International Journal of Emerging Technology and Advanced Engineering

Semantic Web Development Through Ontology Evolution

Rakesh Kumar Singh\textsuperscript{1}, Ranjan Singh\textsuperscript{2}
\textsuperscript{1}Scientist-D (Information Technology), G.B. Pant Institute of Himalayan Environment & Development, Kosi-Katarmal, Almora - 263643, Uttarakhand, India.
\textsuperscript{2}MCA Student, IGNOU, New Delhi, India.

Abstract—The ontology has been widely used in recent years in the field of Artificial Intelligence, computer and information science especially in domains such as, cooperative information systems, intelligent information integration, information retrieval and extraction, knowledge representation, and database management systems. The continual need to be better and better, for a business system is becoming a prerequisite for surviving in the highly changing business world. Although changes encompass several dimensions of a business system (e.g. people, processes, technologies), most of them are reflected on its IT infrastructure. Ontology-based applications are subject to a continual change. Thus, to improve the speed and to reduce costs of their modification, there is a need for an efficient ontology evolution system and a process model that fulfills them. The ontology evolution process enable the customisation of the ontology-evolution process to the current need and ensures the consistency of the underlying ontology and all dependent artefacts and also offers advice to the user for continual ontology reengineering. The implementation of the second generation of the WWW, the so-called Semantic Web, has made the idea of the large-scaled ontology-based applications in a business context real. Ontologies have recently become a key technology for semantics-driven modelling. The explicit representation of the semantics of data through ontologies enables applications to provide a qualitatively new level of services, such as verification, justification, gap analysis, etc.

Keywords—Ontology, Ontology Evolution, WWW, Semantic web, etc.

I. ONTOLOGY – AN INTRODUCTION

The word ontology is borrowed from philosophy, where an ontology is a systematic account of existence. For computer science, what "exists" is that which can be represented. Thus, in the context of computer science, the following definition is adopted:

"An ontology is a formal, explicit specification of a shared conceptualisation of a domain of interest."

Ontology can also be defined as "a set of representational primitives with which to model a domain of knowledge or discourse". Ontology is used in the fields of computer science as artificial intelligence, software engineering, semantic web, language processing.

In the field of computer science, ontology is the foundation of describing a domain of interest; it consists in a collection of terms organized in a hierarchical structure that shape the reality. The following are the components of ontology: 1. concepts, terms; 2. relations between concepts, terms; 3. Properties, attributes of the concepts; 4. Rules, axioms, predicates, constraints.

Conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose. Ontologies have set out to overcome the problem of implicit and hidden knowledge by making the conceptualisation of a domain explicit. Ontology is used to make assumptions about the meaning of a specific concept. It can also be seen as an explication of the context for which the concept is normally used. Moreover, everything (i.e., any knowledge-based system or any knowledge-level agent) is liable to some conceptualisation, explicitly or implicitly. Therefore, since there is consensus of terms, it is a shared conceptualisation.

Machine processibility is the basis for the next generation of the WWW, the so-called Semantic Web, which is based on using ontologies for enhancing with formal semantics. The purpose of an ontology is not to model the whole world, but rather a part of it - so-called domain. A domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc. Therefore, in order to define a domain, it is important to know what an ontology is for. Ontologies serve as a means for establishing a conceptually concise basis for communicating knowledge for many purposes. In order to achieve this, an ontology has to be a formal description of the meaning of concepts and relationships between them.

Ontologies are now ubiquitous in many enterprise-wide information systems: they are used in e-commerce, knowledge management and in various application fields such as bioinformatics and medicine. The ontology evolution is to interpret all requests for changes coming from different sources like users, internal processes and business environment, and to perform them on the ontology and depending artefacts while keeping consistency of all of them.
II. TYPES OF ONTOLOGY

The types of ontologies are as follows:

Domain Ontology: A domain ontology (or domain-specific ontology) represents concepts which belong to part of the world. Particular meanings of terms applied to that domain are provided by domain ontology. For example the word card has many different meanings. Ontology about the domain of poker would model the "playing card" meaning of the word, while an ontology about the domain of computer hardware would model the "punched card" and "video card" meanings. Since domain ontologies represent concepts in very specific and often eclectic ways, they are often incompatible. As systems that rely on domain ontologies expand, they often need to merge domain ontologies into a more general representation.

Upper Ontology: An upper ontology (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It usually employs a core glossary that contains the terms and associated object descriptions as they are used in various relevant domain sets. There are several standardized upper ontologies available for use, including BFO, 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.

Hybrid Ontology: The Gellish ontology is an example of a combination of an upper and a domain ontology.

III. TOOLS FOR ONTOLOGY

A great range of languages have been used for implementing ontologies during the last decade like Ontolingua, LOOM, OCML, FLogic, CARIN, OKBC, Telos, Cycl. Many of these languages had been already used for representing knowledge inside knowledge-based applications, other ones were adapted from existing knowledge representation languages, and there is also a group of languages that were specifically created for representing ontologies. These languages (which called "traditional" languages) are in a stable phase of development, and their syntax consists of plain text where ontologies are specified.

Recently, Web-based ontology specification languages have been developed in the context of the World Wide Web: RDF, RDF Schema, SHOE, XOL, OML, OIL, DAML+OIL and OWL.

Their syntax is based on XML, which has been widely adopted as a ‘standard’ language for exchanging information on the web, except for SHOE, whose syntax is based on HTML. From all these languages, RDF and RDF Schema cannot be considered as ontology languages, but as general languages for the description of metadata in the web. Most of these “markup” languages are still in a development phase; hence, they are continuously evolving. At the same time as these ontology languages have been developed, tools have emerged for creating, editing and managing ontologies written in the various languages like Protege 2000. Ontological tools usually provide a graphical user interface for building ontologies, which allows the ontologist to create ontologies without using directly a specific ontology specification language like OntoEdit, OilEd, WebODE, Ontolingua, Ontosaurus, LinkFactory.

IV. ONTOLOGY EVOLUTION

Ontology evolution can be defined as “Ontology Evolution is the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts”. Since a change in the ontology can cause inconsistencies in other parts of the ontology, as well as in the dependent artefacts, the ontology evolution has to be considered as a process. It encompasses the set of activities, both technical and managerial, that ensures that the ontology continues to meet organizational objectives and user’s needs in an efficient and effective way. The distinction between management, modification, evolution and versioning of the ontologies has been, in some cases, confused. The following characterization will be used by adapting the terminology from the database community:

- **Ontology Management** is the whole set of methods and techniques that is necessary to efficiently use multiple variants of ontologies from possibly different sources for different tasks. Therefore, an ontology management system should be a framework for creating, modifying, versioning, querying, and storing ontologies. It should allow an application to work with an ontology without worrying about how the ontology is stored and accessed, how queries are processed, etc.

- **Ontology Modification** is accommodated when an ontology management system allows changes to the ontology that is in the use, without considering the consistency.
Ontology Evolution is accommodated when an ontology management system facilitates the modification of an ontology by preserving its consistency.

Ontology Versioning is accommodated when an ontology system management allows handling of ontology changes by creating and managing different versions of it.

V. SIGNIFICANCE OF ONTOLOGY EVOLUTION

Research work carried out so far in the field of ontologies has focused on ontology construction issues. It is assumed that domain knowledge encapsulated in an ontology does not evolve in time. However, in a more open and dynamic business environment, the domain knowledge evolves continually. These changes include accounting the modification in the application domain or in the business strategy; incorporating additional functionality according to changes in the users' needs; organizing information in a better way, etc. Fig.1 depicts three basic sources that can cause changes in a business system:

- **The Environment**: The environment in which systems operate can change, thereby invalidating assumptions made when the system was built. For example, acquiring a new subsidiary in an enterprise adds new business areas and functionalities to the existing system.

- **Users**: User’s requirements often change after the system has been built, warranting system adaptation. For example, hiring new employees might lead to new competencies and greater diversity in the enterprise, which the system must reflect.

- **Internal Processes**: The business applications are coupled around the business processes that should be continually reengineered, in order to achieve better performances.

Ontology development is a dynamic process starting with an initial rough ontology, which is later revised, refined and filled in with the details. Further, the ontology must be used, and, during its period of use, the knowledge on which it relies will change and develop. An ontology that has not become rapidly obsolete must change and adapt to the changes in environments, users' needs, etc. Therefore, if an ontology aims at being useful, it is essential that it is able to accommodate the changes that will inevitably occur.

VI. DIFFICULTIES IN UNDERSTANDING ONTOLOGY EVOLUTION

Ontology evolution is not a trivial process, due to the variety of sources and consequences of changes. Building a ontology system has proven to be a difficult task, since there is almost a complete lack of suitable methodology, techniques and tools. Particularly, there are three challenges for the efficient realisation of the ontology evolution:

- **Complexity** - An ontology model is rich and, therefore, an ontology has an interwoven structure. Each change leads to a change specific workaround. Even when the effects of a change are minor, the cumulative effect of all changes realizing a user's request can be enormous.

- **Dependencies** - Ontologies often reuse and extend other ontologies. Changes in an ontology may affect the ontologies that are based on it. Therefore, changes between dependent ontologies are interrelated, and the immediate synchronisation between dependent ontologies is required. Obviously, the complexity of the ontology evolution increases with the number of dependent ontologies being evolved.

- **Physical Distribution** - Ontology development is a decentralized and collaborative process. Therefore, the physical distribution of the dependent ontologies has to be taken into account. The ontology evolution requires tracking the changes applied to an ontology and broadcasting the group of changes when an explicit request arises.
VII. SEMANTIC WEB AND ONTOLOGY

The Semantic Web is the next generation of the WWW, which is based on using ontologies for enhancing content with formal semantics. It is worth noticing that the real power of the Semantic Web is realised when many systems that collect Web content from diverse sources, integrate and process the information as well as exchange the results with other human or machine agents are created. Thereby, the effectiveness of the Semantic Web will increase drastically as more machine-readable Web contents and automated services become available. This level of inter-agent communication will require the exchange of "proofs". Furthermore, the Semantic Web will also be the basis for the Web of Trust, which will provide mechanisms to handle authentication, permission, and validation of attribution in a Web where, by design, anyone can contribute content, links, and services. Two important technologies for developing the Semantic Web are already in place: the eXtensible Markup Language (XML) and the Resource Description Framework (RDF). Expressing meaning of resources that can be found on the web is the main task of the Semantic Web. In order to achieve that objective, several layers of representational structures are needed which are presented in Fig.2. These layers have the following role:

- The XML layer represents the structure of data.
- The RDF layer represents the meaning of data.
- The Ontology layer represents the formal common agreement about meaning of data.
- The Logic layer enables intelligent reasoning with meaningful data.
- The Proof layer supports the exchange of "proofs" in an inter-agent communication enabling the common understanding of how the desired information is derived.
- The Trust layer ranges from digital signatures and security to social network analysis.

XML allows users to add arbitrary structure to their documents, but says nothing about what the structures mean. The meaning of XML-documents is intuitively clear to humans since the "semantic" mark-up and tags are domain-terms. However, computers do not have intuition. Tag-names per se do not provide semantics.

Data Type Definitions (DTDs) are a possibility to structure the content of the documents. However, structure and semantics are not always aligned, they can be orthogonal. Therefore, a DTD is not an appropriate formalism to describe the semantics of an XML document. The same holds for XML-Schema, it only defines structure, though with a richer language.

The Resource Description Framework (RDF) provides a means for adding semantics to a document. RDF is an infrastructure that enables encoding, exchange and reuse of structured metadata. RDF, in combination with RDFS, offers modelling primitives that can be extended according to the needs at hand. Basic class hierarchies and relations between classes and objects are expressible in RDFS. Some parts of the RDF and RDFS vocabularies are not assigned any formal meaning, and in some cases, notably the reification and container vocabularies, it assigns less meaning than one might expect.

Ontologies are well suited for describing heterogeneous, distributed and semi-structured information sources (e.g. XML documents) that can be found on the web or in the intranets. The ontologies, which already exist on the Semantic Web, range from simple taxonomies (such as the Yahoo hierarchy), to metadata schemes, to logical theories.

VIII. TYPES OF ONTOLOGY LANGUAGES

An ontology language is a formal language by which an ontology is built. There have been a number of languages for ontologies both proprietary and standards-based. Based on their formal semantics they can be split into two groups of languages:
Frame-based Ontology Languages: Their central modelling primitives are classes (known as frames) with properties (known as slots). A frame provides a context for modelling a class, which is generally defined as a subclass of one or more other classes, with slot-value pairs being used to specify additional constraints on instances of the new class. For example, the KAON (Karlsruhe Ontology and Semantic Web framework) ontology language, incorporates the essential modelling primitives of frame-based systems, being based on the notion of a concept and the definition of its superclasses and slots. It also treats slots as first class objects that can have their own properties (e.g. domain and range) and can be arranged in a hierarchy.

Description Logic based Ontology Languages: It has been developed in knowledge-representation research, and describe knowledge in terms of concepts (comparable to classes, or frames) and roles (comparable to slots in frame systems). An important aspect of these languages is that they have very well understood theoretical properties. Description logic enables reasoning with concept descriptions and the automatic derivation of classification taxonomies. There are now efficient implementations of description logic reasoners able to perform these tasks. For example, the Ontology Web Language (OWL) inherits from description logic both their formal semantics and efficient reasoning support.

IX. KAON (Karlsruhe Ontology) Ontology Language Definition

During the time of KAON development, RDF and RDFS were the de-facto standard languages for ontology modelling in the Semantic Web. Hence, these languages were chosen to be implemented by the platform. However, as development progressed, certain features of these languages were found to be inadequate for practice. Also, the languages in question have undergone a transformation themselves. Hence, the currently implemented ontology language is based on RDF(S), but contains many additions and changes to the standard. To avoid the pitfalls of self-describing RDFS primitives such as subClassOf, the KAON ontology language has the clean separation of modelling primitives from the ontology itself. Moreover, it provides means for modelling metaclasses and incorporating several commonly used modelling primitives, such as transitive, symmetric and inverse properties, or cardinalities.

As per KAON ontology language, all information is organised in so-called OI-models (ontology-instance models), containing both ontology entities (concepts and properties) as well as their instances. This allows grouping concepts with their well-known instances into self-contained units. An OI-model may include another OI-model, thus making all definitions from the included OI-model automatically available. The mathematical definition of OI-model is given below.

**Definition-1:** An OI-model (ontology-instance model) is a tuple $OIM := (E, INC)$, where:
- $E$ is the set of entities of the OI-model.
- $INC$ is the set of included OI-models.

An OI-model represents a self-contained unit of structured information that may be reused. It consists of entities (the set $E$ in previous definition) and may include other OI-models (represented through the set $INC$). Different OI-models may talk about the same entity, so the set of entities of these models need not to be disjoint. Note that the set of ontology entities $E$ contains the ontology entities and an instance pool associated with it.

**Definition-2:** An ontology structure of an OI-model is an 11-tuple:

$$O(OIM) := (C, P, S, T, INV, HC, HP, domain, range, mincard, maxcard),$$

where:
- $C \subseteq E$ is a set of concepts.
- $P \subseteq E$ is a set of properties.
- $R \subseteq P$ is a set of relational properties, i.e. relations.
- $A=PR$ is a set of attribute properties, i.e. attributes.
- $S \subseteq R$ is a subset of symmetric properties.
- $T \subseteq R$ is a subset of transitive properties.
- $INV \subseteq RXR$ is a symmetric relation that relates inverse properties:
  - if $(p1, p2) \in INV$, then $p1$ is an inverse property of $p2$
- $HC \subseteq CXC$ is an acyclic relation called concept hierarchy:
  - If $(c1, c2) \in HC$ then $c1$ is subconcept (or child) of $c2$, $c2$ is superconcept (or parent) of $c1$, $HC^*$ is the reflexive, antisymmetric and transitive closure of $HC$
- $HP \subseteq PXP$ is an acyclic relation called property hierarchy:
  - If $(p1, p2) \in HP$ then $p1$ is subproperty (or *child property) of $p2$, $p2$ is a superproperty (or parent property) of $p1$, $HP$ is the reflexive, nisymmetric and transitive closure of $HP$
- Function $domain: P \rightarrow 2^C$ gives the set of domain concepts for some property $p \in P$
The example shown in Fig. 3, the concept “PhD Student” is a direct child of the concept “Student”, and an indirect child of the concept “Person”.

In the example shown in Fig. 3, the concept “PhD Student” is a direct child of the concept “Student”, and an indirect child of the concept “Person”. Therefore, (“PhD Student”, “Student”) ∈ HC and (“PhD Student”, “Person”) ∈ HC*.

X. TAXONOMY OF ONTOLOGY CHANGES

Taxonomy of changes was adjusted in most other schema evolution research and represents the most frequently used set of changes. This approach has been adopted by taking into account the underlying ontology model. Changes related to the property hierarchy, cardinality constraints, included ontology models, etc., are specific for ontology models in general.

On the other hand, we use only a subset of this taxonomy by excluding changes related to the methods since they are not a part of the ontology model. Through the enrichment of the referent taxonomy with the additional changes related to the ontology model and through the elimination of changes that are not relevant to the ontology model, the "standard" set of schema changes is completely adjusted to the KAON ontology model. Moreover, the semantics of all changes (including the changes existing in the referent taxonomy) is defined on the basis of the underlying ontology model. For example, a property is a first-class citizen and therefore, it can exist without being attached to any concept.

A complete set of changes, determined by the KAON ontology language, is given in Table 1. These changes represent the ontology modifications at the lowest level of the complexity since they can add or remove one and only one entity in an ontology. Therefore, they build the backbone of an ontology evolution system.

Table 1.

<table>
<thead>
<tr>
<th>Meta Entities/ Meta Changes</th>
<th>Add</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>AddConcept</td>
<td>RemoveConcept</td>
</tr>
<tr>
<td>Concept Hierarchy</td>
<td>AddSubConcept</td>
<td>RemoveSubConcept</td>
</tr>
<tr>
<td>Property</td>
<td>AddProperty</td>
<td>RemoveProperty</td>
</tr>
<tr>
<td>Property Hierarchy</td>
<td>AddSubProperty</td>
<td>RemoveSubProperty</td>
</tr>
<tr>
<td>Property Domain</td>
<td>AddPropertyDomain</td>
<td>RemovePropertyDomain</td>
</tr>
<tr>
<td>Property Range</td>
<td>AddPropertyRange</td>
<td>RemovePropertyRange</td>
</tr>
<tr>
<td>Property Symmetric</td>
<td>AddPropertySymmetric</td>
<td>RemovePropertySymmetric</td>
</tr>
<tr>
<td>Property Transitive</td>
<td>AddPropertyTransitive</td>
<td>RemovePropertyTransitive</td>
</tr>
<tr>
<td>Property Inverse</td>
<td>AddPropertyInverse</td>
<td>RemovePropertyInverse</td>
</tr>
<tr>
<td>Max Cardinality</td>
<td>AddMaxCardinality</td>
<td>RemoveMaxCardinality</td>
</tr>
<tr>
<td>Min Cardinality</td>
<td>AddMinCardinality</td>
<td>RemoveMinCardinality</td>
</tr>
<tr>
<td>Instance</td>
<td>AddInstance</td>
<td>RemoveInstance</td>
</tr>
<tr>
<td>InstanceOf</td>
<td>AddInstanceOf</td>
<td>RemoveInstanceOf</td>
</tr>
<tr>
<td>InstProp</td>
<td>AddPropertyInstance</td>
<td>RemovePropertyInstance</td>
</tr>
<tr>
<td>OI-model</td>
<td>AddOI-model</td>
<td>RemoveOI-model</td>
</tr>
</tbody>
</table>

Table 1 shows that ontology changes can be thought of as additive and subtractive as is illustrated through the second and the third column, respectively.
The formal distinction of these two types of changes, adopted from, is given below:

An ontology change OntoCh is a total mapping between ontologies, i.e.,

\[ OIM_2 = OntoCh(OIM_1) \]

where \( OIM_1 \) and \( OIM_2 \) are ontologies.

We can use the following function:

The function \( extractSet(OIM,i) \) that gives \( i-th \) set of elements for a given ontology \( OIM \), \( 1 \leq i \leq 15 \)

There are 15 meta entities in the KAON ontology model. They are derived from the KAON ontology language definition and are shown in Table-1. For example, \( extractSet(OIM,1) \) returns a set of concepts \( C \) or \( extractSet(OIM,15) \) returns a set of included ontologies INC.

XI. SEMANTICS OF ONTOLOGY CHANGES

The purpose of the ontology evolution is to ensure that the application of ontology changes should result in an ontology conforming to the set of ontology consistency constraints. An ontology change preserves the ontology consistency if and only if it preserves all constraints of the underlying ontology model. However, applying an ontology change alone will not always leave an ontology in a consistent state. As shown in Fig.4, making the concept "PhD Student" a parent concept of the concept "Person" causes the inconsistency since the invariant \( IC_2 - Concept Hierarchy Invariant \) related to the cycle in a concept hierarchy would be violated. Namely, the concept "Person" would be in the same time the parent concept of the concepts "PhD Student" (through the concept "Student") as well as its child concept.

Above example shows that the ontology consistency has to be examined. Since verification in general concerns the correctness, the ontology verification is checking of the correctness of an ontology with the respect to the ontology consistency definition. There are two approaches to confirm ontology consistency:

The first approach first executes a change, and then validates the updated ontology to check whether it satisfies the consistency constraints. The second approach defines a respective set of preconditions for each change.

XII. SEMANTIC WEB DEVELOPMENT

According to Berners-Lee, the Semantic Web is:

"A web of data that can be processed directly and indirectly by machines."

Semantic Web is a network that takes the apparently infinite amount of data on the World Wide Web, but also connects this information with data in relational databases and other non-compatible archives, for example, the EDI system. Given that relational databases house most of the data from the company, the capability of the semantic Web technology to access and process, along with other data from websites, databases, XML documents, and other systems increases the amount of useful Data exponentially. Moreover relational databases already contain a large amount of semantic information. Databases are organized in tables and columns on the basis of relations between the tasks at home, and these relationships show the meaning (semantics) of data. Applications for data integration offer the possibility for the connection of different sources, which requires a distribution of posts between the different databases. Semantic Web, however, can be a machine to connect to another machine and the exchange of data and process effectively on the basis of built-in, available semantic information describing each of the resources. The fact is that the Semantic Web will make it possible to have all the information mentioned as a large database.

XIII. PRESENT STATUS OF SEMANTIC WEB DEVELOPMENT

Much work has made to attain the vision of a machine that could read the World Wide Web, the Semantic Web. Many researchers and practitioners have already begun to think that a new web, on the basis of ongoing research and development. Some industries, and its main players are coming from a wait and see attitude (Cardoso, Miller et al. 2005) to the real world expansion applications that will give them added value and competitive advantage.
To see how Semantic Web is in reality, you need only look at companies like Oracle, Vodafone, Amazon.com, Adobe, Yahoo and Google. All these organizations are working for an intelligent Web and certain provisions are already available on the Web. For example, Oracle has introduced the industry's first RDF (Decker, Melnik et al., 2000) management platform, orientation application areas such as life sciences (Stephens, Morales et al. 2006), data and content integration; integration of business applications, and integration of the supply chain. Current position of the Semantic Web will be described in research, education and industry as follows:

**Research Achievements in Semantic Web:**
- W3C have standards for XML, RDF, OWL and active working groups for these and semantic web services e.g. SPARQL and SWRL
- Ontology alignment system and framework
- Ontology alignment API's
- Editor of the development of ontology, alignment and querying
- Regular lectures and workshops with participants from more different nationalities and geographical location

**Education Achievement in Semantic Web:**
- Virtual institution for semantic web education content curriculum and events organized in a well manner and delivered in the semantic web infrastructure
- Dedicated curriculum for master program and PhD studies
- Annual Summer schools with increasing participation
- Active research teams in the world's best universities

**Industry Achievement in Semantic Web:**
- Oracle introduced the industry's first RDF management system, oracle 10.2 provides the facility to upload the RDF
- Adobe, Google, yahoo all these are organization are using intelligent web
- Jena API of HP, With HP investing more in semantic computing

**XIV. TECHNOLOGIES ASSOCIATED WITH SEMANTIC WEB DEVELOPMENT**

There is lot of technologies available to create semantics on the web. Some of these core technologies are briefly discuss as follows:

**Unified Modeling language (UML):**

UML provides a collection of models and graphs to describe the structural and behavioural semantics of any complex information system. Class and object diagrams are use to specify the semantic structure of the system. Object diagrams are the instance of class diagrams. Use case model and scenarios are use to collect the user requirement and functionality of the system. Scenario is the instance of use case. Activity diagrams to specify workflows. Interactions diagrams define that how group of object work together in some behaviour. State diagrams define the dynamic behaviour of an object in system. Physical diagrams define the implementation structure of the system.

**XML and XML Schema:**

These are the tools used to go beyond the fixed-oriented structure of the HTML page that provides vocabulary. With XML, it is possible to describe the structure of data and documents under a personal or Community defined vocabulary. These vocabularies can be a kind of semantics and support for an open exchange of data within communities and the tools to understand the vocabulary.

**RDF and RDF Schema:**

RDF is based on existing XML and URI (Uniform Resource Identifier) technology, a URI to identify all the resources, and the use of URI to make statements about resources. RDF statements are often called triple, which consists of a subject, predicate and object. It does not structure the syntax of the data, but defines semantic meaning for data on the web. Multiple semantic perspectives of the same data are possible. The technology is based on lower level technologies: URI's to identify web resources and Namespaces to identify different vocabularies. RDFS used to create the vocabulary that describes groups of RDF-related resources and the relationship between these resources. An RDF vocabulary permitted defines properties that can be allocated to the RDF resources within a given domain. RDFS types of resources can be created that share common characteristics. In Fig.5 "triples" is shown for RDF statement which has the subject, predicate and object. After defining the "triples" as graphically it can be coded either in RDF or XML.
Topic Maps:

Topic maps are a form of semantic web technology (in the broadest sense) and some work on interoperability between the W3C RDF / OWL / SPARQL family of standards for the Semantic Web. Topic maps define arbitrarily complex semantic knowledge structures and allow the exchange of information needed to build common and maintaining the index of knowledge. They provide a more comprehensive approach to RDF, in principle, because they are not limited to use on the Web.

XV. WEB ONTOLOGY LANGUAGE (OWL)

Ontology language gives greater machine interpretability of Web content to support for XML, RDF, and RDF Schema. They do this by providing additional vocabulary along with formal semantics. OWL is possible to implement a semantic description of a specific domain, indicating the concepts and relationships between concepts. There are three particular sublanguages: OWL Lite, OWL DL and OWL Full. We can define "ontology" in connection with the Semantic Web as a system that formally defined hierarchies and relations between different resources. Semantic Web ontology is taxonomy and a set of rules for the inference that the machines can make logical conclusions. Taxonomy, in this context is the system of classification, groups of resources into classes and sub-classes based on their relationship and common property. Since the taxonomy to express the hierarchical relationships between resources, OWL can be used to assign characteristics of classes of resources and allow their sub-classes that inherit the same characteristics.

XVI. PROBLEMS AND NEEDS ASSOCIATED IN SEMANTIC WEB DEVELOPMENT

Semantic Web development has been succeeded in its path to become a reality, and offers bright prospects in research, education and industrial areas for Semantic Web.

The following are the barriers and problems in Semantic Web development:

- **Acceptance:** Semantic Web has been changed the way we used to design, develop and deploy technologies on the Web today. It is important to emphasize that it is not just about changing on the other hand there are some arguments that it is the Semantic Web who will break the Internet model entirely. It requires too high degree of precision and trying to wrest control from the user and place in the hands of the Model Maker.

- **Privacy:** How many of us are willing to disclose to its semantic representation of knowledge system as large as the WWW? There are not many off course? And the issue becomes more severe in the case of organizations that not only individuals. The idea to write down each and all on-line resources requires people and organizations to expose their semantics it will mean more challenges for the Semantic Web development to ensure that adaptability for individuals, businesses and industry.

- **Confidence:** In the semantic web development we assume that the one based taxonomy in the first place know and can define to everyone's satisfaction, which means that taxonomy. But this is not how it works in the real world. The reliability of the suggestion of Semantic Web is doubtful. Although there are some fundamental building blocks that are in the Semantic Web development to ensure both the privacy and trust, but most of the requirements that these issues are missing is about the future of the W3C Semantic Web.

- **Taxonomy:** Taxonomy is the basic building blocks for the representation of knowledge on Semantic Web. Before that can come close to a visual Semantic Web development to become reality; we must have a taxonomy of taxonomy to define the form of taxonomy of work and their common terms mean and how to develop, define and disseminate their small issue. And then there is a need for another goal as taxonomy for Meta taxonomy also. All this makes it a real problem situation and needs to be addressed carefully.

- **Layer Structure:** The layers are the languages and have the syntax. The syntax of a layer can be an extension of the previous layer, for example, RDF syntax is part of the syntax of XML, RDF uses XML as syntax, but it is not correct that all XML documents are valid for RDF documents. Similarly OWL would have the same syntax as RDF and its semantics would be an extension of the semantics of RDF (S). This overlap is the relationship you can expect to be used by designers of RDF(S).
As we have read it through different statements on the RDF(S), so it clearly shows that we are extending the relationship from RDF to OWL, which leads it to the semantic web paradoxes. Since this stratification of OWL on top of RDF(S) shows that layering structure is an ambiguous.

- **Alignment**: Although there are many well-organized and serious efforts on the ground, leading to a fair system, frameworks and tools but ontology Adaptation is still a fundamental challenge for the semantic Web today. Main portion of the problem is very crucial and is linked to social, cultural and scientific diversity in the computer world.

- **Effect on Scenario**: Semantic Web technology and specially ontology are not always practicable in all sorts of settings and mechanisms. On some places like Political, Spiritual and Societal issues where ontology does not work every time. People passes different views on these issues and do not like to give an argument every time on these issues. Therefore, we must go back and look at these other methods or types for these kinds of circumstances.

**XVII. OUTLOOK AND FUTURE OF SEMANTIC WEB DEVELOPMENT**

Semantic Web development posses on its enormous success and has shown significant growth in different directions, from research to industry and institutions to business. As well as on the other hand, challenges are also enormous, especially these visual ideas about semantic web have to become a reality according to the Web level in the near future. Semantic Web technologies probably decisive role to play in the semantics of intranet enterprise level in the near future. It is easier for institutions to keep their system best in a more robust, free and formal way. Service-oriented architectures a semantic field that show a great potential for learning and business processes. The main Objective of the self awareness through e-learning is to get the knowledge without explanation and formal description. It is very central to the growing complexity and distributed nature of today's learning societies. Semantic web services and information technology have the potential to meet these requirements.

**XVIII. CONCLUSION**

Semantic Web leaves its impact on information technology and different research areas such as Knowledge Engineering / Management, Software Agents and Web Services. An important objective of the Semantic Web is to hand over most of the information to software agents that users are doing by themselves now a day on the web. Although the Semantic Web has achieved many milestones, but still it faces many challenges in the form of a vision to reality. Ontologies are becoming popular largely because of what they promise: a shared and common understanding that reaches across people and application systems. Undoubtedly, there is no single methodology for developing a learning ontology. Ontology design is a creative process and no two learning ontologies, designed by different people, would be the same. Learning ontology design choices are affected by the potential applications of the ontology and the designer’s understanding and view of the domain.

**REFERENCES**


