LODScape: Ontology-Based Multiple-LOD Object Browser

Kavi Mahesh, Shruthi Chari and Shrinidhi Ramakrishnan

Centre for Knowledge Analytics and Ontological Engineering - KAnOE,
Department of Computer Science,
PES Institute of Technology, Bangalore 560085 INDIA
drkavimahesh@gmail.com   http://kanoe.org

Abstract:

Vast amounts of information are becoming available in Linked Open Datasets (LOD). Users need an effective way to browse LODs in a human-readable form through a rich, natural interface similar to the familiar web browser. While it appears rather straightforward to build a browser for an LOD or one that even integrates data elements on a given subject from multiple LODs, in reality it is not so. The browser must be able to filter uninteresting data elements and also sort and arrange available facts to make the browsing experience natural and effective. In this project we present LODScape, an ontology-based object browser for multiple LODs. A carefully constructed ontology provides an organization and classification of predicates which are used to render a nontrivial subject effectively. We further demonstrate that by making small changes to the ontology, the behavior of the browser can change significantly, thereby allowing customized or personalized browsing experiences. LODScape is an initial version of the browser. In future versions, we intend to support object-based navigation and contextual aides for users to efficiently find the information they are seeking. LODScape is especially useful in domains with a rich set of predicates whose data is aggregated from multiple LODs.

Keywords: LOD browser, semantic browser, object browser, ontology, linked open data.

The *LODScape* application is available on the Web through our landing page at **http://kanoe.org/kanoeapps/lodscape.html** (or directly at http://27.251.237.245/themes/TryFreebase.html).

1 Introduction

Large semantic datasets are becoming available through recent developments in Semantic Web technologies. These datasets are typically represented as RDF triples or quads, often with suitable ontologies, and published in the form of Linked Open Datasets (LOD). However, LODs are primarily meant for machines and are not easy to browse. Users need an effective way to browse LODs in a human-readable form through a rich, natural interface similar to the familiar web browser. While it may seem that it is rather straightforward to build a browser for an LOD or one that even integrates data elements on a given subject from multiple LODs, in reality it is not so. The browser must be able to filter uninteresting data elements and also sort and arrange available facts to make the browsing experience natural and effective. Otherwise, the user's experience will not be far from directly reading RDF triples.

In the LODScape project, we have attempted to design and implement an LOD browser called LODScape that renders a set of triples having the same subject using the classification of the predicates involved in an OWL ontology. Such an LODScape ontology is primarily a property hierarchy which guides LODScape in determining whether to display a triple and where and in which order to place on the screen.

In the rest of this paper, we present an overview of previous work on LOD browsers and the design of LODScape. We illustrate how making a change to the ontology changes the rendering of LOD data in LODScape. We conclude with a discussion of potential extensions to LODScape and further work in ontology-based applications for making linked data human-readable. It must be clarified that we are using the term "object" as in "object-oriented" modeling or programming as the set of all facts known (or inherited or inferred) about a given subject (or resource/entity/topic), not as in the third element of an RDF triple.

2 LOD Browsers

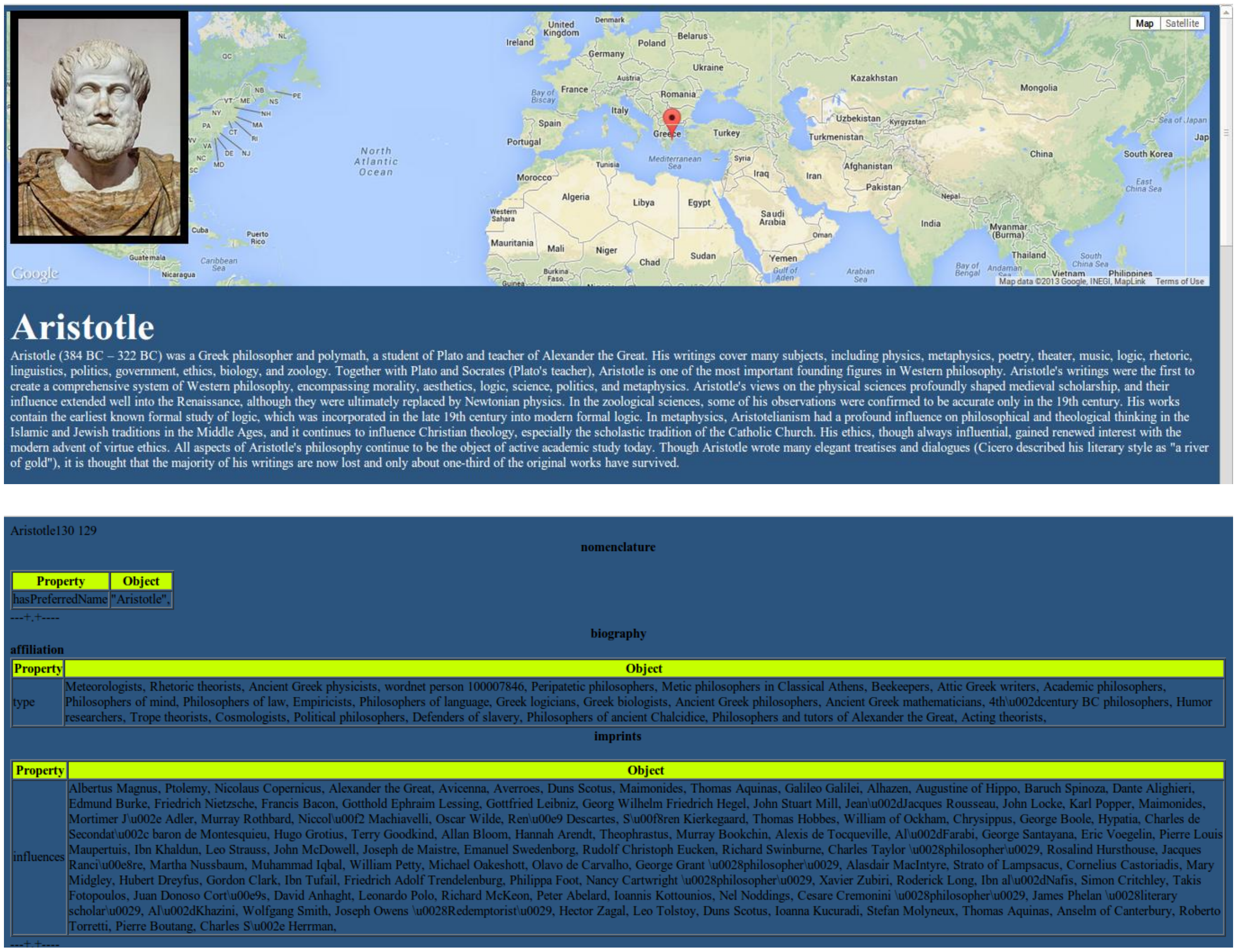
Several LOD browsers have been built already. While the simplest ones merely list triples alphabetically in tabular form, some of them, e.g., Fenfire [1], take the approach of helping users visualize an RDF graph in which the relationships (i.e., predicates) between different objects are shown. Others present one resource at a time and are usually called object viewers or browsers. Tabulator [2] is a file-based browser and works as an extension to FireFox for RDF browsing. Another such browser is OpenLink Data Explorer (ODE, formerly called OperLink RDF Browser) [3] and is suitable for viewing embedded RDF data in web pages, combined with dereferencing of RDF in linked data. Other such popular browser, Disco [4] and Object Viewer [5] presents triples to users without any semantic organization. Zitgist RDF Browser [6] and Marbles [7] take novel approaches using an "information shape shifter" and "Fresnel lenses" respectively, to help users visualize RDF data. Humboldt [8] is a facet browser for objects using the idea of pivoting.

It may be argued that the above types of LOD browsers are not semantic browsers since they do not perform any semantic grouping, ordering or filtering operations on the (predicates in the) data before displaying them. A notable exception is [9] where a semantic LOD browser is built by automatic semantic grouping using predicate similarity and clustering algorithms such as K-Means. The success of such automatic grouping in generating semantically coherent groups is limited largely by the accuracy of available algorithms for automatic clustering or classification of the predicates in a domain.

In LODScape, we take an entirely different approach wherein an ontology of the various predicates of interest is manually constructed. The classification and organization of various predicates in the ontology determine whether, where and how a triple carrying a particular predicate is rendered for the user. Unlike some of the above browsers, however, LODScape in its present form is not intended to either view embedded semantic data or to edit the data being viewed. Unlike file-based RDF viewers, LODScape is designed for users to browse through multiple, large LODs which are themselves available in the back-end in databases, triple stores or through SPARQL endpoints.

3 Design of LODScape

LODScape is designed to support two primary functions: searching for information about a particular object and browsing by category (i.e., a known rdf:type). In either case, further navigation is possible by clicking on any hyperlink in the resulting "page," thereby providing the familiar user experience of Web browsers. Fig. 1 shows a screenshot of a part of the output page from LODScape for the resource "Aristotle".



Aristotle

Aristotle (384 BC – 322 BC) was a Greek philosopher and polymath, a student of Plato and teacher of Alexander the Great. His writings cover many subjects, including physics, metaphysics, poetry, theater, music, logic, rhetoric, linguistics, politics, government, ethics, biology, and zoology. Together with Plato and Socrates (Plato's teacher), Aristotle is one of the most important founding figures in Western philosophy. Aristotle's writings were the first to create a comprehensive system of Western philosophy, encompassing morality, aesthetics, logic, science, politics, and metaphysics. Aristotle's views on the physical sciences profoundly shaped medieval scholarship, and their influence extended well into the Renaissance, although they were ultimately replaced by Newtonian physics. In the zoological sciences, some of his observations were confirmed to be accurate only in the 19th century. His works contain the earliest known formal study of logic, which was incorporated in the late 19th century into modern formal logic. In metaphysics, Aristotelianism had a profound influence on philosophical and theological thinking in the Islamic and Jewish traditions in the Middle Ages, and it continues to influence Christian theology, especially the scholastic tradition of the Catholic Church. His ethics, though always influential, gained renewed interest with the modern advent of virtue ethics. All aspects of Aristotle's philosophy continue to be the object of active academic study today. Though Aristotle wrote many elegant treatises and dialogues (Cicero described his literary style as "a river of gold"), it is thought that the majority of his writings are now lost and only about one-third of the original works have survived.

Aristotle130 129

nomenclature

Property	Object
hasPreferredName	"Aristotle",

---+.+----

biography

affiliation

Property	Object
type	Meteorologists, Rhetoric theorists, Ancient Greek physicists, wordnet person 100007846, Peripatetic philosophers, Metic philosophers in Classical Athens, Beekeepers, Attic Greek writers, Academic philosophers, Philosophers of mind, Philosophers of law, Empiricists, Philosophers of language, Greek logicians, Greek biologists, Ancient Greek philosophers, Ancient Greek mathematicians, 4th\u002dcentury BC philosophers, Humor researchers, Trope theorists, Cosmologists, Political philosophers, Defenders of slavery, Philosophers of ancient Chalcidice, Philosophers and tutors of Alexander the Great, Acting theorists,

imprints

Property	Object
influences	Albertus Magnus, Ptolemy, Nicolaus Copernicus, Alexander the Great, Avicenna, Averroes, Duns Scotus, Maimonides, Thomas Aquinas, Galileo Galilei, Alhazen, Augustine of Hippo, Baruch Spinoza, Dante Alighieri, Edmund Burke, Friedrich Nietzsche, Francis Bacon, Gotthold Ephraim Lessing, Gottfried Leibniz, Georg Wilhelm Friedrich Hegel, John Stuart Mill, Jean\u002dJacques Rousseau, John Locke, Karl Popper, Maimonides, Mortimer J\u002e Adler, Murray Rothbard, Niccol\u00f2 Machiavelli, Oscar Wilde, Ren\u00e9 Descartes, S\u00f8ren Kierkegaard, Thomas Hobbes, William of Ockham, Chrysippus, George Boole, Hypatia, Charles de Secondat\u002c baron de Montesquieu, Hugo Grotius, Terry Goodkind, Allan Bloom, Hannah Arendt, Theophrastus, Murray Bookchin, Alexis de Tocqueville, Al\u002dFarabi, George Santayana, Eric Voegelin, Pierre Louis Maupertuis, Ibn Khaldun, Leo Strauss, John McDowell, Joseph de Maistre, Emanuel Swedenborg, Rudolf Christoph Eucken, Richard Swinburne, Charles Taylor \u0028philosopher\u0029, Rosalind Hursthouse, Jacques Ranci\u00e8re, Martha Nussbaum, Muhammad Iqbal, William Petty, Michael Oakeshott, Olavo de Carvalho, George Grant \u0028philosopher\u0029, Alasdair MacIntyre, Strato of Lampsacus, Cornelius Castoriadis, Mary Midgley, Hubert Dreyfus, Gordon Clark, Ibn Tufail, Friedrich Adolf Trendelenburg, Philippa Foot, Nancy Cartwright \u0028philosopher\u0029, Xavier Zubiri, Roderick Long, Ibn al\u002dNafis, Simon Critchley, Takis Fotopoulos, Juan Donoso Cort\u00e9s, David Anhaght, Leonardo Polo, Richard McKeon, Peter Abelard, Ioannis Kottounios, Nel Noddings, Cesare Cremonini \u0028philosopher\u0029, James Phelan \u0028literary scholar\u0029, Al\u002dKhazini, Wolfgang Smith, Joseph Owens \u0028Redemptorist\u0029, Hector Zagal, Leo Tolstoy, Duns Scotus, Ioanna Kucuradi, Stefan Molyneux, Thomas Aquinas, Anselm of Canterbury, Roberto Torretti, Pierre Boutang, Charles S\u002e Herrman,

---+.+----

Fig. 1. Screenshot of Partial LODScape Output for "Aristotle"

Two LODs are supported in the initial version: DbPedia and Freebase. We plan to add YAGO and other datasets in the near future. Quads from the datasets are imported into the back-end store which is a MySql database in the current version. Subjects and predicates are indexed for efficient retrieval.

LODs are enhanced by applying the Jena engine to their ontology (if available) and RDFS relationships (primarily rdfs:subClassOf) to derive new triples. Additional rdf:type triples are thus added to the available LODs.

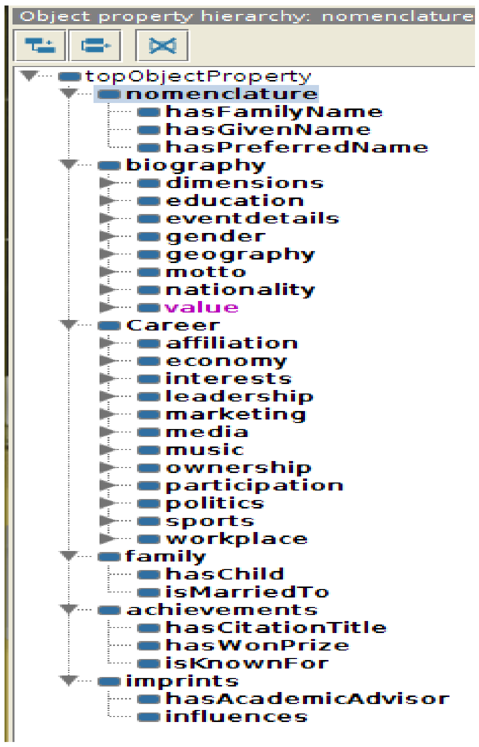
Predicates present in the enhanced LODs are organized manually in an OWL ontology built on top of the Freebase and DbPedia ontologies. The ontology is essentially an owl:ObjectProperty hierarchy. Intermediate categories of predicates were created to combine Freebase and DbPedia predicates into related groups. Fig. 2 shows a snapshot of the ontology used in the current version where predicates used in DbPedia's `PersonData` are organized under the broad categories of `biography`, `career`, `family`, `achievement` and `imprints`. The hierarchical relationships among the predicates are imported into the back-end database via JDBC and indexed to improve the performance of browsing by avoiding having to query the OWL ontology at run time.

Given a resource, all triples with that resource as the subject are retrieved. The triples are then subjected to filtering, grouping and sorting operations. Any predicate that is not found in the ontology is filtered out (unless all predicates for the given resource are unknown to the ontology). The subset of predicates that are retained are then grouped and ordered depending on their organization in the ontology. For example, all predicates under `biography` are grouped together, and, within the group, triples are ordered in the same order in which they are indexed in the imported ontology. The groups (and subgroups) themselves are also ordered in a similar way. Finally, the filtered, grouped and sorted triples are displayed on the Web page along with any image and/or text retrieved for the given resource from Wikipedia or other linked collection on the Web. Fig. 3 shows the architecture of LODScape.

Given a category, all resources which are of rdf:type that category are retrieved and presented to the user in a pull-down menu. When the user selects a resource from the menu, all facts known about the resource are retrieved and rendered as outlined above through the operations of filtering, grouping and sorting.

Both functions support auto-completion as the user is typing a keyword, thereby allowing users to browse through very large, multiple LODs without knowing exact names for the information being sought.

In the implementation of LODScape, PHP was used as the server side language and JavaScript on the client side. Ajax was used to enable auto-completion of user inputs. The frontend user interface was modeled using the Bootstrap framework. LODScape runs on a single HP ProLiant ML110 G7 server with 16 GB RAM and 8TB disks.



```
Object property hierarchy: nomenclature
topObjectProperty
    nomenclature
        hasFamilyName
        hasGivenName
        hasPreferredName
    biography
        dimensions
        education
        eventdetails
        gender
        geography
        motto
        nationality
        value
    Career
        affiliation
        economy
        interests
        leadership
        marketing
        media
        music
        ownership
        participation
        politics
        sports
        workplace
    family
        hasChild
        isMarriedTo
    achievements
        hasCitationTitle
        hasWonPrize
        isKnownFor
    imprints
        hasAcademicAdvisor
        influences
```

Fig. 2. Snapshot of Grouping of Predicates in LODScape OWL Ontology

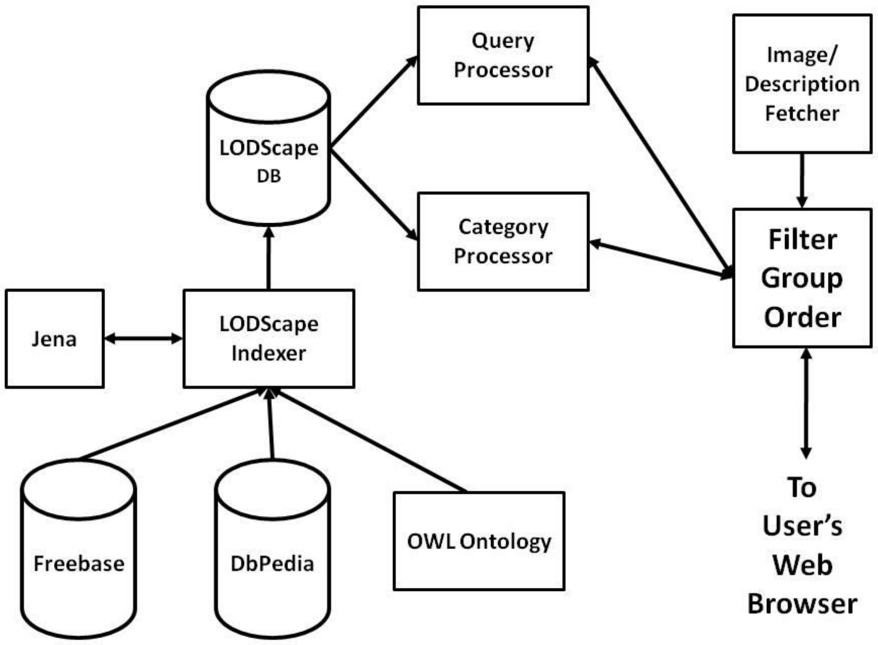
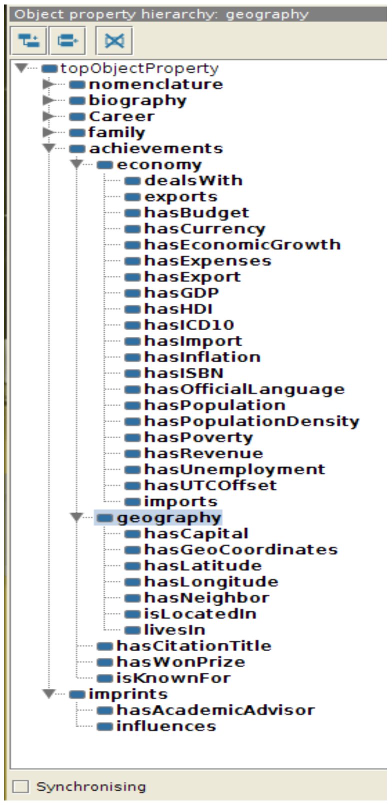


Fig. 3. Architecture of LODScape

4 Modifying the Ontology to Customize Browsing

The Freebase ontology was originally structured and indexed in LODScape such that the `nomenclature` appeared first in the output page, followed by `biography` (e.g., birth details, location, etc.), `career` (e.g., workplace). `family`, `achievements` and `imprints` (i.e., people whom the subject influenced or was influenced by). Although this may be a reasonable order for most users, LODScape allows a user to change this for all of the data in the LODs by simply modifying the ontology. To demonstrate this capability, let us consider a user who modifies the ontology such that the subclasses `geography` and `economics` which were under `biography` and `career` respectively are moved under `achievements`! A snapshot of the modified ontology is shown in Fig. 4. One could argue from this user's perspective, for example, that highly relevant information such as the location of a place, it's population density and GDP are displayed at the end which were left out in the original output.



```
Object property hierarchy: geography
topObjectProperty
    nomenclature
    biography
    Career
    family
    achievements
        economy
            dealsWith
            exports
            hasBudget
            hasCurrency
            hasEconomicGrowth
            hasExpenses
            hasExport
            hasGDP
            hasHDI
            hasICD10
            hasImport
            hasInflation
            hasISBN
            hasOfficialLanguage
            hasPopulation
            hasPopulationDensity
            hasPoverty
            hasRevenue
            hasUnemployment
            hasUTCOffset
            imports
        geography
            hasCapital
            hasGeoCoordinates
            hasLatitude
            hasLongitude
            hasNeighbor
            isLocatedIn
            livesIn
        hasCitationTitle
        hasWonPrize
        isKnownFor
    imprints
        hasAcademicAdvisor
        influences
Synchronising
```

Fig. 4. Modification to the LODScape Ontology: economy and geography are under achievements!

5 Design Choices and Further Work

LODScape is currently providing a minimally functional browser for the combined set of facts from DbPedia and Freebase. We plan to explore further opportunities for ontology-based control of the behavior of the LOD browser. For instance, in the future, in addition to grouping of predicates in an owl:ObjectProperty hierarchy, we would like to enable users to select various categories of interest to them so that facts related to those categories are highlighted during browsing. We will also provide an optional user feedback mechanism for the relevance of various facts on the screen from which LODScape can learn the likes and dislikes of particular users.

A drawback of the current design of LODScape is the lack of a simple user interface to edit the ontology. It is unrealistic to expect ordinary users to edit the ontology with a full-fledged ontology editor such as Protégé. We plan to provide a simple Web-based editor for customizing the LODScape ontology.

Another limitation is the inability to explicitly state predicate ordering preferences in the ontology. Since predicates are owl:ObjectProperty's, they cannot have their own properties. Ordering information is currently encoded only in the imported ontology which is indexed internally by LODScape. We plan to overcome this by making all predicates classes so that they can have their own owl:DatatypeProperty's which can explicitly represent the ordering information for predicates within a group.

Acknowledgments

This work is supported in part by the World Bank/Government of India research grant under the TEQIP programme (subcomponent 1.2.1) to the Centre for Knowledge Analytics and Ontological Engineering (KAnOE) at PES Institute of Technology, Bangalore, India. The authors would also like to acknowledge the work of Mr. Rohan Mahalank in creating the user interface for LODScape.

References

1. Hastrup, T., Cyganiak, R. and Bojars, U. Browsing Linked Open Data with Fenfire, In Proc. LDOW 2008, April 22, 2008, Beijing, China (2008).
2. T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets.
1. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In Proceedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06), (2006).
2. Openlink Data Explorer: http://www.w3.org/wiki/OpenLinkDataExplorer
3. Bizer, C., Gauss, T. Disco – Hyperdata Browser: A simple browser for navigating the Semantic Web, http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/ (2007)
4. Object Viewer: http://projects.semwebcentral.org/projects/objectviewer
5. Zitgist RDF Browser: http://www.w3.org/2001/sw/wiki/Zitgist
6. Marbles: http://dbpedia.org/Marbles
7. Kobilarov, G. and Dickinson, I., Humboldt: Exploring Linked Data, HP Labs, UK (2007)
8. Seeliger, A. and Paulheim, H. A Semantic Browser for Linked Open Data, In Proc. International Semantic Web Conference ISWC (2012).

Appendix: How *LODScape* Meets the Open Track Criteria:

Minimal requirements

1. The application has to be an end-user application, i.e. an application that provides a practical value to general Web users or, if this is not the case, at least to domain experts.
   Yes, LODScape provides a public web site that anyone can visit to search and browse to obtain information derived from multiple LODs in a single interface.

2. The information sources used
   - should be under diverse ownership or control
     LODScape currently uses DbPedia and Freebase and will include other LODs in the near future.
   - should be heterogeneous (syntactically, structurally, and semantically), and

Both DbPedia and Freebase contain facts from a wide range of domains. Further, it is particularly relevant to LODScape that the set of predicates in Freebase is user-created, less clean and more heterogeneous than those in DbPedia.
    - should contain substantial quantities of real world data (i.e. not toy examples).
      LODScape has about 300 million triples taken from DbPedia and Freebase. More LODs will be added in the near future.
3. The meaning of data has to play a central role.
    - Meaning must be represented using Semantic Web technologies.
      All data is represented and processed as RDF triples/quads in addition to the use of the OWL ontology and the Jena semantic engine.
    - Data must be manipulated/processed in interesting ways to derive useful information and
      LODScape shows that modifying the ontology changes the behavior of the browser. Facts are filtered, grouped and ordered based on the ontology.
    - this semantic information processing has to play a central role in achieving things that alternative technologies cannot do as well, or at all;
      Unlike other LOD browsers which merely order the triples alphabetically or show them in a graph structure, LODScape is able to filter, group and order predicates based on their organization in the OWL ontology.

Additional Desirable Features

- The application provides an attractive and functional Web interface (for human users)
  A minimally attractive but fully functional web interface has been provided.
- The application should be scalable (in terms of the amount of data used and in terms of distributed components working together). Ideally, the application should use all data that is currently published on the Semantic Web.
  Additional LODs can be readily added to LODScape by adding the predicates used in the LODs to the ontology.LODScape handles any owl:sameAs relationships among LODs automatically.
- Rigorous evaluations have taken place that demonstrate the benefits of semantic technologies, or validate the results obtained.
  Minimal user evaluations have been done in the limited time available.
- Novelty, in applying semantic technology to a domain or task that have not been considered before
  We believe that modifying the behavior of an LOD browser to personalize or customize it by changing an underlying ontology is a novel idea.
- Functionality is different from or goes beyond pure information retrieval
  LODScape is able to apply semantic filtering, semantic grouping and semantic ordering which are not usually possible through standard information retrieval techniques.
- The application has clear commercial potential and/or large existing user base
  We believe that there is a significant commercial and widespread usage potential for LOD browsers, especially with the semantic capabilities of LODScape.
- Contextual information is used for ratings or rankings: *Not applicable.*
- Multimedia documents are used in some way
  Yes, images of the retrieved resource, if available, is fetched at run-time from Wikimedia and integrated into the resulting page.
- There is a use of dynamic data (e.g. workflows), perhaps in combination with static information
  Current contents of Wikipedia pages including descriptive texts and images are fetched at run-time and integrated into the resulting page.
- The results should be as accurate as possible (e.g. use a ranking of results according to context)
  Not applicable.
- There is support for multiple languages and accessibility on a range of devices
  LODScape is language independent. It is already accessible from any device running a web browser, including smart phones. All of the code and technologies used are Unicode-compliant.