A
Dissertation
On

METHODOLOGY: KNOWLEDGE REPRESENTATION IN SEMANTIC WEB

Submitted in Partial fulfillment of the requirement
For the award of the degree of

MASTER OF ENGINEERING
(Computer Technology and Application)

Submitted by
Kartar Jat
(University Roll No. 10189)

Under the Guidance of
Dr. Daya Gupta
(Head of Department)
Department of Computer Engineering

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
BAWANA ROAD, DELHI -110042
(UNIVERSITY OF DELHI)

2006-2008

CERTIFICATE

DELHI COLLEGE OF ENGINEERING
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI - 110042

Date: ………………

This is certified that the major project report entitled "**Methodology: Knowledge Representation in Semantic Web**" is the work of **Kartar Jat** (University Roll No. 10189) a student of Delhi College of Engineering. This work was completed under my direct supervision and guidance and forms a part of the Master of Engineering (Computer Technology and Application) course and curriculum. He has completed his work with utmost sincerity and diligence.

The work embodied in this major project has not been submitted for the award of any other degree to the best of my knowledge

Dr. DAYA GUPTA
HOD & Project Guide

Department of Computer Engineering
Delhi College of Engineering
University of Delhi, India

ABSTRACT

The enormous increase of data has made it difficult to find, access, present and maintain the information required by a wide variety of users on web. It is facing new problems such as information overload, inefficient keyword searching and heterogeneous information integration. These problems will be tackled by the modern technology known as Semantic Web Technology. The efficient management and presentation of data is called knowledge representation, which will work as the building block for the semantic web and semantic web applications.

Knowledge representation is the main issue with current web which can be resolved by the concept of ontology in semantic web. Ontology is a common set of terms that are used to describe and represent knowledge in an organized way, which is now days Ad-hoc in nature. The second issue for knowledge representation is to provide interoperability between current web and semantic web while switching from current web to semantic web. Most of data over current web are present in Relational database so there is need to convert it into RDF (Resource Description Framework) form, recommended by W3C for the semantic web development and knowledge representation.

The main objective of this thesis is to propose a methodology for the knowledge representation in the semantic web that resolve the problem of knowledge representation in semantic web. Author present a well articulated generic ontology design process for representing knowledge with Ontology that will overcome the previous ontology design approach where ontology design is Ad-hoc in nature. RDB2RDF Converter will convert relational database into RDF form which will be used as a base for any semantic web application.

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extent my heartfelt gratitude to everybody who helped me throughout the course of this project.

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor Dr. Daya Gupta, HOD, Department of Computer Engineering, Delhi College of Engineering, for her invaluable guidance, encouragement and patient review. Their continuous inspiration only has made me complete this dissertation. Without her help and guidance, this dissertation would have been impossible. She provided the conceptions and theoretical background for this study as well as suggested us the rational approach. She remained a pillar of help throughout the project.

I would also like to take this opportunity to present my sincere regards to my teachers Prof. D.Roy Choudhary, Mrs. Rajni Jindal, Dr. S.K. Saxena, Mr. Manuj Sethi and other staff members of Computer Engineering Department for providing me unconditional and anytime access to the resources and guidance.

Finally we are also thankful to my classmates for their unconditional support and motivation during this work. Last but not least, I special thanks to the crowd who are active in the field of Semantic Web and knowledge representation.

Kartar Jat
College Roll no: 10/CTA/06
University Roll no: 10189
M.E. (Computer Technology and Application)
Department of Computer Engineering
Delhi College of Engineering, Delhi-110042.

TABLE OF CONTENTS

	Certificate	ii
	Abstract	iii
	Acknowledgement	iv
	Table of Content	v
	List of figures	vii
1	**Introduction**	**1**
	1.1 Introduction	1
	1.2 Motivation	3
	1.3 Related work	4
	1.4 Proposed work	5
	1.5 Proposed System	5
	1.6 Organization of Work.	6
2	**Knowledge Representation using Ontology**	**8**
	2.1 Knowledge representation.	8
	2.2 Ontology and knowledge representation	9
	2.3 Ontology in semantic web.	11
	2.4 Semantic web and Current web	12
	2.5 Semantic Web Building Blocks	15
	2.6 Knowledge representation techniques	18
3.	**Methodology: Ontology Design**	**19**
	3.1. Ontology design Approaches	19
	3.2. Methodology: Ontology design Process	20
	3.3. ontology design Process	22
	3.4. Uses of Ontology	28
4.	**Tourism Ontology**	**31**
	4.1. Introduction	31
	4.2. Ontology design for tourism domain	33

4.2.1. Tourism domain 33
4.2.2. Scope of Ontology 33
4.2.3. Ontology design process 34
5. Relational to RDF Converter 39
5.1. RDF 39
5.1.1. RDF in Semantic Web 40
5.1.2. RDF Model 40
5.2. RDB2RDF Converter. 42
5.2.1. System architecture 43
5.2.2. Data Flow Diagram 44
5.2.3. Implementation with Java 45
5.3. Application programming interface (API) 50
5.4. Uses of RDB2RDF Converter. 53
6. Implementation Detail 54
6.1. Ontology design with protégé 54
6.2. RDF2RDF Converter 55
6.3. Snapshots 57
6.3.1. Ontology 57
6.3.2. RDB2RDF Converter 59
6.4. Result and Discussion 60
7. Publication from Thesis 64
7.1. Conference paper. 64
7.2. Journal paper 65
8. Conclusion 66
8.1. Conclusion 66
8.2. Future Work 67
9. References 68
Appendix A Protégé 70
Appendix B Jena API 74

LIST OF FIGURES

Figure 1.1: Knowledge and data ………………………………………………………………...….. 2

Figure 2.1: Ontology Structure…………………………………………………………….……..... 10

Figure 2.2 Semantic web architecture ……………………………………………………...….… 14

Figure 2.3 OWL Complexity ……………………...……………………………………….…..…… 15

Figure 3.1 Ontology design process………………………………………………………..…..…… 23

Figure 3.2 Semantic web model……………………………………………………….……... 29

Figure 4.1 Ontology design framework ………………………………………………..……..… 34

Figure 4.2 Defining class using protégé …..…………………………………………..….. 35

Figure 4.3 Instance in ontology …………………………………………………..……... 37

Figure 4.4 Query in ontology ………………………………………………..……..…38

Figure 5.1 RDF Triple …………………………………………………………..…..…….. 41

Figure 5.2 RDB2RDF architecture…………………………………………..………...…44

Figure 5.3 Functional Diagram of RDB2RDF Converter ……………………..……………45

Figure 5.4 Relational database table ……………………………………..…..…………….. 52

Figure 5.5 RDB2RDF Converter output …………………………………..……………..52

Figure 6.1 Class- Subclass hierarchies ……………………………………......…………57

Figure 6.2 Form design in Ontology …………………………………………...……..57

Figure 6.3 Defining constraints ……………………………………………………….58

Figure 6.4 Instances in Ontology ……………………………………………………..58

Figure 6.5 Searching in Tourism Ontology ……………………………..………..……59

Figure 6.6 Input for RDF Converter …………………………………………….…….59

Figure 6.7 Output for RDF Converter ……………………………………….…..60

Figure 6.8 Contact table ……………………………………………………………....60

Figure 6.9 Search result in ontology …………………………………………………...63

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Knowledge representation is the key element for the success of semantic web. In the current web the enormous increase of data has made it difficult to find, access, present and maintain the information required by a wide variety of users. It is facing new problems such as information overload, inefficient keyword searching and heterogeneous information integration. These problems will be tackled by the modern technology known as Semantic Web Technology [1]. The basic building technologies for knowledge representation are RDF, language recommended by W3C[2] for semantic web and ontology design for particular domain. Tim Berners-Lee, Director of the World Wide Web Consortium, referred to the future of the current WWW as the "*semantic web*" - an extended web of machine-readable information and automated services that extend far beyond current capabilities [2]. To represent knowledge in semantic web an appropriate knowledge representation scheme for any domain is important to represent data in effectively way.

Current web are build over data while semantic web is build over knowledge. There is considerable difference between knowledge and data. There are many theories are given to differentiate it. According to Russell Ackoff, a systems theorist and professor of organizational change, data and knowledge can be defined as:

1. **Data:** data is raw. It simply exists and has no significance beyond its existence. It can exist in any form, usable or not. It does not have meaning of itself. In computer parlance, a spreadsheet generally starts out by holding data.

2. **Knowledge:** knowledge is the appropriate collection of information, such that its intent is to be useful. Knowledge is a deterministic process. To build knowledge requires a true cognitive and analytical ability which can be used for understanding. In computer

parlance, most of the applications we use (modeling, simulation, etc.) exercise some type of stored knowledge

Knowledge and data exist at different level of correctness and understanding level. The data are row in nature having null value of correctness and understanding, while knowledge can be presented by different value of correctness and understanding as presented in figure 1.1.

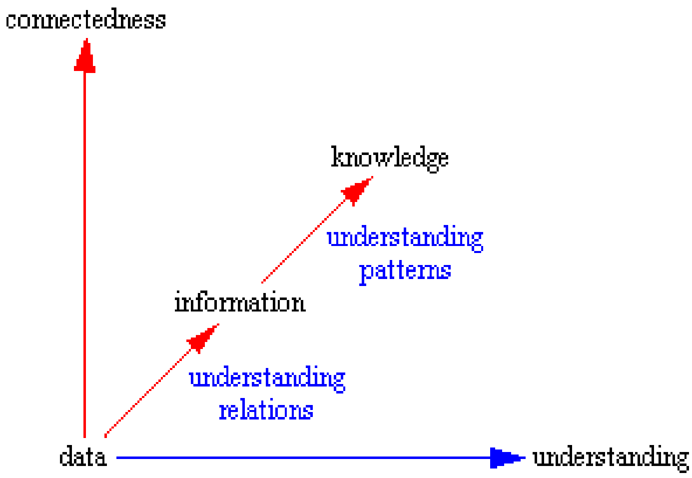


Figure 1.1 Knowledge and data

How to represent knowledge is the main issue with current web which can be resolved by the concept of ontology and machine understandable language such as RDF. Knowledge representation is most fundamentally, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting or it is a medium of human expression, i.e., a language in which we say things about the world. How to represent knowledge is the main question among the researcher. There is not any well defined mechanism for the knowledge representation, people used to do it in ad-hoc manner.

The knowledge representation play vital role in semantic web it is basically used for the representation of knowledge in machine understandable format which can be used for the semantic web applications such as semantic web portal design, semantic search engine, e-commerce and representation of data in Mobile devices in efficient way.

Knowledge representation is the building block for the Artificial Applications based on the machine learning and semantic web. To represent knowledge in semantic web an appropriate knowledge representation scheme for any domain is important to represent data in effectively way. How to represent knowledge is the main issue with current web which can be resolved by the concept of ontology and RDF in semantic web. This knowledge will be used for different semantic web applications because of better representation, organization and extraction of knowledge from different domain.

1.2 MOTIVATION

Knowledge representation and Semantic web are new concept to web; there is not any well defined methodology available for the knowledge representation [3]. Current web is facing many problems such as information overload, knowledge representation, population, inefficient searching and organization while maintaining and representing data over web.

In the current web there are many domains where data are present over different domain creating redundancy still unable to provide the desired result due to lack of an efficient knowledge representation technique. Tourism is one of such domain where data are spread over hundreds of domain but unable to provide the exact result to tourist regarding any domain [4].

To explore these problems of current web we come across solution to present Methodology for the efficient knowledge representation in semantic web. In this thesis we are going to present methodology for knowledge representation with ontology and RDF (Resource Description Framework) with well articulated steps for the ontology design [5] and a conceptual tool for the Relational database to RDF conversion of data for knowledge representation in semantic web.

The main issues for the semantic web are:

- A well defined Ontology design approach.
- Semantic representation of relational data, which is main data source in current web.

Hence we took these challenges and presenting a well articulated ontology design process for ontology design for any domain and also present a tool that will convert the relational data present over different data source into RDF, a language recommended by W3C(World Wide Web consortium) for the semantic web development.

1.3 RELATED WORK

A lot of work is being done in the field of ontology, RDF, semantic web and knowledge representation. A number of approaches are being presented; we come across some of such papers. One of such ontology design approach is presented by Xu Binfeng [6], for medical domain by construction of ontology for medical domain which is basically ad-hoc in nature and cover only tourism domain ontology designs approach.

Earlier approaches for ontology design and semantic web are Ad-hoc in nature and letter some ontology design methods are proposed. To meet the challenge of current web many semantic web design methods are proposed which is based on the Ontology for a particular domain [7].

Recent approaches are based on methodologies for building Ontologies and semantic web. Mariano Fernandez Lopez, Asuncion Gomez-Perez, and Juan Pazos Sierra [8] propose one of the ontology creation methodologies which are used for the ontology creation which is more diverse towards conceptualization. However, acquiring knowledge and review for the consistency is not mentioned which play a great role for efficient ontology design. This Methontology framework includes the identification of the ontology development process, a life cycle based on evolving prototypes and the methodology itself, which specifies the steps for performing each activity, the techniques used, the products to be output, and how the Ontologies are to be evaluated. The main steps for the ontology design are knowledge acquisition, evaluation, documentation, configuration management and integration.

Very recently approach for the ontology design is proposed by U.jan, magda, pavel, tereza which is iterative approach for ontology design [9]. This approach is basically based on the definition of the purpose, class and review of the design at each phase.

RDF play the role of knowledge representation same as HTML in current web. It is new language to web technology to present knowledge in semantic web. There is not too much work done in this field. How to represent and role of RDF is presented by Wajee Teswanich and Suphamit Chittayasothorn [10]. They present the technique of Transformation from RDF Documents to Relational Databases.

1.4 PROPOSED WORK

This research work was basically done to help the semantic web and semantic web applications such as semantic web portals, search engine and Mobile applications for the better representation and management of data. The current websites presents over the internet regarding Tourism did not full fill the need of each user in efficient way. So to present data in better way we have designed ontology for the tourism domain for the Rajasthan, one of the states of India famous for its various attractions to tourists. We have designed a generic approach for the ontology design process which can be used for any domain. Another aspect for the semantic web is data presentation in RDF format to present data in machine understandable format so we present the RDB2RDF converter that will present data in RDF form and facilitate the researchers for the better knowledge representation. To present all of this knowledge representation technique in better way we choose following topic of research for thesis work.

"Methodology: Knowledge representation in semantic web"

1.5 PROPOSED SYSTEM

To present an efficient knowledge management system for semantic web as discussed in section 1.2, we present the Methodology for the knowledge representation for semantic web. To present a methodology for the knowledge representation in semantic web we cover the two main techniques for the knowledge representation:

- Ontology for specific domain
- Data representation in RDF Format

To resolve these two challenges for the semantic web we present an ontology design process which is generic and applicable to all domains. The RDB2RDF converter tool has been designed to present data in RDF Format which is machine understandable for the development of semantic web applications.

1.6 ORGANIZATION OF WORK

The remainder of thesis is organized as follows.

In **Section 2** we need to have an overview of the knowledge, knowledge representation and its role in semantic web and Artificial Applications. Then we give the basic idea about the current web and semantic web and its building blocks such as Knowledge, ontology and OWL and its role in the knowledge.

Section 3 deals with ontology and ontology design approaches. It presents the generic ontology design process for any domain and presents the various steps of ontology design for semantic web.

Section 4 covers the tourism domain on web and presents ontology design for the tourism domain for Rajasthan state and also presents the scope and uses of designed ontology for the tourism domain.

Section 4 covers the basic idea about RDF and its role in semantic web. It also covers the problems with the knowledge representation in semantic web and provide a solution in the form of RDB2RDF (Relational Database to RDF) converter and its various aspects for the knowledge representation.

Section 5 covers the Implementation detail of Ontology design with protégé and RDB2RDF Converter showing the block diagram and different classes which we have implemented with JAVA and JENA API.

Section 6 covers the implementation detail for the ontology and RDB2RDF Converter. It also presents snapshots of implementation and finally discusses the result of work.

Section 7 describes the papers which have been accepted in international Conference on Enterprise Information Systems and Web Technologies (EISWT-08), USA. It also gives details of Paper communicated to International Journal.

Section 8 covers the conclusion and future work related to the original work done by us.

Section 9 finally culminate thesis showing different references including research papers, web sites and books that we had gone through during my project.

Appendix A Covers the introduction to protégé, Tool used for ontology design.

Appendix B covers the Introduction to Jena API, used for RDB2RDF Converter.

CHAPTER II

KNOWLEDGE REPRESENTATION WITH ONTOLOGY

2.1 KNOWLEDGE REPRESENTATION

Knowledge representation term used for how to organize, manipulate and reuse the data in efficient and meaningful way for different applications. The semantic web is an extension of current web where the huge information is given a well-defined meaning and structure using ontology and knowledge representation system. The Knowledge representation [11] is the key element for the development of the semantic web. W3C define a language RDF (Resource Description Framework) for the semantic web development and knowledge representation in machine understandable format to assist semantic web applications. In semantic web the domain specific knowledge is represented in organized way using ontology.

The proper knowledge representation is one of the main goals of the semantic web which can be achieved by ontology for the various domains. Ontology is a common set of terms that are used to describe and represent knowledge in an organized way. Ontology can be designed by various steps such as expert analysis, tool and language, ontology design, validation and maintenance and expert review for further enhancements. In the forgoing domains, the design of ontology has been done is an Ad-hoc way, There are many problems during the ontology design process between concepts and becomes real for particular domain after resolving many bottlenecks for ontology design. There is no well defined methodology and tool support for ontology design process involving construction and maintenance. Usually domain expert are consulted to explain the meaning of domain specific concepts and then previous experience by designers are used for creation of such ontology. But for efficient ontology design a methodology described well articulated steps is essential for transforming current web data to semantic web. There is need to build a knowledge base using ontology. Ontology is a branch of philosophy concerned with the study of what exists. In recent years, semantic web has enrich itself with the help of ontology design in different domains like medial, tourism etc. there are many knowledge representation system was developed using ontology for semantic web development. Protégé_3.3.1 of Stanford is a tool that is used for construction and maintenance for designing ontology for different domains.

The second issue for switching from current web to semantic web is to provide techniques for interoperability between current web and semantic web. While most of data over current web are present in Relational database so there is need to convert it into RDF form, recommended by W3C for the semantic web development and knowledge representation. Relational database to RDF (RDB2RDF) converter will act as a bridge between current web and semantic web knowledge representation. Author presents the methodology for ontology design and techniques for converting for Relational database to RDF conversion. A prototyping tool has been designed and applied to tourism department. This RDB2RDF Converter provide coupling between the current web and semantic web data representation.

These are the two knowledge representation technique which will be used for the representation of knowledge in semantic web and various semantic web applications. Knowledge representation is the meaningful presentation of knowledge after some modification in raw data so that it will be machine understandable for various artificial intelligence applications.

2.2 ONTOLOGY AND KNOWLEDGE REPRESENTATION

In semantic web the domain specific knowledge is represented in organized way using ontology and this representation is called knowledge representation in semantic web.

Ontology Concept:

Ontology plays the central role in knowledge management. Developing ontology is to defining a set of data and their structure at the same time covering the all aspect for a particular domain. Problem-solving methods, domain-independent applications, and software agents use Ontologies as knowledge bases to build the intelligence system for the design and development of a system built from Ontologies as data.

Ontology Structure

In the design of ontology the Basic elements are class, relationship, constraints, slots, forms and instances which will define ontology for a domain. Ontology is defined as one independent, collective representation of all standardized concepts for vocabulary not for simple words, but we are referring to collection of vocabulary, which have relationships and meanings. Ontology expressions are based not only on the logical relations between term definitions and other meanings but also on the bottom-out structure where the interpretation starts from primitive terms. The basic ontology structure for knowledge representation [9] is shown below with class or concept, relationship, constraints, forms and instances.

The building blocks of ontology design include:

1. **Class or concepts** for a domain such as location, city, travel, destination etc.
2. **Relationships** are the properties between two classes such as 'isa'.
3. **Forms** are framework that is used to set the layout for the instances in ontology.
4. **Constraints** are conditions that must be satisfied during the design.
5. **Instance** is values for particular categories in ontology.

The general ontology structure is as shown in figure 2.1

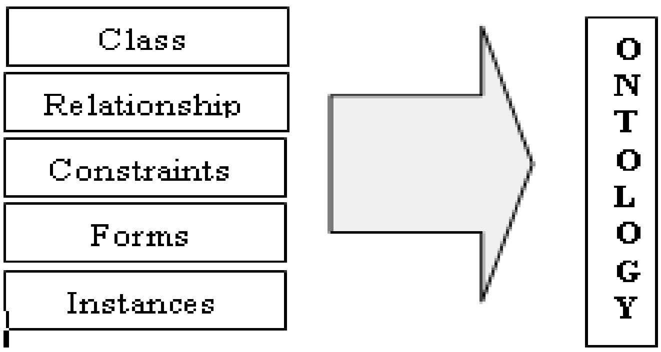


Fig. 2.1 Ontology structure

Using these attribute we have designed the ontology for tourism domain that can be used for the searching purpose and work as knowledge base for semantic web. It includes a collection of domain-specific concepts, and is a system description which includes class-subclass taxonomy, slots, forms, instances, relationships, constraints and performing query in knowledge base. The ontology design process is evolutionary in nature.

2.3 ONTOLOGY IN SEMANTIC WEB

According to Tim Berners-Lee, Director of the World Wide Web Consortium (W3C), refer the future of the current web. Basic idea behind the Semantic Web is to create well-defined ontology in order to provide new capabilities for more effective discovery, automation, integration, and reuse when compared with traditional web technology.

Ontology is a data model that represents a set of concept within a domain and relationship between those concepts. It is used to reason about the objects within that domain. Ontologies are used in artificial intelligence, semantic web, and biomedical informatics as a form of knowledge representation about the world or some part of it [12]. The current Web is based on HTML, which specifies how to layout a web page for human readers. HTML as such cannot be exploited by information retrieval techniques to improve results, which has thus to rely on the words that form the content of the page; hence it is restricted to keywords. In the Semantic Web, pages not only store content as a set of unrelated words in a document, but also code their meaning and structure. The Semantic Web promises to make web content machine understandable, means allowing agents and applications to access a variety of heterogeneous resources, processing and integrating the content, and producing added value output for users.

The success of semantic web is based on the ontology design if ontology is poorly designed then objective of semantic web can't be achieved. Ontology is a description of concepts in a domain of classes or sometimes called concepts, properties of each concept describing various features and attributes of the concept. Classes are the focus of most ontology. Classes describe concepts in the domain. A class can have subclasses that represent concepts that are more specific than the super class. Slots describe properties of classes and instances.

The concept of ontology in tourism area also play a great role to provide exact and sharp search for the domain so that we can get better way to search and manage data over internet than the current methods used to manage data on www.

2.4 SEMANTIC WEB AND CURRENT WEB

2.4.1 Semantic web

The Semantic Web is a mesh of information linked up in such a way as to be easily process able by machines, on a global scale. You can think of it as being an efficient way of representing data on the World Wide Web, or as a globally linked database. The Semantic Web was thought up by Tim Berners-Lee, inventor of the WWW, URIs, HTTP, and HTML. There is a dedicated team of people at the World Wide Web consortium working to improve, extend and standardize the system, and many languages, publications, tools and so on have already been developed. However, Semantic Web technologies are still very much in their infancies, and although the future of the project in general appears to be bright.

The problem with the majority of data on the Web that is in this form at the moment is that it is difficult to use on a large scale, because there is no global system for publishing data in such a way as it can be easily processed by anyone [13].

For example, just think of information about any sanctuary about which don’t know too much you know that it is bird sanctuary in Rajasthan if you search it on current web there are lot false hit links and required information is not available. While in semantic web each data have some URI, property and value which will be defined by data will provide domain specific search so that you can easily find the required information by traversing the hierarchy of data during search. The Semantic Web is generally built on syntaxes which use URIs to represent data, usually in triples based structures [14]: i.e. many triples of URI data that can be held in databases, or interchanged on the World Wide Web using a set of particular syntaxes developed especially for the task. These syntaxes are called "Resource Description Framework" syntaxes.

2.4.2 Semantic web and current web

Semantic web have many advantages and disadvantage as compared to the current web because it is new to the industry and there are lot of work to be done, while current web is on its top span where it get all assistantship from the industry and researchers for the development of the web. But semantic web is the new concept to overcome the pitfalls in current syntactic web and provide a web that will be better and efficient [15]. The main factor on which we can compare the current web and present the advantages of semantic web are-

1. **Data organization:**

   Data in semantic web are arranged in hierarchy of domain while in current web data are unorganized and connected through the hypermedia of links.

2. **Task allocation**

   In the current web computers do the presentation (easy) and people do the linking and interpreting (hard) task Why not get computers to do more of the hard work? That is the main advantage of semantic web.

3. **Searching methodology**

   The current web uses word matching based searching while semantic web is based on knowledge based search.

4. **Data presentation**

   Current web uses the HTML and many other techniques to present data where data is spread freely over the internet while RDF provide data that will have some meaning or relation with each other.

5. **Current position**

   Semantic web is totally new concept and it gets many support and obstacles for the development. It needs too much investment and lot of research to replace the current web. At current scenario there is not too much Ontologies and semantic search engine developed that can present the concept of ontology and semantic web.

2.4.3 Semantic web architecture

The semantic Web architecture is composed of a series of standards. This architecture is often represented using a diagram first proposed by Tim Berners-Lee to present the basic component of semantic web [16] as shown in below figure 2.2

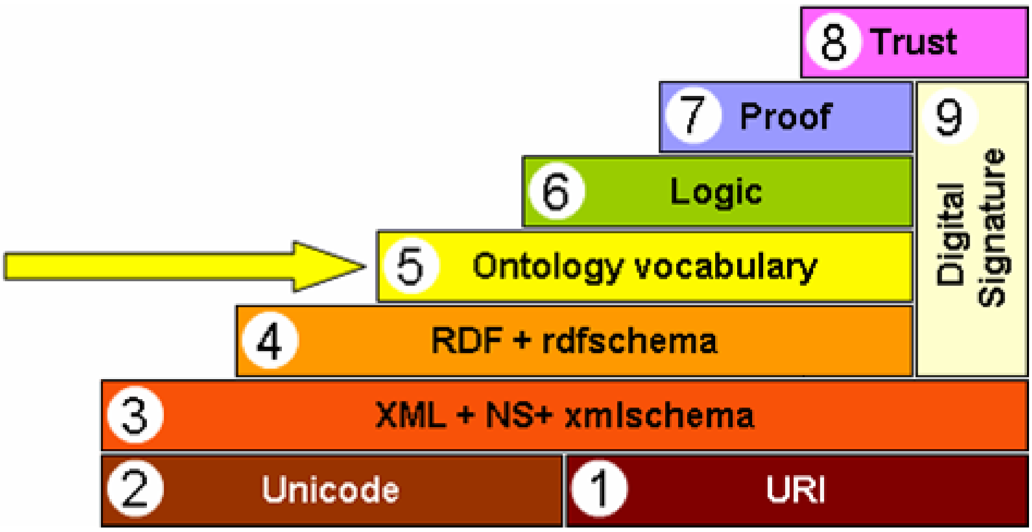


Figure 2.2 Semantic web Architecture

The main components of semantic web are:

- **URI:**

  Uniform resource identifier is used to identify the abstract or physical resource with a unique name.

- **XML**

  XML is the *Extensible Markup Language*. It used to define other languages. XML is a markup language that specifies neither the tag set nor the grammar for that language. The *tag set* for a markup language defines the markup tags that have meaning to a language parser.

- **RDF**

  RDF is the standard defined to present the data in the semantic web

- **Ontology vocabulary**

Ontology vocabulary present the representation of RDF knowledge for any particular domain same as we have used here tourism domain for the representation of tourism domain knowledge. The ontology vocabulary present the concept of ontology used for semantic web as a base for knowledge representation.

- **Logic and proof**

This layer present the query language used to access the RDF data and ontology for search and other application purpose. There are some of query languages such as SPARQL and SRQL that will be used to perform the search. We can put our logic in query facility provided by protégé which is the standard tool for development of any Ontology.

- **Trust**

This layer present the concept that how we can provide the security, interface and other web related properties for better semantic search engine design.

2.5 SEMANTIC WEB BUILDING BLOCK

The main building block and technologies which play central role for the design of semantic web with ontology are following

OWL (Ontology web Language)

OWL has three different sublanguages based on the requirement and complexity of design we will use one of these language [17]. Each sublanguage fulfils different requirements:

a. OWL Lite.
b. OWL DL.
c. OWL Full.

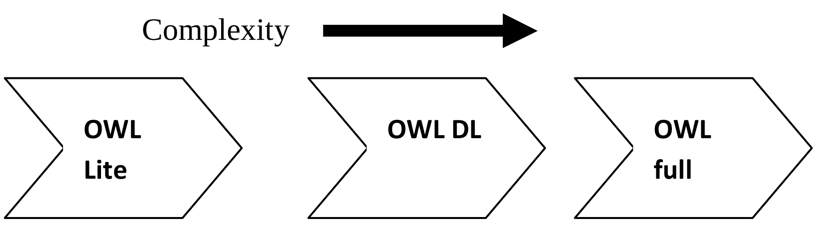


Figure 2.3 OWL complexity

RDF (Resource Descriptive Framework)

The Resource Descriptive Framework (RDF) is a language designed to support the Semantic Web, in much the same way that HTML is the language that helped initiate the original Web. RDF is a framework for supporting resource description, for the web. RDF provides common structures that can be used for interoperable XML data exchange. The main difference between the exiting web and the newer, smarter Web is that rather than a large amount of disorganization of data into knowledge statements – assertions about resources accessible on the web.

RDF Concept and Abstract Syntax defines an abstract syntax on which RDF is based, and which servers to link its concrete syntax to its formal semantics. It also includes discussion of design goals, key concepts, Data typing, character normalization and handling of URI references .RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions of RDF data.

XML

Extensible Markup Language (XML) is a subset of SGML (the Standard Generalized Markup Language), i.e. it is totally compatible with SGML. But it is simple and flexible. It's original aim to tackle the problems of large-scale electronic publishing. However, it is also very important in data exchange on the Web. Despite its name, XML is not a markup language but a set of rules to build markup languages.

Markup language

Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. - Erik T. Ray, Markup language is kind of mechanism organizing the document with a set of symbols, e.g. this article is labeled with different fonts for headings. Markup use similar methods to achieve its aims. Markup is important to implement machine-readable documents since a program need to treat different part of a document individually.

Namespaces

We can expand our vocabulary by namespaces which are groups of element and attribute names. Suppose, if you want to include a symbol encoded in another markup language in an XML document, you can declare the namespace that the symbol belongs to. In addition, we can avoid the situation that two XML objects in different namespaces with the same name have different meaning by the feature of namespaces. The solution is to assign a prefix that indicates which namespace each element or attribute comes from.

The syntax is shown below: ***ns-prefix*****:** ***local-name***

Protégé:

Protégé is the latest tool in an established line of tools developed at Stanford University for knowledge acquisition [18]. Protégé has thousands of users all over the world who use the system for projects ranging from modeling cancer-protocol guidelines to modeling nuclear-power stations. Protégé is freely available for download under the Mozilla open-source license.

One of the major advantages of the Protégé architecture is that the system is constructed in an open, modular fashion. Its component-based architecture enables system builders to add new functionality by creating appropriate plug-in. The Protégé Plug-in Library 3 contains contributions from developers all over the world.

2.6 KNOWLEDGE REPRESENTATION TECHNIQUE

Knowledge representation is a field which is currently seems to have the reputation of being initially interesting, but still it needs a lot of research work to be done. Knowledge representation is basically technique to provide means to the semantic web.

Basic knowledge representation technique for the semantic web is:

- Ontology design
- RDF knowledge base by RDB2RDF Converter

Ontology is the base of semantic web as shown in figure 2.2. Ontology defines a common vocabulary for researchers who need to represent data in meaningful way. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. Ontology is a common set of terms that are used to describe and represent knowledge in an organized way to overcome the problems with current data representation over web and various applications.

RDF converter will be used at base so that we can make a bridge between the semantic web and current web. Ontology create knowledgebase for new domain, perform edition in the ontology while RDF converter provide help for data conversion so that more and more syntactic websites becomes semantic.

CHAPTER III

METHODOLOGY: ONTOLOGY DESIGN

3.1 ONTOLOGY DESIGN APPROACHES:

Ontology and knowledge management have generated considerable interest because they reduce development time and the resources that projects require. Ontologies are used to resolve these obstacles. Initially ontology design process are Ad-hoc in nature and letter some ontology design methods are proposed. To meet the challenge of building Ontologies, we've developed Methontology and a well articulated design techniques.

Earlier ontology design approaches are adhoc in nature latter few domain-independent methodologies for building Ontologies have been reported. Mariano Fernandez Lopez, Asuncion Gomez-Perez, and Juan Pazos Sierra propose one of the ontology creation methodologies [1] which are used for the ontology creation which is more diverse towards conceptualization. However, acquiring knowledge and review for the consistency is not mentioned which play a great role for efficient ontology design. This Methontology framework includes the identification of the ontology development process, a life cycle based on evolving prototypes and the methodology itself, which specifies the steps for performing each activity, the techniques used, the products to be output, and how the Ontologies are to be evaluated. The main steps for the ontology design are knowledge acquisition, evaluation, documentation, configuration management and integration. This approach did not consider the role of domain expert, design structure, languages and maintenance which is one of the important part of any ontology design process.

Recently ontology development approaches are more divorced toward conceptualization one of such ontology design approach is proposed by U.jan, magda, pavel, tereza which is iterative approach for ontology design. This approach is basically based on the definition of the purpose, class and review of the design at each phase and did not consider the expert or domain analysis for the design process.

These ontology design approaches didn't consider the role of Domain Expert which play the central role for the ontology design for any domain so we present an approach with expert analysis and review for the domain and present the methodology for the ontology design which is generic for different domains and also present an approach for the semantic web based on the ontology and considering current web and its data as a base for the semantic web development.

3.2 METHODOLOGY: ONTOLOGY DESIGN PROCESS

There is no one "correct" way or methodology for developing Ontologies Till now. Here we discuss general issues to consider and offer one possible process for developing ontology. To present any approach for ontology design process there is need of deep understanding of ontology process and various issues related to ontology design process. We have made a study and design of ontology for different domain to get the idea about actual things that play role in the ontology design. After the study and design of ontology for various domains such as:

1. Medical ontology design and study
2. Pizza delivery ontology
3. Computer Hardware ontology
4. Panchayati-raj Ontology

With the study and design of these various ontologies we acquired facts and we concede an approach for ontology design which is generic and well articulated. We describe an iterative approach to ontology development: we start with a rough first pass at the ontology. We then revise and refine the evolving ontology and fill in the details. There are some rules that can help while making design decisions design process:

- **a.** There is no one correct way to model a domain while there are always viable alternatives. The best solution always depends on the application.
- **b.** Ontology development is necessarily an iterative process.
- **c.** Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.
- **d.** We also need to remember that ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.

e. After we define an initial version of the ontology, we can evaluate and debug it by using it in applications or problem-solving methods or by discussing it with experts in the field, or both.

We start the development of ontology by defining its domain and scope. This can be decided using several basic questions:

a. What is the domain that the ontology will cover?

b. For what we are going to use the ontology?

c. For what types of questions the information in the ontology should provide answers?

d. Who will use and maintain the ontology?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

There are several possible approaches in developing a class hierarchy in ontology to define any real life group of entity suing class. Classes will correspond to objects, or types of objects, in the domain. In the tourism, there are around 70 classes to represent the concept such as destination, lakes, fort, museum, city etc

There are mainly three approaches to define classes:

a. A top-down development process

This approach starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts. Here abstract class plays a role in the design process for the domain. Abstract class are here used for creating the subclass only, it is not used for defining the subclasses. The top-down approach is based on the hierarchical design of the process. For example, we can start with creating classes for the general concepts of destination. Then we specialize the destination class by creating some of its subclasses: fort, pilgrimages, nature etc.

b. A bottom-up development process

This approach starts with the definition of the most specific concepts in the domain and subsequent generalization of the concepts starts with the definition of the most generic classes, the root of the hierarchy, with subsequent grouping of these classes into more general concepts.

c. **A combination development process**

This approach starts with the definition of the most important class whether it is on top or bottom of hierarchy for the design. Classes are building around any concept for the design process. It defines a combination of the top-down and bottom-up approaches: We define the more salient concepts first and then do generalization and specialization on them appropriately.

During the design process the selection of class should have following characteristics for better ontology design.

a. Class name should be generic, means it is capable to represent the concept behind the class.
b. The class hierarchy should be optimal means there should not be too much class at the same level as well as the depth of specialization should be normal up to some depth based on the application domain.
c. Classes should be arranged in such as way that there will be minimum redundancy in knowledgebase.
d. The instance or value in the class diagram should be present at the leaf node for better search and constraints definition.

3.3 ONTOLOGY DESIGN PROCESS

Ontology design is Ad-hoc in nature now days. There are many problems during the ontology design process between concepts and becomes real for particular domain after resolving many bottlenecks for ontology design. For ontology design process there is no such protocol that tells how to construct ontology for any domain. we present an ontology design process from our past ontology design approach for various domain such as medical, computer hardware, pizza delivery, Panchayati-raj and tourism ontology. To design ontology for a specific domain there are some necessary steps involved for the efficient ontology design process. For efficient ontology design author proposed a general ontology design process involving steps for the ontology design. These ontology design steps provide a generic approach for the ontology design and present a better way to design an efficient ontology for any domain. The process of ontology construction can be divided into several steps as shown in figure 3.1

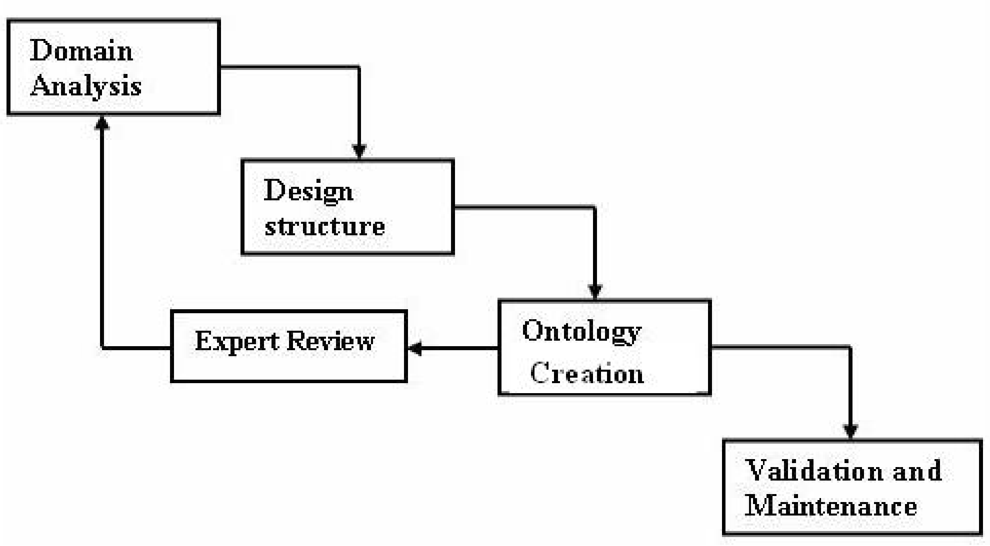


Fig 3.1 Ontology design process

1. *Expert Analysis*

For analysis we need an expert of the particular domain having the knowledge about the knowledge representation for that domain. First step in ontology design process is domain analysis where scope and knowledge source for the domain is analyzed by the domain expert such as for medical it may be any doctor or medical representative who has wide knowledge of medical domain and for tourism domain we have used tourist agent from Rajasthan to analysis the scope and knowledge source for the tourism domain. The expert analysis will help the designer to concede the concept and relationship between these concepts in better way. The expert will cover the following main issues regarding ontology:

a. *Ontology Scope*

One of the main goals of analysis process is to discuss the scope of ontology such as what are its application and advantages with respect to current knowledge representation scheme. During the analysis and discussion of scope for the ontology design the main concern is on the domain that how is will work for particular domain based on the expert analysis. Here we will consider the basic questions regarding the domains such as

a. What are its applications and who are users.

b. How it help users such as tourist.

c. What are general questions from a tourist?

d. What is better way to resolve tourists query.

e. How much time and cost it will take for the knowledge acquisition.

These are some of the general query for the ontology design process and how to overcome it is one of the main challenges.

b. Knowledge Source

In addition to scope, it is important to determine the sources of knowledge base whether it is from current web or start from scrap. The main knowledge source for building ontology is the related data over internet or concern department. The other sources of knowledge include the user, designer, expert, other ontologies or terminologies and most importantly, experts' knowledge. Knowledge source is the most important for ontology design process because the knowledge base of ontology is derived from knowledge base. It will present the whole analysis and representation of the data source and gathering of this information in an appropriate way so that another user will get benefits from it.

2. Tool and Languages

Second step in ontology design process is to decide which tool and language will be used for ontology design. The ontology development tool are used for easily and efficient design of ontology. There are many ontology development tools are available such as

- Protégé from Stanford.
- SWOOP from W3C now days developed by Google.
- Gene ontology tool by Gene Ontology Consortium
- XML Ontology editor by EXL.com

These are some of the ontology development tool available for the ontology design process after the study of various advantages and disadvantages we prefer to use Protégé because of some of advantages over other tools such as

- Better ontology representation
- Freely available to uses
- Less memory space for the ontology design
- Continuous development and updating in the tool

These characteristics made protégé as a better choice for us and we have used it for the ontology design process.

To support these tools we have ontology language which is OWL (Ontology web language). The most recent development in standard ontology languages are OWL from the W3C. Like Protégé, OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts.

The **Web Ontology Language** (**OWL**) is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL DL and OWL Lite semantics are based on Description Logics,[1] which have attractive and well-understood computational properties, while OWL Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax

The W3C-endorsed OWL specification includes the definition of three variants of OWL, with different levels of expressiveness.

OWL Lite was originally intended to support those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems utilizing thesauri and other taxonomies. In practice, however, most of the expressiveness constraints placed on OWL Lite amount to little more than syntactic inconveniences: most of the constructs available in OWL DL can be built using complex combinations of OWL Lite features. Development of OWL Lite tools has thus proven almost as difficult as development of tools for OWL DL, and OWL Lite is not widely used.

OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (all conclusions are guaranteed to be computed), decidability (all computations will finish in finite time), and the availability of practical reasoning algorithms. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, number restrictions may not be placed upon properties which are declared to be transitive). OWL DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL.

OWL Full is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for OWL Full.

So to design ontology in better way we have used **OWL DL** for tourism ontology with the help of protégé.

3. Ontology Creation

Ontology design covers the design of framework for ontology by user, expert and designer to represent knowledge in efficient way. The main issue in ontology design involves:

a. Design a framework

First we design a framework that will be used to define the class, subclass and relations between them. For higher-level ontology, then one also has to use concepts that are more abstract. Many higher-level classes would have to be only abstract, thus containing no individuals. These classes would create a framework for other, more specific classes to fit in. the constructed framework should cover the all aspects of a domain to construct ontology. Framework designed for ontology work as a base during the design of ontology. It will provide the basic outlays of the knowledge representation with ontology defining the class and subclass to understand the system in better way and present knowledge with efficient way. For the tourism domain we will define first the abstract class and then further explore it based on the domain analysis for any domain.

b. ***Knowledge representation***

After the framework design the next step is to represent knowledge using the framework defined by expert and knowledge source. Using some appropriate tool and language we represent the knowledge using ontology that is should be reusable, expandable and evolvable in nature.

4. Validation and maintenance

After the ontology design we have to perform validation check for the ontology to check the quality and knowledge representation efficiency of the ontology. If there is constraints violation during the design then we will remove it using design phase. The main things which will consider during the validations are:

- Quality of designed ontology, whether it full fill the goal of ontology design with respect to the previous knowledge representation technique and goal of ontology design
- Constraints violation, the designed ontology should not violate any constraints in the ontology design such as you can not put any character in the date instance.
- Knowledge representation efficiency, consider that how efficient is the knowledge representation using the ontology and how much it will fulfill the goals of the ontology.

The next step is the maintenance of the ontology on the web. It will consider the changes which we needed after the implementation such as there is need of change in knowledge or there is change in the instance then it will provide the necessary changes in the system without affecting the rest of the system. Maintenance of the system is the main part that decides how successful the designed ontology is for the long term regarding the changes in the system.

5. *Expert Review*

Expert review is used in ontology design to overcome the shortcomings in ontology design and perform the necessary changes in design phase. An expert for particular domain can easily examine the ontology in better way and changes suggested by expert will be performed. The expert review made ontology process evolutionary in nature during the design process of ontology. The main things that will be considered during the expert review are:

- How close is system to goal of design?
- Is there any need of changes?
- How better is it with respect to current system?
- What changes are needed?

These are some of the question which will help the expert to make changes based on the review of the system.

3.4 USES OF ONTOLOGY

The semantic web model presents the uses of the ontology and what is its role in the current web. One of the uses of ontology based knowledge representation is for the various AI Applications. Author presents a conceptual semantic web model that shows how to present the knowledge in semantic web. The composite structural model to present the knowledge in semantic web is as shown in figure 4.1. This model presents the role of knowledge in various applications such as search, web portal and mobile data representations.

Here we present the knowledge representation techniques with ontology and RDF for any domain we also need to present the current relational data into knowledge base which will be done with RDB2RDF Converter so that all data are present in RDF/XML. The fig. 4.1 present a model for knowledge representation in semantic web and applications related to it. The knowledge is basically present in RDF/XML form which is machine understandable format so it is used in areas such as searching, intelligent information integration and knowledge representation in semantic web. Ontology was used to define the elements characterizing the knowledge elements necessary for tourism or any domain. Semantic web application is build over the ontology and RDF as a base for any semantic web application.

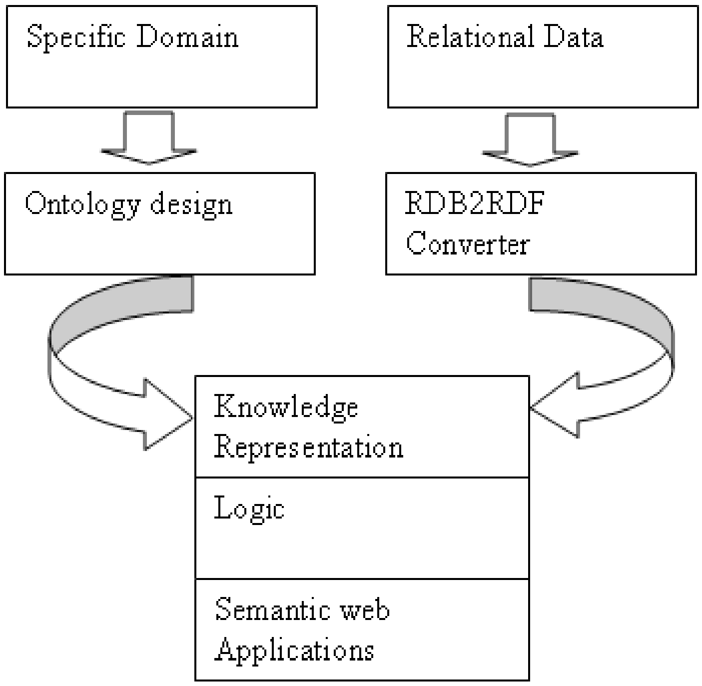


Fig. 3.2 Semantic web model

Semantic web model present the uses of knowledge in various domain and areas according to the need of researchers such as semantic web portal, web search, mobile data representation. So knowledge representation need more work to be done in the area so that we can uses the benefit of knowledge in place of data. Ontology provides a common vocabulary for researchers who need to share information in the domain. Some of uses of ontology are:

a. To share common understanding of the structure of information among people or software agents
b. To enable reuse of domain knowledge
c. To separate domain knowledge from operational knowledge
d. To analyze domain knowledge
e. Intelligent search instead of keyword matching.
f. Query answering instead of information retrieval.

g. Representing: XML, RDF, and OIL are used for describing syntax and semantics of semi-structured information sources.
h. To represent the data in the RDF format that can be used for semantic web development.

CHAPTER IV

TOURISM ONTOLOGY

4.1 INTRODUCTION

The current tourism data representation approach on web couldn't satisfy the great deal of representation and management of exponentially increasing data on web. Tourism is domain where data are unorganized and spread in hundred of data source so there is need to organize and represent it in efficient way. The organization and management of tourism data is the main issue for tourism knowledge representation. Today's tourism environment covers many issue related to specific destination, accommodation, travel, climate, people, and city. To retrieve complex knowledge in current web it fails sometime and gives false hit during searching such as *accommodation at bird sanctuary for particular city*" to overcome these problems author represent the ontology design for tourism domain.

Ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. The knowledge management and organization of data is the key element of Semantic web, which will be illustrated by Ontology. Thus, Ontology provides a fundamental framework for the development of Semantic Web and applications. For the Tourism domain one has to specify the scope of the ontology, acquire tourism knowledge, select a tool and an ontology language, design the ontology, and present it in an appropriate way. Special attention is devoted to the problem of representing relevant knowledge in the form of ontology. The ontology is the heart semantic web and efficient search. We established an ontological structure including destination, travel, city, people and accommodation to organize information in class subclass taxonomy.

Ontology design approach for tourism domain

Ontology design for any domain should cover all aspect of any domain related to entity, instance and relations. Ontology development mainly uses two approaches for the design of ontology for a particular domain. Each has different complexity and advantages during the design of ontology for any domain. These two approaches are:

a. **Ontology using RDF:** In this approach we have to define each and every class, relation and attribute using RDF/XML format with RDF as a language for the development of ontology. This approach is need great skill in RDF and XML to represent knowledge regarding any entity. This approach is time consuming and defining constraints are not so easy while writing codes so generally we will prefer second approach for ontology design.
b. **Ontology design using protégé:** Protégé is an integrated software tool from Stanford University used by system developers and domain experts to develop *knowledge-based systems*. The ontology defines the set of concepts and their relationships. The knowledge-acquisition tool is designed to be domain-specific, allowing domain experts to easily and naturally enter their knowledge of the area. The resulting knowledge base can then be used with a *problem-solving method* to answer questions and solve problems regarding the domain. Finally, an *application* is the end product created when the knowledge base is used in solving an end-user problem employing appropriate problem-solving, expert-system, or decision-support methods. Protégé is designed to allow developers to reuse domain Ontologies and problem-solving methods, thereby shortening the time needed for development and program maintenance. Several applications can use the same domain ontology to solve different problems, and the same problem-solving method can be used with different Ontologies. We have used protégé tool for design of ontology and the expert data is collected from all major tourism system to make ontology just like a domain expert to fulfill all knowledge regarding tourism and that will cover the all aspect of the tourism domain in terms of representation, efficiency and organization of data for particular domain.

4.2 ONTOLOGY DESIGN FOR TOURISM DOMAIN

4.2.1 Tourism Domain:

Tourism play a great role in the GDP of India while data over the internet are not organized which is a great problem for any tourist to decide his destination of his interest so there is need to do something work in the field of tourism web development so I have selected Tourism as a domain with a goal to provide the first ontology for tourism domain in world at the same time provide semantic search engine for tourism domain. To resolve this real life problem I have decided to develop a ontology for tourism domain taking the whole knowledge from tourism domain of Rajasthan state where tourism play a great role in the growth of economic and social life .

Tourism ontology provides a way to achieve integration and interoperability trough the use of a shared vocabulary and meanings for terms with respect to others so that we can further extent it or reuse it for the development of ontology.

The ontology was build to answer the following main questions:

a. What a tourist want to see or visit based on the desired categories he can search the information for a particular categories.
b. Where are the interesting places to see and visit located
c. When can a tourist visit a particular place using the transport services?
d. Information for domain specific such as about people, about tourism department
e. What is your domain for interest from the places such as pilgrimage, fort, garden, lakes, sports etc?
f. Information related to accommodation based on your budget and package you can select hotels.

4.2.2 Scope of ontology

We start the development of ontology by defining its domain and scope. This can be decided using several basic questions:

e. What is the domain that the ontology will cover?
f. For what we are going to use the ontology?
g. For what types of questions the information in the ontology should provide answers?

h. Who will use and maintain the ontology?

The answers to these questions may change during the ontology-design process, but at any given time they help limit the scope of the model.

4.2.3. Ontology Design Process

Class hierarchy of tourism ontology contains around 70 classes that are arranged in such a way that the meaning of each class is clear and we can perform the traversing on the classes based on need we want to search in the knowledge base.

The below fig. 5 show the class diagram for ontology domain:

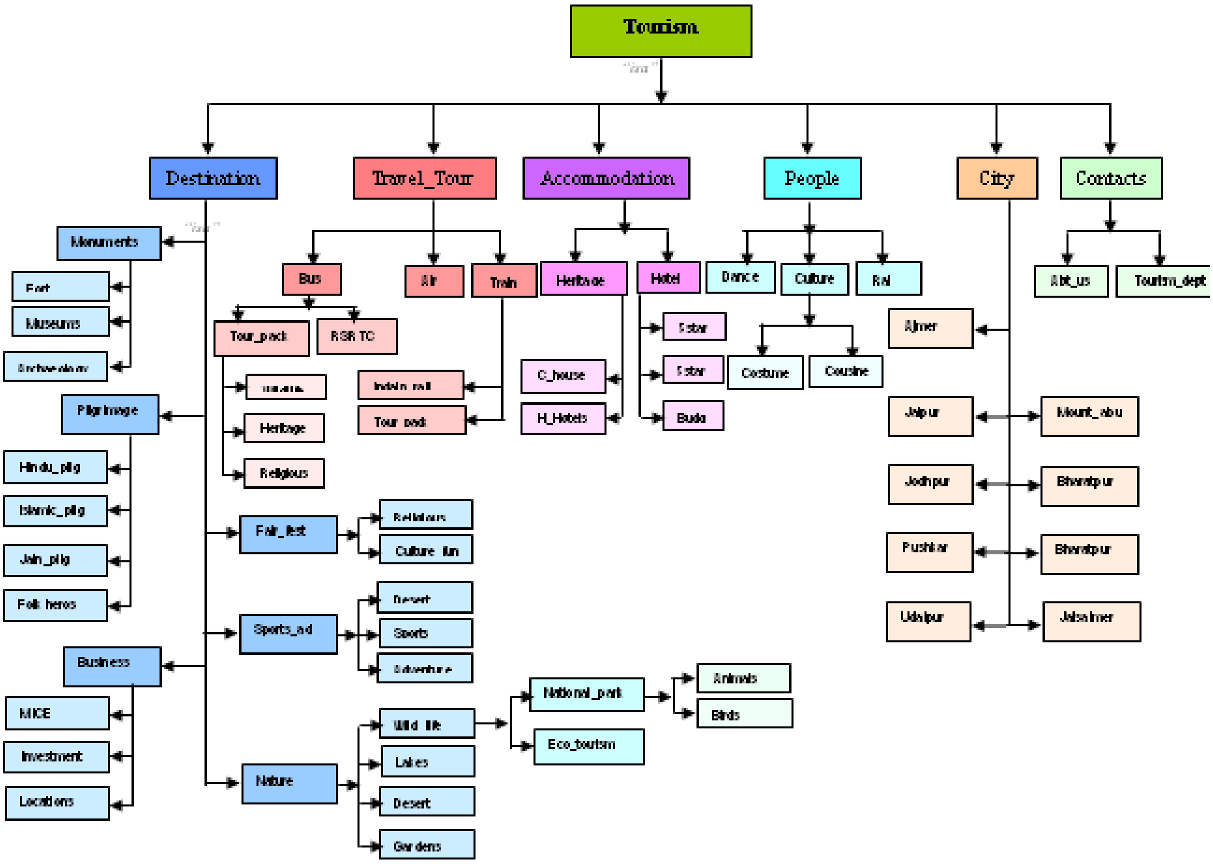


Figure 4.1 Ontology design framework

The ontology design process using protégé contain the four main parts to design a ontology for a particular domain:

a. Define classes

This is the process to define a class and its subclasses using the protégé for the development of ontology. The class window for the ontology is as shown in figure 6

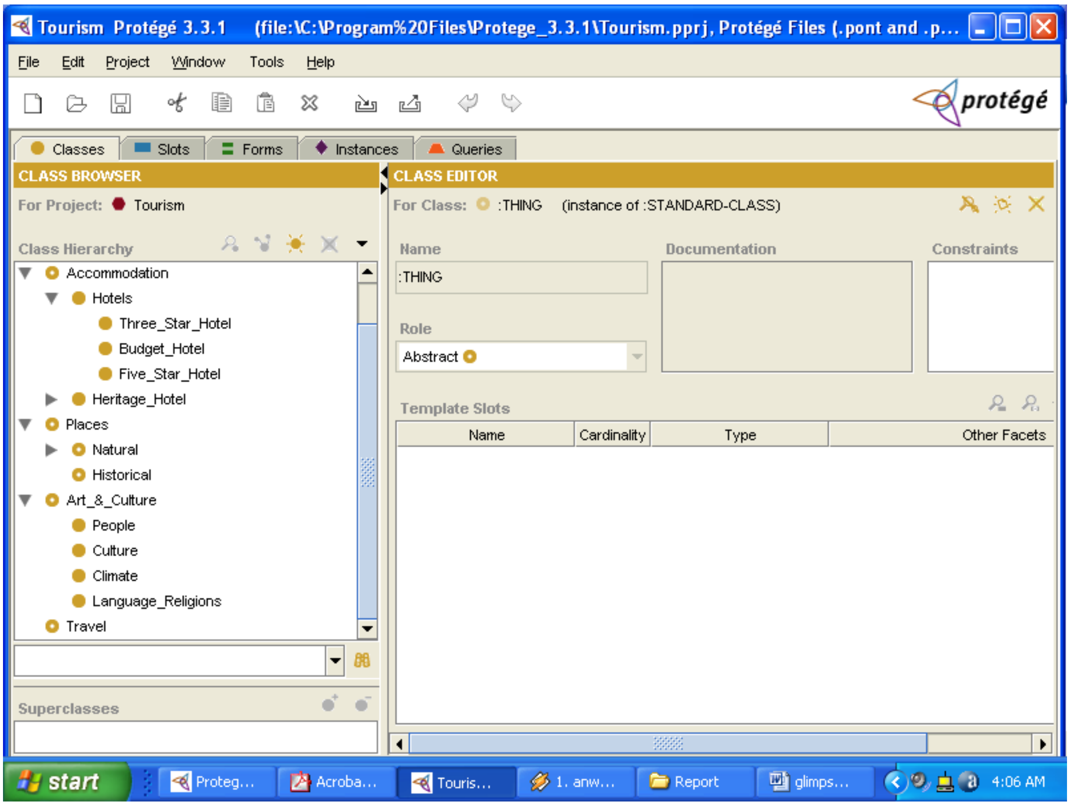


Figure 4.2 defining class using protégé

The class definition window contains the information about the class whether it is concrete or abstract and detail about the sub class, super class hierarchy for data management.

b. Creating slots

In Protégé classes can be thought of as concrete concepts from the domain, such as Editor and Columnist. Classes are more than simple objects arranged in a hierarchy. They can also have attributes, such as a name, phone number, or salary, and relations between them, such as the Author of an Article. Class attributes and relations are described using *slots*.

The below protégé snapshots fig. show that how slots are created and how we will define the value in it.

During the slot values definition we must take care about these things –

a. Slot name should show the meaning so that it will help during the searching process.

b. Data type of slot define that whether slot is string, number or any relation between the two classes.

c. Slot value gives us the facility to provide coordinate, min., max. Value, default value for a slot.

d. We can use some already defined slot for our application.

c. Creating form

Form in the ontology is used to present the layout for the data that will be presented when we search some instance of a class. So it is a frame where we will present the knowledge in the ontology design Form provides the interface kind of structure for providing data a place to represent while searching.

The form provide following advantages:

a. Provide way to represent the data.
b. Layout management facility
c. Layout inheritance facility from other already defined forms.

So these are the main facilities provided by form to the ontology.

d. Entering instances

Instances are the actual data in your knowledge base. In general, it is a good idea to make sure you has structured your project as well as you can before entering extensive numbers of instances. In addition, if you add slots, you will have to go back and fill in the slot values for all instances that were created previously.

The instance is the real data which we have to store and retrieve during the searching operation in ontology.

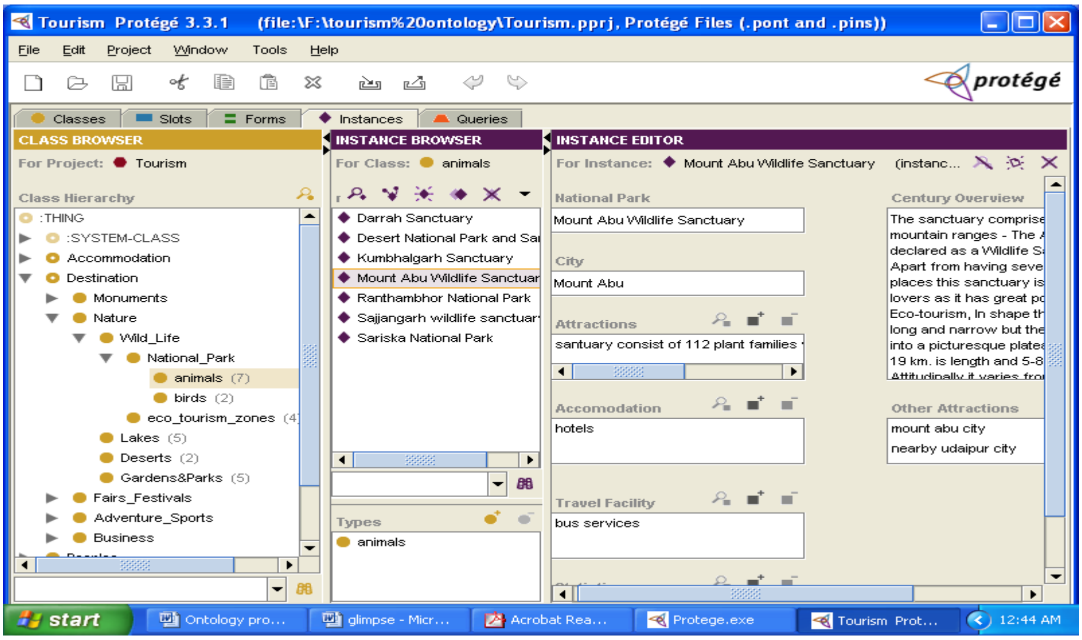


Figure 4.3 Instance in ontology

e. Query in ontology

Performing query is one of the main tasks of ontology which will provide a way to search the desired information by ontology. The Queries tab allows you to query your project and locate all instances that match criteria you specify. To create a query, you must select one or more classes, and one or more slots within that class. You can also save queries in the Query Library for future recall. The main functions that will be provided by ontology are;

a. Creating a query
b. Running a query
c. Saving a query
d. Retrieving a query

The protégé tool can be used to perform the query in knowledgebase suppose you want to find the all "**lakes at Udaipur**". The above fig show that the protégé return the two instances lake pichola and fateh Sagar Lake as a search result which is exact there is not any false hit in the searching approach. To find the result we have to create a query for which it need three things:

a. Select the class from the class hierarchy of your need and interest.
b. Specify the slot for which you are looking.
c. Enter the desired string for search.

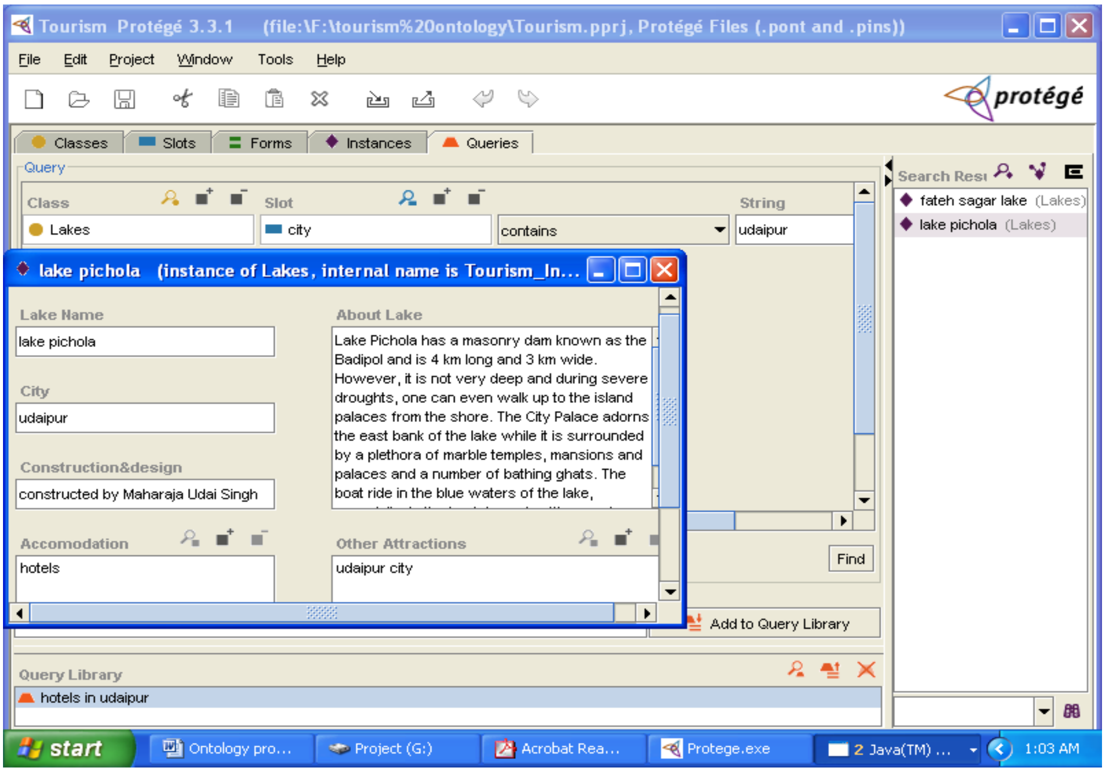


Figure 4.4 Query in ontology

Using these three value protégé ontology tool will return the desired result which can be saved as query for further use and the result will be exact. So query in ontology is basically based on three things class, property and value.

CHAPTER V

RELATIONAL TO RDF CONVERTER

5.1 RDF (Resource Description Framework)

The Resource Descriptive Framework (RDF) is a language designed to support the Semantic Web, in much the same way that HTML is the language that helped initiate the original Web. RDF is a framework for supporting resource description, for the web. RDF [16] provides common structures that can be used for interoperable XML data exchange. The main difference between the exiting web and the newer, smarter Web is that rather than a large amount of disorganization of data into knowledge statements – assertions about resources accessible on the web.

RDF Concept and Abstract Syntax defines an abstract syntax on which RDF is based, and which servers to link its concrete syntax to its formal semantics. It also includes discussion of design goals, key concepts, Data typing, character normalization and handling of URI references .RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions of RDF data. The development of RDF has been motivated by the following uses, among others:

a. Providing information about Wed resource and the systems that use them (e.g. content rating, privacy preference, etc).
b. To do for machine process able information (application data) what the World Wide Web has done for hypertext: to allow data to be processed out side the particular environment in which it was created, in a fashion that can work at Internet scale.
c. Networking among applications: combining data from several applications to arrive at new application.

RDF is designed to represent information in a minimally constraining, flexible way. It can be used in isolated applications, where individually designated formats might be more direct and easily understood, but RDF generality offers greater value from sharing. The value of information thus increases as it becomes accessible to more applications across the entire Internet.

5.1.1 RDF in semantic web

RDF is used as a base for the semantic web because of the following main characteristics of the RDF:

a. Independence

Since a Property is a resource, any independent organization (or even person) can invent them.

b. Interchange

RDF Statements can be converted into XML; they are easy for us to interchange.

c. Scalability

RDF statements are three-part records (Resource, Property, value), so they are easy to handle and look things up by, even in large numbers.

d. Properties are Resources

Properties can have their own properties and can be found and manipulated like any other Resource.

e. Statements Can Be Resources

Statements can also have properties.

These are the main characteristic which will make RDF a special language for the semantic web development and application where it can be used to present the data in efficient way.

5.1.2 RDF Model

RDF Data Model for representing named properties and their values.

These properties serve both to represent attributes of resources and to represent relationships between resources. The RDF Model is defined by RDF Triple, in which we need three piece of information to fully define a single bit of knowledge.

a. Subject / resources

b. Properties

c. Statements / Value

These three objects will make a meaningful RDF statement used for modern web.

a. Resources

- All things being described by RDF expressions.
- Resources correspond to objects.

Various types:

- An entire Web page; e g: "http://www.w3.org/Overview.html".
- A part of a Web page; e.g. a specific HTML within the document source.
- An object that is not directly accessible via the Web; e.g. a printed book.

b. Properties

- A specific aspect, attribute, or relation used to describe a resource.
- Relationships between resources.
- Properties correspond to instance variables.

c. Statements

- A specific resource together with a named property plus the value of that property for that resource.

In RDF, a statement is called a triple because it has three parts. The subject of a statement is in fact called the subject. The equivalent of a verb is called the predicate, and the remaining part is called the object. Other terms are also in common use: property instead of predicate, and value instead of object (because many RDF statements assign property values to their subjects). The following diagram depicts the structure of an RDF triple:

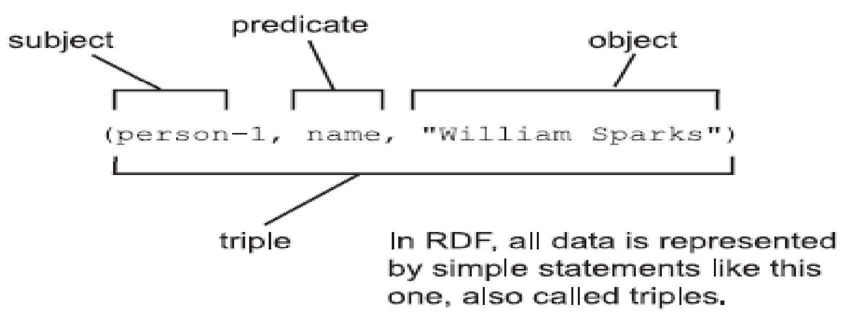


Figure 5.1 RDF Triple

The value of a property can be a simple value, like an ordinary number or a string of characters such as William Sparks in the diagram. Such values are called literals. The value of a property can be either a literal or another resource, as appropriate. RDF has a way of indicating whether a literal value has a data type, meaning that it is intended to be, for example, an integer or a chunk of XML. A literal can't be the subject of a statement. A collection of RDF data has no standard name. It's sometimes called an RDF store, an RDF data set, a knowledge base.

Many kinds of notations can be used to display RDF data. It turns out that directed; labeled graphs are excellent for representing RDF statements. A graph is a collection of nodes or vertices connected by edges (sometimes called arcs). In a labeled graph, the edges carry labels; in a directed graph, the edges point from one node to another.

RDB2RDF CONVERTER

The basic objective of our project is to convert a Relational schema into RDF memory model. To achieve this goal, we intend to make a Relational database in Microsoft Access. This database may contain one or more tables. Then we have to make DSN of this database to enable JDBC/ODBC connectivity. After this we intend to use JENA API which will facilitate to build RDF memory model. This will include gaining knowledge about JENA API, its working and its usage.

Resource Description Framework (RDF) is widely used in Semantic Webs to make machine-understandable of data among Webs. RDF Schemas (RDFS) help RDF defined properties (attributes), kinds, and relationships of resources in RDF documents. At present, maximum data on web are present in Relational Database form so to assist semantic web there is need to convert a relational database into RDF form. When maximum data are in the machine understandable form on web it is easy to design semantic web application using the existing data on the web. The RDB2RDF converter also draws the attention of researcher to design tool for Query processing, optimization and various other issues with SPARQL [20] and SERQL for knowledge management in RDF instead of RDF2RDB Conversion for optimization and query. JENA API forms the core of the project, therefore, a detailed discussion on methods of Jena API are given in the report. Also the desired output in RDF code and RDF triples, and the primary key of the relational tables is provided by the JENA API. We get the output in the text file.

Resource Description Framework (RDF) is designed for specific data about specific Subjects. RDF and similar technologies can represent data and metadata equally. To make all this data accessible to computers, we must do it with a language they can understand. We'll also have to make it addressable over networks like the Internet.
RDF was developed to provide a standard way to model, describe, and exchange Information about "resources"—that is to say, practically anything about practically anything—as long as it is specific data about a specific subject. RDF can be used to model and exchange Ontologies. We are creating data of data which are meta data .In which their are number of database tables which are to be used as the input to the Mapper .then we are connecting that meta data tables to the drives through ODBC.
Then we create the java programs to access the Meta data .the java program are connect by JDBC to the DSN. Then by Jena API methods generates the RDF code and RDF triple. That RDF code and the RDF triple are finally saved in the text file. These RDF code are used in the semantic web for efficient searching.

5.2.1 System Architecture

In order to transform Relational database (RDB) to RDF/XML and RDF Triple form, the relational documents will be loaded into the RDB2RDF Transformation Engine, which will convert Relational database into RDF form with the help of Java and JENA packages, which will provide interface to create memory model and instances for a relational database Tuple. The output of the RDB2RDF converter is stored in text file which will be used for query and semantic web applications. The architecture of RDB2RDF converter is shown below where we will use Relational table as an input and output is in the RDF format that will assist the semantic web applications and efficient searching for semantic web.

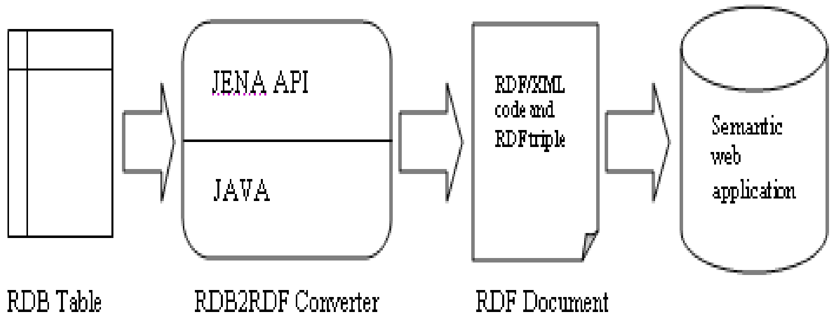


Fig. 5.2 RDB2RDF architecture

5.2.2 Functional Diagram for RDB2RDF Converter

A Functional diagram is a diagram that indicates the functions of the principal parts of a total system and also shows the important relationships and interactions among these parts. It also describes a function between input variables and output variables. A function is described as a set of elementary blocks. Input and output variables are connected to blocks by connection lines. An output of a block may also be connected to an input of another block: Inputs and outputs of the blocks are wired together with connection lines, or links. Single lines may be used to connect two logical points of the functional diagram.

Functional diagram for RDB2RDF Converter present the basic function of the tool. It shows the input and output for the system and also presents the general function of the system presenting Jena, relational schema and output format for the system. It will provide the processing on the data and various processes used in the system design. the below Functional diagram will show that how we provide the input a relational database and that will be converted into RDF Memory Model by Jena API Methods and that will provide RDF/XML Code and RDF Triple as Output in the system, that will be used for the knowledge representation in various application domains.

The functional diagram is shown in figure 5.3

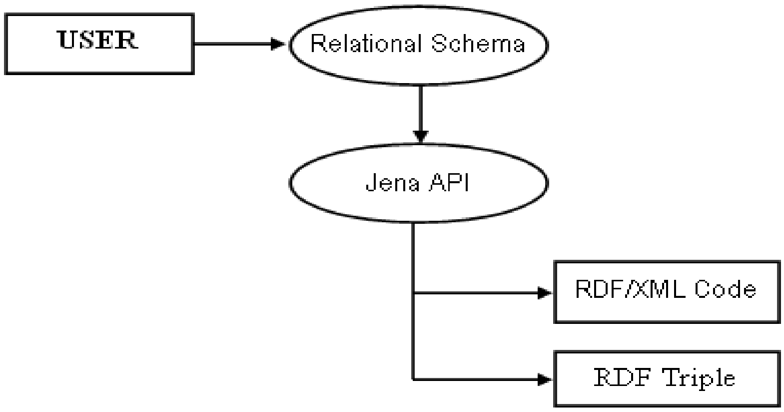


Figure 5.3 Functional Diagram of RDB2RDF Converter

The above DFD show that how a user provide a DSN name of relational data and after the Jena API it becomes converted into RDF code and RDF Triple which will be used to represent the data.

5.2.3 Implementation with Java:

RDFB2RDF Converter is implemented with the help of JAVA and Jena API. The code for the conversion is as presented here:

Jena Classes:

```
import com.hp.hpl.jena.mem.*;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.Property;
```

Java Classes:

```
import java.sql.*;
import javax.swing.*;
import java.io.FileOutputStream;
```

```
import java.io.PrintWriter;
import java.io.*;
import java.io.BufferedReader;
import java.io.File;
```

Implementation Code:

```
import java.sql.*;
import com.hp.hpl.jena.mem.*;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Model;
import com.hp.hpl.jena.rdf.model.Property;
import javax.swing.*;
import java.io.FileOutputStream;
import java.io.PrintWriter;
import java.io.*;
import java.io.BufferedReader;
import java.io.File;
 class s10
{
        public static void main(String args[]) throws Exception
        {
                String suri="";
                        Model model;
                String ch,ch1, con, in1,a,b;
                ResultSet results, pkey, ilist;
        FileWriter file=new FileWriter("test.txt");
        FileWriter file1=new FileWriter("test1.txt");
           String srealted="";
                System.out.println("checking if driver is registered with driver manger");
```

```
try
{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        System.out.println("find the driver");
}catch(ClassNotFoundException cnfe)
{
        System.out.println("couldn`t find the driver!");
        System.out.println("let`s print a stack trace and exit");
        cnfe.printStackTrace();
        System.exit(1);
}
        System.out.println("registered the driver ok,so let`s make a connection");
        Connection c=null;

con="jdbc:odbc:"+JOptionPane.showInputDialog("enter DSN name ");
        try
        {
                c=DriverManager.getConnection(con," "," ");
                Statement smt=c.createStatement();
               String [] tabletypes={"TABLE"};
                DatabaseMetaData dbmd=c.getMetaData();
           ResultSet alltables=dbmd.getTables(null,null,null, tabletypes);
           while (alltables.next())
                {
                String tname=alltables.getString("TABLE_NAME");

                        ilist=dbmd.getIndexInfo(null,null,tname,false,false);
                while (ilist.next())
                {
                try{
                a=ilist.getString("INDEX_NAME");
```

```
		if (a.equals("PrimaryKey"))
			System.out.println("indexname   " + a);
		{
		 b=ilist.getString("COLUMN_NAME");
			System.out.println("cname    " + b);
			suri="http://"+ b;
		}

		//	pkey=dbmd.getPrimaryKeys(null,null,tname);
	//	ilist.close();
		}
		catch(NullPointerException e)
		{
			e.getMessage();
		}
		}
			//	System.out.println(tname);
			in1="Select * from "+tname;
		 results=smt.executeQuery(in1);
		 ResultSetMetaData meta=results.getMetaData();
		if(results!=null)
		{
			int count;
			 count=meta.getColumnCount();

			while(results.next())
			{
			 model=ModelFactory.createDefaultModel();
			 Resource postcon=model.createResource(suri);
			for(int i=0;i<count;i++)
			 {
```

```
String s1=meta.getColumnLabel(i+1);
String s2=results.getString(s1);
String spostcon4="http://"+s1;
Property realted=model.createProperty(spostcon4,srealted);
postcon.addProperty(realted,s2);
}
model.write(new PrintWriter(file));
//        BufferedReader    br=new    BufferedReader(new
InputStreamReader(
//   String p=br.readLine();
//   System.out.println(p);
//   file.write(p);
model.write(file1,"N-TRIPLE");
}
}
results.close();

}
}
catch(SQLException se)
{
System.out.println("couldn`t connect:printout a stack trace and
exit");
se.printStackTrace();
System.exit(1);
}
}
}
```

APPLICATION PROGRAM ITERFACE(API)

Application Programming Interface, a language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol. APIs are implemented by writing function calls in the program, which provide the linkage to the required subroutine for execution. Thus, an API implies that some program module is available in the computer to perform the operation or that it must be linked into the existing program to perform the tasks. The API is designed to facilitate the user and provide a better way of interaction. There are multiple real life problems where it is not possible that each user can understand a problem in a way a designer handle, API provide an interaction so that each end user can handle a problem in an easy way and perform the desired task with an ease.

One of the well known API is the packages are written in java since that runs well on all platforms. These should include:

a. Be as uniform as possible in their interface.

b. be as interactive as possible -- with no "hard wired" input.

c. be as graphical as possible to provide graphical insight so that it will

Provide a better way of interaction with the problem.

API basically provides the way of interaction to user. There are multiple languages which will use the different API according to their need and facilities provided by these APIs. RDF convertor have used mainly two APIs to convert relational to RDF data these are

- **Java API**

JDBC stands for "Java Database Connectivity". It is an API (Application Programming Interface) which consists of a set of Java classes, interfaces and exceptions and a specification to which both JDBC driver vendors and JDBC developers adhere when developing applications. JDBC is a very popular data access standard. RDBMS (Relational Database Management Systems) or third-party vendors develop drivers which adhere to the JDBC specification. The

JDBC API provides universal data access from the Java programming language. Using the JDBC API, we can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built. In simplest terms, a JDBC technology-based driver ("JDBC driver") makes it possible to do three things:

a. Establish a connection with a data source
b. Send queries and update statements to the data source
c. Process the results

- **Jena API**

Jena was developed by Brian McBride of Hewlett-Packard and is derived from earlier work on the SiRPAC API. Jena allows one to parse, create, and search RDF models. Jena has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively. In Jena, a graph is called a model and is represented by the Model interface. The Jena Framework includes:

a. A RDF API
b. Reading and writing RDF in RDF/XML, N3 and N-Triples
c. An OWL API
d. In-memory and persistent storage
e. query engine

RDB to RDF Conversion

The basic concept behind the conversion from RDB to RDF is the use of Jena API which will create the memory model for equivalent data. To convert a relational table author employs use of primary key for the traversal in RDB Tuple for convert it into equivalent RDF code. The relational table "Tourism_officer" containing information regarding name, contact and city can be converted into RDF form to represent the knowledge in RDF format

cid	Officer_name	City	contactno
1	Kartar	Jaipur	1414323234
2	bhasker	Delhi	114324234
3	diwakar	kolkata	224323543

Fig. 5.4 Relational database table

The equivalent code generated in RDF/XML and RDF Triple form by RDB2RDF Converter is as shown in figure 5.5.

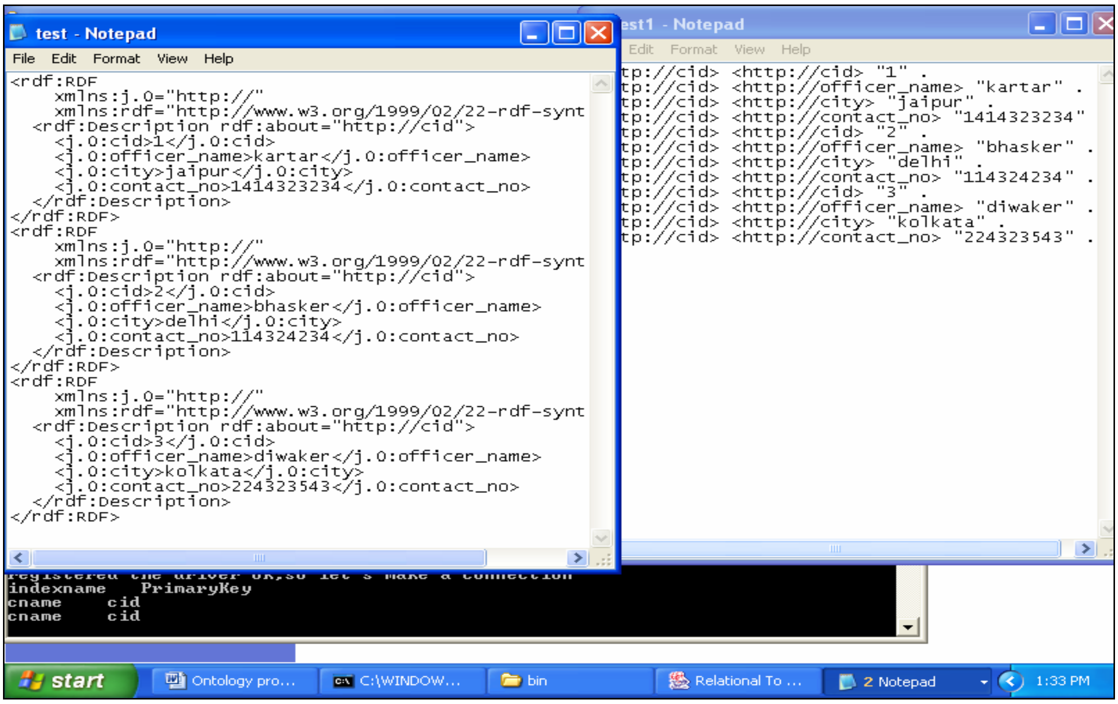


Fig.5.5. RDB2RDF Converter output

5.4 USES OF RDB2RDF CONVERTER

The main uses of RDF converter are following:

a. RDF code generation which will be used for semantic web application and provide a bridge between semantic web and current web using RDF and relational data.
b. Reusability in design means we can convert any relational data without writing equivalent RDF code for the semantic web.
c. Provide the help to semantic web.
d. Combine the data with other data sets that don't follow the data model you've been using.
e. Add more data that doesn't fit the table structures. You could add a book's web site to the data; give an author's nickname, or whatever you like.
f. Use an RDF processor that can do logical reasoning to discover unstated relationships between the properties and resources in your data.
g. Add statements about publications and references that have been defined somewhere else on the Web. All you have to do is to refer to the identifiers (the URIs) that they have published. You aren't limited to talking about things stored in your own database.
h. Do all these things using well-defined standards, so that a wide range of software can process the data.

CHAPTER VI

IMPLEMENTATION DETAIL

6.1 ONTOLOGY DESIGN WITH Protégé

The implementation of the project ontology and relational to RDF Converter will be used for the development of semantic web using the RDF approach for data search and organization of knowledgebase

To implement the ontology project for tourism domain the following steps are involved:

a. Select a domain

The domain selection plays the great role for the implementation of any project in semantic web and ontology. Here we have selected tourism as one of the domain to represent the concept of ontology

b. Domain expert a advice

To design any ontology the best way is to concern with the domain expert so that we can talk with each and every aspect of a particular domain and based on that define a framework for design.

c. Ontology approach

Select a ontology approach whether the writing code in RDF or using an editor or tool for better design within the cost.

d. Define knowledge

After the selection of ontology approach the nest step is to define the knowledge for a domain using class, relation, slots and properties for a domain and enter the information.

e. Perform the query

The next step is to use the ontology knowledge by using the query tab which will show that how we will do the search with the help of triple that is class, slot and variable for a particular query.

6.2 RDB2RDF CONVERTER

To implement the relational to RDF Converter we will use basically following series of steps:

a. **Select a relational data**

   The first step is to select a relational data whether in SQL or in MS Access data format which we have to convert in to RDF Format.

b. **Select API**

   The next step is to select an API that can be used as a medium between operating system, relational databases and RDF format, for this purpose we have selected the two APIs one is JDBC: ODBC API to help relational data and JENA API for the RDF data.

c. **Access relational data**

   The next step is to access relational data in such a way that we will access first a Tuple through column by column and traverse the whole relational database.

d. **JENA API**

   The next step is to understand the JENA API and its functionality that it will provide to the RDF and how to use the different packages from the Jana to fulfill your desired goal. Jena is a platform that will provide the various functionality for RDF.

e. **Mapping in to RDF**

   The next step is to map a relational data into RDF format using the model and various classes that will support RDF format. Using it we have to perform mapping for the RDF.

f. **RDF and RDF Triple**

   Finally we will store the converted relational data in to two different text file name test and test1 for the storage of data in the RDF and RDF Triple format.

Running project

The last step is how to run a project. The main steps that are used for ontology are following:

a. **Install protégé:**

   The first step is to install protégé, a tool that will support the ontology. It is available free of cost on Stanford university website for the use and distribution for any application

b. **Open a project:**

The next step is to open the designed project with the help of protégé to perform various operations and update it.

c. **Perform query:**

The next step is to perform query for the ontology search and save it for the further use.

d. **RDF code generation**

Finally we can get the ontology in the RDF format so that it can be used in the form of RDF data for semantic web design.

The main steps involved in running RDF converter are:

a. **Install jdk**

The first step is to install the jdk for the java and it should be jdk 1.3 or above version for using Jana packages.

b. **Install Jena**

The next step is to install the Jena for the purpose of the RDF manipulation.

c. **RDF Code generation**

In this step we will convert a relation data into RDF code using the Jena API and its various packages defined for the operation.

d. **Storing RDF Code**

Finally we will store the RDF code into text file for the further use for the semantic web applications.

6.3 SNAPSHOTS

6.3.1 Snapshots for Tourism Ontology

Below figure 6.1 – Figure 6.5 present the working snapshots of the tourism ontology.

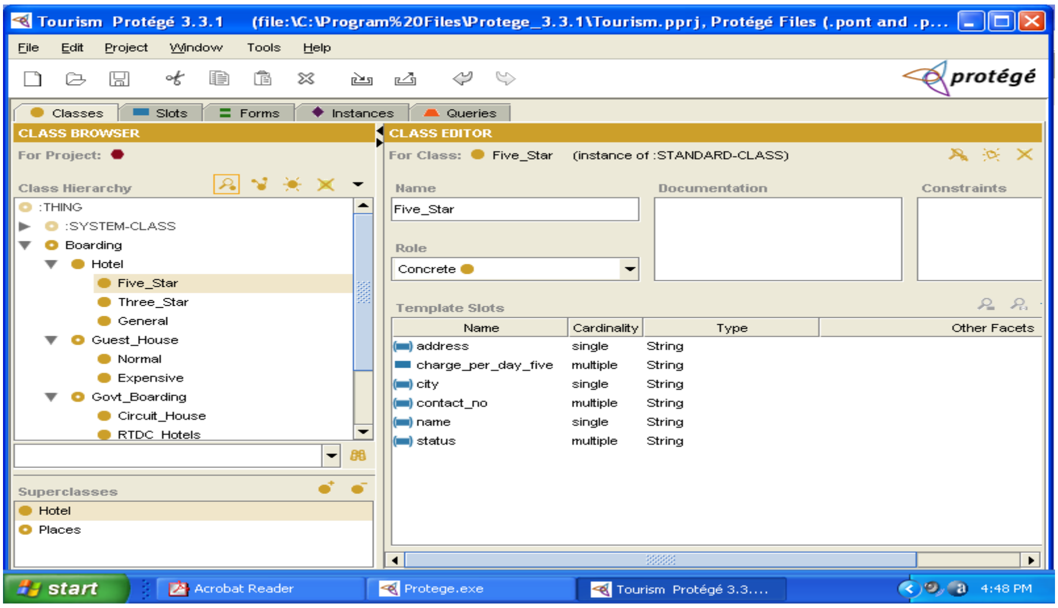


Figure 6.1 Class- Subclass hierarchies

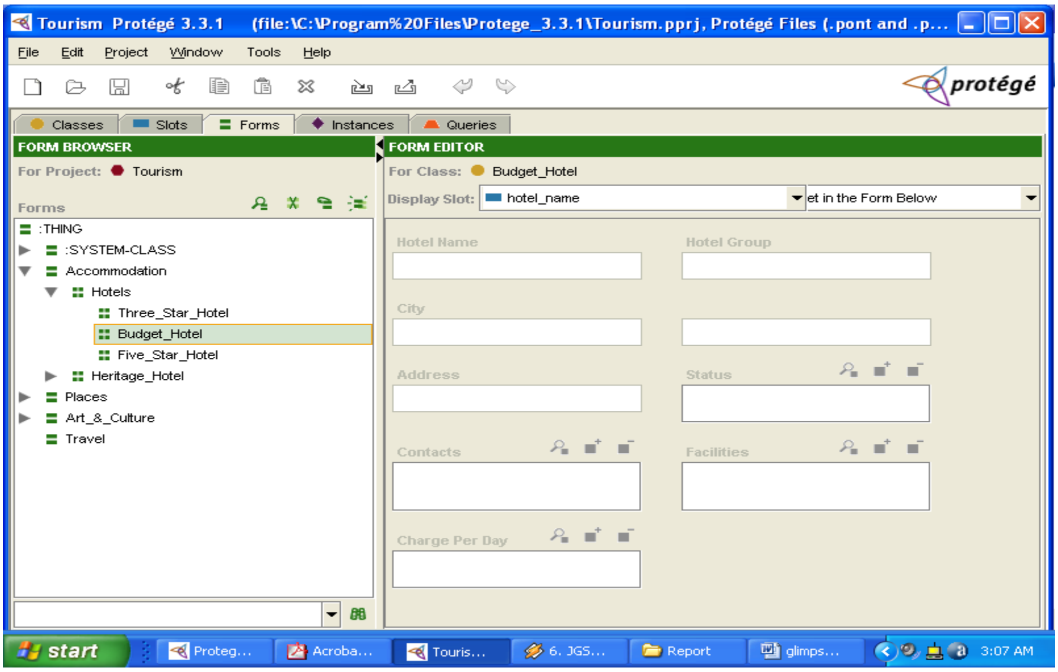


Figure 6.2 Form design in Ontology

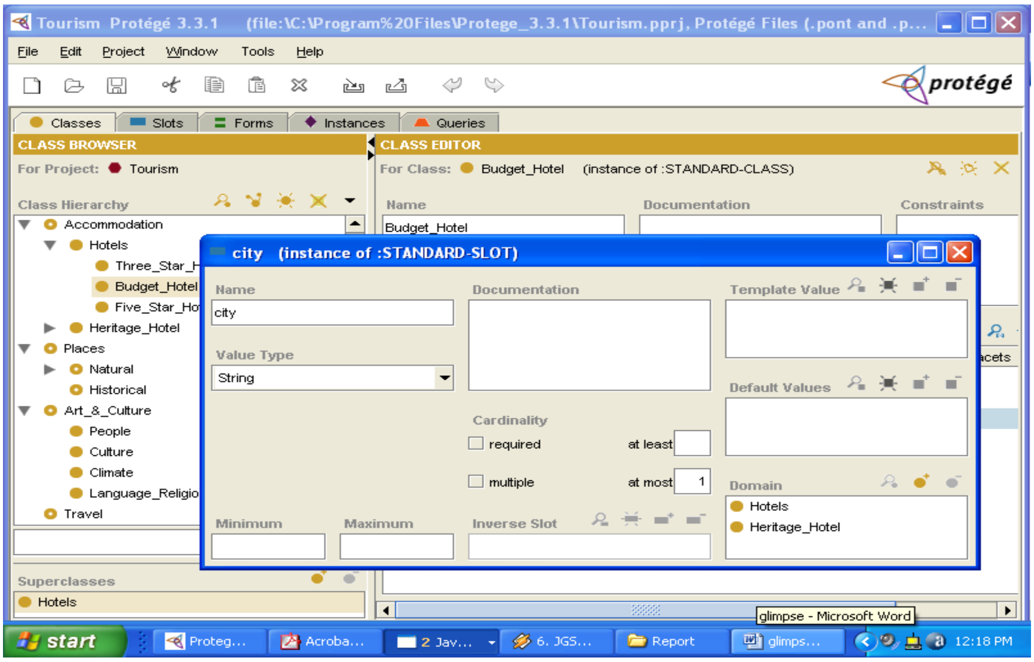


Figure 6.3 Defining Constraints

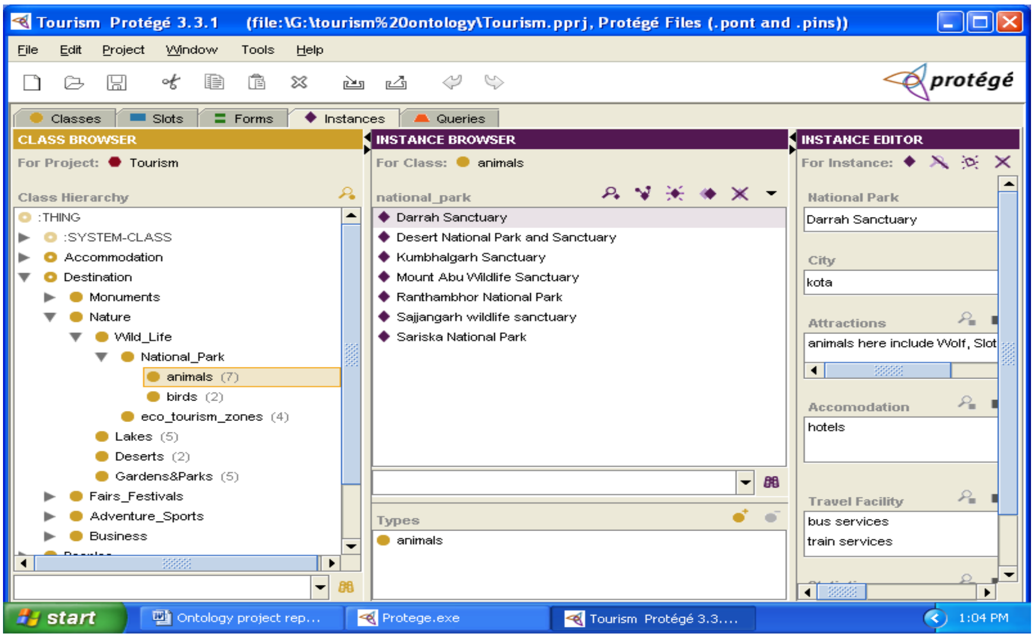


Figure 6.4 Instances in Ontology

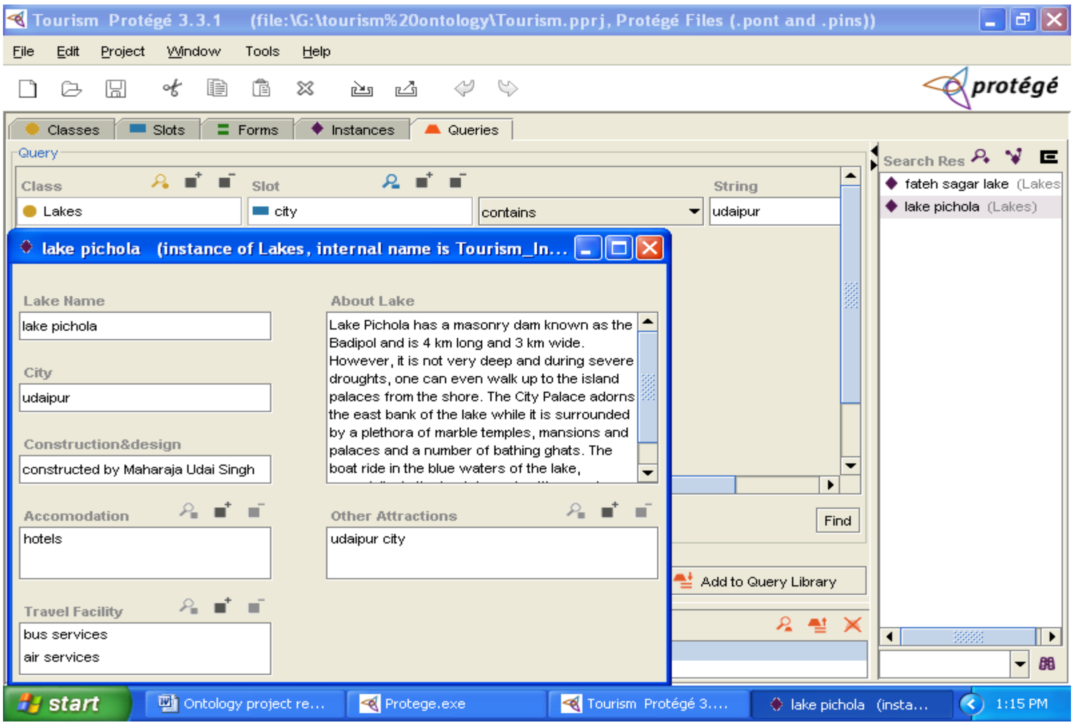


Figure 6.5 Searching in Tourism ontology

6.3.2 Snapshots for RDF converter

The below shown snapshot show the running of Relational to RDF Converter

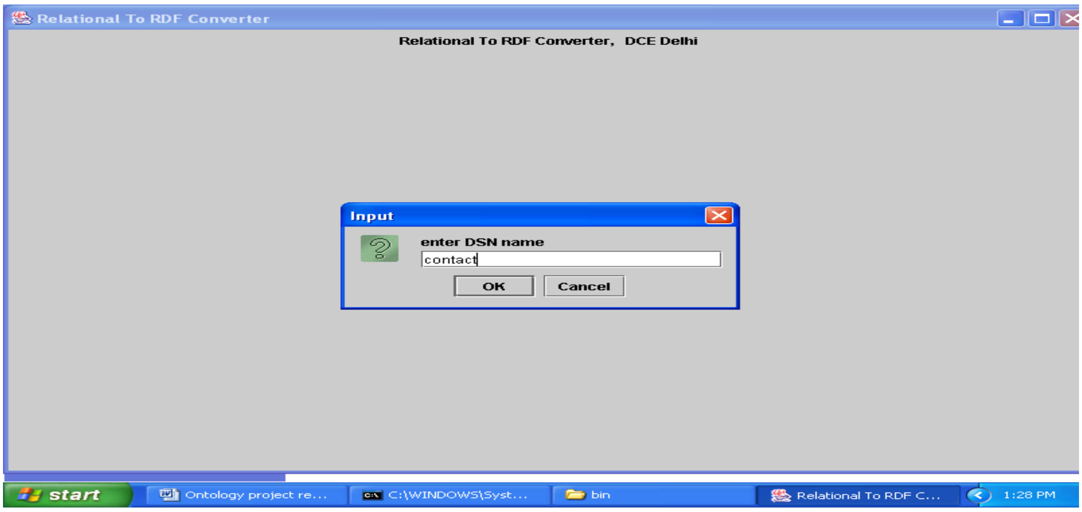


Figure 6.6 Inputs for RDF Converter

The below figure the result of relational to RDF converter

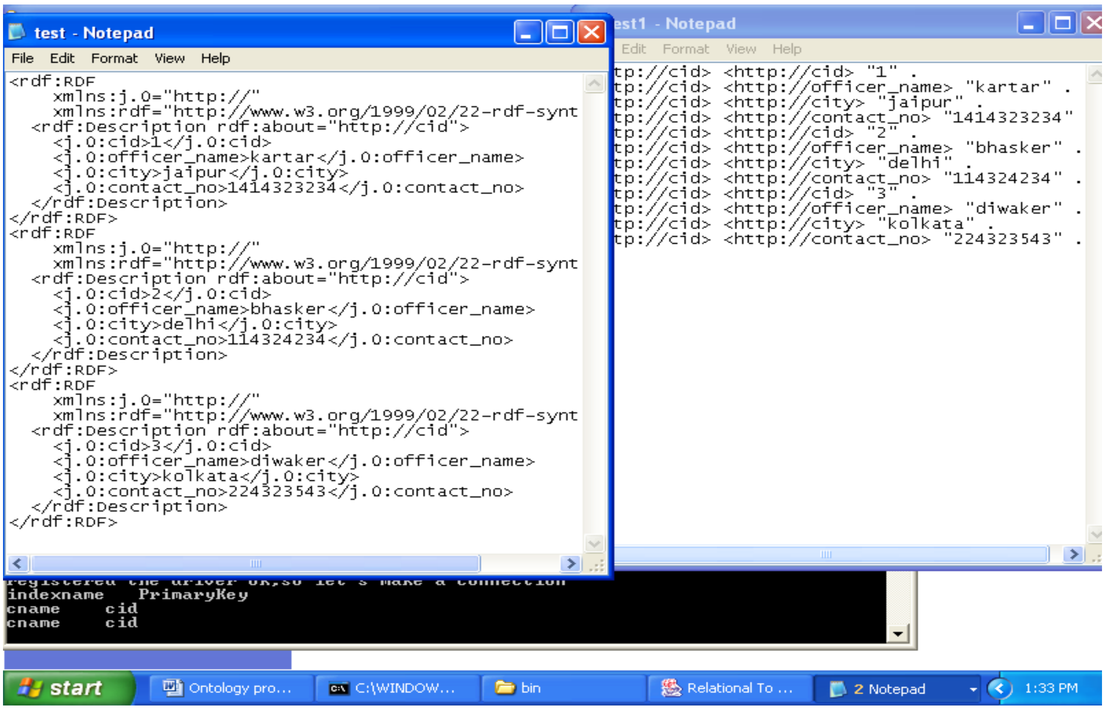


Figure 6.7 Outputs for RDF Converter

6.4 RESULT AND DISCUSSION

6.4.1 RDF Converter

The simple table in the MS Access is taken as input as shown below:

cid	Officer_name	City	contactno
1	Kartar	Jaipur	1414323234
2	bhasker	Delhi	114324234
3	diwakar	kolkata	224323543

Figure 6.8 Contact table

The relational table will be converted into the RDF We connect the database to Microsoft driver. after that using jdbc connectivity we connect the dsn to the java program. it has many methods by which we connect the dsn to a java program. after that use jena api. In jena api there are many methods by which we make a rdf memory model. it has methods for create resource, properties and connect the properties to the object. The output of the sample input is

RDF Code

```
<rdf:RDF
    xmlns:j.0="http://"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="http://cid">
    <j.0:cid>1</j.0:cid>
    <j.0:officer_name>kartar</j.0:officer_name>
    <j.0:city>jaipur</j.0:city>
    <j.0:contact_no>1414323234</j.0:contact_no>
  </rdf:Description>
</rdf:RDF>

<rdf:RDF
    xmlns:j.0="http://"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="http://cid">
    <j.0:cid>2</j.0:cid>
    <j.0:officer_name>bhasker</j.0:officer_name>
    <j.0:city>delhi</j.0:city>
    <j.0:contact_no>114324234</j.0:contact_no>
  </rdf:Description>
</rdf:RDF>

<rdf:RDF
    xmlns:j.0="http://"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <rdf:Description rdf:about="http://cid">
    <j.0:cid>3</j.0:cid>
    <j.0:officer_name>diwaker</j.0:officer_name>
```

```
    <j.0:city>kolkata</j.0:city>
    <j.0:contact_no>224323543</j.0:contact_no>
  </rdf:Description>
</rdf:RDF>
```

RDF Triple:

```
<http://cid> <http://cid> "1" .
<http://cid> <http://officer_name> "kartar" .
<http://cid> <http://city> "jaipur" .
<http://cid> <http://contact_no> "1414323234" .
<http://cid> <http://cid> "2" .
<http://cid> <http://officer_name> "bhasker”.
<http://cid> <http://city> "delhi" .
<http://cid> <http://contact_no> "114324234" .
<http://cid> <http://cid> "3" .
<http://cid> <http://officer_name> "diwaker" .
<http://cid> <http://city> "kolkata" .
<http://cid> <http://contact_no> "224323543".
```

These are the RDF Codes which can be used in searching and performing knowledge management. These RDF Codes are used for the development of semantic web.

The RDF Converter will fulfill the desired goal of the project which is to convert the relational data into RDF code.

6.4.2 Ontology

The ontology was designed with motive of two goals in mind:

- Performing query in particular domain, in which we have designed the knowledgebase which can perform the searching operation using the class, relation and value.
  We have achieved this goal up to some extent for the design of Full ontology we may need a domain expert for the design.

- Generating RDF code, this can be used further in the searching application and other knowledge representation.
  A sample output for the query in RDF about any instance is as shown below figure 16 which will return the efficient information regarding the instance lake pichola as shown below.

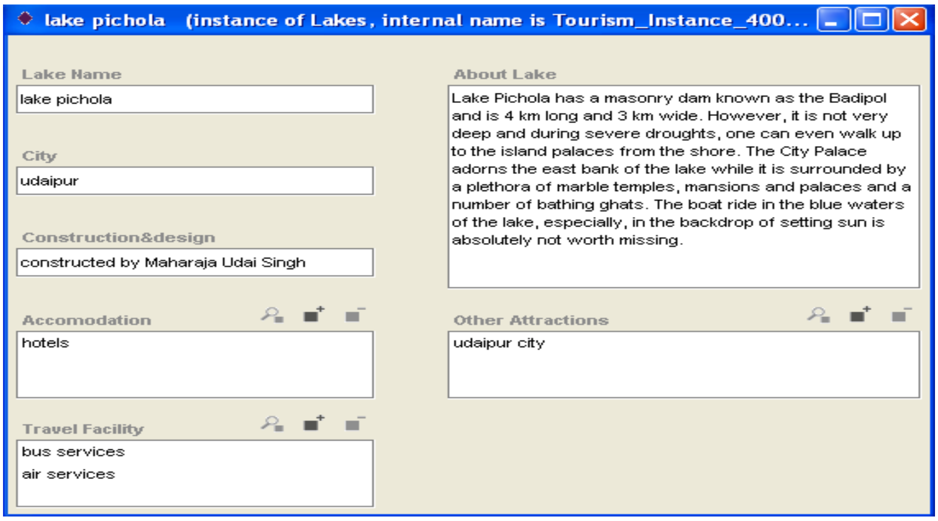


Figure 6.9 Search result in ontology

CHAPTER VII

PUBLICATION FROM THESIS

During the period of working over this project we interacted with International community working on web technologies. We discussed our approach for representing knowledge with them and collected the reviews and worked over the suggestion send to us. One Research papers have been accepted in International conference for presentation and will be published in their proceedings. Also we have communicated a journal paper, so that our work can be recognized and validated.

This paper presents the concept of knowledge representation with Ontology and RDB2RDF Converter. It also specifies a well articulated ontology design process model for the knowledge representation in semantic web.

7.1 The details of Conference publications are as follows:

Conference Name: *International Conference on Enterprise Information Systems and Web Technologies (EISWT-08), USA.*

URL : **http://www.promoteresearch.org/2008/eiswt/index.html**

Paper Title : *"knowledge management with ontology and RDB2RDF Converter"*

Authors : Dr. Daya Gupta and Kartar Jat

Location : Imperial swab hotelOrlando, Florida, USA.

Publishers/ proceedings: The accepted papers will be included in the conference proceedings, which has an ISBN number. The proceedings will be made available during the conference. The proceedings will also be submitted for several database indexes. The previous conferences are submitted for several reputed database indexes. Revised versions of some selected papers will also be considered for publication with several journals as well.

Paper will be indexed at **DBLP** Bibliography Server.

7.2 The details of Journal publications are as follows:

Journal Name: International Journal of Knowledge Management (IJKM)

About Journal: The International Journal of Knowledge Management (IJKM) covers all aspects of the knowledge management discipline, from organizational issues to technology support to knowledge representation. The IJKM provides a forum for global aspects of knowledge management and for differing cultural perspectives on the use of knowledge and knowledge management. It also focuses on cultural and organizational issues as well as technical issues associated with the creation and implementation of knowledge management systems. The International Journal of Knowledge Management actively seeks out submissions from all regions of the world.

URL: http://www.igi-pub.com/journals/details.asp

Journal Topic: Methodology: Knowledge based semantic search

Author: Dr. Daya Gupta, Kartar jat, Anuj Kumar, Nitin Nizhawan

Editor-in-Chief: Murray E.Jennex San Diego State University, USA

Status: Journal paper is under Communication with publisher.

.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This Thesis presents the methodology of knowledge representation using ontology and RDB2RDF converter. We described our knowledge representation architecture in semantic web with ontology and RDB2RDF converter. It presents how to construct ontology for a particular domain using ontology design process. Author created ontology for tourism. The ontology developed provides the information in better way according to the area of interest. At the same time RDB2RDF converter is used to present knowledge in RDF format. A model for the knowledge representation is presented that will be used for the knowledge representation using new concept and existing RDB for the knowledge representation. We demonstrated the ontology development for tourism and a conceptual tool for the RDB2RDF conversion with illustrative example.

The project will fulfill the all goals regarding the initial goal in mind during the design. While there are some sort coming in the design as there is not expert for the domain of ontology. The main objective RDF Converter and ontology is covered and working with the desired output of need. So over all motive of the project is covered in the design process. That RDF code will be used to implement to solve many Net based problems.

The main goal that is fulfilled is:

a. Tourism ontology design that is able to solve many real life problems related to the semantic web and internet.
b. RDF Converter is developed that can be used for the data conversion which will becomes one of the great tool in the semantic web for the peoples.

The main work that was done in the ontology domain was:

a. Ontology for medical, books and some of the domain are already available
   The current ontology uses the different domain so provide help to the semantic web developer at the same time provide one of the alternate for the tourism search related to Rajasthan tourism.
b. In the current scenario there is not any tool that can convert the relational data to RDF format so this project work will provide a new tool and work as a bridge between the semantic web and syntactic web.

These are the main topics that will present the work done in the project.

Ontology design for tourism domain and RDF converter are designed with main motive of to present concept of ontology in semantic web and designing a tool that will convert the relational data into RDF format for knowledge representation which can be used in different domains such as search, web portal and data representation in small devices such as mobile phone.

7.2 FUTURE WORK

Here we have presented technique for the knowledge representation for the various AI Applications. Knowledge can be used in various applications such as search, web portal and mobile data representations. The knowledge we gather from the ontology and RDB2RDF Converter can be used by the semantic web for designing various application based on this domain.

For any domain we also need to present the current relational data into knowledge base which will be done with RDB2RDF Converter so that all data are present in RDF/XML form which can be used to perform query and other logic using SPARQL [16] or SERQL for the semantic web applications.

Knowledge representation model present the uses of knowledge in various domain and areas according to the need of researchers. So knowledge representation need more work to be done in the area so that we can uses the benefit of knowledge in place of data.

CHAPTER VII

REFERENCES

[1]. D. Fensel, “Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce”, Springer-Verlag, Berlin, 2001.

[2]. A. Banerji, C. Bartolini, D. Beringer, V. Chopella, K.Govindarajan, A. Karp, H. Kuno, M. Lemon, G. Pogossiants, S. Sharma, S. Williams: Web Services Conversation Language (WSCL), HP, 2001.

[3]. Karun Bakshi and David R. Karger; “Semantic Web Applications”, MIT Computer Science and Artificial Intelligence Laboratory

[4]. Dan Song, Wudong Liu, Yangfan He, Keqing He; Ontology Application in Software Component Registry to Achieve Semantic Interoperability; Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’05), IEEE.

[5]. Graham Klyne, Nine by Nine; Semantic Web Applications; http://www.ninebynine.net/

[6]. Ackoff, R. L., "From Data to Wisdom", Journal of Applies Systems Analysis, Volume 16, 1989 p 3-9.

[7] Xu Binfeng, Luo Xiaogang, Peng Chenglin and Huang Qian, “Based on Ontology: Construction and application of Medical Knowledge Base”, 2007 IEEE/ICME Inrenational conference on complex medical engineering

[8]. Mariano Fernindez Lopez, Asuncion Gomez-Perez, and Juan Pazos Sierra, Building a Chemical Ontology Using MeMmontology and the Ontology Design Environment; IEEE,1999.

[9]. U.jan, magda, pavel, tereza; creation of architectural ontology: user’s expericene; 14th International Workshop on Database and Expert Systems Applications (DEXA’03).

[10] Wajee Teswanich and Suphamit Chittayasothorn, “A Transformation from RDF Documents and Schemas to Relational Databases”, published at IEEE 2007.

[11]. Seung Ki Moon, Soundar R.T. Kumara, and Timothy W. Simpson, “Knowledge Representation for Product Design Using Techspecs Concept Ontology”.

[12]. Martin Hepp, “Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies”, 2007, IEEE Computer society.

[13] Matthew Horridge1, Holger Knublauch2, Alan Rector1, Robert Stevens1, Chris Wroe, A Practical Guide To Building OWL ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0, the University Of Manchester, August 27, 2004.

[14]. D. Fensel, "Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce", Springer-Verlag, Berlin, 2001.

[15].Staab, S., Studer, R. editors: "Handbook on Ontologies", Springer, 2004

[16]. Grigoris Antoniou and Frank van Harmelen, "A semantic web primer", The MIT Press Cambridge, Massachusetts London, England, 2004

[17] Matthew Horridge1, Holger Knublauch2, Alan Rector1, Robert Stevens1, Chris Wroe, A Practical Guide To Building OWL ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0, the University Of Manchester, August 27, 2004.

[18]. Taboada, D.Martinez, J.Mira; Experiences in reusing knowledge sources using Protege and PROMPT; Int.J.Human-Computer Studies 62(2005) 597-618.

[19]. D. Brickley and R.V. Guha, Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation, March 27, 2000.

[20]. E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF, W3C Working Draft, October 4, 2006.

[21] IBM Web site www.ibm.com

[22] HP Labs website www.hplabs.com

[23] Protégé http://protege.stanford.edu.

[24] "Protégé Web Interface",
http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeWebBrowser

[25] Jena API http://jena.sourceforge.net/

APENDIX A

An Introduction to Protégé

URL: http://protege.stanford.edu.//

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with Ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

Ontology describes the concepts and relationships that are important in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontologies range from taxonomies and classifications, database schemas, to fully animalized theories. In recent years, ontologies have been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge. Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services.

Protégé is the latest tool in an established line of tools developed at Stanford University for knowledge acquisition. Protégé has thousands of users all over the world who use the system for projects ranging from modeling cancer-protocol guidelines to modeling nuclear-power stations. Protégé is freely available for download under the Mozilla open-source license.

Protégé provides a graphical and interactive ontology-design and knowledge-base–development environment. It helps knowledge engineers and domain experts to perform knowledge-management tasks. Ontology developers can access relevant information quickly whenever they need it, and can use direct manipulation to navigate and manage ontology. Tree controls allow quick and simple navigation through a class hierarchy. Protégé uses forms as the interface for filling in slot values. The **knowledge model** of Protégé-2000 includes support for classes and the class hierarchy with multiple inheritance; template and own slots; specification of pre-defined and arbitrary facets for slots, which include allowed values, cardinality restrictions, default values, and inverse slots, meta classes and meta class hierarchy. In addition to highly usable interface, two other important features distinguish Protégé from most ontology-editing environments: its **scalability** and **extensibility**.

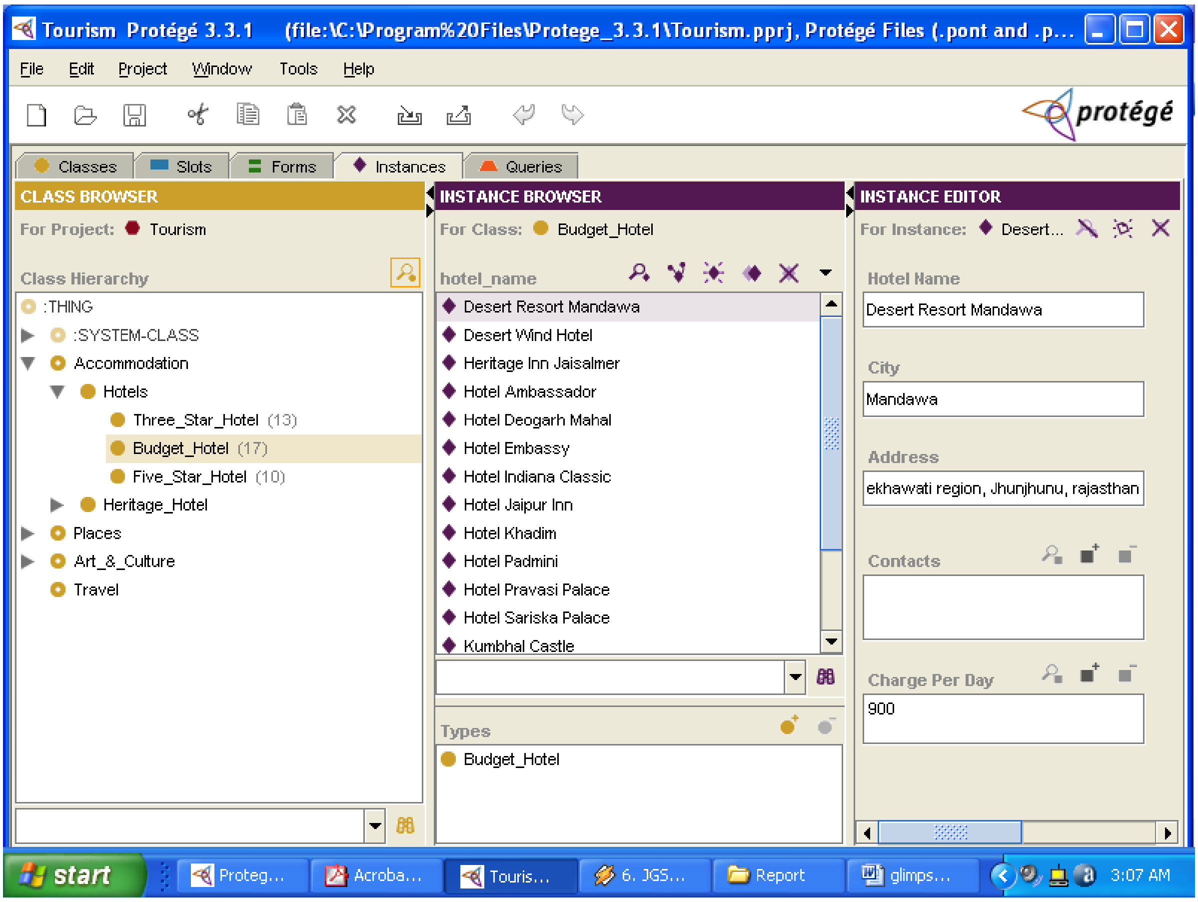


Figure (i) Protégé

One of the major advantages of the Protégé architecture is that the system is constructed in an open, modular fashion. Its component-based architecture enables system builders to add new functionality by creating appropriate plugins. The Protégé Plugin Library3 contains contributions from developers all over the world.

The Protégé-OWL API is an open-source Java library for the Web Ontology Language (OWL) and RDF(S). The API provides classes and methods to load and save OWL files, to query and manipulate OWL data models, and to perform reasoning based on Description Logic engines. Furthermore, the API is optimized for the implementation of graphical user interfaces. The Protégé-OWL API is centred around a collection of Java interfaces from the model package. These interfaces provide access to the OWL model and its elements like classes, properties, and individuals.

The API is designed to be used in two contexts:

- For the development of components that are executed inside of the Protégé-OWL editor's user interface
- For the development of stand-alone applications (e.g., Swing applications, Servlets, or Eclipse plug-ins)

Protégé is a flexible, configurable platform for the development of arbitrary model-driven applications and components. Protégé has an open architecture that allows programmers to integrate plug-ins, which can appear as separate tabs, specific user interface components (widgets), or perform any other task on the current model. The Protégé-OWL editor provides many editing and browsing facilities for OWL models, and therefore can serve as an attractive starting point for rapid application development. Developers can initially wrap their components into a Protégé tab widget and later extract them to distribute them as part of a stand-alone application.

Protégé OWL:

The Protégé-OWL editor is an extension protégé that supports the web ontology languages (OWL). OWL is the most recent development in standard ontology languages, endorsed by the world wide consortium to promote the semantic web. "OWL ontology may include descriptions of classes, properties and their instances. Given such ontology, the OWL formal semantics

specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.
The Protégé-OWL editor enables users to:

- Load and save OWL and RDF Ontologies.
- Edit and visualize classes, properties, and SWRL rules.
- Define logical class characteristics as OWL expressions.
- Execute reasoned such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup.

Protégé-Owl’s flexible architecture makes it easy to configure and extend the tool. Protégé-OWL tightly integrated with Jana and has an open source Jena API for the development of custom-tailored user interface components or arbitrary Semantic Web services.

APENDIX B

An Introduction to Jena API

URL: http://jena.sourceforge.net/

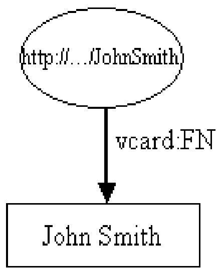
Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Jena is open source and grown out of work with the HP Labs Semantic Web Programme.

The Jena Framework includes:

- A RDF API
- Reading and writing RDF in RDF/XML, N3 and N-Triples
- An OWL API
- In-memory and persistent storage
- SPARQL query engine

Support is provided by the jena-dev mailing list.

The Resource Description Framework (RDF) is a standard (technically a W3C Recommendation) for describing resources. What is a resource? That is rather a deep question and the precise definition is still the subject of debate. For our purposes we can think of it as anything we can identify. You are a resource, as is your home page, this tutorial, the number one and the great white whale in Moby Dick. Our examples in this tutorial will be about people. They use an RDF representation of VCARDS. RDF is best thought of in the form of node and arc diagrams. A simple vcard might look like this in RDF:



The *resource*, John Smith, is shown as an ellipse and is identified by a Uniform Resource Identifier (URI)[1], in this case "http://.../JohnSmith". If you try to access that resource using your browser, you are unlikely to be successful; April the first jokes not withstanding, you would be rather surprised if your browser were able to deliver John Smith to your desk top. If you are unfamiliar with URI's, think of them simply as rather strange looking names.

Resources have *properties*. In these examples we are interested in the sort of properties that would appear on John Smith's business card. Figure 1 shows only one property, John Smith's full name. A property is represented by an arc, labeled with the name of a property. The name of a property is also a URI, but as URI's are rather long and cumbersome, the diagram shows it in XML qname form. The part before the ':' is called a namespace prefix and represents a namespace. The part after the ':' is called a local name and represents a name in that namespace. Properties are usually represented in this name form when written as RDF XML and it is a convenient shorthand for representing them in diagrams and in text. Strictly, however, properties are identified by a URI. The nsprefix:localname form is a shorthand for the URI of the namespace concatenated with the local name. There is no requirement that the URI of a property resolve to anything when accessed by a browser. Each property has a value. In this case the value is a *literal*, which for now we can think of as strings of characters[2]. Literals are shown in rectangles. Jena is a Java API which can be used to create and manipulate RDF graphs like this one. Jena has object classes to represent graphs, resources, properties and literals. The interfaces representing resources, properties and literals are called Resource, Property and Literal respectively. In Jena, a graph is called a model and is represented by the Model interface.

The code to create this graph, or model, is simple:

```
// some definitions
static String personURI    = "http://somewhere/JohnSmith";
static String fullName     = "John Smith";
// create an empty Model
Model model = ModelFactory.createDefaultModel();
// create the resource
Resource johnSmith = model.createResource(personURI);
// add the property
 johnSmith.addProperty(VCARD.FN, fullName);
```

It begins with some constant definitions and then creates an empty Model or model, using the ModelFactory method createDefaultModel () to create a memory-based model. Jena contains other implementations of the Model interface, e.g one which uses a relational database: these types of Model are also available from ModelFactory.