# Generalizing Concept Aggregation Framework and Automating Property Binding For CAFSIAL

by

Muhammad Asif Raza Khan

2008-NUST-MS PHD- CSE-13

MS-08 (SE)

Submitted to the Department of Computer Engineering in fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
SOFTWARE ENGINEERING

Thesis Supervisor

Dr Aasia Khanum

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

2012


In the name of Allah, the most

Beneficent and the most Merciful

# DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor (Dr Aasia Khanum). All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Muhammad Asif Raza Khan

# TABLE OF CONTENTS

# REFERENCES


[1] Tim Berner Lee, J. Hendler, and O. Lassila, “The Semantic Web”. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[2] Jon Bosak and Tim Bray, XML and the Second Generation Web

[3] http://www.wikipedia.org/RDF

[4] http://www.w3.org/TR/owl-features/

[5] http://www.w3.org/TR/rdf-sparql-query/

[6] http://dbpedia.org

[7] http://www.foaf-project.org/

[8] http://www.heppnetz.de/projects/goodrelations/

[9] http://en.wikipedia.org/wiki/GoPubMed

[10] http://en.wikipedia.org/wiki/GoPubMed

[11] Tim Berner Lee “Linked Data Design Issues” http://www.w3.org/DesignIssues /LinkedData.html.

[12] http://linkeddata.org

[13] LOD cloud

[14] A. Latif, M.T. Afzal, A.U. Saeed, P. Hoefler and K. Tochtermann “CAF-SIAL: Concept Aggregation Framework for Structuring Informational Aspects of Linked Open Data”.

[15] http://en.wikipedia.org/wiki/Agile_software_development

[16] Linked Data – The Story So Far
Christian Bizer, Tom Heath, Tim Berners-Lee

[17] How will we interact with the Web of Data?
Tom Heath

[18] http://www.faviki.com

[19] DBpedia Mobile: A LocationEnabled Linked Data Browser
Christian Becker, Christian Bizer

[20] http://www.bbc.co.uk/music/beta/

[21] M. Dzbor, J. Domingue, and E. Motta. "Magpie – towards a semantic web browser".

[22] http://magpie.kmi.open.ac.uk/

[23] D. Huynh, S. Mazzocchi, and D. Krager. "Piggy Bank: Experience the Semantic Web Inside Your Web Browser".

[24] D. Quan, D. Huynh, and D.R. Karger. "Haystack: A Platform for Authoring End User Semantic Web Applications".

[25] M.C. Schraefel, M. Karam, and S. Zhao "mSpace: interaction design for user-Determined, adaptable domain exploration in hypermedia".

[26] Hyv¨onen, E. Junnila, M. Kettula, S. Makel¨, E. Saarela, S. Salminen, M. Syreeni, A. Valo, and A. Viljanen. "Finnish Museums on the Semantic Web. User's Perspective on Museum Finland".

[27] T.B. Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. "Tabulator: Exploring and Analyzing linked data on the Semantic Web".

[28] S.F.C. deAraújo and D. Schwabe. "Explorator: A tool for exploring RDF data through direct manipulation".

[29] A. Russell, P. R. Smart, and D. Braines. "NITELIGHT: A Graphical Tool for Semantic Query Construction".

[30] M. Hildebrand, J.V. Ossenbruggen and L. Hardman. "/facet: A Browser for Heterogeneous Semantic Web Repositories".

[31] A. Harth, J. Umbrich, A. Hogan, and S. Decker. "YARS2:A Federated Repository for Querying Graph Structured Data from the Web"

[32] http://www.dbpedia.org/snorql

[33] A Platform for Scalable, Collaborative, Structured Information Integration.

Kurt Bollacker, Patrick Tufts, Tomi Pierce, Robert Cook

.

# CHAPTER 1: INTRODUCTION

## 1.1 Semantic Web

The Semantic Web is development of web. It comprises of data that will allow computer to understand the details or meanings, of content present on the Internet. Semantic Web was introduced by Tim Berners-Lee, the originator of the World Wide Web (www). He administers the progress of Semantic Web. Lee defines the Semantic Web as A web of data that can be used and processed by computers.

The Semantic Web is not a new Web but a new development in present web, in which information is defined with respect to some rules. Web pages contain information inside different type of tags along with some metadata and are interlinked with each other through hyperlinks. Aim of semantic web is to put together this information in machine readable format so that computers can also understand it. In the near future, machines will be able to use and recognize the data. Currently the machines only display this data. The Semantic Web will enable machines to comprehend semantic documents and data [1].

In order to make web understandable for machines different new technologies have been introduced. These technologies together form world of Semantic Web. Here we briefly look at some of the main technologies that are fundamental elements of the semantic web.

### 1.1.1 Extensible Markup Language (XML) [2]

XML is a language used for creating the documents and information into machine readable format. Information encoded using XML can be transferred over the Internet, across different applications and platforms. Although XML has been designed for documents, it is widely used for arbitrary data structures. In XML users make tags such as <CNIC No>, <Lane No> etc. These tags are then used in scripts and programs by developers and programmers. XML allows adding arbitrary data structures to documents but does not say anything about the semantics of these structures.

### 1.1.2 Resource Description Framework (RDF)

RDF is a framework used to define a resource. It has been designed by World Wide Web Consortium (W3C). RDF data model is similar to other data models like Entity Relationship Model. A resource is defined in form of a triple known as an RDF triple. An RDF triple consists of a subject and an object. Between the subject and the object is a Predicate that expresses the relationship between them e.g. in the statement "Pakistan Flag has the color Green", "Pakistan Flag" is a subject, "has the color" is predicate showing the relationship and "Green" is the object. Subject and Objects are represented by a URI.

URI stands for Universal Resource Identifier. It is used to identify a resource. It is similar to URL. A URL refers to a web page while a URI points to a resource. Different URI's can reside inside a web page. URL of College of E&ME is "http://www.ceme.nust.edu.pk", while URI of logo of College of E&ME is "http://www.ceme.nust.edu.pk/logo.jpg".
An RDF triple is a directed graph whose nodes and arcs are labeled by URI's [3].

### 1.1.3 Web Ontology Language (OWL)

OWL is a web language used to describe ontology. Ontology describes structure of any resource type. The OWL Web Ontology Language has been developed for use by programs that require to process the information and not its presentation to users. OWL allows computers to process the data in XML and RDF. [4].

### 1.1.4 SPARQL

SPARQL stands for SPARQL protocol and RDF Query Language. SPARQL is used to make query on distributed data sources. These queries are made on the data which is in RDF form and the result is also in RDF form. SPARQL can also query for graphs and additional information along with these graphs. SPARQL also allows the testing of a graph and constraints in it. The output of SPARQL queries are the results in form of datasets or in RDF graph. [5].

## 1.2 Semantic Web Projects

A number of Semantic Web Projects have been introduced since introduction of Semantic Web. We will briefly discuss some of these projects in this section.

### 1.2.1 DBpedia [6]

DBpedia is an attempt to make public semantic data mined from Wikipedia[1]. The data is presented in RDF and made available for public on the Web. This data can be inferred, reused, queried and extended in other data sources. DBpedia serves as a hub for linked data.

### 1.2.2 Friend of a Friend (FOAF) [7]

FOAF is an interesting application of semantic web. It describes relationships between people to people and things around them. FOAF is technology for linking social Web sites, and the people. FOAF is move towards a Web where people can select the web pages and applications, while remaining in contact with the people with different choice to applications and web pages. FOAF allows to link information from different pages, pass it on , and use it as the user wants.

1. http://en.wikipedia.org

### 1.2.3 Good Relations for E-Commerce [8]

"Good Relations" is a glossary for presenting all of the information about products and services on the internet. By adding a bit of extra code to Web content, businessmen and companies can make convince the public for starting business. People can extract the information from such web sites very easily. Hence, this project is an application of semantic web for Business Community.

### 1.2.4 GoPubMed [9]

It is a semantic web application for information related to life sciences. It makes use of different extracts from medical research papers. People can go through these papers and can extract information. This project has thousands of members which help in converting this data into semantic form. In future the work will be extended to ontologies for improving the results.

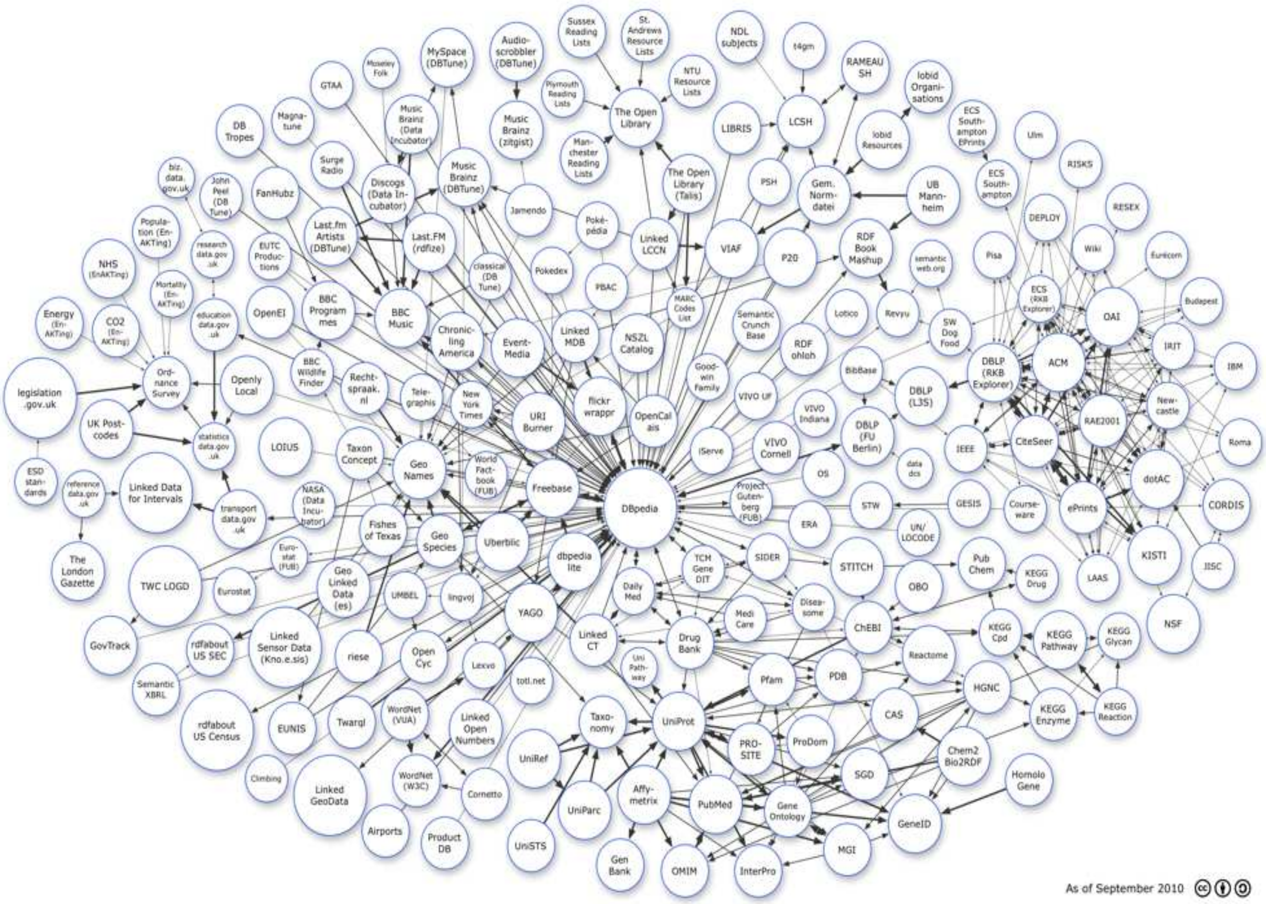
### 1.2.4 Semantically-Interlinked Online Communities (SIOC) [10]

This project is about linking the information about online groups of people. SIOC contains information in form of ontologies and RDF. It is being used in many different applications on internet. It makes use of FOAF.

### 1.2.5 Linked Open Data (LOD)

Linked Data Project is a way for connecting the different data sets and making these data sets available for the people on internet. It makes use of RDF and URI's. Using this methodology the information can be connected on the internet in such a way that other people can use it/ Four rules for linked data defined by Tim Berners Lee are:- [11].

1. Things are defined in URI's.
2. HTTP is used for allowing the people to search the data.




3. When the data is searched, useful information is displayed.

4. Additional links are included so that more information can be searched.

The Linking Open Data is a community project which allow people to publish their data in a semantic form. LOD converts the information in semantic form and also interlinks the data. [12]. According to September 2010 version of LOD Cloud Diagram, it is composed of 203 connected datasets which have 25 billion RDF triples and are linked by 395 million RDF links. It is quite clear from Figure 1 that DBpedia serves as a hub for linked data.

Figure 1: LOD Cloud 2010 [13]

### 1.2.6 Motivation

Linked data is enabling us to connect different data sources over the Internet. This project of semantic web is still in its early days. There is tremendous potential for research work to be carried out in this area.  Three types of research problems are available inside LOD project. First, conversion of existing data into semantic web format, for opening it on the web. Second, extraction and processing of the present data for LOD applications. And the third is presentation of the data extracted from LOD to the common users in organized form. In current experiment the issue of presentation of data extracted from LOD resources will addressed.

### 1.3 Background

A number of semantic web research applications have been developed in recent past addressing semantic web problems. Extraction of data from source and its presentation to the users is an important problem in LOD. CAFSIAL [14] is a research based application which addresses the same problem. It stands for Concept Aggregation Frame Structuring Information Aspects from Linked Open Data.

CAFSIAL is currently operational on web with only one type of resource “Person”. It has been populated with data extracted from DBpedia dumps, which are freely available on DBpedia website. Mechanism adopted for CAFSIAL is based on a “Concept Aggregation Framework”. According to this framework, for any “Informational Aspect” of a resource only “Related Properties” should be selected. This will result in presentation of information in a much organized way to the users. Using this bottom to top approach, information presented to users is organized as shown in Figure 2.

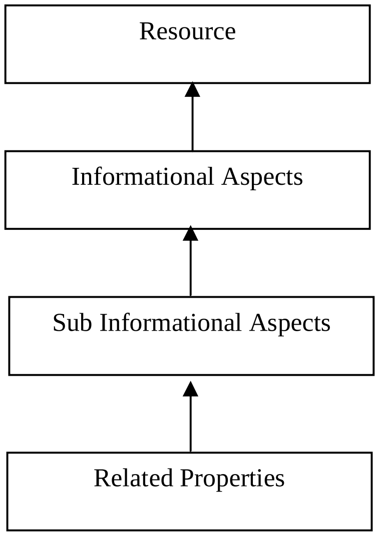



Figure 2: Information Presentation Hierarchy

However, CAFSIAL is at an early stage, containing information about only one type of resource i.e. “Persons”. Also, the mechanism adopted for selection of related properties, binding of related properties to sub informational aspects and sub informational aspects to informational aspects is manual. In order to make CAFSIAL a complete application, we need to shift it from manual mode of processing to auto mode.

### 1.4 Problem Statement

The aim of this work is to make CAFSIAL an application for a common user. In order to achieve this aim the following tasks must be achieved:-

a. Formulate a generalized mechanism for inclusion of any new resource in CAFSIAL.
b. Develop an algorithm for binding of similar properties to sub informational aspects and sub informational aspects to informational aspects.
c. Presentation of results to users in an appealing way.

d. Compare proposed system with other existing systems.

### 1.4.1 Methodology

This is a research project where modest information about solving the above mentioned problems exists. In order to try a variety of techniques before reaching to a final solution or algorithm an “Agile Software Development Methodology” [15] was followed for the work. In this methodology the different steps can be started in parallel. Some type of delay in one step does not affect other tasks.

### 1.4.2 Project Scope

Scope of this project is not just limited to CAFSIAL. A number of LOD applications have been developed / under development. Different approaches are being adopted for clustering of semantic data. The method of grouping similar properties that will be formulated for CAFSIAL may be used in other applications in future. The algorithm for clustering of information will need further research and improvement until it becomes a state of the art method for clustering of semantic data. The application can be linked with other semantic applications such as “Google maps” and “Geonames” for providing vaster information to the users in future.

## 1.5 Summary

This chapter introduced semantic web and its application in today’s technology. We described main elements of semantic web that are necessary for any semantic web project. A brief study of some of the semantic web projects has been presented that puts light on vastness of the field. Linked Open Data is one of the most important semantic web projects which aims in linking different type of data sets over the Internet, thus turning it into a widely distributed database. CAFSIAL is a research based application that focuses on presentation layer problem of LOD project. The problem statement has been given. Main contribution will be developing an algorithm for clustering of similar data and its presentation to user. Thus, chapter 1 gave an introduction about the problem and the research area.

# CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

Linked Data is a leading Semantic Web project aiming to open and interconnect different datasets over the Internet. Although the Internet connects large numbers of documents with each other, it does not provide this connection between datasets. This deficiency will overcome by using linked data techniques. Linked data uses current web architecture along with RDF and URI's. As described in [16] some of the properties of linked data are similar to classic web. These include:-

- Linked Data is generic and contains any type of data.
- Any one can publish data on linked data web.
- Data publishers can choose any type of vocabulary for data representation.
- Different entities are linked to each other by RDF Links, creating global data graph.
- It enables the discovery of new data sources.

From an application development point of view, Linked data has following characteristics:-

- Data is separated from its formatting and presentation.
- Data is self describing.
- Combination of HTTP and RDF simplifies data access as compared to web API's.
- Linked data is open. Applications are not meant to work on particular data sets, instead it can use RDF to search data from any of the data set.

## 2.1 Use of Linked Data in Web Applications: [17]

Although linked data is a new concept, still it is being quickly adopted for building web applications. Many web developers use different API's in their applications for enhancing them, but the scope of these API's is limited. Therefore, developers can take benefit of Linked Open Data as it is based on a Common Data Model. Use of linked data in some of the existing applications is given below.

### 2.1.1 FAVIKI [18]

FAVIKI is a social book marking tool. It allows creating Wikipedia concepts as tags. It also allows creating new tags and connecting them to common universal concepts present in the knowledge world. A user interesting in any of these tags can dereference the URI of the tag to obtain the information.

### 2.1.2 DBpedia – Mobile: [19]

DBpedia mobile is a client application meant for mobiles. It uses GPS signals from a mobile to get its current geographic position. After locating its position it renders its map indicating the nearby locations from the DBpedia datasets. Using this map user can further navigate into interlinked datasets and can obtain background knowledge about locations nearby. This is an interesting application of linked open data.

### 2.1.3 BBC Music Beta: [20]

It is a web application by BBC for music. It is built on Musicbrainz metadata and identifiers. Information like name of the artist is picked from Musicbrainz and information like introduction of the artist is extracted from Wikipedia.

From the three examples discussed above we can say that Linked Open Data is being used along with existing web standards.

## 2.2 Challenges in Linked Data:

Linked data has linked and published a large number of data sets and has formed a test bed for linked data technologies. However, in order to convert linked data into a single global database, following challenges need to be addressed.

### 2.2.1 User interfaces and interaction Paradigm:

Linked data extracts data from various data sources. However, presentation of extracted this semantic data to users is a major issue. Different applications are presenting this data in different ways. As there is no model or standard for user interaction with LOD applications and for its presentation to users.

### 2.2.2 Application Architectures:

There is no standard way of building a linked data application, therefore architecture of different applications differs from each other.

### 2.2.3 Schema Mapping and Data Fusion:

Linked data extracts data from various sources and presents it to user. However, data extracted from data sources is represented side by side and is not integrated. It requires mapping of terms from different vocabularies to target schema.

Data Fusion is a concept in which several things representing same real world object are interconnected. The major problem is the conflict in data. This means that a situation in which output for a same thing is different from different sources.

### 2.2.3 Link Maintenance:

In linked data, addition, edition and deletion of objects and sources are routine tasks. However, there is no standard way for pointing to newly added object links and for deletion of unused URI's. Hence research work is needed for link maintenance.

### 2.2.4 Licensing:

Licensing laws are not applicable to data. Therefore, licensing laws and such frame works should be made available and adopted in this field.

### 2.2.5 Trust, Quality and Relevance:

It must be ensured that the data provided to user by LOD applications is the same as needed by the user. It is correct and relevant as per user needs. This will ensure trust of the user on LOD applications.

### 2.2.6 Privacy:

As many distinct data sources are being interconnected for LOD, issues of privacy may arise. This issue needs both technical and legal efforts to be resolved.

## 2.3 Semantic Data Search Applications:

A number of semantic web applications have been introduced in recent past for handling semantic data search. As semantic web is in research phase, most of these applications are for research and analysis purpose. Here we discuss some of these applications.

### 2.3.1. Semantic Browsers

These applications have to be downloaded and then installed on a machine to use. In semantic web browsing context of data is extracted from the semantic data. Aim of semantic web browsing is to find the right web page i.e. the one containing the most related information needed to user. A number of semantic browsers have been presented in recent past.

“Magpie” [21] is a semantic browser which supports the interpretation of web pages. Magpie offers matching knowledge sources, which a user searches for finding the background knowledge relevant to a web resource. It contains a semantic layer which associates ontology to web resources. Users access these services using common web browsers. However, it does not make use of RDF, instead it uses RDF(S).

“PowerMagpie” [22] is a new version of original “Magpie”. The primary goal of PowerMagpie is that user has to make very little effort of semantic understanding of a web content. It automatically combines the similar information present in the text of the web page and discover the ontologies. In order to achieve this goal, PowerMagpie has to handle four major tasks.

a) Identifying relevant terms in the currently browse web page.

b) Selecting ontologies from internet.

c) Relating the text to semantic information

d) Navigating textual and semantic information together.

“Piggy Bank” [23] is another semantic browser. It makes use of semantic data within a web page during browsing. In absence of semantic data, it collects information from a web page using a screen scrapper and organizes it into semantic format.

“Haystack” [24] also belongs to the group of semantic browsers. Haystack has been built as an extensible platform that allows various kinds of functionality to be developed easily and independently. It makes use of RDF triples. It facilitates Semantic Web developers for building end user applications based on RDF.

#### 2.3.2. Semantic Data Browsers

These applications are some what similar to common web database applications. Semantic Data Browsers have a front end for users and a database at the backend containing data in RDF Triples.

“mSpace” [25] is a Sematic Data Browser. It is an experimental application. Its initially carried data related to music. Normally, people don’t have in depth information about music; they just search information by artist, album of instrumental tool. So we can say that people does not know underlying semantics of music. So when a user searches information about music we should not present its underlying semantic information and just present the information user needs.

By semantic web techniques, it is possible to make collections semantically process the information and provide the museum visitors with accurate data searchfacility. It presents information related to a range of museums on web [26]. This concept is illustrated in a case study: the prototype of Museum Finland, a semantic portal for Finnish museums to publish their collections on the Semantic Web.

### 2.3.3. Direct Manipulation:

Third type of applications deals with semantic data directly. “Tabulator” [27] is an example of this type. The Tabulator is an RDF browser. People can use it to search the information semantically. The data searched and accessed is in RDF form. RDF linked are searched and these links are used for data searching.

### 2.3.4 Visual Query Tools:

Visual Query Tools allow users to form their own queries using a visual interface. Users should have knowledge of RDF and SPARQL in order to use construct queries and use these tools. Few examples of these tools are discussed below. “Exploratar” [28] is a visual query tool. It allows user to construct queries visually. Exploratar uses an operational model. A visual interface allows user to give criteria for query and the under lying operational model implements it. It allows information searching, exploration and visualization facilities. User can extract information without having domain knowledge.

“NITELIGHT” [29] is also allows construction of visual queries. Users construct queries on a visual interface. A set of graphical notations that represent semantic query language constructs are available to help the user in query construction. It uses vSPARQL which is visual counterpart of SPARQL. It also provides a graphical editing environment along with combines ontology navigation capabilities.

Tools like Exploratar and NITELIGHT require SPARQL knowledge from the user for data searching and exploration.

### 2.3.5 Faceted Searching:

Faceted Searching is also known as Faceted Navigation or faceted browsing. Wikipedia defines faceted searching as “A method for accessing group of information presented in a faceted classification, allowing people to investigate by sorting out accessible information.

Faceted searching technique is also being used in semantic web applications. Below are some examples to it.

“/Facet” [30] is a semantic browser which addresses three problems a) users should be able to select and navigate through facets of resources of any type and to make selections based on properties of other semantically related types b) any new dataset can easily be added without disturbing the existing system and c) semantic data is not in a form for browsing, therefore a search mechanism should be provided. “/Facet” is allows semantic web developers an instant interface to their datasets. Automated facet searching further refines it to be used by the end users. It is a open source software.

“YARS2” [31] is also faceted semantic web tool. It uses a graphical data model for and provides interactive query mechanism to end users. It collects structured and interlinked data from various distributed resources on the Internet. It provides object’s description and returns answers instead of links.

### 2.3.6 Problems in Existing Systems:

After studying the above mentioned systems we can list some of the problems.

(a) In order to use most of the systems, user must possess knowledge about RDF, OWL and SPARQL.

(b) Most of the applications lack filtering mechanism.

(c) Making a visual SPARQL query requires background knowledge.

(d) A novice user cannot use these systems.

(f) Most of the systems are based on limited resource types.

(g) Presentation of data to user in textual form.

From these problems we can say that, there is no application which is using the semantic web techniques for information processing and user is using it like commonly used web applications.

## 2.4 CAFSIAL:

CAFSIAL [14] is a new addition to LOD applications. It stands for "Concept Aggregation Framework Structuring Informational Aspects from Linked Open Data". CAFSIAL contains data extracted from DBpedia which is considered central point in linked Open Data. DBpedia contains 23 different types of resources like (Person, Place and Organization etc). Initial experiment of CAFSIAL was carried out with resource type "Person".

CAFSIAL is based on a "Concept Aggregation Framework". According to this framework, relevant concepts of a resource can be aggregation from a knowledge base and the most related informational aspects can be organized [14]. CAFSIAL describes this concept in three layers; we discuss it briefly with the initial CAFSIAL framework based on resource type "Person" below:-

### 2.4.1 Aggregation Knowledge Bases Layer:

In this layer two knowledge bases were generated. First is the DBpedia Property Dump. In this dump each type of "Person" was queried using SNORQL Query Explorer. [31]. Then all the distinct property sets for “Persons” were aggregated. A query used for this purpose is as under:-

```
SELECT DISTINCT ?p WHERE {
?s ?p ?o .
 ?s rdf:type <http://DBpedia.org/ontology/Artist> .
}
```

The above SPARQL query is searching for all the distinct properties that are related to resource type Artists, present in DBpedia. Since in RDF a resource is defined in form of

Subject-Predicate-Object, the SPAQRL queries also require Subject-Predicate-Object in the query along with the criteria for search.

**2.4.2 Properties Aggregation Layer:**

In this layer, similar properties of resource are aggregated. These properties are then grouped together to form sub aspects of a resource. In this step system tries to find a related property from DBpedia dump. This information is in RDF form. In case no information is retrieved it tries to find related properties from Yago Classification dump. After retrieving the information, these properties are *manually* mapped to logical informational aspects discussed above. Multiple concepts can be related to a single aspect. Figure 2.1 illustrates it.

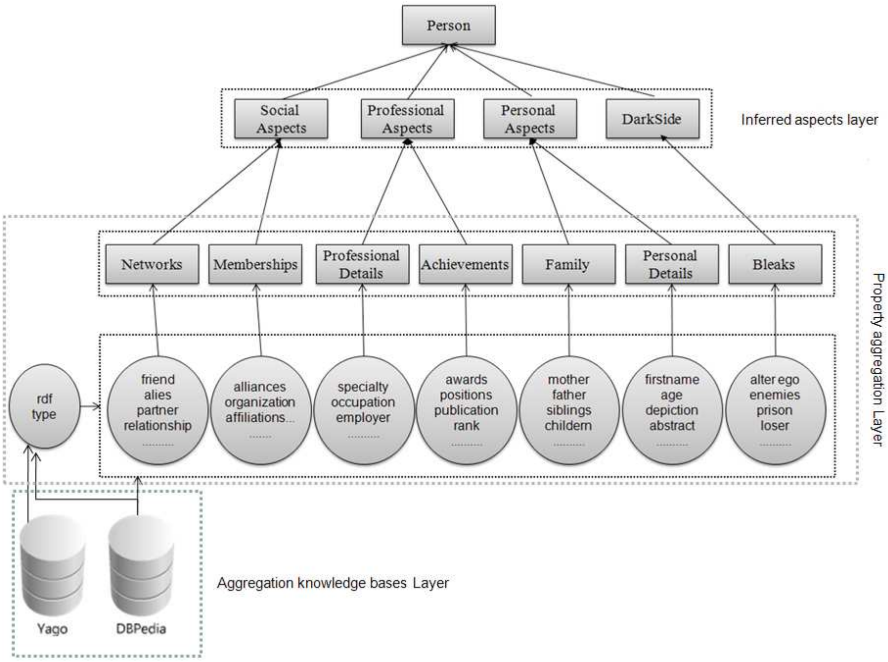



Figure 2.3: Concept Aggregation Framework (Resource Type Person) [14]

### 2.4.3 Inferred Aspects Layer:

In this layer, related sub aspects are grouped together to form aspects of a resource. In case of resource type "Person" the four informational aspects are "Personal", "Professional", "Social" and "Dark Side" as shown in the above figure.

## 2.5 Architecture of CAFSIAL:

The system comprises of four parts namely, Query Manager, Auto Suggestion Module, Information Retrieval Module and Search with Property Module. Figure 2.2.

### 2.5.1 Query Manager:

It is the main controlling part of application. It converts keyword typed by user into a SPARQL query.

### 2.5.2. Auto Suggestion Module:

After transforming keyword into SPARQL Query, Query Manager starts the Auto Suggestion Module. The local DBpedia triple store is searched for understandable concepts. If results are retrieved, they are presented to user for selection. There may be case when no results are returned. Concept selected by the user or the keyword (in case of no selection) is forwarded to the Information Retrieval Module.

#### 2.5.3 Information Retrieval Module:

This module locates URI's and extracts related information. First, it takes the searched term and passes it to URI locator Process. This process searches the term in local DBpedia Triple Store for locating a URI. If no URI is located, the searched term is converted into a query and the web service SINDICE is used to locate its URI. The located URI is dereferenced using LOD Retrieval process and its RDF description is taken from DBpedia Server.

This RDF description is parsed into RDF triples using a parser, which also stores them locally. In the end, important aspects of the resource are sorted out manually using Concept Aggregation Process and output is presented to user.

#### 2.5.4 Search with in Property Module:

User can search with in all the retrieved properties using this Module. The term searched by the user is queried to wordnet for acquiring its synset. This task is carried out by the synset extraction process. Query Manger retrieves related triples from the local triple store that are matched with synset. Property Locator retrieves matches the triples to the keyword and results are presented to user.

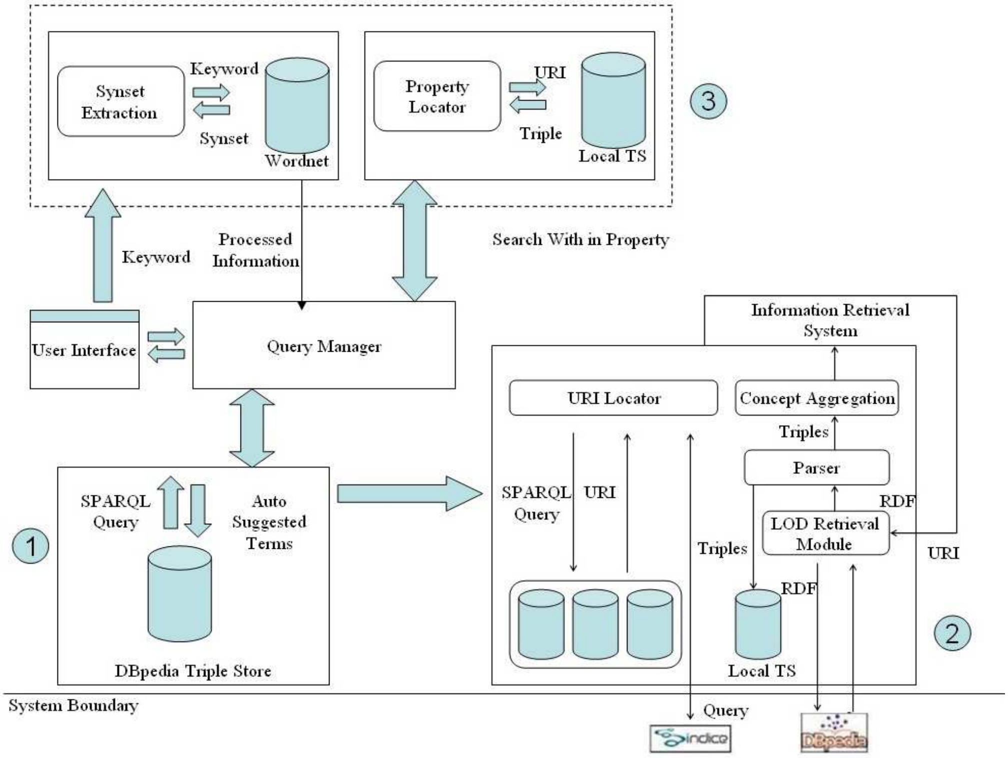



Figure 2.2: CAFSIAL's Architecture [14]

## 2.6 Conclusion

In this chapter we have briefly discussed use of LOD in semantic web applications and applications dealing with semantic data search. Problems in existing applications and their way of working have been discussed. We have also discussed problems of LOD that need further research work. CAFSIAL is basically a semantic web application that deals with searching and presentation of linked data. Architecture of CAFSIAL has also been discussed in this chapter.

# CHAPTER 3: METHODOLOGY

As discussed in Chapter 2, the initial version of CAFSIAL holds data about persons, and properties are manually mapped to aspects and sub aspects. This experiment was a proof of concept for Concept Aggregation Framework. In this chapter methodology of automating the process of property binding will be discussed in detail.

## 3.1 Selection of a Dataset (Resource Type):

To demonstrate the strength of proposed system, we need a comprehensive dataset. Further more, for studying the property binding mechanism we need a resource type with ample number of properties. DBpedia contains 17 different types of resources.. We have selected the resource type of "Organization" from LOD because of following three reasons:

(1) Broad Range of Organizations are available on DBpedia.

(2) Ample number of properties are available for each Organization.

(3) Concept of an "Organization" is easily to understand because we come across different type of organizations in our routine life like college, hospitals, companies and universities etc.

## 3.2 Organization's Aspects and Sub Aspects:

To structure the information for resource type: organization, we engineered/model an ontology which conceptualizes main aspects of organization such as its Personal aspects, Professional Aspects, Social Aspects and Dark sides etc. The main concepts have further been linked with sub-concepts. For example, the main concept Personal has been linked with its sub-concepts Important Persons, Geographical and web etc. This ontology was



1 http://mappings.dbpedia.org/server/ontology/classes

conceptualized by critical analysis of all properties made available by DBPEDIA[1]. This exercise enabled us in understanding the concept aggregation framework and relationship between properties and aspects.

### 3.2.1 Filling Organization's Ontology:

In order to fill the ontology we queried DBpedia for all the sub classes of Organization class. Each query returned the properties of the respective sub class of Organization. These properties have been saved automatically in a database.

In the example below, query retrieving all properties that a "Hospital" holds is shown.

```
SELECT DISTINCT ?p
WHERE {
?s ?p ?o .
?s rdf:type <http://dbpedia.org/ontology/Hospital> .
        }
```

### 3.2.2 Counting Number of Records:

There are large numbers of properties for each resource. All of these properties are not much of interest. During the study of properties, it has been found that number of records some of the properties for a particular resource is very less i.e. 1 or 2. After detailed analysis it was decided the properties having less then 10 records for a particular resource can be ignored. A script has been written in PHP which returned the number of records for each property against a particular resource by selecting one property at a time and querying for the number of records on DBpedia server. An example query used is shown in the example below.

```
SELECT COUNT(DISTINCT ?s) AS ?counts
WHERE {
?s property_name ?o .
```

*?s rdf:type <http://dbpedia.org/ontology/College> .*

}

Table 3.1 gives the details of properties of five resource types and the record count.

Table 3.1 Resource Types and Relevant Properties

| **Ser No** | **Resource Name** | **Total No of Properties** | **Properties with Record Count < 10** | **Properties with Record Count >= 10** |
|---|---|---|---|---|
| 1 | Airline | 254 | 179 | 75 |
| 2 | College | 133 | 82 | 51 |
| 3 | Hospital | 83 | 23 | 50 |
| 4 | Legislature | 101 | 35 | 66 |
| 5 | Library | 150 | 103 | 47 |

### 3.3 Automating Property Binding:

Binding properties of aspects is major goal of this work. Existing version of CAFSIAL binds the properties to aspects manually. Since there are large number of properties and each these properties belong to different type of Domains, hence, clustering of properties needs some detailed study for formulating some strategy to resolve the issue. Two different strategies has been adopted for this purpose:-

a. Using Semantic Relatedness

b. Using Domain and Range of Properties

### 3.3.1 Semantic Relatedness:

Semantic Relatedness is a way of measuring the relationship between different concepts or terms. We used this technique and measured the relationship between different resources and their properties and properties with the properties. Using the relatedness results we tried to group the properties. However the results were not satisfactory in terms of grouping the similar type of properties at this stage and needs further research work. Due to this reason the use of this technique is not implemented at this stage.

### 3.3.2 Using Domain and Range of the Properties:

Each DBpedia property has a Domain and Range. During the study we found that the Domain and Range of properties can be quite useful in clustering the properties. After a number of experiments it has been inferred that grouping the properties with respect to Domain can result into related sub-aspects, while grouping the data with respect to Range can group together the correlated properties. This strategy is fairly straight forward and is easy to implement. And since the results are also up to expectation, it has been decided that for this research activity we will use Domain and Range for property binding. Table below shows distinct Domains for Organizations.

Table 3.2 Distinct Domains For Organizations

| S# | Domain | S# | Domain |
|---|---|---|---|
| 1. | Airline | 2. | Airport |
| 3. | College | 4. | Hospital |
| 5. | Library | 6. | Country |

| | | | |
|---|---|---|---|
| 7. | School | 8. | Person |
| 9. | Geo Spatial Thing | 10. | Owl:Thing |
| 11. | Populated Place | 12. | Military Unit |
| 13. | Broad Caster | 14. | Radio Station |
| 15. | Legislature | 16. | Television Station |
| 17. | University | 18. | Company |

Similarly, Distinct Ranges for Organizations are given in the table below:-

Table 3.3 Distinct Ranges For Organizations

| S# | Range | S# | Range |
|---|---|---|---|
| 1. | Airline | 2. | Airport |
| 3. | Company | 4. | College |
| 5. | Currency | 6. | Country |
| 7. | Legislature | 8. | Educational Institute |
| 9. | Hospital | 10. | Library |
| 11. | Person | 12. | Place |
| 13. | Populated Place | 14. | owl:Thing |
| 15. | xsd:nonNegativeInteger | 16. | xsd:date |
| 17. | geo:SpatialThing | 18. | University |

### 3.3.3 Saving Domain & Range:

As discussed above property binding will be automated on the basis of Domain and Range. For this purpose we need Domain and Range of each property. It has been mentioned in first chapter that the ontology of any resource is defined in OWL for manipulating it on Web. In order to get the Domain of a property we need to query as shown in an example below.

*SELECT DISTINCT ?domain*
*WHERE {*
*property_name http://www.w3.org/2000/01/rdf-schema#domain ?domain*
*}*

Similarly the Range of a property can be obtained using the query

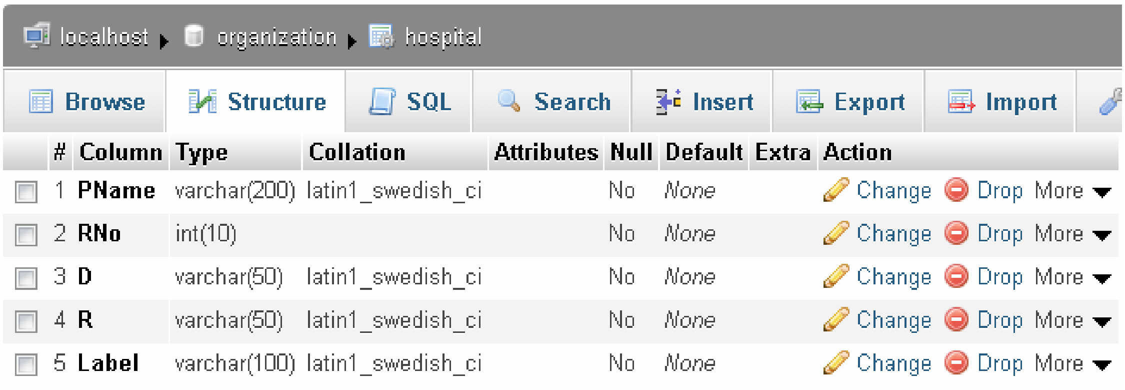
*SELECT DISTINCT ?range*
*WHERE {*
*property_name http://www.w3.org/2000/01/rdf-schema#range ?range*
*}*

However, most of the properties on DBpedia have still not been converted in OWL format, so getting the Domain and Range of all the properties programmatically is not possible.

To resolve this issue each of the property of a resource type has been queried. Result of the queries has been analyzed. The Domain and Range of the properties has been decided on the basis of the output. It is important to mention that DBpedia ontologies can also be studied on DBpedia website. This information has also been taken into account for deciding the Domain and the Range of each property.

## 3.4 Database Creation:

A database has been created in mySQL DBMS. This database has been used to fill the Organization's Ontology discussed above in this chapter. For each resource type a separate table has been created. Structure of table for each resource type is similar. Figure below illustrates the structure of “Hospital” table.




| # | Column | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|---|---|---|---|---|---|---|---|
| 1 | PName | varchar(200) | latin1_swedish_ci | | No | None | | Change Drop More |
| 2 | RNo | int(10) | | | No | None | | Change Drop More |
| 3 | D | varchar(50) | latin1_swedish_ci | | No | None | | Change Drop More |
| 4 | R | varchar(50) | latin1_swedish_ci | | No | None | | Change Drop More |
| 5 | Label | varchar(100) | latin1_swedish_ci | | No | None | | Change Drop More |

Figure 3.1: Structure of Table “Hospital”

### 3.4.1 Organization Dump:

Selecting the type of organization from its name is an important issue. For example if the user searches “Quaid-i-Azam” from organization categories, how will we decided that whether user wants “Quaid-i-Azam Airport” or “Quaid-Azam University”. To resolve this issue a table has been created in the database which holds the names of all the organizations present in DBpedia, their types.

## 3.5 CAFSIAL Architecture:

Architecture of existing CAFSIAL architecture has already been discussed in chapter two. Since, the current experiment has automated the process of property binding, the architecture of CAFSIAL has changed. Figure 3.1 shows architecture of new CAFSIAL application.

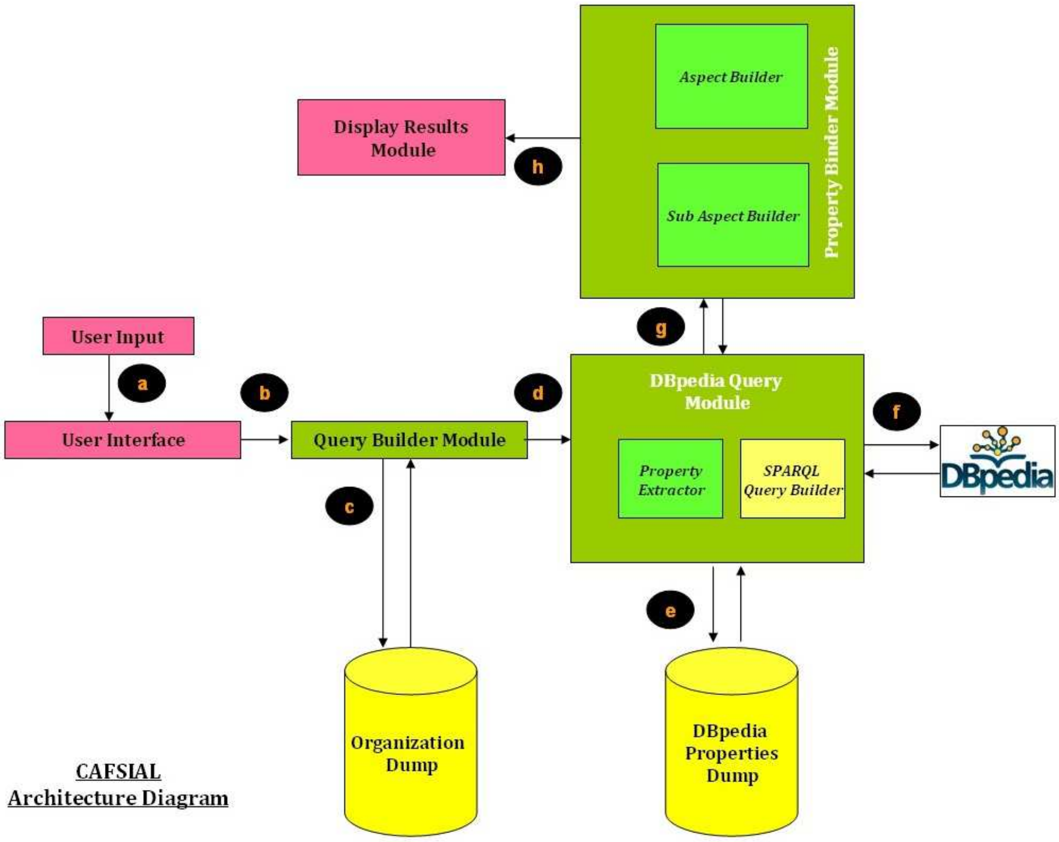



Figure 3.1: CAFSIAL Architecture Diagram

### 3.5.1 Query Builder Module:

User enters the keyword to search in text box provided on User Interface. User Interface passes the keyword to Query Builder Module. Query Builder Module works in two steps. In the first step it turns the key word into a mySQL query and retrieves the related concepts from Organization Dump. The results are presented to user for concept selection. User selects the concepts and the as a second step, this module passes the name and type of the concept to DBpedia Query Module.

### 3.5.2 SPARQL Query Module:

DBpedia query module comprises of two sub modules property extractor and SPARQL Query Builder.

***Property Extractor:*** Property Extractor is responsible for extracting the properties from the DBpedia properties dump. It extracts the properties that relate to the searched concept and all of these properties are passed on to the second Sub Module i.e. SPARQL Query Builder. It also extracts the Domain and Range of each property.

***SPARQL Query Builder:*** Properties extracted by the property extractor are passed on to this sub module. It picks up properties one by one and queries to DBpedia Server. Since there can be large number of properties which can consume time, it makes use of asynchronous queries for making the system efficient.

DBpedia Query Module passes the property names, Domain, Range and the results returned from DBpedia Server to the next module which is Property Binder Module.

### 3.5.3 Property Binding Module:

Property Binding Module has two sub modules Sub Aspects Builder and Aspects Builder.

***Sub Aspects Builder:*** This sub module binds the related properties into sub aspects. This binding is carried out on the bases of their Ranges. All the distinct ranges are noted. Then sorting of properties is carried out on the basis of the range of properties. In this way the properties are grouped together. These groups are then passed onto the Aspect Builder.

***Aspects Builder:*** Grouping together the similar sub aspect to form the aspects is responsibility of this sub module. Each group of properties formed by sub aspect builder are re-sorted on the basis of Domain of properties. This result in forming different aspects.

After forming the aspects and sub aspects the information is passed on to the results module.

### 3.5.4 Display Results Module:

This module is responsible for presentation of the results to the user. It organizes the results in a structure form and displays it to the user.

## 3.6 Summary:

In this chapter we discussed methodology for the implementation of CAFSIAL system. Architecture of the system, details of aspects, sub aspects and properties. We also discussed Domain and Range types and other issues related to the system. In addition we given the details of the database and the tables created in it.

# CHAPTER 4: IMPLEMENTATION AND RESULTS

In the previous chapter we discussed in detail the methodology adopted for automating the process of property binding for CAFSIAL. Following that methodology we have implemented new version of CAFSIAL. In this chapter we will discuss the implementation and the results achieved. We will also present the comparison between CAFSIAL and other state of the art system available.

## 4.1 System Implementation:

CAFSIAL is a Semantic Web application which makes use of Linked Open Data which is a famous project of Semantic Web. In order to implement the new version of CAFSIAL following languages and technologies have been used:-

### 4.1.1 PHP:

PHP is a famous language used to develop web based applications. It is platform independent and can be scripted in any editor like notepad. It can easily be used with HTML and other scripting languages. It is used for server side scripting.

### 4.1.2 Java script:

Java script is also a scripting language for web applications. It is used for client side scripting.

### 4.1.3 AJAX:

As discussed in previous chapter asynchronous queries are made on DBpedia server for improving the efficiency of the system. This has been achieved by the use of AJAX. AJAX stands for Asynchronous Java and XML. It provides facility of updating a portion of a web page with out posting back the whole page to the server.

### 4.1.4 SPARQL:

SPARQL stands for SPARQ Protocol and RDF Query language. It is a query language for RDF. It has been used for querying on the DBpedia server. Properties of a concept are queried and the output is displayed to the user.

### 4.1.5 RAP Library:

Since PHP by default does not has ability to support SPARQL queries. We have used RAP library for making use of SPARQL inside PHP scripts. It is written in PHP and can be configured very easily.

### 4.1.6 MySQL:

Database for CAFSIAL has been created in MySQL. MySQL is a famous DBMS. It is open source and can easily be incorporated with PHP to build web database applications.

## 4.2 Results:

CAFSIAL is a system for searching any concept semantically. Output of the system is information searched, which is presented to the user in a structured form. The basic unit of the information is properties. So the results of the new CAFSIAL system are first evaluated on the basis of properties.

### 4.2.1 Properties:

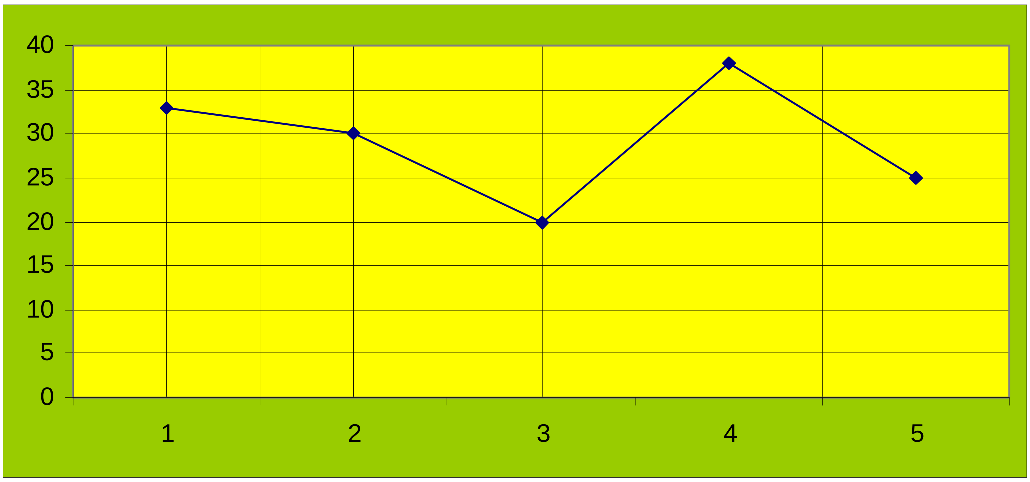
In order to evaluate CAFSIAL on the basis of properties of a resource we selected five different resource types: Airline, College, Hospital, Legislature and Library. CAFSIAL has been used to search different concepts. The results achieved are discussed below.

A total of 182 concepts has been searched. Average number of distinct properties searched is 121. The detail of the result is given in the table below.

Table 4.1: Results of Properties Searched by CAFSIAL

| Organization Type | No of concepts Searched | CAFSIAL | | | |
|---|---|---|---|---|---|
| | | Distinct Properties Searched | Min Searched | Max Searched | Avg No of Properties Searched |
| Airline | 39 | 33 | 2 | 21 | 15.33 |
| College | 27 | 30 | 18 | 26 | 21.67 |
| Hospital | 52 | 20 | 9 | 19 | 14.34 |
| Legislature | 40 | 38 | 32 | 20 | 23.47 |
| Library | 24 | 25 | 7 | 22 | 14.54 |
| **TOTAL** | **182** | **121** | | | **17.87** |

To explain the results two different graphs have been plotted below. Graph in Figure 4.1 (a) shows the variation between numbers of distinct properties searched per concept for a particular resource type; whereas graph in Figure 4.1 (b) shows average number of properties searched per concept.




-

Figure 4.1 (a)

The second graph shows the average number of properties searched per concept.

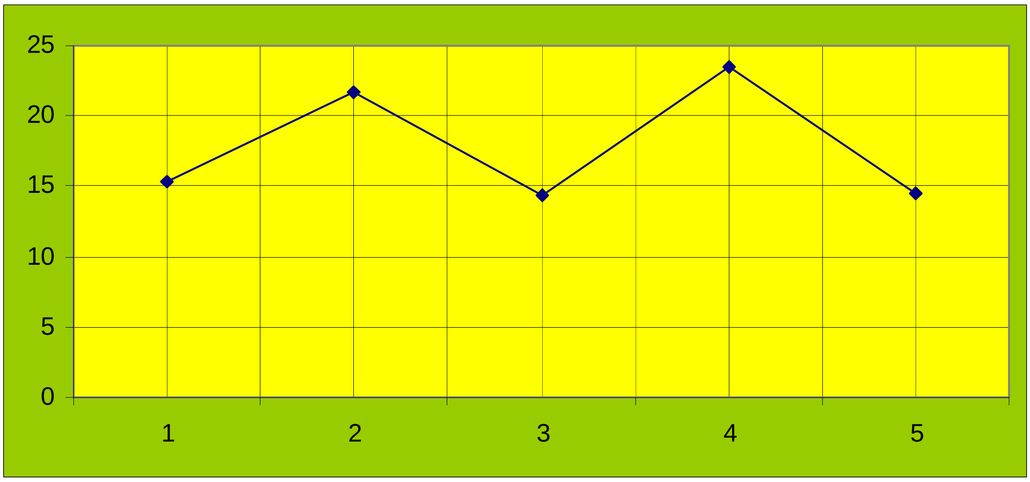



Figure 4.1 (b)

### 4.2.2 Time:

The efficiency of the system can be measured the by measuring the time consumed by the system in displaying the output. The time measured is as follows:-

Table 4.2 (a): Efficiency before using AJAX

| **S.No** | **Subject** | **Time / Value** |
|---|---|---|
| a. | Average time taken by a SPARQL query | 1.5 sec |
| b. | Average number of properties displayed | 20 |
| c. | Average time for display | 1.5 * 20 = 30 seconds |
| d. | Best case Time | 15 seconds |
| f. | Worst case Time | 165 seconds |

The efficiency of the system can be improved by decreasing the query processing time. Since the output is displayed to the user after all the queries are processed and results are returned by DBpedia server. Hence adopting a mechanism in which all queries are run in parallel and output is displayed as soon as query results are returned will be more beneficial. For this purpose AJAX has been used. This increases the efficiency of the system by reducing the display time as shown in the table below.

Table 4.2 (a): Efficiency after using AJAX

| **S.No** | **Subject** | **Time / Value** |
|---|---|---|
| a. | Average time taken by a SPARQL query | 1.5 sec |
| b. | Average number of properties displayed | 20 |

| | | |
|---|---|---|
| c. | Average time for display the whole page | 10 seconds |
| d. | Best case Time | 5 seconds |
| f. | Worst case Time | 24 seconds |

### 4.3 Comparison:

In order to evaluate the system's performance it has to be compared with some standard or with some state of the art system. In this section we discuss these comparisons.

#### 4.3.1 Comparison with Standard:

In order to compare the system with a standard we first need some standard. In order to find a standard we first tried to find some standard ontology for each resource type. But we did not find any standard ontology for most of the resource types.

So as a second option we used top three search engines google, yahoo and bing. We used the auto suggestion facility provided in these search engines and recorded properties searched by users against each resource type. Using these properties we have a standard set (dataset) of properties for each resource type. We can refer these properties as a standard and can compare the properties searched by our system with this dataset.

#### 4.3.2 Comparison with Freebase:

In the second comparison we have selected a state of the art system present on the web. This system is known as Freebase [33]. It is a semantic web search application which allows searching different concepts. It contains data about large number of concepts. Freebase has facility of manual editors, so the data present on freebase is well managed.

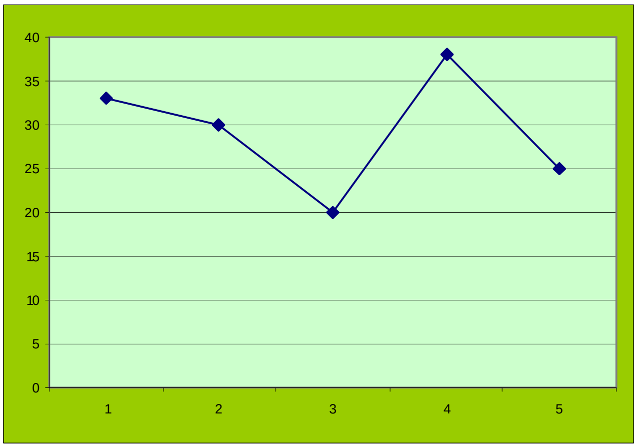
Table 4.3 gives the comparison between CAFSIAL and Freebase.

Table 4.3: Comparison between CAFSIAL and Freebase

| Organization Type | | Airline | College | Hospital | Legislature | Library | TOTAL |
|---|---|---|---|---|---|---|---|
| **No of concepts searched** | | 39 | 27 | 52 | 40 | 24 | **182** |
| **CAFSIAL** | **Distinct Properties Searched** | 33 | 30 | 20 | 38 | 25 | **121** |
| | **Min Searched** | 2 | 18 | 9 | 32 | 7 | **61** |
| | **Max Searched** | 21 | 26 | 19 | 20 | 22 | **86** |
| | **Avg Searched** | **15.33** | **21.67** | **14.34** | **23.47** | **14.54** | **17.87** |
| **Freebase** | **Distinct Properties Searched** | 45 | 43 | 25 | 10 | 27 | **150** |
| | **Min Searched** | 4 | 12 | 1 | 4 | 3 | **24** |
| | **Max Searched** | 25 | 26 | 17 | 10 | 14 | **92** |
| | **Avg Searched** | **15.33** | **20.12** | **6.56** | **5.96** | **6.28** | **10.85** |

It is quite clear from the above comparison that in resource type Airline and resource type College both the systems are almost at the same level with respect to number of properties searched. However, in the other three resources Hospital, Legislature and Library CAFSIAL is much better than Freebase.

This comparison has been illustrated in more detail in forms of Graphs in Figure 4.2. Figure 4.2 (a) and Figure 4.2 (b) give a comparison between number of distinct properties searched by CAFSIAL and number of distinct properties searched by Freebase. Figure 4.2 (c) and Figure 4.2 (d) give comparison between average number of properties searched by CAFSIAL and average number of properties searched by freebase system.



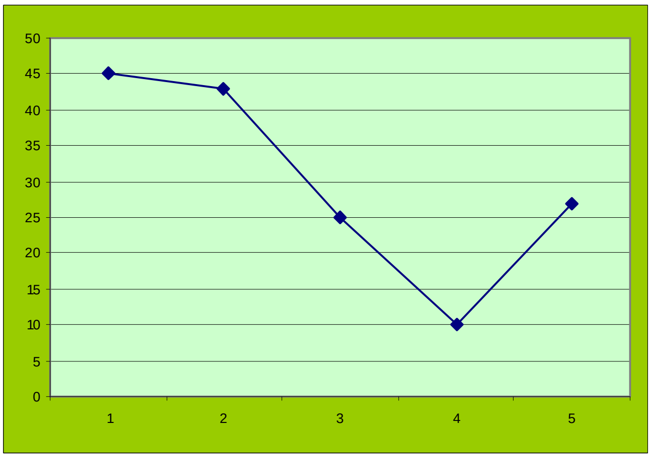
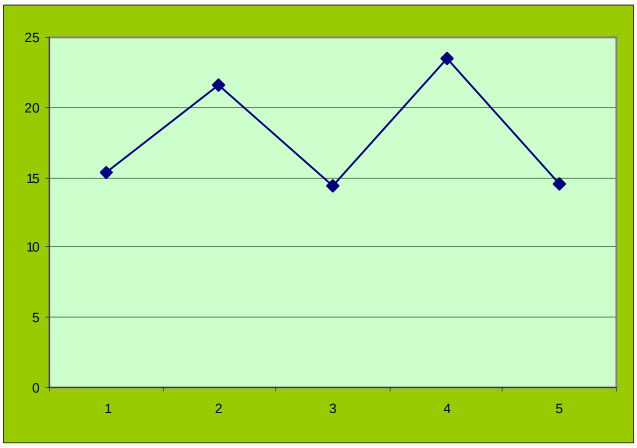




Figure 4.2 (a) Distinct Properties of Five Resource types Searched by CAFSIAL.

Figure 4.2 (b) Distinct Properties of Five Resource types Searched by Freebase.



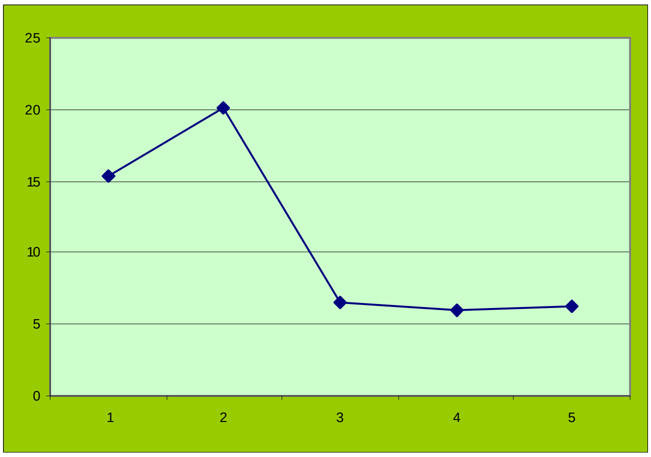




Figure 4.2 (c) Average number of Properties of Five Resource types Searched by CAFSIAL.

Figure 4.2 (d) Average number of Properties of Five Resource types Searched by Freebase.

## 4.4 Aspects and Related Properties:

Properties are grouped together to form aspects. Each aspect consists of some properties that are related to it. Maximum number of aspects displayed is 8. Name of the aspects and its details is given below.

**a. <u>Abstract:</u>** Abstract describes the searched term in descriptive form. It gives a brief introduction about the subject.

**b. <u>Organization Aspect:</u>** Properties whose Domain and Organization types are similar are bind aspect named similar to the organization type, for example, if the Domain type is "Airline" and the Organization is also "Airline" the name of aspect will be "Airline".

**c. <u>Related Persons:</u>** Every organization has some important persons related to it. Properties related to these persons are displayed in this aspect.

**d. <u>Financial Aspects:</u>** Financial aspect is an important aspect. It contains information related to financial issues of the Organizations.

**e. <u>Important Dates:</u>** This aspect contains different dates related to the organization. For example founding data, closed date etc.

**f. <u>Geographical Aspects:</u>** Information which describes geographical properties of an organization like its geographical coordinates and location etc are described in this aspect.

**g. <u>Important Values:</u>** The properties describing different figures or values of the organization are displayed in this aspect. For example if the organization is a college it contains number of students, graduates and undergraduates etc.

**h. <u>Web Aspect:</u>** Properties that are related to web are shown in this aspect, for example web address of the organization and its wiki page.

### 4.5 Conclusion:

In this chapter we have discussed the implementation and results of our new system. In addition to it we have compared the results of CAFSIAL with another state of the art system present on the web.

# CHAPTER 5: CASE STUDY

CAFSIAL is a system to search semantic data on web. Currently it can search only those concepts which are present on DBpedia. More over, resource type Organization has been selected for our current experiment. Although it is just one resource type but it has a number of sub types and contains large number of different concepts.

## 5.1 Organization's Concepts held in DBpedia:

Table below shows the number of distinct concepts of different resource types present in DBpedia.

| Ser No | Resource Type | Number of Concepts Held in DBpedia |
|---|---|---|
| 1 | Airline | 2805 |
| 2 | College | 78 |
| 3 | Company | 10000 |
| 4 | Government Agency | 2155 |
| 5 | Hospital | 1800 |
| 6 | Legislature | 105 |
| 7 | Library | 480 |
| 8 | Military Units | 10000 |
| 9 | Political Parties | 2756 |
| 10 | Radio Station | 10000 |

| | | |
|---|---|---|
| 11 | Schools | 10000 |
| 12 | Sports League | 1269 |
| 13 | Television Station | 6576 |
| 14 | University | 10000 |

### 5.2 Case Study:

It is very clear from the above table that a large number of Organizations can be searched from CAFSIAL.

In this case study we will search a hospital named “Castle Peak”. It is a famous hospital of Hong Kong. The Figure 5.1 shows CAFSIAL interface.


Figure 5.1: CAFSIAL Interface

In the second screen shot (Figure 5.2) user enters the key word Castle in the text box and submits it.

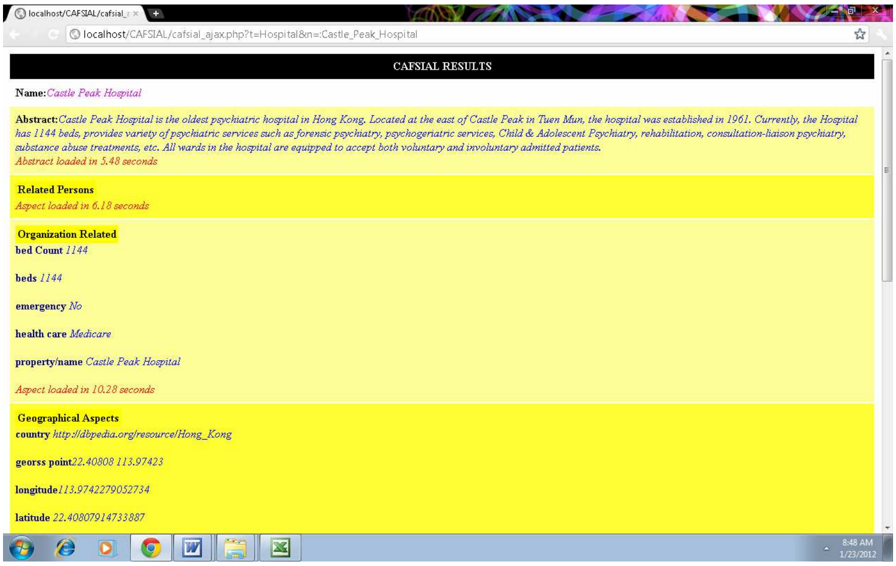

Figure 5.2: Key word entered by User

The CAFSIAL system retrives all the concepts which match with the work "Castle" along with their types Figure 5.3. User selects Castle Peak Hospital and system starts semantic search.

In Figure 5.4 system has started to display different aspects and properties related to Castle Peak Hospital.


Figure 5.4: Loading Aspects

In the Figure 5.5 system has displayed some aspects and properties related to it while some of the aspects are still being loaded.



In the Figure 5.6 system has displayed the remaining aspects of the concept along with the properties.

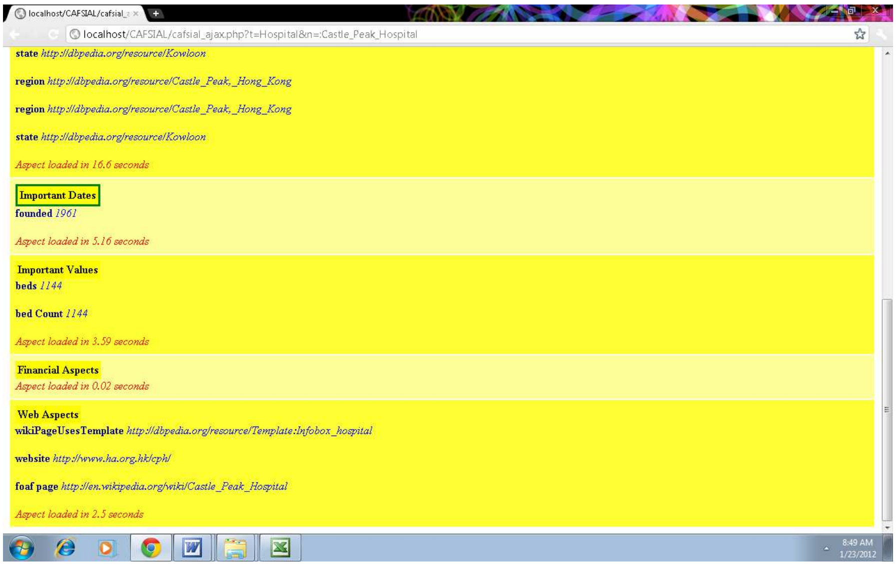



Figure 5.6: Remaining Aspects Loaded

## 5.3 Summary:

A case study has been presented in this chapter for explaining the working mechanism of CAFSIAL. Screen shots of the system has also been provided for understanding the working of CAFSIAL.

# CHAPTER 6: CONCULSION AND FUTURE WORK

A number of applications making use of Semantic Web Paradigms and Linked Open Data are available. Most of these applications focus on semantic data search and retrieval. A brief discussion about the existing semantic web applications was done in the literature review. Some of the problems related to semantic web applications discussed there are:-

- Users are required to possess technical knowledge related to semantic web constructs like RDF etc.
- Most of the applications are restricted to only a single domain.
- There are no standard rules for semantic web application development.
- There are problems in presentation of the output to the users in an intuitive and organized manner.

In order to address these issues the idea of automated CAFSIAL application has been presented in this work. Working mechanism of automated CAFSIAL application and the underlying details of the system have been discussed in detailed in previous chapters.

## 6.1 Automated CAFSIAL:

The main goals that have been addressed during current research work are:-

a. Populate CAFSIAL with a resource type; “Organizations” was selected for this purpose.

b. Develop an approach for binding properties to aspects.

c. Presentation of data to users in an appealing and easily understandable way.

### 6.1.1 Populating Organizations:

As discussed in chapter 3, we populated the properties of Organizations in a local data base, recorded the number of records against each property with respect to each sub class of the organization. And at last, we stored the Domain and Range of properties in the local database. It is important to mention here that only the names of the properties have been saved and not the entire data of "Organizations". The semantic data of the Organizations is present on DBpedia server.

### 6.1.2 Property Binding:

As binding of properties to aspects and sub-aspects automatically is main task. In the new strategy developed for CAFSIAL we have achieved this goal by using the Domain and Range of the properties. Grouping data with respect to Domain and Range separates the information at different levels, i.e. aspects and sub-aspects. The domain and range of most of the properties have been taken from ontology of the resource, while some of the properties whose domain and range was not found from ontology were queried, and on their Domain / Range was decided on the basis of output of the query.

### 6.1.3 Results Presentation:

Results are displayed to user in an appealing way. Data in DBpedia is in RDF form, but in CAFSIAL it is shown in simple text format so that users can easily conceive it. This is done by de-referencing the URI's and displaying property labels instead of property names in the output. Thus the complex semantic details of the information are hided from the user.

Common aspects are displayed first; the user is allowed to browse all the aspects and properties if wanted. Each aspect is shown separately, presenting the information in an appealing structured way.

### 6.1.4 No Local Data Storage:

In the initial version of CAFSIAL, there is a local database which holds the data related to "Persons". This data was taken from DBpedia. In the new version of CAFSIAL, local database is just holding the property names and not the complete data related to "Organizations". The data is fetched at runtime from DBpedia Server. Thus in future linking of other data sources and addition of new resource types will easily be carried out.

## 6.2 Conclusion:

In this experiment we have tried to solve the problem of presentation of data extracted from LOD sources. Since the data extracted from LOD sources is in semantic form and a common user cannot understand it, there should be some mechanism for presenting the searched data in a structured and easy to understand way. The mechanism adopted is to property bind the properties of a resource into different aspects and sub aspects and hide the underlying complex logic before displaying it to users. The proposed system has been compared with an available state of the art system and results have been discussed.

## 6.3 Future Work:

Research is a continuous process. There are many key issues in CAFSIAL which are potential research problems and are to be addressed in future. This section briefly describes these issues.

### 6.3.1 Adding other resource types:

There are 17 different types of resources available on DBpedia. In order to add all of them in CAFSIAL proper study of their classes, sub classes and properties is needed.

### 6.3.2 Adding data from other LOD sources:

The resource type Organization used for current experiment has been taken from DBpedia. There are many different LOD sources with data of different resources available with them. Research work is needed to connect these sources with DBpedia and extraction of data from these new sources.

### 6.3.3 Editing of Data:

Most of the semantic data search applications have manual editors which modify the data according to the users needs. CAFSIAL also holds large amount of data. A complete exercise is needed to edit it for increasing the readability of users.

### 6.3.4 Adding audio Feature:

Some of the properties like anthem, title song, theme song etc have audio files in the background. Work is needed to add audio features in the application.

### 6.3.5 Adding video Feature:

Similar to audio, video feature can also be added for playing the external video links retrieved from DBpedia against any concept.

### 6.3.6 Use google Maps:

Since DBpedia also returns latitude and longitude of an organization, google map API' can be used to display the location of the organization on map.

## 6.4 Summary:

In this chapter the main research problems addressed in the current project have been discussed. In addition to it, features which need research work and can be added in future have also been discussed.