Ontology design and development using ontology editors along with semantic search patterns towards intelligent retrieval of information on web: case studies

Sanjay Kumar Malik*

University School of Information Technology,
GGS Indraprastha University,
Dwarka, New Delhi, India
E-mail: sdmalik@hotmail.com
*Corresponding author

S.A.M. Rizvi

Department of Computer Science,
Faculty of Natural Sciences,
Jamia Millia Islamia,
New Delhi, India
E-mail: samsam_rizvi@yahoo.com

Abstract: In the present era of transition from Web 2.0 to an Intelligent Web 3.0, Sir Tim Berner's LEE Semantic Web layered architecture, is playing a key role for its significant layer issues where ontology is a milestone in incorporating semantics and in conceptualising shared knowledge of a particular domain or several domains. Ontology development using ontology editors is crucial which may be best realised with case studies. To endow the web itself with intelligence, the need is to provide semantic annotations search patterns on web documents of an ontology where semantic annotation attaches metadata to the documents, pointing to concepts and properties in an ontology and is the glue that ties ontologies into document spaces, via metadata where information is typically exported as text annotated with links to the ontology. Two case studies on ontology development have been presented using protégé and swoop ontology editors. First, an extended case study of a 'Sports Complex Ontology' is presented using protégé 3.4.4 along with an ontology flowchart and using SWOOP 2.3 Beta 4 Ontology Editor is presented highlighting various features like Venn diagram of class hierarchy, fly the mothership, pellet query instance search, ontology partition, adding annotation property, etc. Second, various semantic annotation search patterns based on 'pattern', 'Boolean', and 'facets' have been presented using KIM platform. Third, another case study illustration of a 'Dwarka Shopping Complex Ontology' is presented using SWOOP 2.3 Beta 4 Editor highlighting LHS and RHS of GCI, instances with and without GCI axioms, query retrieval look up, adding property, class expression table and ontology statistics.

Keywords: intelligent web; ontology design; swoop; protégé; class hierarchy; fly the mothership, pellet, ontology partition, semantic annotation search patterns; KIM; proton; Venn diagram; ontology statistics; class expression table; query retrieval; annotation property; GCI axioms

Reference to this paper should be made as follows: Malik, S.K. and Rizvi, S.A.M. (2014) 'Ontology design and development using ontology editors along with semantic search patterns towards intelligent retrieval of information on web: case studies', *Int. J. Autonomic Computing*, Vol. 2, No. 1, pp.73–98.

Biographical notes: Sanjay Kumar Malik has more than 15 years of experience in academics and industry in India and abroad (Dubai and USA) and is presently working as Assistant Professor in University School of Information and Communication Technology, GGS Indraprastha University, Dwarka, New Delhi. He is MTech (IT) and currently pursuing PhD from GGS IP University and published various research papers in international journals/conferences and attended international conferences in USA.

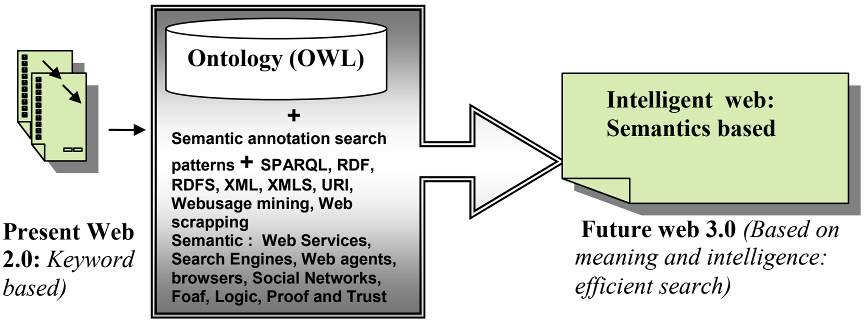
S.A.M. Rizvi is PhD in Computer Science and Engineering, currently working as an Associate Professor at the Department of Computer Science, Jamia Millia Islamia, New Delhi, India having more than 26 years of experience in India and abroad. He is an expert in software engineering, who has published numerous papers in the field of software engineering, MIS, mathematical modelling, bioinformatics and web-based applications.

This paper is a revised and expanded version of a paper entitled 'Ontology design towards web intelligence: a sports complex ontology case study' presented at CASoN-IAS-NWeSP joint IEEE International Conference, University of Sao Paulo, Sao Carlos, Brazil, 21–23 November 2012.

1 Introduction

With Web 3.0, search engines will be more and more user requirements specific and produce precise search results, supposed to handle more complex queries in very much the way we speak making the web, humans and machines more communicative and sharable. A search engine will not only look for a string of keywords that are entered, but it will understand its relevance and context with respect to the needs. It will be a web organised according to 'meaning' of information and its context, which will make emergence of intelligent search engines possible. We all realise that the need of the day is to further extend the present human readable web to a more valuable machine understandable intelligent web with better information sharing or exchange in a meaningful way by encoding some semantics of resources in a machine-processable form. This is a challenging task which may be achieved by Semantic Web which dreams about a scenario of a web where computers will be better able to search, process, integrate and present the content of web resources in a meaningful or intelligent manner and therefore issues of intelligent information retrieval of information on web are of significant concern. In this direction, ontology and semantic search patterns are two prime aspects where ontologies enable a machine-understandable form of content description, such that the retrieval process can be driven by the meaning of the content and semantic annotation with semantic search patterns endows the web with intelligence.

Figure 1 Key issues towards an intelligent web (see online version for colours)

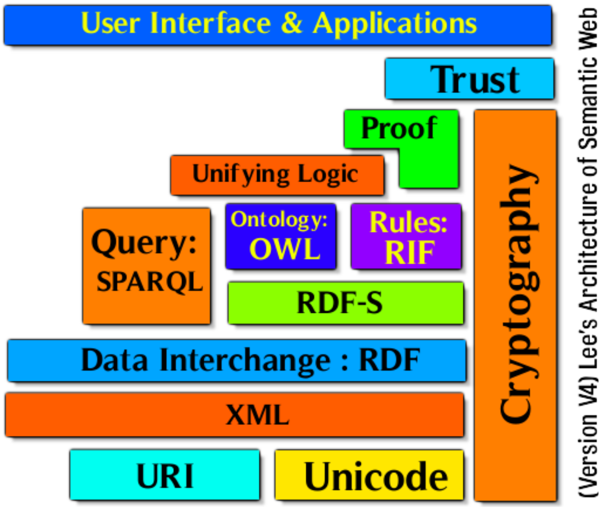

Ontology (OWL)
+
Semantic annotation search patterns + SPARQL, RDF, RDFS, XML, XMLS, URI, Webusage mining, Web scrapping
Semantic : Web Services, Search Engines, Web agents, browsers, Social Networks, Foaf, Logic, Proof and Trust
Intelligent web: Semantics based
Present Web 2.0: *Keyword based)*
Future web 3.0 *(Based on meaning and intelligence: efficient search)*

An intelligent web refers to the next generation web which will lead to an IT revolution in the area of information search on web in terms of a more efficient, intelligent or meaningful search. It refers to a collaboration of humans with machines where machines will also be capable of understanding and analysing transactions on the web like humans. Semantic Web refers to a dream concept of how computers, people, and the web will be able to work together more effectively than is possible now for a better information web search and agents (software programs) will be performing various tasks by communicating with other agents and seeking information from various web resources (Shadbolt et al., 2006; Passin, 2004; Berners-Lee et al., 2006).

According to Tim Berners-Lee vision of Semantic Web, 1999 "I have a dream for the web in which computers become capable of analyzing all the data on the web-the content, links and transactions between people and computers". For making this happen true, several approaches are being proposed like annotating web which is very labour intensive or to build several software agents carrying information from one web page to another or using techniques like scrappers to extract chunks of information from web pages or to have several ontologies etc where ontologies and semantic search patterns are critical and the need is to focus on them. An ontology, represents an area of knowledge to provide a formal semantics that can be employed to process and integrate information on the web, is an explicit specification of conceptualisation (Gruber, 1993).

Intelligent information retrieval on web refers to a web where information will be retrieved intelligently or meaningfully in a more efficient manner where the user will query the search engine about some information, the search engine, web and computer machines will understand what has been queried and accordingly answer. Semantic and syntactical analysis along with various key issues are of significance. Sir Tim Berners-Lee proposed an architecture of Semantic Web (Shadbolt et al., 2006; Berners-Lee et al., 2001; Hebler et al., 2009; Taniar and Rahayu, 2006) which comprises of several components like XML [Extended Markup Language, RDF/RDFS (Resource Description Framework Schema), Ontology/OWL (Web Ontology Language), SPARQL (SPARQL Protocol and RDF Query Language), URI (Uniform Resource Identifier)]. The layered architecture for Semantic Web as proposed by Sir Tim Berners-Lee is as presented below in Figure 2 (Shadbolt et al., 2006; Berners-Lee et al., 2001; Hebler et al., 2009; Taniar and Rahayu, 2006).

Figure 2 Layered architecture for Semantic Web as proposed by Berners's LEE (see online version for colours)



Source: Hebler et al. (2009) and Taniar and Rahayu (2006)

The above architecture provides a foundation for fulfilling the dream of retrieving information on web intelligently or meaningfully and efficiently. Besides above, various other significant issues are: semantic annotation, search engines, Semantic Web browsers, web services, web usage mining, web scrapping, PSP, software agents or robots, Jena framework, eclipse environment, semantic search engines, FOAF, social networks, semantic search engines, etc. Similarly, ontology has multifarious issues in which design and development is significant and our concern. The summary is as follows:

Table 1 Summary of a few significant issues of intelligent information retrieval on web including ontology

Intelligent information retrieval on web: key issues	*Refers to:*
XML/XMLS (Extended Markup Language Schema)	Text-based meta-language for data exchange and express relations among them and a query language to express queries based on XML (XQuery). Advantage: allows to add its own tags unlike HTML. Drawback: XML does not offer semantics and logic capabilities and therefore, the next step is resource description framework (RDF) which begins to establish semantics on the web
RDF/RDFS (Resource Description Framework Schema)/RDF XML	Framework for representing the different resources and their relationships on the web based on XML language and mainly represents the meta data on the web and contains statements called triples as its basic building block containing three parts viz., subject, predicate and object (triple). Provides the syntactic and semantic model for defining machine processable ontologies and its role in annotating content is better than XML. RDF/XML is for recording and exchanging relationships and for representing RDF triples which is the standard exchange syntax for RDF serialisation.

Source: Shadbolt et al. (2006), Lee and Lassila (2001), Hebler et al. (2009), and Taniar and Rahayu (2006)

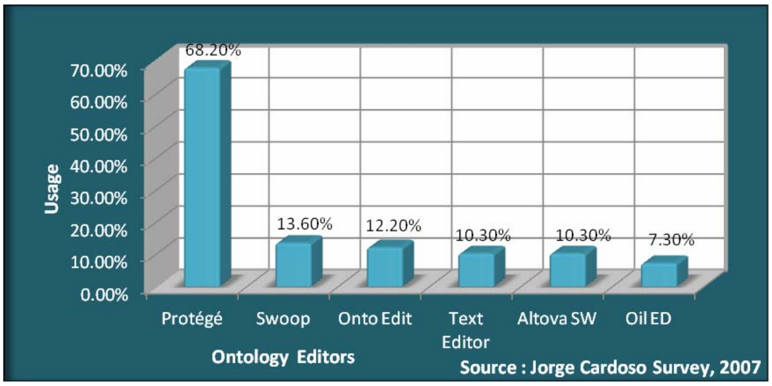
Table 1 Summary of a few significant issues of intelligent information retrieval on web including ontology (continued)

Intelligent information retrieval on web: key issues	*Refers to:*
Uniform Resource Identifier (URI)	Determines what the object is that is being described (the subject of the statement) and it gives each resource a unique identity.
SPARQL (SPARQL Protocol and Query Language)	Query language for querying RDF which is a directed and labelled graph data format for representing information on the web
Semantic Web services	Software applications that can be discovered, described and accessed based on XML and standard web protocols aiming to support interaction between networks mechanisms and enhance the web services usability by incorporating meaning and structure to available information. They are based on XML (data description and exchanged format), SOAP (protocol of service call up and message transmission), WSDL (service description format) and UDDI (catalogue for registering and searching of services).
Semantic Web browsers	Developing meta driven concise search browsers which support RDF dataset and can store information in a centralised fashion or in a RDF Store after extracting information from various sites, and provide data with intelligent searches allowing the users to traverse among the semantically connected documents easily
Search engines	Indexing the entire web is a challenge for traditional keyword-based search engines and the need is to incorporate semantics in them and managing the complex queries intelligently.
Semantic annotation	The need is to extract useful pieces of knowledge within the data with little or no human involvement and it enables machines to discover required information from raw data.
Web agents	An agent can be defined as a persistent software entity that perceives its environment through sensors and acts upon that environment through effectors and provide smart support for locating resources on web.
Web usage mining	To extract some useful information and it mines the required information of access routes kept back in servers, in the form of server logs, i.e., the information of access manners of users visit the websites
Web scrapping	Extracting useful information from HTML pages and it is implemented with powerful string matching operations
Ontology and its key issues	Means for sharing the knowledge of various domains on the web. Issues like: design and development, merging, integrating, implementation, validation and evaluation, maintenance, import/export, mapping/matching/alignment, comparison/ranking, comparison and ranking, validation, FOAF (standard ontology used to identify social relations, all one has to do is generate a FOAF profile describing himself and publish it on the web)

Source: Shadbolt et al. (2006), Lee and Lassila (2001), Hebler et al. (2009), and Taniar and Rahayu (2006)

In this paper, prime concern is focus on ontology's issue of 'Design and Development using Ontology Editors'. Various ontology editors are available for ontology design and development but Protégé (http://protege.stanford.edu/) and SWOOP (http://mindswap.org/2004/swoop/) are most widely used open source editors. A survey by Sir Jorge Cardoso in 2007 on a few ontology editors is as shown.

Figure 3 Jorge Cardoso survey on ontology editors used by respondents (researcher, professional, programmer, etc.) domains (see online version for colours)



Source: Adapted from Cardoso (2007)

Inference

Table 2 Inference of highest usage of ontology editors

Ontology editor	*Market share/usage*	*Inference*
Protégé	68.20%	Most widely used
SWOOP	13.60%	Second most widely used

Ontology forms the heart of any system of knowledge representation for any given domain and without ontologies, or the conceptualisations that underlie knowledge, there cannot be a vocabulary for representing knowledge (Chandrasekran et al., 1999). Ontology development includes: defining classes in the ontology, arranging the classes in a taxonomic (subclass-superclass) hierarchy, defining slots (*relationships*) and describing allowed values for these slots, filling in the values for slots for instances (Noy and McGuinness, 2000).

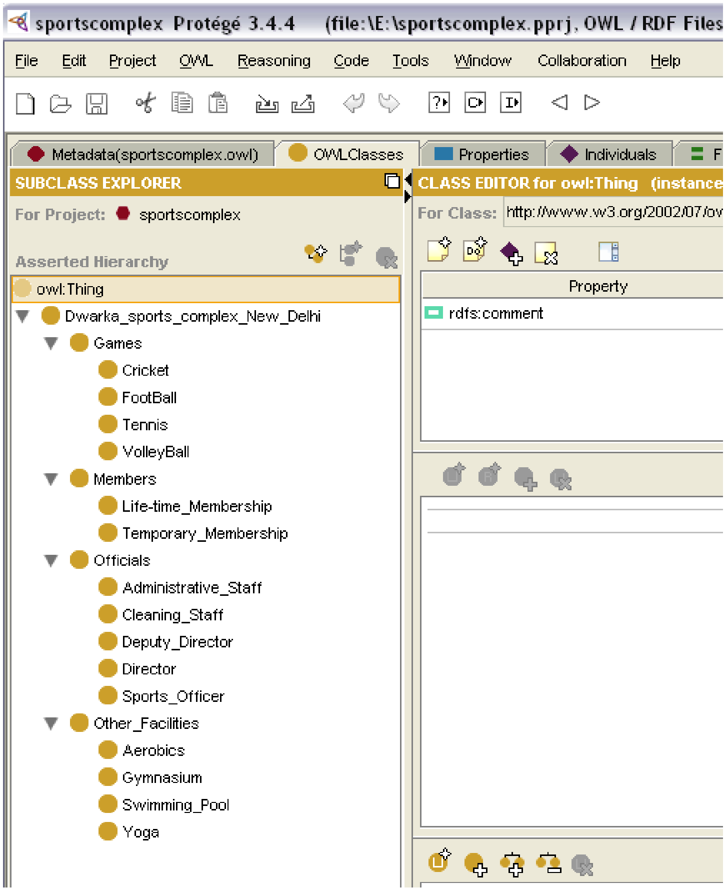
In this paper, introduction summarises the various significant issues of intelligent information retrieval on web focusing 'ontology design and development using Protégé and SWOOP'. In the first section, a revised case study of a 'Sports Complex Ontology' is presented using Protégé 3.4.4 along with an ontology flowchart and its extension work of development in SWOOP 2.3 Beta 4 Ontology Editor is presented highlighting Venn diagram of class hierarchy, fly the mothership, pellet query, ontology partition summary, adding annotation property etc. Second, various semantic annotation search patterns based on 'pattern', 'Boolean', and 'facets' have been presented using KIM platform. Third, another case study is illustrated of a 'Dwarka Shopping Complex Ontology' is presented using SWOOP 2.3 Beta 4 Editor highlighting LHS and RHS of GCI, instances with and without GCI axioms, query retrieval look up, adding property, class expression table and ontology statistics.

2 Design and development of 'Sports Complex Ontology' in Protégé and SWOOP ontology editors

2.1 Ontology design and development using Protégé 3.4.4

An ontology extended version is created using Protégé 3.4.4 of 'Sports Complex, Dwarka, New Delhi'. By clicking on 'OWL Classes' tab on the top, a screen is obtained with hierarchy: 'Dwarka_Sports_Complex_New Delhi' as super class and the sub-classes as: 'Officials', 'Members', 'Games' and 'Other _Facilities' where 'Officials' has further subclasses like: 'Director', 'Deputy_Director', 'Sports_Officer', etc. The sub-classes of 'Dwarka_Sports_Complex_New Delhi' are 'Officials', 'Members', 'Games' and 'Other_Facilities' which are shown in figure created using OWLVIZ plugin of Protégé 3.4.4 (refer to following figures).

Figure 4 Class hierarchy of 'Dwarka Sports Complex Ontology' using Protégé 3.4.4 (see online version for colours)


sportscomplex Protégé 3.4.4 (file:\E:\sportscomplex.pprj, OWL / RDF Files
File
Edit
Project
OWL
Reasoning
Code
Tools
Window
Collaboration
Help
Metadata(sportscomplex.owl)
OWLClasses
Properties
Individuals
F
SUBCLASS EXPLORER
CLASS EDITOR for owl:Thing (instance
Asserted Hierarchy
Property

For Project: sportscomplex

For Class: http://www.w3.org/2002/07/ow

- owl:Thing
- Dwarka_sports_complex_New_Delhi
  - Games
    - Cricket
    - FootBall
    - Tennis
    - VolleyBall
  - Members
    - Life-time_Membership
    - Temporary_Membership
  - Officials
    - Administrative_Staff
    - Cleaning_Staff
    - Deputy_Director
    - Director
    - Sports_Officer
  - Other_Facilities
    - Aerobics
    - Gymnasium
    - Swimming_Pool
    - Yoga

rdfs:comment

Figure 5 Subclasses using OWLVIZ plugin of Protégé 3.4.4: graphical representation (see online version for colours)

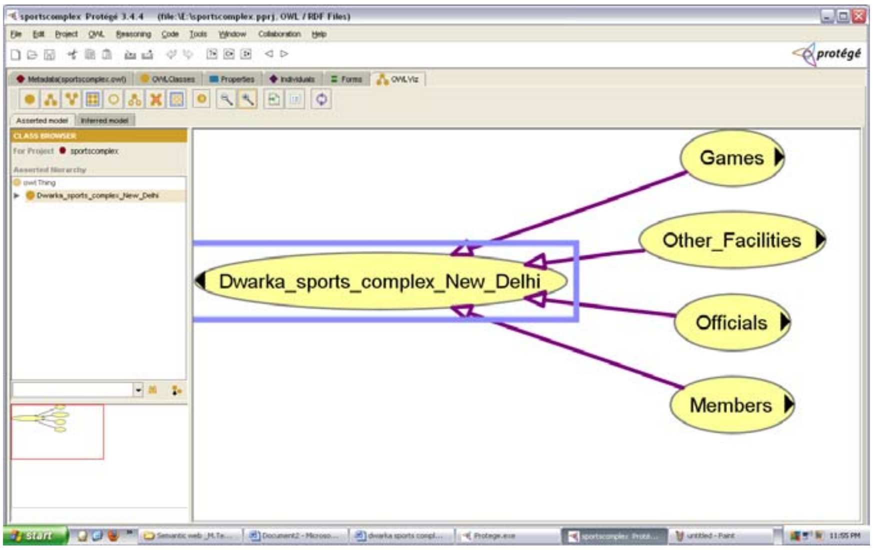
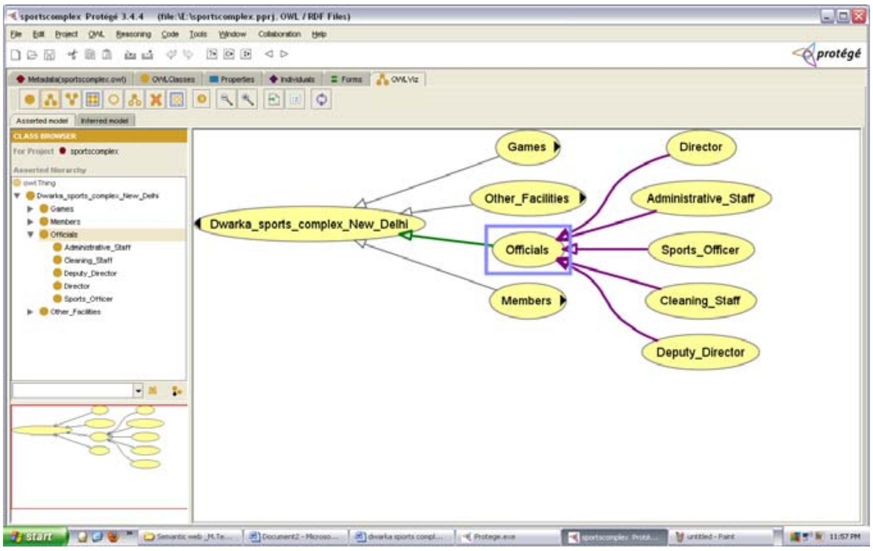

sportscomplex Protégé 3.4.4 (file:\E:\sportscomplex.pprj, OWL / RDF Files)
File Edit Project OWL Reasoning Code Tools Window Collaboration Help
protégé
Metadata(sportscomplex.owl)
OWLClasses
Properties
Individuals
Forms
OWLViz
Asserted model
Inferred model
CLASS BROWSER
For Project sportscomplex
Asserted Hierarchy
owl:Thing
Dwarka_sports_complex_New_Delhi
Games
Other_Facilities
Officials
Members
Dwarka_sports_complex_New_Delhi
start

Figure 6 Subclasses of 'Officials' class using OWLVIZ plugin of Protégé 3.4.4 (see online version for colours)



For query retrieval, following may be executed which results into various 'subclass-superclass' relationships. SPARQL query:

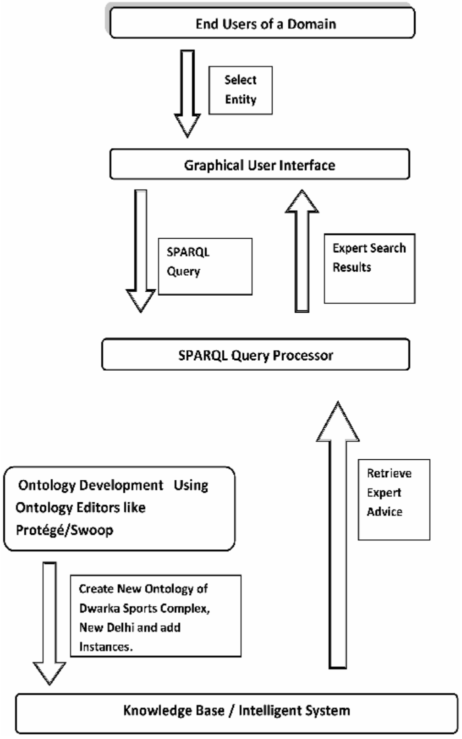
SELECT?subject ?object

WHERE{?subject rdfs:subClassOf ?object}

Now, we present an ontology flowchart for the said ontology as follows:

Ontology flowchart

Figure 7 A flowchart of 'Dwarka Sports Complex Ontology'

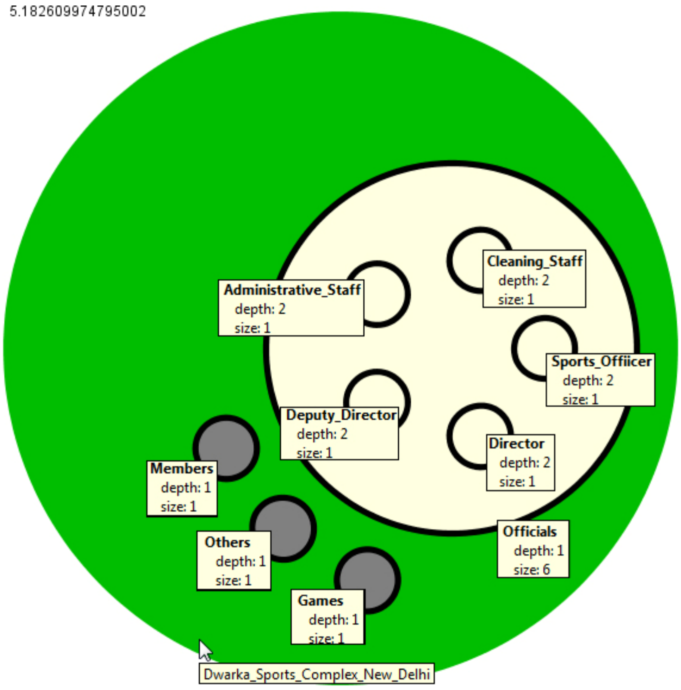

End Users of a Domain
Select Entity
Graphical User Interface
SPARQL Query
Expert Search Results
SPARQL Query Processor
Ontology Development Using Ontology Editors like Protégé/Swoop
Retrieve Expert Advice
Create New Ontology of Dwarka Sports Complex, New Delhi and add Instances.
Knowledge Base / Intelligent System

The flowchart of 'Dwarka Sports Complex Ontology' comprises of various components like selecting entity, expert search results, retrieve expert advice, add instances, creating ontology, SPARQL query processor etc. whose flow has been shown in the above flowchart. Developing an ontology includes: defining classes in the ontology, arranging the classes in a taxonomic (subclass-superclass) hierarchy, defining slots (*relationships*) and describing allowed values for these slots, filling in the values for slots for instances (Noy and McGuinness, 2000).

2.2 Ontology design and development using SWOOP

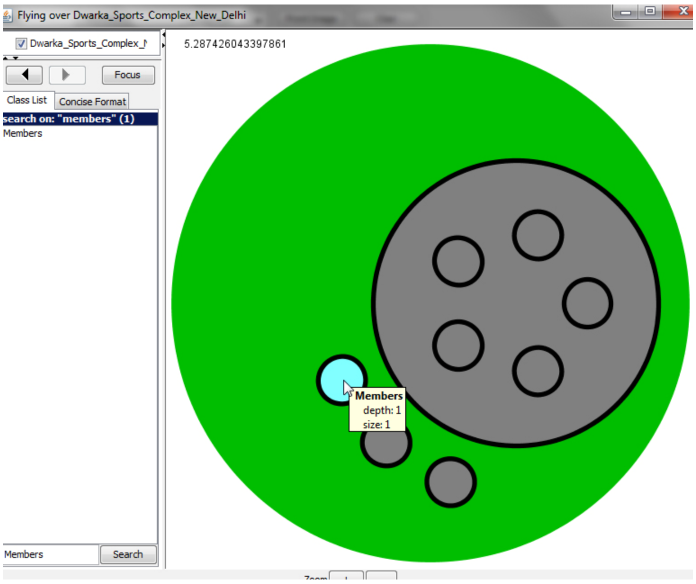
Now, We illustrate the 'Dwarka Sports Complex New Delhi' ontology, using SWOOP 2.3 Beta 4 Ontology Editor to presents expected Venn diagram of class hierarchy and GCI [various aspects like GCI (general concept inclusion axiom) by using classes, properties and instances along with Query Retrieval by options of 'fly the mothership' option with output as in Venn diagram and 'look up query' and also presenting ontology statistics and observation with class expression table]. SWOOP is an ontology browser and editor, designed specifically for use with web ontology language. It meant for rapid and easy browsing and development of web ontologies. It creates primary identifiers for ontologies, classes, properties, and individuals to support hyper textual navigation through, and between ontologies.

Figure 8 Display class hierarchy of ‘Dwarka Sports Complex New Delhi’ ontology in form of Venn diagram (see online version for colours)



Fly The Mothership

Figure 9 Using ‘Fly The Mothership’ option (see online version for colours)

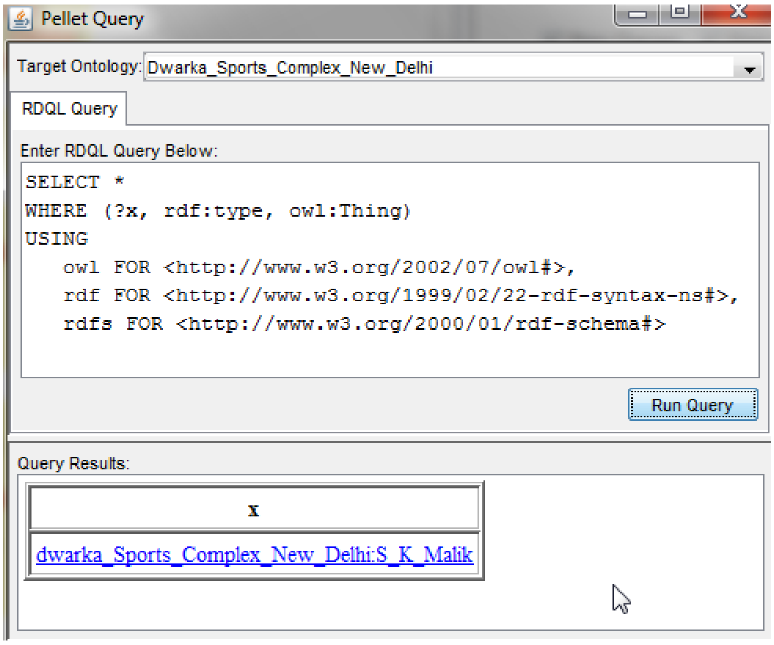


search on: "members" (1)
Members

Members

Pellet query

Figure 10 Searching for instance ‘S_K_Malik’ of class ‘Members’ by pellet query (see online version for colours)



Target Ontology: Dwarka_Sports_Complex_New_Delhi

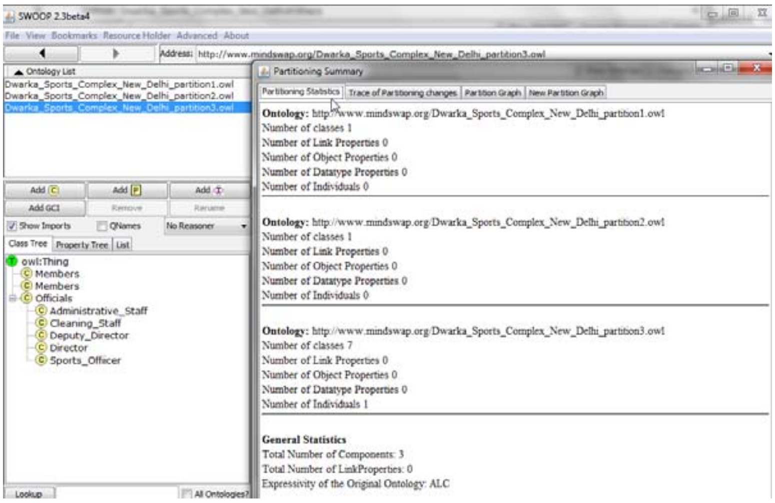
Enter RDQL Query Below:

```
SELECT *
WHERE (?x, rdf:type, owl:Thing)
USING
   owl FOR <http://www.w3.org/2002/07/owl#>,
   rdf FOR <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
   rdfs FOR <http://www.w3.org/2000/01/rdf-schema#>
```

Query Results:

x
dwarka_Sports_Complex_New_Delhi:S_K_Malik

Figure 11 Partition the ontology and observing the partitioning statistics (see online version for colours)



Address: http://www.mindswap.org/Dwarka_Sports_Complex_New_Delhi_partition3.owl

Dwarka_Sports_Complex_New_Delhi_partition1.owl
Dwarka_Sports_Complex_New_Delhi_partition2.owl
Dwarka_Sports_Complex_New_Delhi_partition3.owl

Ontology: http://www.mindswap.org/Dwarka_Sports_Complex_New_Delhi_partition1.owl
Number of classes 1
Number of Link Properties 0
Number of Object Properties 0
Number of Datatype Properties 0
Number of Individuals 0

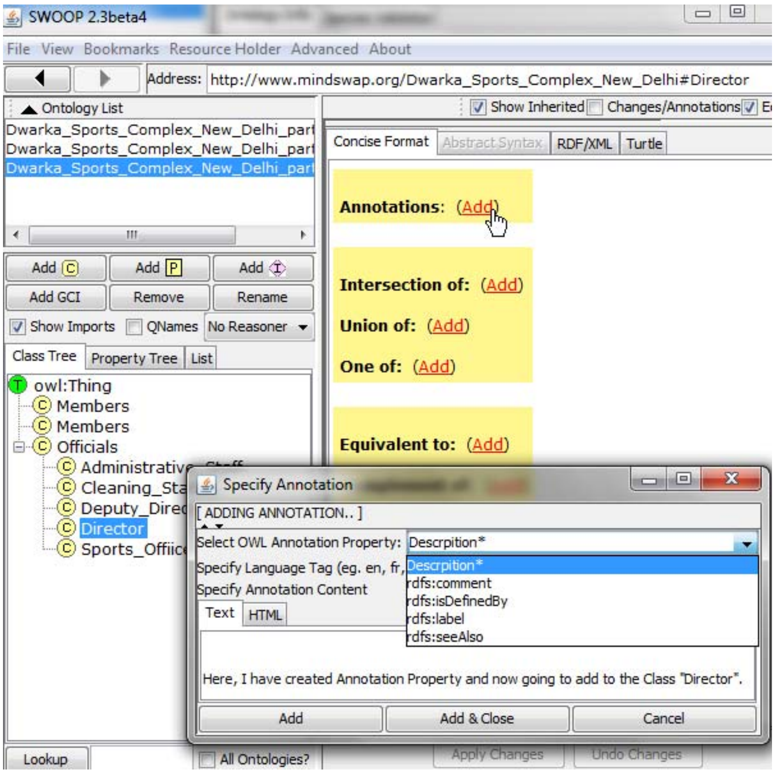
Ontology: http://www.mindswap.org/Dwarka_Sports_Complex_New_Delhi_partition2.owl
Number of classes 1
Number of Link Properties 0
Number of Object Properties 0
Number of Datatype Properties 0
Number of Individuals 0

Ontology: http://www.mindswap.org/Dwarka_Sports_Complex_New_Delhi_partition3.owl
Number of classes 7
Number of Link Properties 0
Number of Object Properties 0
Number of Datatype Properties 0
Number of Individuals 1

General Statistics
Total Number of Components: 3
Total Number of LinkProperties: 0
Expressivity of the Original Ontology: ALC

Adding annotation property

Figure 12 Adding annotation property (see online version for colours)



Address: http://www.mindswap.org/Dwarka_Sports_Complex_New_Delhi#Director

[ADDING ANNOTATION..]

Select OWL Annotation Property: Descrpition*

Specify Language Tag (eg. en, fr,

Specify Annotation Content

Here, I have created Annotation Property and now going to add to the Class "Director".

Selecting 'Partition Automatically' which is a sub-option of 'Advanced' tab, then 'Continue?' and after selecting 'yes', it will Transform an OWL ontology to an e-connection which results all super classes to be separated into respective ontologies.

Abstract syntax of ontology snippet

Namespace(rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)

Namespace(owl = <http://www.w3.org/2002/07/owl#>)

Namespace(xsd = <http://www.w3.org/2001/XMLSchema#>)

Namespace(rdfs = <http://www.w3.org/2000/01/rdf-schema#>)

Namespace(a = <Dwarka_Sports_Complex_New_Delhi#>)

Ontology(<Dwarka_Sports_Complex_New_Delhi>

Annotation(rdfs:label "Dwarka Sports Complex New Delhi Ontology")

Annotation(owl:versionInfo "Dwarka_Sports_Complex_New_Delhi")

Class(a:Administrative_Staff partial a:Officials)

```
Class(a:Cleaning_Staff partial a:Officials)

Class(a:Deputy_Director partial a:Officials)

Class(a:Director partial a:Officials)

Class(a:Games partial annotation(rdfs:label "Games/Class of
    Dwarka_Sports_Complex_New_Delhi"))

Class(a:Members partial)

Class(a:Officials partial)

Class(a:Others partial)

Class(a:Sports_Offiicer partial a:Officials)

AnnotationProperty(rdfs:comment)

AnnotationProperty(rdfs:label)

AnnotationProperty(owl:versionInfo)

Individual(a:S_K_Malik

annotation(rdfs:comment "SK Malik is Instance of Member Class.")

annotation(rdfs:label "Instance of \"Member\"Class")

type(a:Members)))
```

3 Semantic annotation search patterns: semantic search on a web document using KIM

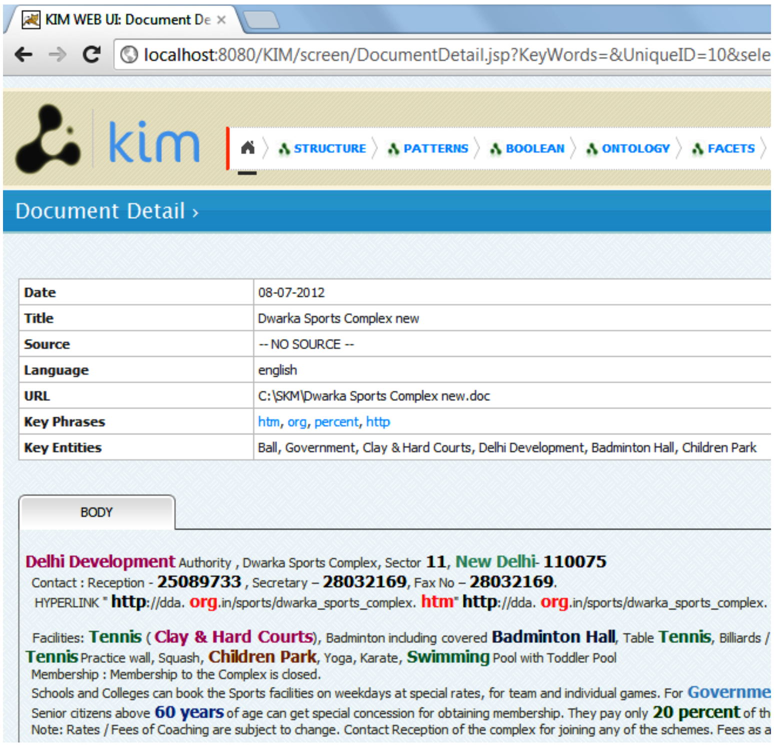
Semantic annotation

Semantic annotation is about assigning to the entities in the text and links to their semantic descriptions and makes it possible to add semantics to unstructured and semi-structured documents on the web trying semantic models and natural language together which may be characterised as dynamic creation of interrelationships between ontologies and documents which may be performed using tools like KIM (Kiryakaov et al., 2004; Davies et al., 2006). KIM provides a web user interface that allows traditional access methods (keyword search) and semantic search (like entity search, pattern search, etc.) and KIM Web UI is front end of KIM providing a powerful search interface. After populating the knowledge base with ontology and documents using KIM document store population tool, one can search for queries where one may reuse the information extraction functionality regarding pattern matching grammars in the system which is based GATE platform which makes it possible for KIM to relate grammar rules to ontology concepts.

Semantic search patterns

Various types of semantic search annotation on a web document of the said proton ontology are illustrated using KIM, a platform for semantic information extraction or retrieval and thus endowing the web itself with intelligence. For better and intelligent search, a document may be semantically annotated. The various terms/entities are tagged on the basis of proton ontology, knowledge base and repositories. Once the document is annotated, we can perform search operations semantically. In the above illustration, we have a web document for Dwarka sports complex, New Delhi. After semantic annotation using KIM, the various entities are tagged/annotated and now various types of semantic search can be performed on the document.

Figure 13 Semantic annotation on a web document of 'Sports Complex Ontology' using KIM platform (see online version for colours)


KIM WEB UI: Document De
localhost:8080/KIM/screen/DocumentDetail.jsp?KeyWords=&UniqueID=10&sele
kim
STRUCTURE
PATTERNS
BOOLEAN
ONTOLOGY
FACETS
BODY

Document Detail ›

Date	08-07-2012
Title	Dwarka Sports Complex new
Source	-- NO SOURCE --
Language	english
URL	C:\SKM\Dwarka Sports Complex new.doc
Key Phrases	htm, org, percent, http
Key Entities	Ball, Government, Clay & Hard Courts, Delhi Development, Badminton Hall, Children Park

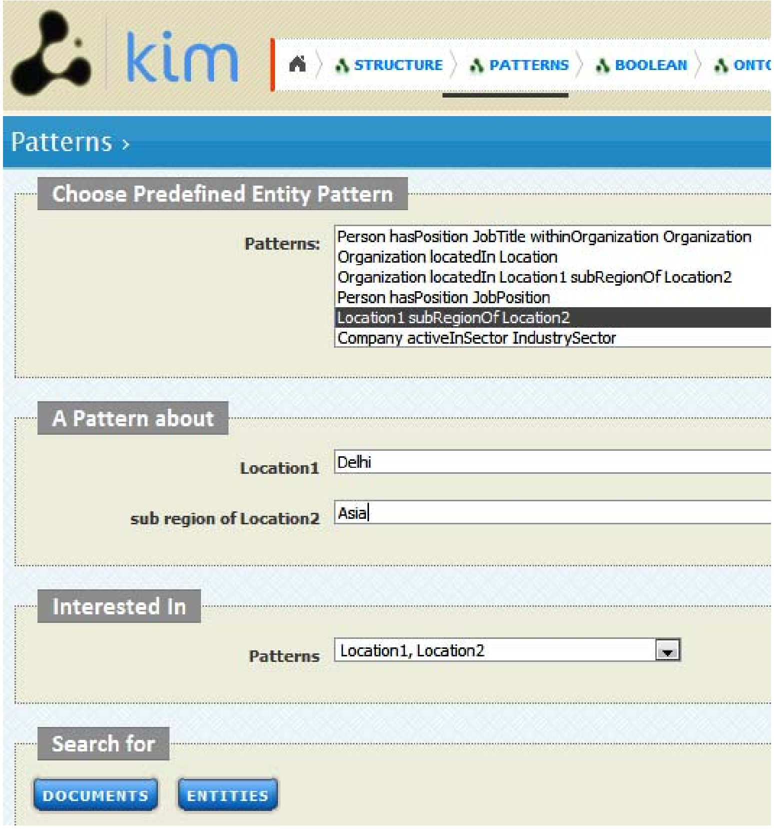
Delhi Development Authority , Dwarka Sports Complex, Sector 11, New Delhi- 110075
Contact : Reception - 25089733 , Secretary – 28032169, Fax No – 28032169.
HYPERLINK " http://dda. org.in/sports/dwarka_sports_complex. htm" http://dda. org.in/sports/dwarka_sports_complex.

Facilities: Tennis (Clay & Hard Courts), Badminton including covered Badminton Hall, Table Tennis, Billiards /
Tennis Practice wall, Squash, Children Park, Yoga, Karate, Swimming Pool with Toddler Pool
Membership : Membership to the Complex is closed.
Schools and Colleges can book the Sports facilities on weekdays at special rates, for team and individual games. For Governme
Senior citizens above 60 years of age can get special concession for obtaining membership. They pay only 20 percent of th
Note: Rates / Fees of Coaching are subject to change. Contact Reception of the complex for joining any of the schemes. Fees as a

Search based on 'pattern'

It provides a set of frequently used queries. These queries incorporate a predefined pattern frame of entities with specified types. For example, it can be a location which is a sub region of another location. E.g., Delhi may be considered as a sub location of Asia as Delhi is a part of it.

Figure 14 Searching documents having location ‘Delhi’ which is sub-region of ‘Asia’ (see online version for colours)



Patterns:
Person hasPosition JobTitle withinOrganization Organization
Organization locatedIn Location
Organization locatedIn Location1 subRegionOf Location2
Person hasPosition JobPosition
Location1 subRegionOf Location2
Company activeInSector IndustrySector

Location1: Delhi

sub region of Location2: Asia

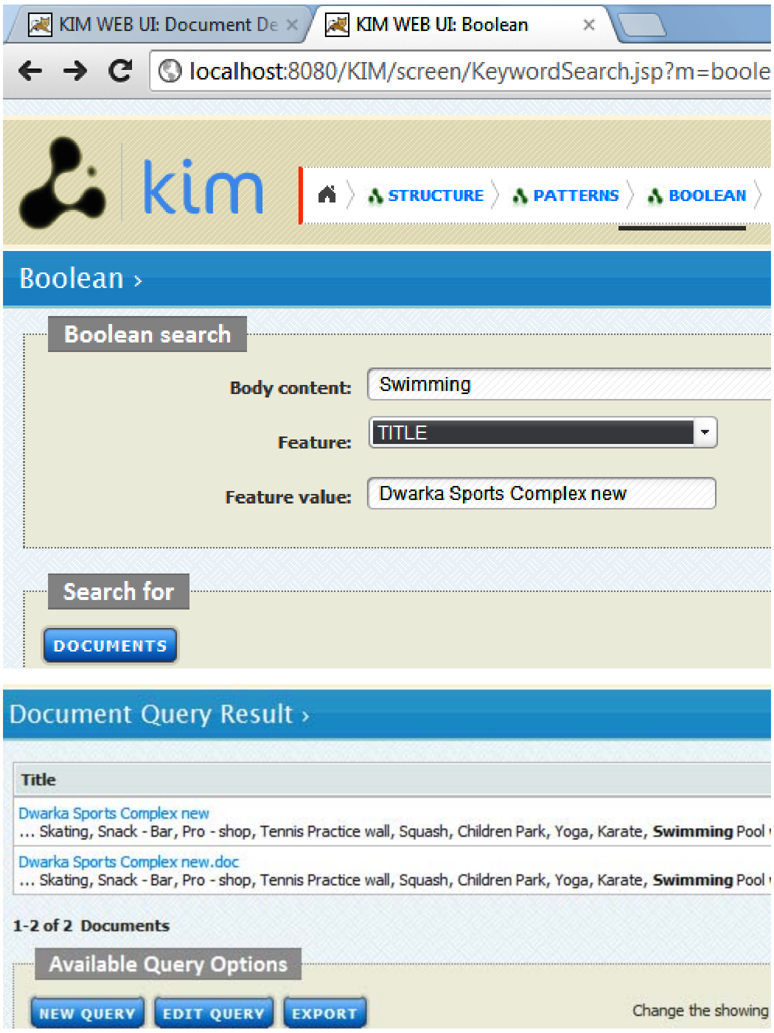
Patterns: Location1, Location2

Search based on ‘Boolean’

It allows to combine words and phrases using logical (Boolean) operators (such as AND, OR, etc.) to limit, widen, or define our search.

The search operation for the documents which consists of body content as ‘Swimming’ and the Title of the document as ‘Dwarka Sports Complex new’ which results in the search operation based on Boolean method which clearly shows the document fulfilling the search criteria.

Figure 15 Search based on 'Boolean' (see online version for colours)



Body content: Swimming

Feature: TITLE

Feature value: Dwarka Sports Complex new

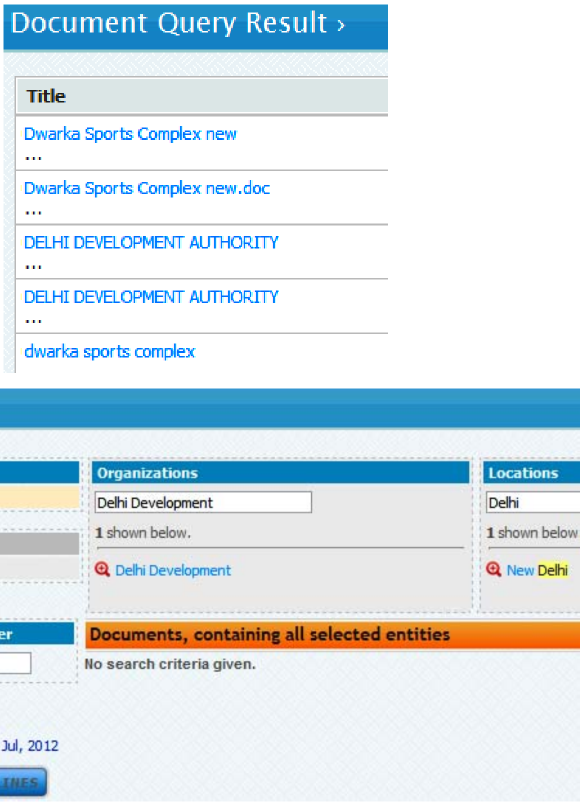
Title
Dwarka Sports Complex new … Skating, Snack - Bar, Pro - shop, Tennis Practice wall, Squash, Children Park, Yoga, Karate, **Swimming** Pool
Dwarka Sports Complex new.doc … Skating, Snack - Bar, Pro - shop, Tennis Practice wall, Squash, Children Park, Yoga, Karate, **Swimming** Pool

1-2 of 2 Documents

Search based on 'facets'

It offers an advanced search by tracking associative relations between entities, based on their co-occurrence in the same context. Its scalability allows to search quickly and effectively through an extensive set of documents. KIM web interface offers facet based search for entities of three classes viz.; 'People', 'Organization' and 'Location'.

Figure 16 Search based on 'facets' (see online version for colours)



Document Query Result ›

Title

Dwarka Sports Complex new
...

Dwarka Sports Complex new.doc
...

DELHI DEVELOPMENT AUTHORITY
...

DELHI DEVELOPMENT AUTHORITY
...

dwarka sports complex

Facets ›

Selected Items
(No items selected)

Recent Items
(No recent items)

Organizations
Delhi Development
1 shown below.
Delhi Development

Locations
Delhi
1 shown below
New Delhi

Document Keyword Filter

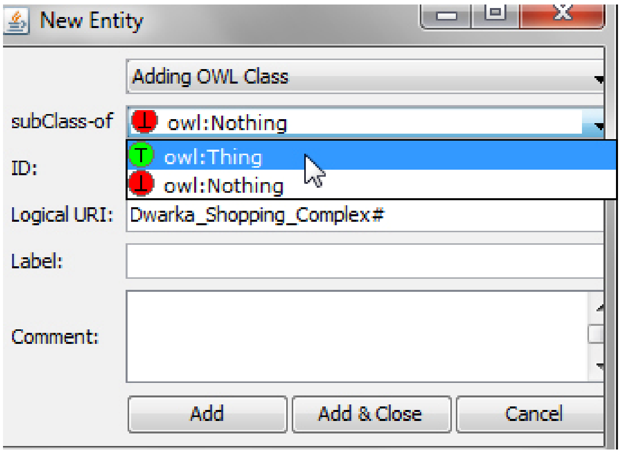
Documents, containing all selected entities
No search criteria given.

Matching documents: 103
Between: 25 Feb, 2001 - 09 Jul, 2012

4 Case study 2: design and development of a new project ontology in SWOOP: 'Dwarka Shopping Complex Ontology'

Here, We illustrate the 'Dwarka Shopping Complex Ontology' using SWOOP 2.3 Beta 4 Ontology Editor to present various aspects like GCI (General Concept Inclusion Axiom) by using classes, properties and instances along with query retrieval by options of 'fly the mothership' option with output as in Venn diagram and 'look up query'. We also present ontology statistics and observation with class expression table. SWOOP is an ontology browser and editor, designed specifically for use with web ontology language. It meant for rapid and easy browsing and development of web ontologies. It creates primary identifiers for ontologies, classes, properties, and individuals to support hyper textual navigation through, and between ontologies. It presents various features like ontology statistics, observation table and class expression table. By selecting 'New Ontology' option, we start creating the new ontology in SWOOP 2.3 Beta Ontology Editor.

Figure 17 Starting to create a superclass of the ontology (see online version for colours)



Adding OWL Class

subClass-of: owl:Nothing

owl:Thing

owl:Nothing

ID:

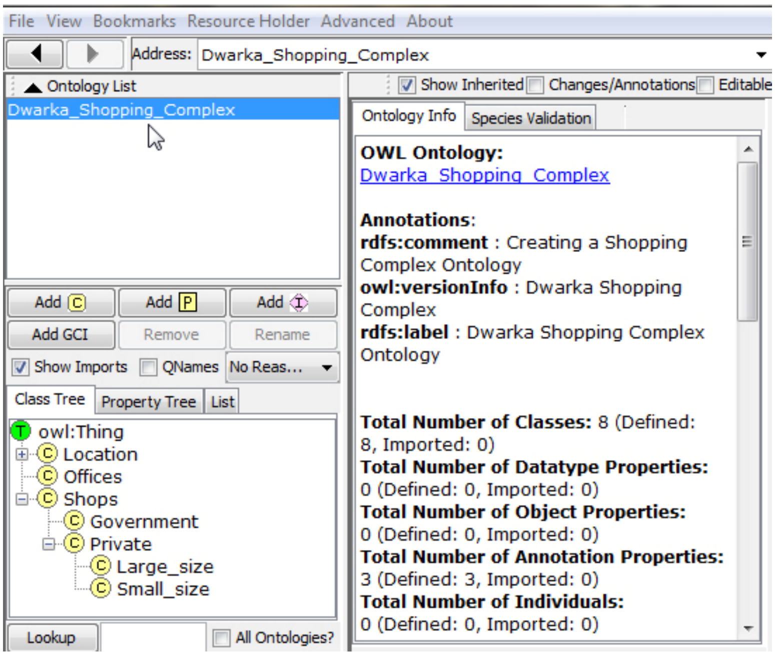
Logical URI: Dwarka_Shopping_Complex#

Label:

Comment:

Selecting 'ADD C' option of SWOOP editor, the new entity screen appears as above which consists of 'subclasss-of', 'ID' and 'Logical URI'. In option, 'subclass-of', there will be two superclasses: owl:Thing (symbol T in green circle) for creating classes within ontology and owl:Nothing (symbol as reverse T in red circle) for creating classes outside the said ontology. The ontology is presented as below showing a few classes and subclasses.

Figure 18 Super class/sub class hierarchy of 'Shopping Complex Ontology' (see online version for colours)



Address: Dwarka_Shopping_Complex

Dwarka_Shopping_Complex

☑ Show Inherited □ Changes/Annotations □ Editable

☑ Show Imports □ QNames No Reas...

□ All Ontologies?

- owl:Thing
  - Location
  - Offices
  - Shops
    - Government
    - Private
      - Large_size
      - Small_size

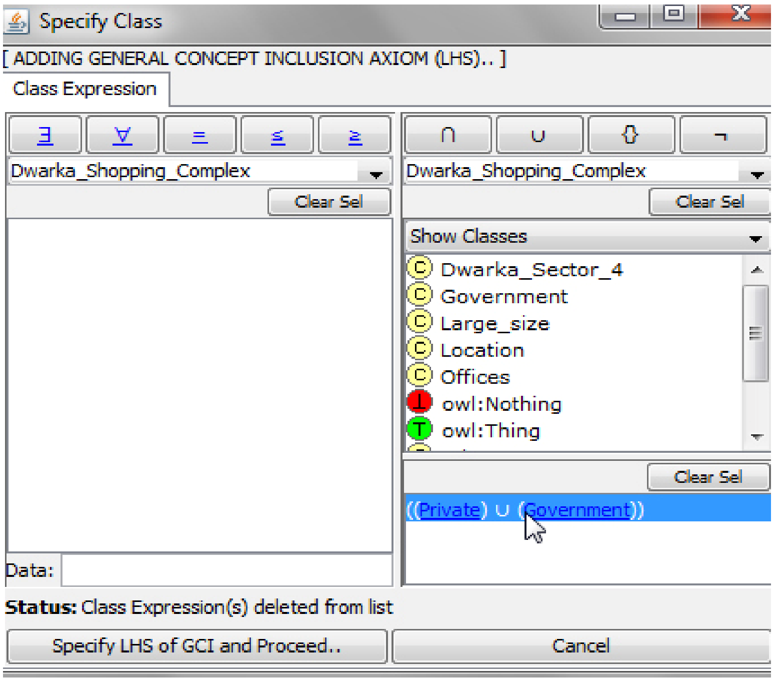
OWL Ontology:
Dwarka_Shopping_Complex

Annotations:
rdfs:comment : Creating a Shopping Complex Ontology
owl:versionInfo : Dwarka Shopping Complex
rdfs:label : Dwarka Shopping Complex Ontology

Total Number of Classes: 8 (Defined: 8, Imported: 0)
Total Number of Datatype Properties: 0 (Defined: 0, Imported: 0)
Total Number of Object Properties: 0 (Defined: 0, Imported: 0)
Total Number of Annotation Properties: 3 (Defined: 3, Imported: 0)
Total Number of Individuals: 0 (Defined: 0, Imported: 0)

Adding instances, specifying GCI axioms (LHS and RHS)

Figure 19 Adding instances, specifying GCI Axioms (LHS and RHS) (see online version for colours)



Specify Class

[ADDING GENERAL CONCEPT INCLUSION AXIOM (LHS)..]

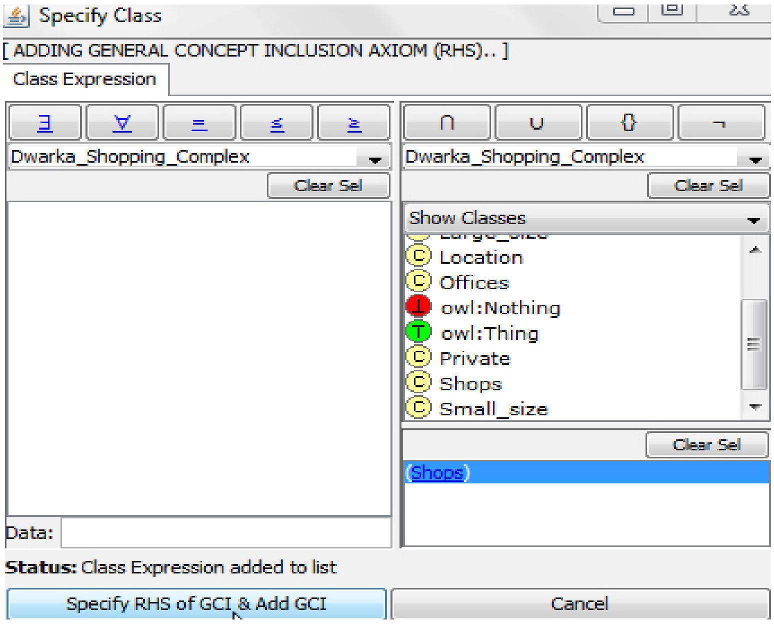
Dwarka_Shopping_Complex

((Private) ∪ (Government))

Data:

Status: Class Expression(s) deleted from list

Figure 20 Specifying LHS and RHS of GCI (see online version for colours)



Specify Class

[ADDING GENERAL CONCEPT INCLUSION AXIOM (RHS)..]

Dwarka_Shopping_Complex

(Shops)

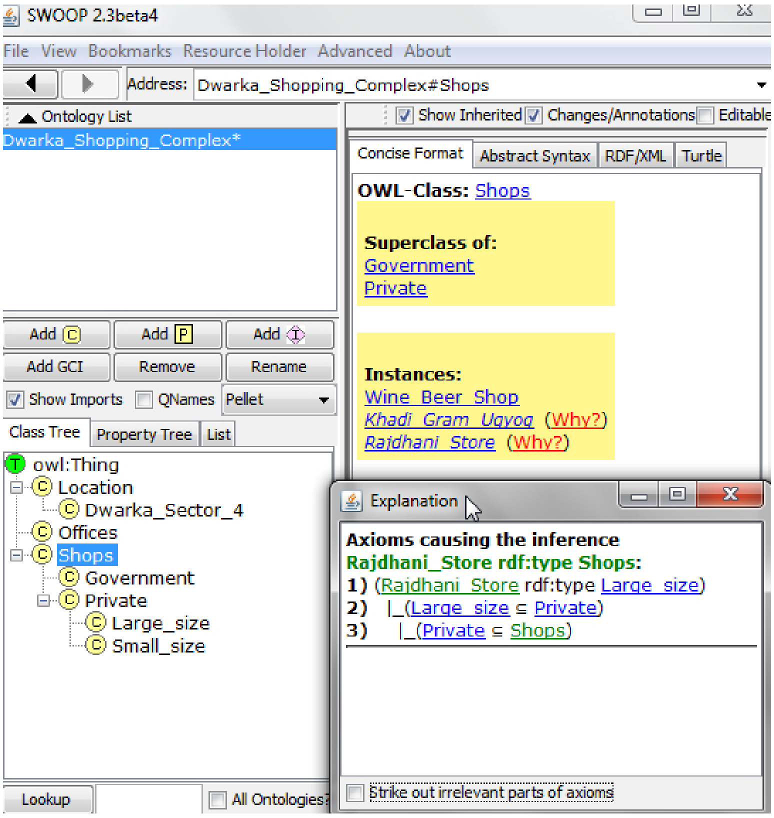
Data:

Status: Class Expression added to list

We can add instances by using ‘ADD I’ option in the SWOOP ontology editor (The class in which the instance is to be created may be selected from ‘subclasses-of’ option of ‘New Entity’.). Then, the name of instance is entered in ‘ID’ text box. For illustration we have created the instance ‘Wine_Beer_Shop’, ‘Khadi_Udyog_Gram’ and ‘Rajdhani Store’ in ‘Shop’, ‘Government’, and ‘Private’ classes respectively (can be observed in Figure 21). For adding GCI axioms in ‘Instances’, ‘ADD GCI’ option is used and select the relative classes ‘Shop’, ‘Government’ and ‘Private’ and perform union operation between ‘Private’ and ‘Government’ which is subset of class ‘Shop’ in the ontology. Following results/observations are obtained.

Observations of instances

Figure 21 Observations of instances (see online version for colours)



Address: Dwarka_Shopping_Complex#Shops

☑ Show Inherited ☑ Changes/Annotations □ Editable

☑ Show Imports □ QNames Pellet

□ All Ontologies?

OWL-Class: Shops

Superclass of:
Government
Private

Instances:
Wine_Beer_Shop
Khadi_Gram_Ugyog (Why?)
Rajdhani_Store (Why?)

Axioms causing the inference
Rajdhani_Store rdf:type Shops:
1) (Rajdhani_Store rdf:type Large_size)
2) |_(Large_size ⊆ Private)
3) |_(Private ⊆ Shops)

□ Strike out irrelevant parts of axioms

Instances 'Without GCI axioms' and 'With GCI axioms'

Table 3 Instances without GCI axioms and with GCI axioms

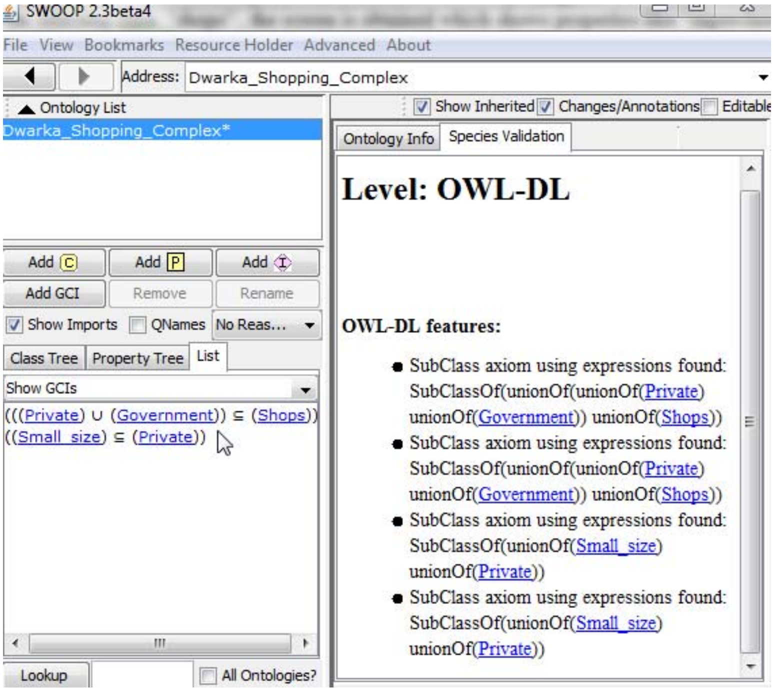
Class name	*Instances name*	*Class hierarchy*
Shops	Wine Beer Shop	Super class of 'Government' and 'Private', sub-class of 'Dwarka_Shopping_Complex'.
Government	Khadi Gram Ugyog	Sub class of 'Shops'
Private	Rajdhani Store	Super class of 'Large_size' and 'Small_size', and sub class of 'Shops'.

On selecting class 'shops', the screen is obtained which shows properties like 'superclass' and 'instances'. On clicking, 'Rajdhani Store (Why?)', 'Explanation' screen is obtained showing 'GCI axioms causing the inference'.

Display list

On selecting 'List' option and 'Show GCIs' options, various GCI axioms are displayed as shown in Figure 21.

Figure 22 Display list with 'Show GCIs' (see online version for colours)



(((Private) ∪ (Government)) ⊆ (Shops))
((Small_size) ⊆ (Private))

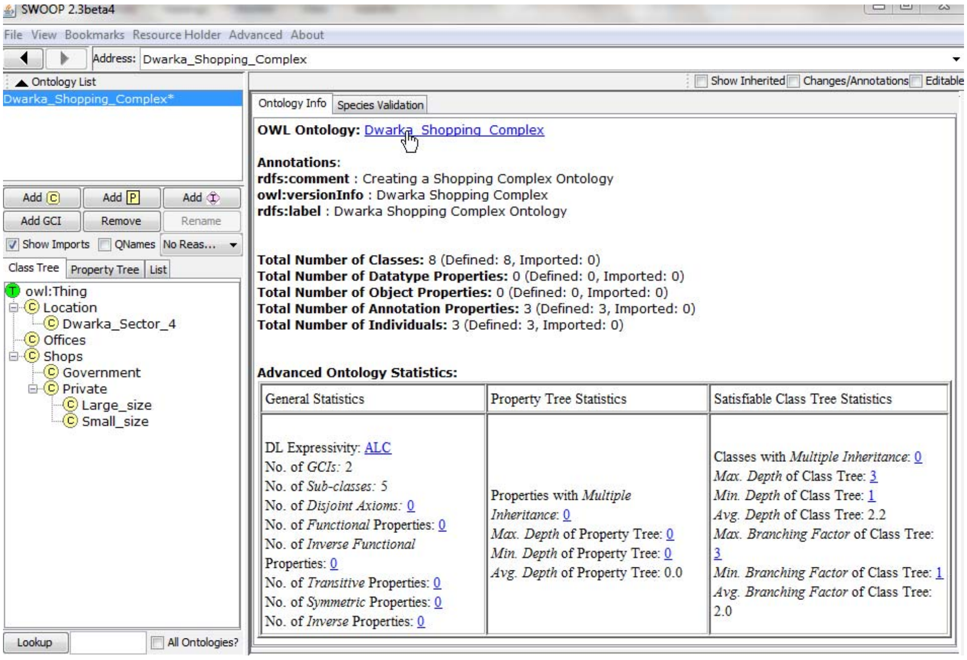
Level: OWL-DL

OWL-DL features:

- SubClass axiom using expressions found: SubClassOf(unionOf(unionOf(Private) unionOf(Government)) unionOf(Shops))
- SubClass axiom using expressions found: SubClassOf(unionOf(unionOf(Private) unionOf(Government)) unionOf(Shops))
- SubClass axiom using expressions found: SubClassOf(unionOf(Small_size) unionOf(Private))
- SubClass axiom using expressions found: SubClassOf(unionOf(Small_size) unionOf(Private))

Ontology statistics

Figure 23 Ontology statistics (see online version for colours)



OWL Ontology: Dwarka_Shopping_Complex

Annotations:
rdfs:comment : Creating a Shopping Complex Ontology
owl:versionInfo : Dwarka Shopping Complex
rdfs:label : Dwarka Shopping Complex Ontology

Total Number of Classes: 8 (Defined: 8, Imported: 0)
Total Number of Datatype Properties: 0 (Defined: 0, Imported: 0)
Total Number of Object Properties: 0 (Defined: 0, Imported: 0)
Total Number of Annotation Properties: 3 (Defined: 3, Imported: 0)
Total Number of Individuals: 3 (Defined: 3, Imported: 0)

Advanced Ontology Statistics:

General Statistics	Property Tree Statistics	Satisfiable Class Tree Statistics
DL Expressivity: ALC No. of *GCIs:* 2 No. of *Sub-classes:* 5 No. of *Disjoint Axioms:* 0 No. of *Functional* Properties: 0 No. of *Inverse Functional* Properties: 0 No. of *Transitive* Properties: 0 No. of *Symmetric* Properties: 0 No. of *Inverse* Properties: 0	Properties with *Multiple Inheritance:* 0 *Max. Depth* of Property Tree: 0 *Min. Depth* of Property Tree: 0 *Avg. Depth* of Property Tree: 0.0	Classes with *Multiple Inheritance:* 0 *Max. Depth* of Class Tree: 3 *Min. Depth* of Class Tree: 1 *Avg. Depth* of Class Tree: 2.2 *Max. Branching Factor* of Class Tree: 3 *Min. Branching Factor* of Class Tree: 1 *Avg. Branching Factor* of Class Tree: 2.0

Observation table

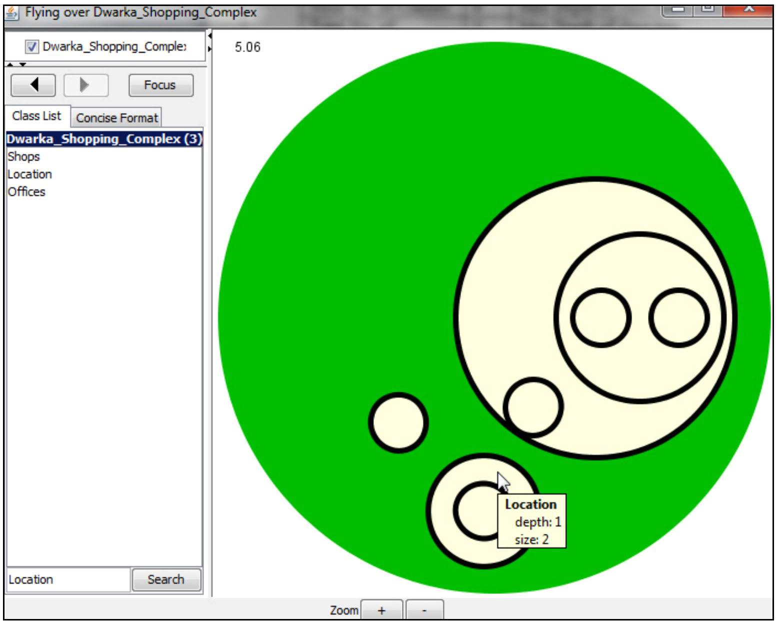
Table 4 Ontoloy statistics observation table

Total no. of classes	8	Max depth of class tree	3
Total no. of annotation properties	3	Max depth of class tree	1(always 1)
Total no. of individuals	3	Avg. depth of class tree	2.2
No. of GCI's	2	Max branching factor of class tree	3
No. of subclasses	5	Avg. branching factor of class tree	2

Query retrieval process: 'Fly The Mothership' option

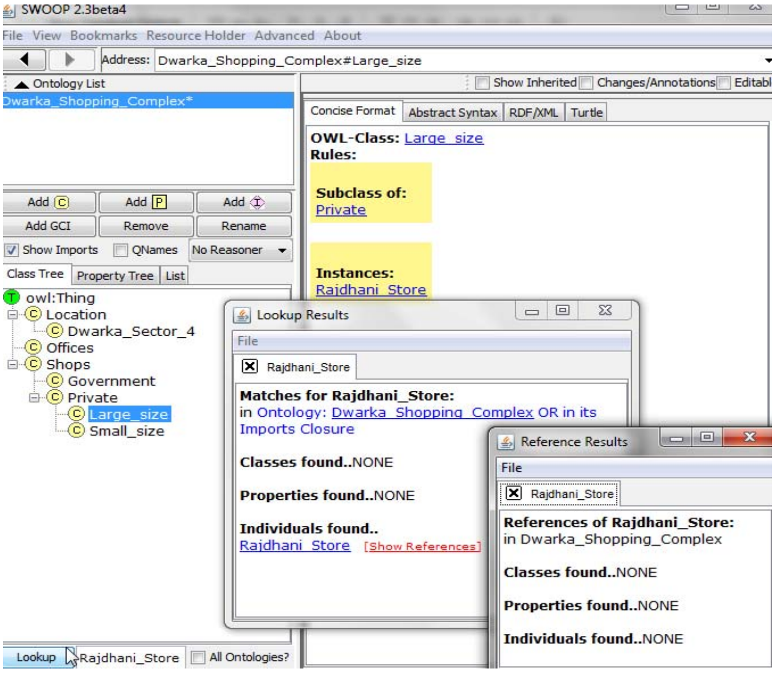
Now we consider the query retrieval process to show that how we run the query using the 'Fly The Mothership' option of 'Advanced' tab as shown in Figure 23. Here, class 'Location' is searched and output appears in a form of Venn diagram where cursor pointing to the class 'Location' and also displaying 'Depth and Size' of the class.

Figure 24 ‘Fly The Mothership’ option (see online version for colours)



Query retrieval process: ‘Lookup’ option

Figure 25 Searching instance by using ‘Lookup’ option in SWOOP ontology editor (see online version for colours)



Address: Dwarka_Shopping_Complex#Large_size

OWL-Class: Large_size
Rules:

Subclass of:
Private

Instances:
Rajdhani_Store

Matches for Rajdhani_Store:
in Ontology: Dwarka_Shopping_Complex OR in its Imports Closure

Classes found..NONE

Properties found..NONE

Individuals found..
Rajdhani_Store [Show References]

References of Rajdhani_Store:
in Dwarka_Shopping_Complex

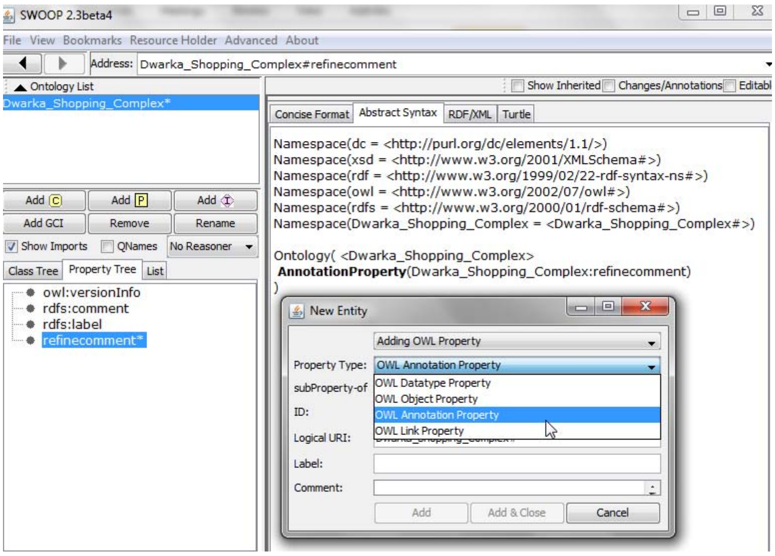
Classes found..NONE

Properties found..NONE

Individuals found..NONE

Adding property

Figure 26 Types of properties (see online version for colours)



```
Namespace(dc = <http://purl.org/dc/elements/1.1/>)
Namespace(xsd = <http://www.w3.org/2001/XMLSchema#>)
Namespace(rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Namespace(owl = <http://www.w3.org/2002/07/owl#>)
Namespace(rdfs = <http://www.w3.org/2000/01/rdf-schema#>)
Namespace(Dwarka_Shopping_Complex = <Dwarka_Shopping_Complex#>)

Ontology( <Dwarka_Shopping_Complex>
 AnnotationProperty(Dwarka_Shopping_Complex:refinecomment)
)
```

Adding OWL Property

Property Type: OWL Annotation Property

- OWL Datatype Property
- OWL Object Property
- OWL Annotation Property
- OWL Link Property

subProperty-of

ID:

Logical URI:

Label:

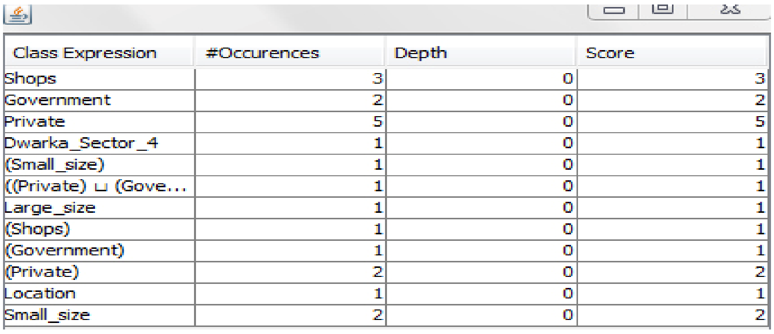
Comment:

In Figure 25, for searching instance by ‘lookup’, type the name of instance in text box of ‘lookup’, then it will display the ‘Lookup Result’ panel which is consist of result. It commonly shows the result in four ways, i.e., ‘Matches for’, classes found, properties found, individuals found. For given illustration, instance ‘Rajdhani_Store’ is searched and Individuals result is found. It is also shows ‘References of Rajdhani_Store’. In Figure 26, by selecting add P option, new entity panel is opened which display four types of properties like: data type, object, annotation and link property. A property, ‘refinecomment’ is defined which is a annotation type property.

Class expression table

To display class expression table using the ‘Show Class Expression Table’ option of ‘Advanced’ tab as shown in SWOOP ontology editor.

Figure 27 Displaying the class expression using ‘Show Class Expression Table’ sub-option of ‘Advanced’ tab (see online version for colours)



Class Expression	#Occurences	Depth	Score
Shops	3	0	3
Government	2	0	2
Private	5	0	5
Dwarka_Sector_4	1	0	1
(Small_size)	1	0	1
((Private) ⊔ (Gove…	1	0	1
Large_size	1	0	1
(Shops)	1	0	1
(Government)	1	0	1
(Private)	2	0	2
Location	1	0	1
Small_size	2	0	2

On selecting this option, it will show the table with the column class expression, occurrence, depth, and score. In the row all the classes will be listed with their respective values. Now, if we select one of the classes then it will also display the particular relation of class with respect to others.

Code snippet

The code snippet is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
 <!ENTITY Dwarka_Shopping_Complex "Dwarka_Shopping_Complex#">
 <rdf:RDF xml:base="Dwarka_Shopping_Complex"
     xmlns:Dwarka_Shopping_Complex="&Dwarka_Shopping_Complex;"
     xmlns:owl="&owl;"
     xmlns:rdf="&rdf;"
     xmlns:rdfs="&rdfs;">
 <owl:Ontology rdf:about=""
        rdfs:comment="Creating a Shopping Complex Ontology"
        rdfs:label="Dwarka Shopping Complex Ontology"
        owl:versionInfo="Dwarka Shopping Complex"/>
 <owl:Class rdf:about="#Dwarka_Sector_4">
   <rdfs:subClassOf rdf:resource="#Location"/>
 </owl:Class>
```

4 Conclusions and future work

The significance of the paper is developing ontologies (case studies of two ontologies) with various enhanced features and extended version in most widely used ontology editors Protégé and Swoop (Cardoso survey), presenting an ontology flowchart and annotation search patterns based on ‘pattern’, ‘Boolean’, and ‘facets’ using KIM. Based on LEE’s architecture of Semantic Web, various significant issues towards an intelligent web are summarised and Cardoso survey (2007) highlighting highest usage of ontology editors is presented to justify the ontology editors chosen. Two case studies have been developed, ‘Sports Complex Ontology’ and ‘Shopping Complex Ontology’. The case study on a ‘Sports Complex Ontology’ is an enhanced version developed by ontology editor, SWOOP 2.3 Beta 4, extended to various results and observations highlighting Venn diagram of class hierarchy, fly the mothership, pellet query, instance search, ontology partition summary, adding annotation property etc and then it is also developed in another ontology editor, Protégé 3.4.4 for various observations presenting a flowchart for the said ontology. The work is appended with various semantic annotation search patterns based on 'pattern', 'Boolean', and 'facets' have been presented using KIM platform. For the purpose of future comparisons and other objectives, a second case study illustration of a 'Dwarka Shopping Complex Ontology' is also presented using SWOOP 2.3 Beta 4 Editor highlighting LHS and RHS of GCI, instances with and without GCI axioms, query retrieval look up, adding property, class expression table and ontology statistics, etc. Various other concerned options needs to be explored and worked in more depth towards the vision of an ontology for an intelligent web. The work may serve as a good reference for researchers and students to start developing ontologies and in future, the work may be further extended by exploring more or better options towards ontology development and making a comparative study.

References

Berners-Lee, T., Godel, K. and Turing, A. (2006) *Thinking on the Web*, Wiley-Interscience, John Wiley & Sons, pp.xv, xxvi, 108,14,127,128.

Berners-Lee, T., Handler, J. and Lassila, O. (2001) 'The semantic web', *Scientific American Magazine*, 17 May.

Cardoso, J. (2007) 'The semantic web vision: where are we?', *IEEE Intelligent Systems*, September/October, pp.22–26.

Chandrasekran, B., Josephson, J.R. and Benjamins, V.R. (1999) 'What are ontologies and why do we need them', *IEEE Intelligent Systems*, January–February, Vol. 14, No. 1, p.1.

Davies, J., Studer, R. and Warren, P. (2006) *Semantic Web Technologies, Trends and Research in Ontology-Based Systems*, John Wiley & Sons, pp.29, 36, 42, 43, 44, 45.

Gruber, T. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*.

Hebler, J., Fisher, M., Blace, R. and Perez-Lopez, A. (2009) *Semantic Web Programming*, p.xxv, Wiley.

Kiryako, A., Popov, B., Manov, D., Ognyanof, D., Marinov, R. and Terziev, I. (2004) 'Automatic semantic annotation with KIM', *ISWC 2004* [online] http://iswc2004.semanticweb.org/demos/17/paper.pdf.

Noy, N.F. and McGuinness, D.L. (2000) *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford University, CA, 94305.

Passin, T. (2004) 'Explorer's guide to semantic web', *Manning*, p.18.

Shadbolt, N., Hall, W. and Berners-Lee, T. (2006) 'The semantic web revisited', *IEEE Intelligent Systems*, IEEE Computer Society.

Taniar, D. and Rahayu, J.W. (2006) *Web Semantics and Ontology*, p.viii, Idea Group.