A Survey on Ontology Reasoners and Comparison

Sunitha Abburu, PhD.
Professor &Director,Dept. of M.C.A
Adhiyamaan College of
Engineering,Tamilnadu.

ABSTRACT

Reasoner is a software that is used to derive new facts from the existing ontologies. Some of the popular reasoners developed in the last few years are: Pellet, RACER, FACT++, Snorocket, Hermit, CEL, ELK, SWRL-IQ, TrOWL and others. This survey describes the reasoners that can be used as plug-in for either protégé or NeOn toolkit since these are most widely used ontology development tools. The current study describes the reasoners with their important features such as soundness, completeness, reasoning method, incremental classification etc. Finally this paper presents comparison of the reasoners with respect to their features.

General Terms
Semantic Knowledge, Ontology, Reasoner.

Keywords
Description Logic, Ontology Reasoners, Reasoner Attributes, Protégé, NeOn toolkit.

1. INTRODUCTION

An ontology is a theory which uses a specific vocabulary to describe entities, classes, properties and related functions with certain point of view [1]. Ontologies enable the computer to understand the information by its own. Researchers have developed ontology languages for knowledge representation, tools to construct ontologies and reasoners to infer the ontologies. Among many ontology languages OWL ontology language is most popular. OWL [2] is an ontology language designed for Semantic Web and is the language recommended by W3C. OWL has influences from quite a number of sources, but its main representational facilities are directly based on Description Logics. OWL2 [3] developed in 2007 to address some of the issues around OWL and it is recommended by W3C since 2009. OWL2 has three good profiles [4]. These profiles are called OWL 2 EL, OWL 2 QL, and OWL 2 RL. These logical fragments also provide expressive power for the efficiency of reasoning. OWL2EL provides polynomial time algorithms for all the standard reasoning tasks of description logic, OWL2QL enables efficient query answering over large instance populations, and OWL2RL restricts the expressiveness with respect to extensibility toward rule languages.

Protégé and NeOn toolkit ontology construction tools provide more features to ontology developers for constructing ontology. This paper describes ontology reasoners that support OWL ontology language and can be used as plug-in for either protégé or NeOn toolkit or both.

A reasoner is a program that infers logical consequences from a set of explicitly asserted facts or axioms and typically provides automated support for reasoning tasks such as classification, debugging and querying [5]. Among the large number of reasoners available, the reasoners that can support protégé or NeOn toolkit are: Pellet [6], RACER [7], FACT++ [8], Snorocket [9], HermiT [10], CEL [11], ELK [12] and SWRL-IQ [13], TrOWL [14]. These reasoners have various important attributes (characteristics) such as inference algorithm, supporting logic, degree of completeness of reasoning, implementation language and others.

The rest of the paper is organized as follows. Section 2 briefly describes Description Logic and its notations as a part of related work, section 3 gives a short note on the importance of ontology reasoners, section 4 outline the attributes of ontology reasoners, section 5 presents description of popular ontology reasoners, section 6 compares the reasoners with respect to their attributes and finally section 7 concludes.

2. RELATED WORK

Description Logic

The Description Logic (DL) is a subset of the first-order-predication logic [15].It models *concepts*, *roles* and *individuals*, and their relationships [16]. Concept is a collection of objects and role is a relation ship between objects.DL is designed to mainly represent the knowledge.

The DL provides the Boolean concept constructors plus the existential and universal restriction constructors are called ALC [17], where the Boolean concept constructors are concept disjunction, concept conjunction and concept negation. Various constructs and the logical languages based on the description logic are represented using Table 1. For example the extensions of ALC with transitive roles and role inclusion axioms are called ALCR+ and ALCH, respectively. The logic ALCHR+ is commonly known as SH. The extension of SH with inverse roles, nominals, qualified number restrictions and concrete domains is known as SHOIQ (D).

DL reasoners verify whether there are any logical contradictions in the ontology axioms. Furthermore [18] they can also be used to derive inferences from the asserted information.

Table1: Notations of description language

Notation	Description
S	ALC Description logic extended with transitive roles i.e. ALCR+
H	Role hierarchies
O/B	Nominals
Q	Qualified number restriction
N	Unqualified number restriction
(D)	Data types
A	Atomic concept
AL	Top, bottom, intersection and value restriction
U	Union
C	Negation
E	Existential quantifier
F	Agreement and disagreement with equality for feature chains
f	Agreement and disagreement without equality for feature chains
I	Role constructor inverse
R+	at subscript-restriction that some roles are
H	Role hierarchy with single inheritance
R	Role conjunctions

3. IMPORTANCE OF REASONERS

The quality and correctness of ontologies playsvital role in semantic representation and knowledge sharing [19]. To ensure the quality of ontologies, there is a need for dealing with the inconsistency and uncertainty in the ontologies of real-world applications.

An inconsistent ontology means that an error or a conflict exist in an ontology, as a result some concepts in the ontology cannot be interpreted correctly. The inconsistency will result in false semantic understanding and knowledge representation. An uncertain ontology means that the correctness of the ontology is probabilistic. Ontology reasoning reduces the redundancy of information in knowledge base and finds the conflicts in knowledge content.

4. REASONER ATTRIBUTES

This section describes important attributes of ontology reasoners. Kathrin Dentler et al categorized ontology reasoner attributes into three types [5]:

- *Reasoning characteristics*: this category describes basic features of ontology reasoners. E.g: methodology, sound and completeness, expressivity, incremental classification, rule support, justification and support of ABOX reasoning task.
- *Practical usability characteristics*: This category of attributes determines whether the reasoner implements the OWL API. They also describe the availability and license of the reasoners.
- *Performance indicators*: Performance indicators are used to measure the performance of ontology reasoners. E.g: classification performance, TBOX consistency checking performance etc.

This paper describes the reasoners that are suitable for protégé and NeOn with the reasoning characteristics, practical usability characteristics and some other special characteristics.

Methodology:It indicates the procedure or an algorithm that is used by the reasoner for solving basic reasoning problems in description logics. E.g: Tableau [20], Tableaux [21] etc.

Soundness and Completeness:This feature of ontology reasoner evaluates whether all possible inferences are inferred or not. Soundness or completeness can helps for a significant speed-up of reasoning [22].

Expressivity: It allows different sorts of axioms such as transitivity axioms and role inclusion axioms in RBOX.

*Incremental Classification:*When an ontology has been classified and is updated afterwards (by additions or removals), it makes sense for a reasoner to reuse the previous classification information together with the updated axioms to produce the new concept hierarchy.

*Rule Support:*Itenables the ontology reasoner to combine ontologies with rules.

Platforms:This feature indicates operating system on which a reasoner can work. e.g: windows, Linux etc.

Justifications:This feature of ontology reasoners provide explanations for inconsistency that exist in the ontologies.

ABOX Reasoning:ABox reasoning is reasoning with individuals and includes instance checking, conjunctive query answering and ABox consistency checking.

OWL API: The OWL API [23] is an Application Programming Interface (API) for working with OWL ontologies. It provides a standard interface to OWL reasoners, so that an application can insert different reasoners without having to change its implementation.

OWL Link:OWLlink [24] provides an extensible, implementation neutral protocol to interact with OWL 2 reasoners. It allows to turn OWL API aware reasoners into OWLlink servers and to access remote OWLlink servers from OWL API based applications (such as Protégé).

Protégé Support:It indicates whether the reasoner can be used with protégé tool or not.

NeOn Support: It indicates whether the reasoner can be used with NeOn toolkit or not.

License:The major distinguishing feature concerning licenses is whether the license is a recognized open source license or not.

Jena Support:It indicates whether the reasoner can be used with Jena API or not.

Impl. Language:It indicates the language which is used to implement a reasoner.

Availability:It indicates availability of reasoner. Many reasoners are free and open.

Native Profile: It indicates logical fragments that provide expressive power for the efficiency of reasoning.

5. ONTOLOGY REASONERS

"A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms". The inference rules are commonly specified by means of a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining [25]. Forward chaining and backward chaining are the strategies of ontology reasoners [26].

- *Forward-chaining*: According to this strategy the reasoner starts from the known facts and derive valid inferences. The goals of such reasoning are
  - To compute the inferred closure
  - To answer a particular query
  - To infer a particular sort of knowledge (e.g., the class taxonomy).
- *Backward-chaining*: According to this strategy the reasoner starts from a particular fact or a query and to verify it or to find all possible solutions.

The current section describes the summary of various ontology reasoners. Mishra et al. [27] presented an extensive survey of nineteen reasoners that have been released between 1975 and 2009. The authors have categorized the reasoners into four generations.

Among the large number of reasoners available the popular reasoners and suited for either protégé or NeOn toolkit are: Pellet, RACER, FACT++, Snorocket, HermiT, CEL, ELK, SWRL-IQ and TrOWL.

Pellet

Pellet is an open source java based OWL-DL reasoner developed by The Mind Swap group. It is based on the tableau algorithm and supports expressive description logics. It is the first reasoner that supported all of OWL DL SHOIN(D) and has been extended to OWL2 (SROIQ(D)) [28]. Pellet supports OWL 2 profiles. It reasons ontologies through Jenaas well as OWL-API interfaces. Pellet also supports the explanation of bugs. Fig1 shows various components of the pellet reasoner.

RACER

Horrocks et al., developed a reasoning model called as RACER (Renamed ABoxes and Concept Expression Reasoner). RACER, also known as RacerPro [29], is the first OWL reasoner. It supports the optimization techniques of FaCT as well as the new optimization techniques for dealing with number restrictions and ABoxes. RACER implements TBox and ABox reasoner for the SHIQ logic. Fig2 shows clear architecture of the reasoner RacerPro.

FACT ++

Horrockshas presented a reasoner known as FaCT (Fast Classification of Terminologies). It can be used as a description logic classifier and for modal logic satisfiability testing. The FaCT system has sound and

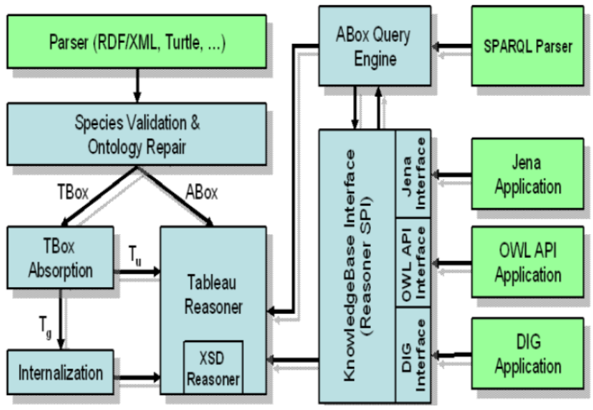

Fig1: Main components of the Pellet reasoner

complete tableaux algorithm for expressive description logics. An updated version of FaCT is FaCT++. This reasoner uses the same algorithm as in FaCT, but with a different internal structure. It is implemented in C++. The first version of the FaCT++ was only supporting the reasoning in SHOIQ, OWL-DL. However, the latest version of FaCT++ supports OWL and is based on the description logic SROIQ. FaCT++ implements a tableau-based decision procedure for general TBoxes and incomplete support for ABoxes [30].

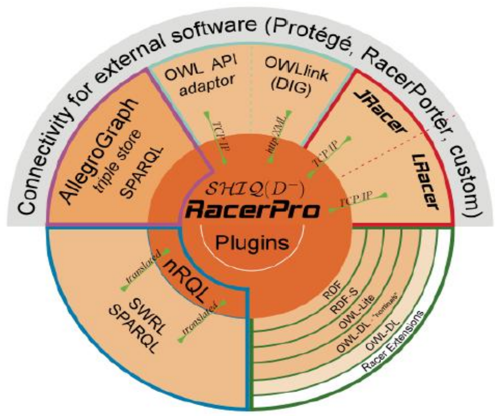

Fig2: RacerPro System Architecture

Snorocket

Snorocket [9][31] is a high-performance implementation of the polynomial-time classification algorithm for the lightweight Description Logic EL+. It is implemented in java. Snorocket has been developed as part of the CSIRO's Health Informatics and Clinical Terminologies research program.

SWRL-IQ

SWRL-IQ (Semantic Web Rule Language Inference and Query tool) [13][32] is a plugin for Protege 3.5 that allows users to edit, save, and submit queries to an underlying inference engine based on XSB Prolog. Fig3 shows architecture of the reasoner SWRL-IQ. The tool has number of features:

- Goal-oriented backward-chaining Prolog-style reasoning
- Tracing and debugging inference results.
- Saving queries.
- Exporting query results in XML or CSV format
- No dependency on proprietary or closed-source components.
- Uses XSB Prolog, which is freely available under the LGPL license.

ELK

ELK [12] [33] [34] is a free and open source reasoner for the lightweight ontology language OWL 2 EL.The ELK reasoner is based on Java and can be controlled using the OWL API. ELK is available under the Apache License 2.0. ELK runs in all operating systems that support Java 1.5 or above. Fig4 shows Main software modules of ELK reasoner and information flow during classification.

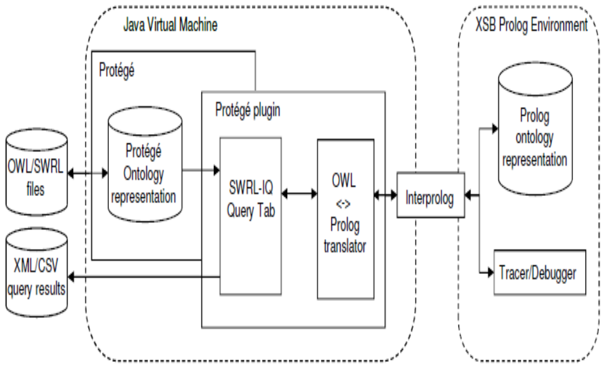

Fig3: SWRL-IQ System Architecture

HermiT

HermiT is the first publicly available OWL reasoner. It is written using OWL. HermiT can check the OWL files to determine the consistency of the ontologies and to identify the hierarchical relationships between the classes. This reasoner is based upon the hypertableau calculus. It also provides the faster process for classifying the ontologies.

CEL

CEL (Classifier for ε£) is a LISP based reasoner. It has very simple shell like interface. It provides users with all essential

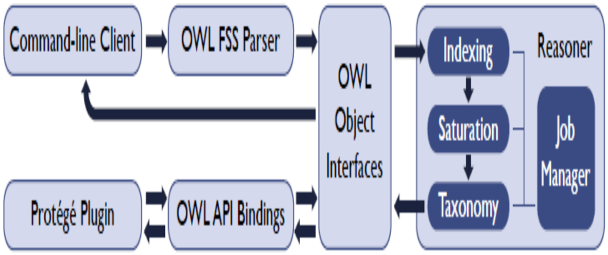

Fig4: Main software modules of ELK and information flow during classification

functionalities including a simple interactive help command. CEL is based on the refined version of polynomial-time classification algorithm. It mainly provides the classification reasoning services involving the computation of the complete subsumption hierarchy between all concept names occurring in the input ontology. Fig5 shows CEL system architecture**.**

TrOWL

TrOWL (Tractable OWL2 infrastructure) [14] is the common interface to a number of reasoners developed in Aberdeen University. TrOWL Quill provides reasoning services over OWL 2 QL. TrOWL REL is an optimized implementation of the CEL algorithm that provides reasoning over OWL 2 EL. To support full DL reasoning, TrOWL allows for the use of heavy weight plugin reasoners, such as FaCT++, Pellet, HermiT and RacerPro.

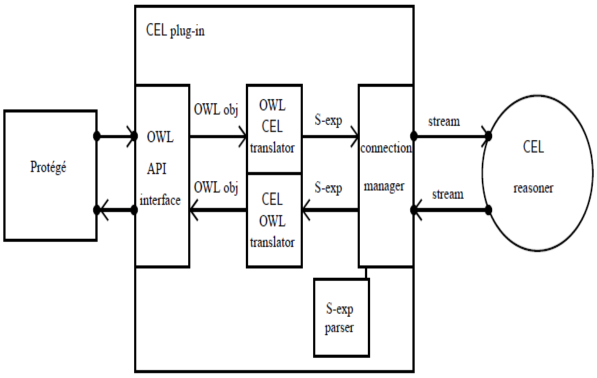

Fig5: CEL System Architecture

The current study observed that all the above reasoners can be used with protégé and NeOn toolkit. But the reasoners are not compatible with all the versions of protégé [35]. Table 2 shows compatibility of versions of reasoners with the versions of protégé and NeOn toolkit.

6. COMPARISON OF REASONERS

After a detailed study on ontology reasoners, this section presents a comparison of ontology reasoners with respect to their attributes. Table 3 shows comparison of the reasoners.

		Pellet	RACER	FACT++	Snorocket	SWRL-IQ	HermiT	CEL	TrOWL	ELK
Methodology		Tableau based	Tableaux based	tableau based	Completion rules	SWRL rules	Hypertableau based	Completion rules	Completion rules	Consequence based
Soundness		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Completeness		Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Expressivity		SROIQ(D	SHIQ	SROIQ(D	EL+	-	SROIQ(D)	EL+	SROIQ	EL
Native Profile		DL, EL	DL	DL	EL	-	DL	EL	DL, EL	EL
Incremental Classification	**Addition**	Yes	No	No	Yes	Y/N	No	Yes	No	Yes
	Removal	Yes	No	No	No	Y/N	No	No	No	Yes
Rule Support		Yes (SWRL)	Yes (SWRL)	No	No	Yes (SWRL)	Yes (SWRL)	No	No	Yes (Own rule format)
Platforms		all	all	all	all	all	all	Linux	all	all
Justifications		Yes	Yes	No	No	Yes	No	Yes	No	No
ABOX Reasoning		Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
OWL API		Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
OWL Link API		Yes	Yes	Yes	No	No	Yes	Yes	No	Y/N
Protégé Support		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
NeOn Support		Yes	No	No	No	No	Yes	No	No	No
License		DULI: AGPL	own	GLGPL	own	Y/N	GLGPL	Apache License 2.0	DULI: AGPL	Apache License 2.0
Jena Support		Yes	No	No	No	No	No	No	Yes	Y/N
Impl. Language		Java	LISP	C++	Java	Prolog	Java	LISP	Java	Java
Availability		Open source	Commercial	Open Source	Commercial	Y/N	Open source	Open source	Commercial	Open source

Table 3: Comparison of reasoners (Y represents supported feature, N represents non-supported feature, Y/N represents need further explanation)

Table 3: Reasoner versions compatible with protégé and NeOn Toolkit versions (- represents not supported)

Reasoner	Version	Compatible with	
		Protégé Version	NeOn Toolkit Version
Pellet	1.x	3.x, 4.x	-
	2.x	-	2.x
RacerPro	1.1.10	4.1, 4.2	-
FACT++	-	4.x	-
Snorocket	1.3.4	4.0, 4.1	-
	1.3.1	4.0	-
	1.3.0	4.0	-
HermiT	1.3.6	4.2	2.x
	1.3.5	4.2	
	1.2.2	4.1	
	1.2.1	4.1	
CEL	1.0	4.x	-
ELK	0.3.0	4.1, 4.2	-
	0.2.0	4.1	-
	0.1.0	4.1	-
SWRL-IQ	1.1b	3.5	-
	1.1	3.4	-
	1.0	3.4	-
TrOWL	any Version	4.0.x, 4.1.x	-

7. CONCLUSION

This paper have provides a comprehensive study and a survey of current available and popular ontology reasoners which can be used as plug-in for either protégé or NeOn toolkit. We have compared the reasoners with respect to their features. From the comparison table among the all reasoners some of them have common methodology and others are have different reasoning methodology. All reasoners are sound and complete except SWRL-IQ. Only the reasoners pellet and ELK supports full incremental classification and others are not. The reasoners pellet, RACER, SWRL-IQ, HermiT and ELK provide rule support. All reasoners works on all platforms except CEL. Only Pellet, RACER, SWRL-IQ and CEL give explanations for inconsistency exist in the ontologies. All reasoners perform ABOX reasoning except Snorocket and ELK. All reasoners have OWL API support except SWRL-IQ. The reasoners cannot be operated by Jena API except pellet.

The reasoners Pellet, FACT++, HermiT and ELK are open sources and others are not. All the reasoners can be used as plug-in for protégé but for NeOn toolkit only Pellet and HermiT. The main conclusion from this study is that reasoners vary significantly with respect to all included reasoner attributes, a critical estimation and evaluation of requirements is needed before selecting a reasoner for a real-life application.

8. ACKNOWLEDGMENT

The work presented in this paper is done as part of a sponsored project founded by government of India, Ministry of Defence, DRDO (ER&IPR), and done in the labs of Adhiyamaan College of Engineering where the author is working as a Professor& Director in the department of Master of Computer Applications. The author would like to express her sincere thanks to DRDO for providing the support.

9. REFERENCES

[1] Fonseca, F. Egenhofer M., Agouris, P., Camara G, "Using Ontologies for Integrated Geographic Information Systems". Transactions in GIS, − (6):3 in print, 2002.

[2] http://www.w3.org/TR/owl2-primer/.

[3] Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Lutz, C. (eds.), " OWL 2 web ontology language profiles", W3C Recommendation (Oct 2009).

[4] http://www.w3.org/TR/owl2-profiles/.

[5] Kathrin Dentler, Ronald Cornet, Annette ten Teije and Nicolette de Keizer, "Comparison of Reasoners for Large Ontologies in the OWL 2 EL Profile", *Semantic Web Journal*, 2011, pp.1-5.

[6] Parsia B, Sirin E, "Pellet: An OWL DL Reasoner", MINDSWAP Research Group, Supporting Reasoners and Softwares, 2003.

[7] Haarslev V, Moller R, "Consistence Testing: the RACE Experience", In: Dyckhoff R (ed) Proceedings, automated reasoning with analytic tableaux and related methods. Lect Notes Artif Intell 1847:57–61, 2000.

[8] Horrocks I, "The FaCT System", http://www.cs.man.ac.uk/~horrocks/FaCT/, website last updated on April, 2003.

[9] M.J. Lawley, C. Bousquet, "Fast Classification in Protégé: Snorocket as an OWL 2 EL Reasoner", *Australasian Ontology Workshop 2010 (AOW 2010): Advances in Ontologies*, volume 122 of CRPIT,2010, pp. 45–50.

[10] Glimm B,Horrocks I,Motik B, StoilosG, "HermiT: Reasoning with Large Ontologies", http://www.comlab. ox.ac.uk/projects/HermiT/index.html, Computing Laboratory, Oxford University, 2009.

[11] Baader F, Lutz C, Suntisrivaraporn B, "Efficient reasoning in EL ˆ+", In Proceedings of the 2006 international workshop on description logics (DL2006), CEUR-WS, 2006.

[12] Yevgeny Kazakov, Markus Krötzsch, František Simančík, "ELK Reasoner: Architecture and Evaluation"**,** In Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012). CEUR Workshop Proceedings 2012.

[13] Daniel Elenius, Susanne Riehemann, "SWRL-IQ User Manual", January 16, 2012 Version 1.0.

[14] E. Thomas, J. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In Proceedings of the Extended Semantic Web Conference. Springer, 2010.

[15] Zhu ChuangLu, "Research on the Semantic Web Reasoning Technology", in proceedings of 2012 AASRI Conference on Computational Intelligence and Bioinformatics, Elsevier, 2012, pp.87-91 ; doi: 10.1016/j.aasri.2012.06.016.

[16] Baader F, Nutt W, "Basic Description Logics", in: F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic.

[17] Schauss MS, Smolka G, "Attribute Concept Descriptions with Complements", Artif Intell 1991;48:1–26.

[18] Grigoris Antonion and Frank Van Harmelen, "Web Ontology Languages", Handbook on Ontologies, Second Edition, Springer, 2009, pp. 91-110.

[19] Bo Liu, Jianqiang Li, Yu Zhao, "Repairing and Reasoning with Inconsistent and Uncertain Ontologies", Elsevier, 2011, pp.380-390; doi:10.1016/j.advengsoft.2011.10.015.

[20] *F. Baader and W. Nutt, "*Basic Description Logics*",* THE DESCRIPTION LOGIC HANDBOOK Theory, implementation, and applications, Cambridge University Press, 2003, pp.41-95.

[21] Horrocks, I., Sattler, U, "A tableaux decision procedure for SHOIQ", In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), Edinburgh, pp. 448–453 (2005).

[22] S. Rudolph, T. Tserendorj, and P. Hitzler. "What Is Approximate Reasoning? Web Reasoning and Rule Systems", Springer, 2009, pp.150–164.

[23] M. Horridge and S. Bechhofer, "The OWL API: A Java API for Working with OWL 2 Ontologies", In 6th OWL Experienced and Directions Workshop, 2009.

[24] T. Liebig, M. Luther, O. Noppens, M. Rodriguez, D. Calvanese, M. Wessel, R. Möller, M. Horridge, S. Bechhofer, D. Tsarkov, and Others. OWLlink: DIG for OWL 2. In 5th OWL Experienced and Directions Workshop, 2008.

[25] Introduction to Semantic Web", available at http://www.mphasis.com/knowledge-center/white-papers-all.asp .

[26] Atanas Kiryakov, mariana Damova, "Reasoning in the Semantic Repositories", Handbook of Semantic Web Technologies, Springer, 2011, pp. 245-258.

[27] Mishra R B, Sandeep Kumar, "Semantic Web Reasoners and Languages", Springer, 2010**;** DOI 10.1007/s10462-010-9197-3 .

[28] E. Sirin, B. Parsia, B. Grau, a. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL Reasoner". Web Semantics: Science, Services and Agents on the World Wide Web, 5(2):51–53, June 2007.

[29] Volker Haarslev, Kay Hidde, Ralf Moller, and Michael Wessel, " The RacerPro Knowledge Representation and Reasoning Systems", Semantic Web Journal, IOS Press, 2011, available at:www.franz.com/agraph/cresources/white_papers.

[30] Sattler U, "Description Logic Reasoners", http://www.cs.man.ac.uk/~sattler/reasoners.html, website accessed on Feb 19, 2007.

[31] http://research.ict.csiro.au/software/snorocket.

[32] https://www.onistt.org/display/SWRLIQ/SWRL-IQ.

[33] Yevgeny Kazakov, Markus Krötzsch, František Simančík, "ELK: A Reasoner for OWL EL Ontologies"**,** System Description 2012.

[34] http://www.cs.ox.ac.uk/isg/tools/ELK/.

[35] http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library.