Garden Planner: A Semantic Web Search Application based on Jena and OWL Reasoner

Nibu Jacob
Indiana University
Bloomington
jacobnibu@gmail.com

ABSTRACT

Semantic search provides significant advantages over conventional text search. Widespread adoption of semantic technologies has been hindered by a lack of skills and easy-to-use software tools. A website is user-friendly when the complexities of implementation and underlying architecture are hidden from the viewer. This project is an attempt to create a user-friendly website that provides semantic search capability in finding relevant information for a kitchen garden. It explores a workflow that employs suitable tools and best-practices to build a prototype web application that provides custom garden planning facility using semantic search and reasoning.

Categories and Subject Descriptors

A.1 [**General Literature**]: Introductory and Survey; H.2.3 [**Database Management**]: Languages---query languages; H.4.2 [**Information Systems Applications**]: Types of Systems---decision support; K.6.3 [**Management of Computing and Information Systems**]: Software Management---software selection.

General Terms

Documentation, Design, Experimentation, Standardization, Languages, Theory.

Keywords

Semantic search, OWL reasoning, Jena, Gardening.

1. INTRODUCTION

Increasing volume and complexity of data makes traditional search systems increasingly inadequate to find relevant information. At the current pace of growth in data generated precision search algorithms become critical in all fields of life. The Semantic Web concept was proposed by Berners-Lee in 2001 as the architecture for data on the WWW that enables such precision searches [Berners-Lee, 2001]. Semantic web is an extension of current technologies using hierarchical layers of standards. The different layers are at different levels of maturity and presently the ontology layer is the near mature highest layer [Fazzinga, 2011].

Web Ontology Language (OWL) was developed by W3C as a recommended ontology language [Horrocks, 2003]. Meant to provide a shared and common understanding of a domain, there has been a lot of research and academic work in this field as revealed by usage analyses [Ashraf, 2015]. However, the uptake of OWL on the web has been patchy, even though it has been slowly increasing [Glimm, 2012]. Matching ontologies still remains a challenge that hinders interoperability and data integration [Nacer, 2014]. The performance of semantic web services is worse than similar technologies using Java and other languages. In spite of the current limitations, ontology-based web technology offers the best interoperability and potential for future growth.

Apache Jena is a semantic tool that can store, process, manipulate, and query semantic data [Jena, 2007]. Comparative studies have shown that Jena is one of the most robust and high-performing semantic tools (in a large-scale computing environment) [d'Aquin, 2010]. Jena is based on Java and provides APIs for SPARQL query as well as reasoning among other things.

Simple Protocol and RDF Query Language (SPARQL) is a W3C recommendation for querying RDF data. Performance benchmark tests have shown that SPARQL performs slower than SQL and using wrappers to convert SPARQL to SQL for querying relational database may be a more efficient solution [Sequeda, 2013]. However, SPARQL has high expressive power and is directly compatible with semantic web services [Sbodio, 2010]. Efficient search systems that can query very large datasets [Tablan, 2015] and fetch information from multiple datasets [Nesi, 2014] have been successfully built based on SPARQL.

Semantic search systems differ in their design and features offered. Some of them use crawler-based searching and indexing of data while others use keywords-based searching [Batzios, 2012]. A semantic search system could retrieve only certain types of data, such as, data in XML format or OWL format. Search systems using inductive reasoning and probabilistic ontologies have potential for further research and adoption [Fazzinga, 2011].

The purpose of this project is to build a prototype web application that allows garden enthusiasts to search horticulture data semantically and find relevant information. The search system makes use of an OWL reasoner to infer additional data based on an ontology provided.

2. METHOD

The project involves four major steps: creation of an ontology, annotating data in a suitable semantic format, creating java methods to handle persistence and reasoning, and finally building a web application and deploying it.

2.1 Ontology development

Developing ontologies is a knowledge-intensive and time-consuming process and often requires teamwork and collaboration [Tudorache, 2008]. The best-practice in this regard is to re-use existing ontologies whenever possible [Simperl, 2009]. For the purpose of this project, a simple ontology was created using Protégé tool [Protégé, 2000].

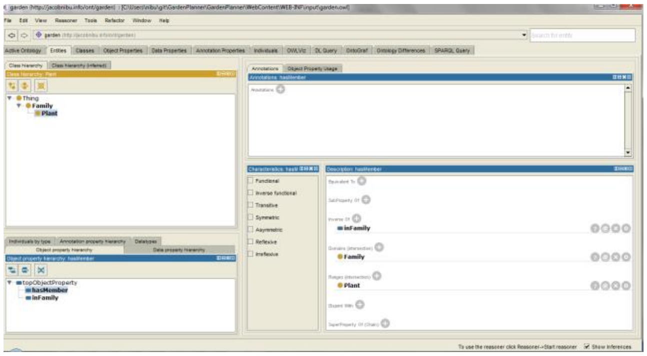

Figure 1. Ontology development in Protégé

The garden ontology consists of a class Family and its subclass Plant (Figure 1). Two object properties link these classes and are inverse of each other. The two classes also have six data properties declared for them to hold hardiness zone values (minimum and maximum), harvest duration values (minimum and maximum), and sun preference values (declared for Family class). The created ontology was saved to disk as an OWL file in RDF/XML syntax.

The Plant class being a subclass of Family class would inherit all the properties of the Family class. Inheritance is a powerful feature of ontology that makes it much more than just a taxonomical system. Moreover, classes in ontology are different from the concept of classes in an object-oriented programming language in that classes in ontology are not inherently connected to any properties.

Protégé allows adding individuals of classes into the ontology itself. Adding data to the ontology is convenient for small projects and may be necessary when the ontology is complex and you need to verify the data against the ontology for correct constraints [Bozzato, 2012]. However, when the data is in a relational database and is constantly being updated, the best practice is to separate data from ontology so that they can be updated and managed more efficiently.

2.2 Semantic annotation of data

Semantic annotation is the process of adding metadata to the data such that the data can be searched semantically rather than just structurally. Fully automatic semantic annotation algorithms have the potential of being scalable to large corpus of data. For semantic annotation of data in relational databases, there is no integrated and user-friendly system available yet [Vavliakis, 2011].

Data about common vegetable plants was obtained from the open data site of United States Department of Agriculture (USDA) and it was annotated using vocabularies in the created ontology. The data was written in RDF/XML format using Notepad++. The data being small, manual annotation was carried out.

2.3 Jena for reasoning with persistent data

Free and open source triple stores like Jena TDB, Jena SDB, Virtuoso, Mulgara, Sesame, etc. offer a way to store RDF triplets data in a persistent storage. Jena SDB is a non-native triple store that can be connected to any relational database. For this project, Jena TDB was chosen as the persistent storage since it has higher performance being a native triple store for use on a single machine [TDB].

Wrote a Java class with methods for loading the ontology and RDF data files from the disk and combining them with an OWL reasoner to create an inference model that can be stored in TDB. Since the Garden Planner needs to use reasoning applied to a relatively stable data, storing the ontology together with data as an inference model was chosen as the best strategy.

2.4 Deploying as a web application

The Garden Planner needs to have a front-end, a webpage, that the users can access over the internet. The webpage would allow a user to provide characteristics of his geographical location, such as hardiness zone and sunlight availability, and his or her desired harvest duration for the plant. Based on these conditions, the search system would retrieve data from the inference model and suggest best plants for the user's garden.

The Java class was modified into a JSP servlet page by extending HttpServlet interface in Apache Tomcat server. A new Dynamic Web Project was created in Eclipse and configured to run on Tomcat server. The JSP file was added to the project as index.jsp homepage. This being a small project, no frameworks like Maven or Ant were used for the project. The entire application logic was written in JSP files.

3. RESULTS

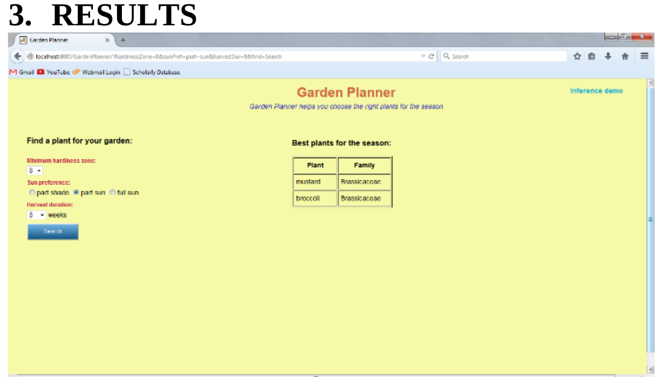

Figure 2. Results from semantic search

Screenshots of the web application opened in a browser window show that the search functionality is implemented successfully (Figure 2) and reasoning is being done to fetch additional inferred data (Figure 3). Codebase of the project is openly available at GitHub [GitHub]. A demonstration of the project is available at YouTube [YouTube].

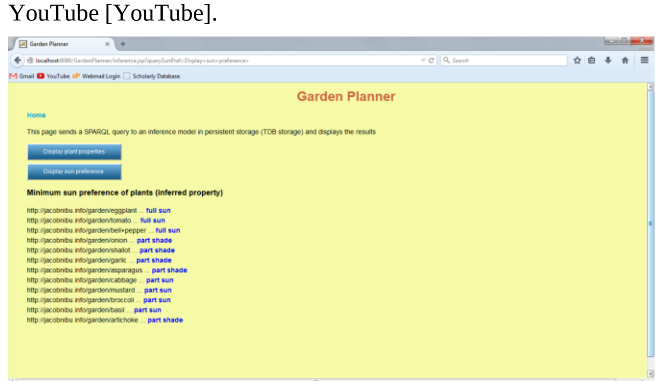

Figure 3. Results from data inferred using OWL reasoner

4. DISCUSSION

The outcome of the project is a prototype web application with a small set of data that has been manually collected and annotated. The domain of gardening has much larger datasets available on the internet, which must be utilized through SPARQL endpoints available. Once connections to other SPARQL endpoints have been established, this Garden Planner can include multiple variables to base the search and become truly useful for users.

There are a few best-practices that could be potentially implemented in this project. Apart from linking to other open datasets in the domain, the architecture of the web application must be changed to an MVC (Model View Controller) architecture so that more dynamic and complex data input and output can be managed. Collection and annotation of data has to be made fully or partially automatic and a suitable PageRank algorithm has to be implemented to display most relevant results to the user.

Connecting the latest version of Jena (version 2.12.1) with MySQL database using SDB storage model was initially attempted but later dropped in favor of TDB which turned out to be much easier to implement. Full implementation in JSP pages was feasible for this project, however, the best-practice would be to confine JSP to the front-end and use Java for the middle layer.

An integrated framework for semantic web application development and general web authoring with annotated data is highly desired for the widespread uptake of semantic technologies by the web community. Currently available tools require technical knowledge about the technologies and are not user-friendly for general users. Most of the SPARQL-based search engines available require users to write SPARQL queries themselves. The display of results also tends to be not user-friendly as full URIs are often displayed and the results are often in a graph format.

5. CONCLUSION

Semantic web search is the next step in the evolution of information search as it offers the best solution for the fast-growing domain of big data in terms of integration and efficient search. More efficient algorithms and user-friendly tools are necessary to achieve this progression. An integrated framework for semantic web authoring and development meant for the general web users can boost the uptake of semantic technologies. Jena can be used to deploy a web application with semantic search and inference capability. Such a search system in the domain of gardening can be used by garden enthusiasts as a very user-friendly tool to dig relevant information from the large corpus of horticulture data available.

REFERENCES

[1] Ashraf, J., Chang, E., Hussain, O. K., and Hussain, F. K. 2015. Ontology usage analysis in the ontology lifecycle: A state-of-the-art review. *Knowl.-Based Syst*., 80, 34-47.

[2] Batzios, A. and Mitkas, P. A. 2012. WebOWL: A Semantic Web search engine development experiment. *Expert Syst. Appl*., 39, 5052-5060.

[3] Berners-Lee, T., Hendler, J., and Lassila, O. 2001. The Semantic *Web. Scientific American* (2001).

[4] Bozzato, L., Braghin, S., and Trombetta, A. 2012. A Method and Guidelines for the Cooperation of Ontologies and Relational Databases in Semantic Web Applications. *Proceedings of the 2nd International Workshop on Semantic Digital Archives* (SDA 2012).

[5] d'Aquin, M., Nikolov, A., and Motta, E. 2010. How Much Semantic Data on Small Devices? In P. Cimiano & H. S. Pinto (eds.), EKAW (p./pp. 565-575), Springer. ISBN: 978-3-642-16437-8

[6] Fazzinga, B., Gianforme, G., Gottlob, G., and Lukasiewicz, T. 2011. Semantic Web search based on ontological conjunctive queries. *J. Web Sem.*, 9, 453-473.

[7] GitHub. Repository available at https://github.com/jacobnibu/GardenPlanner

[8] Glimm, B., Hogan, A., Krötzsch, M., and Polleres, A. 2012. OWL: Yet to arrive on the Web of Data? (cite arxiv:1202.0984)

[9] Horrocks, I., Patel-Schneider, P. F., and van Harmelen, F. 2003. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1.

[10] Jena, A. 2007. Semantic web framework for Java (http://jena.apache.org/).

[11] Nacer, H. and Aissani, D. 2014. Semantic Webservices: Standards, applications, challenges and solutions. *Journal of Network and Computer Applications*, 44, 134–151.

[12] Nesi, P., Bellini, P., and Venturi, A. 2014. Linked Open Graph: browsing multiple SPARQL entry points to build your own LOD views. In E. Jungert (ed.), DMS, 94-103.

[13] Protégé. 2000. The Protege Ontology Editor and Knowledge Acqisition System http://protege.stanford.edu/

[14] Sbodio, M. L., Martin, D., and Moulin, C. 2010. Discovering Semantic Web services using SPARQL and intelligent agents. *J. Web Sem*., 8, 310-328.

[15] Sequeda, J. and Miranker, D. P. 2013. Ultrawrap: SPARQL execution on relational data. *J. Web Sem*., 22, 19-39.

[16] Simperl, E. P. B. 2009. Reusing ontologies on the Semantic Web: A feasibility study. *Data Knowl. Eng*., 68, 905-925.

[17] Tablan, V., Bontcheva, K., Roberts, I., and Cunningham, H. 2015. Mímir: An open-source semantic search framework for interactive information seeking and discovery. *Web Semantics: Science, Services and Agents on the World Wide Web*, 30, 52-68.

[18] TDB. Documentation available at https://jena.apache.org/documentation/tdb/

[19] Tudorache T., Vendetti J., and Noy N. F. 2008. Web-protégé: a lightweight OWL ontology editor for the web. In: *Proceedings of 5th OWL Experiences and Directions Workshop* (OWLED 2008).

[20] Vavliakis, K. N., Symeonidis, A. L., Karagiannis, G. T., Mitkas, P. A., and Mitkas, P. A. 2011. An integrated framework for enhancing the semantic transformation, editing and querying of relational databases. *Expert Systems with Applications*, 38, 3844–3856.

[21] YouTube. Demonstration video available at https://youtu.be/-MYM3KsIyOs