THE MANCHESTER SUSHI FINDER

Progress Report
COMP60990 Research Methods and Professional Skills
(June 6, 2014)

Name: Al Abbas, Hani Ahmed
Project Supervisor: Sean Bechhofer

School of Computer Science
The University of Manchester

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT:

1 INTRODUCTION:

As the trend nowadays to try making machines more intelligent, sharing knowledge of information instead of sharing the raw information in the web is becoming more desirable. Ontologies are considered the main pillar of the semantic web [1]. Semantic web is all about sharing knowledge ,that is understandable for machines, on the web [1]. Creating ontologies is a way to capture our knowledge of the world [1].

Computers do not understand information stored on the web as xml and html. They are just codes to the machine and they display it to users regardless of what knowledge needed. So, ontologies came along for machines to make sense of information. Ontologies are a way to represent knowledge and make inferences from that knowledge using machines computational capabilities and some description logic.

An intelligent way of representing knowledge needs an intelligent way of browsing it. There are a lot of intelligent browsers that is ontology driven user interfaces. Browsing and constructing queries through such user interfaces would be easy and save time. This is because that the user interface guides the user to construct the wanted query with no worries about previous knowledge of the domain. Ideas like manuals and the help in the menu bar of a user interface would same absolute comparing to the self-guided user interface.

Additional technique to make the user interface smart is to use faceted browsing. The idea behind faceted browsing is to personalize the search and get more specific results by suggesting some filters. Faceted browsing is very related to ontology driven interface since both provide some information about the query while been constructed [2].

There are systems, that are ontology driven user interface, exist such as Transparent Access to Multiple Bioinformatics Information Sources (TAMBIS) and SEmantic Webs and AgentS in Integrated Economies (SEWASIE). TAMBIS is system that gather and analysis bioinformatics information from different sources through one

interactive user interface [3]. While SEWASIE meant to access multiple sources of data and help user through constructing exact query needed [4].

The idea of ontology driven interface is not new. In this project, will try to build an application that has some functionalities of interactive interface.

1.1 Project Aims And Objectives:

The aim of this project is to investigate and demonstrate the benefits of using OWL ontologies within ontology driven interface [5]. This will be shown be implementing a configurable and flexible application, so that most configuration will lay in the ontology file itself and could browse other ontologies that contain some specific configuration [5].

To achieve this aim, project was divided into number of objectives. The objectives of this project are:

1. Research ontology driven interfaces.
2. Research faceted browsing.
3. Research OWL API.
4. Look at existing applications that are similar in the basic functionalities.
5. Gather project requirements.
6. Develop a project plan.
7. Design user stories and tests evaluations.
8. Determine the tools that will be used in the project.
9. Start implementing and divide and add functionalities one at a time.

1.2 Project Scope:

This project is going to demonstrate the use of ontologies using an interactive user interface. The functionalities that will be included in the interface are:

1. The user interface will be built using java.
2. User interface will be configurable for generic use.
3. Flexible query building.
4. Preferences of the user will be saved as configuration for reusability.
5. The sushi finder will be a desktop application.

6. The sushi finder should work for any ontology with standard annotations defined in them (Configuration will be stored in the ontology).
7. User can load ontologies one at a time.
8. User can decides which thing to query about (Sushi-Sushi dishes).
9. User can query for specific sushi or sushi dish type based on wanted or unwanted ingredients.

1.3 Report Structure:

2 BACKGROUND:

2.1 OWL API:

Application Programming Interface (API) is a set of protocols that make sure the software components interact with each other in the right way [6]. The job of OWL API is to communicate between the Web Ontology Language (OWL) and any other program.

2.1.1 OWL

OWL is a semantic web language to represent things about the world, group of things, and the relations between them [7]. It is a way to represent knowledge such that it is a representation of the world and our knowledge of it and it is accessible to programs and can be used [8]. OWL is a W3C recommendation since 2004, and then OWL 2 was published in 2009, followed with a second edition in 2012 [7, 9]. OWL 2 is just an extension and revision of the original OWL publish in 2004 [7]. OWL has several defined syntaxes including Functional Syntax, RDF/XML, OWL/XML and the Manchester OWL Syntax [7, 9].

OWL documents are called ontologies [7]. These ontologies can be put into the web or into a local computer depending on the need. The advantage of ontologies in the web is that they can be referenced from or reference to other ontologies [7]. Ontologies station in a local computer used local in the same level of the local machine.

2.1.2 OWL API

OWL API is an Application Programming Interface for the purpose of specifying how to interact with OWL Ontologies [9]. OWL ontologies can be created, manipulated, and reasoned over using OWL API [9]. It has been available since almost the same time of OWL [9]. OWL API went through several revisions following the development of OWL [9]. OWL API has the ability to parse and serialize OWL ontologies to different syntaxes such as Functional Syntax, RDF/XML, OWL/XML and the Manchester OWL Syntax [9].

OWL API comes with free java implementation that takes out the burden of parsing and serializing OWL ontologies from the developers back [9]. OWL API comes also with loading and saving ontologies capabilities [9].

OWL ontologies being accessed using OWL API only through OntologyManager interface [9]. OntologyManager interface manage all changes in ontology as seen in Figure 1 below [9].

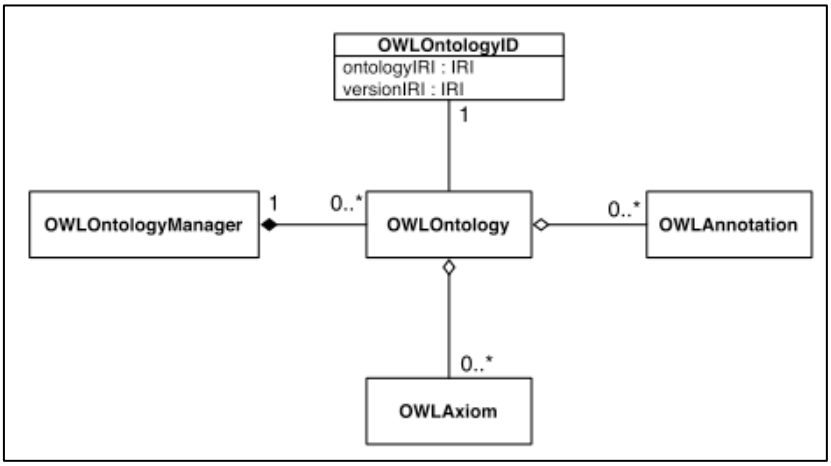

Figure 2.1.2-1: UML diagram showing the management of ontologies using OWL API [9]

Inference is applied on the OWL ontologies using OWLReasoner interface [9]. This interface provides some useful check like consistency, checking computation of class and axiom entailments [9]. Since the reasoning functionality is separate, developers either can use the available or can provide their own implementation [9]. There are some exist implementations of reasoners such as FaCT++, HermiT, and Pellet [9].

As for Query using OWL API, it does not offer any query mechanism [9]. However, it provides some sort of basic querying which is based on entailment checking functionality [9].

2.2 Information Retrieval

Information retrieval is the method in which some information is retrieved from a source or multiple sources contains needed data. Every retrieval system needs some mechanisms to retrieval relevant needed information. Nowadays, there is more information probably resides on the cloud than it was in the recent years. As the amount of data on the web grows dramatically, the need to find and retrieve relevant

information becomes more important. Getting the wanted results is becoming more problematic because of the amount of the data on the web and the technique used. Usually, in a search engine the results range between relevant and irrelevant. It would be nice for a user to query for something and get the most relevant result that meets his/her needs.

There are different methods of retrieving data from the web. Keyword-based search method could be the oldest among the others.

2.2.1 Keyword-Based Information Retrieval

Keyword-based search method use specific words call "Keyword" that are linked with database records [10]. Keyword-based search method would be the default and the usual choice to use in search, since it has been used over long time. This method seems easy to use, as it is resemble natural language which could be understandable by humans but not by machines [10]. Because of the human factor that exists in writing the search query, things inevitably could go wrong. Simplicity and ease come with a cost, keyword-based search method suffers from some serious issues [10]. (1) One of these issues is the lack of accuracy and recall because of all of the synonyms and the homonyms which are based on memorizing terms rather than concepts [10]. (2) Another major issue is that using keyword-based search add more ambiguity, when the user want just to browse around to find out what is there or the user does not know the right term used in specific content [10].

According to [10], there are solutions for both issues. The lack of precision and recall can be treated by ontology-based information retrieval method [10]. The growth in the ambiguity issue, would be solve be using multi-faceted search method which would guide the user during constructing the search query [10]. So, the use of knowledge base and concept base would be more desirable than just providing arbitrary information. In addition, a sense of Artificial Intelligence is also felt since machines can make inferences based on some rules.

Ontogator is a system that combines the two methods ontology and multi-faceted based search [10].

2.3 Ontology-Based User Interface

In general, developing user interfaces hindered by the knowledge of the user [11]. Letting the user kwon what can he ask for and constructing a meaningful search query using the user interface is the major issue [11]. To remedy this issue, several solutions have been proposed. Some of those solutions would be making every option in the user interface available to reduce the recall issue. Another solution would be writing manuals to the user to follow. These solutions might be providing more complexity and other problem. The former solution could overwhelm the user with all of the options available whether needed options or not. The latter solution could increase the load on the user to study and spend time on something that needed to be recalled eventually. Ontology driven user interface would be the most suited solution, since ontologies are based on conceptual model rather than just terms [10, 11]. This conceptual model gives a map for the user to follow upon constructing queries [11].

Ontology-based user interface is a user interface that allows the user to construct and manipulate queries based on some concept domain stored in ontology [4, 11]. The concept domain drives the user interface [11]. There is no need for manuals or shove all available options in the user interface, since the ontology based one should act as a guide for the user [4, 11]. It depends on recognizing knowledge instead of memorizing keywords [11]. It allows the user to build a complex and meaningful queries and return the needed results [11]. In addition, it offers the user the option of browsing around to find out what he/she needs [11].

The user interface offers choices and some scenarios for the user, so that the user would be guided toward constructing meaningful queries that return the intended results [4, 11]. Query expressions are Description Logics (DLs) expressions and they are incremental and compositional [4, 11]. Users would not face the no-result status after running queries. DL is a way for knowledge representation used by the conceptual model [12]. DL model is not easy because of the need knowledge about the DL syntax along with understanding it, so a friendly user interface need to build to separate the user from dealing with DL [11]. DL provide hierarchal model based on

conceptual model that represent classes of specific domain and the relationships between the instances of those classes [4, 11].

There are two kind of concepts that the model support [11].

-Metadata
-Annotations

2.4 Faceted-Based Search

Faceted search is an intelligent and efficient retrieval mechanism that allows the users to filter a collections of information based on some dimensions which called Facets [13, 14]. This information ordered based on multiple taxonomies, which are called faceted classification [13, 15].

This technique can be applied in two ways, either unidirectional or bidirectional. In unidirectional way, either applies it beforehand on a collection of selection that is browsed to construct the query or on the result of a query so it can be refined more. In the bidirectional way, it is to apply it both beforehand and afterward. Both serve the same purpose which is to personalized the search and make it easy to suit the user's needs.

2.5 Ontology Visual Querying

Ontology Visual Querying is the use of the user interface and ontologies to guide for interactive query building and provide meaningful queries in intelligent way [2, 4]. In addition, the user interface help in constructing valid and exact queries [2, 4].

Visual querying is not new. It has been there since almost the beginning of textual query languages [2]. Almost all visual querying languages have two features in common [2]. The two features are: (1) a model to represent the stated query and (2) a way to of constructing the query [2]. Since visual querying languages invented to query from a data structure, it is only natural for its evolution to follow the development of data structure [2, 16]. A simple example of visual querying would in Microsoft Access.

A major benefit from ontology visual querying is user does not have to remember or know the vocabulary, since user can survey the domain [2, 4]. As a result, forming queries for naïve users becomes easier [2]. In ontologies, new concept can be defined either directly like defining class or indirectly like making inference of something. Therefore, creating query is the same as creating new concept such as the TAMBIS system and SEWASIE system [2-4]. Another advantage would be helping users, who not experienced with the system, to create satisfiable queries according to the constraints [2].

3 RESEARCH METHODS:

3.1 Project Plan:

3.2 Project Deliverables:

3.3 Project Evaluation Plan:

3.4 Project Tools:

4 PROGRESS:

4.1 Users Stories:

4.2 Acceptance Tests:

4.3 Prototype:

5 REFERENCES:

1. Ding, L., et al., *Using Ontologies in the Semantic Web: A Survey*, in *Ontologies*. 2007, Springer US. p. 79-113.
2. Bechhofer, S. and N.W. Paton, *Ontology Visual Querying*, in *Encyclopedia of Database Systems*. 2009, Springer.
3. Stevens, R., et al., *TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources.* IBM System Journal. **40**(2): p. 532-551.
4. Catarci, T., et al., *An Ontology Based Visual Tool for Query Formulation Support.* ECAI, 2004: p. 308-312.
5. Bechhofer, S. *The Manchester Sushi Finder - Project Page*. 2014 [cited 2014 March 5, 2014]; Available from: http://studentnet.cs.manchester.ac.uk/pgt/2013/COMP60990/project/projectbookdetails.php?projectid=20889.
6. Wikipedia. *Application programming interface*. [cited 2014 April 21, 2014]; Available from: http://en.wikipedia.org/wiki/Application_programming_interface.
7. Group, W.C.O.W. *Web Ontology Language (OWL)*. 2012 [cited 2014 April 19, 2014]; Available from: http://www.w3.org/2001/sw/wiki/OWL.
8. Davis, R., H. Shrobe, and P. Szolovits, *What is a Knowledge Representation?*, in *AI Magazine*. 1993. p. 17-33.
9. M, H. and B. S, *The OWL API: A Java API for OWL ontologies.* Semantic Web, 2011. **2**(Number 1 / 2011): p. 11-21.
10. Hyvönen, E., S. Saarela, and K. Viljanen, *Application of ontology techniques to view-based semantic search and browsing*, in *The Semantic Web: Research and Applications*. 2004, Springer. p. 92-106.
11. Bechhofer, S., et al., *Guiding the User: An Ontology Driven Interface.*
12. Bechhofer, S. and C. Goble, *Classification Based Navigation and Retrieval for Picture Archives*, in *Database Semantics*. 1999, Springer. p. 291-310.
13. Wikipedia. *Faceted search*. [cited 2014 April 23, 2014]; Available from: http://en.wikipedia.org/wiki/Faceted_search.
14. Smith, D.A. and N.R. Shadbolt, *FacetOntology: Expressive Descriptions of Facets in the Semantic Web*, in *Semantic Technology*. 2013, Springer. p. 223-238.
15. Wikipedia. *Faceted classification*. [cited 2014 April 23, 2014]; Available from: http://en.wikipedia.org/wiki/Faceted_classification.
16. Catarci, T., et al., *Visual query systems for databases: A survey.* Journal of Visual Languages and Computing, 1997. **8**: p. 215-260.