Towards Faceted Browsing over Linked Data

Zhenning Shangguan and Deborah L. McGuinness

Tetherless World Constellation, Rensselaer Polytechnic Institute
110 8th Street, Troy, New York 12180, USA
shangz@cs.rpi.edu, dlm@cs.rpi.edu

Abstract

As the pace of Linked data generation and usage increases, so does the interest in intelligent, usable, and scalable browsing tools. Faceted browsing has potential to provide a foundation for effective dataset navigation. In this paper, we will discuss some of the anticipated benefits along with some associated challenges in building the next-generation faceted browsing system for the Web of Linked Data. We also present our initial system design and implementation.

Introduction

As more and more data following the Linked Data rules (Berners-Lee 2007) becomes publicly available on the web, the Web of Linked Data can now be considered as a global dataspace. One benefit of this huge dataspace is that building applications by dynamically meshing up data from heterogeneous sources now becomes possible (Bizer et al. 2009). However, research challenges still remain in the area of interaction paradigms that support user-friendly and efficient navigation through the growing dataspace (Heath 2008).

Faceted browsing (also called faceted search or faceted navigation) as a popular mechanism to accomplish exploratory tasks has been widely studied over the past years. Numerous research efforts (Hildebrand et al. 2006, Oren et al. 2006, Schraefel et al. 2003) have demonstrated benefits, particularly usability and flexibility, of faceted browsers when interacting with structured data (e.g., relational databases, XML data, RDF data, and etc.). Since the Web of Linked Data can be viewed as structured data connected by typed links, we are exploring the benefits of applying faceted search on the Web of Linked Data. In this paper, we focus on the following two key benefits:

• Interactivity: Users will be able to arrive at the desired information by progressively applying constraints and narrowing down the target dataspace.

• Understandability: Users will be able to gain a better understanding of the dataspace when navigating forward and backward within it. And consequently, the cognitive load (Chandler et al. 1996) needed to locate the final target information shall also be lowered.

In this paper, we argue that faceted browsing provides a promising foundation for exploratory Linked Data navigation because of the benefits aforementioned. However, building such a solution is not without its research challenges. Therefore, we will identify some of these challenges and then discuss our initial research and design efforts in this direction. Finally, our current conclusions and future directions will also be presented.

Challenges

The sheer size and growth rate of the Linked Data space impose great challenges on building the faceted browsing facilities. These challenges can be roughly divided into areas that are predominantly front-end and back-end.

Front-end challenges

A well designed user interface (UI) should be able to lower the cognitive load (Chandler et al. 1996) imposed on the user in general by requiring less information to be kept in the user’s working memory. We argue that this principle requires the UI to be informative, thereby providing access to relevant and required information that the user would otherwise have had to find or know on their own. Challenges increase related to determining relevant information when faced with a large and growing dataspace.

If the faceted navigation process is viewed as a sequence of consecutive decision-making procedures, an informative UI should be able to provide context information to guide the users to the final target. There are many existing research efforts contributing to this general principle from different perspectives. mSpace (Schraefel et al. 2006), Exhibit (Huynh et al. 2007), and Flamenco (Yee et al. 2003) all present users with a dynamic UI and narrow the target dataspace.
dataset leveraging user click information. One common characteristic of these systems is that they all employ a single-focus paradigm in which the target information and the facets are pre-defined by the system developers (e.g., the online newsfilm archive in mSpace\(^1\)). Thus, the changing facet values can be viewed as the most significant context information that helps users in their decision making.

Beyond these single-focus systems, there are also multi-focus faceted browsing tools like Humbolt (Kobilarov et al 2008) and Freebase Parallax\(^2\) that enable users to pivot or refocus to different target datasets. Therefore, these systems provide additional context information in the form of paths traversed and dynamically generated facets. This not only enables the user to be aware of his current position in the dataspace, but also provides the capability of returning to a previous decision to follow a different path. However, ideal solutions to generate these additional context information remains a great research challenge because extra computation might be needed given the huge dataspace we are facing.

Although these faceted browsing systems provide an informative and responsive interface to the user, they are all limited as they either do not scale well to large datasets or only handle a fixed dataset. Most of them can be viewed as faceted browsers targeting a fixed structured dataset that is either light-weight or independent from other datasets. Therefore, their benefits begin to decrease when dealing with datasets consisting of very large sets of interconnected entities. Unless they evolve, they are unlikely to address large web-scale datasets.

### Back-end challenges

When faced with Linked Data at web scale, scalability becomes a major concern. The interactions between the users and the faceted browsing systems often generate multiple SPARQL queries, sometimes with arbitrary join, aggregation, and filtering operations (Erling et al 2008). It remains an open research question on how to evaluate these complex queries at web scale under critical time constraints.

Some recent research efforts are beginning to address these issues. Hartig et al (Hartig et al 2009) proposed an online approach to query the Web of Linked Data by traversing RDF links during run-time to discover data that might be relevant to the query. In contrast, there are also research efforts on federated query mechanisms over distributed data sources that are cached in a central data store. Erling and Mikhailov (Erling et al 2009) discuss various aspects of SPARQL query evaluation techniques used by the Virtuoso Cluster Edition over a large corpus of linked data, including runtime inference, partial query evaluation, and identity reasoning. These approaches all try to solve the scalability issue either by suitable pre-fetching and caching strategies or by increasing the computational power of the servers. However, numerous challenges remain for back-end technologies to truly support scalable Linked Data browsing. For example, appropriate mechanisms to rank the entities taking the semantics of the typed links into consideration may greatly facilitate populating the UI of the faceted browser by generating facets that are most relevant to the target information.

### Initial System Implementation

Our system is under active development as of this writing. Currently most of our work is focused on design and implementation of the user interface, aiming at addressing the challenges discussed in the previous section. Key features of our current interface design include:

- **Non-directional Facets**: We do not provide directional associativity between facets as mSpace does. Instead, a non-directional paradigm is used in which different facets have equal significance. Using non-directional facets in a Linked Data faceted browser might bring about intensive computation because sometimes it may be necessary to recompute the facets and their values depending on the user selection.

- **Facet Filters**: In many existing systems, facets are limited only to the properties of the target dataset. However, there is no reason that a facet itself cannot be described by other facets. Such facets-of-facets are called indirect facets. We implement indirect facets as filters on direct facets and we refer to these as facet filters. In this way, we can avoid facet chains of arbitrary length and facet cycles discussed in (Clarkson et al 2009).

- **Paging**: When a SPARQL query returns too many results to be displayed within one screen, paging techniques are used to show different subsets of the result set. Currently, since we only retrieve data from SPARQL endpoints, paging is well supported by using the LIMIT and OFFSET modifiers in the SPARQL query.

Figure 1 shows one experimental UI of our prototype. The data is retrieved from a SPARQL endpoint backed by Jena TDB\(^3\) and Joseki\(^4\). The current layout is similar to that used in most of the existing systems: the focus dataset is displayed in the center panel, while facets are presented on the right and from top to bottom.

To implement an interactive user interface, we use Google’s Web Toolkit\(^5\), which is convenient for building optimized cross-browser Ajax applications. Currently we also have a helper Java Servlet to issue SPARQL queries against SPARQL endpoints that are publicly available. We currently fetch results in XML format, which are then parsed to populate the user interface.

---

1. mSpace Demo: [http://mspace.fm/](http://mspace.fm/)
Conclusion and Future Work

In this paper, we outlined some of the key challenges, both front-end and back-end, of building a faceted browsing system over the Web of Linked Data. Our initial efforts to build such a system and address some of these challenges, highlighting some front-end issues, are also presented. Our design and implementation is evolving and a future release is expected to be online by symposium time for public test and evaluation.

There are various aspects that we consider are worthy of investigation in our future work. We plan to approach these aspects from both front-end and the back-end.

• Front-end: We intend to add features that will further improve the UI and facilitate user decision-making. For example, we plan to support facets with hierarchical structure, e.g., geographic locations, by showing a tree view of the parent-child relationship within a single facet. We also plan to investigate how to combine the single-focus and the multi-focus paradigm, with the former offering the ability to narrow the data space as quickly as possible, and the latter being able to help explore the data space as widely as possible. Additionally we plan to carry out a user study of both qualitative and quantitative nature to evaluate and refine our user interface design.

• Back-end: Currently our prototype system is only able to fetch data from SPARQL endpoints using RESTful web services over HTTP. As a further step, we are beginning to study how to effectively retrieve data published on the Web exploiting the knowledge of data publication subject to the Linked Data rules. Moreover, additional work on scalability is required and we plan to investigate appropriate strategies related to pre-fetching and caching to improve performance.

References


