine* (see section 4.1.3) is used to retrieve related information. Context and type of information are used to select suitable formats for presentation; combined with filtering to hide information of low relevance to the user this allows optimal use of small screen size on mobile devices. The results are processed on the server before being returned to the user in a text overlay that contains also links to third party LD browsers. Support is provided for querying via a simple query UI, and pre-specified SPARQL queries may be reused in the mobile version.

DBPedia Mobile also allows end users to publish information about their current location and any photos taken (via their FOAF profile), in addition to reviews (via Revyu) of selected resources (e.g., buildings, people associated with the area) in the vicinity; this feeds into the shared DBPedia repository as new linked data.

4.2.2. Fenfire

Hastrup et al. [12] describe the use of interactive graph visualisation in *Fenfire* to browse linked data. Fenfire was developed as a generic tool, to remove the restriction to specific domains encountered in other RDF browsers. The application has been extended to support (more specific requirements of) browsing linked data, in order to support both authors and end users to verify, explore the content and make use of such data. To allow scalability to the very large amounts of data typical in this domain, Fenfire displays the detail for the focus and its immediate neighbours, and fades away the graph with distance from the focus. An alternative list view for the focus is also available. Hastrup et al. [12] note the importance of using `rdfs:labels` to provide user-readable labels for nodes in the RDF graph.

Multiple documents may be loaded in the view; Fenfire will follow links defined between documents, and also attempt to generate dummy links between disconnected data. New graphs may be created and saved locally; changes to external documents loaded into the view may however not be published to its source.

4.2.3. IsaViz

IsaViz[54] is an interactive RDF graph browser and editor, with functionality for navigation through a 2.5D interface. Graph style sheets (GSS) [28] are overlaid on the graph built according to the underlying RDF model, in order to provide a more intuitive view on the underlying data. Using GSS also helps to reduce the high occlusion that occurs in graph layouts for large amounts of dense, highly interlinked data, by merging related information about an entity, for instance, into a meaningful visual representation (e.g., using a person icon to represent a `foaf:Person`). IsaViz also supports the use of Fresnel lenses. Detail for ROIs is provided using text overlays in tables.

4.2.4. LESS

Auer et al. [1] aim to lower the barrier to mainstream consumption of linked data through the use of *LESS* web-based templates to define visual representations of linked data [1]. LESS templates are written using LeTL (the LESS Template Language) and rendered using PHP[55] (PHP: Hypertext Preprocessor), producing HTML, RDF, RDFa[56] or other textual output such as JavaScript[57] or JSON. LESS templates may take as input a single data source, or synthesise multiple sources via SPARQL queries. The publicly shared web-based templates enable simplified re-use of the resulting visual output, e.g., by importing into other web pages and even mobile applications' UIs.

It should be noted that Auer et al. [1] do not report usability evaluation with their target (mainstream) end users. They do however indicate as future work further development to provide a visual front-end for the template design and query construction, and integration with relevant web services, to better support non-technical users to both generate and consume linked data.

4.2.5. OpenLink Data Explorer

OpenLink Data Explorer (ODE) is a web-based RDF data browser for interacting with linked data. ODE extracts metadata about SW resources and also relevant non-SW resources, and converts the latter to RDF.

A number of perspectives (or views) are available: ***What:*** lists the sources from which related information has been retrieved. The result set may be grouped by resource category (e.g., `sioc:link`); ***Where:*** plots data that contains geo-location information on a map view; ***When:*** plots any data with temporal attributes on an interactive timeline; ***Who:*** displays information about `Person` entities using a Fresnel template based on FOAF; ***Images:*** displays image data extracted from the sources; ***Grid view:*** displays the triples extracted as a table of subject-predicate-object (spo) entries; ***Tag Cloud:*** ; ***SVG Graph:*** generates a node-link graph to display the relationships between up to the first 100 triples extracted; filters may be applied to restrict the graph to an ROI; ***Navigator:*** a list of entities grouped by type; click-through may be used to navigate to successive levels of detail for each; ***Custom:*** this view allows the end user to load a Fresnel lens with which to format the display of selected resources.

A public web service is available, and ODE may also be installed as a Firefox extension. The latter integrates the extraction of metadata and the retrieval of additional relevant sources of information with the end user's normal web browsing and information retrieval activity (as does PiggyBank – see section4.1.4).

4.2.6. RDF Gravity

RDF Gravity is a visual browser for RDF and OWL documents. Support for basic text search to advanced querying using formal query syntax complements a visual filter to hide information of lower relevance. This

[54] `http://www.w3.org/2001/11/IsaViz`
[55] `http://php.net`
[56] Resource Description Framework âĂŞ- in âĂŞ-attributes: `http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014`
[57] `https://developer.mozilla.org/en/JavaScript_Language_Resources`

[57] `http://semweb.salzburgresearch.at/apps/rdf-gravity`

Fig. 14. The results are shown for a request for the URI `http://www.bbc.co.uk/music/beta` (a), with alternative views – a filtered and the default RDF graphs in (b) and (c) respectively, two snapshots from the image view (d), and the navigator view in (e). The relation to the query for the string "BBC Music" is shown bottom, left (f), with a start of a new browsing chain in (g).

is especially useful for managing the high occlusion that occurs for very large graphs, and for analysing detail in ROIs. RDF Gravity uses basic icons and colour coding to distinguish different resource types.

Point and select functionality may also be used to select and reposition nodes, to provide more physical space for ROIs, or to bring distant nodes closer together to aid visual inspection. Text overlays are used to display detail about nodes related to the focus in a table. Fig. 15 shows the URI `http://data.dcs.shef.ac.uk/person/Matthew-Rowe` from the *Data.dcs* dataset [30] loaded into RDF Gravity.

4.2.7. *RelFinder*

RelFinder aims to provide more usable interaction with linked data through graphical rendering of (automatically discovered) links between pairs of specified resources [15]. Support for analysing detail for the focus relies on interactively restricting and/or aggregating links drawn in the graph, based on the data attributes. A full data overview is maintained in a coupled table view. Auto-completion of input labels and validation of URIs support users in identifying valid nodes to search from. Colour coding is used to highlight related data properties.

A restriction in RelFinder is that the user must supply two known, linked entry points, a SPARQL query endpoint and the repositories to query on. This restricts

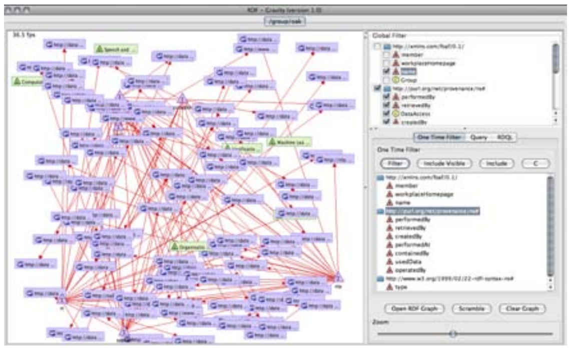


Fig. 15. Loading the URI `http://data.dcs.shef.ac.uk/person/Matthew-Rowe` into RDF Gravity displays a subset of the Data.dcs dataset, focusing on relationships between research groups and their members.

use to tech-users with at least a fair knowledge of data content. Fig. 16 illustrates two queries for relationships between entities in the music domain.

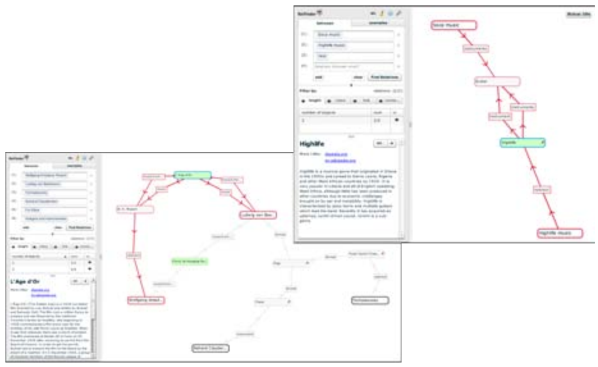


Fig. 16. RelFinder is used to search for any relationship between Soca and Highlife – an instrument used in both (snapshot top, right). No relationship is found between either and Jazz.
Bottom, left, a set of relationships are found between three classical composers and a modern pianist (Clayderman)

4.2.8. *Tabulator*

Tabulator was specifically developed as a generic linked data browser [4], using globally referenced concepts (such as FOAF) to interpret the content of linked data. Tabulator was developed to support, among others, the content of linked data resources and identify effective methods for presenting this to end users (both tech- and lay-users), using an interface that is easy to set up and learn to use.

The latest version of Tabulator also provides support for writing to the underlying data [5]. Tabulator uses a tree view (or a nested hierarchy) to display increasing levels of refinement as a user browses through a dataset. Data with temporal or geo-location properties may be displayed using a timeline or a map view respectively. Support for both naïve and expert SPARQL querying is provided.

4.2.9. Summary

Our review recognises the good level of support provided by text-based LD browsers for tech-users, but a dearth in support for lay-users (see section 4.1). A review of visualisation-based browsers reveals more intuitive support for especially lay-users, for two main reasons. First of all, visualisation provides a wider range of options for encoding data attributes, making it a good candidate for effective representation of heterogeneous, multi-attribute linked data. The other reason for the popularity of (especially well designed) visualisations is that they are able to take advantage of advanced perceptual ability in humans to reduce cognitive load, especially useful in complex, large-scale data analysis and knowledge retrieval. Visualisation therefore lowers the entry bar in the use of advanced technology for both domain experts and mainstream users.

Examining the top level requirements for effective visualisation support for consuming linked data, we find that half of the tools in this group provide a visual overview, in addition to functionality for detailed analysis of ROIs available in all 8 tools reviewed. 6 out of the 8 tools provide at least some degree of functionality for highlighting the links in the data, either explicitly, using graph views, and/or via simple, intuitive filtering mechanisms that highlight related data, within the context of the larger linked data cloud. Availability of basic text search and in some cases, advanced querying using formal syntax, combined with the alternate views in the visualisation tools, support both guided information retrieval and navigation through the data.

Although validation of the RDF model is explicitly supported only in IsaViz, which was developed as an RDF editor, those tools that provide a graph view allow end users to browse the data structure, and recognise quickly anomalies that would be more difficult to recognise large amounts of data in a textual view. Further, mapping the data to other representations that correspond to appropriate visual metaphors (e.g., geographical attributes to a map view) also reveals different perspectives on the data in addition to any anomalies in each view.

A major limitation in the visualisation-based approaches reviewed is limited support for browsing the unbounded Web. This restricts (exploratory) analysis and discovery to relatively small data stores that serve as static snapshots of selected regions of the larger Web of Data. This is partly due to the fact that most of the tools available were originally designed to support the browsing of XML and/or RDF models rather than linked data and the Semantic Web as an unbounded resource.

Analysing end user requirements according to the tasks they normally perform, the visualisation-based tools appear to offer more intuitive support for the exploratory knowledge discovery that is expected to guide lay-users' tasks in the consumption of linked data. The interactive visualisations also provide a fair degree of support to tech-users, as advanced functionality is provided in addition to the more intuitive but also more basic support for lay-users. The dearth in especially generic, basic visualisation support reveals a gap in the field that must be filled if linked data is to see consumption – use, reuse and enrichment of the linked data cloud – by the different end users targeted.

5. Findings

A number of RDF and linked data-specific browsers exist, in addition those we specifically review. These include *SIMILE Exhibit*[58], *LENA* [25], *Noadster*[59] [31], *mSpace*[60] [33], *Humboldt* [23], *Revyu* [14], *Semanlink*[61] [35], and the explore mode of the *Sesame*[62] semantic store.

This survey captures the range of functionality available for browsing, querying, editing and publishing linked data, employing text-based and visualisation-based UIs. Tables 1 and 2 summarise functionality we have identified as key to intuitive consumption of linked data. Tools are grouped into text-based presentation of linked data and those that are visualisation-based or provide visual representations in addition to textual output, in order to allow a comparison of the functionality available based on the target users and tasks of each tool.

We review the requirements for consuming linked data listed in section 3, to determine whether the approaches examined are able to support intuitive exploration of linked data that leads to knowledge discovery

[58] http://www.simile-widgets.org/exhibit

[59] http://media.cwi.nl/cocoon/cuypers/noadster/form

[60] http://mspace.fm

[61] http://www.semanlink.net

[62] Sesame: http://www.openrdf.org

Table 1
Comparison of functionality in the text-based LD browsers.

	LD Browser						
Usability Criterion	Dipper	Disco	Marbles	Piggy Bank	Sig.ma	URI Burner	Zitgist
Visual representation							
Presentation Templates	x*		x		x		x
Data overview							
Highlight of links in data			x*				
Detail on demand for focus	x*	x*	x*	x	x*	x*	x
Multi-dimensional data					x*		x*
Support for scalability					x		x*
Reusable output				x	x	x	x*
Edit underlying data		x*					
Non-domain specific		x	x	x	x	x	x
Support for querying			x		x	x	
Faceted Search / Browsing				x			x
Filtering					x		x
History				x			
Publication / Syndication				x			
Target – Lay-users				x			
Target – Tech-users	x	x	x		x	x	x

and the achievement of end users' overall goals. The discussion is broken down according to the distinct, high level requirements identified. It should be noted however that a degree of overlap exists between requirements; one feature may depend on the implementation of the functionality for another. Intuitive navigation, for example, is to a large extent dependent on the generation of effective data overviews and is also influenced by the ability to filter out irrelevant information. We will discuss the support provided by the different LD browsers, and highlight gaps that must be filled in order to result in truly intuitive functionality for consuming linked data, especially for the lay-users who face significant challenges in the use of this rich data.

5.1. Accessing Linked Data

The tech-user is able to read and understand RDF in its raw form, and may prefer to browse long lists of URI resources and RDF triples presented with little or no formatting. Domain experts may, to a lesser extent, be able to make use of their knowledge of data content in their field of expertise to browse large amounts of linked data in textual format, especially where labels in the RDF provide human-readable descriptions of the data. However, one of the most significant barriers to linked data consumption is access by lay-users to the rich knowledge contained when presented as RDF. Most LD browsers require input in the form of absolute, dereferenceable URIs. A sub-set of these require that these also point to RDF data; this requires both an understanding of SW protocols and a knowledge of data sources and content. This explains in large part the significantly lower consumption of linked data outside the SW domain.

Two examples of the use of linked data to support access and consumption by a lay audience are presented in this paper: a web-based application providing a set of perspectives with which to retrieve, browse and visualise health indicators in the US by geographical region (see section 2.2.2), by linking data in the health domain with cartographic data. The application allows end users to query the data using meaningful text labels (corresponding to data attributes) and browse the results using commonly available text (sortable table) and visual (map and scatter plot) representations. The second example is the BBC Music Beta project (see

Table 2

Comparison of functionality in the LD browsers with visual representations available

Usability Criterion	LD Browser: DBPedia Mobile	Fenfire	IsaViz	LESS	OpenLink	RDF Gravity	RelFinder	Tabulator
Visual representation	x	x	x	x	x	x	x	x
RDF graph view		x	x		x	x	x	
Presentation Templates			x	x	x			
Data overview							x	
Visual overview	x*		x			x*		
Highlight of links in data		x	x		x	x	x	x*
Detail on demand for focus	x	x	x	x*	x	x	x	x
Multi-dimensional data	x		x*		x			x
Support for scalability	x*	x	x		x*	x*	x	
Reusable output		x	x	x				
Edit underlying data			x					x*
Non-domain specific	x*	x	x	x	x	x	x	x
Support for querying	x	x*	x	x	x*	x	x*	x
Faceted Search / Browsing								
Filtering	x	x*	x		x	x	x	
History	x				x			
Publication / Syndication	x	x	x					x
Target – Lay-users	x	x*	x	x*	x*	x*		x*
Target – Tech-users		x	x	x	x	x	x	x

section 4), which is harnessing the principles of LOD to link related information across independent, previously unconnected legacy data repositories, in order to improve services to the BBC's audience. The data that has been converted to RDF may be accessed by lay-users directly from the BBC site, formatted as HTML pages and via other services that allow the retrieval of related information in other sub-domains.

In each case a layer is placed over the linked data to present it in a format suitable for the sub-tasks (e.g., information retrieval, analysis) and overall goals of the end user – invariably knowledge discovery toward a specific end (e.g., review strategies for public health policies in a local council; research the relationship between (West Indian) Soca music and (Ghanaian) Highlife, respectively).

To retrieve information about a resource of interest from the Web of Data its URI may be input to an LD browser or SW search engine. Where results are obtained the end user may interact with the linked data using the functionality available in their application of choice, as illustrated in the tool review in section 4. Disco and Dipper require input as an absolute URI that points to a document containing RDF data. Dipper further requires a data source to be specified. Marbles and Zitgist require only that the input URI be dereferenceable. Sig.ma and URI Burner accept input as a text string or a URI; both will attempt to match free text input to the most appropriate resource in the Web of Data, and query for a list of potentially relevant sources based on this. Piggy Bank integrates the retrieval and generation of linked data into regular web browsing activity. It transparently retrieves URIs where available for information of interest to the user, or extracts to new RDF data information stored in non-RDF format. A point and click interface is used to navigate to and select points of interest (available as linked data nodes) on a map in DBPedia Mobile. Fenfire and IsaViz require as input the URI for a document containing RDF data. RDF Gravity, RelFinder and Tabulator require as input an absolute URI that points to a document containing RDF data. RelFinder requires additionally, the specification of a SPARQL endpoint and the data source from which to retrieve related information. ODE requires a resolvable URI as input or a text string (for which it will attempt to locate a matching resource URI); where this does not point to a document containing RDF data, ODE attempts to extract metadata to an RDF representation and displays the result. When installed as a FireFox extension ODE works like Piggy Bank in that it automatically extracts SW resources and metadata about the current focus, and switches from this to the LD browser view on request, to allow direct interaction with the underlying inked data. LESS takes as input a (LESS) template, data source properties and a SPARQL query (which may include multiple files).

5.2. *Obtaining an Overview*

Advances in technology have resulted in ever increasing amounts of complex data; support for managing information overload is key to the usability of data analysis and knowledge discovery tools [22]. One technique for doing this involves the generation of data overviews. While visual overviews are most common they may take any other form suitable for the user and their task (section 5.7 provides a more detailed discussion of methods for general data presentation). Overviews help end users to obtain an understanding of overall data structure, and often serve as starting points for navigation to ROIs and detailed analysis [36,38]. IsaViz is the only tool that provides a dedicated overview, in a coupled pane, in addition to the central detail view. A limitation of text-based browsers is the poor support text provides for generating overviews. Where in use, presentation templates in the tools reviewed help to aggregate and format the information retrieved for the request, providing a descriptive summary and pointers to relevant information.

Support for handling scalability, in terms of data size and dimensionality, is also important for (large scale) linked data. All the tools surveyed perform some degree of processing to cluster or merge data and minimise redundancy. For instance, `owl:sameAs` constructs are used to highlight resources that point to the same information, and `rdfs:seeAlso` to point to additional information about the resource of interest. Custom methods that format output using templates (see section 5.7) use them to merge related information into a more human-readable format and to ensure correct interpretation of the processed data.

Multiple dimensions in text-based tools tend to be presented in sortable tables or grouped in lists. Faceted browsing such as in Piggy Bank exploits the multi-dimensionality of the underlying data to support intuitive filtering and browsing to ROIs. Sig.ma provides basic support for filtering to only selected data attributes. Multiple, linked views are also often used to highlight and compare different perspectives on a data set, by focusing on a sub-set of the data attributes in each. Visual representations are able to exploit pre-attentive human perception, in addition to making use of a larger number of cues for encoding multiple dimensions, including variation in colour (see e.g., Fig. 1b), representative icons (a visual cue often adopted in text-based representations, as done in Marbles).

5.3. *Navigation & Exploratory Discovery*

In order to support knowledge discovery, especially for the lay-user who will have little knowledge about data content and structure, LD browsers must support intuitive navigation within individual data sets and across the links between data. This must be combined with effective strategies for managing information overload, in order that end users do not simply get lost, but instead are able to identify the nuggets of valuable, relevant knowledge hidden within the very large Web of highly interlinked data. The presentation of data (see section 5.7) – from high level overviews to detail in ROIs, and support for intuitive filtering are among the most important contributors to effective navigation [31,36] through especially large, complex interlinked data of the kind we address.

The text-based LD browsers reviewed all make use of in-built navigation features in web browsers to provide click-through navigation from a resource of interest to other related information. Embedding applications in a browser however often disables the history mechanism built into the forward and backward navigation buttons on web browser tool bars, or results in inconsistent behaviour. Dipper, LESS, Marbles, Piggy Bank and URI Burner, however, serve output as HTML pages, allowing normal use of the navigation buttons. Piggy Bank users may browse information previously saved to their local or shared repositories. Sig.ma and Zitgist provide basic filtering, but neither has a history mechanism. Sig.ma additionally allows sorting and categorisation of resources retrieved, allowing users to stack information in order of highest relevance. ODE has an in-built history mechanism, allowing end users to return to previously executed queries. Tabulator uses a click-through mechanism to expand a collapsed tree view to reveal increasingly refined levels of detail for a selected resource. DBPedia Mobile, Piggy Bank and the ODE Firefox extension use semantic stores to collect linked data and tags attached to resources, to aid the retrieval of related information.

The visualisation-based representations make use of custom functionality for navigating through the data layouts, with the most basic support including point and click functionality for 'opening' or jumping to resources of interest. Fenfire additionally provides support for navigation using the keyboard. Pan and zoom are typical features available for browsing to ROIs in interactive visualisation, and filters are often also provided for suppressing less relevant information. Such features are used to navigate through the graph layouts in Fenfire, RDF Gravity and RelFinder. Level of detail is used in DBPedia Mobile to provide increasing detail as one moves to a previously unpopulated area of the map or zooms into ROIs. ODE uses filters to display the RDF graph and the links between up to 100 nodes at a time, supporting navigation within a sub-set of the data.

5.4. *Information Retrieval*

Directed information retrieval is carried out after end users obtain an understanding of data content and structure, to enrich knowledge discovered through initial exploration of a data space. The ability to directly query linked data using formal query syntax is key to successfully carrying out tech-users' normal activity. SPARQL end points are available for Zitgist (requires sign-up), Marbles and URI Burner. DBPedia Mobile, IsaViz, LESS, RDF Gravity, RelFinder, Tabulator provide functionality for the use of formal query syntax (including at least SPARQL). Openlink enables visual querying via ISPARQL; this is however not intuitive enough to be used effectively by lay-users..

While the power and flexibility of formal querying cannot be denied, simple, intuitive filtering and information retrieval is important, especially for lay-users; ODE, RDF Gravity, RelFinder, Sig.ma and ZitGist provide simple click through lists of resources to (de)select filters. Fenfire provides functionality for filtering based on data attributes. DBPedia Mobile provides a form-based filter & query interface. URI Burner's form-based precision search for text input and URI lookup support information retrieval by lay-users. Options for ordering and grouping information by data type and property are provided by Piggy Bank, Sig.ma and ODE, helping to identify and retrieve related information. PiggyBank makes use of this to provide faceted search/browse. Zitgist also provides a basic form of faceted browse/filter, based on data attributes of the resources retrieved.

5.5. *Detailed Analysis of ROIs*

Existing LD browsers focus on information retrieval across multiple, related data sets. While some degree of functionality is available for browsing through the Web of Data from a specified entry point for all the browsers reviewed, detailed inspection of ROIs and further processing and analysis tend to require more specialised functionality than is available in generic tools. This is especially important for text-based browsers, as human cognitive limits restrict the amount of information that can be processed effectively in text, especially where a large number of relationships must be followed to obtain a complete understanding of data. ODE allows detailed inspection of resources retrieved, across multiple views that highlight specific data attributes (such as location and temporal properties of data). Fenfire supports detailed inspection of the focus and allows the RDF being browsed to be saved to a local file. The graphical view in IsaViz allows a high degree of detail to be examined for the focus. Local editing and creation of data is also supported.

5.6. *Data Reuse*

The ability to export query and analysis results for reuse in other tools is often a requirement for end users; a significant advantage in linked data is that the common data encoding format simplifies information exchange between SW tools. Zitgist supports reuse of queries, by allowing the URLs to the HTML pages generated to serve the output to be saved and shared; it does not however provide support for extracting the underlying data. Sig.ma and URI Burner allow RDF to be exported directly from the browser interface to formats including JSON. Piggy Bank exports data – both resources and tags – to a local and a remote, shared semantic store. Marbles and ODE, installed locally, also make use of local data stores. It should be possible to extract RDF data directly from these semantic stores for reuse by third parties. Fenfire saves edited graphs and new RDF graph data locally.

5.7. Presentation

We have discussed the benefits encoding linked data in (machine-readable) RDF. However, it also poses significant challenges for consumption by especially lay-users (see, among others, [31,34]). Descriptive labels present in RDF are often used to format output, to improve readability and interpretation by end users. Where a URI points to data of type image or text/HTML content, the URI may be dereferenced to embed the corresponding image or web page respectively (see Fig. 11a), and served with the rest of the output in an HTML page.

However, this still leaves a large amount of the output in RDF, some of which may employ unintelligible identifiers (especially to non-domain experts). Generic templates provide an extra layer, formatting selected data and entity types to obtain more human-readable output. Pietriga [28] demonstrates the use of GSS in IsaViz to provide more intuitive presentation of RDF graphs and ease understanding of underlying data content. Fresnel lenses [29] are popular for defining implementation-independent templates for data presentation. These may be used to format, for example, information on entities of type `foaf:Person` or `foaf:Organization`, based on information encoded according to the FOAF ontology. Information extracted about a music artist or an album may be formatted according to the MO.

Fig. 13 shows the use of a template for the MO, to format and display information about the `mo:MusicArtist` Joe McElderry in the Zitgist mini dataviewer on an iPhone. Marbles and Sig.ma also use presentation templates to format selected entity types. Marbles supports the use of Fresnel lenses. Sig.ma provides basic, interactive editing of its presentation templates. DBPedia Mobile uses Fresnel lenses to display data in its LD browser overlay. LESS provides a custom language for building reusable templates for formatting output for consumption by lay-users. ODE allows end users to import Fresnel lenses to generate custom views on data of interest.

Well designed (reusable and/or extensible) templates that target specific users and tasks help to manage cognitive load and support more intuitive knowledge discovery, by transforming the complex data to a more easily interpreted representation, by merging and clustering related information and suppressing less important data attributes [29,31]. While visualisation provides more options for doing so, these examples illustrate that even text-based representations benefit from this approach.

5.7.1. Visualisation

Whether or not linked data is reused beyond its initial point of generation is influenced by data quality, the ability to identify links across data sets, and access to usable tools for directed information retrieval, exploratory knowledge discovery, analysis and the presentation of the results of interaction with the data [32]. This review has found significant support for LD consumption by tech-users, using predominantly text-based presentations. However, support for lay-users is much lower, being restricted largely to custom applications and visualisations built to suit specific end goals, using selected sub-sets of the Web of Data, as seen in the example in section 2.2.2. Such tools tend to make use of basic, commonly used techniques to generate one-off visual representations, often maps and timelines for data with geographical location and temporal attributes, respectively, and graphical plots and charts, such as scatter plots and bar and pie charts respectively, to support statistical analysis.

Visualisation has seen lower than expected coverage in the LD application domain, especially in the development of generic LD browsers, beyond the provision of node-link graphs used to display the structure of the RDF graphs described by linked data. A contributory factor may be the newness of the field; a fair portion of applications are still at the prototype stage. Further, these are often built to meet the requirements of tech-users, who are still the main consumers of linked data, chiefly to browse data content, determine its structure, carry out error checking and validate new data. It is not surprising that functionality in these tools is geared to support for these tasks.

We have however made a case for the benefits that visualisation could bring to the consumption of linked data, especially for mainstream use, but also for tech-users (see section 2.3). Generic browsers are necessary to promote mainstream LD consumption. Existing visualisation-based tools such as RDF Gravity and IsaViz, two well-known RDF browsers, attempt to parse the RDF graph structure and display this in a node-link layout. Such tools commonly use icons, node and edge colour and size to encode data attributes. High level data overviews are often provided, especially useful for very large data sets, in addition to support for navigation to and detailed analysis in ROIs, to support intuitive discovery. Clustering and level of detail filters are also often provided to support interaction with especially dense data.

RDF Gravity currently provides greater support for the tech-user than the lay-user. RelFinder provides interactive graph visualisation targeted at the tech-user with at least basic knowledge of data content. RelFinder focuses on reveal interlinking between data; it should be very useful to application developers working with domain experts, for instance, to reveal data structure and relationships across independent data set. Fenfire aims to support predominantly those tasks carried out by tech-users, but also knowledge discovery by lay-users, with its simple graph layout that focuses on detail at the user's point of interest. Tabulator provides a degree of support for visual, interactive querying, to select data to be displayed in its map, calendar and timeline views. ODE also provides map and time line views, in addition to a number of other views that allow focus on specific data entities and attributes. Both Tabulator and ODE however still provide more intuitive support for tech-users than for lay-users.

IsaViz provides the most intuitive support for visual exploration and detailed analysis and editing of ROIs, suitable for both tech- and lay-users, through the use of GSS and presentation templates that render a human-readable layer on top of the RDF graph. Tech users who wish to explore the plain graph may disable the template overlays. Support for advanced querying also helps to filter out data of low relevance from the view. Fresnel and similar support for defining generic, reusable and extensible templates will be especially useful for the design of new, intuitive visualisation-based LD browsers. LESS generates formatted output targeted at lay-users; however a degree of technical expertise is required to learn how to create new templates.

DBPedia Mobile makes use of a familiar metaphor for navigation through unfamiliar geographical spaces ([38] explains the power in adopting such approaches). The DBPedia Mobile interface is a good example of the use of a custom visualisation as an entry point to the Web of Data.

A challenge to exploratory discovery in the Web of Data is the requirement for the user to specify an entry point, normally as a valid, dereferenceable URI. DBPedia Mobile serves as a good example of an application that infers a suitable entry point for the end user, by automatically revealing resources based on geographical location. The technical barrier is removed, and the lay-user is able to make use of experience in the real world to navigate the Web of Data. Two other browsers, Sig.ma and URI Burner, are able to parse free text entry, allowing a more familiar method of information retrieval for the lay-user. Piggy Bank and the ODE Firefox plugin also infer user interest based on their current focus, and allow the user to switch seamlessly to navigation of related resources as linked data.

A final challenge in the use of LD browsers configured to browse the unbounded Web is handling responses such as "No data found in this store" or "No further information is available" for valid URIs. While the latter response implies that no information was found for the URI in question, the former corresponds to a failure to locate the resource. A particular advantage in the use of visualisation to explore unbounded data is the ability to provide a preview of the next step for not just the current focus, but also linked nodes, using level of detail functionality that reveals more detail as a user navigates through a data space. DBPedia Mobile provides similar functionality – as a user zooms in to an ROI or pans across the map previously hidden resources are revealed. This gives the user a preview of the knowledge in new, previously unexplored regions. If the user fails to locate a known resource in a region of the data space other information in the surrounding area may provide related information and/or relevant information that may help to locate and retrieve the resource desired. Advances in technology support the generation of both simple and augmented visual layouts able to support such views, which relying on intuitive, preattentive perception in humans, place little cognitive load on the end user. Greater cognitive load associated with managing large amounts of textual information and limitations to screen real estate however prevent the generation of similar detailed previews in text-based representations.

5.8. Publication / Syndication

Intuitive support for publishing new data into the LOD cloud is necessary to ensure high quality and reusability, and therefore promote the value and consumption of linked data by especially lay-users [34]. Tech-users are likely to write custom or extend existing scripts to automatically generate and dump large-scale data into the LOD, and where necessary, extract metadata from legacy data and convert it to RDF. Lay-users on the other hand require more transparent and intuitive support to publish smaller amounts of data; weaving data syndication into users' normal information retrieval and analysis activity is one of the more effective routes to this end. For instance, a free text tag attached to a photograph of a cultural icon uploaded to shared repository as part of a tourist's review may be automatically extracted as metadata to an RDF representation and attached to the original description of tourist attractions in the geographical area encoded as a resource in the Web of Data. DBPedia Mobile takes such an approach, using where necessary, third party applications such as Revyu [14] as a portal to the Web of Data. Piggy Bank also provides functionality for capturing tags as metadata to local and/or shared semantic repositories.

Fenfire supports local editing of data and the creation of new content. Plans to enable direct publishing to the Web of Data should provide more effective support for especially lay-users interacting with the graphs. IsaViz was developed to support editing of RDF graph models. It therefore provides extensive support for both simple and complex graphs. While lay-users are likely to encounter a relatively steep learning curve IsaViz represents the most usable of the graph editors reviewed.

5.8.1. Data Validation

Both publishing new and editing existing data sets requires validation, i.e., checking for errors, noise and redundancy in the data. To carry out such tasks tech-users visually inspect the plain RDF or use simple graph views that display the structure of the RDF graphs described by linked data. These tasks are however normally carried out transparently in applications targeted at lay-users; data entry forms such as provided in DBPedia Mobile guide input to ensure that it can be transformed to valid RDF. Tools such as IsaViz provide good support for error identification and validation of the RDF model.

6. Discussion & Conclusions:

In this paper we have presented an analysis of current approaches used to browse and visualise linked data. The motivation behind this survey was to present the reader with a clear definition of the differences between such approaches, and their advantages and limitations. By identifying requirements that such approaches must address, requirements which in turn were motivated by the context of application over the Web of Data and the huge expanse of data that is to be dealt with, our analysis has identified where linked data can be used by both tech-savvy and lay-users.

We divided the approaches for browsing linked data into two types: text-based and browsers with visualisation options. Both types of approaches have their places and usage. For instance, the former allows for fine grained analysis of a given datum, whereas the latter allows the bigger picture to be seen. In essence the provision of visualisation options supports human perceptual ability [22,32,36]. Bombarding end users with rows of text is not effective in conveying a structural description of a resource and how it is associated with other resources in disparate datasets. However, large-scale data visualisation is not always the best solution to complex analysis, instead presentation should be based on the requirements of the end user and their specific tasks [20,31,32,36].

Of the array of approaches and tools surveyed we found only two which, based on our requirements, offered a usable solution to browsing linked data by lay-users. The uptake of linked data by a mainstream audience is dependent on its utility to those outside the Semantic Web and Linked Data communities, therefore this lack of support for non-tech-savvy users could inhibit its adoption.

Analysis of the approaches did however identify several useful features that support end users. Firstly, breadcrumb features, which are present in the majority of web-browser based approaches – particularly if the browser states are maintained – enable users to maintain a path of exploration through the Web of Data. As a consequence, end users are able to keep track of their progress through the duration of their exploration activity, thereby providing a history of the paths that were traversed. Second, the use of templates associated with given ontological concepts transforms data returned by a URI into a legible, usable form. We found this to be particularly useful for URIs that contained many pieces of information within the instance description. When dereferenced, such URIs, depending on the linked data browser used, would display several pieces of information, thereby making it hard for the end users to decipher and interpret the response.

It is evident that within the Web of Data, a chicken and egg problem exists [1,8,17,27,35]: where tools are needed to locate, retrieve and use linked data, and without *good quality* linked data, there is little motivation to design and build such tools for end users external to the Linked Data community. Mainstream consumption requires *reusability* [1] where data can be combined together, or the current focus of exploration saved and then re-loaded, or passed on at a later point. Heath [13] states that the sum is greater than its parts, where using the Web of Data to discover information about information enables expansion on existing knowledge, something which needs to be shared.

Formalising data using RDF principles goes a long way to obtaining a uniform representation of self-describing data, allowing the links across data to be recognised, as seen in LOD, and promoting data reuse [17], among others. This lowers the restrictions peculiar to specific domains and tasks, and increases the ability to develop more generic tools for data exploration and analysis. Using Fresnel lenses [29] and other methods for creating generic presentation templates [31,34], for instance, is one step toward independent declaration of the presentation of data. This will not completely remove the need for custom tools - they are better at performing specific tasks [4,31]; however the design of tools for the Semantic Web are increasingly able to provide support for user customisation, allowing a truly user-centred perspective on analysis.

A key conclusion that we drew from our work was the limited number of linked data browsers currently available. We assume that this can be attributed to the infancy of this domain. Although the Semantic Web vision has been pursued for circa 10 years, it is only over the past 3 years that the production of machine-readable data has been generated at a sufficient scale that consuming it has now become a large-scale task. We therefore believe that the consumption of linked data has now become a major research challenge, one in which the visualisation and browsing of linked data will play a crucial role.

Acknowledgements

Aba-Sah Dadzie is funded under the European Commission (EC) 7th Framework Programme, by the SmartProducts project (EC grant number 231204). Matthew Rowe is funded under the EC 7th Framework Programme, by the WeGov project (EC grant number 248512).

References

[1] Sören Auer, Raphael Doehring, and Sebastian Dietzold. LESS - template-based syndication and presentation of linked data. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *ESWC 2010: The Semantic Web: Research and Applications, Proc., 7th Extended Semantic Web Conference*, pages 211–224, 2010.

[2] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *AAAI 2009: Proc., International AAAI Conference on Weblogs and Social Media*, 2009.

[3] Christian Becker and Christian Bizer. Exploring the geospatial semantic web with dbpedia mobile. *Journal of Web Semantics (Web Semantics: Science, Services and Agents on the World Wide Web)*, 7(4):278–286, 2009.

[4] 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 *SWUI 2006: Proc., 3rd International Semantic Web User Interaction Workshop*, 2006.

[5] T Berners-Lee, J. Hollenbach, Kanghao Lu, J. Presbrey, E. Prud'ommeaux, and mc schraefel. Tabulator redux: Browsing and writing linked data. In *LDOW 2008: Proc., Linked Data on the Web Workshop at WWW'2008*, 2008.

[6] Tim Berners-Lee. Linked data – design issues. Technical report, W3C, 2006.

[7] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[8] Gong Cheng and Yuzhong Qu. Searching linked objects with falcons: Approach, implementation and evaluation. *International Journal on Semantic Web and Information Systems*, 5 (3):49–70, 2009.

[9] Richard Cyganiak and Anja Jentzsch. Linking open data cloud diagram. `http://lod-cloud.net`, 2010.

[10] Jean-Daniel Fekete, Jarke J. van Wijk, John T. Stasko, and Chris North. The value of information visualization. In A. Fekete, Kerren C., J. North, and T. Stasko, editors, *Information Visualization: Human-Centered Issues in Visual Representation, Interaction*, pages 1–18. Springer, 2008.

[11] Christiaan Fluit, Marta Sabou, and Frank van Harmelen. Supporting user tasks through visualisation of light-weight ontologies. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 415–434. Springer, 2004.

[12] Tuukka Hastrup, Richard Cyganiak, and Uldis Bojars. Browsing linked data with fenfire. In *LDOW 2008: Proc., Linked Data on the Web Workshop at WWW'2008*, 2008.

[13] T. Heath. How will we interact with the web of data? *IEEE Internet Computing*, 12(5):88–91, 2008.

[14] Tom Heath and Enrico Motta. Revyu: Linking reviews and ratings into the web of data. *Journal of Web Semantics: Special Issue on Science, Services and Agents on the World Wide Web*, 6(4):266–273, 2008.

[15] Philipp Heim, Steffen Lohmann, and Timo Stegemann. Interactive relationship discovery via the semantic web. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *ESWC 2010: The Semantic Web: Research and Applications, Proc., 7th Extended Semantic Web Conference*, pages 303–317, 2010.

[16] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the Pedantic Web. In *LDOW 2010: Proc., Linked Data on the Web Workshop at WWW'2010*, 2010.

[17] David Huynh, Stefano Mazzocchi, and David R. Karger. Piggy bank: Experience the semantic web inside your web browser.

Journal of Web Semantics, 5(1):16–27, 2007.

[18] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.

[19] M. Jern, T. Astrom, and S. Johansson. Geoanalytics tools applied to large geospatial datasets. In *IV '08: Proc., 12th International Conference on Information Visualisation*, pages 362–372, 2008.

[20] David Karger and m.c. schraefel. The pathetic fallacy of RDF. In *SWUI 2006: Proc., 3rd International Semantic Web User Interaction Workshop*, 2006.

[21] Daniel A. Keim. Visual exploration of large data sets. *Communications of the ACM*, 44(8):38–44, 2001.

[22] Tanja Keller and Sigmar-Olaf Tergan. Visualizing knowledge and information: An introduction. In Sigmar-Olaf Tergan and Tanja Keller, editors, *Knowledge and Information Visualization*, volume 3426 of *LNCS*, pages 1–23. Springer, 2005.

[23] Georgi Kobilarov and Ian Dickinson. Humboldt: Exploring linked data. In *LDOW 2008: Proc., Linked Data on the Web Workshop at WWW'2008*, 2008.

[24] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets Semantic Web - how the BBC uses DBpedia and linked data to make connections. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, *ESWC 2009: The Semantic Web: Research and Applications, Proc., 6th European Semantic Web Conference*, pages 723–737, 2009.

[25] Jörg Koch and Thomas Franz. LENA - browsing RDF data more complex than FOAF. In Christian Bizer and Anupam Joshi, editors, *ISWC2008: Proc., 7th International Semantic Web Conference (Poster and Demonstration Session)*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[26] P. Mutton and J. Golbeck. Visualization of semantic metadata and ontologies. In *Proc. Seventh International Conference on Information Visualization, 2003. IV 2003.*, pages 300–305, 2003.

[27] Tope Omitola, Christos L. Koumenides, Igor O. Popov, Yang Yang, Manuel Salvadores, Martin Szomszor, Tim Berners-Lee, Nicholas Gibbins, Wendy Hall, m. c. schraefel, and Nigel Shadbolt. Put in your postcode, out comes the data: A case study. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *ESWC 2010: The Semantic Web: Research and Applications, Proc., 7th Extended Semantic Web Conference*, pages 318–332, 2010.

[28] Emmanuel Pietriga. Semantic web data visualization with graph style sheets. In *SoftVis '06: Proc., 2006 ACM symposium on Software visualization*, pages 177–178, 2006.

[29] Emmanuel Pietriga, Christian Bizer, David R. Karger, and Ryan Lee. Fresnel: A browser-independent presentation vocabulary for RDF. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web: Proc., 5th International Semantic Web Conference*, pages 158–171, 2006.

[30] Matthew Rowe. Data.dcs: Converting legacy data into linked data. In *LDOW 2010: Proc., Linked Data on the Web Workshop at WWW'2010*, 2010.

[31] Lloyd Rutledge, Jacco van Ossenbruggen, and Lynda Hardman. Making RDF presentable: integrated global and local semantic web browsing. In *WWW '05: Proc., 14th International Conference on World Wide Web*, pages 199–206, 2005.

[32] J. Scholtz. Beyond usability: Evaluation aspects of visual analytic environments. In *IEEE VAST 2006: Proc., IEEE Symposium On Visual Analytics Science And Technology*, pages 145–150, 2006.

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

[34] m.c. schraefel, Jennifer Golbeck, Duane Degler, Abraham Bernstein, and Lloyd Rutledge. Semantic web user interactions: exploring HCI challenges. In *CHI '08: Proc., 26th annual SIGCHI conference on Human factors in computing systems*, pages 3929–3932, 2008.

[35] François-Paul Servant. Linking enterprise data. In *LDOW 2008: Proc., Linked Data on the Web Workshop at WWW'2008*, 2008.

[36] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc., IEEE Symposium on Visual Languages, 1996*, pages 336–343, 1996.

[37] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: live views on the web of data. In *WWW '10: Proc., 19th international conference on World wide web*, pages 1301–1304, 2010.

[38] Ji Soo Yi, Youn-Ah Kang, John T. Stasko, and Julie A. Jacko. Understanding and characterizing insights: how do people gain insights using information visualization? In *BELIV '08: Proc., 2008 conference on BEyond time and errors*, pages 1–6, 2008.