# Development of a Linked Open Vocabulary for a Feedback Survey

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

in Computer Science

University of Regina

By

Parita Dipakkumar Amin

Regina, Saskatchewan

August, 2022

## Abstract

The World Wide Web (WWW), as defined by the World Wide Web Consortium (W3C) at `https://www.w3.org/WWW/`, is "the universe of network-accessible information, the embodiment of human knowledge". With development over the years, the global web of linked documents has become the global web of linked data, which is popularly known as the Semantic Web.

The Semantic Web is now an established topic of research but there remain many unanswered questions. Linked Open Data (LOD), as the highest standard for linked data, is a worthy goal for researchers and data providers who seek to contribute to the development of the Semantic Web. Data that are linked and open facilitate discovery and reuse. Likewise, vocabularies used to describe the semantic relationships amongst data that are linked and open facilitate discovery and reuse. Yet, it is not always a straightforward matter to take data that may be somehow accessible on a webpage written in Hyper Text Markup Language (HTML) and publish it as LOD. Many complimentary technologies are used in handling LOD on the web, particularly the Resource Description Framework (RDF), the Web Ontology Language (OWL), and the Simple Protocol and RDF Query Language (SPARQL).


last paragraph news some attention – presently commented out

## Acknowledgements

I would like to take an opportunity to thank all the people who have supported me during my research period in some or the other way.

I would like to express my soulful gratitude to my supervisor Dr. Daryl H. Hepting for his valuable time, guidance, suggestions and comments throughout the course of the research. He was always open to all my questions and queries related to my research work as well as while writing this thesis. He provided me with the space required to make this study as my own work, at the same time he made sure that I am leading on the right path by giving constant feedback and suggestions whenever I was lost or stuck. I would also like to thank the Department of Computer Science and Faculty of Graduate Studies and Research for their academic as well as administrative support as and when required.

Lastly, I am grateful to my parents for always being there for me regardless of the distance and time-difference between the two different parts of the world. This would have been next to impossible for me if not for them. I would like to specially thank my brother Shivam for being the family here and for encouraging and motivating me during the times I was low. My special thanks to all my friends for all their support.

## Post-Defense Acknowledgements

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| API | Application Programming Interface p. 25 |
| CERN | Conseil Européen pour la Recherche Nucléaire p. 1 |
| CSV | Comma Separated Values p. 23 |
| DL | Description Logics p. 39 |
| ER | Entity Relationship p. 12 |
| GIF | Graphics Interchange Format p. 62 |
| GREL | Google Refine Expression Language p. 24 |
| GUI | Graphical User Interface p. 24 |
| HTML | Hyper Text Markup Language p. i |
| HTTP | Hyper Text Transfer Protocol p. 2 |
| HTTPS | Hypertext Transfer Protocol Secure p. 2 |
| IDE | Integrated Development Environment p. 38 |
| IRI | Internationalized Resource Identifier p. 46 |
| ISO | International Organization for Standardization p. 43 |
| JPEG | Joint Photographic Experts Group p. 62 |
| JSON | JavaScript Object Notation p. 64 |
| LOD | Linked Open Data p. i |
| N3 | Notation3 p. 14 |
| OBDA | Ontology-Based Data Access p. 39 |
| OWL | Web Ontology Language p. i |
| PNG | Portable Network Graphics p. 62 |
| RDF | Resource Description Framework p. i |
| RDFS | Resource Description Framework Schema p. 14 |
| SPARQL | Simple Protocol and RDF Query Language p. i |
| SVG | Scalable Vector Graphics p. 64 |
| Turtle | Terse RDF Triple Language p. 14 |
| URI | Uniform Resource Identifiers p. 2 |
| URL | Uniform Resource Location p. 21 |
| VOWL | Visual Web Ontology Language p. 63 |
| W3C | World Wide Web Consortium p. i |
| WWW | World Wide Web p. i |
| XML | eXtensible Markup Language p. 8 |

# Chapter 1

# Introduction

Tim Berners-Lee is the father of the WWW. He wrote *Information Management: A Proposal*[1] in 1989 and 1990 to present the need for a "global hypertext system" at the Conseil Européen pour la Recherche Nucléaire (CERN). In his conclusion to that document, he wrote:

> "we should work toward a universal linked information system, in which generality and portability are more important than fancy graphics techniques and complex extra facilities. The aim would be to allow a place to be found for any information or reference which one felt was important and a way of finding it afterward. The result should be sufficiently attractive to use, that is, the information contained would grow past a critical threshold so that the usefulness of the scheme would, in turn, encourage its increased use."

In the more than 30 years that have followed that initial proposal, a great deal of progress towards Berners-Lee's vision has been realized. The original "web of documents" is becoming the "web of data" or the "semantic web". Is this next web sufficiently attractive to use? This thesis will examine this question in the context of an example.

[1] `https://www.w3.org/History/1989/proposal.html`, dated March 1989, May 1990

## 1.1 Motivation

In his teaching practise, my supervisor collects feedback from students in his classes each semester and makes the responses available on his website[2]. Typically, student responses were collected on paper and then entered manually into computer-readable form. This was not done during the COVID-19 pandemic.

Although it is valuable to have this data available on the web, it is not published as LOD. The motivation for the research presented here was to examine the process by which existing data can be published as LOD and to consider whether that process is "sufficiently attractive to use".

The published linked (structured) data which is available on the web can be efficiently retrieved and reused as, according to Ngomo et al. [?], linked data is "a set of best practices for publishing and connecting structured data on the web". After the introduction of the semantic web and the four principles of linked data for publishing and connecting structured datasets on the web by Berners-Lee, the web of documents has been gradually moving towards the development of the web of data. With time, these principles have been accepted and considered as the standard to publish structured data on the semantic web [?, ?, ?, ?]. The four principles of linked data by Berners-Lee [?] are as follows:

- Use Uniform Resource Identifiers (URI) as names for things
- Use Hyper Text Transfer Protocol (HTTP)[3] URIs to look up those names
- Provide useful information, using the standards (RDF[4], SPARQL)
- Include links to other URIs to discover more things

[2] `http://www2.cs.uregina.ca/ hepting/teaching/evaluation.html`
[3] Both HTTP and Hypertext Transfer Protocol Secure (HTTPS)
[4] includes all the standards belonging to RDF family

After the development of Linked Data for the semantic web, Berners-Lee also contributed to web engineering by introducing a 5-star rating[5] system for the rating of linked data implementations. In this system, using the linked data principles, the publishers of the data on the web link their data to the existing linked data to provide reference and make it available for lookup. The data on the web, according to the 5-star system, is accepted to be 5-star data if the data:

- is structured (machine-readable)
- has open-license for access
- is described and linked to other data using URIs

A query[6] is used to retrieve information from a database that consists of multiple tables. According to the W3C recommendation, a semantic (web) query depicts the techniques and protocols that are "used to retrieve information from the web of data" using computer programs. Web (semantic) queries use the semantics and the constitution of data on the web to perform data retrieval operations. Various semantic web technologies have been developed for data retrieval on the web. SPARQL is one of the semantic web technologies used for web queries, where it gathers all the data from different domains and treats all those datasets as local.

## 1.2 Contribution

## 1.3 Organization

Following this introduction, the rest of the thesis is organized as follows. Chapter 2... Chapter 3... Chapter 4... Finally, Chapter 5 presents some conclusions and

[5] https://www.w3.org/2011/gld/wiki/5_Star_Linked_Data
[6] https://www.techopedia.com/definition/5736/query

opportunities for future work.

# Chapter 2

# Related Work

This chapter includes the theoretical and conceptual information that builds the foundation for this thesis. Broadly, this chapter covers the semantic web and linked open data from the two perspectives of technology and usability.

## 2.1 The Web

The idea of the Web, as per Berners-Lee [?], was derived from a positive experience of a small "home-brew", a personal hypertext system. This "home-brew" system was used to keep track of personal information on a distributed project. The Web can be used independently for multiple projects that can be later integrated with unique relationships among them using the newly shaped information to represent the new state of linked knowledge. The web was first designed by Berners-Lee in 1989[1] while working at the CERN. The web, today, has advanced more, evolved, and progressed to develop multiple versions. Different versions of the Web were designed and declared as generations like Web 2.0, Web 3.0, and Web 4.0. Web 2.0 has been accepted and included for a long time now, Web 3.0 is comparatively new, whereas Web 4.0 is at the initial stage of research and application. An idea of different versions of the web can

[1] https://www.w3.org/History/1989/proposal.html

be derived from Figure 2.1.

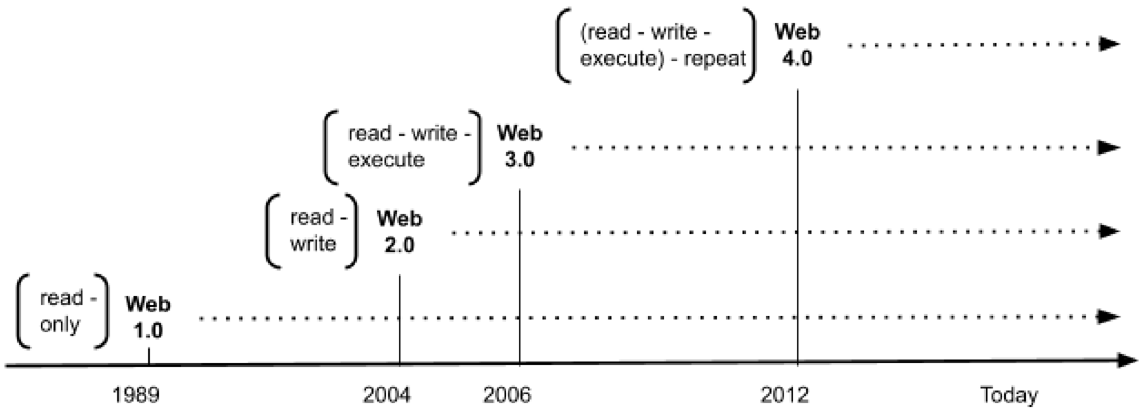



Figure 2.1: Different Versions of the Web, according to Aghaei et al. [?]

The first version of the Web designed in 1989, as per Aghaei et al. [?], was read-only. The users of the web browsers for Web 1.0 could only read the information present on the web pages. The right to modify the contents of the web page was only with the website owners. In 2004, Dale Dougherty at O'Reilly Media [?] formally designed an improved version of Web 1.0 and called it Web 2.0 where the users had the liberty to read as well as modify the data on the web. The web browsers for this generation of the web offered comparatively more features than Web 1.0, making the websites more efficient and engaging. Features like web designing, creation, and modification also helped Web 2.0 in the introduction of blogs, vlogs, wikis[2] and social networking technologies [?].

## 2.2 Semantic Web

In 2006, John Markoff [?] introduced the new read-write-execute web as Web 3.0 in New York Times. Structured datasets and the links between them are used for the successful working of Web 3.0, also known as the Semantic Web. The web pages for the

[2]www.wiki.org/wiki.cgi?WhatIsWiki

semantic web can be written using HTML and RDF as it is the web of data, contrary to the web of documents, where the web pages were written using HTML and the links were described using hyperlinks. In the web of data, arbitrary things can be expressed using links to represent the relationship between two entities, as shown in Figure 2.2.

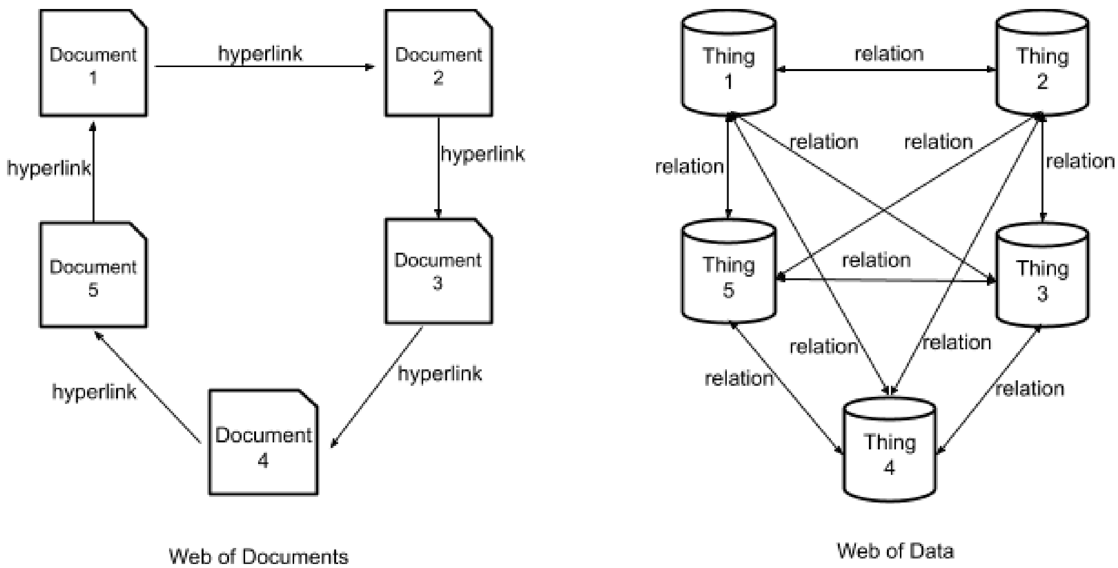



Figure 2.2: The Web to Semantic Web, according to Aghaei et al. [?]

The latest generation of web that is still at its research and development stage is Web 4.0 which is supposed to be read-write-execute with the property of concurrency [?]. This version of the web is intended to work with artificial intelligence to give users an exceptional quality of experience. While describing how Web 4.0 would be, Berners-Lee [?] stated “if HTML and the Web made all the online documents look like one huge book, RDF, schema, and inference languages will make all the data in the world look like one huge database”.

### 2.2.1 The Semantic Web Stack

The Semantic Web Stack [3] is also known as Semantic Web Cake or Semantic Web Layer Cake. The stack represents the architecture of semantic web. Figure 2.3[4] shows how semantic web principles are implemented in the layers of technologies.

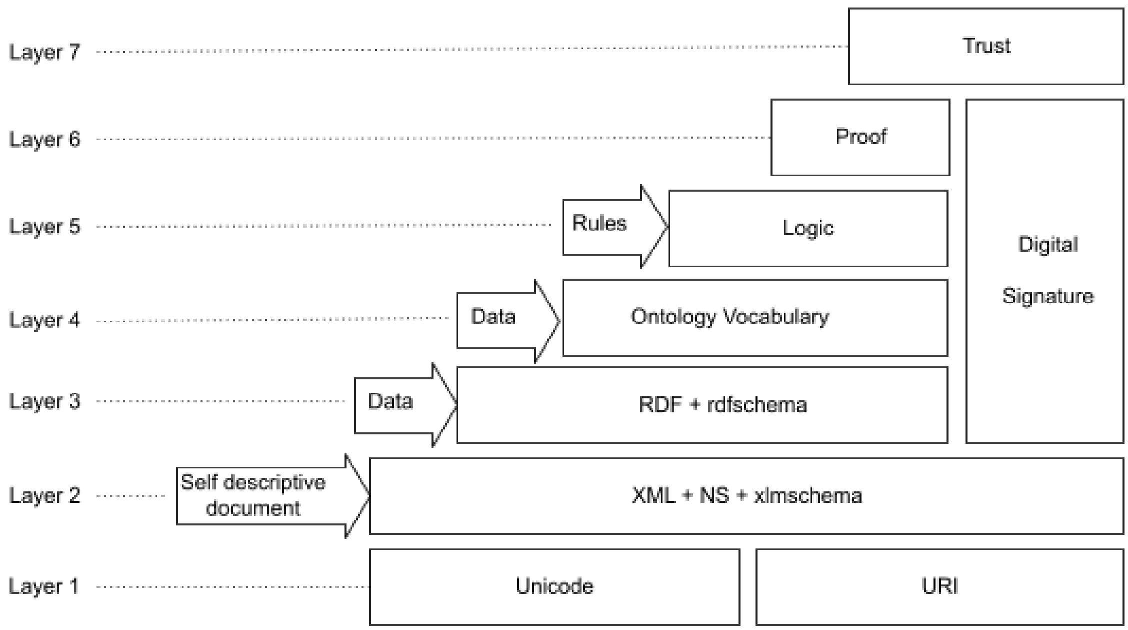



Figure 2.3: Semantic Web Stack

The layers in the stack consist of different technologies. Going from the lowest level to the top level, the technologies are Unicode and URI; eXtensible Markup Language (XML), NS and xmlschema; RDF and rdfschema; Ontology vocabulary; Logic; Proof; and Trust. Digital signatures are used as security at different levels. The layers belong to three different groups, namely, self-descriptive document, data and rules.

[3] https://en.wikipedia.org/wiki/Semantic_Web_Stack
[4] https://www.w3.org/2001/12/semweb-fin/w3csw

## 2.3 Linked Open Data

According to Berners-Lee et al. [?], "Semantic web is not limited to publication of data on the web; it is about making links to connect related data." The concept of Linked Open Data[5] is an amalgamation of two technologies, Linked Data and Open Data. GraphDB[6] by Ontotext, an RDF database, is an example of LOD. GraphDB promotes knowledge-discovery and efficient data-driven analytics by linking large datasets from distinct sources to Open Data.

### 2.3.1 Linked Data

The relationships shared among the data from different sources on the web are equally important as the access to the data for efficient working of the semantic web. The technique used to manage these datasets and their relationships is referred to as Linked Data. In other words, Linked Data[7] can be termed as a set of various techniques of publishing structured data on the web. The W3C standards and various semantic web technologies handle the distribution and engagement of the structured data across the web to aggregate the data available on the web into one huge database [?].

In 2006, Berners-Lee [?] framed four basic rules (mentioned in section 1.1) to publish and connect the structured datasets across the web. This set of rules was termed as Linked Data principles. A brief description of these principles is as follows:

1. Use URIs as names for things:

   The web identifiers, the (URIs) should be used to give unique names to all the entities (or things) on the web. URI of an entity can be used to understand that the entity in this dataset is the same as the other entity in a disparate dataset.

[5]https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/
[6]https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/
[7]https://www.w3.org/wiki/LinkedData

2. Use HTTP[8] URIs to look up for those names:

   The HTTP/HTTPS protocols allow easy retrieval of resources. Hence, along with URIs, either of the protocols can be used to search things (entities) easily. This promotes the publication of data on the web and its addition to the global data space.

3. Provide useful information, using the standards (RDF[9], SPARQL) to look up for a URI:

   When an entity is searched using the URI associated with it, it is beneficial to use standards like RDF and SPARQL to query the datasets. RDF is a framework used for graphical representation of publication and distribution of data on the web, as described in detail in section 2.4. On the contrary, SPARQL is a query language used for retrieval and manipulation of data (in RDF format) on the web. It also helps with the information on relationships related to the data being searched.

4. Include links to other URIs to discover more thing:

   In the semantic web, links between two entity defines the relationship using the basic entity-relationship model. Using links for URIs of the entities promotes interconnection of the data and search of things on the web. Connecting new data to already existing entities increases the reuse and interlinking within the domain and creates a computer-understandable network.

Publishing data on the web as per linked data principles can benefit the data providers by allowing them to use one global space to add all their data. An RDF data model and URIs are used by the linked data to name the things (entities). The

[8]Both HTTP and HTTPS

[9]includes all the standards belonging to RDF family

linked datasets publish and locate the class-object data which in turn is accessed using the HTTP/HTTPS and maintains the connection among them [?].

A wide network of structured data, from a variety of domains, on the web has been formed obeying the principles of linked data by Berners-Lee. This has resulted in the formation of a huge database of structured linked data on the web which has further led to the concept of the LOD Cloud. The LOD cloud is a set of publicly [?] accessible/available linked datasets.

### 2.3.2 Linked (Open) Data

According to a handbook[10] by the Open Knowledge Foundation, "Open Data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and share-alike". To make open data available to everyone on the web, the data don't need to be interlinked. Important features of open data are:

- Availability and access: the data should be available on the web. The users should be able to access the data easily by downloading them over the internet. It should be easy to access and modify the data.

- Re-use and redistribution: the data on the web must have permissions to re-use and redistribute in combination to other datasets.

- Universal participation: the data should be available to use, re-use, and redistribute for all the users without any discrimination to an individual or a group.

[10] https://opendatahandbook.org/guide/en/what-is-open-data/

## 2.4 Resource Description Framework

The RDF, initially designed as a metadata data model, is a computer-understandable specification written in a particular format. According to a proposal by W3C[11], RDF plays an important role in their "Semantic Web Vision". RDF describes an entity (or resources) and the relationships among them on the web using basic data models like the Entity Relationship (ER) model.

RDF is a W3C[12] specification for describing resources on the web. RDF uses "properties" and "property values" to describe the resource which is identified by web identifiers called URI (section 2.4.1). There are three types of objects in RDF, namely, resources, properties, and statements. Each data object in a dataset on the web is considered a resource and is named using a URI with an optional ID. To search for the resources on the web, the resources must be named using the HTTP/HTTPS URIs. The W3C has defined the "rdf:type" property to represent the "object of one or more classes" [?]. Resources are described by their characteristics, called "property". The object of RDF that is used to combine a resource with a property of its value is a "statement". A statement, in general terms, can be seen as building blocks of a statement in English, which are a subject, an object, and a predicate. An object in a statement can be either a property value, a resource, or a literal. RDF, according to Swick and Lassila [?], can be represented in three ways:

1. Using XML: an RDF/XML document
2. As triples: subject-predicate-object form
3. In graphical form

[11] https://www.w3.org/TR/PR-rdf-syntax/Overview.html
[12] https://www.w3schools.com/xml/xml_rdf.asp

Figure 2.4 shows an example of an RDF/XML[13] document. The resource being described is “https://www.semanticweb.org/pda324/ontologies/survey” and “Survey Details” is the property value. The document consists of various XML tags enclosed between the angular brackets (“<” and “>”). An RDF/XML document starts and ends with <rdf : RDF> and </rdf : RDF> respectively. <rdf : Description> consists of the elements that are responsible for the creation of the statement. Description tag holds an identification for the resource.

```
<rdf : RDF>
  <rdf:Description rdf:about="https://www.semanticweb.org/pda324/ontologies/survey">
    <s:istypeof>Survey Details</s:istypeof>
  </rdf:Description>
</rdf : RDF>
```

Figure 2.4: An Illustration of Code in an RDF/XML Document

Figure 2.5 shows an example of an RDF as a triple. An RDF triple, generally, follows subject-predicate-object format. For the illustration in the figure, the subject is “https://www.semanticweb.org/pda324/ontologies/survey”, predicate is “includes” and the object is “Survey Details”.

**Subject : https://www.semanticweb.org/pda324/ontologies/survey**

**Predicate : includes**

**Object : Survey Details**

Figure 2.5: An Example of an RDF Triple

Figure 2.6 show an RDF in graphical format. Resource is indicated using an oval shape and a rectangle is used to represent a property value. A connector (an arrow) is

[13]RDF written using XML

used to represent a property to show that property (predicate) connects the resource (subject) with the value (object).

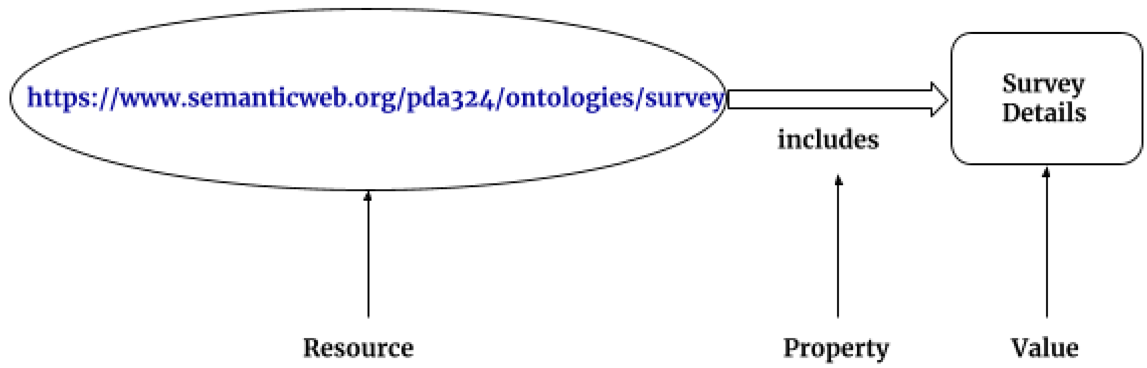



Figure 2.6: Graphical Representation of an RDF

The subject and object of the triples of RDF act as nodes in the graph. In addition to RDF/XML documentation, the RDF data model has a variety of serialization formats such as N-triples, which is a line base format where an RDF triple is used to represent each line [?]; Terse RDF Triple Language (Turtle); Notation3 (N3), which is a subset of Turtle and N-Quads, which is an extension of N-Triples (line based) and adds a graph label to each line that allows multiple graph encoding [?].

Resource Description Framework Schema (RDFS), according to McBride [?], was introduced by the W3C as a technique to define resource types and property names. The data publishers can use RDFS to define various vocabularies in an RDF model. Some of the commonly used RDFS classes and properties[14] along with their description are listed in Table 2.1 [?].

Figure 2.7 shows a graph for the representation of RDFS for a comparatively complex and abstract example based on RDF Primer [?]. The graph has been drawn for the following statement:

*Romeo and Juliet is written by William Shakespeare.*

[14]https://www.w3.org/TR/rdf-schema/

| Class/Property | Description |
|---|---|
| rdfs:Resource | All things described by RDF are called resources |
| rdfs:Class | The class of resources that are RDF classes |
| rdf:Property | The class of RDF properties. It is an instance of rdfs:Class |
| rdfs:Literal | The class of literal values like integers and strings |
| rdfs:Datatype | The class of datatypes. All the objects of this class correspond to the RDF model of datatype |
| rdf:HTML | The class of HTML literal values and subclass of rdfs:Literal |
| rdfs:range | An instance of rdf:Property that is used to state that the values of a property are instances of one or more classes |
| rdfs:domain | An instance of rdf:Property that is used to state that any resource that has a given property is an instance of one or more classes. |
| rdf:type | An instance of rdf:Property that is used to state that a resource is an instance of a class |
| rdfs:subClassOf | An instance of rdf:Property that is used to state that all the instances of one class are instances of another |
| rdfs:label | an instance of rdf:Property that may be used to provide a human-readable version of a resource's name |
| rdfs:subPropertyOf | An instance of rdf:Property that is used to state that all resources related by one property are also related by another |

Table 2.1: Commonly used Classes and Properties of RDFS

Figure 2.7 is the graphical representation of an RDF with schema as well as data. In the graph, the rdf:type of "Romeo and Juliet" is "Play" which in turn is an rdf:subClassOf "Literature". Also, the rdf:type of "William Shakespeare" is "Person" which, further, is an rdfs:range of "is written by" of rdfs:domain "Literature".

The actual RDF looks quite different from the previous example (Figures 2.4, 2.5, 2.6 and 2.7). For the representation of an actual RDF, consider Figure 2.8 for the corresponding English statement. "http://www.semanticweb.org/survey.html" is a subject URI with respondent and Robert James having URIs as "http://xyz.org/1/respondent" and "http://www.semanticweb.org/responseID/1234" respectively. The example represents a survey system which has responses as well as respondents and related details as a part of the environment.

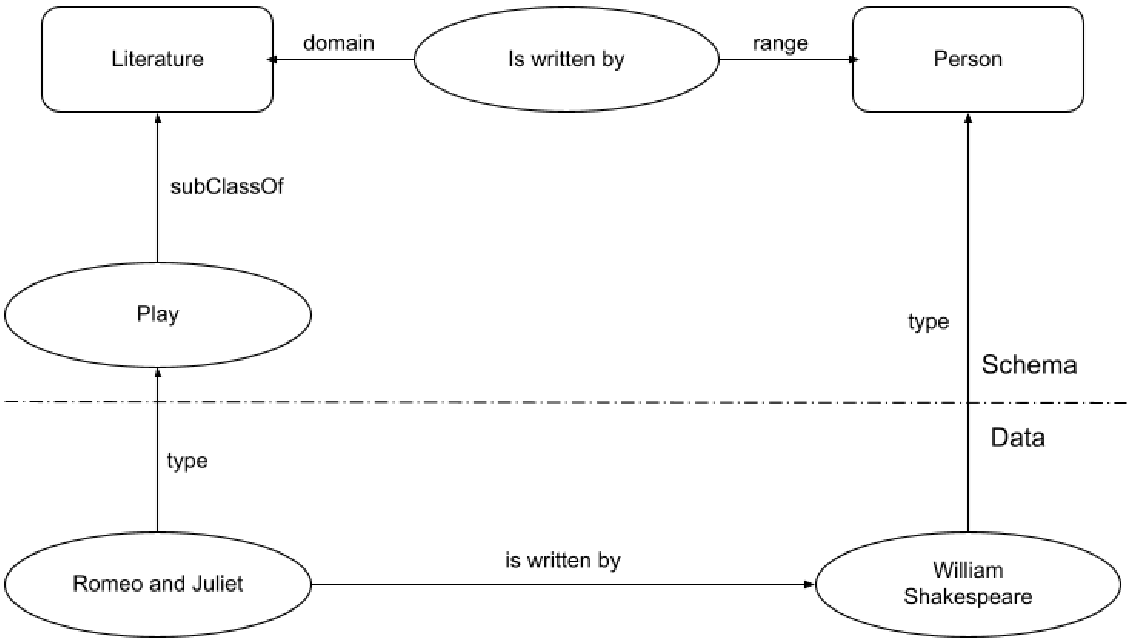



Figure 2.7: Graphical Representation of an RDFS for a Comparatively Complex Illustration, after McBride [?]

**http://www.semanticweb.org/survey.html** has a **respondent** whose value is **Robert James**

Figure 2.8: English Statement for an Example of Actual RDF

In the form of RDF triple, the statement in Figure 2.8 can be represented as shown in Figure 2.9. URI is assigned to the subject survey.html and to the object which is the response with responseID 1234 which corresponds to the property value Robert James. A URI is also assigned to the predicate (respondent).

**SUBJECT - http://www.semanticweb.org/survey.html**
**PREDICATE - http://xyz.org/1/respondent**
**OBJECT - http://www.semanticweb.org/responseID/1234**

Figure 2.9: RDF Triple for English Statement in Figure 2.8, after RDF Primer [?]

Figure 2.10 shows a graphical representation of the RDF triple in Figure 2.9. The subject and the object are two entities and the predicate being the third entity acts as a connector between them. Considering an entity-relationship data model, the predicate can be treated as a relationship shared by the subject and the object. There can be more details added to the system, that is, more statements can be added which are in relation to the one in Figure 2.8. Suppose the new statement added is:

*http://www.semanticweb.org/survey.html* has a *date* with value *Sept. 08, 2020*

and the URI for the date is "http://www.semanticweb.org/respDetails/date".

The new graph would be as shown in Figure 2.11 where as the set of statements in Table 2.2 shows the RDF triple notation for Figure 2.11. The first notation represents the entities, survey.html (subject) and Robert James (object), along with respondent (predicate) as the relation between them, using the URIs. The second notation is for the date, where instead of the URI, the value of the property, Sept. 08, 2020, is used. Here, survey.html is the subject and Sept. 08, 2020 is the object with date acting as the predicate.

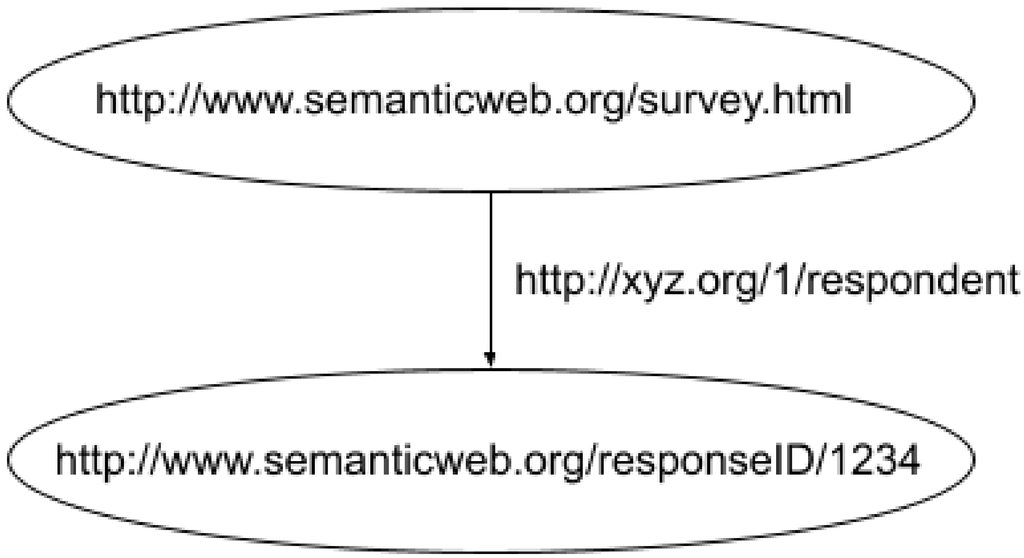



Figure 2.10: RDF Graph for English Statement in Figure 2.8, after RDF Primer [?]

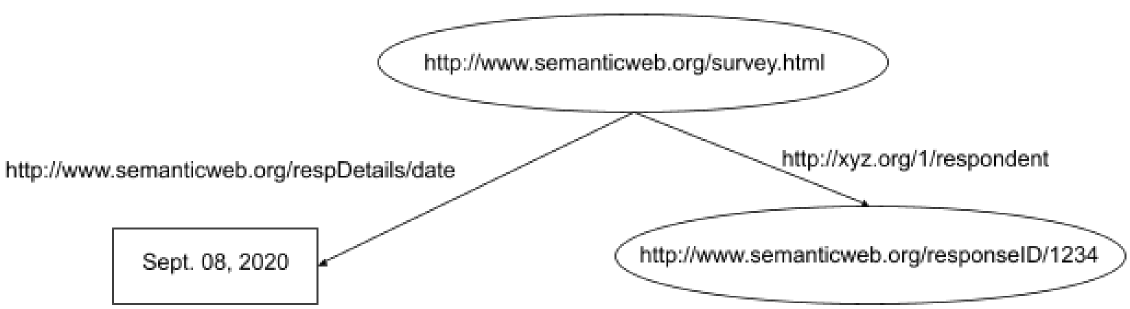



Figure 2.11: RDF Graph for Two Relative Statements, after RDF Primer [?]

| | |
|---|---|
| <http://www.semanticweb.org/survey.html> <http://xyz.org/1/respondent> <http://www.semanticweb.org/responseID/1234> . | |
| <http://www.semanticweb.org/respDetails/date> "Sept. 08, 2020" . | |

Table 2.2: RDF Notations for Figure 2.11, after RDF Primer [?]

Consider a more complex system, where the details of respondent are collected. Suppose, one of the details is name and the other is address of the respondent. Figure 2.12 shows a graphical representation of this RDF system.

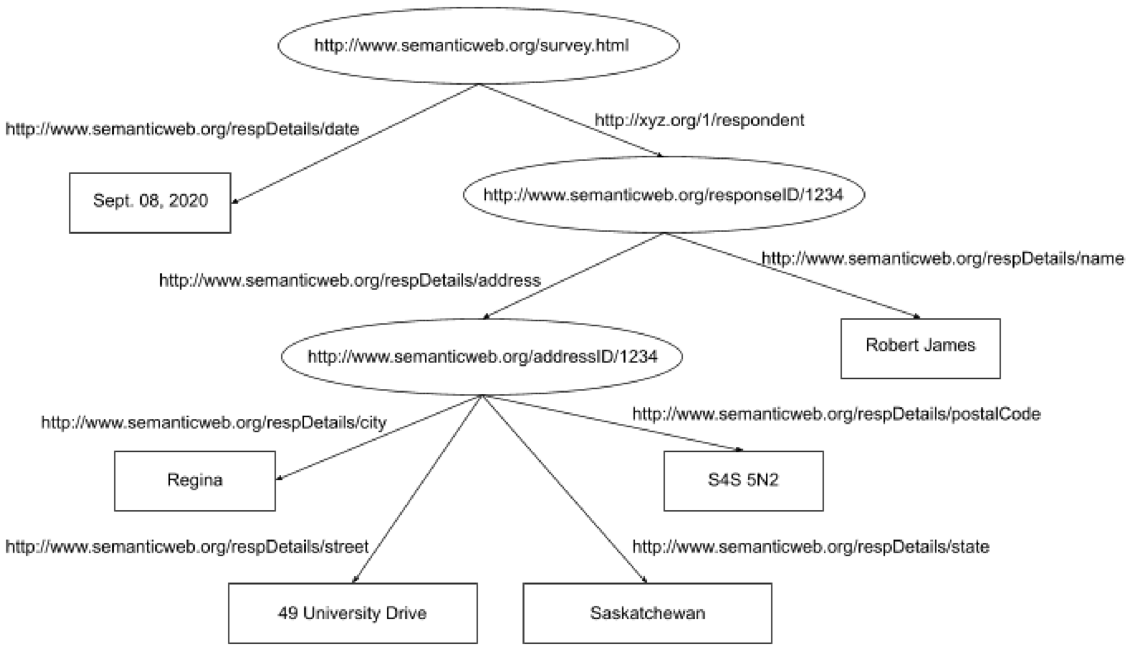



Figure 2.12: Graphical Representation of RDF with Details of the Respondent of the Survey, after RDF Primer [?]

For name, the respondent is subject and Robert James acts as object whereas respName is the predicate. The URI for respName is "http://www.semanticweb.org/respDetails/name" and the value, Robert James, is the object. On the other hand, respondent (subject) is connected to address (object) node, with URI "http://www.semanticweb.org/addressID/1234", through respAddress (predicate) with URI "http://www.semanticweb.org/respDetails/address". Further, the address node has multiple child nodes, namely, street, city, state and postal code. The address (subject) URI is connected to the values of four different object values, 49 University Drive (street), Regina (city), Saskatchewan (state) and S4S 5N2 (postalCode) where the URIs for the predicates are "http://www.semanticweb.org/respDetails/street", "http://www.semanticweb.org/respDetails/city", "http://www.semanticweb.org/respDetails/state" and "http://www.semanticweb.org/respDetails/postalCode" respectively. Instead of using an unnecessary URI for the address entity, a blank node can be used to denote the entity, as shown in Figure 2.13. As the relationships of the node with other nodes give sufficient details without hindering the flow of the data tree, the node does not require a URIreference[15].

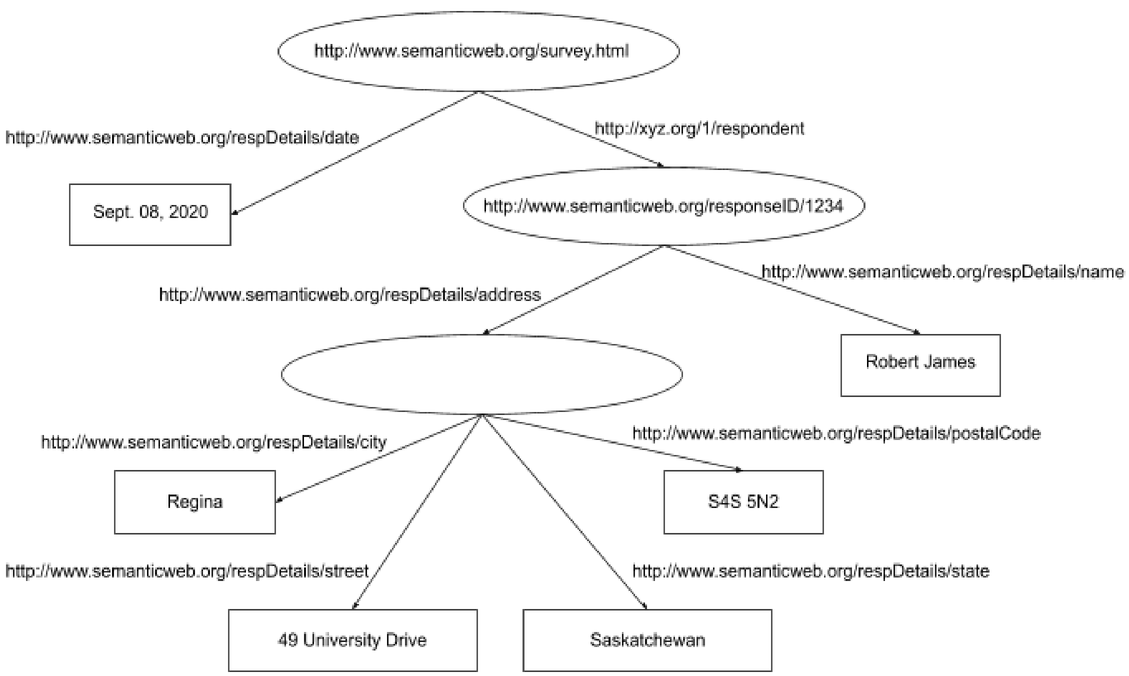



Figure 2.13: RDF Representation in Figure 2.12 without a URI (blank node) for Address Entity, after RDF Primer [**?**]

In 2004, W3C recommended intended goals that are to be achieved whenever an RDF is to be designed. According to the W3CRecommendation[16], the design should "have a simple data model; formal semantics and provable inference; should use an

[15]https://lists.w3.org/Archives/Public/uri/2007Jul/0027.html
[16]https://www.w3.org/TR/rdf-concepts/

extensible URI-based vocabulary and an XML-based syntax; should support the use of XML schema datatypes; and should allow any one to make statements about any resource".

### 2.4.1 Uniform Resource Identifier

An abstract (a logical) or a physical resource on the web is identified by a unique set of ASCII characters which is referred to as a URI[17]. According to Berners-Lee et. al. [?] in RFC 3986, the unique string, on the web, follows a set of guidelines and security considerations while defining a syntax for the generic URI and determining a process for URI references' resolution. As mentioned by W3C, for the operating systems on URIs, some specific "protocols are defined depending on the UriScheme[18]". A system can perform multiple operations like "access", "update", "replace" and "find attributes" on the identified resources. In RFC 2396, Berners-Lee et al. [?] characterized URI using three definitions, as shown in Table 2.3.

#### 2.4.1.1 Syntax of a URI

According to Berners-Lee et al. [?], the generic syntax of a URI comprises five components in a hierarchical sequence which are as follows:

URI = scheme: [//authority] path [?query] [#fragment]

where the authority component consists of three more components which are:

authority = [userinfo@] host [:port]

A syntax diagram can be modelled to represent the components of Generic Syntax[19] of URI along with their hierarchy as shown in Figure 2.14. Consider the following

[17] https://www.w3.org/wiki/URI
[18] https://www.w3.org/2001/tag/doc/SchemeProtocols.html
[19] https://www.w3.org/Addressing/URL/uri-spec.html

| Term | Definition |
|---|---|
| Uniform | There are various benefits of achieving uniformity, like, it facilitates different procedures to access resources in the same environment using different types of resource identifiers; new types of resource identifiers can be introduced without modifying the actual usage of the existing identifiers; common syntactic conventions can be interpreted uniformly for a variety of resource identifiers on the semantic web; and the new applications or protocols can utilize the already existing large and widely-used resource identifiers as the identifiers can be reused in different contexts. |
| Resource | Anything on the web that can be uniquely identified is referred to as a resource. Some day-to-day examples of a resource are a computerized file or folder, weather-broadcasting service, an image, and a set of other resources. Some resources like books in a library, an animal, a human being,and a corporation are a set of resources that cannot be retrieved over the network. A resource is the "conceptual mapping" [?] to a set of entities independent of the corresponding entities at a particular time. Hence, if the conceptual mapping isn't changed, a resource can remain constant irrespective of the changes in its content. |
| Identifier | An object which is used as a reference to a thing with unique identity on the web is termed as an identifier. In URI, a string of ASCII characters restricted by a specific syntax is object. |

Table 2.3: Definitions Used for Characterizing URI, by Berners-Lee et al. [?]

Uniform Resource Location (URL). Figure 2.15 shows its components.

https://parita@www.semanticweb.org:695/ontology/?tag=survey&order=newest#top

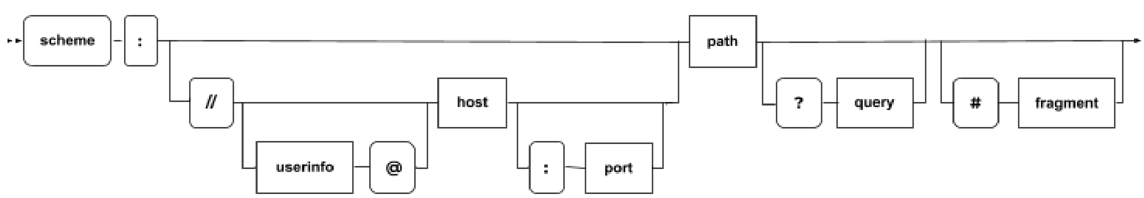



Figure 2.14: Syntax Diagram of a Generic URI, after Berners-Lee et al. [?]

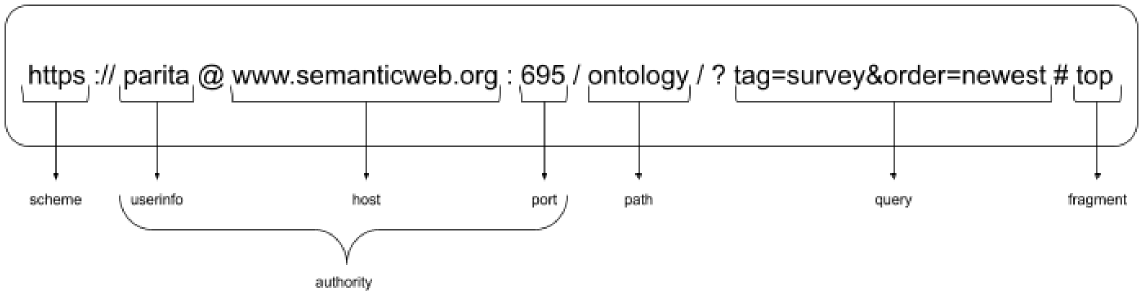



Figure 2.15: Components of an Example of URL, after Berners-Lee et al. [?]

## 2.5 Linked Data Life Cycle

The Linked Data Life Cycle includes multiple stages as described by Ngomo et al. [?] which are listed in Table 2.4. Each stage of this life cycle is independent yet linked to one another. So, it is not mandatory to go through all the stages sequentially to publish the data on the web. It is necessary to use a procedure that follows the life cycle to ensure the quality of data to be published as linked data. The first stage, generally, is data extraction along with conversion of the extracted data (in forms other than RDF) into RDF. Once the data is in RDF form, any or all of the other stages listed in Table 2.4 can be performed.

## 2.6 Conversion of Data from CSV to RDF

The preferred format of the data published on the semantic web is RDF format. Many data scientists use tables and spreadsheets to store and publish data using Comma Separated Values (CSV) files over RDF format as conversion of data to RDF is comparatively difficult. The data used by Dr. Hepting has been stored in tabular format (Sample CSV data consisting of all responses is available in Appendix A ). To convert tabular data into RDF (Appendix B consists of RDF code for the CSV data in

| Stage | Description |
| --- | --- |
| Extraction | Extraction and conversion of unstructured and semi-structured data into RDF |
| Storage & Querying | Storing the RDF data in a format that supports data querying efficiently |
| Authoring | Allowing the users to create new data or to access and modify the existing data |
| Linking | Creating links the data belonging to various domains and users based on their entities |
| Enrichment | Data enrichment for efficient query support using various top-tier structures |
| Quality Analysis | Managing data using different strategies to ensure good quality of available data on the web |
| Browsing & Exploration | Making the structured data available on the web for the users to browse and explore effectively |

Table 2.4: Stages in Linked Data Life Cycle, after Ngomo et al. [?]

Appendix A), W3C[20] recommends the following open source tools [?],

- OpenRefine:

  OpenRefine[21], an RDF extension of Google Refine, is a powerful java-based set of tools that work with messy data using Google Refine Expression Language (GREL)[22]. It allows the users to extract, transform, export, and convert CSV, Ex cel, spreadsheets and other tabular data into RDF where a Graphical User Interface (GUI) is used to define schema mapping.

- RDF123:

  RDF123[23] is a java application and servlet for the conversion of CSV and spreadsheets to RDF which uses arbitrary graphs to define schema mapping. According to Han et. al [?], it uses a graphical RDF123 template to allow users to convert the data from spreadsheets to RDF.

[20]https://www.w3.org/wiki/ConverterToRdf
[21]https://openrefine.org
[22]https://libjohn.github.io/openrefine/grel.html
[23]https://ebiquity.umbc.edu/resource/html/id/237/RDF123-java-application-v1-0

- csv2rdf4lod:

  csv2rdf4lod[24] is a tool that is used for the conversion of tabular data into structured and linked RDF using specifications that are based on declarative RDF enhancement parameters. It forms default namespaces for URIs and provides VoID and original metada for the conversion using identifiers for source organization, dataset and version.

- XLWrap:

  XLWrap[25] is a graphical template based application that allows execution of various SPARQL queries [?]. The spreadsheets are wrapped to arbitrary RDF graphs which allows:

  - Streamed processing of tabluar data like Excel, OpenDocument and CSV
  - Loading Local/HTTP
  - Expressions that include calculator operations for Excel and OpenDocument, and Custom functions
  - Use of Application Programming Interface (API) and SPARQL endpoint

- Tarql:

  Tarql[26] operates as a command-line application for the conversion of CSV to RDF using a user-defined mapping that is written in SPARQL standard 1.1.

[24] https://github.com/timrdf/csv2rdf4lod-automation/wiki
[25] https://xlwrap.sourceforge.io
[26] https://tarql.github.io

## 2.7 Ontology Editors

Ontology editors are used by publishers to create new ontologies when the existing vocabulary is not suitable for their purpose. W3C[27] has recommended a list of ontology editors that can be used to edit, manage, organize and visualize ontology. The list of editors includes Protégé, NeOn Toolkit, SWOOP, Neologism, TopBraid Composer, Vitro, Knoodl, Anzo for Excel, OWLGrEd, Fluent Editor, Semantic Turkey, VocBench. The page does not include valid links to the editors' websites. Protégé is the most popular ontology editor among researchers. There exists negligible information about most of the editors in the list which has driven the decision of the use of Protégé for ontology development and OntoGraf and WebVOWL for visualization.

## 2.8 Applications of Linked Open Data

A variety of projects and websites, that use linked open data, have been developed. Wikidata and DBpedia are two major projects of linked open data that present structured data on the semantic web. This section consists of a brief description of both applications.

### 2.8.1 Wikidata

According to a report by Krotzsch and Vrandecic [?], Wikidata was introduced to the world of information engineering in 2012 by Wikimedia. Wikidata can be defined as "the community-created knowledge base of Wikipedia and the central data management platform for Wikipedia and most of its Sister Projects[28]", according to Erxleben et al. [?]. Wikidata was established for the collection and integration of data on the

[27] https://www.w3.org/wiki/Ontology_editors

[28] Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others

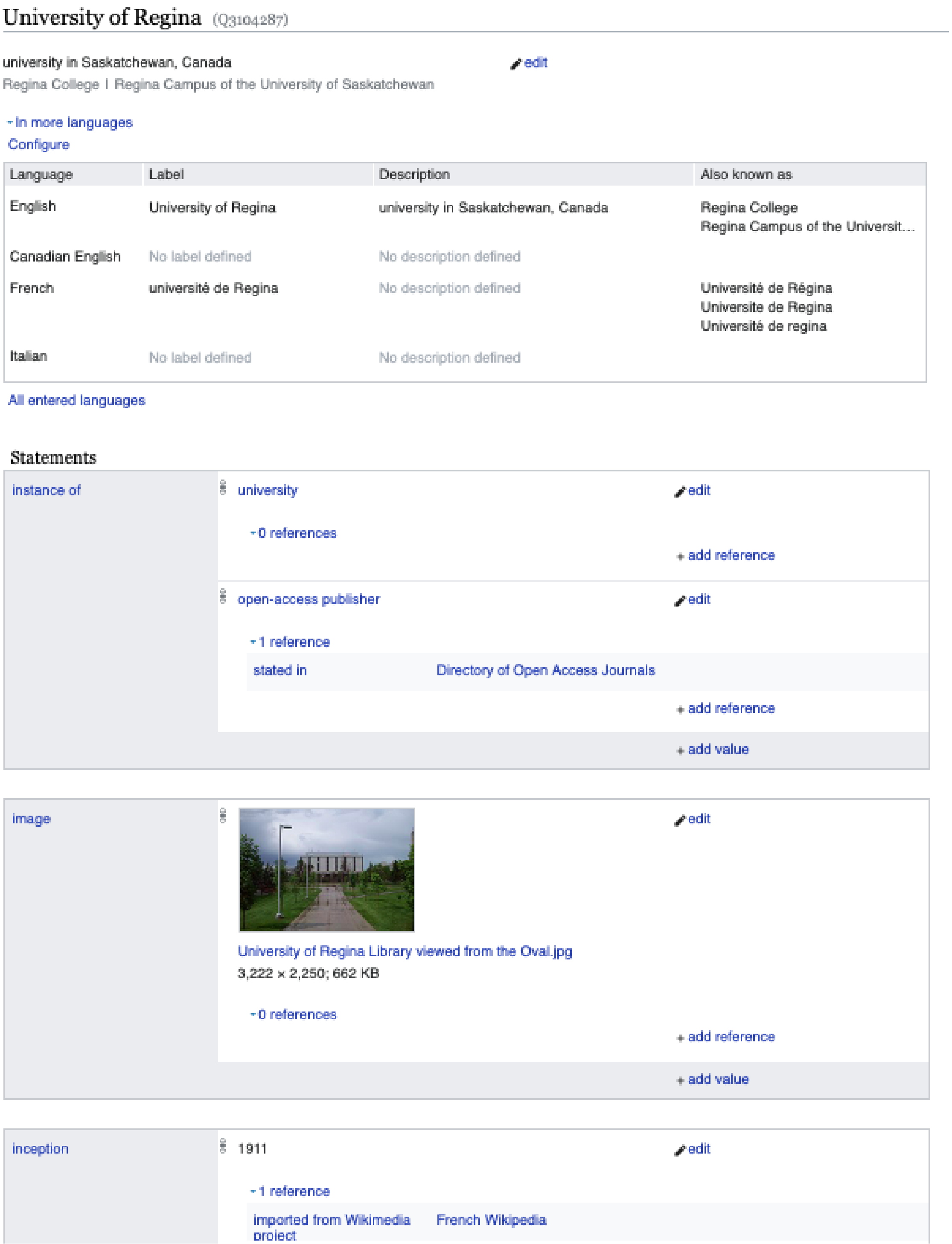
semantic web with Wikipedia. Wikidata is a free open-source tool that handles a large amount of structured data and maintains pages that store the data. Each entity on Wikidata represents the subject of a triple with a dedicated page. The properties of Wikidata are similar to that of RDF where the individuals and classes are indicated using items. As Wikidata is an open-source tool, the data on a page of the Wikidata is easily available for the users to access as well as edit.

For instance, the University of Regina, a university in the provincial capital of Saskatchewan—the city of Regina, has an item page allotted for it. Figure 2.16 shows the Wikidata item page of the University of Regina that can be accessed using the link: https://www.wikidata.org/wiki/Q3104287. Wikidata uses a method of automatically assigning unique identifiers to the entities, which are used as the labels of the item pages instead of the actual names of the entities. These identifiers are a string starting with a capital Q followed by a sequence of digits. The identifiers are assigned irrespective of the language as the Wikidata website supports multiple languages. For the University of Regina, the label used is "Q3104287". The item page on Wikidata consists of multiple sections:

- label: The name of the subject that is treated as an entity. For the instance in Figure 2.16, the label is "University of Regina" along with the unique identifier "Q3104287".

- description: This section consists of general introductory details about the subject. For the University of Regina, the description on wikidata is "university in Saskatchewan, Canada".

- statements: This section consists of a list of subsections called "property". The properties along with their property IDs for the University of Regina are listed in Table 2.5.



## University of Regina (Q3104287)

university in Saskatchewan, Canada edit

Regina College | Regina Campus of the University of Saskatchewan

In more languages

Configure

| Language | Label | Description | Also known as |
|---|---|---|---|
| English | University of Regina | university in Saskatchewan, Canada | Regina College<br>Regina Campus of the Universit... |
| Canadian English | No label defined | No description defined | |
| French | université de Regina | No description defined | Université de Régina<br>Universite de Regina<br>Université de regina |
| Italian | No label defined | No description defined | |

All entered languages

### Statements

instance of

university edit

0 references

add reference

open-access publisher edit

1 reference

stated in Directory of Open Access Journals

add reference

add value

image

edit

University of Regina Library viewed from the Oval.jpg

3,222 × 2,250; 662 KB

0 references

add reference

add value

inception

1911 edit

1 reference

imported from Wikimedia project French Wikipedia

Figure 2.16: Details of University of Regina, Regina, SK on Wikidata

| Statement Property | Property ID |
|---|---|
| instance of | P31 |
| image | P18 |
| inception | P571 |
| country | P17 |
| located in the administrative territorial entity | P131 |
| coordinate location | P625 |
| subsidiary | P355 |
| official website | P856 |
| commons category | P373 |
| topic's main category | P910 |
| category for employees of the organization | P4195 |
| category for alumni of educational institution | P3876 |
| API endpoint | P6269 |
| social media followers | P8687 |

Table 2.5: Statement Properties and Property IDs for the University of Regina on Wikidata

- identifiers: This section consists of various IDs belonging to the subject. For the University of Regina, the set of properties used for identifiers and their IDs are listed in Table 2.6.

The URI for an entity on Wikidata is "https://www.wikidata.org/wiki/<id>", where <id> stands for the unique identifier. For the University of Regina, the <id> is the label identifier—Q3104287. Figure 2.17 shows the RDF/XML description of the University of Regina on Wikidata.

### 2.8.2 DBpedia

Jimmy Wales and Larry Sanger, in 2001, created the largest general reference work on the internet consisting of articles[29] in over 280 languages, the Wikipedia[30], which

[29] include data comprising of templates, images, videos, categorizations, and links to other articles (pages) in multiple languages

[30] https://en.wikipedia.org/wiki/Main_Page

```
<?xml version=”1.0” encoding=”utf-8” ?>
    <rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:ontolex="http://www.w3.org/ns/lemon/ontolex#"
        xmlns:wdt="http://www.wikidata.org/prop/direct/"
        xmlns:wdtn="http://www.wikidata.org/prop/direct-normalized/">
    <rdf:Description rdf:about="https://www.wikidata.org/wiki/Special:EntityData/Q3104287">
        <rdf:type rdf:resource="http://schema.org/Dataset"/>
        <schema:about rdf:resource="http://www.wikidata.org/entity/Q3104287"/>
        <cc:license rdf:resource="http://creativecommons.org/publicdomain/zero/1.0/"/>
        <rdf:Description rdf:about="https://en.wikipedia.org/wiki/University_of_Regina">
            <rdf:type rdf:resource="http://schema.org/Article"/>
            <schema:about rdf:resource="http://www.wikidata.org/entity/Q3104287"/>
            <schema:inLanguage>en</schema:inLanguage>
            <schema:isPartOf rdf:resource="https://en.wikipedia.org/"/>
            <schema:name xml:lang="en">University of Regina</schema:name>
            <schema:description xml:lang="en">university in Saskatchewan,
            Canada</schema:description>
            <wdt:P18
            rdf:resource="http://commons.wikimedia.org/wiki/Special:FilePath/University%2
            0of%20Regina%20Library%20viewed%20from%20the%20Oval.jpg"/>
        </rdf:Description>
    </rdf:Description>
</rdf:RDF>
```


Figure 2.17: Wikidata Entry (RDF/XML) for University of Regina, Regina, SK

| Identifier Property | Property ID |
|---|---|
| VIAF ID | P214 |
| ISNI | P213 |
| GND ID | P227 |
| Library of Congress authority ID | P244 |
| WorldCat Identities | P7859 |
| ARWU university ID | P5242 |
| Crossref funder ID | P3153 |
| Facebook ID | P2013 |
| Freebase ID | P646 |
| Google Maps Customer ID | P3749 |
| GRID ID | P2427 |
| HAL structure ID | P6773 |
| LittleSis organization ID | P3393 |
| Microsoft Academic ID | P6366 |
| Ringgold ID | P3500 |
| ROR ID | P6782 |
| Times Higher Education World University ID | P5586 |
| Twitter username | P2002 |
| U-Multirank university ID | P5600 |

Table 2.6: Identifier Properties and Property IDs for University of Regina on Wikidata

acts as a source of data for many web projects like DBpedia[31], one of the major linked open data projects. The main aim behind the development of DBpedia is the conversion of unstructured data on Wikipedia into structured datasets. DBpedia enables querying the Wikipedia data as well as the generation of links among the structured datasets, resulting in the establishment of a massive web of data.

Initially, the template of the page is discovered using a Template Extraction Algorithm[32] which results in the identification of the structure of the template using various Pattern Matching Techniques[33]. Once the appropriate template is selected, the template data is parsed using a suitable algorithm and transformed into RDF triples.

[31]https://wiki.dbpedia.org/

[32]like Principle Components Analysis, Linear Discriminant Analysis, Logically Linear Embedding

[33]https://www.dummies.com/programming/big-data/data-science/how-pattern-matching-works-in-data-science

MediaWiki[34] links are identified and assigned specific URIs. On the other hand, the common units of data are recognized and assigned appropriate datatypes whereas RDF lists are created for the object lists. According to Auer et al. [?], DBpedia in 2007 was a platform that delivered information on "more than 1.95 million 'things' that includes at least 80,000 persons, 70,000 places, 35,000 music albums, 12,000 films". The DBpedia semantic web services are handled using the GNU Free Documentation License terms and policies. The data on DBpedia is easily accessible in different ways like accessing the datasets as linked data; using SPARQL queries for data extraction and accessibility; and in the form of a downloadable RDF snippets [?]. Figure 2.18 shows DBpedia entry for the "University of Regina, Regina, SK". Like Wikidata, DBpedia also has a label, a description section and a set of important property-value pairs that give information related to the entity that has been described.

Figure 2.19 shows the data architecture of DBpedia. An open-link software, Virtuoso Universal Server[35], is a data virtualization tool used to host and publish RDF datasets on DBpedia, which can be easily accessed using SPARQL endpoint. The HTML or RDF representation of the DBpedia resources is enabled by the data architecture using HTTP that supports the standard HTTP GET method calls from the web clients.

[34] https://www.mediawiki.org/wiki/MediaWiki
[35] https://dbpedia.org/page/Virtuoso_Universal_Server

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

## About: University of Regina

An Entity of Type : جامعة عامة, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

جامعة ريجينا هي جامعة عامة في كندا، تأسست عام 1911. تقع في مدن ريجينا، ساسكاتون، سويفت كورنت.

| Property | Value |
|---|---|
| dbo:abstract | ▪ The University of Regina is a public research university located in Regina, Saskatchewan, Canada. Founded in 1911 as a private denominational high school of the Methodist Church of Canada, it began an association with the University of Saskatchewan as a junior college in 1925, and was disaffiliated by the Church and fully ceded to the University in 1934; in 1961 it attained degree-granting status as the Regina Campus of the University of Saskatchewan. It became an autonomous university in 1974. The University of Regina has an enrollment of over 15,000 full and part-time students. The university's student newspaper, The Carillon, is a member of CUP. The University of Regina is well-reputed for having a focus on experiential learning and offers internships, professional placements and practicums in addition to cooperative education placements in 41 programs. This experiential learning and career-preparation focus was further highlighted when, in 2009 the University of Regina launched the UR Guarantee Program, a unique program guaranteeing participating students a successful career launch after graduation by supplementing education with experience to achieve specific educational, career and life goals. Partnership agreements with provincial crown corporations, government departments and private corporations have helped the University of Regina both place students in work experience opportunities and help gain employment post-study. (en) |
| dbo:affiliation | ▪ dbr:University_of_the_Arctic<br>▪ dbr:Canadian_Association_of_Research_Libraries<br>▪ dbr:Canadian_Bureau_for_International_Education<br>▪ dbr:Canadian_University_Society_for_Intercollegiate_Debate<br>▪ dbr:International_Association_of_Universities<br>▪ dbr:L'Association_des_universités_de_la_francophonie_canadienne<br>▪ dbr:Canadian_University_Press<br>▪ dbr:Association_of_Universities_and_Colleges_of_Canada |
| dbo:athletics | ▪ dbr:U_Sports |
| dbo:city | ▪ dbr:Regina,_Saskatchewan |
| dbo:endowment | ▪ 9.75E7 |
| dbo:formerName | ▪ Regina College (1911–1961) (en)<br>▪ Regina Campus of theUniversity of Saskatchewan(1961–1974) (en) |
| dbo:motto | ▪ As one who serves |
| dbo:numberOfPostgraduateStudents | ▪ 2027 (xsd:integer) |
| dbo:numberOfStudents | ▪ 16501 (xsd:integer) |

Figure 2.18: Description of University of Regina, Regina, SK on DBpedia

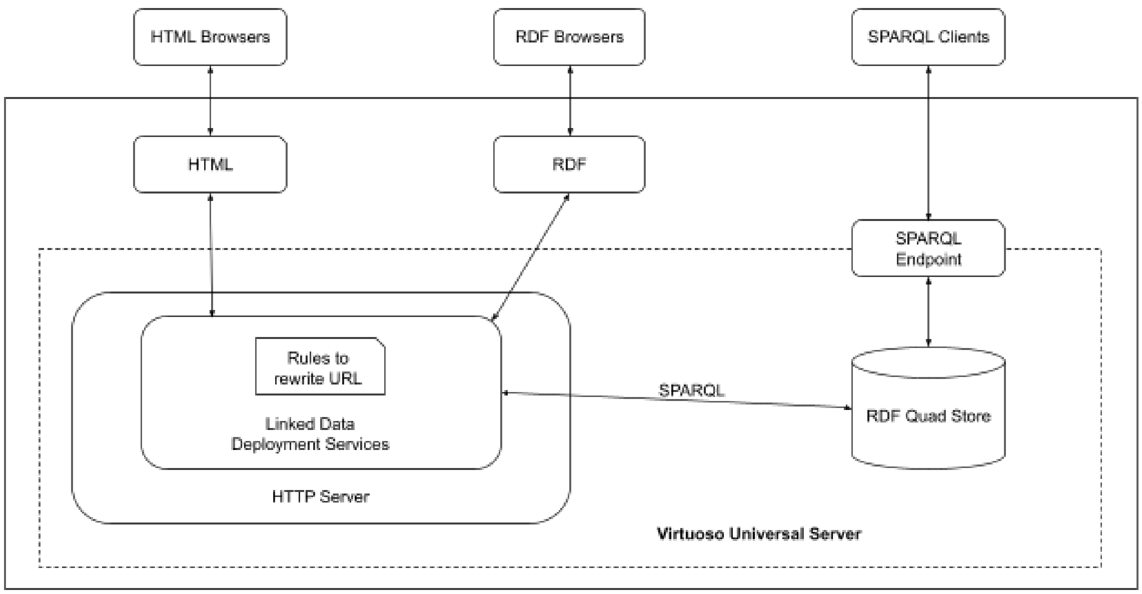



Figure 2.19: DBpedia Data Architecture

# Chapter 3

# Proof Of Concept

This chapter focuses on some fundamental concepts of OWL ontologies that were considered while developing the survey instrument for this research work.

## 3.1 Ontology in Web Semantics

Ontology is a branch of information science that encloses representation, nomenclature, and definition of various properties, categories, and relationships shared among the entities[1] that support any number of domains the user is interested in. In other words, Ontology is a form of defining a set of concepts for representing the subject properties and the relationships shared among them. According to Tom Gruber [?], ontology is "an explicit specification of a conceptualization", where conceptualization is "an abstract, simplified view of the world (includes objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them) that we wish to represent for some purpose". In web semantics, an ontology is used to define the semantics of the resources briefly and methodically. The ontologies have have been used to "specify the physical as well as conceptual characteristics of resources, in the form of metadata schema on the semantic web, for a fixed set of

[1] includes all types of data, concepts and entities

users" [?].

## 3.2 OWL Ontologies

W3C developed a standard ontology language called OWL[2] to define and describe concepts for ontology using different facilities and operations like intersection, union and negation. It is based on a logical model due to which a reasoner[3] can be used to check the mutual consistency among the statements and definitions in the ontology. The reasoner is also responsible for the identification of a suitable concept for particular definitions [?].

### 3.2.1 Components of OWL Ontologies

An OWL ontology consists of a set of components that define and describe the concepts collectively. The basic components of any OWL ontology, Individuals, Properties and Classes, are briefly explained in this section.

- Individuals:

  According to Horridge et al. [?], "Individuals represent objects in the domain in which we are interested". Individuals are the objects of classes. Figure 3.1 shows the representation of different individuals in various domains. The triangles are used here to denote the individuals in free space.

- Properties:

  The Binary Relationships[4] between the individuals (things) are called properties. For instance, the individuals "Max" and "Kieron" from Figure 3.1 are linked to

[2]http://www.w3.org/TR/owl- guide/
[3]maintains hierarchy accurately
[4]links shared by two individuals


Figure 3.1: Representation of Individuals in OWL

each other by the property “hasSibling”. Each property can have its inverse. For instance, “hasSibling” can have an inverse property “isSiblingOf”. The properties can be symmetric or transitive. It is possible to have functional properties where the properties can be restricted to having only one value. Figure 3.2 shows representation of properties between the individuals “Max” and “India”, and “Max” and “hasSibling”. The relation between “Max” and “India” is “livesIn”. The arrow between two individuals in the figure denotes the property (relationship) between them.

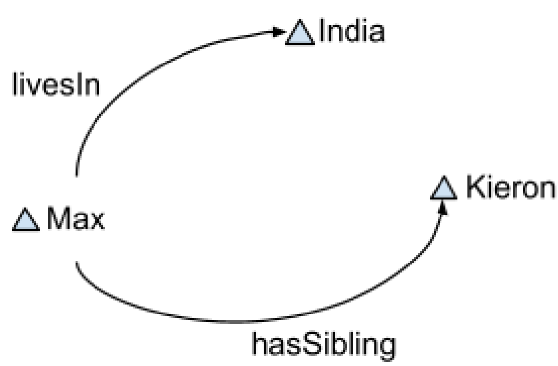



Figure 3.2: Representation of Properties in OWL

- Classes:

  In OWL, the sets consisting of individuals are called OWL classes. Mathematical (formal) descriptions are used to describe the OWL classes. These descriptions include accurate statements on the requirements for the class membership. For instance, the class "Country" consists of all the individuals that are countries in the Domain of Discourse[5]. Inheritance in OWL classes (Class Hierarchy) may exist with the classes having subclasses and/or super-classes which is technically termed as taxonomy. "Subclasses specialise ('are subsumed by') their super-classes" [?]. For instance, consider the classes "Lady" and "Human", where "Lady" is a subclass of "Human". This implies the following statements:

- All ladies are humans
- All members of class "Lady" are members of class "Human"
- Being a "Lady" implies that individual is "Human"
- "Lady" is subsumed by "Human"

Figure 3.3 shows the representation of the classes for the individuals and properties that are shown in Figure 3.2.

## 3.3 OWL Ontology Inconsistencies and Reasoner

A Reasoner[6] is used in Protégé to address and solve the inconsistencies. The current version of the Protégé Integrated Development Environment (IDE) offers support for some built-in semantic reasoners which are listed below:

- ELK 0.4.3[7] : A Java-based fast reasoner for lightweight OWL2 EL which is available under Apache License 2.0

[5] can contain one or more classes

[6] https://en.wikipedia.org/wiki/Semantic_reasoner

[7] https://www.w3.org/2001/sw/wiki/ELK

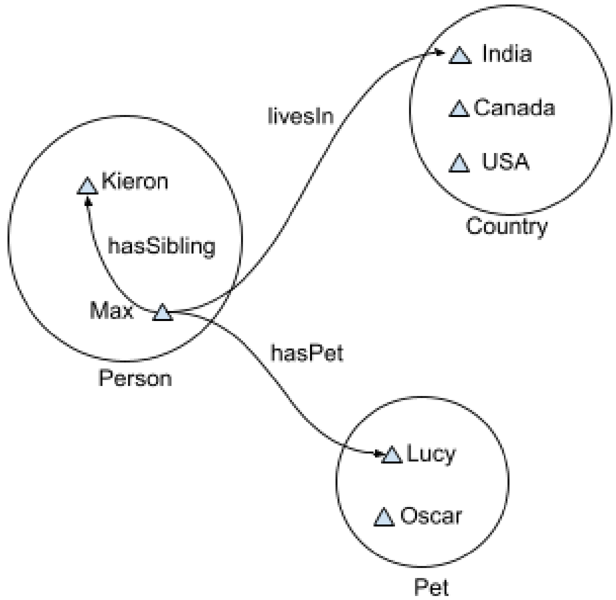



Figure 3.3: Representation of OWL Classes Consisting of Individuals and Properties

- FaCT++ 1.6.5[8] : A Tableaux-based reasoner for expressive OWL Description Logics (DL) which is implemented using C++

- HermiT 1.4.3.456[9] : A Java-based reasoner, written using OWL, that can be used to determine consistency of the ontology as well as for the identification of relationships among the classes of the ontology

- Mastro DL-Lite[10] : An Ontology-Based Data Access (OBDA) management system in which the DL-Lite family of languages for lightweight DL are used for ontology specifications

- Ontop 4.1.0[11] : A reasoner which uses SPARQL for querying datasets as Virtual RDF Graphs

[8]https://www.w3.org/2001/sw/wiki/Fact
[9]https://www.w3.org/2001/sw/wiki/Hermit
[10]https://protegewiki.stanford.edu/wiki/Mastro_DL-Lite_Reasoner
[11]https://www.w3.org/2001/sw/wiki/Ontop

- Pellet and Pellet (incremental)[12] : A Java-based OWL2 reasoner for the optimization of nominals, conjunctive query answering, and incremental reasoning

[12]https://www.w3.org/2001/sw/wiki/Pellet

# Chapter 4

# OWL Ontology for Survey

## 4.1 Development Tool for the Creation of New OWL Vocabularies

In data science, for a fresh and developing branch like ontology engineering, it is difficult to find enough knowledge that can be reused. The field is still growing with research and development in progress as well as a lot still required to be explored in the future. As a result of this, linked open vocabularies are limited currently, and hence, the data publishers will have to create a vocabulary in case they do not find the required vocabulary from the already existing vocabularies.

There are editors available today that help the publishers to create their own linked open vocabularies. Alatrish at the University of Belgrad [?], in 2013, compared the most used ontology editors, namely, Apollo, OntoStudio, Protégé, Swoop, and TopBraid Composer. The editors were compared based on the following features:

- General description: developers' details, and accessibility on the internet (open source or software licensed)
- Software architecture and tools evaluation: semantic web architecture, support for plug-ins, backup management, and ontology storage

- Interoperability: interoperability with other ontology tools, support and translation for other languages
- Knowledge representation: Knowledge Representation (KR) paradigm of knowledge model, axiom language, and methodological support
- Inference services: built-in inference engine, other attached inference engines, and constraint/consistency checking
- Usability: graphical taxonomy, graphical views, zooms, collaborative working, and ontology libraries

After studying the details and comparison among the most used editors based on the above features by Alatrish [?], Protégé has been used for the development of the instrument to support this research and thesis work.

## 4.2 Protégé

Stanford University developed a free, open-source ontology development platform, Protégé[1], that offers a set of substantial tools[2] to build ontologies for knowledge-based applications and to implement various "knowledge-modeling structures and actions that support the creation, visualization and manipulation of ontologies in various representation formats" [?]. Protégé offers customization options to build knowledge models and to add data.

[1]https://protege.stanford.edu

[2]along with various plug-ins and Java-based API

### 4.2.1 Usability of Protégé

The International Organization for Standardization (ISO) standard 9241[3] defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". In usability engineering, Jacob Neilson [?] suggested five qualities of a usable product: "learnability, efficiency, memorability, errors (low rate, easy to recover), and satisfaction". These were the original five dimensions of usability which were then re-termed as all five words starting with "E" by Quesenbery [?], and they became popular as the five Es of usability [?]. The new terms are as shown in Figure 4.1.

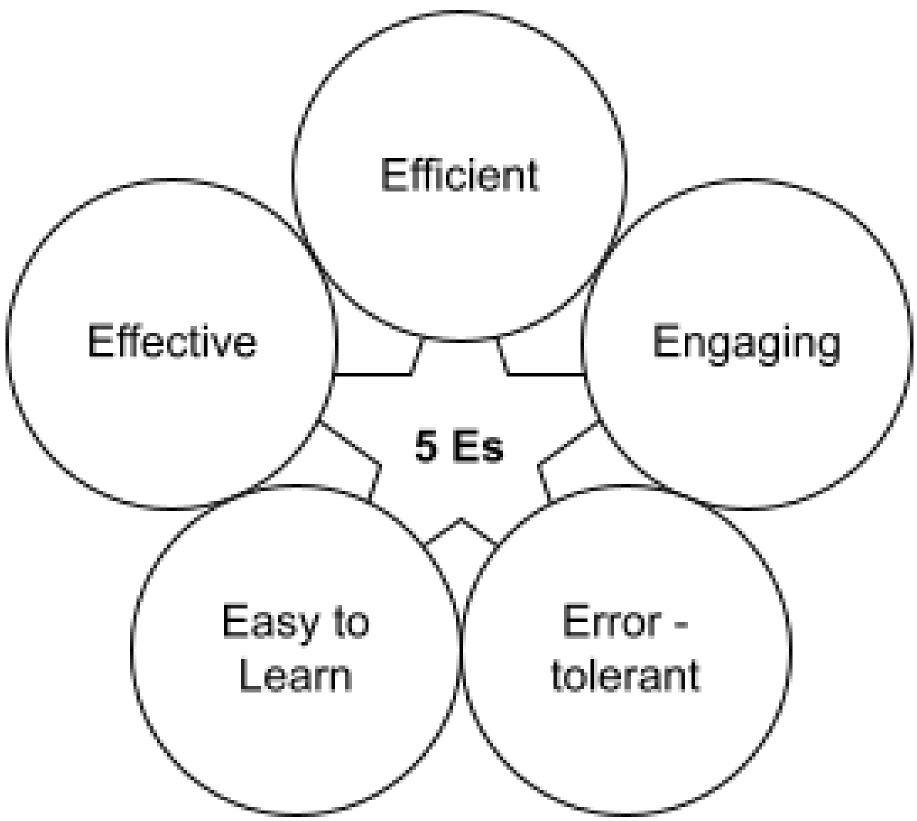



Figure 4.1: Five Es of Usability

- Effective: The task is completed accurately

[3]https://www.iso.org/standard/63500.html

The user must be able to complete the desired task accurately using the design. If the user is unable to complete the task, irrespective of the time spent and efforts made to complete the task, the design is said to have failed to reach meet the design's goals. To measure effectiveness in a design, it is necessary to know users' perspectives of success and accuracy.

- Efficient: The task is completed quickly

  The user must be able to complete the task easily. The interface should be designed to allow the users to complete the desired tasks with the least complexity and within the minimum possible time. If a simple task takes more than the required time for completion, the design is not feasible and would be said to not meet the goals.

- Engaging: The look and feel of the design is interactive and engaging

  The look and feel of the interface must be interactive so that the user finds it interesting and continues using it. The user of the interface should be able to connect with the presentation and the organization of the components of the interface and hence should be satisfied by using the design to complete the desired task.

- Error-tolerant: Error prevention and data recovery in case any error is committed

  The interface should be able to easily handle errors. This means in case the user commits mistakes, the interface should be helpful enough to not lose the work done by the user along with providing the error correction options.

- Easy to learn: The interface should be easy to learn

  The user should get the feeling of familiarity and adaption while using the interface. The names of the components and tools in the design must be appropriate so that the user can understand their purpose from the name itself. For instance, in a banking application, using the keyword "balance owed" to denote the debit amount might confuse the user by not giving a clear idea about who owes the money, that is, whether the customer owes the amount to the bank or vice versa.

The five dimensions of usability can be considered while designing as well as for evaluation. While designing, the engineers consider the five dimensions as goals to be achieved whereas while testing and evaluation, they are considered as the requirements necessary for the designed model. The five dimensions of usability were used to assess the usability of Protégé and the report of the evaluation is as follows:

- Effective: Protégé is an effective design as it allows the users to build ontologies and use other tools precisely. The users can complete the desired tasks accurately using the set of facilities available.

- Efficient: It is easy and straightforward to describe the vocabularies and build ontologies. All the desired tasks in Protégé can be completed quickly (without wasting much time) as it is easy to look for the facilities using the menus and tools available. Hence, Protégé is an efficient ontology builder.

- Engaging: The tools and menus in Protégé are interactive which leads to a better experience for the user. Also, the layout and the components are attractive and decent for the user to use this ontology builder whenever necessary. Overall, Protégé gives an engaging and satisfactory user experience.

- Error-tolerant: Protégé provides options for modifications, additions, and deletions of classes, objects, and properties, which allow the users to correct the errors that were committed. Also, the deleted classes can be recovered using "Undo" from the edit menu resulting in proving Protégé as an error-tolerant design.

- Easy to learn: Protégé is easy to learn as it has a familiar look and commonly used names for the tools and menus making it easy for the users to adapt and know them nicely. Also, the "Help" document along with a list of some frequently asked questions is available which helps the users in the correct direction whenever they are lost.

## 4.3 Procedure to Model the Survey Ontology in OWL Using Protégé Tools

This section focuses on a step-wise methodology that has been followed to create new OWL vocabularies for a survey using Protégé. Figure 4.2 shows the student feedback form for which the new vocabularies are described. Protégé is available for use as a web application. Protégé can also be downloaded[4] to use as a desktop application for ontology construction and visualization.

To commence with the process of building the survey ontology, a new project needs to be created with a unique Internationalized Resource Identifier (IRI)[5] and the ontology version. Figure 4.3 shows the Protégé home-screen with a new project created and an IRI with the name and version of ontology.

### 4.3.1 Classes and Class Hierarchy

The next step is to create classes and build the hierarchy for the class nodes. Each entity in Protégé is directly or indirectly a child node of the "Thing" class. Also, each object has a name-value pair, which generally are the nodes of the "Domain_Entity" class, an inherited class of Thing. Initially, four classes, "Thing", "Domain_Entity",

[4] https://protege.stanford.edu/products.php#desktop-protege
[5] https://www.w3.org/2011/rdf-wg/wiki/IRIs/RDFConceptsProposal

Student Feedback Form

(adapted from http://cft.vanderbilt.edu/teaching-guides/reflecting/student-feedback/#inclass)

**Course**: ______________ **Instructor**: Daryl Hepting **Date**: ____________________

1 = Never; 5 = Always

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | The instructor is well-prepared for class. | 1 | 2 | 3 | 4 | 5 |
| 2 | The instructor clearly communicates his expectations for student preparation and participation. | 1 | 2 | 3 | 4 | 5 |
| 3 | The instructor uses class time effectively. | 1 | 2 | 3 | 4 | 5 |
| 4 | The instructor has clear expectations for assigned work. | 1 | 2 | 3 | 4 | 5 |
| 5 | The instructor encourages student participation. | 1 | 2 | 3 | 4 | 5 |
| 6 | The instructor clearly answers questions. | 1 | 2 | 3 | 4 | 5 |
| 7 | The instructor treats students with respect. | 1 | 2 | 3 | 4 | 5 |
| 8 | The instructor effectively directs and stimulates discussion. | 1 | 2 | 3 | 4 | 5 |
| 9 | The instructor effectively encourages students to ask questions and give answers. | 1 | 2 | 3 | 4 | 5 |

What do you like best about this course?

What would you like to change about this course?

What do you think the instructor's greatest strengths are?

Figure 4.2: The Student Feedback Form, from Hepting [?]

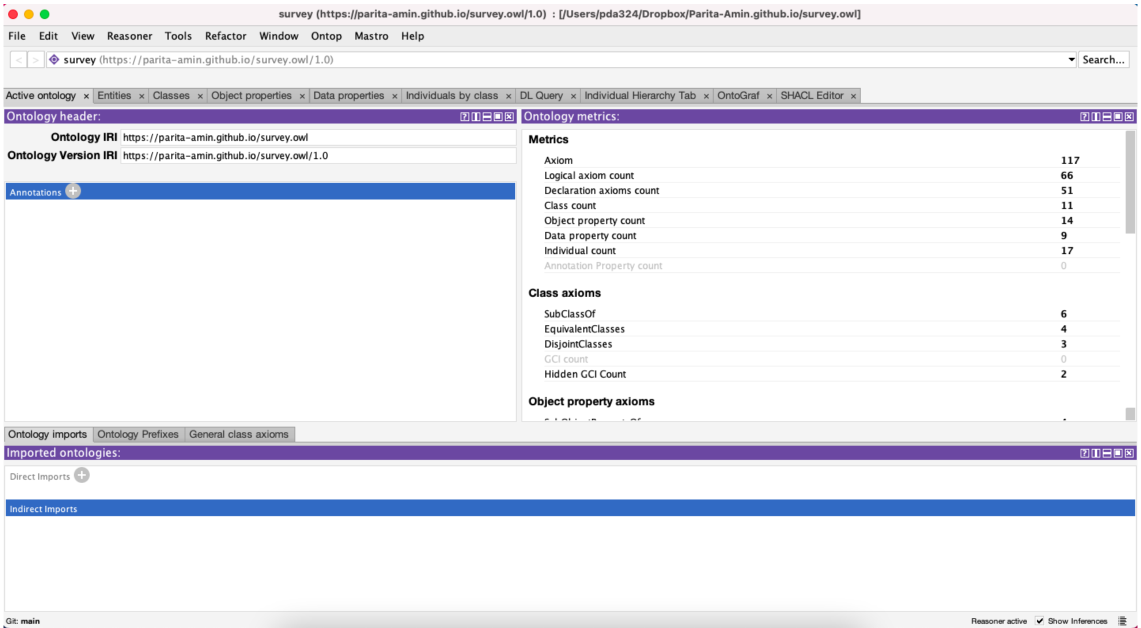



Figure 4.3: Protégé Home-screen for a New Project with an Ontology Name and an IRI

"Independent_Entity", and "Value" are created using class hierarchy, as shown in Figure 4.4. To create a class hierarchy for the classes that represent entities in the survey, there are three steps to be followed:

1. Select "owl:Thing" in the Class Hierarchy View of the Classes tab from the Windows menu followed by selecting "Create class hierarchy..." from the Tools menu, as shown in Figure 4.5 to open a "Enter hierarchy" pop-up window which allows the users to add classes to the ontology.

2. Enter the class names in the pop-up window starting from the left. For the child nodes, leave some spaces before writing the class name. Each horizontal line consists of one class name, as shown in Figure 4.6.

3. On clicking "Continue" on the pop-up window in Figure 4.6, it opens up a new pop-up window. Here, it is important to state whether one wants the sibling classes to be disjoint or not. The checkbox needs to be checked accordingly.

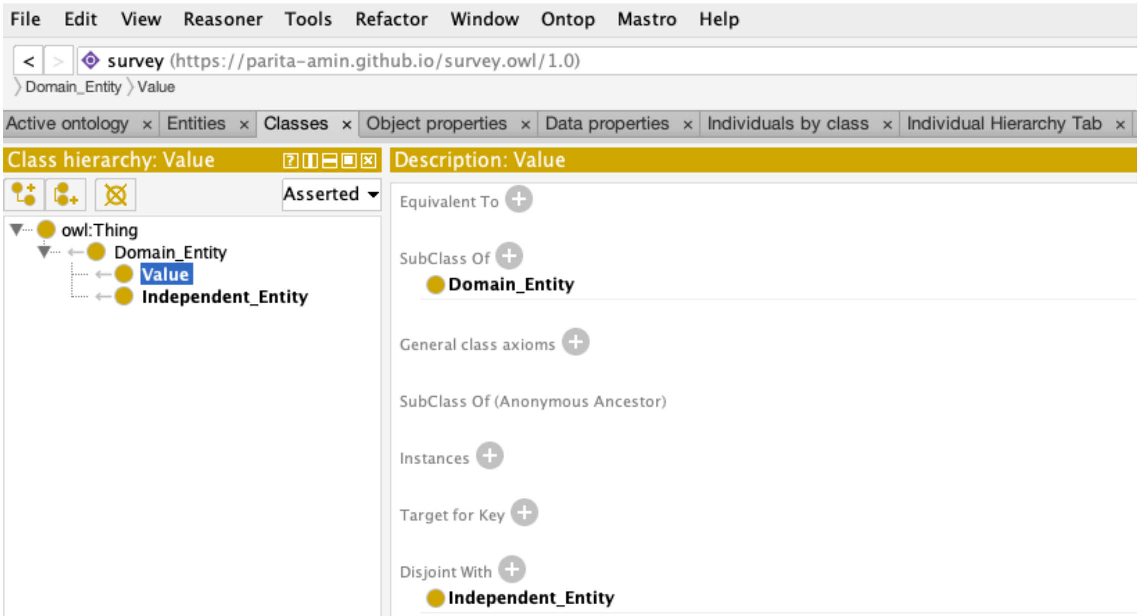



Figure 4.4: Class Hierarchy of Thing, Domain_Entity, Independent_Entity, and Value

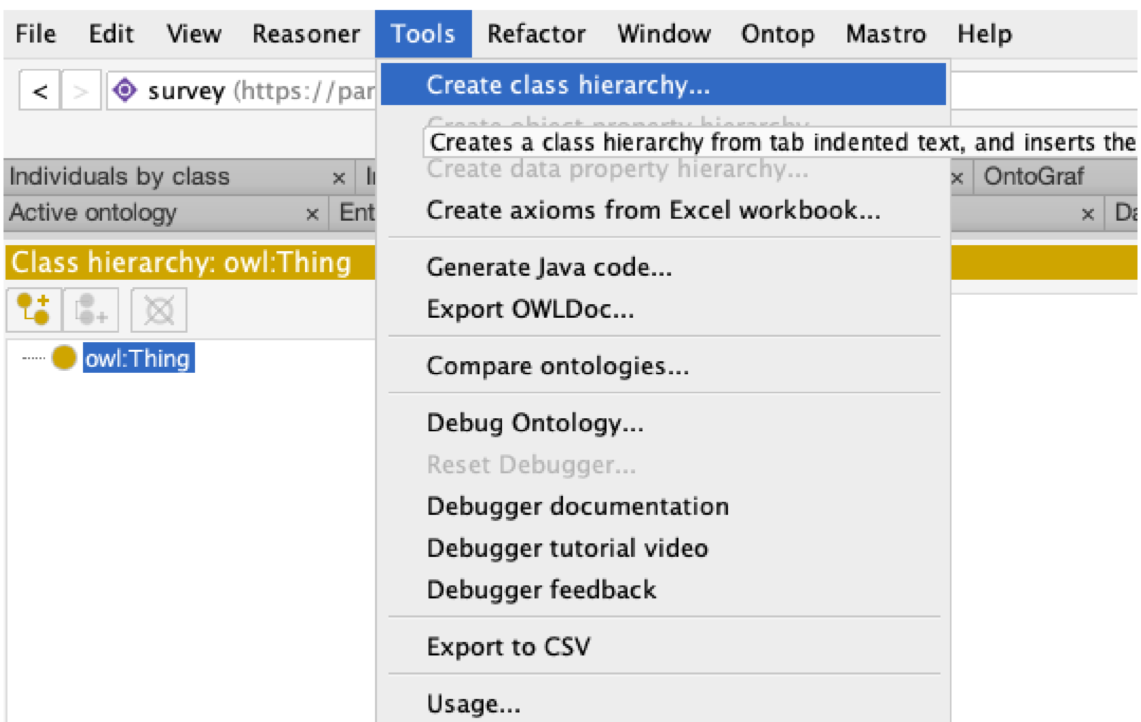



Figure 4.5: Tools Menu to Create Class Hierarchy

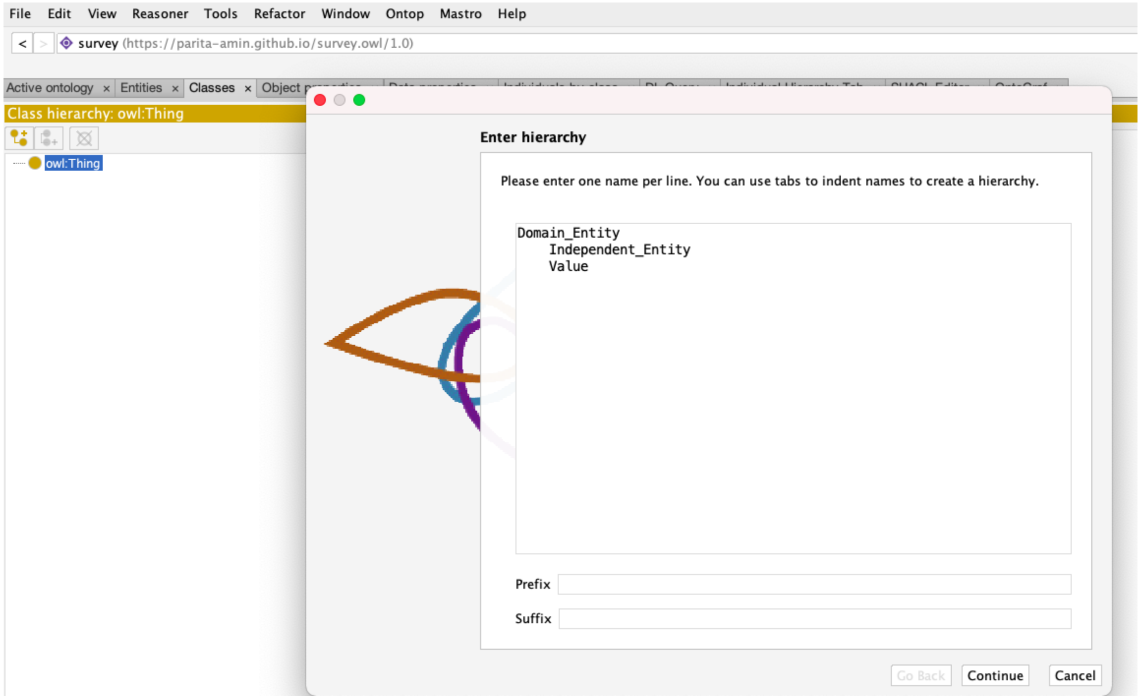



Figure 4.6: “Enter hierarchy” Pop-up Window

Usually, it is recommended to create disjoint sibling classes which can be changed later as necessary, as shown in Figure 4.7.

Once the classes are added, the next step is to describe the classes. Descriptions of classes like “Equivalent to”, “SubClass Of”, “Instances”, “Disjoint With” and so on can be added using the Description view of the Class tab (shown in Figure 4.4). Each class can be described individually by selecting a particular class and adding details to the description view.

All the classes can be created and described using the above steps. In the end, the class hierarchy for the survey ontology would be as shown in Figure 4.8. The survey can have “Respondent”, “Response Date”, “Type of Survey”, “Survey Questions”, “Survey Answers”, and “Likert Scale”, which are considered for this ontology as well. Here, the type of survey used is “feedback survey”. The survey questions and survey answers would further have entities corresponding to various forms of question-answer.

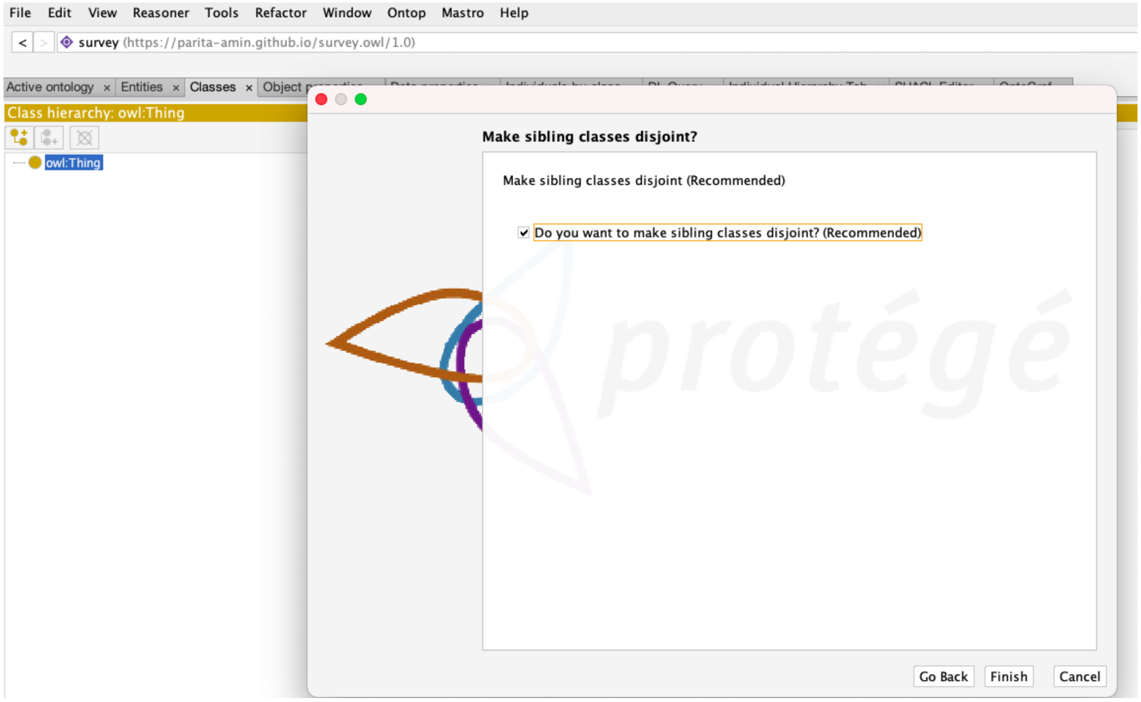



Figure 4.7: Pop-up Window to Create Disjoint Sibling Classes

Generally, feedback surveys consist of two types of questions, namely, "Likert Questions" and "Open-ended Questions". The type of questions might differ depending on the designer of the survey, some of which, as stated on the *SurveyMonkey*[6] website, are listed below.

- Multiple Choice Questions
- Rating Scale Questions
- Likert Scale Questions
- Matrix Questions
- Dropdown Questions
- Open-ended Questions
- Demographic Questions
- Ranking Questions
- Image Choice Questions
- Click Map Questions
- File Upload Questions
- Slider Questions
- Benchmarkable Questions

[6]https://www.surveymonkey.com/mp/survey-question-types/

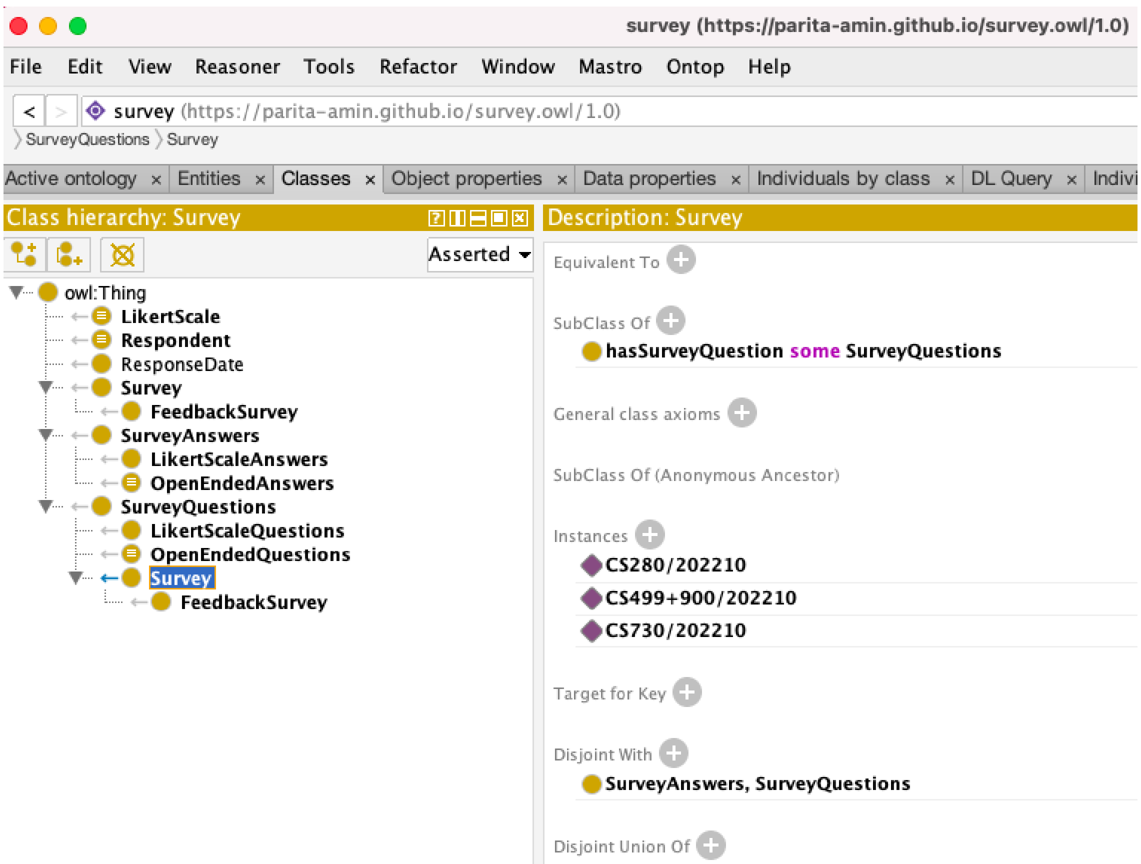


Figure 4.8: Class Hierarchy in Survey Ontology

For this research, the feedback survey that has been considered has Likert and Open-ended questions. The entities described for the questions and answers are "LikertScale-Questions", and "OpenEndedQuestions" as sub-classes of "SurveyQuestions", and "LikertScaleAnswers", and "OpenEndedAnswers" as sub-classes of "SurveyAnswers".

For all the survey questions, there would be a question and an answer, where both of them would be considered as distinct entities. Likert questions are the questions for which the answers are from the preset list of answers. For the Likert questions, the questions can be either of type "Never-Always-4" or "Never-Always-5" or "Never-Always-7" or "Never-Always-*", of which Never-Always-5 has been used by Hepting [?]. In a Never-Always-5 type of survey, 5 degrees of agreement are used as shown in Table 4.1. In addition to these degrees, this ontology also considers the instances where the question wouldn't be answered. In this case, the field corresponding to the answer is left blank. On the other hand, Open-ended questions have unpredictable answers and hence there cannot be any degree of measurement for Open-ended questions.

| Number | Agreement |
|---|---|
| 1 | Never |
| 2 | [Rarely] |
| 3 | [Sometimes] |
| 4 | [Often] |
| 5 | Always |

Table 4.1: Degrees of Agreement for Likert-scale Questions

### 4.3.2 Object Properties and Object Property Hierarchy

The next step in building the ontology, after describing the classes, is describing the "Object Properties". Object properties are used to link the entities (classes in Protégé). The "owl:topObjectProperty" is the root node of the object properties' hierarchy. The hierarchy for object properties can be built by selecting a property in the Object Property Hierarchy view of the Object Property Views tab and then going to the tools menu as shown in Figure 4.9. The remaining steps are similar to the ones that are followed for building class hierarchy (Figure 4.5, 4.6, and 4.7).

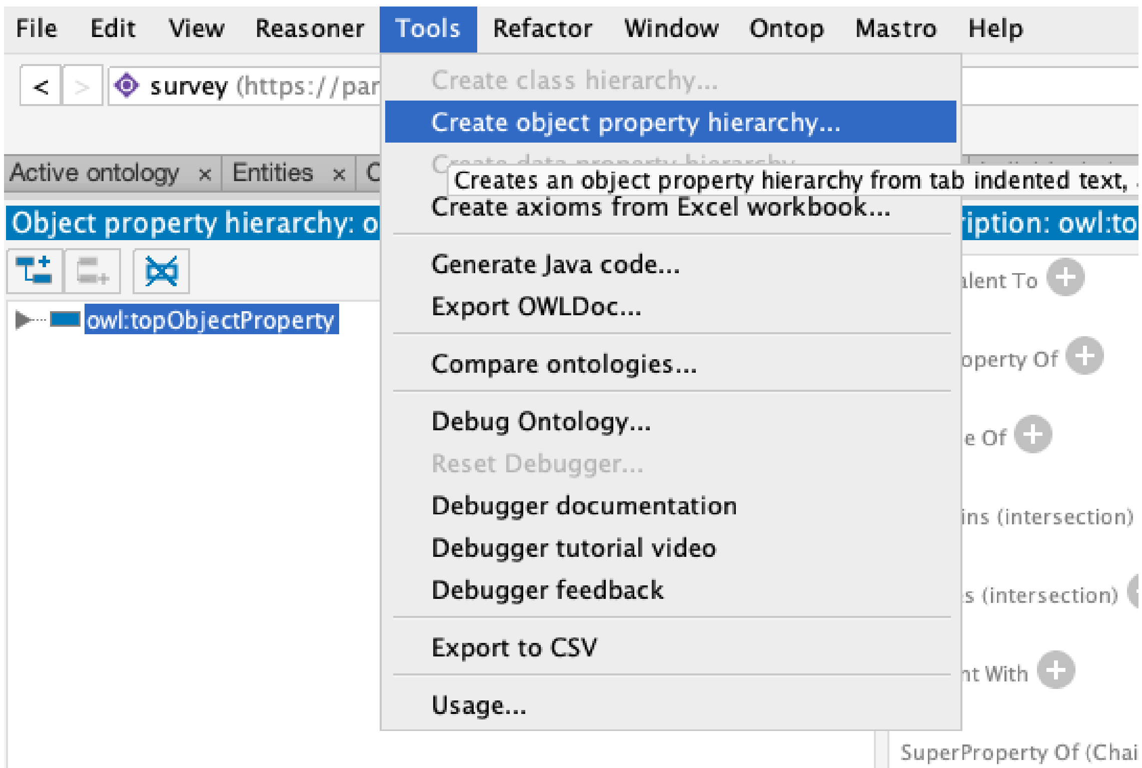



Figure 4.9: Tools Menu to Create Object Property Hierarchy

Also, the description view of object properties hierarchy allows the user to describe the object properties. The details that can be added or modified using the description view are “Equivalent To”, “SubProperty Of”, “Inverse Of”, “Domains (intersection)”, “Ranges (intersection)”, “Disjoint With”, and “SuperProperty Of (Chain)”.

The names of the object properties usually start with an “is” or a “has”. For instance, consider two classes: “Lady” (class 1) and “Parita” (class 2). To describe the link between class 1 (Lady) and class 2 (Parita) such that class 1 has value class 2 and class 2 is value of class 1, the properties “hasName” and “isNameOf” would be used respectively, as shown in Figure 4.10.

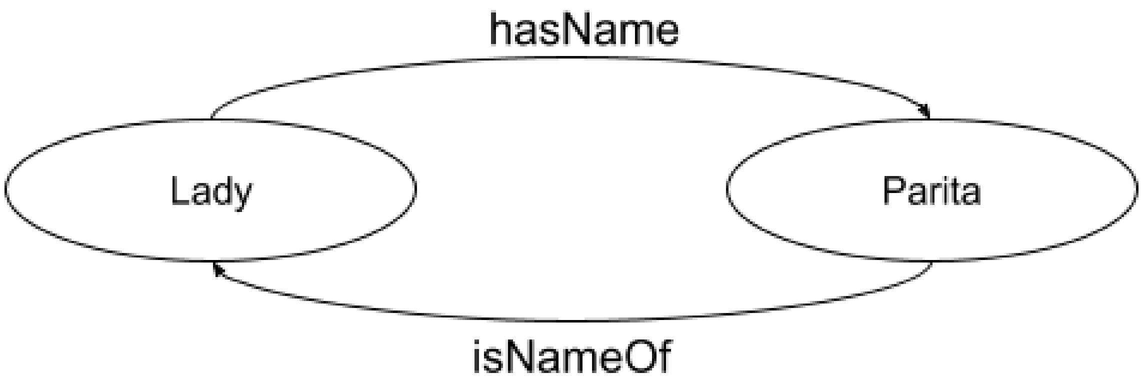



Figure 4.10: Object Properties for Classes "Lady" and "Parita"

Each object property might have a corresponding inverse property. For the above instance, it can be said that:

Lady hasName Parita **is equivalent to** Parita isNameOf Lady

Here, "hasName" is inverse of "isNameOf" and "isNameOf" is inverse of "hasName". Apart from inverse, domain and range[7] are two other important parts of description of object properties. According to Horridge et al. [**?**], in OWL, "domain and range are not constraints to be checked. They are axioms which are used by the reasoner to make inferences". As shown in Figure 4.11, "Lady" acts as the "Domain" and "Parita" acts as the "Range" for "hasName" whereas for "isNameOf", "Parita" is the "Domain" and "Lady" is the "Range".

Using the above method of naming and building object properties, some of the primary object properties are described for the survey ontology as shown in Figure 4.12 where as Table 4.2 shows the list of the object properties along with their domain and range classes (entities). Table 4.3 consists of the parent nodes and inverse properties of the object properties. The three parent nodes involved in object property hierarchy

[7]http://protegeproject.github.io/protege/views/object-property-description/

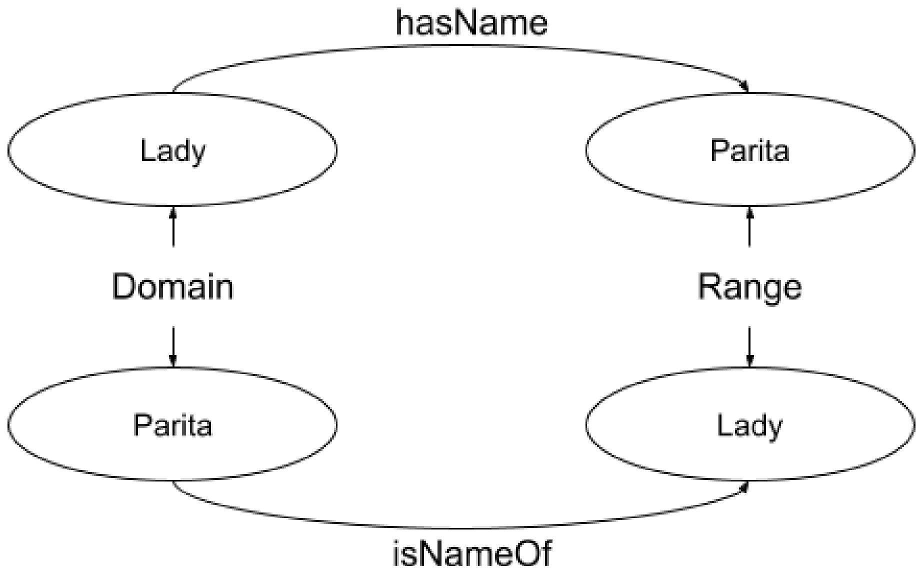



Figure 4.11: Domain and Range Classes for “hasName” and “isNameOf”

are “owl:TopObjectProperty”, “hasContent”, and “isContentOf”. Protégé has a built-in “Reasoner” that allows the users to determine inconsistencies of a class as well as to discover implicit information.

| Object Property | Domain | Range |
|---|---|---|
| hasSurveyQuestion | Survey | SurveyQuestions |
| hasLikertScale | LikertScaleQuestions | LikertScale |
| hasLikertScaleAnswer | Respondent | LikertScale |
| hasRespondedOn | Respondent | ResponseDate |
| respondedToSurvey | Respondent | Survey |
| hasOEAnswer_Change | OpenEndedQuestions | OpenEndedAnswers |
| hasOEAnswer_LikeBest | OpenEndedQuestions | OpenEndedAnswers |
| hasOEAnswer_Strengths | OpenEndedQuestions | OpenEndedAnswers |

Table 4.2: Object Properties and Their Domain and Range

### 4.3.3 Data Properties and Data Property Hierarchy

Furthermore, the “Data Properties” can also be described for ontology. “An ontology data property provides a relation to attach an entity instance to some literal datatype

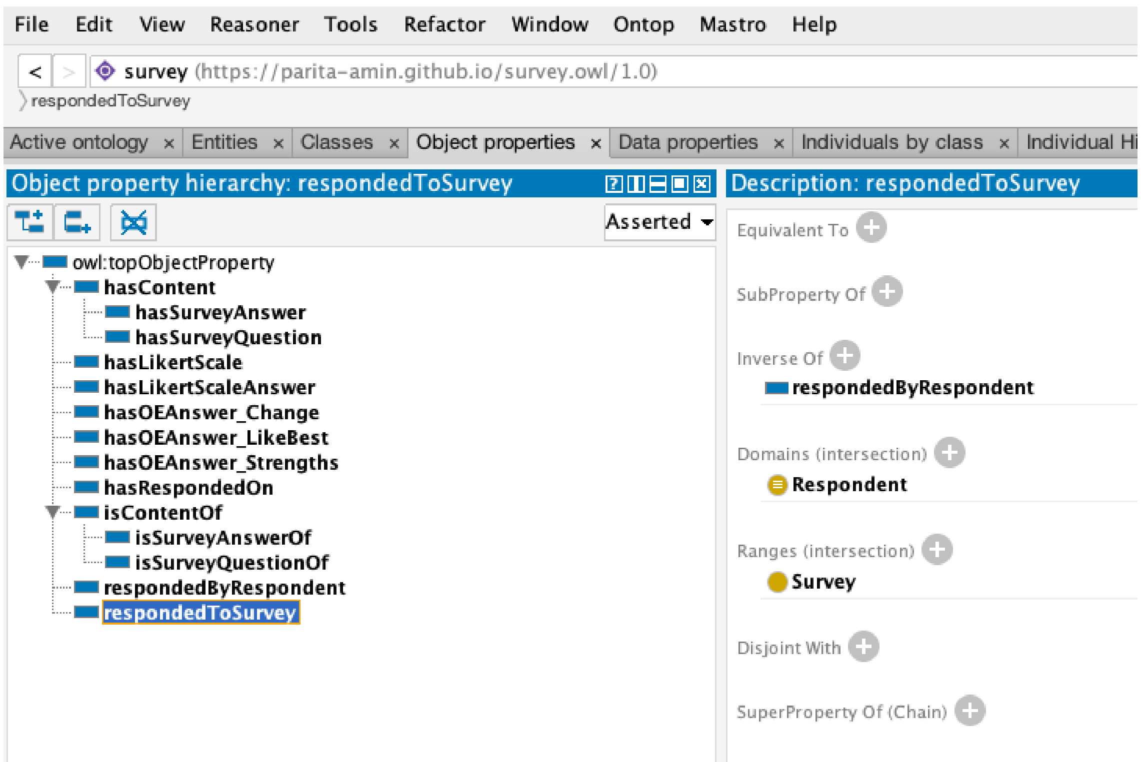



Figure 4.12: Object Property Hierarchies, Domain and Range for Classes

| Object Property | Parent Node | Inverse Property |
|---|---|---|
| hasContent | owl:TopObjectProperty | isContentOf |
| isContentOf | owl:TopObjectProperty | hasContent |
| hasLikertScale | owl:TopObjectProperty | |
| hasLikertScaleAnswer | owl:TopObjectProperty | |
| hasOEAnswer_Change | owl:TopObjectProperty | |
| hasOEAnswer_LikeBest | owl:TopObjectProperty | |
| hasOEAnswer_Strengths | owl:TopObjectProperty | |
| hasRespondedOn | owl:TopObjectProperty | |
| respondedByRespondent | owl:TopObjectProperty | respondedToSurvey |
| respondedToSurvey | owl:TopObjectProperty | respondedByRespondent |
| hasSurveyQuestion | hasContent | isSurveyQuestionOf |
| hasSurveyAnswer | hasContent | isSurveyAnswerOf |
| isSurveyQuestionOf | isContentOf | hasSurveyQuestion |
| isSurveyAnswerOf | isContentOf | hasSurveyAnswer |

Table 4.3: Object Properties with Their Parent Nodes and Inverse Properties

value (an RDF number, string or data for example) that is a measure or estimate of what that data property is about."[8] In Protégé, the "owl:topDataProperty" is the root node of the data properties' hierarchy. The data property hierarchy can be built by selecting a property in the Data Property Hierarchy view of the Data Property Views tab and then going to the tools menu as shown in Figure 4.13. The remaining steps are similar to the ones that are followed for building class and object property hierarchy (Figure 4.5, 4.6, and 4.7).

Data properties can be described using the description view of data properties of hierarchy which included details like "Equivalent To", "SubProperty Of", "Domains (intersection)", "Ranges", and "Disjoint With". Figure 4.14 shows the data properties described for the survey ontology. The parent node of all the data properties is "owl:topDataProperty" and the domain and range for all the data properties are "Respondent" and "xsd:positiveInteger" respectively. Table 4.4 consists of list of data properties along with their domain and range classes (entities).

[8]https://ddooley.github.io/docs/data-properties/

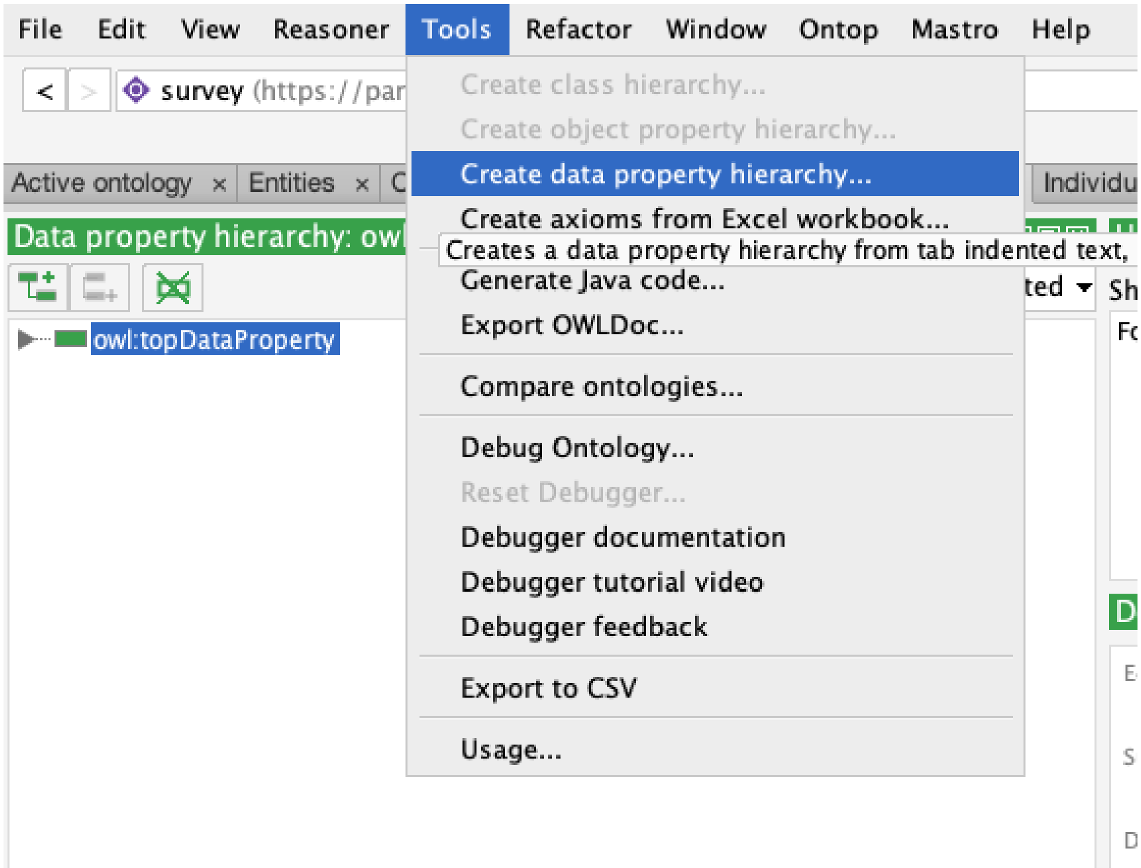



Figure 4.13: Tools Menu to Create Data Property Hierarchy

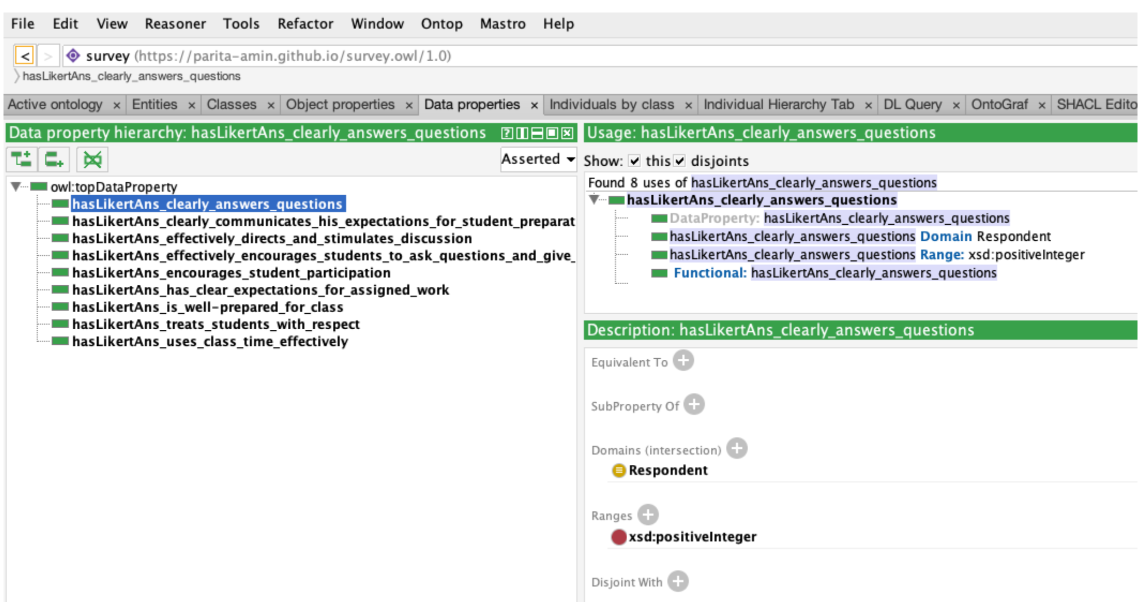



Figure 4.14: Data Property Hierarchies, Domain and Range for Classes

| Data Property | Domain | Range |
|---|---|---|
| hasLikertAns_is_well-prepared_for _class | Respondent | xsd:positiveInteger |
| hasLikertAns_clearly_communicates _his_expectations_for_student _preparation_and_participation | Respondent | xsd:positiveInteger |
| hasLikertAns_uses_class_time _effectively | Respondent | xsd:positiveInteger |
| hasLikertAns_has_clear_expectations _for_assigned_work | Respondent | xsd:positiveInteger |
| hasLikertAns_encourages_student _participation | Respondent | xsd:positiveInteger |
| hasLikertAns_clearly_answers _questions | Respondent | xsd:positiveInteger |
| hasLikertAns_treats_students _with_respect | Respondent | xsd:positiveInteger |
| hasLikertAns_effectively_directs _and_stimulates_discussion | Respondent | xsd:positiveInteger |
| hasLikertAns_effectively_encourages _students_to_ask_questions_and _give_answers | Respondent | xsd:positiveInteger |

Table 4.4: Data Properties and Their Domain and Range

### 4.3.4 Individuals

After describing the classes, object properties, data properties, and individuals, the next step is to determine and fix the errors and inconsistencies in the ontology using a Reasoner. The final class hierarchy of the OWL ontology in Protégé, after using the HermiT 1.4.3.456 reasoner, would be as shown in Figure 4.15.

The grey (all entities) and blue ("Survey" entity) back arrows in front of the yellow solid dot, which indicates class, show parent/child relationship other than that of SubClassOf. This feature can be enabled and disabled by selecting "Display relationship in class hierarchy" option from the "View" menu of Protégé. Figure 4.16 shows the relation indicated by the blue back arrow.

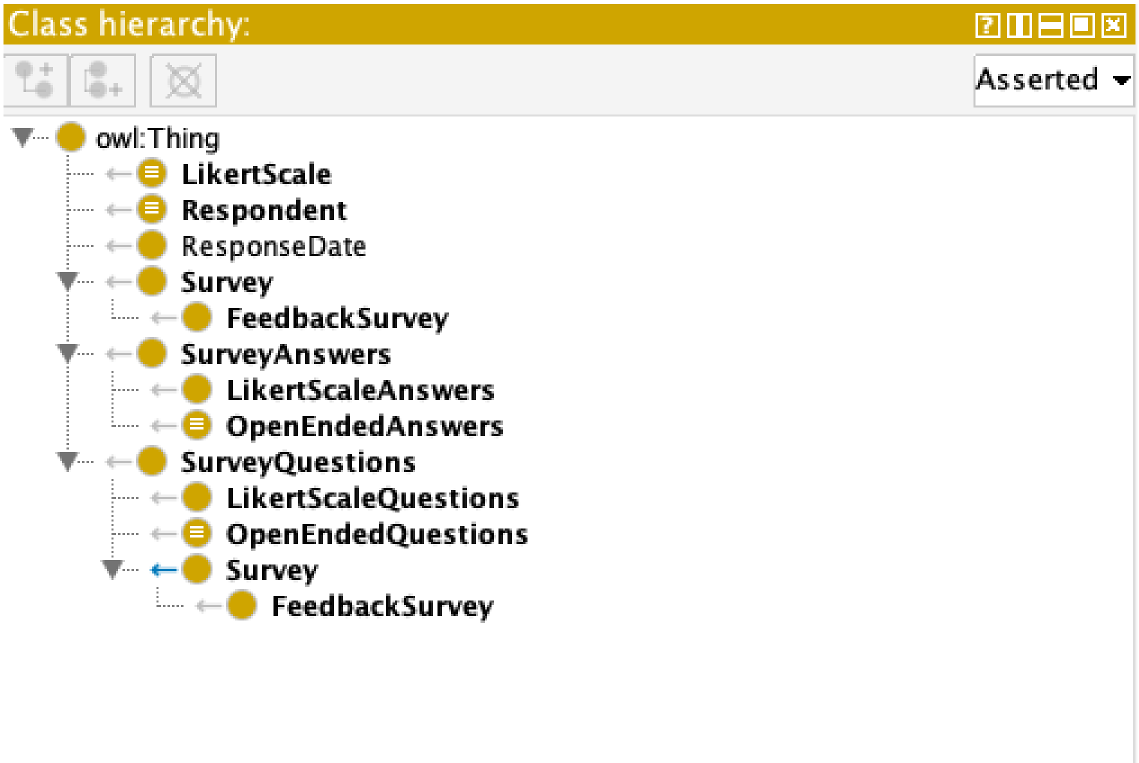



Figure 4.15: Class Hierarchy for the Final Version of Survey Ontology

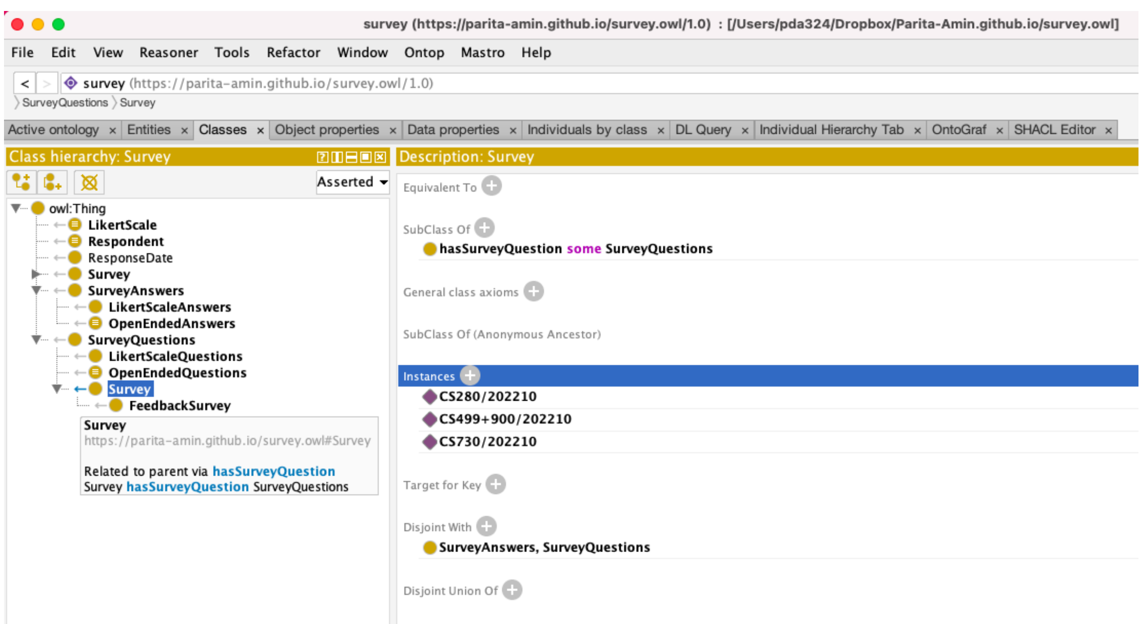



Figure 4.16: Parent/Child Relation indicated by "Blue Back" Arrow

## 4.4 Visualization Tools for OWL Ontology

The next step is visualization of the built OWL ontology using various visualization tools. Protégé supports plug-ins for some visualization tools like OntoGraf and OWLViz whereas tools like WebVOWL, RelFinder and OWLGrEd are available for use as web-based tools. This section focuses on the two tools, OntoGraf and WebVOWL, used for the visualization of the survey ontology in this research.

### 4.4.1 OntoGraf

Falconer [?] introduced OntoGraf as a Protégé plug-in for the visualization of the OWL ontologies. This tool helps the users to represent the ontologies, built using Protégé, by repetitively allowing and restricting the desired classes. OntoGraf offers various graph layout options for visualization like Grid Layout[9], Spring Layout, and Tree Layout[10]. Antoniazzi and Viola [?] stated that the "individuals of a class can be visualized in its tooltip, but this is uncomfortable when dealing with a high number of assertional statements". The graphs generated using OntoGraf can be exported as image files of various formats such as Portable Network Graphics (PNG), Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF) and as a dot file[11].

Figure 4.17 shows the graph generated using OntoGraf for the survey ontology. To start with the graph formation, the classes "Thing", "Domain_entity", "Independent_entity" and "Value" were selected followed by expansion of "Independent_entity" and "Value" classes by double-clicking them. This expansion adds all the classes that are connected to the classes. The classes connected includes both the subclasses as well as the ones linked through the object properties. Solid blue lines are used to denote the

[9] classes are arranged in a grid alphabetically
[10] both horizontal and vertical
[11] a Microsoft Word Template with the details of default settings

relationships between the classes and their subclasses whereas dashed lines are used to represent the object properties.

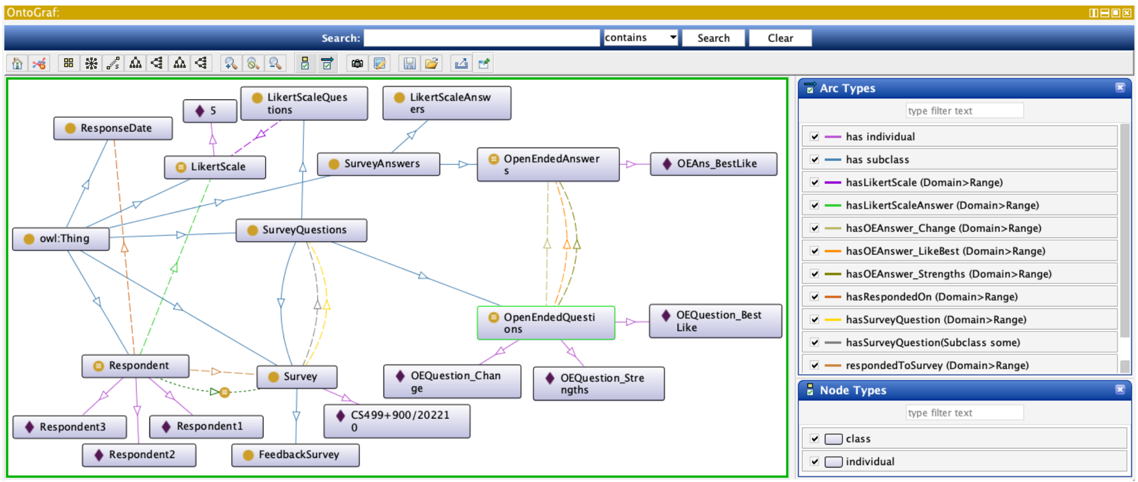



Figure 4.17: OntoGraf for Survey Ontology

### 4.4.2 WebVOWL

WebVOWL [?], the Web-based Visual Web Ontology Language (VOWL), one of the variety of forms in which the VOWL visualization tool is available, represents the "ontologies graphically using a force-directed graph layout" [?]. VOWL follows some ground rules for the representation of OWL ontologies which are:

1. Circles are used to denote classes. Also, different types are assigned different colors. For instance, OWL classes are denoted using "light blue circles".

2. Black solid lines are used to represent OWL object and datatype properties. The object properties use light blue labels where as the datatype properties are labelled in green.

3. The relationships between the classes and their subclasses are represented using dashed lines.

Figure 4.18 shows the graphical representation of the survey ontology using WebVOWL. The figure focuses on only some of the class-subclass relationships and some of the object properties. The metadata and the statistics of the node (circle) or the edge (line) can be retrieved by clicked on them. The entities (classes) and their relationships (object and datatype properties) can be presented or restricted using filters. The VOWL graph can be exported as an Scalable Vector Graphics (SVG) image file or a JavaScript Object Notation (JSON) code file.

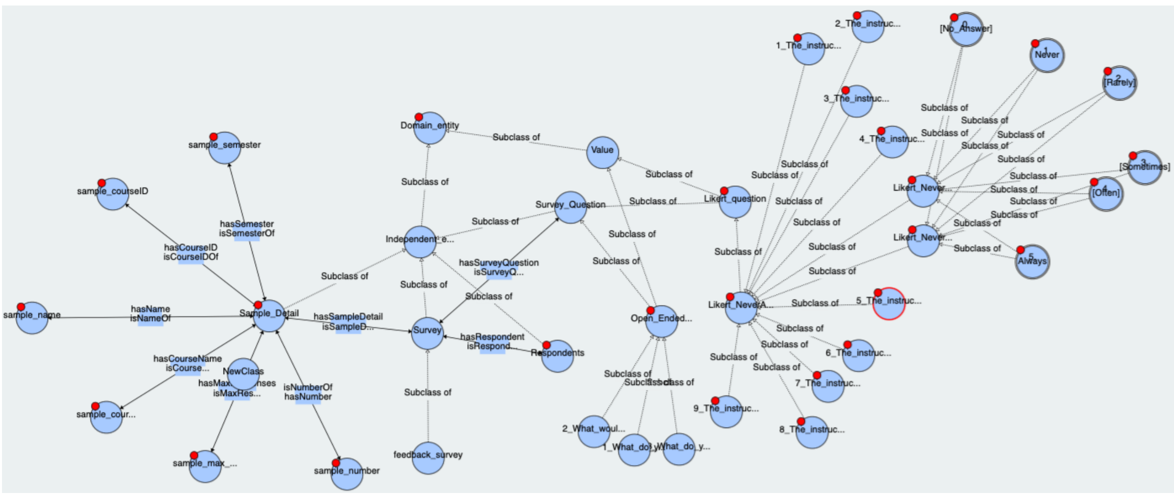



Figure 4.18: WebVOWL for Survey Ontology

# Chapter 5

# Conclusion and Future Work

This Chapter focuses on the prime offerings of this research work where one of the sections includes the conclusions drawn from the work done and the other discusses future work that can be done to succeed in this research.

## 5.1 Conclusion

This thesis work describes a new attempt on developing a linked open data instrument for feedback surveys. It focuses on the issues encountered and the user experience of various tools and references used for the modelling of the instrument and for data visualization. The thesis revolves around some fundamental topics like Semantic Web, Linked Open Data, RDF, and OWL ontologies. It also describes how and why Protégé has been used over other ontology development tools to describe survey vocabularies. The thesis can be considered as a guide for the researchers who are interested in exploring this lesser known aspect of semantic web.

The thesis includes a detailed background study on the fundamental topics, the usability of Protégé, and discussion on some tools for linked open data visualizations. Chapter 4 describes the steps involved in building the survey ontology using Protégé, various parts and facilities offered by Protégé, and the visual representation of the ontology toward the end.

## 5.2 Future Work

This thesis also offers a number of future development options like building a real-time linked open data application for feedback surveys that follows the linked data life cycle. Web Scraping (or Web Extraction) is a method of extracting desired data from the resources on the WWW into a separate file for data retrieval and data analysis. Data Extraction is one of the steps in the linked data life cycle that can be explored in future research to facilitate use of data that has already been published to the web with HTML tables.

Versioning of the survey ontology is a topic of interest. The ontology vocabularies (available in Appendix C) can be reused for appropriate applications but they may need to be updated over time. Hence, ontology versioning and version management of the survey instrument for this feedback survey, as well as for other types of surveys is an aspect of this research that can be explored.

Developing a data entry form for the feedback survey instrument that can be defined, in part, directly from the survey ontology would facilitate recording of survey responses collected offline.

# Appendix A

# Sample CSV Data

```
Likert Item,5 Always,4 [Mostly],3 [Sometimes],2 [Rarely],1 Never,Total
1. The instructor is well-prepared for class,29,16,8,0,0,53
2. The instructor clearly communicates his expectations for student
   preparation and participation,33,15,4,1,0,53
3. The instructor uses class time effectively,26,18,7,2,0,53
4. The instructor has clear expectations for assigned work,34,10,8,1,0,53
5. The instructor encourages student participation,45,6,2,0,0,53
6. The instructor clearly answers questions,31,16,5,1,0,53
7. The instructor treats students with respect,49,3,1,0,0,53
8. The instructor effectively directs and stimulates discussion,
   38,10,4,1,0,53
9. The instructor effectively encourages students to ask questions and
   give answers,43,8,2,0,0,53
Total,328,102,41,6,0,477
Weighted Average: 91% 'Always'
Response,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Avg,SD,Mode,Median
1,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
2,4,3,3,2,3,4,5,3,4,3.44,0.88,3,3
```

```
3,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
4,5,5,4,5,5,5,5,5,5,4.89,0.33,5,5
5,5,5,4,5,5,5,5,4,5,4.78,0.44,5,5
6,3,4,4,3,5,4,5,5,5,4.22,0.83,5,4
7,3,4,4,4,4,4,5,3,3,3.78,0.67,4,4
8,4,4,4,4,5,5,5,5,5,4.56,0.53,5,5
9,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
10,4,5,4,5,5,5,4,4,5,4.56,0.53,5,5
11,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
12,5,5,5,4,5,4,5,5,5,4.78,0.44,5,5
13,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
14,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
15,4,5,4,4,5,5,5,5,5,4.67,0.50,5,5
16,5,5,4,5,5,5,5,5,5,4.89,0.33,5,5
17,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
18,3,4,5,3,3,2,3,4,4,3.44,0.88,3,3
19,3,3,3,3,5,3,5,4,5,3.78,0.97,3,3
20,4,4,4,5,5,5,5,5,5,4.67,0.50,5,5
21,4,4,3,5,5,4,5,5,5,4.44,0.73,5,5
22,3,3,3,3,5,3,5,4,5,3.78,0.97,3,3
23,4,4,4,3,5,4,5,5,5,4.33,0.71,4,4
24,4,4,5,5,5,4,5,5,5,4.67,0.50,5,5
25,3,2,2,3,4,3,5,3,4,3.22,0.97,3,3
26,3,3,2,3,4,3,4,2,3,3.00,0.71,3,3
27,5,5,4,5,5,5,5,4,4,4.67,0.50,5,5
28,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5
```

29,4,4,4,3,5,4,5,5,4,4.22,0.67,4,4

30,4,5,4,5,5,4,5,5,4,4.56,0.53,5,5

31,5,5,5,5,5,4,5,5,5,4.89,0.33,5,5

32,5,5,5,5,4,5,5,4,5,4.78,0.44,5,5

33,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

34,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

35,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

36,4,4,3,4,5,5,5,5,5,4.44,0.73,5,5

37,5,5,5,5,5,5,5,4,5,4.89,0.33,5,5

38,4,4,3,4,5,3,5,4,5,4.11,0.78,4,4

39,5,5,5,4,5,4,5,5,4,4.67,0.50,5,5

40,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

41,5,4,5,5,5,5,5,5,5,4.89,0.33,5,5

42,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

43,4,5,4,5,5,5,5,5,5,4.78,0.44,5,5

44,4,5,4,5,4,4,5,4,5,4.44,0.53,4,4

45,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

46,3,4,4,4,4,4,5,3,4,3.89,0.60,4,4

47,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

48,4,4,4,5,5,4,4,5,5,4.44,0.53,4,4

49,4,4,3,4,5,5,5,5,5,4.44,0.73,5,5

50,5,5,4,4,5,4,5,5,5,4.67,0.50,5,5

51,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

52,5,5,5,5,5,5,5,5,5,5.00,0.00,5,5

53,5,5,5,5,5,4,5,5,5,4.89,0.33,5,5

Avg,4.40,4.51,4.28,4.45,4.81,4.45,4.91,4.60,4.77,4.58,0.21

SD,0.74 ,0.72 ,0.84 ,0.82 ,0.48 ,0.75 ,0.35 ,0.72 ,0.51

Mode,5 ,5 ,5 ,5 ,5 ,5 ,5 ,5 ,5

Median ,5 ,5 ,4 ,5 ,5 ,5 ,5 ,5 ,5

# Appendix B

# Expression of CSV Data in RDF

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/">
<rdf:Description rdf:about="http://survey.org/Question_1.
%20The%20instructor%20is%20well-prepared%20for%20class">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">29</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">16</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">8</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">0</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_2.
%20The%20instructor%20clearly%20communicates%20his%20expectations
%20for%20student%20preparation%20and%
20participation">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">33</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">15</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#int">4</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">1</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_3.
%20The%20instructor%20uses%20class%20time%20effectively">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">26</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">18</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">7</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">2</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_4.
%20The%20instructor%20has%20clear%20expectations%20for
%20assigned%20work">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">34</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">10</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">8</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">1</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_5.%20The
%20instructor%20encourages%20student%20participation">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">45</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">6</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">2</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">0</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_6.%20The
%20instructor%20clearly%20answers%20questions">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">31</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">16</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">5</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">1</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_7.%20The
%20instructor%20treats%20students%20with%20respect">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">49</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">3</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">1</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">0</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_8.%20The
%20instructor%20effectively%20directs%20and%20stimulates
%20discussion">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">38</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">10</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">4</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">1</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/Question_9.%20The
%20instructor%20effectively%20encourages%20students%20to
%20ask%20questions%20and%20give%20answers">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">43</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">8</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">2</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">0</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">53</Responses>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Question_Total">
  <Always xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">328</Always>
  <Mostly xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">102</Mostly>
  <Sometimes xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">41</Sometimes>
  <Rarely xmlns="http://survey.org/likert/">6</Rarely>
  <Never xmlns="http://survey.org/likert/">0</Never>
</rdf:Description>

<rdf:Description rdf:about="http://survey.org/likert/Total">
  <Responses xmlns="http://survey.org/likert/" rdf:datatype=
  "http://www.w3.org/2001/XMLSchema#int">477</Responses>
</rdf:Description>

</rdf:RDF>
```

# Appendix C

# OWL Classes to be Used to Describe Surveys

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.semanticweb.org/pda324/ontologies/survey"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     ontologyIRI="http://www.semanticweb.org/pda324/ontologies/survey"
     versionIRI="http://www.semanticweb.org/pda324/ontologies/survey/1.10
   ">
    <Prefix name="" IRI="http://www.semanticweb.org/pda324/
    ontologies/survey"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/
   >
    <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        <Class IRI="#Always"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Domain_entity"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Independent_entity"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Likert_NeverAlways5"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </Declaration>
```

```
    <Declaration>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Likert_question"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Never"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Open_Ended_question"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Respondents"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Sample_Detail"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Survey"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Survey_Question"/>
    </Declaration>
    <Declaration>
        <Class IRI="#Value"/>
    </Declaration>
    <Declaration>
        <Class IRI="#feedback_survey"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_courseID"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_course_name"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_max_responses"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_name"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_number"/>
    </Declaration>
    <Declaration>
        <Class IRI="#sample_semester"/>
    </Declaration>
    <Declaration>
        <Class IRI="#0"/>
    </Declaration>
    <Declaration>
```

```
        <Class IRI="#1"/>
    </Declaration>
    <Declaration>
        <Class IRI="#1_The_instructor_is_well-prepared_for_class"/>
    </Declaration>
    <Declaration>
        <Class IRI="#1_What_do_you_like_the_best_about_this_course?"/>
    </Declaration>
    <Declaration>
        <Class IRI="#2"/>
    </Declaration>
    <Declaration>
        <Class IRI="#2_The_instructor_clearly_communicates_his
         _expectations_for_student_preparation_and_participation"/>
    </Declaration>
    <Declaration>
        <Class IRI="#2_What_would_you_like_to_change_about_this_course?"/
   >
    </Declaration>
    <Declaration>
        <Class IRI="#3"/>
    </Declaration>
    <Declaration>
        <Class IRI="#3_The_instructor_uses_class_time_effectively"/>
    </Declaration>
    <Declaration>
        <Class IRI="#3_What_do_you_think_the_instructor's_greatest
         _strengths_are?"/>
    </Declaration>
    <Declaration>
        <Class IRI="#4"/>
    </Declaration>
    <Declaration>
        <Class IRI="#4_The_instructor_has_clear_expectations_for
         _assigned_work"/>
    </Declaration>
    <Declaration>
        <Class IRI="#5"/>
    </Declaration>
    <Declaration>
        <Class IRI="#5_The_instructor_encourages_student_participation"/>
    </Declaration>
    <Declaration>
        <Class IRI="#6_The_instructor_clearly_answers_questions"/>
    </Declaration>
    <Declaration>
        <Class IRI="#7_The_instructor_treats_students_with_respect"/>
    </Declaration>
    <Declaration>
        <Class IRI="#8_The_instructor_easily_directs_and_simulates
         _discussion"/>
    </Declaration>
```

```
    <Declaration>
        <Class IRI="#9_The_instructor_effectively_encourages_students
        _to_ask_questions_and_give_answers"/>
    </Declaration>
    <Declaration>
        <Class IRI="#[No_Answer]"/>
    </Declaration>
    <Declaration>
        <Class IRI="#[Often]"/>
    </Declaration>
    <Declaration>
        <Class IRI="#[Rarely]"/>
    </Declaration>
    <Declaration>
        <Class IRI="#[Sometimes]"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasCourseID"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasCourseName"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasMaxResponses"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasName"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasNumber"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasRespondent"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasSemester"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#hasSurveyQuestion"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isCourseIDOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isCourseNameOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isMaxResponseOf"/>
    </Declaration>
```

```
    <Declaration>
        <ObjectProperty IRI="#isNameOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isNumberOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isRespondentOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isSemesterOf"/>
    </Declaration>
    <Declaration>
        <ObjectProperty IRI="#isSurveyQuestionOf"/>
    </Declaration>
    <EquivalentClasses>
        <Class IRI="#Always"/>
        <Class IRI="#5"/>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#Never"/>
        <Class IRI="#1"/>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#0"/>
        <Class IRI="#[No_Answer]"/>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#2"/>
        <Class IRI="#[Rarely]"/>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#3"/>
        <Class IRI="#[Sometimes]"/>
    </EquivalentClasses>
    <EquivalentClasses>
        <Class IRI="#4"/>
        <Class IRI="#[Often]"/>
    </EquivalentClasses>
    <SubClassOf>
        <Class IRI="#Always"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Independent_entity"/>
        <Class IRI="#Domain_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Likert_NeverAlways5"/>
```

```
        <Class IRI="#Likert_question"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Likert_NeverAlways5_names"/>
        <Class IRI="#Likert_NeverAlways5"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
        <Class IRI="#Likert_NeverAlways5"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Likert_question"/>
        <Class IRI="#Survey_Question"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Likert_question"/>
        <Class IRI="#Value"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Never"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Open_Ended_question"/>
        <Class IRI="#Survey_Question"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Open_Ended_question"/>
        <Class IRI="#Value"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Respondents"/>
        <Class IRI="#Independent_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Sample_Detail"/>
        <Class IRI="#Independent_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Survey"/>
        <Class IRI="#Independent_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Survey_Question"/>
        <Class IRI="#Independent_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Value"/>
        <Class IRI="#Domain_entity"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#feedback_survey"/>
```

```
        <Class IRI="#Survey"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_courseID"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_course_name"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_max_responses"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_name"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_number"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#sample_semester"/>
        <Class IRI="#Sample_Detail"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#0"/>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#1"/>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#1_The_instructor_is_well-prepared_for_class"/>
        <Class IRI="#Likert_NeverAlways5"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#1_What_do_you_like_the_best_about_this_course?"/>
        <Class IRI="#Open_Ended_question"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#2"/>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#2_The_instructor_clearly_communicates_his
         _expectations_for_student_preparation_and_participation"/>
        <Class IRI="#Likert_NeverAlways5"/>
    </SubClassOf>
    <SubClassOf>
```

```
            <Class IRI="#2_What_would_you_like_to_change_about_this_course?"/
        >
            <Class IRI="#Open_Ended_question"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#3"/>
            <Class IRI="#Likert_NeverAlways5_numbers"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#3_The_instructor_uses_class_time_effectively"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#3_What_do_you_think_the_instructor's_greatest
            _strengths_are?"/>
            <Class IRI="#Open_Ended_question"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#4"/>
            <Class IRI="#Likert_NeverAlways5_numbers"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#4_The_instructor_has_clear_expectations_for
            _assigned_work"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#5"/>
            <Class IRI="#Likert_NeverAlways5_numbers"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#5_The_instructor_encourages_student
            _participation"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#6_The_instructor_clearly_answers_questions"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#7_The_instructor_treats_students_with_respect"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#8_The_instructor_easily_directs_and_simulates
            _discussion"/>
            <Class IRI="#Likert_NeverAlways5"/>
        </SubClassOf>
        <SubClassOf>
            <Class IRI="#9_The_instructor_effectively_encourages_students
            _to_ask_questions_and_give_answers"/>
            <Class IRI="#Likert_NeverAlways5"/>
```

```
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#[No_Answer]"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#[Often]"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#[Rarely]"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#[Sometimes]"/>
        <Class IRI="#Likert_NeverAlways5_names"/>
    </SubClassOf>
    <DisjointClasses>
        <Class IRI="#Always"/>
        <Class IRI="#Never"/>
        <Class IRI="#[No_Answer]"/>
        <Class IRI="#[Often]"/>
        <Class IRI="#[Rarely]"/>
        <Class IRI="#[Sometimes]"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Independent_entity"/>
        <Class IRI="#Value"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Likert_NeverAlways5_names"/>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Likert_NeverAlways5_names"/>
        <Class IRI="#Likert_NeverAlways5_numbers"/>
        <Class IRI="#1_The_instructor_is_well-prepared_for_class"/>
        <Class IRI="#2_The_instructor_clearly_communicates_his
        _expectations_for_student_preparation_and_participation"/>
        <Class IRI="#3_The_instructor_uses_class_time_effectively"/>
        <Class IRI="#4_The_instructor_has_clear_expectations_for
        _assigned_work"/>
        <Class IRI="#5_The_instructor_encourages_student
        _participation"/>
        <Class IRI="#6_The_instructor_clearly_answers_questions"/>
        <Class IRI="#7_The_instructor_treats_students_with_respect"/>
        <Class IRI="#8_The_instructor_easily_directs_and_simulates
        _discussion"/>
        <Class IRI="#9_The_instructor_effectively_encourages_students
        _to_ask_questions_and_give_answers"/>
    </DisjointClasses>
    <DisjointClasses>
```

```
        <Class IRI="#Likert_question"/>
        <Class IRI="#Open_Ended_question"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Respondents"/>
        <Class IRI="#Sample_Detail"/>
        <Class IRI="#Survey"/>
        <Class IRI="#Survey_Question"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#Sample_Detail"/>
        <Class IRI="#Survey"/>
        <Class IRI="#Survey_Question"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#sample_courseID"/>
        <Class IRI="#sample_course_name"/>
        <Class IRI="#sample_max_responses"/>
        <Class IRI="#sample_name"/>
        <Class IRI="#sample_number"/>
        <Class IRI="#sample_semester"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#sample_courseID"/>
        <Class IRI="#sample_max_responses"/>
        <Class IRI="#sample_name"/>
        <Class IRI="#sample_number"/>
        <Class IRI="#sample_semester"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#0"/>
        <Class IRI="#1"/>
        <Class IRI="#2"/>
        <Class IRI="#3"/>
        <Class IRI="#4"/>
        <Class IRI="#5"/>
    </DisjointClasses>
    <DisjointClasses>
        <Class IRI="#1_What_do_you_like_the_best_about_this_course?"/>
        <Class IRI="#2_What_would_you_like_to_change_about_this
        _course?"/>
        <Class IRI="#3_What_do_you_think_the_instructor's_greatest
        _strengths_are?"/>
    </DisjointClasses>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasCourseID"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasCourseName"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
```

```
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasMaxResponses"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasName"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasNumber"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasRespondent"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasSampleDetail"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasSemester"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#hasSurveyQuestion"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isCourseIDOf"/>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isCourseNameOf"/>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isMaxResponseOf"/>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isNameOf"/>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isNumberOf"/>
        <ObjectProperty IRI="#isSampleDetailOf"/>
    </SubObjectPropertyOf>
    <SubObjectPropertyOf>
        <ObjectProperty IRI="#isRespondentOf"/>
        <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    </SubObjectPropertyOf>
```

```
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isSampleDetailOf"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isSemesterOf"/>
    <ObjectProperty IRI="#isSampleDetailOf"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
    <ObjectProperty IRI="#isSurveyQuestionOf"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasCourseID"/>
    <ObjectProperty IRI="#isCourseIDOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasCourseName"/>
    <ObjectProperty IRI="#isCourseNameOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasMaxResponses"/>
    <ObjectProperty IRI="#isMaxResponseOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasName"/>
    <ObjectProperty IRI="#isNameOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasNumber"/>
    <ObjectProperty IRI="#isNumberOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasRespondent"/>
    <ObjectProperty IRI="#isRespondentOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasSampleDetail"/>
    <ObjectProperty IRI="#isSampleDetailOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasSemester"/>
    <ObjectProperty IRI="#isSemesterOf"/>
</InverseObjectProperties>
<InverseObjectProperties>
    <ObjectProperty IRI="#hasSurveyQuestion"/>
    <ObjectProperty IRI="#isSurveyQuestionOf"/>
</InverseObjectProperties>
<ObjectPropertyDomain>
    <ObjectProperty IRI="#hasCourseID"/>
    <Class IRI="#Sample_Detail"/>
</ObjectPropertyDomain>
```

```
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasCourseName"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasMaxResponses"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasName"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasNumber"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasRespondent"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasSampleDetail"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasSemester"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#hasSurveyQuestion"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isRespondentOf"/>
        <Class IRI="#Respondents"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isSampleDetailOf"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyDomain>
    <ObjectPropertyDomain>
        <ObjectProperty IRI="#isSurveyQuestionOf"/>
        <Class IRI="#Survey_Question"/>
    </ObjectPropertyDomain>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasCourseID"/>
        <Class IRI="#sample_courseID"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasCourseName"/>
        <Class IRI="#sample_course_name"/>
    </ObjectPropertyRange>
```

```
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasMaxResponses"/>
        <Class IRI="#sample_max_responses"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasName"/>
        <Class IRI="#sample_name"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasNumber"/>
        <Class IRI="#sample_number"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasRespondent"/>
        <Class IRI="#Respondents"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasSampleDetail"/>
        <Class IRI="#Sample_Detail"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasSemester"/>
        <Class IRI="#sample_semester"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#hasSurveyQuestion"/>
        <Class IRI="#Survey_Question"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isRespondentOf"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isSampleDetailOf"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyRange>
    <ObjectPropertyRange>
        <ObjectProperty IRI="#isSurveyQuestionOf"/>
        <Class IRI="#Survey"/>
    </ObjectPropertyRange>
    <DisjointObjectProperties>
        <ObjectProperty IRI="#hasCourseID"/>
        <ObjectProperty IRI="#hasCourseName"/>
        <ObjectProperty IRI="#hasMaxResponses"/>
        <ObjectProperty IRI="#hasName"/>
        <ObjectProperty IRI="#hasNumber"/>
        <ObjectProperty IRI="#hasSemester"/>
    </DisjointObjectProperties>
    <DisjointObjectProperties>
        <ObjectProperty IRI="#hasRespondent"/>
        <ObjectProperty IRI="#hasSampleDetail"/>
        <ObjectProperty IRI="#hasSurveyQuestion"/>
```

```
            <ObjectProperty IRI="#isRespondentOf"/>
            <ObjectProperty IRI="#isSampleDetailOf"/>
            <ObjectProperty IRI="#isSurveyQuestionOf"/>
        </DisjointObjectProperties>
        <DisjointObjectProperties>
            <ObjectProperty IRI="#isCourseIDOf"/>
            <ObjectProperty IRI="#isCourseNameOf"/>
            <ObjectProperty IRI="#isMaxResponseOf"/>
            <ObjectProperty IRI="#isNameOf"/>
            <ObjectProperty IRI="#isNumberOf"/>
            <ObjectProperty IRI="#isSemesterOf"/>
        </DisjointObjectProperties>
</Ontology>

<!-- Generated by the OWL API (version 4.5.9.2019-02-01T07:24:44Z) https://github.com/owlcs/owlapi -->
```

# Bibliography


[1] Sareh Aghaei, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani. Evolution of the World Wide Web: From WEB 1.0 TO WEB 4.0. *International Journal of Web and Semantic Technology*, 3(1):1–10, 2012.

[2] Emhimed Alatrish. Comparison Some of Ontology Editors. *Management Information Systems*, 8(2):18–24, 2013.

[3] Francesco Antoniazzi and Fabio Viola. RDF Graph Visualization Tools: A Survey. In *Proceedings of the 23rd Conference of Open Innovations Association (FRUCT)*, pages 25–36, 2018.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Conference on the Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, pages 722–735. Springer-Verlag, Berlin, Heidelberg, 2007.

[5] Florian Bauer and Martin Kaltenböck. Linked Open Data: The Essentials - A Quick Start Guide for Decision Makers. *Edition Mono/Monochrom*, Vienna, 2011.

[6] Dave Beckett and Brian McBride. RDF/XML Syntax Specification (Revised). *W3C Recommendation*, 2004. https://www.w3.org/TR/REC-rdf-syntax/ , accessed August 14, 2019.

[7] David Beckett. RDF 1.1 N-Triples. *W3C Recommendation*, 2014. https://www.w3.org/TR/2014/REC-n-triples-20140225/ , accessed October 29, 2019.

[8] Tim Berners-Lee. Linked Data. *W3C Recommendation*, 1996. https://www.w3.org/DesignIssues/LinkedData , accessed January 23, 2019.

[9] Tim Berners-Lee. Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web. *RFC 1630*, June 1998. https://www.rfc-editor.org/info/rfc1630 , accessed September 24, 2020.

[10] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. The World Wide Web. *Communications of the ACM*, 37(8):76–83, 1994.

[11] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. *Internet Engineering Task Force, RFC 2396*, August 1998. https://www.rfc-editor.org/info/rfc2396 , accessed October 01, 2020.

[12] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. *RFC 3986*, January 2005. https://www.rfc-editor.org/info/rfc3986 , accessed October 02, 2020.

[13] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Harper San Francisco, 1999.

[14] Christian Bizer. The Emerging Web of Linked Data. *IEEE Intelligent Systems*, 24(5):87–92, 2009.

[15] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal of Semantic Web Information Systems*, 5:1–22, 2009.

[16] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data: The Story So Far. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227. IGI global, USA, 2011.

[17] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked Data on the Web (LDOW2008). In *Proceedings of the 17th International Conference on World Wide Web*, pages 1265–1266. ACM, 2008.

[18] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia-A Crystallization Point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[19] Nupur Choudhury. World Wide Web and Its Journey from Web 1.0 to Web 4.0. *International Journal of Computer Science and Information Technologies*, 5(6):8096–8100, 2014.

[20] Richard Cyganiak, Andreas Harth, and Aidan Hogan. N-Quads: Extending N-Triples with Context. *W3C Recommendation*, pages 1–41, 2008.

[21] Richard Cyganiak, David Wood, Markus Lanthaler, Graham Klyne, Jeremy J Carroll, and Brian McBride. RDF 1.1 Concepts and Abstract Syntax. *W3C Recommendation*, 25(02):1–22, January 2014. http://hdl.handle.net/10421/2427 , accessed January 05, 2020.

[22] Ivan Ermilov, Sören Auer, and Claus Stadler. CSV2RDF: User-Driven CSV to RDF Mass Conversion Framework. In *Proceedings of the International Symposium on Applied Electromagnetics and Mechanics*, volume 13, pages 4–6. ACM, 2013.

[23] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing Wikidata to the Linked Data Web. In *Proceedings of the International Semantic Web Conference*, pages 50–65. Springer, 2014.

[24] Sean Falconer. Ontograf Protege Plugin. *Available at: http://protegewiki.stanford.edu/wiki/OntoGraf*, 2010.

[25] Thomas R Gruber. Toward Principles for the Design of Ontologies used for Knowledge Sharing? *International Journal of Human-Computer Studies*, 43(5):907–928, 1995.

[26] Lushan Han, Tim Finin, Cynthia Parr, Joel Sachs, and Anupam Joshi. RDF123: From Spreadsheets to RDF. In *Proceedings of the International Semantic Web Conference*, pages 451–466. Springer, 2008.

[27] Albert L. Harris and Alan Rea. Web 2.0 and Virtual World Technologies: A Growing Impact on IS Education. *Journal of Information Systems Education*, 20(2):137–144, 2009.

[28] Tom Heath and Christian Bizer. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, 2011.

[29] Daryl Hepting. Student Feedback Instrument. *Available at: `http://www2.cs.uregina.ca/~hepting/assets/teaching/pdf/feedback-instrument.pdf`*, 2020.

[30] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, and Chris Wroe. A Practical Guide to Building OWL Ontologies using Protégé 4 and CO-ODE Tools Edition 1.3. *The University of Manchester*, March 2011.

[31] Elin K Jacob. Ontologies and the Semantic Web. *Bulletin of the American Society for Information Science and Technology*, 29(4):19–19, 2003.

[32] Andreas Langegger and Wolfram Wob. XLWrap–Querying and Integrating Arbitrary Spreadsheets with SPARQL. In *Proceedings of the International Semantic Web Conference*, pages 359–374. Springer, 2009.

[33] Steffen Lohmann, Vincent Link, Eduard Marbach, and Stefan Negru. WebVOWL: Web-based Visualization of Ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 154–158. Springer, 2014.

[34] Frank Manola, Eric Miller, and Brian McBride. RDF Primer. *W3C Recommendation*, 10(1):1–6, 2004. https://www.w3.org/TR/2004/REC-rdf-primer-20040210/ , accessed December 2017.

[35] Brian McBride. The Resource Description Framework (RDF) and Its Vocabulary Description Language RDFS. In *Handbook on Ontologies*, pages 51–65. Springer, Berlin, 2004.

[36] Axel-Cyrille Ngonga Ngomo, Sören Auer, Jens Lehmann, and Amrapali Zaveri. Introduction to Linked Data and Its Life Cycle on the Web. In *Proceedings of Reasoning Web 2014: Reasoning on the Web in the Big Data Era*, pages 1–99, Cham. Springer International, September 2014.

[37] Jakob Nielsen. Usability Engineering. *Morgan Kaufmann Publichers Inc., San Franscisco, USA*, pages 23–48, 1994.

[38] Whitney Quesenbery and Whitney Interactive Design. Dimensions of Usability: Defining the Conversation, Driving the Process. In *UPA 2003 Conference*, pages 23–27, 2003.

[39] Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.