ISSN 2348 – 7968

An Introduction To Ontology: Survey

Mr. Pranit Chettri
M.tech,CSE,
Department of CSE,
Sikkim Manipal Institute of Technology,
Majhitar,Sikkim

Mr. Dhruba Ningombam
Assistant Professor,
Department of CSE,
Sikkim Manipal Institute of Technology,
Majhitar,Sikkim

Dr. C.T. Singh
Professor,
Department of CSE,
Sikkim Manipal Institute of Technology,
Majhitar,Sikkim

Dr. M.K. Ghose,
Dean Academics,
Department of CSE,
Sikkim Manipal Institue of Technology,
Majhitar,Sikkim

ABSTRACT- Ontology is the specification of conceptualization. This paper provides concept about the Ontologies and its importance in advancement of the field of Artificial intelligence. It also focuses on some of the guidelines that should be followed while implementing it. It also focuses on necessary field where various ontology can be implemented and proposes a future work on the same.

I.INTRODUCTION

Ontology is defines as a set of primitives that is used to design the domain of knowledge. Ontology language is a formal language that is used to design ontologies. It allows the encoding of the knowledge about specific domain and often includes reasoning rule that supports the processing of that knowledge and they are declarative language. The Ontology separates the variables that is used for some set of computation and also build relationship between them. The Ontology is created to limit the complexity and to organize the information in the field of Artificial Intelligence, the semantic web, System engineering, Software Engineering, Biomedical Informatics, Library Science, Enterprise bookmarking and Information architecture. It can be applied to problem solving. Ontology reduces the conceptual or terminological confusion and brings shared understanding that can function in unifying framework for the different viewpoints and serves basis for:

a) Communication-among the system with different viewpoints and the needs.

b) Inter-operability-among systems achieved by translating between different modeling methods, paradigms, languages and software tools.

Ontology can be used as inter-lingua between different system. To translate language Li to Lj and vice versa ontology can be used to support the translation between different system and representation. There are three kinds of ontology:

a) Domain Ontology: It represents the concept with respect to domain. For example, the word 'BOW' has different meaning. An ontology in the domain of place is "a District in London" and ontology in the domain of weapons is "things to shoot projectiles with". Domain Ontology are often incompatible since it represents the concept in very specific and often eclectic ways.
b) Upper Ontology-It is usually applicable to wide range of domain ontologies. There are several standardized upper ontologies available for use, including BFO, BORO method, Dublin Core, GFO, OpenCyc/ResearchCyc, SUMO, the Unified Foundational Ontology (UFO) and DOLCE. WordNet, while considered an upper ontology by some, is not strictly an ontology. However, it has been employed as a linguistic tool for learning domain ontologies.

c) Hybrid Ontology-It is combination of both domain and upper ontology.

The formal language used to encode the ontology is ontology Languages. There are number of such languages for ontology.

I.I.OWL

Web ontology Language is the family of knowledge representation language authority ontology. It is an international standard for encoding and exchanging ontologies and is designed to support the semantic web. The concept of the semantic web is that information should be given explicit meaning, so that machine can process it more intelligently. It is an ontology for the web and is based on the description logics. Descriptive logics are a family of logics that are decidable fragments of first order logics. The logics focus on describing classes and roles ,and have a set-theoretic semantics.OWL are build upon a W3C XML standard for objects called the Resource Description Framework(RDF).RDF is closely related to semantic networks since it has a graph based data model with labeled nodes and directed, labeled edges. It is a very flexible model for representing data.It has a fundamental unit called statement and it corresponds to an edge in the graph.The RDF statement has three components: a subject, a predicate, and an object.

Subject: It is the source of the edges and must be a resource. Resource can be uniquely identifiable via Uniform Resource Identifier (URI) and it can be anything. Identifier are more often than not are Uniform Resource Locator (URL),URIs are more general than URLs.

Object: it is the target of the egde.

Predicate: It determines the kind of relationship established between the subject and the object.

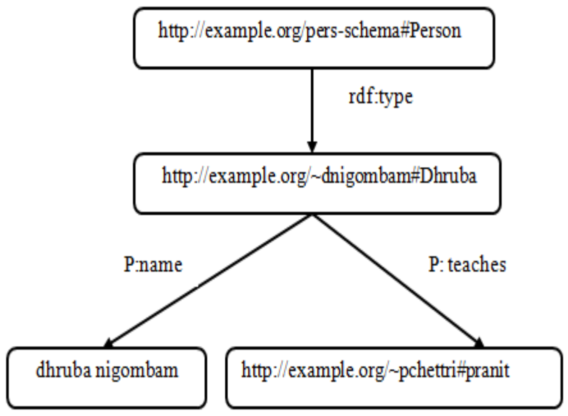
http://example.org/pers-schema#Person
rdf:type
http://example.org/~dnigombam#Dhruba
P:name
P: teaches
dhruba nigombam
http://example.org/~pchettri#pranit

Fig 1: example of RDF graph.

In the above figure, it shows an example of RDF graph. It contains three statements. One statement has subject http://example.org/~dnigombam#Dhruba, predicate p:teaches and object http://example.org/~pchettri#pranit. This all statement altogether represents that "Dhruba teaches Pranit".The statement with predicate p:name is an example of a statement tha has a literal value(i.e, "dhruba nigombam") as its object. This statement denotes that dhruba's name is "Dhruba nigombam". Here p:teaches and p:name are the example of qualified names. The third statement declares dhruba to be a person.

I.I.I.XML SERIALIZATION SYNTAX FOR RDF

The W3C recommendation defines an XML syntaxin order to exchange the RDF graphs. XML is a markup language. It uses tags to provide additional information about the text. The tags it uses are denoted by the angle brackets, such as the <p>, <img> and <a> tags in the Hypertext Markup Language (HTML). The main syntactic unit in XML is the element, which typically consists of a start tag (such as <p>), and the end tag (such as </p> and some contents enclosed between the tags. If it contains other element then it is called subelement. Element can have attributes which are name-value pairs. The attributes are listed inside of the element's start tag.

```
< rdf : RDF
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-
syntax-ns#
xmlns:p=http://example.org/pers-schema#>
<rdf:Description
rdf:about=http://example.org/~dnigombam#dhruba">
   <p:teaches
rdf:resource=http://example.org/~pchettri#pranit />
<p:name>dhruba nigombam</p:name>
<rdf:type
          Rdf:resource=http://example.org/pers-
schema#person" />
</rdf:Description>
</rdf : RDF>
```

Fig 2: RDF/XML syntax for the RDF graph in figure 1.

In the above figure, the rdf:RDF contains an rdf:Description subelement that is used to identify a resource and to describe some of its properties. All rdf:RDF elements encodes one or more RDF statements. As per above example the subject of each of the statements is a resource given by the "rdf:about" attributes and which has the URI http://example.org/~dnigombam#dhruba as it value. The rdf:Description element contains three property subelement and it encodes three statement.The first subelement is empty element with the qualified name p:teaches:based on the namespace declaration at the beginning of the document and this refer to the resource http://example.org/pers-schema#teaches and this is the predicate of the statement. The resources that is used by the predicate is known as property. To specify that "http://example.org/~pchettri#pranit" is the object of the statement we use rdf:resource attribute. The second subelement of the rdf:Description encodes such statement which have literals as objects. This element is a textual content. The statement has predicate "http://example.org/pers-schema#name" and object "Dhruba Nigombam". The third subelement of the rdf:Description is rdf:type By using the namespace declaration at the beginning of the document, we can find that it refers to the predicate http://www w3.org/1999/02/22-rdf-syntax-ns#type.This property in RDF allows to categorize resources. The rdf:resources attribute is used to specify the category such as "http://example.org/pers-schema#person".

I.I.II.OBO

The Open Biomedical Ontologies (OBO) is an ontology language that is often being used for modeling ontologies is life science. It is an effort to provide a way to organized knowledge for subsequent retrieval for shared use across different biological and medical domain. The majority of the ontologies in that repository are written in OBD Flat File Format. It is an ontology generally design for Gene Ontology (GO). The OBO language use simple textual syntax. It was designed to be compact, readable by human and easy to parse. The OBO Ontology is a collection of stanzas. Each stanzas describes one element of the ontology. It is introduced by the line containing a stanza name which identifies the type of the element being described. The stanza consists of lines, each of which has a tag followed by a colon, a value, and an optional comment introduced by "!".

```
[Term]
Id: GO:0000367
name: oocyte growth
is_a:GO:0000321 ! cell growth
relationship: part_of GO:0000459 ! oocyte morphogenesis
intersection_of: GO:0000310 !  growth
intersection_of: has_central_participant CL:0000212 ! oocyte
```

fig 3.example of OBO ontology.

OBO stanza defines the term GO:0000367 with the name oocyte growth. This term is the subclass of the term GO:0000321 (This term is named as cell growth).The term GO:0000367 has a part_of relationship to the term GO 0000459(which is named as oocyte morphogenesis) and GO:0000367 is defined as the intersection of GO:0000310 (which is names as growth) and of the relationship has_centrel_participant to CL:0000212 (which is named as oocyte).

```
[Typedef]
id: propreo:is_described_by
domain: propreo:chemical_entity
range: __Description123
```

The above stanza defines the relationship type (the OBO equivalent of a property) propreo:is_described_by. The terms propreo:chemical_entity and _Description123 are used as the domain and range, respectively, of the relationship type being defined.

```
[Instance]
id: propreo:water_molecule
instance_of: propreo:inorganic_solvent_molecule
property_value: propreo:is_described_by propreo:CHEBI_1234
```

Finally, the above stanza defines the instance (the OBO equivalent of an individual) propreo:water_molecule. The instance is a member of the term propreo:inorganic_solvent_molecule and has propreo:CHEBI_1234 for the value of the relationship propreo:is_described_by.

I.I.III.OntoUML

The OntoUML is an ontology which is well founded profile of UML for the conceptual modeling of domain ontologies. It has build with a lightweight expansion of UML metamodel. Guizzardi utilized UFO ontology for the improvement of the conceptual modeling language, with a target to resolve the overload, excessiveness and redundancy problems of the modeling language components. UFO (Universal Foundation Ontology) specifies the various catalogues of the ontology in the actual world, for example, things, situation, role and complex. These catalogues are classified into different natures such as rigidity, anti-rigidity and non-rigidity.The composition of UFO ontology is showed below:

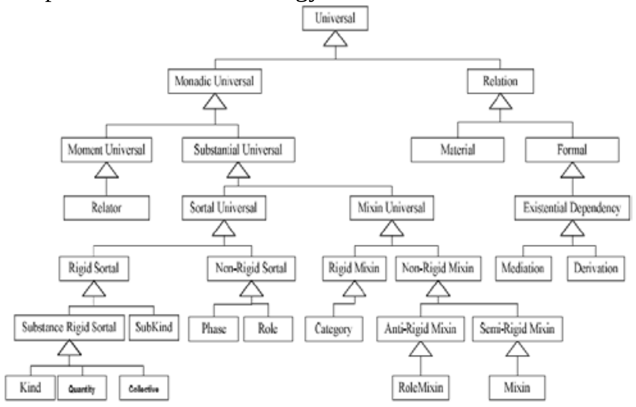

Fig 3. Composition of UFO[4].

The conceptual modeling in terms of ontology can be fulfilled in four steps:

i. Recognition of ontology concept and language component;
ii. Mapping the ontology concept onto the language components;
iii. Recognition of the relationship and bound between the ontology assumption and concept; and
iv. Transformation of ontology relationship by utilizing mapping method and formulation of modeling rules.

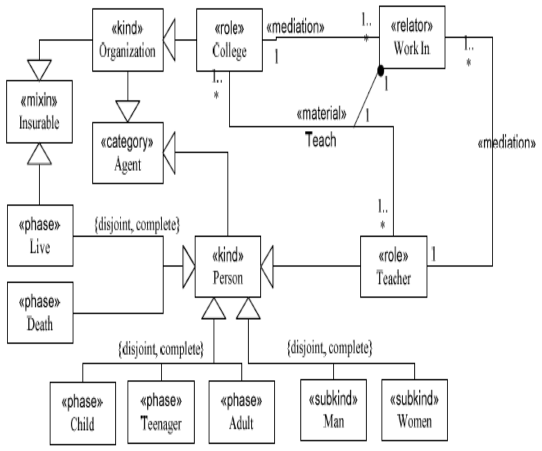

Fig 4: Example of conceptual model based ontoUML [4].

According to the original modelling language in terms of UFO concept in OntoUML, the explanation for UML structures is stated as follows:
«phase»: class-in-state, one kind of class-in-time, which is of anti-rigidity, the lifecycle of an entity class can be divided by different states, «kind»: entity class, e.g. functional complex, «relator»: class-in-relator, at a certain time, the existence of the example shall demand at least two entities on an outward basis, «collective»: entity class, e.g. collection object, «roleMixin»: class-in-mixed-role, which is featured by anti-rigidity and outward demanding, «mixin»: mixing class, the attribute for some entities are of built-in type, but for the others are not of built-in type, «quantity»: entity class, e.g. object with nominal quantity, «role»: class-in-role, one kind of class-in-time, which is of anti-rigidity, an object can be the example of different entity classes at different time, «subkind»: sub-class entity, which is of rigidity, and the entity with various concepts can be differentiated and marked, «mediation»: class-in-media-relations, representing the formal relationship between the relation concept and the continuous material concept, «Derivation»: derived class, representing the formal relationship between the relation concept and the material concept, «category»: class-in-category, which is of rigidity, concentrated with build-in attributes of various things, «material»: class-in-material-relationship, representing the material relationship between various entities.
Through expansion of UML and enhance the UFO ontology concept, clear and complete semantic mapping is established between OntoUML and the conceptual model, so that the correctness of the model ontology can be guaranteed.

I.I.IV.RIF

The Rule Interchange Format is a W3C Recommendation and is a part of the infrastructure for the semantic web along with the SPARQL, RDF and OWL. The design of RIF is based on the observation that there are many “rules languages” in existence and what is needed is to exchange rules between them.The RIF has three dialects:

i. Core-It is the common subset of the most rule engine and a “safe” positive datalog with bulltins. RIF-Core is a subset of both RIF-BLD and RIF-PRD.

ii. BLD-It stands for Basic Logic Dialect. It adds features to the core that are not directly available such as: logic function, equality in the then-part and named arguments. It corresponds to positive datalogs, that is ,logic programs without function or negations. It has a model-theoretic semantics.

iii. PRD-It stands for Production Rules Dialect. It adds action with the side effect in the rule conclusion. It can be used to model production rules. It has an operational semantics.

RIF focused on exchange rather than trying to develop a single one-fits-all rule language because, in contrast to other Semantic Web standards, such as RDF, OWL, and SPARQL, it was immediately clear that a single language would not satisfy the needs of

many popular paradigms for using rules in knowledge representation and business modeling.
For the example of RIF, we concern with the integration of the data about film and plays across the semantic web. Suppose, for example, that one wants to combine data about films from IMDb, the Internet Movie Data Base (at http://imdb.com) with DBpedia (at http://dbpedia.org). Both resources contain facts about actors being in the cast of films, but DBpedia expresses these facts as a binary relation (also known as predicate or RDF property). In DBpedia, for example, one can express the fact that an actor is in the cast of a film: starring(?Film ?Actor)-where we use '?'-prefixed variables as placeholders. The names of the variables used in this example are meaningful to human readers, but not to a machine. These variable names are intended to convey to readers that the first argument of the DBpedia starring relation is a film, and the second an actor who stars in the film. In IMDb, however, one does not have an analogous relation. Rather, one can state facts of the following form about actors playing roles: playsRole(?Actor ?Role)-and one can state facts of the following form about roles (characters) being in films: roleInFilm(?Role ?Film).Thus, for example, in DBpedia, one represents the information that Lily James was in the cast of Cinderella, as a fact starring(Cinderella Lily James).In IMDb, however, one represents two pieces of factual information, that Lily James played the role of Princess: playsRole(Lily James Princess) and that Princess was a character in Cinderella: roleInFilm(Princess Cinderella)

I.I.V.KIF

Knowledge Interchange format (KIF) is the computer oriented language that enable system to share and reuse the information from knowledge base system. It will have declarative semantics (that is the meaning of the expressions in the representation can be understood without appeal to an interpreter for manipulating those expression); and will expresses arbitrary statement in first order logic. It also supports non-monotonic reasoning. The language provides the representation of knowledge about the representation of knowledge. This allows us to make all knowledge representation decisions explicit and permits us to introduce new knowledge representation constructs without changing the language. It also provides for the definition of object, function , and relations. KIF is not intended as a primary language for interaction with human users and also not intended to be an internal representation for knowledge within computer programs or within closely related sets of programs.Typically, when a program reads a knowledge base in KIF, it converts the data into its own internal form (specialized pointer structures, arrays, etc.). All computation is done using these internal forms. When the program needs to communicate with another program, it maps its internal data structures into KIF. There are two types of KIF:

i. Linear KIF-In linear KIF, all expressions are strings of ASCII characters and, as such, are suitable for storage on serial devices (such as magnetic disks) and for transmission on serial media (such as phone Lines).

ii. Structured KIF-In structured KIF, the legal \expressions" of the language are structured objects. Structured KIF is of special use in communication between programs operating in the same address space.

The linear KIF is consists of 128 characters in the ASCII characters set. The 93 characters has a printed representation and rest character do not.The printed characters are as follows:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 ( ) [ ] { } < >
= + - * / \ & ^ ~ ' ` " _ @ # $ % : ; , . ! ?

The KIF originated in the Lisp application and it inherits its syntax from the Lisp. The structure KIF has a notion of word which is taken as primitive. An expression can be either word or a finite sequence of the expression. In structured KIF we use enclosing parenthesis to bound the items in a composite expression.

<word>::= a primitive syntactic object
<expression>::=<word>|(<expression>*)

The listed below are set of all words is divided into categories.
<indvar> ::= a word beginning with the character ?
<seqvar> ::= a word beginning with the character @
<termop> ::= listof | setof | quote | if | cond |
the | setofall | kappa | lambda
<sentop> ::= = | /= | not | and | or | => | <= | <=> |
forall | exists

```
<ruleop> ::= =>> | <<= | consis
<defop> ::= defobject | defunction | defrelation | := |
                          :=> | :&
<objconst> ::= a word denoting an object
<funconst> ::= a word denoting a function
<relconst> ::= a word denoting a relation
<logconst> ::= a word denoting a truth value
```

From these fundamental categories, we can make more complex categories, viz. variables, operators, and constants.

```
<variable>::= <indvar> | <seqvar>
<operator>::=<termop> | <sentop> |<ruleop><defop>
<constant>::=<objconst>                                  |
                           <funconst>|<relconst>|<logconst
                           >
```

The relationship between linear KIF and Structured KIF is most easily specified by appeal to the commom Lisp reader.

The following set of legal sentences in KIF is defined by BNF. In total, there are six types of sentences.

```
<sentence>::=<logconst>|<equation>|<inequality>|
                           <relsent>|<logsent>|
                           <quantsent>
<equation> ::= (= <term> <term>)
<inequality> ::= (/= <term> <term>)
<relsent>::=(<relconst> <term>* [<seqvar>])|
                  (<funconst> <term>* <term>)
<logsent> ::= (not <sentence>)|
                   (and <sentence>*)|
                   (or <sentence>*)|
                   (=><sentence>* <sentence>)|
                   (<=<sentence> <sentence>*)|
                   (<=><sentence> <sentence>)
<quantsent>::=(forall <indvar> <sentence>)|
                   (forall(<indvar>*) <sentence>)|
                  (exists<indvar> <sentence>)|
                  (exists(<indvar>*)  <sentence>)
```

Let's take simple example for representation of sentence "The father of john is either richard or peter" in KIF.

```
(forall (?x)
                  (=>(father john ?x)
                  (or (= ?x richard)
                  (= ?x peter))))
```

Everyone's marks must be greater than 50.

```
(forall (?x)
       (greater (marks ?x) 50))
```

II.APPLICATION

The application of ontology can be seen in many fields of technological advancement. It can be used beneficially in enterprise application. The existing example is SAPPHIRE (Health care) or Situational Awareness and Preparedness for Public Health Incidences and Reasoning Engines which is a semantics-based health information system capable of tracking and evaluating situations and occurrences that may affect public health. Other various project is related to ontology are Ontology Lookup service ,Gene Ontology Consortium, Sequence Ontology, Generic Model Organism Databases, Functional Genomics Data(FGED), Plant Ontology. Nowadays, several companies and projects rely on semantics to implement more advanced applications. Examples include semantic wikis, semantic social networks, semantic blogs, and semantic product descriptions. One particularly interesting application was the use of semantics to develop the language called Linked USDL (Unified Service Description Language) which enables to describe services (e.g. from healthcare, education, and cloud computing).The application of ontology can be seen in the field of robotics in designing the knowledge base which allow multiple robot to obtain not only the prior corpus of knowledge, but to augment it with the additional knowledge obtained by previous exploration by the other robots.

III.CONCLUSION AND FUTURE WORK

The Ontology can be used to developed a new generation information system. This can be used to enriched the system with full artificial intelligence. The one of the Hottest IT topic where ontology can be implemented is information Integration. This needs the merging and mapping of the information from different data source and preparing them in such a way that end user can use it. The Ontology can be deploy in the field of cloud robotics where robots can interact with each other and enables each robot to share and reuse their existing information/ knowledge.

IV.REFERENCES

[1] Jorge Cardosa ,Alexandre Miguel pinto,"The Web Ontology Language(OWL) and its Application",IGI global,page no:754-766,2015.

[2] D. Lorencik, P. Sincak," Cloud Robotics: Current trends and possible use as a service", • IEEE 11th International Symposium on Applied Machine Intelligence and Informatics, page no:85-88, January 31 - February 2, 2013.

[3] Guoqiang Hu, Wee Peng Tay and Yonggang Wen, “Cloud Robotics: Architecture, Challenges and Applications”, IEEE Network Magazine, pg. no.-21-27, May/June 2012.

[4]Y.Bin et. al,"Research on Consistency Checking of OntoUML Model",Advanced Material Research,vol.433-440,pg.no-2862-2867,2012.

[5] Dejan Pangercic, Benjamin Pitzer,Moritz Tenorth,Michael Beetz,” Semantic Object Maps for Robotic Housework - Representation, Acquisition and Use”, 2012 IEEE/RSJ International Conference on Intelligent Robot and System, Portugal, pg. no.-4644-4651,October 7-12, 2012.

[6] A. Bjorkelund, J. Malec, K. Nilsson, P. Nugues and H. Bruyninckx, "Knowledge for Intelligent Industrial Robots", AAAI Technical Report SS-12-02,2012.

[7] Rajesh Arumugam, Vikas Reddy Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran Foong Foo Kong, A.Senthil Kumar, Kang Dee Meng, and Goh Wai Kit,” DAvinCi: A Cloud Computing Framework for Service Robots”, 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA.

[8] G. Benet, F. Blanes, J.E. Simo, P. Perez,"Using infrared sensors for distance measurement in mobile robots",Robotics and Autonomous Systems 40 (2002) 255–266(Elsevier),27 March 2002.

[9] Matthew L. Ginsberg,"Knowledge Interchange Format: The KIF of Death",AI Magazine Volume 12 Number 3,pg. no-57-63,1991