Generating Domain Specific Ontology for Retrieving Hidden Web Contents

***Abstract* --A Large amount of information on the present Web is available only through search interfaces; the users have to type in a set of keywords in a search form in order to access the pages from certain websites. These pages are often referred to as Hidden Web. The components like hidden web crawler which wants to retrieve this hidden information has to fill the search interfaces automatically. For automatically filling up these type of search interfaces precise and relevant data is needed. A database that stores semantic information about objects and their relations may solve this purpose. This database can be defined with the help of ontology which defines a common vocabulary. The work presented here mainly focuses on generating the domain specific Ontology for retrieving hidden web contents. In this paper a knowledge base of Book domain has been created using Protégé. All the information is stored as knowledge base in form of RDF triples<Subject Predicate Object>. This knowledge base may be used in automatically filling up the search interfaces for retrieving hidden web Data.**

Keywords: hidden web, book, domain, ontology, knowledge base.

I. INTRODUCTION

WWW comprises two parts: Surface Web and Hidden Web. The Surface Web consists of billions of browsable pages, while the Hidden Web contains hundreds of thousands of data sources. The Surface of the Web refers to the part of the Web that can be crawled and indexed by general purpose search engines, while the hidden Web refers to the abundant information that is "hidden" behind the query interfaces and not directly accessible to the search engines. Hidden Web sources store their content in searchable databases that only produce results dynamically in response to a direct request. There is no keyword matching scheme and no url following for accessing hidden web data.

To solve the problem of retrieving hidden web content there is a need of addressing three main issues [2]:

1. Generating Ontology for a particular domain.
2. Creating Common Interface for Users.
3. Mapping of Interface.
4. Generation of Queries.

This paper concentrates on first step of developing an ontology for book domain. Ontology technology allows arbitrary user-defined relationships among classes and allows adding properties to relationships such as symmetry, transitivity, and inversion. These properties are used in reasoning that's why Ontology supports inference process in knowledge base.

Digital library is one of the domain that comes under hidden web. Analysis shows 60 to 70 percent of hidden web information buries under these websites [3][4]. So it will be useful for users if one can create common search interface through which all available books can be accessed, leading to generation of Book Domain Ontology. The Book Domain that is chosen for generating Ontology is very general and known to all. In day to day life one has to access WWW for gathering information about various research work, research papers or to find study material. Ontology is preferred instead of relational database as it avoid data redundancy, allow sharing of information, reuse of information, gives common understanding of domain structure and easy reasoning in terms of <S P O> triples that are not possible with relational database.

II. RELATED WORK

Recent studies have estimated the size of this hidden web at around 500 times the size of PIW. As the volume of hidden information grows, there has been increased interest in techniques that allow users and applications to leverage this information. To address the problem of crawling and retrieving the contents from hidden web many work has to be done in this area

A. *Hidden Web Exposer (HiWE) :* Raghavan and Garcia-Molina's work was to learn Query Interfaces. Here they proposed HiWE, a task-specific hidden-Web crawler [6].

B. *Framework for Downloading Hidden Web Content:* Ntoulas et al. provided a theoretical framework for analyzing the process of generating queries, The crawler has to come up with meaningful queries to issue to the query interface[4]. This body of work was often referred to as query selection problem over the Hidden Web. Disadvantage of this work was that it focuses on single attribute queries rather than multi attribute or structural queries.

C. *Exploiting Ontology for Retrieving Data Behind Searchable Web Forms:Okasha et al.*[] Classify each searchable form to its relevant domain then exploit the suitable ontology to automatically fill out these forms. Then retrieve and analyze the result pages for further indexing process in the search

III. PROBLEM IDENTIFICATION

After having a critical look over the work done in this field and considering the limitation of each, It is observed that Hidden Web services which are mostly form-based interfaces need to be described using new approach which involve semantic specifications. The parameters or the service behaviours as well as non-functional properties need to be extracted or inferred. It is further noticed that various researchers have given algorithms to access deep web using ontology but not much work has been done on constructing that ontology.

IV. Proposed Work

As WWW is growing tremendously and people are relying on it for finding study materials including Academics and non Academics books. Also for doing research a person downloads the material from Internet. For Example if a user wants to access an ebook he/she has to specify book's name along with other information like author's name, ISSN number etc. or if a user wants to access some research work he has to specify the paper name or journal name, issue number, conference name etc. After specifying all these field and after inputing values for all the fields user will be brought to various links which further contain the URL's for various websites which contain the relevant data. Hence there is a huge requirement for developing a common framework (interface) where a user can enter a single query and get the desired result.

For generating and responding to user query in hidden web environment we need a database that is populated knowledge base which satisfies the user's need of information. As user specify queries using different keywords but having same meaning, Ontology is one of the best ways for creating Domain knowledge that has common understanding of the structure of information among people.

Protégé is an open source platform which is used for creating OWL ontology.

Work begins with the specification of hierarchal models that is used to generate ontology for book domain, covering the description of various tools that are used for developing the same and how one can create knowledge base.

A. Book Domain Ontological Model

Book Domain contains the collection of books that are related to Author, Publication, Award and many more. For proposing the idea four sub domain has been chosen to generate book domain ontology that are Book, Author, Award and Publication.

These domain related to each other with some relationship.

e.g. All books has been written by some author & has been published by some publication. Here ***written by*** and ***published by*** is relation that connect these two domains.

Further we classify a particular domain into its sub domain that collectively contain the information to that domain and these sub domains connected with domain with **has a** relation.

a) *Author:* Author is one that originates or writes books of his interest. Two fields of Author may relate to books are Author _Name as Book is written by particular Author and Author_Email_Id that may describe contact point to author.

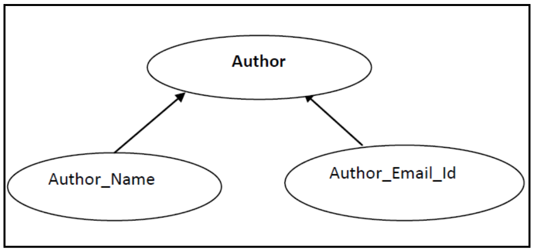


Figure 1. Author Class and its properties.

These sub domain Author_Name is set of all authors name that wrote some books. This domain related to Author with ***has*** Relation.

An author must relate to its name and email-id that uniquely identify its identity.

<Author has Author_Name>

<Author has Author_Email_id>

b) *Award:* Award is materialistic thing that may be given to any book or any author for his book to recognise its excellence in particular field.

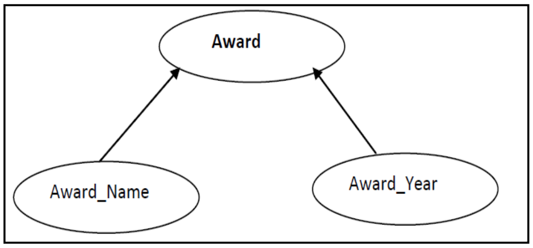


Figure 2. Award Class and its properties.

Here the sub domains Award_Name and Award_ Year related to Award with has a relation.

<Award **has** Award_Name>

<Award **has** Award_year>

c) *Publication:* Publication is an Authority that makes content available to general public by publishing it. The preparation and issuing of a book, journal, piece of music, or other work for public sale.

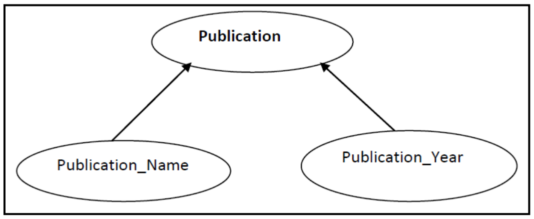


Figure 3. Publication Class and its properties.

Here Publication_Name and Publication_year can be related to Publication domain with similar has relation.
<Publication has Publication_Name>
<Publication has Publication_Year>

d) *Book:* Book may refers to source of information or work of literature that may have Book_Title that represents its unique name, Book_Price that provide information that at what amount it is available to user of interest, Book_category that details about the type of book and Book_ISBN that defines International standard Book number. Book category may contain vast varieties of books so it is required proper organisation of books under book category that is divided into Academics, Sports, Research category, Computer , Internet, story, Novels etc.

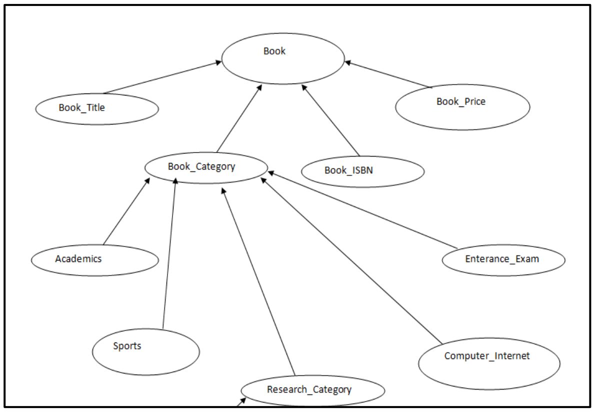


Figure 4. Class Hierarchy.

.

Each book must conatin its name .
<Book has Book_Title>
Each book must belong to some category.
<Book has Book_category>
Each book has unique Book_ISBN .
<Book has Book_ISBN>

e) *Subject-Property-Object relationship* Any relation is in triple<S P O>. Here subject and object may take entity from concepts set and predicate that may link the subject and object. Here Domain is defined as collection of subject and range defines collection on object. Object property maps the Domain and Range as mapped in functional mapping.

In this Book Domain Ontology various sub domains are related to each other with some relation. A domain which is related to other is named as subject and object and they are related with some relation called as object property.e.g. Sub domain of Author is Author_Name related to some book from Book domain with object property ***has Written.***

<Author_Name hasWritten Book_Title>

An **instance** from author name is related to an instance of book title with hasWritten relation. An instance from author name is related to an instance of book title with hasWritten relation.

Similarly we use various objectproperty as describe in table :

Table 1.Table showing <S,P,O> triples

Object Property	Subject (Domain)	Object (Range)
datedOn	Author_Name	Author_Year
hasAmount	Book_Title	Book_Price
hasId	Author_Name	Author_EmailID
hasISBN	Book_Title Book_Ctaegory	Book_ISBN
hasName	Book_Category	Book_Title
hasPublished	Author_Name or Publication_Name	Research_Title or Book_Title
hasWritten	Author_Name	Book_Title
publishedBy	Book_Title Research_Title	Publication_Name
publishedIn	Book_Title Research_Title	Published_Year
wonAward	Book_Title Author_Name	Award_Name
writtenBy	Book_title	Author_Name

A symmetric relation to **hasWritten** is **IsWritten By** can also be defined that can swap subject and object value means a particular book is written by some author.
<Book_Title IsWrittenBY Author_Name>

V. IMPLEMENTATION AND RESULTS

After having all design aspects of proposed model it is being implemented using Protégé 4.2 Desktop ontological editor tool. Protégé is a free, open source ontology editor and knowledge-base framework.

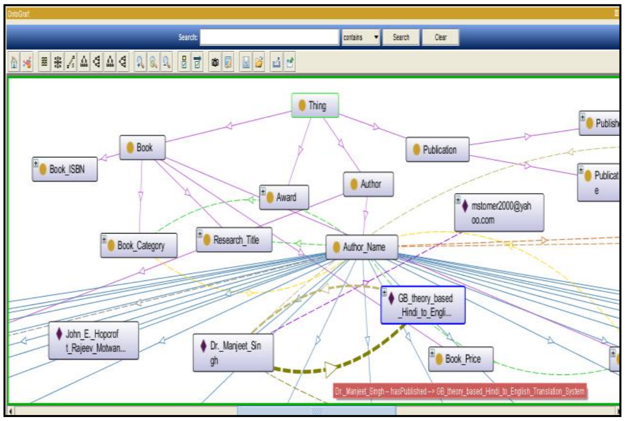


Figure 5. Ontograph

Now to provide support for interactively navigating the relationship of OWL created we use Ontograf utility that support display of superclass-subclass relationship, individual, Domain, Range, Equivalence, Property, Relationship. This provides the familiar view to overall ontology generated.

After populating the entire Book Domain ontology we have knowledge base ready to feed as input for reasoning.

After populating the entire Book Domain ontology we have knowledge base ready to feed as input for reasoning.

Here we have taken Dr.A_K_Sharma as instance showing all its relationship to other concepts. This Knowledge base showing following <S PO> triples.

1. < Dr ._A_K_Sharma **haswritten** Elements_of_Computer_Science >
2. < Dr._A_K_Sharma **hasPublished** Moment_to_Moment_Node_Transition_Awareness_Protocol_ (MOMENTAP) >
3. Dr._A_K_Sharma **hasid** ashokkale2 @ rediffmail .com >

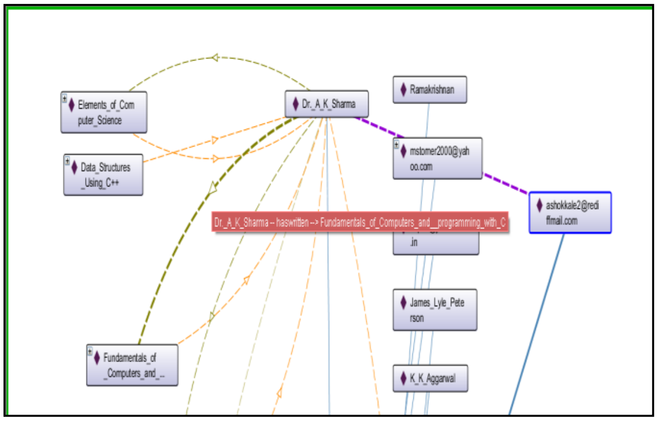


Figure 6. Knowledge Base

Now the system is having complete knowledge base of book domain for which Ontology has been created which is ready to handle user's book domain related query. For available proposed ontology there is the need for designing an ontology-based querying system which maps the information asked by the user on the knowledge stored in the ontology.

A. *SPARQL Query Language*

SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF. SPARQL contains capabilities for querying required and optional graph patterns. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.

VI. USER QUERY AND RESULT

If user wants to the list of all subjects and objects that are related.

Input: "*SELECT ?subject ?object WHERE { ?subject rdfs:subClassOf ?object }*"

e.g. If user wants to the list of all subjects and objects that are related.

Input: "SELECT ?property ?object WHERE { base:Dr_Komal_Kumar_Bhatia ?property ?object}";

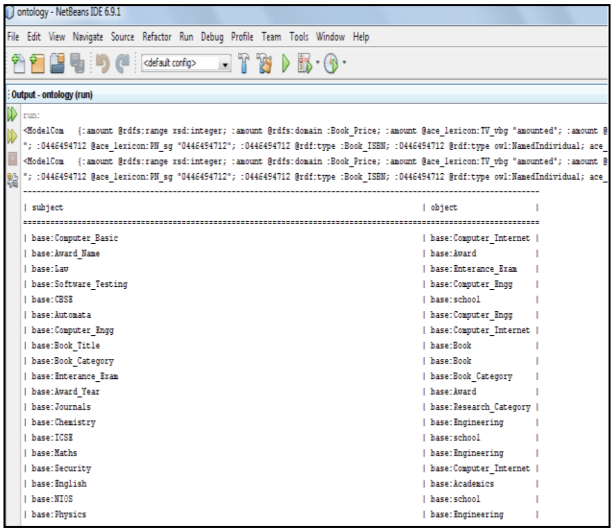
Here Query is in the form < Dr_Komal_Kumar_Bhatia ? ? >

The object and predicate field is unknown , it will result the set of all those property and object that is related to given object instance from knowledge base.

< Dr_Komal_Kumar_Bhatia hasPublished Design_of_Task-Oriented_Hidden_Web_Crawler>

< Dr_Komal_Kumar_Bhatia type Author_Name>

Output's Snapshot:


ontology - NetBeans IDE 6.9.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Output - ontology (run)

```
run:
<ModelCom   {:amount @rdfs:range xsd:integer; :amount @rdfs:domain :Book_Price; :amount @ace_lexicon:TV_vbg "amounted"; :amount @
"; :0446494712 @ace_lexicon:PN_sg "0446494712"; :0446494712 @rdf:type :Book_ISBN; :0446494712 @rdf:type owl:NamedIndividual; ace_
<ModelCom   {:amount @rdfs:range xsd:integer; :amount @rdfs:domain :Book_Price; :amount @ace_lexicon:TV_vbg "amounted"; :amount @
"; :0446494712 @ace_lexicon:PN_sg "0446494712"; :0446494712 @rdf:type :Book_ISBN; :0446494712 @rdf:type owl:NamedIndividual; ace_
------------------------------------------------------------------------------------------------
| subject                                                               | object                 |
================================================================================================
| base:Computer_Basic                                                   | base:Computer_Internet |
| base:Award_Name                                                       | base:Award             |
| base:Law                                                              | base:Enterance_Exam    |
| base:Software_Testing                                                 | base:Computer_Engg     |
| base:CBSE                                                             | base:school            |
| base:Automata                                                         | base:Computer_Engg     |
| base:Computer_Engg                                                    | base:Computer_Internet |
| base:Book_Title                                                       | base:Book              |
| base:Book_Category                                                    | base:Book              |
| base:Enterance_Exam                                                   | base:Book_Category      |
| base:Award_Year                                                       | base:Award             |
| base:Journals                                                         | base:Research_Category |
| base:Chemistry                                                        | base:Engineering       |
| base:ICSE                                                             | base:school            |
| base:Maths                                                            | base:Engineering       |
| base:Security                                                         | base:Computer_Internet |
| base:English                                                          | base:Academics         |
| base:NIOS                                                             | base:school            |
| base:Physics                                                          | base:Engineering       |
```

Figure 7.Output of Query

VII. CONCLUSION

Many systems have been developed to organize domain specific information into relational table and perform user query over it but with the involvement of Ontology one can create semantic knowledge base of domain specific information. The work that has been in this paper mainly focuses on process of building Book Domain Ontology for readers and to show how effective the results of queries can be obtains by storing knowledge base in RDF triplets in terms of Ontology. System is designed which handles reader's book related query where he can find the information of book of his interest. The system handles this by searching a knowledge base Ontology stored as a graph in database.

VIII. Future Scope

Ontology design is a creative process and no two ontology designed by different people would be the same. The applications of the ontology and the designer's understanding and view of the domain will affect ontology design process.

There are some issues which need to be addressed by further research. The first issue which needs to be addressed is to store the RDF triplets obtained from graphical model of knowledge base as semantic database in Oracle 11g as to provide scalability feature to the system so that large amount of domain information can be stored and processed. Secondly there should be scope for merging two ontology that may have common concepts and relationship so that ontology domain can be extended.

REFRENCES

[1] Sergey Brin and Lawrence Page, "The anatomy web search engine", Proc. of 7th International World Wide Web Conference, volume 30, Computer Networks and ISDN Systems, pp 107-117, April 1998.

[2] Manvi et al." Design of an Ontology based Adaptive Crawler for Hidden Web", CSNT,2013

[3] A. K. Sharma, Komal Kumar Bhatia "A Framework for Domain-Specific Interface Mapper (DSIM)", International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.

[4] Alexandro Ntoulas. "Downloading Textual Hidden-Web Content Through Keyword Queries", University of California Los Angeles, Computer Science Department, In Proceedings of the Joint Conference on Digital Libraries (JCDL), 2005, Denver, USA.

[5] B. Chandrasekaran and John R. Josephson ,"What Are Ontologies, and Why Do We Need Them?", , Ohio State University , http://www.w3.org/TR/owl-ref/.

[6] S.Raghavan and H. Garcia-Molina. Crawling the hidden web. In *VLDB*, 2001,Stanford Digital Libraries Technical Report. Retrieved 2008-12-27.

[7] Luciano Barbosa, Juliana Freire. An Adaptive Crawler for Locating Hidden Web Entry Points, IW3C2 2007, May 8–12, 2007, Banff, Alberta,Canada.

[8] M. E. Okasha "Exploiting Ontology for Retrieving Data Behind Searchable Web Forms", ©2009 IEEE, Dept. of Computers and Systems, Mansoura University, Egypt

[9] Ontology Development 101: A Guide to Creating Your First Ontology Natalya F. Noy and Deborah L. McGuinness Stanford University, Stanford, CA.

[10] The Rough Guide to the OWL API: a tutorial Version 3.2.3 for OWL.

[11] Resource Description Framework (RDF). http://www.w3.org/RDF/.

[12] OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/.

[13] protégé, http://protege.stanford.edu/.

[14] A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools.