Interacting with Linked Data through an Automatic Information Architecture

Roberto García, Josep Maria Brunetti, Antonio López-Muzás,
Juan Manuel Gimeno, Rosa Gil

Universitat de Lleida
Jaume II, 69. 25001 Lleida, Spain
{rgarcia, josepmbrunetti, jmgimeno, rgil}@diei.udl.cat, lopezmuzas@gmail.com

Abstract. Thanks to Open Data initiatives the amount of data available on the Web is rapidly increasing. Unfortunately, most of these initiatives only publish raw tabular data, what makes its analysis and reuse very difficult. Linked Data principles allow for a more sophisticated approach by making explicit both the structure and semantics of the data. However, from the user experience viewpoint, published datasets continue to be monolithic files completely opaque or difficult to explore by making tedious semantic queries. Our objective is to facilitate the user to grasp what kind of entities are in the dataset, how they are interrelated, which are their main properties and values, etc. Rhizomer constitutes a tool for data publishing whose interface provides a set of components borrowed from Information Architecture (IA) that facilitate awareness of the dataset at hand. It automatically generates navigation menus and facets based on the kinds of things in the dataset and how they are described through metadata properties and values. Rhizomer is currently being evaluated with end-users that discover a whole new perspective of the Web of Data.

Keywords. Ontology, Semantic Web, Linked Data, metadata, human-computer interaction, usability.

1 Introduction

The amount of data available in the Web, in its transition to a Web of Data[1], is increasing, especially thanks to Open Data initiatives[2]. Unfortunately, most of these initiatives are limited to publishing raw data very difficult to analyse and reuse. A more sophisticated approach is to use Semantic Web and Linked Data principles[1] when publishing data. They make the structure and semantics of the data explicit so data is easier to explore and integrate.

However, from the user perspective, the problem continues to be that datasets are monolithic files completely opaque, or that can only be browsed using complex

[1] Tim Berners-Lee on the next Web. TED Talks, 2009.

[2] World Government Data. The Guardian, 2011.
http://www.guardian.co.uk/world-government-data

semantic queries. The objective is to make all this data more usable so users facing a dataset can easily grasp what kind of entities are in there, how they are interrelated, what are their main properties and values, etc.

This way, users can search, browse and analyse the data. This will facilitate, for instance, that application developer not experienced in Semantic Web technologies can easily realise what a dataset is and thus facilitate the development of new and innovative applications on top of it. The overall outcome is that available data increases its impact and the society as a whole benefits more from data openness.

The best approach to make a dataset more usable to a wider range of users is to use some sort of Web publishing tool. At least, this kind of tools usually provides a HTML rendering for each resource in the dataset. Each HTML page lists all the properties for the corresponding resource. Pages are interlinked based on the connections among resources and the user can follow HTML links to browse through them.

However, this feature is only useful if the user has some a priori knowledge about the dataset, especially the URI identifier of a given resource. There is no way to get an overview of the kind of resources in the dataset. Some data publishing platforms like OpenLink's Virtuoso do provide a faceted view on a specific subset of the data, but in order to get it, it is necessary to provide an URI or some keywords for textual search.

Our proposal is to draw from the experience accumulated in the Information Architecture (IA) domain [2] and reuse and adapt existing IA components to provide this kind of guides to users. This kind of components is well known to Web users, as they are present in most web pages. They are navigation bars, navigation facets, sitemaps, breadcrumbs, etc.

This approach is tested in an application, called Rhizomer, that can be deployed on top of any dataset based on Semantic Web technologies and publish it, while facilitating user awareness of what is in there. This awareness is accomplished by the IA components. For instance, navigation bars show the main kinds of resources in the dataset and facets show the more significant properties for different kinds of resources and their values.

The rest of this paper is organised as follows. First, the related work is presented in Subsection 1.1. Then, Rhizomer is introduced in Section 2 together with the main Information Architecture components. The tool has been evaluated with end-users and a short summary of the results are presented in Section 3. Finally, conclusions and future plans are presented in Section 0.

1.1 Related Work

The first tool that comes to mind when trying to realise what a dataset is about is a browser. Semantic Web browsers differ from Web browsers because they are not prepared for navigating documents but triples, the fundamental building block of the Semantic Web.

Browsers are especially useful when dealing with a dataset published as Linked Data because they provide a smooth browsing experience through the graph, e.g. Disco [3]. However, they just provide a view of a subject, or set of subjects, and their properties. Moreover, users must know the identifier of the subject of interest or perform a query to retrieve it.

Most Semantic Web browsers do not provide a broader view of the dataset being browsed. They provide just a view on the current resource or at most of the steps followed so far using something similar to breadcrumbs. In some cases, it is also possible to get more informative components like facets but they are not generated dynamically for any input dataset, they are tied to specific applications that allow browsing specific datasets, for instance DBPedia Faceted Browser [4]. To our knowledge, the only browser capable of dynamically generating a faceted view for Semantic Web datasets is \facet [5].

It is important to note that this kind of tools makes it possible to browse datasets previously published as Linked Data. If the dataset is just available as a dump file or SPARQL service, then some extra tools or a different approach is required.

Explorator [6] is a tool that makes it possible to explore a dataset available through a SPARQL endpoint. Though Explorator makes it possible to explore the dataset by combining search, facets or operations on sets of resources, like /facet, it makes it also difficult to get a broader view on the dataset other than a list of all the classes or properties used.

Other alternatives are Content Management Systems (CMS) or Wikis with semantic capabilities. Some mainstream CMSs and wiki systems have started to incorporate semantic technologies. The most significant case is the last version of Drupal[3] that provides features such as semantic metadata storage, querying, importation or rendering.

However, semantic CMSs, like Drupal, or portal building solutions, like ODESeW [7], are intended for content creation. Consequently, they make it harder to publish an existing dataset and making its structure available to the user for exploration. It is usually necessary to build this structure manually using features like templates or content creation kits.

The same applies to semantic wikis, such as the semantic extension for MediaWiki. This extension, called Semantic MediaWiki [8], makes it possible to mix wiki mark-up with semantic annotations. As in the case of CMSs, it is also possible to import existing data but the wiki does not provide mechanisms that facilitate user awareness about the structure of the data that has been imported.

Finally, there are some tools that are specialised in publishing Linked Data. The aspect that we consider here is the kind of support they provide to users when accessing the dataset and try to get an idea about what it is about.

One kind of these tools is the one that publishes relational databases as linked data, like D2R Server [9] or ODEMapster [10]. However, they provide the same kind of view on the published data that Semantic Web browsers provide, i.e. resources and their associated properties but no general view on the dataset.

There are also specialised tools that publish existing datasets as Linked Data, like Paget[4] or Pubby[5]. Both build an HTML frontend with dereferenceable URIs for the

[3] Drupal 7, http://drupal.org/drupal-7.0

[4] http://code.google.com/p/paget/

[5] http://www4.wiwiss.fu-berlin.de/pubby/

subjects in the served dataset and content negotiation. However, in both cases, the frontends provided are just like those Semantic Web browsers have.

To conclude, it is also possible to consider platforms for semantic data storage and publishing like Talis Platform[6] or OpenLink Virtuoso[7]. In both cases, in addition to the data stores, there is an HTML frontend for the datasets. However, like with previous tools, the support for broader awareness of the dataset structure is very limited.

The most significant contribution is in OpenLink Virtuoso, which provides a faceted view on a specific subset of the data, but in order to get it, it is necessary to provide an URI or some keywords for textual search. Consequently, the facets view is limited to the resources retrieved from a previous search and there is no way to previously get an overview of the kinds of resources in the dataset.

2 Rhizomer

Rhizomer[8] is a Semantic Web data-publishing tool. Its architecture is rooted on simple HTTP mechanisms and follows a REST approach [11]. It also implements content negotiation taking into account the requested content type thus providing the requested data in the desired format, and HTML rendering for human users or a RDF/XML one for computers.

Each resource is managed through the URI referencing where it is published, thus basing the whole system on a Resource Oriented Approach. The basic HTTP commands allow managing each resource. All the HTTP commands (and the SPARQL queries) are forwarded to the underlying data store. Currently, Rhizomer features connectors to Jena, Virtuoso and OWLIM stores. These connectors make it possible to implement all the data management operations, especially updates, using the REST mechanism independently of the SPARQL Update availability.

On the other hand, client-side functionalities have been developed with the aim of improving the usability of the user interface. They are deployed in the user's browser and implemented using JavaScript and asynchronous HTTP calls (AJAX [12]), thought most of the functionality is also available without JavaScript in order to improve accessibility [13].

All the interactions with the users are built on top of the REST operations. However, the RDF syntax of semantic data is completely hidden in order to increase usability. Like many Semantic Web browsers of data publishing tools, Rhizomer provides an HTML view on the data that also facilitates the navigation across the data graph based on an XSLT transformation that also embeds RDFa in the resulting HTML[9].

However, as it has been stressed in the related work section, this approach does not contribute towards an awareness of the overall structure of a dataset. In order to provide such functionality, Rhizomer features a set of components inspired by those

[6] http://www.talis.com/platform

[7] http://virtuoso.openlinksw.com

[8] http://rhizomik.net/rhizomer

[9] http://rhizomik.net/html/redefer/rdf2html-form/

common in Information Architecture. The added value is that in the case of Rhizomer, these components are automatically generated and updated from the data, thesaurus and ontologies in the published dataset, as described in Sections 2.1 and 2.2.

2.1 Navigation Menus

Navigation menus, in the case of websites, let users navigate through different sections and pages of the site. They tend to be the only consistent navigation element, being present on every page of the site.

Traditionally, user-centred design techniques like Card Sorting [14] are used to develop them. This technique requires a lot of time and effort from developers and if new kinds of items are introduced or a part of the content becomes more relevant, the Card Sorting should be repeated.

The opportunity in the case of web sites built on top of semantic data is to automate part of this generation and maintenance process. This is possible because semantic data is structured by thesaurus and ontologies that hierarchically organise the kinds of things described in the dataset. If there are few members of some class or category, or there are not at all, it should be less relevant in the menu bar. On the contrary, those that do have a lot of members should be shown more prominently.

To do this, we obtain the hierarchy of all kinds of entities and apply inference rules to get their members. Then, the hierarchy is flattened to the amount of levels defined because this component can generate both global and local menus, i.e. a menu for the whole dataset or for a subset of it. The site administrator can define the following parameters:

- The number of levels in the hierarchical menu.
- The number of items in each level of the menu.
- The order of items: alphabetical or by number of members.
- A list of classes or categories to omit.

According to these parameters, this component generates the menu applying a recursive algorithm that mainly performs two operations:

1. Split the categories or classes with more members (direct or inherited from subcategories or subclasses) if there are free slots at a given level.
2. Merge those with few members into a supercategory or superclass if there are more elements at a given menu level than available slots.

This approach makes it possible to show the user the navigation bar that best fits the data in the dataset at that particular moment. For instance, if the dataset changes from containing mainly data about projects to publications, the menu would change accordingly to show more prominently the dataset part about publications.

For instance, Fig. 1 illustrates how the 7 slots defined for a submenu corresponding to DBPedia class “Species” are filled starting from the original class hierarchy and instantiation counts from DBPedia. At first, there are just 3 slots occupied, the direct subclasses of “Species”, so the most instantiated one, “Eukariote” is split and its direct subclasses become direct subclasses of “Species”. They occupy 2 additional slots, 3 still free, so the most instantiated one of the 4 is split, i.e. “Animal”. It has more than 4 direct subclasses so now it is necessary to merge the less instantiated classes till there are just 4 direct subclasses. The class resulting from the merging is automatically labelled “Other Animal”, their direct superclass.

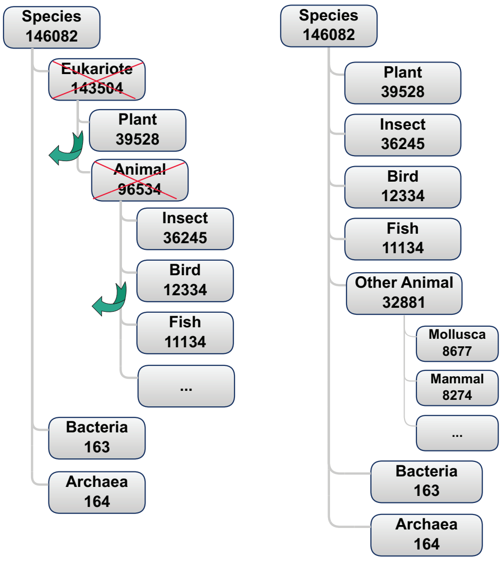


Fig. 1. Generating a navigation submenu for DBPedia Species with 7 slots (left original, right result)

2.2 Facets

Users don’t always know exactly what they are looking for and, sometimes, they don’t even know what its name is. Other times, they are unfamiliar with the domain or they want to learn about a topic. This is particularly true when facing for the first time a particular Semantic Web datasets. In these cases, exploratory search is a strategy that allows users to refine their search by successive iterations. An exploratory interface such as faceted browsing allows users to find information without a priori knowledge of its schema.

With navigation menus we can make the user aware of the hierarchical structure of a dataset but, once they choose the class of things they are interested in, they face the barrier of not knowing how they are described. In other words, what are the main properties that describe them, which ones are the more relevant for that particular kind of things, the range of values they have in that particular case, etc.

Traditional facet browsers rely on manual identification of the facets and on previous knowledge of the target domain. When dealing with semantic data, it is possible to automate this process and a semantic faceted browser should be able to handle any RDF dataset without any configuration. Since Linked Data facilitates integrating data from different sources, we cannot assume a single fixed schema for all data. Consequently, a Linked Data faceted browser should be scalable and generic, not depending on a particular dataset structure.

Facets are pre-computed when Rhizomer is first launched for a dataset by issuing SPARQL queries that, for each class instantiated in the dataset, retrieve all the properties used in the descriptions of the instances of that class, as shown in the left part of Table 1. The queries also retrieve the different values each of those properties have and the cardinality for each value, i.e. how many times a property for a class takes a value, right part of Table 1. From this query results, facets are then calculated and stored in a data structure. And from that moment, they are incrementally updated whenever the dataset is edited through Rhizomer.

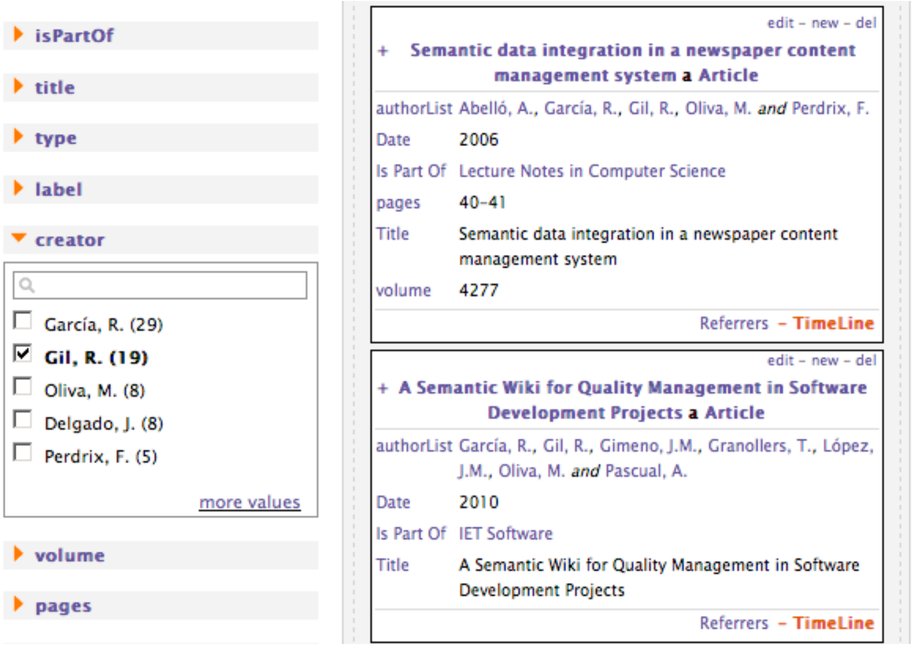
Table 1. Templates fort he queries to compute facets

Properties for [CLASS]	**Values and Counts for [CLASS] and [PROPERTY]**

`SELECT DISTINCT(?p) ?r`
`WHERE { ?i a <[CLASS]> ; ?p ?o.`
`OPTIONAL { ?p rdfs:range ?r } }`
	`SELECT ?o (COUNT(?o) AS ?n) ?label`
	`WHERE { ?r a <[CLASS]>; <[PROPERTY]> ?o.`
	`OPTIONAL{ ?o rdfs:label ?label } }`
	`GROUP BY ?o ?label ORDER BY DESC(?n)`

These pre-computed facets are used whenever a user wants to browse a class, usually selecting it from the navigation bar. Facets are rendered as HTML, as it shown in Fig. 2. However, when a user starts interacting with them, by selecting values, facets are dynamically recalculated starting from the pre-computed ones.

This makes possible to provide reasonable response times to users because most of the computing effort has already been done and what user interaction does is to restrict the set of resources to work with. The set of instances used for facets generation is constrained by the choices made so far and the facets are recalculated for that subset.



□ García, R. (29)
☑ **Gil, R. (19)**
□ Oliva, M. (8)
□ Delgado, J. (8)
□ Perdrix, F. (5)

+ **Semantic data integration in a newspaper content management system a Article**

authorList Abelló, A., García, R., Gil, R., Oliva, M. *and* Perdrix, F.
Date 2006
Is Part Of Lecture Notes in Computer Science
pages 40-41
Title Semantic data integration in a newspaper content management system
volume 4277

+ **A Semantic Wiki for Quality Management in Software Development Projects a Article**

authorList García, R., Gil, R., Gimeno, J.M., Granollers, T., López, J.M., Oliva, M. *and* Pascual, A.
Date 2010
Is Part Of IET Software
Title A Semantic Wiki for Quality Management in Software Development Projects

Fig. 2. Automatically generated facets component screenshot

Finally, as it can be seen in Table 1, the queries also retrieve additional optional information. In the case of the first query, the property ranges are also retrieved. They are used, when available, to customise the facet depending on the kind of values. For instance, a histogram facet for numeric values is work in progress. The second query also retrieves value labels, when available, so it is possible to present them in the facets instead of the URI, or part of it, making the result more usable and even capable of supporting language selection when xml:lang attribute is present.

3 Evaluation

Rhizomer, though still at a prototype stage, has already been tested with end-users in order to evaluate its functionality and usability. The goal of the test conduced so far was to preliminary evaluation of the Information Architecture components, if they are understood and if they improve the awareness of the structure of a particular dataset by improving user performance when looking for a specific piece of information.

We used a real test dataset called the Linked Movie Data Base (LinkedMDB)[10]. We chose the movies domain because it is well known for most people and quite appealing. LinkedMDB is generated from the Internet Movie Database[11] (IMDB). Therefore, we considered interesting to compare the evaluation results with those for IMDB and thus be able to test if the same data as Linked Data can become more usable than from the original web site.

We established one scenario with one task to be performed with IMDB and another one with one task for Rhizomer. The proposed tasks where like **"**Find three films where Woody Allen is director and actor at the same time" using IMDB and "the same for Woody Allen" using Rhizomer. The summary of the evaluation results is presented in Table 2.

Table 2. User test results for IMDB and Rhizomer for LinkedMDB

IMDB metrics			
		Task time (minutes)	**3.37**
Success	**100%**	Success	3.37
		Not terminated	-
Asking for assistance	5	**Efficiency**	**32%**
		Success	32%
Rhizomer-LinkedMDB metrics			
		Task time (minutes)	**2.41**
Success	**100%**	Success	2.41
		Not terminated	-
Asking for assistance	7	**Efficiency**	**54%**
		Success	54%

[10] LinkedMDB by O. Hassanzadeh and M. Consens, awarded 1st prize at the Linked Open Data Triplification Challenge 2008, http://triplify.org/Challenge/2008

[11] IMDB database http://www.imdb.com/

The main outcome was that, though most users needed some assistance to complete the tasks with both tools, Rhizomer was more efficient with 83% of the users completing the task with Rhizomer in less time than with the IMDB interface. The problem with Rhizomer was that all participants began the navigation from actors instead than from movies. This was the reason why users required assistance but as soon as they realized they were able to start from movies and restrict them using the actor and director facets, the task was easily solved.

4 Conclusions and Future Work

As it has been shown, Rhizomer implements a set of features that make it possible to deploy it on top of any dataset based on Semantic Web technologies and Linked Data principles and publish it, while facilitating user awareness of what is in there. This awareness is accomplished by components borrowed from the Information Architecture discipline.

Currently, these components are navigation bars, which show the main kinds of items in the dataset, and facets, which show the more significant properties for different kinds of items and their values. And what is more important, these components are automatically generated and maintained.

Our preliminary tests with users show that Rhizomer facilitates publishing and browsing a dataset, like many other similar tools, but also allows that users realise what is the value of the dataset in the context of their particular needs. It has also shown its scalability from small datasets like the one for the Rhizomik initiative[12] to really big ones like LinkedMDB[13]. Both datasets can be accessed online through Rhizomer at the provided URIs.

The future work focuses now on exploring ways to combine facets, and facets for inverse properties, so it is possible to avoid the problems detected during the evaluation tests that made users unable to complete the tasks if they started navigation from actor or director instead of from movie. Other areas of future work are related with more generic aspects of facets.

For facets, one objective is to generate facets customised to the kind of values being managed, i.e. numerical values, alphabetical values, dates, geographical points, etc. Another objective is to improve the functionality of this component by adding other operators for restricting the results: inverse selection, ranges, etc.

To conclude, the evaluation with end-users has been particularly fruitful. Though the results for Rhizomer have been better than those for the non-semantic version of the data and user interfaces, the main contribution of the test has been qualitative, i.e. the amount of improvements spotted by the way real users interact with the tool.

[12] http://rhizomik.net

[13] http://rhizomik.net/linkedmdb

Acknowledgments

The work described in this paper has been partially supported by Spanish Ministry of Science and Innovation through the Open Platform for Multichannel Content Distribution Management (OMediaDis) research project (TIN2008-06228).

References

[1] Berners-Lee, T.: Linked Data. Design Issues, (2009). http://www.w3.org/DesignIssues/LinkedData.html

[2] Morville, P., Rosenfeld, L: Information Architecture for the World Wide Web. O'Reilly Media, Cambridge, MA (2006).

[3] Bojars, U., Passant, A., Giasson, F., Breslin, J. G.: An Architecture to Discover and Query Decentralized RDF Data. Proceedings of Workshop on Scripting for the Semantic Web, SFSW 2007. CEUR Workshop Proceedings, Vol. 248 (2007).

[4] Hahn, R., Bizer, B., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., Scheel, U.: Faceted Wikipedia Search. 13th International Conference on Business Information Systems (BIS'10), Berlin, Germany (2010).

[5] Hildebrand, M., Ossenbruggen, J., and Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. The Semantic Web - ISWC 2006. Springer. (2006) 272-285.

[6] Araujo S., Schwabe D., Barbosa S.: Experimenting with Explorator: a Direct Manipulation Generic RDF Browser and Querying Tool. Visual Interfaces to the Social and the Semantic Web (VISSW 2009), Sanibel Island, Florida (2009).

[7] Corcho, O., López-Cima, A., Gómez-Pérez, A.: The ODESeW 2.0 semantic web application framework. Proceedings of the 15th International Conference on World Wide Web, WWW '06, ACM Press (2006) 1049-1050.

[8] Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic MediaWiki. Proceedings of the Int. Semantic Web Conference, ISWC'06. LNCS Vol. 4273, (2006) 935-942.

[9] Bizer, C. and Cyganiak, R.: D2R Server - Publishing Releational Databases on the Semantic Web. Poster at the 5th International Semantic Web Conference, Athens, USA (2006).

[10] Rodriguez, J.B., Gómez-Pérez, A.: Upgrading relational legacy data to the semantic web. Proceedings of the 15th international conference on World Wide Web, New York, NY, USA (2006) 1069-1070.

[11] Richardson, L. and Ruby, S.: Restful Web Services. O'Reilly, Cambridge, MA (2007).

[12] Crane, D., Pascarello, E., James, D.: Ajax in Action. Manning, Greenwich, CO (2005).

[13] García, R., Gimeno, J.M., Perdrix, F., Gil, R., Oliva, M., López, J.M., Pascual, A., Sendín, M.: Building a Usable and Accessible Semantic Web Interaction Platform, World Wide Web, Vol. 13, No. 1-2 (2010) 143–167.

[14] Spencer, D.: Card Sorting, Rosenfeld Media, New York, USA (2009).