March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

Extending Description Logics for Semantic Web Ontology Implementation Domains

Ajeet K. Jain1, Dr .PVRD Prasad Rao 2, Dr. K. Venkatesh Sharma 3

1Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India; (Association: CSE, KMIT, Hyderabad, India)
2 Professor, CSE, KLEF, Vaddeswaram, AP, India
3Professor, CSE, CVR College of Engg., Hyderabad, India
1jainajeet123@gmail.com, jainajeet1@rediffmail.com, 2 pvrdprasad@kluniversity.in, 3 venkateshsharma.cse@gmail.com

Article Info
Volume 83
Page Number: 7385 - 7393
Publication Issue:
March - April 2020

Abstract

Semantic web has emerged as the field of integrating various logic families and their extensions. The Description Logic extensions provided considerable attention and research for better expressivity and decidability. The exactness of validity and soundness is provided by extending the rules and axioms as facts and the completeness as every valid statement is associated with a proof.

This article provides basics of most powerful way of logic expression – Description Logic (DL) and their interplay with various semantics extensions and also highlights a brief overview of research in nearly last 2 decades for various approaches for reasoning in expressive DLs. The foundation of OWL DL with tractable reasoning is also presented. The blending of logic families are discussed with extensions of the basic representation language systems.

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 07 April 2020*

Keywords; *Description Logics, Knowledge Representation, ABox, TBox, RBox, Web Ontology Language*

I. INTRODUCTION

Description Logics are family of logic based knowledge representations (KR) formalisms representing the conceptual knowledge of an application domain in a structured way and at present abundantly used in ontological languages modelling. The extension using conjunction ($\sqcap$) and existential ($\exists$) and top (T) concepts has provided important inferences to subsumption and polynomial problems and solutions in axioms. Importantly, they have been successfully adopted in Web Ontology Language (OWL) for semantic web. Moreover, they are also finding applications in areas such as software engineering, medical informatics, digital libraries, natural language processing, and databases [1].

II. DESCRIPTION LOGIC

Description logics (DL) are logics serving primarily for formal description of concepts and roles (relations). These logics were created from the attempts to formalize semantic networks and frame based systems. Semantically they are found on predicate logic, but their language is formed so that it would be enough for practical modelling purposes and also so that the logic would have good computational properties such as decidability. The focus of research in DLs is how various DL constructs and extension impact the decidability and complexity issues.

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

The classical knowledge representation system based on DLs consists of two components - **TBox** and **ABox** [1]. The **TBox** describes ***terminological knowledge***, i.e., the ontology in the form of concepts and roles definitions, while the **ABox** contains *assertions* about individuals using the terms from the ontology. Concepts describe sets of individuals, roles describe relations between individuals is depicted in Fig.1.

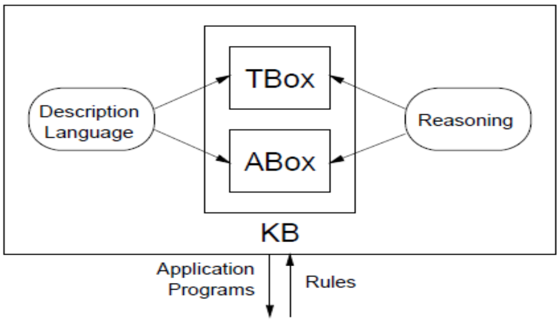


Fig. 1 A KR representation

For instance, the statement:

Every employee is a person - belongs to TBox (1)

Jain is an employee – belongs to ABox (2)

Here, the TBox /ABox distinction is not significant in the sense that the two "kinds" of sentences are not treated differently in First Order Logic - FOL. When we translate them into FOL, a subsumption axiom like (1) is simply a conditional restriction to unary predicates (concepts) with only variables appearing in it. Clearly, a sentence of this form is not privileged or special over sentences in which only constants ("grounded" values) appear like (2).

In fact the software engineering principle known as ' *separation of concern* ' is a niche for keeping the distinction separate between these boxes. There are basically two reasons: primarily, the separation can be useful when describing and formulating decision-procedures for various DL. For instance , a reasoner might process the TBox and ABox separately, in part because certain key inference problems are tied to one but not the other one ('classification' is related to the TBox, 'instance checking' to the ABox). The complexity of the TBox can greatly affect the performance of a given decision-procedure for a certain DL, independently of the ABox. Thus, it provides a useful a way to think about that specific part of the knowledge base representation.

The third box is termed as **RBox** (Role **Box**) which entails knowledge base consisting of axiom so the roles, like:

$$\left.\begin{array}{l} hasChild \sqsubseteq hasDescendant \\ has\ Descendant\ \circ hasDescendant\ \sqsubseteq \\ has\ Descendant \\ hasParent = hasChild^{-} \end{array}\right\} \quad (3)$$

The RBox is a set of axioms of the form

$$\left.\begin{array}{l} R\ \sqsubseteq S\ (Hierarchy\ of\ Roles\ \ or \\ R \circ R\ \ \sqsubseteq R\ (transitive\ Role) \end{array}\right\} \quad (4)$$

2.1 **Semantic of DL :** is a formalism of knowledge representation(KR), possessing more expressiveness than propositional logic but less expressive than FOL The core reasoning problems for DLs are (usually) decidable, and efficient decision procedures have been designed and implemented for these problems. There are general, spatial, temporal, spatiotemporal, and fuzzy descriptions logics, and each one features a different balance between DL expressivity and reasoning complexity by supporting different sets of mathematical constructors [2, 3].

DL belongs to a family of formal KR languages and is more expressive than the conventional propositional logic but lesser expressiveness than the FOL. The core reasoning power of DL are decidable and can be designed and implemented for many problems. There are advance versions of the extensions to DL as Fuzzy DL and spatiotemporal – with a balance of expressivity. A concepts of a happy man (and so also of a woman) can be described in DL as:

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

$$\textbf{Human}\sqcap\textbf{Male}\sqcap\textbf{Healthy}\sqcap\textbf{Handsome}\sqcap\exists\textbf{spouse}(\textbf{Rich}.\sqcap\textbf{Intelligent}\sqcap\textbf{Female})\sqcap \exists\textbf{child.Male}\sqcap\exists\textbf{child.Female}\sqcap\exists\textbf{Friend.T} \quad (5)$$

This concept construct (5) suggests a male human who is healthy and handsome and have a rich intelligent wife, a son and a daughter and a friend.

With (5), all the above properties need to be satisfied, however in seldom they are! It is worth calling man happy if most of the properties are satisfied and not all. This entails that a degree of satisfiability is needed in an appropriate way. The idea is to extend the DL with a graded membership function between a value of 0 and 1 —unlike Boolean values of 0 and 1. Conclusively, we could put 0.8 as a degree of happiness when most of the properties are met with and thereby unhappy as 0.2 as (or less than this). This closely resembles with Fuzzy logic [4]. However, there are some differences —firstly, in fuzzy DLs, the semantic is extended to fuzzy interpretations where concept and role are interpreted as fuzzy sets and relations. The degree of membership can then be computed. Secondly, a threshold concept applied could be crisp rather than a fuzzy. For more details on Fuzzy based DL systems, refer to [4].

2.2 Constructors for Concepts and Roles in DL Systems - DL systems allow new concepts and roles to be built using a variety of different constructors distinguishing between them depending on whether concept or role expressions are constructed. For concepts, we can further separate basic Boolean constructors, role restrictions and nominals / enumerations. A nominal is a concept that has exactly one instance. For example, { **jain** } is the concept whose only instance is , ie., the individual represented by is **jain.** By the same token, we may define the concept **ClassicalIndianMixedMusic** (**CIMM**) by enumerating its instances: **RDBurman, AshaBhosale, KishoreKumar, SDBurman**. Incidentally, the Enumerations are not supported natively in DLs, but the good point is that they can be simulated in DLs using *nominals*. Combining nominal with union, the enumeration can be expressed as

$$\textbf{CIMM} \equiv \{\textbf{RDBurman}\} \sqcup \{\boldsymbol{AshaBhosale}\} \sqcup \{\boldsymbol{KishoreKumar}\} \sqcup \{\boldsymbol{SDBurman}\}$$

Further, using nominal, a concept assertion Mother(Jamuna) can be converted into a concept inclusion {Jamuna} ⊑ Mother and a role assertion parentOf(Jamuna) into a concept inclusion {Jamuna} ⊑ ∃parentOf.{Janaki}.

2.3 General Concept Inclusion (**GCI**) - The GCI state that all mothers are female and that all mothers are parents, but what we really meant is that mothers are exactly the female parents. DLs support such statements by allowing us to form complex concepts as conjunction

$$\boldsymbol{Female} \sqcap \boldsymbol{Parent}$$

which in turn tells us about the set of individuals who are both female and parents. A complex concept can be used in axioms the same way as an atomic concept - like the equivalence

$$\textbf{Mother} \equiv \textbf{Female} \sqcap \textbf{Parent}$$

Similarly, disjunction is the dual of intersection as the concept

$$\textbf{Father} \sqcup \textbf{Mother}$$

describes those individuals that are either fathers or mothers. Thus, it can be used in an axiom such as

$$\textbf{Parent} \equiv \textbf{Father} \sqcup \textbf{Mother}$$

which states that a parent is either a father or a mother.

At times, we are also interested in individuals that do not belong to a certain concept, like wo(men) who are not married. This could be described by the complex concept

$$\textbf{Female} \sqcap \neg\, \textbf{Married}$$

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

where the complement (*negation*) ¬ Married represents the set of all individuals that are not married.

If we want to state that everybody is either male or female then the axiom

$$\top \sqsubseteq \textbf{Male} \sqcup \textbf{Female}$$

where the *top concept* ⊤ is a special concept with every individual as an instance. However, this modelling is rather smutty as it assumes that every individual has a gender, which may not be the case such as Computer or a Printer_Device.

Furthermore, in order to express that, for the purposes of our modelling, nobody can be both a male and a female at the same time, we can declare the set of male and the set of female individuals to be disjoint. While ontology languages like OWL provide a basic constructor for disjointness, it is naturally captured in DLs with the axiom

$$\textbf{Male} \sqcap \textbf{Female} \sqsubseteq \bot$$

where the bottom concept ⊥ is the dual of ⊤, that is the special concept with no individuals as instances; it can be seen as an abbreviation for K ⊓ ¬ K for an arbitrary concept K. This implies that the intersection of the two concepts is empty indeed.

2.4 Role Restrictions and Constraints

The best stimulating feature of DLs is their ability to construct the statements that link concepts and roles together. As we already know that there is an obvious relationship between the concept Parent and the role parentOf, viz., a parent is someone who is a parent of at least one individual. Hence to depict this into DLs, the relationship can be formulated as concept equivalence

$$\textbf{Parent} \equiv \exists\textbf{parentOf}.\top$$

Here, the existential restriction, namely **∃parentOf.⊤** is a complex concept that describe those parents of at least one individual . In the same manner, **∃parentOf.Female** informs us about those individuals that are parents of at least one female – meaning that they have a daughter.

In the same way, we can extend the concept to sate that those parents who have Fe(male) children as

∃parentOf.Female | ∃parentOf.Male

Implies those ones whose all children's are Fe(Male). The other important point to ponder here is that, this type of axiomatic declaration has a flaw-meaning that it also includes all those individuals who have no children at all! A better way to extend this is to have a conceptual DL statement, expressed as

$(\exists parentOf.T) \sqcap (\forall \text{parentOf.Female}) \sqcap (\forall parentsOf.Male)$

These two qualifiers are very useful in combining with top concept for stating domain expressivity and range restrictions on the roles. Stating other way, if we wish to restrict the domain to **sonOf** to make individual, we can this using

$(\exists sonOf.T\ ) \sqsubseteq Male$

$(\exists daughterOf.T\ ) \sqsubseteq female$

and restricting the range to the parents, we can depict them as

$T\ \sqsubseteq \forall sonOf.Parent$

Now providing the assertion **sonOf(Jatin,Jamuna),** this asserts that **Jatin** is a male and **Jamuna** is his parent. This way of describing the semantic ontology have a distinct meaning when compared to a traditional databases schemas while embedding the constraints in a typical DBMS. This way of specifying constraints allow the language with intuitive meaning that all sons must be male. On the other hand, giving only the fact that Jatin is a son of **Jamuna**, making this straight constraint may simply be dishonoured – thus encountering a semantic error- instead of the natural implication that **Jatin** is a male. Thus , this kind of errors are quite frequent

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

while extending the DL axioms for constraint and one has to take care while making such axioms for their consistency violation and checks.

2.5 DL Language Semantics

In this section, we briefly provide the language semantics and further more can be refereed in [6]. To pave with, DL axioms are model-theoretic semantics and provide guidelines as tool that computes the logical significances of ontology. As stated in the previous section, we have described the 'family and relationship' domain. Given this kind of ontology, we cannot fully specify the situation that these semantics describe while extending them for intuitive meanings. An important point o note here is that, there is no formal or otherwise relationships between the symbols we use and the corresponding objects they describe - meaning Jatin and Jamuna are merely syntactic identifiers and have no effect on desiring their formal intrinsic meaning in any way. Instead, many times the ontological axioms in DL do not provide full information- as it follows the 'open world assumption' in general – meaning absence of information does only mean that it's unavailable.

This way of accomplishment of DL system extension are being designed in scenarios where partial of incomplete information are available and they can be inferred using reasoner tools- which are available currently for the purpose. The best way to state this is to say "OWA keeps the unspecified information open". Moving on further, we briefly describe the meaning of 'interpretation' in the context of DL semantics. We formally define the same way as it has been rationally done so far, the interpretation as I consists of a set $\boldsymbol{\Delta^I}$ called the domain and an interpretation function $\cdot^{I}$ that maps atomic concepts A to a set $A^I \subseteq \Delta^I$, each role R to a binary relation $R^I \subseteq \Delta^I \times \Delta^I$ and each individual name α to an element $\alpha^I \varepsilon \Delta^I$.

Table 1 depicts the details of expressions for such parts of the semantics. It is evident from the entries that the interpretation I injects the meaning of all objects ("Things") we can thus explicitly define such axioms whether they hold or not! Intuitively, we define an axiom m holds in I (stating other way as I satisfies **m**) and we express this as $I \vdash m$ when the corresponding conditions given in Table 2 are met with. Thus we can infer that if all the axioms in an ontology O holds in interpretation I , i.e., if I satisfies O ; written as $I \vdash O$, then I is the model of *O*. This in turn infers us that an inconsistent ontology entails every axiom and thus paves roads for doing further research into.

Table 1. (Resource : www.DL.org)

	Syntax	Semantics
Individuals:		
individual name	a	a^I
Roles:		
atomic role	R	R^I
inverse role	R^-	$\{\langle x,y\rangle \mid \langle y,x\rangle \in R^I\}$
universal role	U	$\Delta^I \times \Delta^I$
Concepts:		
atomic concept	A	A^I
intersection	$C \sqcap D$	$C^I \cap D^I$
union	$C \sqcup D$	$C^I \cup D^I$
complement	$\neg C$	$\Delta^I \setminus C^I$
top concept	$\top$	Δ^I
bottom concept	$\bot$	$\emptyset$
existential restriction	$\exists R.C$	$\{x \mid$ some R^I-successor of x is in $C^I\}$
universal restriction	$\forall R.C$	$\{x \mid$ all R^I-successors of x are in $C^I\}$
at-least restriction	$\geqslant n\, R.C$	$\{x \mid$ at least n R^I-successors of x are in $C^I\}$
at-most restriction	$\leqslant n\, R.C$	$\{x \mid$ at most n R^I-successors of x are in $C^I\}$
local reflexivity	$\exists R.Self$	$\{x \mid \langle x,x\rangle \in R^I\}$
nominal	$\{a\}$	$\{a^I\}$

where $a, b \in \mathsf{N}_I$ are individual names, $A \in \mathsf{N}_C$ is a concept name, $C, D \in \mathbf{C}$ are concepts, $R \in \mathbf{R}$ is a role

Table 2 (Resource ://www/DL.org)

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

	Syntax	Semantics
ABox:		
concept assertion	$C(a)$	$a^I \in C^I$
role assertion	$R(a,b)$	$\langle a^I, b^I \rangle \in R^I$
individual equality	$a \approx b$	$a^I = b^I$
individual inequality	$a \not\approx b$	$a^I \neq b^I$
TBox:		
concept inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
concept equivalence	$C \equiv D$	$C^I = D^I$
RBox:		
role inclusion	$R \sqsubseteq S$	$R^I \subseteq S^I$
role equivalence	$R \equiv S$	$R^I = S^I$
complex role inclusion	$R_1 \circ R_2 \sqsubseteq S$	$R_1^I \circ R_2^I \subseteq S^I$
role disjointness	$Disjoint(R,S)$	$R^I \cap S^I = \emptyset$

A noteworthy thing of this kind of semantics is that the DL does not make Unique Name assumption (UNA) in a formal definition and two individual name may be interpreted in either same way or other depending on the domain they belong to. For instance, 'Ford' may signify a name od an individual as well as the car name too. However their context, i.e., domain are different and some time it may lead to confusion as well. Once the domain are well recognised, this leads to simplification. In the same way, an another kind of confusion may occur when individual are represented by names – meaning named individuals. By the same token, we could wrongly the following axiom

parentOf(Jamuna,Jatin)
manyChildren(Jamuna)

manyChildren $\sqsubseteq \geq 4\ \boldsymbol{parentOf.T}$

leads to inconsistency as the axiom stares that Jamuna should have at least(minimum) 4 children , whereas we have provided only one named child as Jatin. This kind of logical interpretation errors can be simply be understood when one keeps the OPA thing of DL in consideration.

III. GOING FROM DL TO WEB ONTOLOGY LANGUAGE (OWL)

As we have seen that DL is provide a full rich set of operators and qualifiers, moving from DL to the web is provided by WWW as OWL for implied knowledge representation, proving most important application for web.

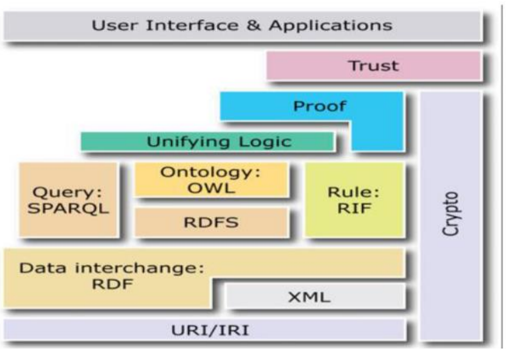


Fig. 2 A semantic RDF Fragment (courtesy : Semantic web primer [7])

A brief of OWL implementation is line and wec can refer to [7] for more details. To begin with, an axiom

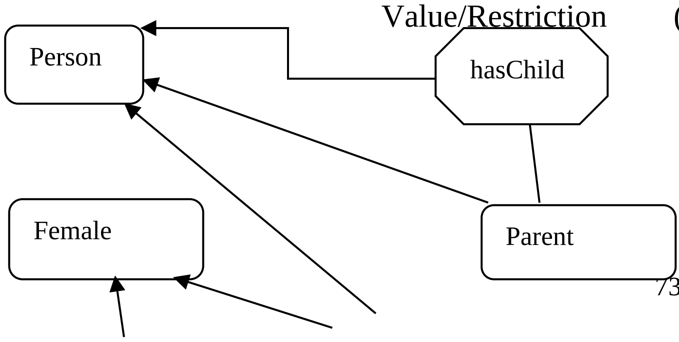
Father = Male ⊓ Parent could equivalent be written as

EquivalentClasses(Father ObjectIntersectionOf(Male Parent))

We can see from the above semantic declaration that OWL provides are similar to the DL syntax with more string like declaration. This kind of functional style gives a leverage for OWL standards a number of features, especially used in ontology based systems – notably the RDF/XML serialization. Additionally, OWL also provides a number of other aspects where some of them are not even provided in DL systems even. Just to mention a few, other features include the meta- modelling ones, non – logical axioms and annotations to add arbitrary axiom and many more. A full description of them, one can refer to [7].

3.1 A Semantic Network For Dl As Example

Value/Restriction (1,



March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

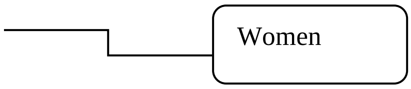


Fig. 3 A semantic network of family knowledge base

A network representing knowledge base concerning persons, parents, children, etc., is shown in Fig.3. This kind of structure is also referred to as a terminology, and meant to represent the generality/specificity of the concepts involved – like , the link between Mother and Parent says that "*mothers are parents*"; depicting "**IS-A**" relationship. This indeed defines a hierarchy over the concepts and provides a basis for the "inheritance of properties": when a concept is more specific than some other concept, it inherits the properties of the more general one. For instance, a person has an age and then a mother has an age, too. This is called monotonic inheritance networks and has a concept of Parent and exhibit a property called a "*role*," expressed by a link from the concept to a node for the role labelled ***hasChild.*** The role has a "value restriction," which expresses a limitation on the range of types of objects that can be applied to a particular role. Additionally, the node also has a number restriction expressed as (1,NIL), where the first number is a lower bound on the number of children and the second element is the upper bound, and NIL denotes infinity. Thus, the representation of the concept of Parent here can be read as "*A parent is a person having at least one child, and all of his/her children are persons*." This kind of relationships inherits the concepts to their subconcepts – like, the concept Mother, i.e., a female parent, is more specific descendant of both the concepts **Female** and **Parent**, and as a result inherits from Parent the link to Person through the role **hasChild;** so the concept Mother inherits the restriction on its **hasChild** role from Parent.

3.2 Terminological Axioms for FamilyExample

As depicted in Fig .2 and 3, the TBox incorporates axioms in most general senses as:

$$C \sqsubseteq D \quad (R \sqsubseteq S) \quad \text{or} \quad C \equiv D \quad (R \equiv S) \quad (6)$$

Here C, D are concepts and R, S are roles showing axioms of inclusions and equalities. Further, an equality whose left-hand side is an atomic concept is a definition and is used to introduce symbolic names for complex descriptions. For instance, the concept:

$$\text{Mother} \equiv \text{Woman} \sqcap \exists \text{hasChild.Person} \quad (7)$$

is associated to the description on the right side as the name **Mother** and same way we define **Father** analogously to Mother, we can define **Parent** as:

$$\text{Parent} \equiv \text{Mother} \sqcup \text{Father} \quad (8)$$

Going further, we can define a TBox for a family relationships in Fig. 3 and its expansion as followings :

<u>Family relationships</u>

Woman≡ Person⊓ Female

Man ≡ Person ⊓ ¬ Woman

Mother ≡ Woman⊓∃Child.Person

Father ≡ Man⊓∃hasChild.Person

Parent ≡ Mother⊔Father

GrandMother ≡ Mother⊓∃hasChild.Parent

MotherWithManyChildren ≡ Mother⊓ ≥3 hasChild

MotherWithoutDaughter ≡ Mother⊓ ∀hasChild.¬ Woman

Wife ≡ Woman⊓∃hasHusband.Man

<u>Family relationships expansion snippet</u>

Woman ≡ Person⊓Female

Man ≡ Perso⊓¬ (Person⊓Female)

Parent≡((Perso⊓¬(Person⊓Female))⊓∃hasChild. Person) ⊔((Person⊓Female)⊓∃hasChild.Person)

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

IV. APPLICABILITY

DLs are used in AI to describe and reason about the relevant concepts of an application domain (known as *terminological knowledge*). It is of particular importance in providing a logical formalism for ontology and the Semantic Web: the Web Ontology Language (OWL) and its profile is based on DLs. The most notable application of DLs and OWL is in biomedical informatics where DL assists in the codification of biomedical knowledge [8, 9]. Further, with these extensions and using OWL semantics, Oracle 11g uses these principles for query processing and knowledge discovery.

V. CONCLUSION AND FURTHER DISCUSSION

The paper describes the capabilities of inferring additional knowledge with the modeling power of DLs. An important goal of DL systems is to ensure that reasoning algorithms are sound and intractable and with possible extension, one can accomplish these virtues. This also makes a promising reason that there is just not a single DL to make it, but instead , it proves a balance n\between the expressivity of extension and complexity of reasoning depending on the intended domain application in mind. The DL axioms are model-theoretic semantics specifying the logical consequences on ontology systems. The ontology discussed in the article highlights ‘ family and relationship’ domain. To suffice, the ontology cannot fully specify the situation that they describe. In this quest, we have attempted to extend the expressive power of DL systems ideas- as these are being designed to deal with incomplete information. DL systems generally consider all possible situations where axioms of an ontology won’t fit appropriately.

This article has delved into the chronological advancement in description logic and blended with other logic families for better expressivity. The expressiveness and contradictions and constraints of various logic families which are very intrigue part of the semantic web. The attempt highlights the mix of various logic families for web ontology language OWL. Additionally, the concepts presented are well established over the semantic-ness of the web and thus provide a better understanding and interweaving of varied kinds of semantic data (structured, semi and un-structured) .

The critical element in developing application based on DLs is the usability of the knowledge based system and the article provide a summative approach which had made a significant contribution for ontology based systems. More interesting research areas with extension to DL system have already come and many more possibilities can thus be explored[10.11,12]

REFERENCES

[1]. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2010.
[2]. Barwise, Jon , "An Introduction to First-Order Logic", in Barwise, Jon, ed 1977.
[3]. Hodges, Wilfrid ; "Classical Logic I: First Order Logic", in Goble, Lou (ed.),1998
[4]. Borgwardt, S, Diestel F and Penaloza R, The Limits of decidability in fuzzy description logic with general concept inclusions. AI 218, pp. 23-55, 2015
[5]. The Blackwell Guide to Philosophical Logic, Blackwell , 2001
[6]. F. Baader. Description logics. In Proceedings of Reasoning Web: Semantic Technologies for Information Systems, volume 5689 of Lecture Notes in Computer Science, pp. 1–39, 2009.
[7]. G Antoniou, et al; Semantic Web Primer , MIT Press 2012
[8]. H. Peter Alesso and Craig F. Smith : Thinking on the Web – Wiley India , 2009

Published by: The Mattingley Publishing Co., Inc.

March - April 2020
ISSN: 0193-4120 Page No. 7385 - 7393

[9]. Berners-Lee T, Hendler J and Ora L: The semantic Web, Scientific American , May 2001 pp. 35-43
[10]. Peter F. Patel-Schneider, Deborah L. McGuiness, Ronald J. Brachman, Lori Alperin Resnick, and Alexander Borgida. The CLASSIC knowledge Representation system: Guiding principles and implementation rational. SIGART Bull., 2(3):108-113, 1991
[11]. Ronald J. Brachman. “Reducing" CLASSIC to practice: Knowledge representation meets reality. In Proc. of the 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'92), pp. 247- 258. Morgan Kaufmann, Los Altos, 1992.
[12]. Boontawee Suntisrivaraporn. Module extraction and incremental classification: A pragmatic approach For EL+ ontologies. In Sean Bechhofer, Manfred Hauswirth,Joerg Ho_mann, and Manolis Koubarakis, editors, Proceedings of the 5th European Semantic Web Conference (ESWC'08), volume 5021 of Lecture Notes in Computer Science, pp. 230-244, Springer-Verlag, 2008.