ISSN:0975-9646

SEMANTIC WEB AND COMPARATIVE ANALYSIS OF INFERENCE ENGINES

Ms. Neha Dalwadi[1], Prof. Bhaumik Nagar[2], Prof. Ashwin Makwana[1]

[1] Computer Engineering, Chandubhai S Patel Institute of Technology
Changa, Dist. Anand, India
[2] National Institute of Design
Paldi, Ahmedabad, India

***Abstract*— Semantic Web is an emerging technology for efficient reasoning support over the knowledge represented on the Web. This paper presents the semantic web standards and survey a number of Inference Engines that supports reasoning with OWL. Also analyzed the reasoner with set of ontologies and based on supported features.**

***Keywords*— OWL, RDF, Inference, Description Logic**

I. INTRODUCTION

Semantic Web is an emerging technology regarded as the next generation Web paradigm providing machine-understandable information that is based on meaning. Motivate creation of new technologies and standards to analyze and understand large amount of data on web and infer new knowledge. Semantic Web is not a separate Web but an extension of the current one, in which information is given well defined meaning, better enabling computers and people to work in cooperation (Berners-Lee et al 2001). It is a web of data, where data should be related to one another just like documents in the web. Used in data integration, classification, resource discovery, search algorithm. It makes Web more collaborative medium, more understandable and thus processable by machines. If you have a lot of information, there are implied and hidden relationships in your data. Today's Web is based on documents written in Hypertext Mark-up Language (HTML) that is used for coding a body of text with images and interactive forms. HTML can make document-level assertions such as layout details like title of document, span of text etc. The Semantic Web takes the solution, involves languages such as Resource Description Framework (RDF), Web Ontology Language (OWL) and Extensible Mark-Up Language (XML) specifically designed for data that can describe arbitrary things such as people, electronic parts, animals etc. To visualize semantic web, for example there are two stories with different syntax and synonyms but same meanings (semantics). Though these two stories are same, current web cannot find it but using semantic web technologies we can get the satisfactory results. Because current web compares syntax structure only, where as semantic web finds semantics and compare it.

II. SEMANTIC WEB STANDARDS

A. RDF (Resource Description Framework)

Semantic Web project is started by W3C with the purpose of realizing the data on the web defined and linked in a way that it can be used by machines not just for display but for automation, integration and reuse of data across various applications. RDF provides information representational framework that makes statements about web resources in machine understandable format. RDF references are labelled, to indicate kind of relationships that exists between linked data. RDF statement model contains 3 parts: Subject that identifies things/concepts, Predicate that identifies property/characteristics of the subject and Object that identifies value of that property.

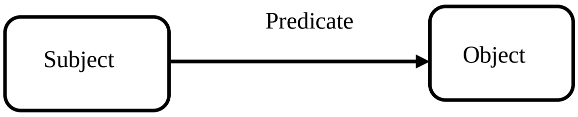

Fig. 1 RDF graph structure

For example: Neha is the creator of http://www.cs.citc.edu/~Neha. Here, subject is of http://www.cs.citc.edu/~Neha, Object is Neha and property is Creator.

RDF has adopted syntax of XML and inherits benefits of XML. XML does not describe any means of semantics of data as there is no intended meaning associated with the nested tags; it is up to each application to interpret the nesting. RDF has same type of structure as XML but it is different in a way that RDF allows assigning global identifiers to information resources and allows one resource document to refer to and extend statements made in other resource documents.

B. RDFS (Resource Description Framework Schema)

RDF is a universal language that allows users to describe resources using their own vocabularies. It does not make assumptions about any particular domain and does not define semantics of any domain. This can be done using RDFS. RDFS allows the representation of classes, class hierarchies, properties, property hierarchies, and domain and range restrictions on properties [1].

RDF Schema is based on RDF. RDF Schema can be viewed as a primitive language for writing ontologies. But there is a need for more powerful ontology languages that expand RDF Schema and allow the representations of more complex relationships between Web objects. Statements written in a language such as RDF define the relations between concepts and specify logical rules for reasoning about them. Computers will “understand” the meaning of semantic data on a Web page by following links to specified ontologies.

C. OWL (Web Ontology Language)

Ontology is an approach for knowledge representation which describes basic concepts and relationships among them. Ontology makes metadata interoperable and ready for efficient sharing and reuse. It provides shared and common understanding of a domain that can be used both by people and machines. Ontology helps in data integration when more ambiguity in datasets. Ontology is important for the purpose of enabling knowledge sharing and reuse. RDF Schema and OWL provide languages to express Ontology. OWL is an ontology language that extends expressiveness of RDFS. There is a need to develop ontology to share reuse common understanding of the structure of information and domain knowledge.

Components of Ontology are: Concepts: i.e. human, animal, movie, picture etc, Instances, Properties, Relations, and Rules. Small example of wine ontology expressed in OWL as follows:

```
<owl:Class rdf:ID="Wine">

<rdfs:subClassOf     rdf:resource="&food;PotableLiquid"/>

 <rdfs:subClassOf>

   <owl:Restriction>

     <owl:onProperty rdf:resource="#madeFromGrape"/>

     <owl:minCardinality
rdf:datatype="&xsd;NonNegativeInteger">1</owl:minCar
dinality>

   </owl:Restriction>

  </rdfs:subClassOf>

  <rdfs:subClassOf>

   <owl:Restriction>

     <owl:onProperty rdf:resource="#locatedIn"/>

     <owl:minCardinality
rdf:datatype="&xsd;NonNegativeInteger">1</owl:minCar
dinality>

   </owl:Restriction>

  </rdfs:subClassOf>

</owl:Class>
```

OWL allows support for a well-defined syntax, efficient reasoning support, a formal semantics sufficient expressive power, and convenience of expression. OWL itself is further classified as: OWL Lite and OWL DL that are based on Description Logics [2]; OWL Full combines the flexibility of RDF(S) with the expressiveness of OWL DL. Because of its high expressiveness, reasoning in OWL Full is generally hard, and the most important reasoning tasks for the language are un-decidable [3]. Furthermore, the formalism is non-standard with respect to existing knowledge representation and database languages; therefore, most Semantic Web research related to OWL is limited to the OWL Lite and OWL DL (see [3]).

III. OVERVIEW OF INFERENCE

Inference on semantic web is defined as discovering new relationships between resources. Inference is a process that can generate new relationships based on additional information in the form of vocabulary or rule set. It is used to improve the quality of data integration on the Web, by discovering new relationships, automatically analyzing the content of data, or managing knowledge on the Web. Inference based techniques are important in discovering possible inconsistencies in the integrated data.

A semantic reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. An inference engine is needed for the processing of the knowledge encoded in the semantic web language OWL.

IV. MOTIVATION

Motivations for choosing inference engine for ontology are the various challenges faced by ontology reasoning in semantic web: 1.There is a need for a data representation to enable software products (agents) to provide intelligent access to heterogeneous and distributed information. 2. To provide tools that drive down the cost of establishing interoperability between different data providers. 3. Reasoning rules play a primary role in the reasoning process to retrieve useful knowledge on the Semantic Web. In maintaining Semantic Web, one of the key issues is to keep rule set consistent and highly efficient. From the survey of existing inference engines it was found that some of them can only support reasoning over a fixed a format of data representation like OWL-DL, OWL-Lite, RDF and

Triplet. Some of them cannot handle large size of ontology database.

V. APPROACHES USED FOR OWL INFERENCE ENGINE

Approach 1: Description Logic based Inference Engine

This is the most widely used approach to design OWL-DL reasoner. Because OWL is originated from Description Logic, it is easy to apply the concept of DL reasoner with OWL-DL. They are used to perform basic reasoning tasks like consistency checking and subsumption concepts. It has the advantage of using decidability. Semantic reasoners FaCT [Horrocks 1999], FaCT++, Racer [Haarslev 2001] and Pellet are DL based reasoner and implements different types of Description Logic.

Approach 2: First Order Logic (FOL) theorem prover based inference engine

This approach works on First Order Logic. Reasoner takes OWL file as input and first translated into FOL. Then inference is processed by using any one of existing automated theorem prover. Semantic reasoners like Hoolet uses Vampire theorem prover, Surnia uses Otter theorem prover are FOL based reasoner.

Approach 3: Combination of FOL and general Logic based inference engine

In this approach fragment of FOL and general logic is used to design OWL inference engine. Horn logic is most widely used due to its simplicity and availability. Semantic reasoners like JESS, JENA and F-OWL are based on this combine approach. F-OWL uses F –logic and XSB for implementation.

VI. SEMANTIC REASONER SERVICES

The standard DL reasoning services are used for OWL reasoning to explore new facts and conclusion about the knowledge base. Knowledge base consists of two parts TBox – where OWL concept ontologies are defined and ABox- where OWL instances are defined.

Common reasoning services applied on TBox and ABox reasoning are as follows:

- Consistency Checking: Consistency checks whether ABox instances are consistent with the TBox concepts and ensures that ABox meet all of the restrictions.

- Satisfiability Checking: It checks OWL concept C can have instances according to current ontology. That is ABox instances should be satisfied by available OWL concepts of TBox.

- Subsumption Checking: It checks whether a class D subsumes another class C. That is instances or properties of class C is also part of class D.

- Query Processing: OWL inference engines need powerful language to support queries so that users and software agents can query on knowledge base to retrieve useful data or facts.

- Reasoning with Rules: Because the rules are capable of expressing the OWL classes, properties and instances, OWL inference engine needs to provide interface to process rules that represented with OWL classes, properties and instance data.

VII. INFERENCE ENGINES AND COMPARISONS

In Comparative analysis include following Inference Engines and shown in Table I:

A. FaCT and FaCT++

FaCT++ [6] an improved version of FaCT [7] employs tableaux algorithms for SHOIQ description logic and implemented in C++ but has very limited user interface and services as compared to other reasoners. The strategies followed are absorption, model merging, told cycle elimination, synonym replacement, ordering heuristics and taxonomic classification.

B. Pellet

Pellet employs reasoning on SHIN (D) and SHON (D) and implemented in Java with the strategies of TBox partitioning, nominal support, absorption, semantic branching, lazy unfolding, and dependency directed back jumping [4]. Datatype reasoning, individual reasoning, and optimization in ABox query answering makes it more attractive for sound semantic web applications. It provides standard reasoning services for OWL ontologies. It incorporates various optimization techniques including optimization for nominal, conjunctive query answering and incremental reasoning.

C. Racer

Racer was implemented in Lisp to demonstrate the tableaux calculus for SHIQ, and follows the multiple optimization strategies for better reasoning support including dependency-directed backtracking, transformation of axioms, model caching and merging, etc, [5].

D. Hoolet

OWL-DL reasoner uses first order logic. The ontology is translated to a collection of axioms and is then given to first order logic for consistency checking [8]. It is extended to handle the rules through the addition of parser for RDF rule syntax [9]. Hoolet is based on OWL-DL reasoner with support of SWRL rules. It uses naïve approach that is classified under homogeneous translation based approach for implementation. Reasoning support is straightforward

TABLE I: COMPARATIVE ANALYSIS OF INFERENCE ENGINE

	FACT	**FACT++**	**PELLET**	**RACER**	**HOOLET**	**F-OWL**	**HERMIT**	**KAON2**
Availability	Free (Open source)	Free (Open \Source)	Free (Open Source)	Non Free (Closed Source)	Free (Open Source)	Free (Open Source)	Free (Open Source)	Free (Open Source)
Platform	Windows /Linux	Windows /Linux	Windows	Windows	Linux	Windows	Windows	Windows/ Linux
OWL Support	OWL-DL	OWL – DL	OWL- DL	OWL-DL	OWL-DL	OWL-Full	OWL – DL	OWL – DL
Logic	SHIQ(D) (DL)	SHIN(D) (DL)	Combination of SHIF(D) and SHON(D) (DL)	SHIQ(D) (DL)	First Order Logic	Horn Logic, Frame Logic	SHIQ(D) DL	Horn logic, SHIQ(D) (DL)
Reasoning Algorithm	Tableau	Tableau	Tableau	Tableau	First Order Prover	Tableau	Hyper-Tableau	First Order Resolution Calculus
Consistency Checking	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
ABox Reasoning	No	No	Yes	Yes	-	Yes	Yes	Yes (Without Nominals)
Interface	DIG, Command Line	DIG, Command Line	DIG, JAVA	DIG, JAVA, GUI	JAVA	JAVA, GUI, Command Line	Command Line	JAVA, GUI, Command Line
Query Support	-	-	RDQL, SPARQL	Racer Query Language	-	Frame Style, RDQL	-	SPARQL
Supported Rule Language	No support	No support	SWRL – DL safe rules	SWRL-Not fully support	SWRL	SWRL	SWRL – DL safe rules	SWRL – DL safe rules

translation of the ontology into a collection of axioms which is communicated to the FOL based theorem prover Vampire for consistency checking. Approach used is not scalable and FOL is undecidable

E. F-OWL

F-OWL is the OWL inference engine uses frame based system to reason with OWL ontologies. F-OWL reads OWL ontology from URI and extracts RDF triples out of the ontology which are further converted into format appropriate for F-OWL's frame style and fed into the engine. It uses flora rules defined in flora-2 language to check the consistency of the ontology and extract hidden knowledge [10].

F. Hermit

Hermit reasoner employs reasoning on SHIQ(D). It is available free for non-commercial usage. Takes OWL file as input and perform various reasoning tasks like consistency checking, identify subsumption relationships between classes and more. It also computes partial order of classes occurring in OWL. It is different from other reasoner like Pellet and FaCT such a way that it implements hyper-tableau reasoning algorithm that is much less deterministic than existing tableau algorithm.

G. KaON2

It is based on OWL-DL and Frame Logic. It is an infrastructure of managing OWL-DL, SWRL and F-Logic ontologies. It supports answering conjunctive queries, although without true non-distinguished variables. KAON2 is a successor of KAON project used extension of RDFS [11]. The main idea behind its implementation is to reduce SHIQ knowledge based to disjunctive datalog program with finite DL-safe rules, reason with hybrid logic by reusing database optimization techniques. KAON2 translates ontology and a rule into axioms in a common logic language due to that it is classified under homogeneous translation based approach. KAON2 currently cannot handle nominals. If an ontology contains an owl: oneOf class or an owl: hasValue restriction (which is just a shortcut for a nominal concept), each reasoning task will throw an error. KAON2 currently cannot handle large numbers in cardinality statements. We have observed problems even on an ontology which contains a maximum cardinality restriction of two: KAON2 on this ontology is not capable of answering any queries.

VIII. CONCLUSION

Semantic Web brings the structure of meaningful content of the Web pages, creating an environment where software agents can take information from pages to pages that can readily carry out sophisticated tasks for users. This paper has analyzed the different reasoning techniques, discussed existing inference engines and identified their issues and opportunities. To understand and use the semantic data on the web there is a requirement of inference engine. If a reasoning tool utilized in the real world settings, it should provide rich practical features. On the semantic web, the reasoning tools inheriting the advantages of rule engines will survive as the efficient reasoning mechanism. Given comparison table will help to select an appropriate reasoner based on strengths and weaknesses.

REFERENCES

[1] Dan Brickley and Ramanathan V. Guh, *RDF vocabulary description language 1.0: RDF schema*. a Recommendation 10 February 2004, W3C, 2004..

[2] *Reducing OWL entailment to description logic satisfiability*, Ian Horrocks and Peter Patel-Schneider Journal of Web Semantics, 1(4):345-357, 2004.

[3] Tim Berners-Lee, James Hendler, and Ora Lassila. “*The semantic web*”. Scienti_c American, 284(5):34-43, 2001. Tim Berners-Lee, James Hendler, and Ora Lassila.

[4] *Pellet: An owl dl reasoner*. B. Parsia, E. Sirin, In: Proc. International Semantic Web Conference., 2005.

[5] *Racer: A Core Inference Engine for the Semantic Web*, Volker Haarslev and Ralf M¨oller. (2003).

[6] “A tableaux decision procedure for SHOIQ”, I. Horrocks, U. Sattler In: Proceedings of Nineteenth International Joint Conference on Artificial Intelligence. (2005).

[7] “The FaCT System”, International conference on Analytic Tableaux and Related Methods (TABLEAUX'98), pp 307-312, vol 1397, Springer-Verlag, (1998).

[8] “Hoolet OWL Reasoner”, Bechhofer (2003).

[9] http://owl.man.ac.uk/hoolet/

[10] “F-OWL: an Inference Engine for the Semantic Web by Youyong Zou, Tim Finin and Harry Chen, 2003.

[11] http://kaon2.semanticweb.org

[12] “Structured Objects in OWL representation and reasoning”, Boris Motik, Burnard Grau and Ulrike Sattler, (2008).

[13] “Evaluation of Ontologies and DL Reasoners”, Muhammad Fahad, Abdul Qadir and Syed Adnan Hussain Shah.