Semantic Search using Ontology and RDBMS for Cricket

S. M. Patil
Information Technology Department,
BVCOE, Navi Mumbai,
Maharashtra, India

D. M. Jadhav
Information Technology Department,
PIIT, New Panvel,
Maharashtra, India.

ABSTRACT

Ontology is being increasingly used for building the applications for the specific domain. Ontology enables users to capture the semantic of the documents. System performance is improved drastically by domain specific information extraction. To interpret information and perform reasoning, we need to store Ontologies in a way that is correct, consistent, scalable and efficient to retrieve. From the years, relational database technology has ensured the best facilities for updating, storing and manipulating the information of problem domain. RDBMS (Relational Database Management Systems) is the most efficient and reliable Data Structure in terms of storage and retrieval. One of the ways is to store Ontologies in RDBMS. To store OWL documents in RDBMS multiple techniques have been proposed, but they either deal with single ontology or they do not store complete semantics expressed in OWL ontologies. Some of the techniques are not really scalable, as the ontology is dynamic and extensible where as the RDBMS schema is not dynamically extensible. So, we need to store the OWL document in such a way that all the data should be stored and system is able to utilize the advantages of relational database. System defined tables are provided to store OWL ontology. Our approach enables users to reference the ontology data directly from SQL using semantic match operators. This paper focuses on semantic search based on ontology and RDBMS for cricket.

Keywords

Ontology, RDF, OWL, SparQL, RDBMS, Inference, Semantic search, Semantic Indexing, Semantic Web.

1. INTRODUCTION

The amount of data available on World Wide Web has increased tremendously. To search the useful information on such a huge data is the topic in discussion today. Research is going on for designing the tools and techniques which can handle the data semantically. Current web (web 2.0) mostly relies on keyword based search. The performance of the system depends on matching the keyword with the available data. Web2.0 doesn’t interpret the contents in the available data. Such a model misses the actual semantic information in text.

To deal with the issues in the Web2.0, Web3.0 comes in the market by W3C. This is called the next generation web or intelligent web. Semantic web can understand the meaning of the contents. The data in semantic web can be interpreted by machine. To interpret the data by machine Ontologies are used [1] for storing the semantic data. Ontology is core part of the next generation web. Both the information extraction and retrieval processes can benefit from ontology. Ontology gives the semantic to simple text content. First we extract the information and then storing in the form of ontology. After ontology storage next step is to search the content from ontology instances. SPARQL is the one of the most efficient query language for the Semantic Web. But problem with this language is that, it is not made for end users [2]. Therefore, Semantic Web community works on simplifying the process of query formulating for the end-user. In Jan 2008, W3C made the SparQL as recommendation for semantic query language [3]. SparQL provides the form based query interface for semantic search.

Information extraction and reasoning requires Ontologies to be stored in such format which is efficient for retrieval, correct, scalable and consistent. RDBMS is the most well-organized and consistent option in terms of storage and retrieval. One of the options is to store Ontologies in RDBMS. For storing ontology documents in relational databases number of techniques has been proposed. Problem with most of the methods are either they are not completely able to store all the construct of ontology or they are concerned with only single ontology. Some of the techniques are not really scalable, as the ontology is dynamic and extensible where as the RDBMS schema is not dynamically extensible [4]. So, there is a need to preserve the dynamic OWL documents in the Relational structure in such a way that no data or relationship is lost and advantages of RDBMS are also gained. The proposed approach enables users to reference ontological data through the number of semantic SQL operators. Semantic matching operators can be combined with other operations such as join which eases the application development [5] [6].

In this paper, we are designing semantic search using ontology and RDBMS. The system is applied to cricket domain. The system consists of automated information extraction module, ontology mapping module, transformation module for conversion of OWL ontology to RDBMS, inference module, keyword based search module. In the first stage ontology building is done for cricket domain. Ontology is stored in RDBMS using different conversion techniques [4]. PelletDB is used for inference module [7]. Semantic searching is done with the combination of traditional SQL query and semantic query. Main concern in this system is achieving high retrieval performance, scalability, utilization of relational database concept, high recall values and preserving the user-friendliness of system.

After introduction, rest of the paper is organized as follows. Section II provides the necessary background and related work. Section III states the detailed problem statement. Section IV gives overview of proposed system. Section V is about the implementation details of the system.

2. RELATED WORK

Semantic search generally deal with knowledge representation using ontology and querying the data from ontology. Ontology-based information extraction and retrieval system and its application to soccer domain is presented in [2]. In general, it deals with three issues in semantic search, namely, usability, scalability and retrieval performance. It proposes a keyword-based semantic retrieval approach. The performance of the system is improved considerably using domain-specific information extraction, inference and rules. System uses the concept of semantic indexing based on Lucene [2]. Ontological data is stored in the flat files which are not able to provide maturity, performance, robustness, reliability and availability.

With the increase in the semantic web, number of ontologies also grows tremendously. It creates the problem of storing the ontology. Ontologies should be stored carefully so that later retrieval of information is simplified. Number techniques are proposed to store the ontology in RDBMS [8, 9, 10, 11], but they lack in achieving the advantages of RDBMS. Data in RDBMS is closed in nature but OWL document are highly dynamic and semi-structured. So, there is a need to preserve the dynamic OWL documents in the relational structure in such a way that no data or relationship is lost and advantages of RDBMS are also gained.

RDBMS are used for efficient storage of Ontologies to provide the fast operations such as search and retrieval, and to gain the benefits of relational databases management systems such as scalability, security and retrieval performance. On the other hand, there appear more and more OWL files that contain Ontologies. Therefore [10] proposes to store ontology by extracting from OWL file to relational databases.

The important factors that we need to see while looking at other techniques are that the indexing technique should preserve the structure of the ontology; it should be scalable in such a way that the number of tables should be constant while the number of Ontologies being indexed by the indexer grows. The last few factors are that there should be well defined rules for the transformation and there should be no or minimal data loss after transformation. The techniques used to store ontology in RDBMS [8, 9] focused on the mapping / transformation techniques. During mapping we require both the low level schema constructs of ontology and RDBMS to define which ontology component maps to which RDBMS component. As a result of mapping we get set of rules. While during transformation we only have transformation rules and ontology. As a result of transformation we get a database schema.

After doing a comparison between ontology indexing techniques we have concluded that only the technique explained in [10] and Jena [12] preserve the ontology in the optimized and fixed RDBMS schema but, the problem with Jena is that the database layer is totally transparent from the user. Jana automatically manages the database end. The problem with [10] is that they are not completely preserving all the constructs of the ontology although they have a static/optimized structure for all the Ontologies and any new ontology can be stored in the same fixed RDBMS schema. The problem with other techniques is that they create separate set of tables in the database for each new ontology. The problem with this methodology is that as the number of Ontologies or the number of concepts in the ontology grow, so do the number of tables in the database. This makes it difficult for the programmer to make an application for such dynamic schema.

The problem with rest of the techniques [10, 11] is that they create separate set of tables in the database for new ontology. The issue with this methodology is that as the number of Ontologies or the number of concepts in the ontology grow, so do the number of tables in the database. This compromises the scalability of the system. To address these issues we have proposed an Indexing scheme which is consistent, scalable, correctly stores Ontologies and retrieval is quick and efficient.

The reason for storing OWL Ontologies in Relational database is that when Ontologies are stored in the form of tables (Relational Schema) then the old application can easily access, and interact with the data stored in Ontologies without ever needing the knowledge of semantic query languages . The Ontologies stored in RDBMS can also interact with non semantic data stored in other tables [4]. Before storing Ontologies in a database we need to apply some rules which will show how the concepts, properties and all the constructs are stored in the database. The procedure of applying these rules is known as Transformation which is presented in [4].

Our approach enables users to reference ontology data directly from SQL using the semantic match operators, thereby opening up possibilities of combining with other operations such as joins as well as making the ontology-driven applications easy to develop and efficient. In contrast, other approaches use RDBMS only for storage of ontologies and querying of ontology data is typically done via APIs.

3. PROBLEM STATEMENT

From the above discussion, we come to the point that current web search are keyword based. Current web are unable to understand the meaning of the content. Web 3.0 or semantic web or intelligent web is approach towards understanding the meaning of the contents by machine. To interpret the meaning of the text by machine, text is represented in Ontology. Ontologies are efficiently implemented with the help of RDF, RDFS, OWL, etc. For correct, consistent. Scalabe and efficient storage of ontology, RDBMS is best choice. The ontology data is stored in the RDBMS by applying the number of conversion technique. RDBMS provides the system defined tables for storing the OWL ontology. After ontology storage, inference is applied on the stored data in RDBMS. Inference is able to find the new relation from the existing data. PelletDb is used for inference purpose. PelletDb finds the new relation from the existing data and store it back in RDBMS. Finally the searching is done with the help of SQL and SparQL. Number of build-in operators and features of SQL are used for searching the desired information from the RDBMS data. By using Ontology and RDBMS scalability, correctness, consistency and effective retrieval of information is possible. Precision and recall value is also improved considerably.

4. PROPOSED SYSTEM

As shown in fig 1, the system is divided into different module for better management. The important module within the system are crawling, plain text search, Information extraction, Ontology design, Ontology mapping, ontology storage in RDBMS, Inference with PelletDB, semantic search.

5. IMPLEMENTATION DETAILS

System is implemented in Java with the support of Jena framework [12]. Implementations details of the modules are as follows.

5.1 Crawling

This is the first module in our system. It takes the web URL as input and crawl all the pages of the website. After crawling, the web pages are stored on hard disk for further processing. HTML parser is used to remove the unwanted contents from the HTML pages. The parser removes the HTML tag, image tag, link tag, etc from the original files. Only plain text is remaining which is used for information extraction purposes and for developing the simple search engine.

5.2 Plain text search

In this module index is prepared on plain text. Index preparation is done with the help of Apache Lucene. This type of search engines is used traditionally for searching the plain text which depends on matching the content within available data. Apache Lucene is providing the facility for indexing, searching, query highlighting, etc [24]. This search engine is used for comparison purpose only.

5.3 Information extraction

It is one of the important parts of ontology based semantic web applications. It is a process of adding structured information from unstructured resources. In this phases system uses the data crawled from the cricket websites. Natural language processing is used for the information extraction. Input for this module is basic information and narrations of the matches. The details of information extraction from HTML pages are given in [14]. It is a template based approach used for information extraction from HTML pages. Extracted information is stored in the XML format on disk.

5.4 Ontology Design

Central cricket ontology is designed which is used for information extraction and inference. Overall performance of the system is dependant on the quality of ontology design. Iterative development strategy is used for ontology design. First different classes, subclasses, properties, sub properties are identified for the cricket domain. E.g. cricket, wicket, century can be considered as the classes whereas one-day cricket, test cricket, IPL can be subclasses of cricket. Properties for one day cricket are no of days, no of over’s, no of innings, participants, etc. Protégé is used for ontology design [15] which provides the user interface for developing the ontology. Protégé provides the ontology development in RDF/XML, OWL/XML and many other formats. This system uses the OWL/XML for storing the ontology. The fig 2 shows some important classes in cricket ontology with the help of protégé.

5.5 Ontology mapping

It is process of mapping unstructured, structured, semi-structured data into ontology instances. Our information extraction module done lots of work by extracting most of structured information. After mapping data in ontology instances, OWL individual for each event is created. If IE module not able to extract some attributes of the event then also we create an instance with empty property. Jena framework is used for ontology mapping [12]. Jena framework provides the facility for reading, writing and manipulating the data in RDF [25] and OWL [26] format.

5.6 Ontology storage in RDBMS

Ontology data are inherently dynamic and semi-structured. To store ontology in RDBMS multiple techniques are used. Our system proposes the technique in which overall schema of the system remain same even if the number of Ontologies grow. Ontology is stored in more effective way which eases the retrieval process. To elaborate the technique the different rules have been defined. Some of the rules are as follows.

Rule 1: (New Ontology / Parent Ontology Rule) For every new domain ontology an entry is done in the Onto_tbl. This table contains the ontology ID, its domain, URI, description, version, label, comments and prior versionThere is also importedontology_tbl table which contains the list of Ontologies imported by certain ontology.

Rule 2: (New Concept Rule) For each known concept a unique new concept ID is assigned and their entry is done in the Concept_tbl with the ontology ID as a foreign key with each concept. Concept_tbl also contains Concept Name, its URI, label and comments.

Rule 3: (Property/ Property characteristics Rule) Entry of every property that applies on certain concept is done in the Property_tbl. That property can be data-type property or an Object property. A unique property ID is allotted to each property which acts as a primary key. Property_tbl contains name of property, property type, Domain, Range and Range value. Where property type represents Object property or Data type property.

The details rules are stated in [4]. Process of ontology transformation is shown in figure 3.

5.7 Inference

New relation can be added from the existing database with the help of class, subclass relation. Pellet provides the complete OWL-DL reasoner with good performance and number of unique features. Pellet is the first and complete OWL-DL reasoner. It is written in java and open source [18]. Oracle 11g releases 2 also support the Pellet reasoner. Pellet in Oracle is referred as PelletDb. PelletDb is build on top of Pellet. It is optimized for Oracle 11g. PelletDb significantly improves performance compared to other systems and offers convenient access to Pellet’s advanced reasoning features, including inference explanation [7]. PelletDb provides access to Pellet’s reasoning services, including consistency checking, concept satisfiability, classification, realization; as well as non-standard reasoning services like SparQL-DL conjunctive query answering, data type reasoning, rules reasoning, inference explanation, and incremental reasoning. PelletDb is integrated with the help of Jena adaptor [7]. Basic idea is to load the ontology schema from oracle database to PelletDb reasoner, computer class subsumption tree and store it back in oracle instances.

5.8 Searching

Searching is done in combination of SQL and SparQL [3]. Set of SQL operators are introduced for semantic matching. The details of the operators can be referred in [5]. The similar operators are provided by Oracle 11g release 2 [1]. The Oracle provides SEM_MATCH operators, which can be embedded in SQL. SEM_MATCH provides the support for UNION, UNION ALL, FILTER, and OPTIONAL keywords. Other operators are SEM_DISTANCE, SEM_RELATED, etc.

[1] http://www.oracle.com

To motivate the need for ontology-based semantic matching, consider a restaurant guide application, which recommends restaurants to a user based on her/his preferences [5].

Consider a table served_food that contains the types of cuisines served at restaurant.

Table 1: Served food

Rid	Cuisine
1	American
2	Mexican
2	American
14	Portuguese

In the absence of semantic matching, the application would most likely resort to syntactic matching via the '=' operator as shown below:

```
    SELECT    *    FROM    served_food    WHERE
cuisine = 'Latin American';
```

This query generates no rows since none of Cuisine values in the table will match 'Latin American'. In contrast, the user can get more meaningful results by performing semantic matching that consults an ontology for computing the results. Specifically, a user can issue the following query:

```
    SELECT * FROM served_food
    WHERE ONT_RELATED(cuisine,
                 'IS_A',
                 'Latin American',
                 'Cuisine_ontology')=1;
```

Here the ONT_RELATED operator determines if the two input terms are related by the input relationship type argument by consulting the specified ontology. If they are related, then the operator will return 1, otherwise 0.

6. CONCLUSION

The system presented the novel approach for semantic information retrieval in cricket domain. System is able to overcome the limitations of web 2.0 by representing the knowledge in ontology. Ontology represents the knowledge in terms of classes and subclasses. Knowledge represented in ontology can be interpreted by machine. Machine can add more data and relations on behalf of users. Due to increase in use of semantic web, number of ontology has been increased which created the problem for ontology storage. Problem was there for scalable storage of ontology data. Ontology storage problem is solved by storing the ontology in RDBMS. Ontologies are stored in RDBMS for efficient maintenance, sharing and scalability. Relational databases able to achieve maturity, performance, robustness, reliability, and availability. Semantic searching is done in combination with SQL and SparQL. System uses the different semantic matching operators for ontology search. Querying the semantic data is simplified because of relational databases. In future, the same concept can be easily adapted for another domain by doing the certain changes in Information extraction, ontology design and database design. The concept can be used to build ontology for multilingual domain which collects the data from different language repository.

7. REFERENCES

[1] David Vallet, Miriam Fernández, and Pablo Castells, "An Ontology-Based Information Retrieval Model" citeseers, 2005.

[2] Soner Kara, Ozgur Alan, Orkunt Sabuncu, Samet Akpınar, Nihan K. Cicekli, Ferda N. Alpaslan, "An Ontology-Based Retrieval System Using Semantic Indexing", IEEE, ICDE workshop 2010.

[3] SparQL Protocol for RDF, K. Clark, Editor, W3C Recommendation, 15 January 2008, http://www.w3.org/TR/2008/REC- rdf-sparql-protocol-20080115.

[4] Adnan Khalid, Syed Adnan Hussain Shah, Muhammad Abdul Qadir, "OntRel: An Optimized Relational Structure for Storage of Dynamic OWL-DL Ontologies", 2009, International Conference of Soft Computing and Pattern Recognition.

[5] Souripriya Das, Eugene Inseok Chong, George Eadon, Jagannathan Srinivasan, Oracle Corporation, "Supporting Ontology-based Semantic Matching in RDBMS", 2004, Proceedings of the 30th VLDB Conference, Toronto, Canada.

[6] Eugene Inseok Chong Souripriya Das George Eadon Jagannathan Srinivasan, "Supporting Keyword Columns with Ontology-based Referential Constraints in DBMS", 2006, IEEE, Proceedings of the 22nd ICDE Conference.

[7] "Introducing PelletDb (Expressive, Scalable Semantic Reasoning for the Enterprise)", 2009, Oracle white papers.

[8] Irina Astrova, Ahto Kalja, "Automatic Transformation Of Owl Ontologies To Sql Relational Databases", IADIS European Conference Data Ming 2007.

[9] R. Garcia, M.M., A. Montes, J.F. "A Tool for Storing OWL Using Database Technology", (November 2005).

[10] Irina Astrova, Nahum Korda, and Ahto Kalja, "Storing OWL Ontologies in SQL Relational Databases", World Academy of Science, Engineering and Technology 29 2007.

[11] E. Vysniauskas, and L. Nemuraite, "Transforming ontology representation from OWL to relational database," Information Technology and Control, vol. 35A, no. 3, 2006.

[12] Jena, A.: Semantic Web Framework for Java, http://jena.sourceforge.net/ontology/index.html.

[13] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.

[14] Wen-tau Yih, "Template-based Information Extraction from Tree-structured HTML documents", Citeseer, 1997.

[15] Naveen Malviya, Nishchol Mishra, Santosh Sahu, "Developing University Ontology using protégé OWL Tool: Process and Reasoning", International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011.

[16] "Oracle Semantic Technologies Inference Best Practices with RDFS/OWL", 2009, Oracle White paper.

[17] Wen-tau Yih, "Template-based Information Extraction from Tree-structured HTML documents", Citeseerx, 1997.

[18] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur , Yarden Katz, "Pellet: A Practical OWL-DL Reasoner", publiced by Elsevier B.V, 2007.

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

[20] Michael Grobe , "RDF, Jena, SparQL and the Semantic Web", SIGUCCS'09, October 11– 14, 2009, St. Louis, Missouri, USA.

[21] Raman Kumar Goyal, Vikas Gupta, Vipul Sharma, Pardeep Mittal, "Ontology based web retrieval", 2008.

[22] Abad Shah, Amjad Farooq, Syed Ahsan and Mohammad Imran, "An Indexing Technique for web ontologies",

Journal of computing, volume 2, issue 7, july 2010, issn 2151-9617.

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

[24] Apache Lucene, http://lucene.apache.org/core/.

[25] RDF, http://www.w3.org/TR/rdf-primer/

[26] OWL, http://www.w3.org/TR/owl-features/

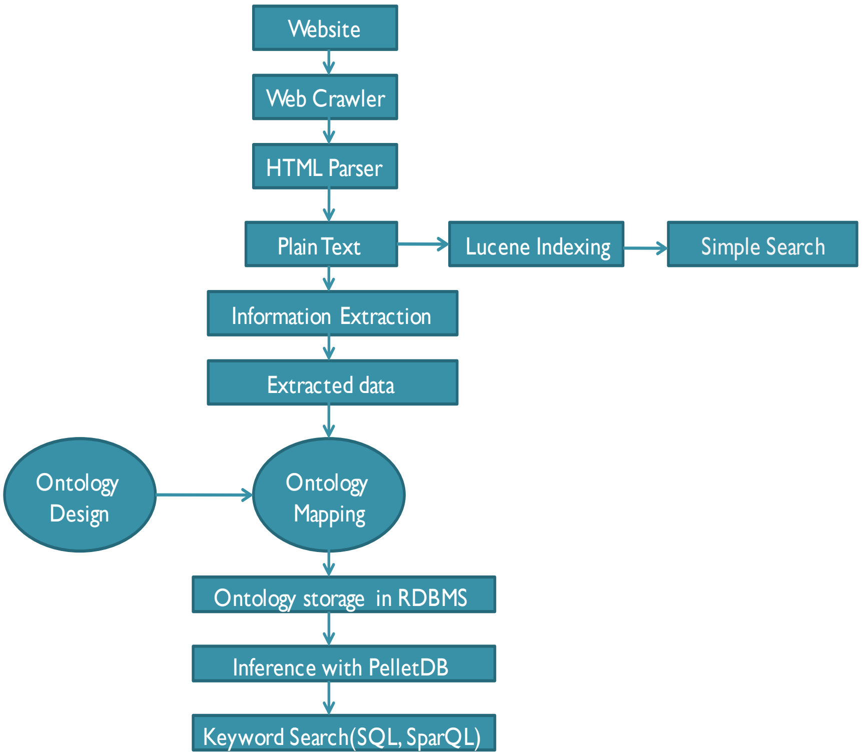

Fig 1: Proposed system

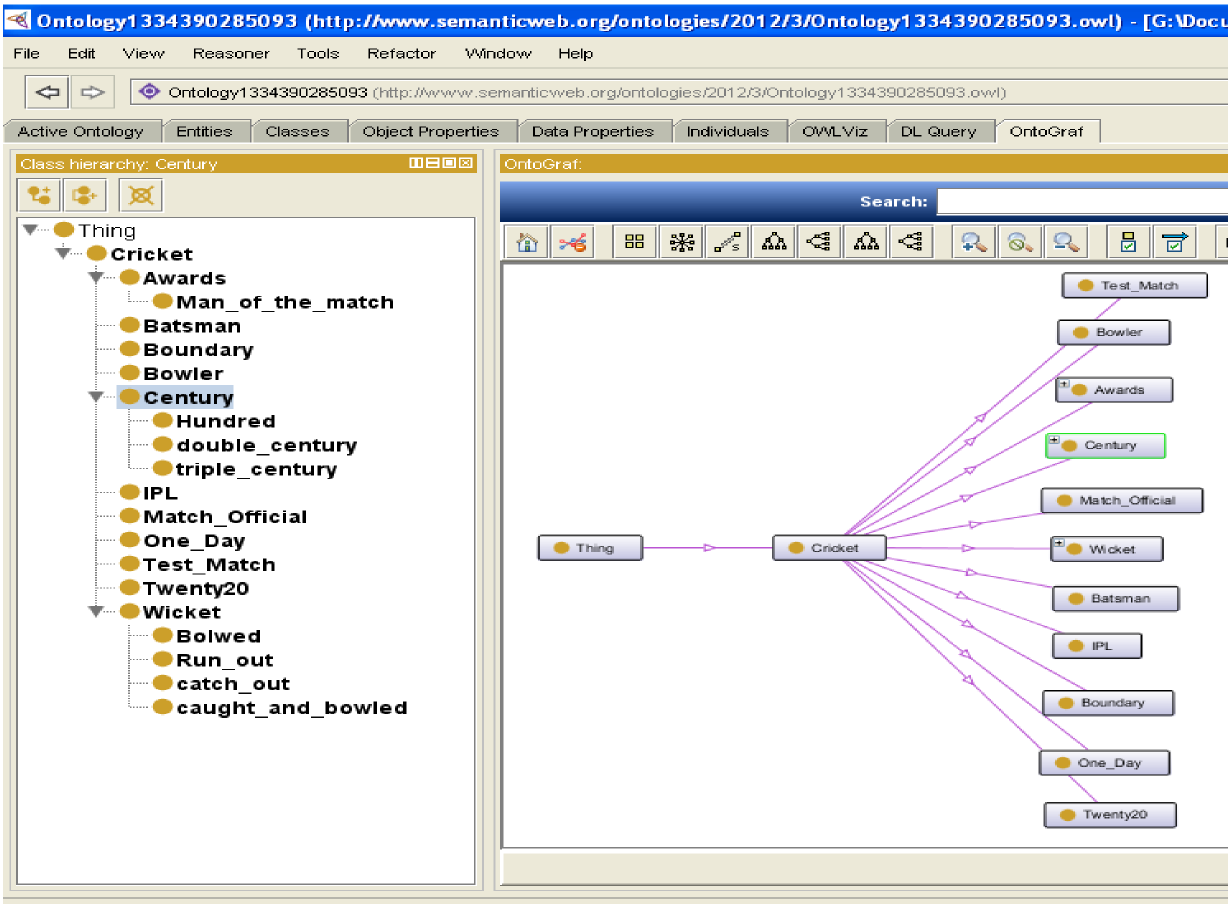

Fig 2: Cricket Ontology using Protégé.

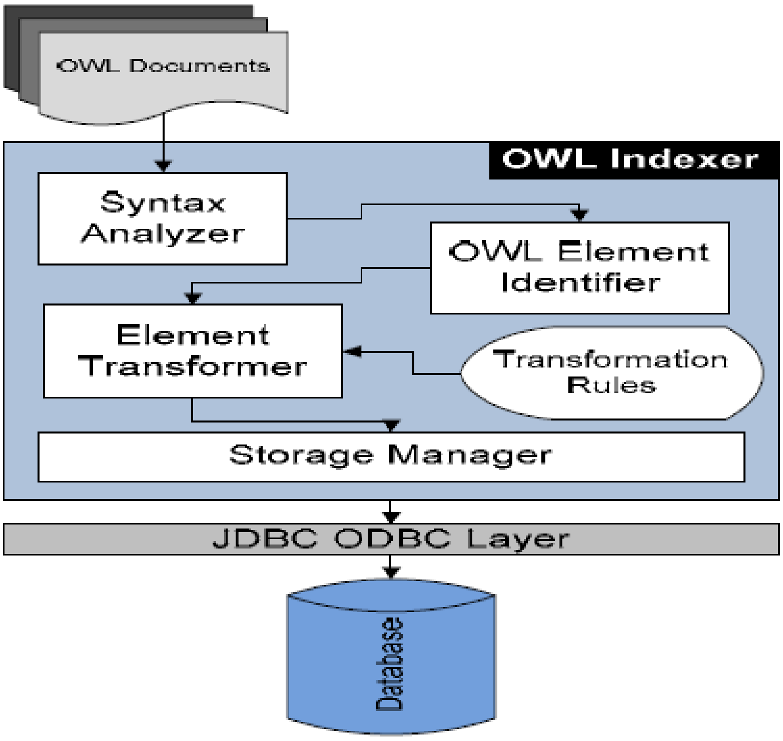

Fig 3: Transforming the OWL document to RDBMS.