Data of Semantic Web as Unit of Knowledge

Archana Patel[1], Sarika Jain[1] and Shishir K. Shandilya[2]

[1]*National Institute of Technology, Kurukshetra, India*
[2]*Vellore Institute of Technology, VIT Bhopal University, India*
E-mail: archana_6170005@nitkkr.ac.in; jasarika@nitkkr.ac.in; shishir.sam@gmail.com

Received 25 January 2019;
Accepted 30 March 2019

Abstract

In service to the state of the art, advances are required toward redesigning the framework over which web applications are built. The semantic web lies at the intersection of web and machine understandable meaningful data, turning it into intelligent 'web of data'. The key requirement with any intelligent system has been to find a concrete knowledge representation that can make the inferences within time and space constraints; that is, reasoning effectively and efficiently within the resource constraints posed to the problem at one hand and with insufficient data as well as incomplete knowledge on the other hand. Various Knowledge representation schemes have been proposed in the literature, each having its limitation over the others. Ontology is the key component for semantic web engineering. Ontologies are conceptual knowledge bases providing a systematic and taxonomical description of the concepts and instances under consideration. Conceptual clarity in the computational representation of a concept is vital for holistic thinking and knowledge engineering. In order to meet the needs of an application/enterprise, knowledge should be presented taking care of all possible perspectives; and represented in a hierarchical structure with differing levels of granularity. This paper discusses about bringing all the manifestations of an ontological

Journal of Web Engineering, Vol. 17_8, 647–674.
doi: 10.13052/jwe1540-9589.1783

concept/decision under one umbrella; hence describing the resultant scheme as a unit of knowledge. By representing a concept as a knowledge unit, classical ontology is claimed to be sufficient in dealing with imprecise, vague and heterogeneous knowledge for real-world web applications; and portrays the capability to acquire fresh knowledge through its thorough interaction with the external world in a given working environment.

Keywords: Semantic Web, Ontology, Unit of Knowledge, Knowledge Representation scheme.

1 Introduction

The more information over the web generates more problems to find out the terms of specific contexts. Simply keyword based searching is not sufficient due to ambiguous behaviour of the words [1, 2]. The goal of the semantic web is to bring semantic or meaning in the information structure so that appropriate information according to the specific context can be find out. This is the place where knowledge representation formalisms play an important role. A common requirement of knowledge representation formalisms is to provide complete knowledge of concepts which is understandable by machine and human both. This objective can be achieved by only enriched knowledge representation scheme which has the capacity to represent a large amount of knowledge and to perform inference over the resultant knowledge base. A knowledge representation scheme has a set of semantic and syntactic conventions to explain and understand various entities like object, classes, event and properties [3]. Various knowledge representation schemes such as semantic network, frame, production rule, and logic have been presented in the literature [4]. These schemes differ from one another in one dimension or the other. Out of available schemes, production rule and logic are widely used to represent knowledge. Production rule defines the knowledge in "If_then" format. Later, Michalski and Winston [5] have added Unless operator in the production rule framing the Censor Production Rule (CPR). A CPR has the capability to reason with incomplete and uncertain information. A CPR is inept to hold the structure inheritance in the knowledge about real-world hereby would not grant control over one view of precision called specific decision. To address this problem, a new variation in production rule called Hierarchical Censored Production Rule (HCPR) as a knowledge representation scheme has been presented [6]. This scheme exploits the hierarchical structure of rules and makes it explicit. Any concept holds two types of attributes namely characteristic attributes and defining attributes. An instance of a concept should hold all the defining features of that concept under any circumstances; whereas characteristic features of the concept are cancellable and are allowed to be overridden by instances. An extended HCPR [7–9] presents this idea by adding two more operators to a rule corresponding to the characteristic properties of a concept.

In logic-based knowledge representation scheme, Description Logics (DLs) are widely used as an underlying representation structure. DLs are a family of class or concept based knowledge representation formalism [10]. They build complex concept and provides sound, decidable, tractable and complete reasoning services. DLs are most popular logic for ontology language such as OIL, DAML+OIL and web ontology language (OWL). The knowledge base of DLs is set of <T-box, A-box, R-box>. T-box contains terminological axioms or conceptual knowledge whereas A-box deals with factual knowledge or assertional axioms. R-box is rule box to perform reasoning. Nowadays OWL is widely used in knowledge representation and solves the problem of poor communication over the web by introducing ontologies. One popular application of ontology is semantic web where ontology play very important role during information exchange process. The current version of OWL is OWL2 which is equivalent to DL SROIQ(D). The syntax of OWL can be represented in various forms such as RDF/XML syntax, Turtle syntax, functional-style syntax, OWL/XML syntax and Manchester Syntax [11]. The main syntax of OWL2 is represented in RDF/XML format.

An Ontology is a vocabulary which consists of classes, relationships, axioms, data and object properties [12]. Through ontology, users and systems can interact and share information with each other. Ontologies are used in various domains to overcome the existing issue of heterogeneity and data integration. Despite the undisputed success of ontologies, unfortunately, conceptual formalism supported by the ontology structure is not sufficient for dealing imprecise, incomplete, temporal, spatial and large scale of data, which is inherent to most of the real world application domains [13]. This is because; the user does not take care of representing every entity as an atomic unit during the development of ontologies. We can reduce existing problems of classical ontology by representing and storing ontological concepts as units of knowledge. Once this is achieved; it will be easier for applications to deal with redundancy and consistency problems. Such applications will possess all features desirable in an intelligent system as efficient access to the stored knowledge, be able to provide context-sensitive inference power, and capacity to acquire and view a new piece of knowledge, to list a few. Our contributions in this paper are as follows:

- Define knowledge as a unit.
- Introduce ontological concept as a unit of knowledge.
- Provide Managerial Implications.

All the attributes, relationships and objects pertaining to a concept are encapsulated in a knowledge packet. A case study of emergency situations has been taken to explain the implications this proposal brings in. The rest of the paper is organized as follows: Section 2 discusses related work. In Section 3, we define a unit of knowledge and its ontological representation. Some foreseen managerial implications of the proposal have been presented in Section 4. Section 5 concludes the findings.

2 State of the Art

The semantic web also called the web of data represents knowledge by using ontologies; and allows machines to semantically process the acquired information and provides the meaningfully correct answer to imposed queries. The major problem in the development of any real time systems and applications lie within the processes of representing and eliciting knowledge. Knowledge representation schemes combine procedures and knowledge structure to exhibit implicit knowledge from stored knowledge structure [14]. There are four standard knowledge representation schemes namely semantic network, frame, logic, and rule. Semantic network captures knowledge in graph structure where nodes represent classes and edges between nodes depict the relationship between the classes. It is a semi-formal knowledge representation scheme and tradition semantic does not really lie [15]. Patel and Jain [4] described that frame scheme of knowledge representation lacks in semantic. There is no standard interpretation of links described between the specific frame and generic frame. Logics are widely used in knowledge representation languages. Knowledge Interchange Format (KIF) and Web Ontology Language (OWL) are two prevalent logic based languages [16]. Description Logics (DL) are a house of knowledge representation formalisms; they are a decidable fragment of First-Order Logic (FOL). OWL is widely used knowledge representation language based on DLs [17]. The current version of OWL is OWL2 which has qualified cardinality restrictions and significantly extended expressivity w.r.t. properties as compared to OWL [18]. Before OWL various languages for a description of semantic knowledge have been developed such as OIL, DAML+OIL, RDF and RDFs [19]. DAML+OIL [20] is a semantic markup language based on description logic designed for use in Web resources. McGuinness et al. [21] described that DAML+OIL has various boundaries to represent knowledge like it has no composition or transitive closure, less property constructors, no composition in data value, only unary and binary relationship, no concept of default and variables. Jain and Singh [22] stated that RDF is not a very expressive language. It has less degree to describe resources along with descriptions, domain and range constraints, cardinality and properties (transitive, inverse and symmetrical). There are also many obstacles in RDFs such as incapability to manifest equality and inequality, limited strength to express enumeration of property value. There are also other formal languages like LOOM, CycL, Ontolingua, KIF, and F-Logic used to manifest concepts terms [17].

Several extension of OWL has been proposed by various authors because it does not deal with uncertain, imprecise, temporal and spatial information. To play with uncertainty, various mathematical frameworks for extension of OWL such as fuzzy, possibilistic and probabilistic extensions with DL formalism have been reported in the literature. Stoilos et al. [23] proposed fuzzy OWL to capture imprecise and vague knowledge. They presented a reasoning platform called a fuzzy reasoning engine. Bobillo and Straccia [24] proposed a methodology to represent uncertain knowledge in ontology called fuzzy ontology using OWL2 annotation properties. They presented a plug-in for development of fuzzy ontology. Milea et al. [25] have presented a temporal extension of OWL called temporal OWL which is expressive fragment (SHIN(D)) of DL. They used layer approach and introduced three extensions in OWL namely concrete domain (allow to present restriction), temporal representation (introduce time points and its relation, intervals) and fluents/timeslices (implement perdurantist view and complex temporal aspects). Thomas et al. [26] told that thousands of ontologies are available online but there are no large scale infrastructures for storing, querying and reasoning these ontologies. They presented TrOWL which uses Quality Guaranteed Approximations to transforming, querying and reasoning OWL2 ontologies. Lee et al. [27] have proposed type2 fuzzy ontology model (T2FO) which rooted on interval type-2 fuzzy sets. They applied the proposed model for the representation of knowledge in the domain of diabetic-diet recommendation. The T2FO consists of T2F personal ontology, T2F food ontology, and T2F personal food ontology. Ausin et al. [28] presented an approach called TURAMBAR to interfuse OWL2 reasoning and expressibility power by using Bayesian networks to overcome the well-known limitations of OWL.

Bobillo and Straccia [29] explained that there are two ways to deal with vague information in OWL. First one is to extend the current language and the second one is to provide the procedure to represent such information within the language. They follow the second approach and proposed a methodology to build fuzzy ontology by using annotation property of OWL2. Malik et al. [30] proposed a new knowledge representation scheme called multimedia OWL. They introduced a probabilistic reasoning scheme by observation of media properties for belief propagation across the domain. Ding and Peng [31] proposed a probabilistic extension of OWL which is consistent with OWL. They used a Bayesian network and defined a set of rule for translation of annotated OWL into a directed acyclic graph of Bayesian network. Stoilos et al. [32] have proposed f-OWL which is a fuzzy extension of OWL and offered a translation method that reduces the inference problem of f-OWL into fuzzy DLs. Lera et al. [33] presented OWL-M to improve the matching process by representing multiple types of correspondences among ontology entities. Each function of OWL-M computes the similarity degree and defined an annotation set to explain the logical decision.

Production Rule (If-then) formalism does not pay attention of variable precision logic (VPL). It often provides several fragments of knowledge thus resulting in a large set of rules. Michalski and Winston [5] have added unless operator in production rule called CPR as a primordial computational and representational scheme for VPL in which specificity remains constant and only certainty varies. Hewahi [34] has applied security technique over CPR to ensure that censors condition are more secure and cannot be modified easily. Hewahi [35] has presented a rule structure called Concept Based Censor Production Rule (CBCPR), an extension of CPR. Every rule is written according to the behavior of the concept that specifies its job. He claimed that this structure of rule provides more certain decisions within a specified time. Bharadwaj and Jain [6] have presented HCPR (If, Unless, Generality, Specificity operators) as a knowledge representation scheme. HCPR deals with certainty as well as specificity, two main facets of precision. Later on, Jain et al. [7] presented an extension of HCPR called EHCPR where properties are divided into two categories namely defining and characteristic property. The characteristic properties of concepts are Has-part and Has-property. These properties may or may not hold in some situation. Jain and Jain [36] have provided learning techniques in EHCPR. They explained three major cases namely modification of existing EHCPRs, Restructuring of EHCPRs tree and adding of new EHCPR by using fission and fusion algorithm. Fission algorithm divides the tree into different clusters of EHCPRs tree whereas Fusion algorithm merged two EHCPRs trees. Jain et al. [37] described reasoning in the EHCPRs system along with the importance of default and constraints list of concept. They mentioned that EHCPRs is a system which provides reasoning with real-life problems. Jain and Jain [38] represented the structure of constraints and defaults in EHCPRs system. They described seven conditions where constraints have to be imposed on the system. Jain and Jain [14] added temporal, fuzzy and spatial information into EHCPR and termed it as generalized HCPR. EHCPR knowledge representation scheme is a collection of the goodness of all representation schemes. Jain and Jain [9] implemented multilingual thinking machine based on EHCPR and GHCPR knowledge representation scheme. They separated knowledge base into declarative knowledge and procedural knowledge. They also implemented EHCPR as an online globalized real-time intelligent system, which is multilingual, multimodal and context sensitive and it has the capacity to grow with domain knowledge.

Nowadays we deal with large scale data which are vague, incomplete and inconsistent in nature. Available representation schemes and their varied flavours like OWL, OWL2, fuzzy OWL, PR, CPR, HCPR, EHCPR, and GHCPR vary in level of power and expressiveness. Every knowledge representation scheme has demerits that degrade the performance of that scheme. For better representation with greater efficiency, a combination of two or more may be needed for structuring a particular knowledge base. EHCPR knowledge representation scheme provides access to concrete knowledge, spatial and temporal information of the entities but it is not well-established knowledge structure. On the other hand, the major problem of classical ontology is that it does not deal with incomplete, vague, inconsistent, temporal and spatial information. A knowledge representation scheme and language is not complete until it addresses these problems. There are two ways to reduce these problems, one is extend the existing language and second is to use the constructs of existing language. We follow latter approach to reduce the existing problem by introducing knowledge as a unit in OWL and enable to develop a realistic ontology. However, up now, no standard way of expressing a unit of knowledge in OWL has been provided. In this paper, we introduce knowledge as a unit and then express it in RDF/XML format according to available ontological constructs. This unit consists of three operators (Generality, Specificity, and Unless), three parameters (γ,δ and ω), two types of properties namely defining properties and characteristic properties (Has_part and Has_property) along with constraints list and default value, two type of encryptions (multimodal and multilingual) of concept and spatial and temporal details of instances.

3 Proposed Work

An intelligent web is gladly admissible to all, if it has a high degree of integrity, having undershot possible redundancy in representation and exquisitely consistent. The web should accumulate the power to earn new knowledge from the complex external world in a given working environment. The strength of automatic updating the knowledge base and to perceive new knowledge along with already procured knowledge is the main aspect of learning web. This objective can be achieved properly if the conceptual structure of every concept is represented as a unit. The soul of the semantic web is ontology that maintains a hierarchy of entities and provides inheritance mechanism. In this section, we explore and store knowledge as a unit into the ontological knowledge base. It has the capacity to expand horizontally i.e. rectification in already procured knowledge, as well as vertically, i.e. acquaintance of altogether new knowledge.

3.1 Defining Knowledge as a Unit

Rule, exception, and hierarchy of concepts offer a realistic description of the real world entities and a comprehensible knowledge representation in order to manage the complexity of large knowledge bases. Moreover, it gives inference power to the web and fetches the desired result according to different contexts. Unit of knowledge helps the semantic web to deal with incomplete and uncertain information and it can be incorporated in any knowledge representation model.

***Definition*:** A unit of knowledge to represent a concept/decision takes the form of following tuple:

$$\mathrm{D}\ [\mathrm{TE},\ \mathrm{AE},\ \mathrm{VE},\ \mathrm{PE}]\,(\boldsymbol{\omega}) = \ < DP\,(\gamma)\,,\ \mathrm{C}\,(\boldsymbol{\delta})\,,\ \mathrm{G},\ \mathrm{S},\ \mathrm{CP},\ \mathrm{I} >$$

A detailed view of this tuple is presented in Figure 1.

where,

DP is the set of m preconditions, which should be satisfied to draw the decision ***D***.

C is the set of n exceptions to the rule. Every exception is associated with a probability value denoted by $\delta_1, \delta_2 \ldots\ldots \delta_n$ respectively. The Unless operator refers to the list of exceptions.

G is the general concept of concept ***D*** and is referred to by the Generality operator.

If	**DP**	$\{DP_1{:}\omega_{P1}; DP_2{:}\omega_{P2}; \ldots\ DP_m{:}\omega_{Pm}\}\ \boldsymbol{\gamma}$	*{preconditions or defining properties (AND condition)}*
Then	**D**	[TE, AE, VE, PE], $\boldsymbol{\omega}$	*{decision/concept}*
Unless	**C**	$\{C_1{:}\delta_1; C_2{:}\delta_2; \ldots, C_n{:}\delta_n\}, \boldsymbol{\delta}$	*{exception to the rule (OR condition)}*
Generality	**G**		*{general concept}*
Specificity	**S**	$\{S_1,S_2, \ldots, S_k\}$	*{specific concepts (mutually exclusive set)}*
Has-Part	**CP**	$\{Pa_1{:}P_1C_1; Pa_2{:}P_2C_3;\ldots,Pa_p{:}P_3C_1\}$	*{characteristic parts of concept}*
Has-Property	**CP**	$\{Pr_1{:}P_3C_2; Pr_2{:}P_4C_1; \ldots Pr_q{:}P_1C_1\}$	*{characteristic properties of concept}*
Has_Instance	**I**	[Temporal Details, Spatial Details]	*{instances}*

Figure 1 Unit of Knowledge.

S is the next set of k specific concepts in the hierarchy and is referred to by the Specificity operator.

CP is the set of characteristic attributes associated with the concept ***D***. Characteristic attributes are of two types: physical parts and abstract properties. The Has_Part operator refers to the list of physical parts and the Has_Property operator refers to the list of abstract properties. P_1C_1 is the default value of part Pa_1 for the concept D chosen from the constraints list of part Pa_1 $\{P_1C_1, P_1C_2\ldots\}$.

I is the list of known individuals/instances of the concept ***D*** and is relegated with the Has_Instance operator.

All premises (DP_1, DP_2,$\ldots$.DP_m) are associated with a probability value (ω_{P1} , ω_{P2} , $\ldots$.ω_{Pm}) respectively which shows confidence of the truthfulness of DP. TE, AE, VE and PE are textual encryption, audio encryption, video encryption and pictorial encryption of the concept D. Factor γ is the precision of the decision with 0-degree of strength (if-then relationship), factor δ is the precision of the decision with 1-degree of strength (also considering exceptions) and factor ω is the precision of the decision with 2-degree of strength (also considering hierarchy).

Now, we discuss the various intrinsic representational aspects a concept must possess in order for it to be called complete in all respects.

a) Interpretation of Properties

Every concept is embellished by two sets of properties: one which is necessary for an object to be created as an instance of that concept, and other which may be cancellable for some instance or change its default value. The first is termed as the defining set of properties and the second is termed as the characteristic set of properties.

- Defining Properties relegated with the If operator, must be true and cannot be altered in any circumstances if an individual or instance satisfies that particular concept or class. By default, each class or concept derives defining properties of its super class or parent concept up in the hierarchy till the root. At any level, all defining properties of only one node will be true.
- Characteristic properties of a concept mentioned with the Has_Property and Has_Part operators, usually hold for an instance but are not mandatory. Every property in the system has an appropriate default value and constraints list. Characteristic properties for a concept in the hierarchy may or may not be derived. These properties can be cancelled or new properties can be added in the hierarchy. For example the concept "Emergency" has Has_Property operator with property "warning" with constraints list {yes, no}, and default value say no. An instance can override default value by the acquired value of "warning" property for that instance. So wherever values are assigned they will be verified for range or type mismatch. If the user gives integer value for "warning" property then it will not be granted and reasonable action has to be taken by the system.

b) Constraints List and Default value

Constraints and default value stipulate the modality of real-world problems. Constraints represent relations, rules, requirements, conventions, natural laws and principles that appoint the concept to be preserved. The default value is the most epochal value of object it draws from the constraint list. This value can be overridden. The different types of constraints and default that can be imposed on a concept treated as a knowledge unit are mentioned below:

- Constraints on characteristic properties of a concept: The characteristic properties like Has_Part and Has_Property of a concept imposes a constraints list and the concept draws default value from it. The Constraints list is a set of values or range of qualitative or quantitative detail of a concept. Instances will take values from constraints list and new constraint can be added at any time. Every concept occupies appropriate default value for every part and property from out of its constraints list.
- Constraints on individual of class: All individuals or instances catalogued in the operator 'Has_Instance' inflict constraints on the member set of that class or concept.

- Constraints on concept/decision: All censors of class relegated with operator ‘Unless’ interpolates constraints on the decision. When all premises of ‘If’ operator is satisfied and Unless is not checked then a decision will be less certain.
- Constraints on precision: Precision depends on the degree of certainty and specificity. Certainty and specificity work like two faces of coin, means certainty is inversely proportional to specificity. Suppose constraint on precision = 0.5 then decision will have moderate certainty and specificity. The value of precision should be lie between [0, 1].
- Constraints on resources: Available resources impose constraints on the decision. If resources are less then less number of the level will be explored whereas in case of high resources all level of the hierarchy will be explored and the result will be more specific and less certain.
- Constraints on user requirement: There are different types of users such as very high priority user, high priority user, medium priority user, low priority user, and very low priority user etc. These users have different constraints on the decision. For example; high priority user may have surplus resources and a high threshold of precision.
- Constraints on reasoning: Constraints in term of time and memory, also imposes at the time of reasoning. The constraints set defines how deep to go in the hierarchy? How many exceptions will be checked? When stopped the execution? For the same query, the system can response different behavior according to the applied constraints.

All these constraints and default values are dynamic in nature and can be changed at any stage of reasoning.

c) Exceptions to the rule

Exceptions manage the degree of uncertainty in the representation of knowledge. When we deal with complete information then all exception can be easily recognized and assigned appropriate values. In the case of incomplete information, we select one value for all unknown exceptions. For a more certain answer it is mandatory to take care of exceptions which are an unusual condition of concept. Exceptions in a unit of knowledge are managed by Unless operator. This operator has two logical aspects

- Expositive aspects: allow to explore certain expectations
- Control aspects: deploy a variety of problem-solving schemes

From an inference point of view, Unlessoperator acts like a Switch or Exclusion-OR operator that validate or invalidate the decision. Every censor like C_1, C_2,. . . ,C_n is associated with numeric values (δ_1,. . . , δ_n) and regarded as a probability value of the respective censor. To supplement for unknown exceptions namely UNK is proposed and be given a δ value. These values may affect the strength of final decision in the presence of exceptions.

d) Taxonomic Knowledge

Taxonomical representation is a powerful way to organize knowledge. The basic idea of taxonomy is that some item is subsidiary to another item. Classes are inserted in taxonomy and more specialized class inherits all properties of the more general class automatically which is directly subordinated in the taxonomy. The more general class is called super-class and specialized classes are called sub-classes. Inheritance property of taxonomical knowledge representation allows to inherit properties from multiple super-classes. Specificity and generality are two operators which are used to derive inheritance in the hierarchy. Taxonomical representation of knowledge provides two major benefits:

- When we move towards the bottom of the taxonomy, the class becomes increasingly specific. If the class becomes more specific, then a number of preconditions in its If operator increases (some are directly provided, rest are inherited). The taxonomical representation of knowledge inherits all preconditions of parent classes automatically. So there is no need to mention all such elements explicitly again and again. Hence redundancy is minimized in the taxonomical representation of knowledge.
- At any level, if any exception of a concept is true in the taxonomy, its specific classes and next level need not be explored hence reducing the searching complexity.

e) Precision of Decision

Precision is the strength of the decision which varies due to constraints on exceptions and the levels of hierachy. A concept is more certain if all premises are satisfied under the surveillance of all censors. The precision of decision may be calculated with varying degrees of strength:

- Precision with 0-degree of strength (γ): The parameter γ is the measurement of the strength of 'If' relationship between premises and decision. It has a constant value.
- Precision with 1-degree of strength (δ): This determines the relationship between If, Then and 'Unless' operators. It depends on 0-degree of strength of precision and all censors. The value of δ should be between [0, 1].

$$\delta = \gamma + \text{Summation of all } \delta'\text{s} \tag{1}$$

For useful implication, the value of δ should lie between $0.5 < \delta \leq 1$.

- Precision with 2-degree of strength (ω): Every concept has a certainty factor which determines the strength of the decision as a whole. The value of this parameter is calculated as [4]:

$$\omega\ (i) = \min\left(\omega\ (i-1)\ , \omega_{P1}\ ,\ \omega_{P2}, \ldots, \omega_{Pm}\right) \times \delta_{\mathrm{i}} \tag{2}$$

where, $\omega\ (i-1)$ is the precision with 2-degree strength of super-class. We always consider $\omega = 1$ for the root node of the hierarchy.

f) Multimodal and Multilingual Encryption

Multimodal means, knowledge modeled in all possible forms; text, multimedia (image, audio and video). Multilingual means that the knowledge about any concept is depicted in the language of choice of the user. For worldwide acceptance of a concept across applications over the globe, it is required to be modeled in all possible encryption and in all languages. Multilingual and multimodal encryptions are two aspects to globalize the concepts. Globalization of concepts may be represented according to Equation (3). This equation shows that the concept is stored in variable format and at the time of localization only variable will be replaced by desired language. Internationalization is making the application locale independent and localization is adapting an application to a specific locale (N). Only the translation of concepts into the local language is not sufficient. It should be incorporated culturalization of that country. For example: "10/31/2001, 3:24 PM" is acceptable for Indian user but not for a German user. They expect to see "31.10.2001, 15:24". Equation (4) depicts the localization of concept.

$$Globalisation = Internationalisation + N * Localisation \tag{3}$$

$$Locatisation = Translation + Culturalization \tag{4}$$

Figure 2 shows the multi-encryption of concept. Every concept has the following types of encryptions: multilingual text encryption (TE), multilingual audio encryption (AE), multilingual video encryption (VE) and picture encryption (PE). Every encryption is a linked list of the dictionaries of various languages. Text$_1$, Text$_2$ ...Text$_n$ are different synonyms of the concept. Different forms of video and audio of the concept are stored. Picture$_1$, Picture$_2$ ... and Picture$_n$ are the different pictures of the same concepts at different angles.

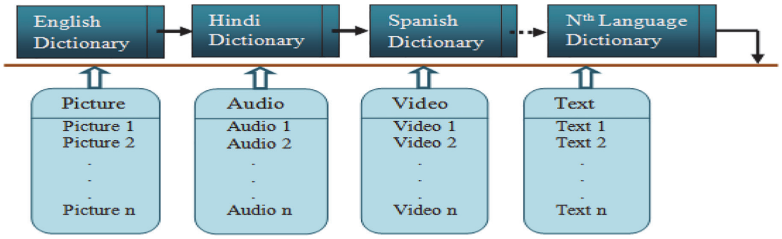


Figure 2 Multi-Encryption of concept.

g) Temporal and Spatial Details

Temporal details provide reasoning with respect to time whereas spatial details infer information according to the present state of the instance. All instances are stored in the instance base and all concepts are part of the concept base. At the time of instance creation, temporal and spatial information of instances will be stored and it cannot be deleted from the system. They can be moved or refined to archive or also be moved to concept base. Whenever an instance in the instance base is used during reasoning then it can be updated. Suppose, Army_Camp_2016, an instance of Terrorist_Attack has increased the value of the property has_injured say 122 then this information will be updated into the instance base with change in the attributes 'modification time' and 'has_Injured'. An instance has a serious impact until it is alive. Suppose below-mentioned data item is in the insatnce base:

Lashkar-e-Taiba
Instance_Of *Terrorist_Organization*
Has_Attributes *has_injured=122, Time_Of_Creation:26/11/2008 7:15:25*

After the death of instance of a terrorist organization, there is no need to maintain the record of that organization in the instance base. So, an instance Lashkar-e-Taiba will be withdrawn from the instance base and will be placed in the archive. Those facts who are part of long term memory are always kept into the concept base but an instance Lashkar-e-Taiba will never exist in the concept base because it is temporal short-lived fact and after the death of Lashkar-e-Taiba organization, it has to be put into the archive from the instance base. These details provide a variable response for the same query, based on variable context say time, location, and user background. Section 4 shows how contexts are changed for the same query.

3.2 Representing Unit of Knowledge in Machine

Thinking, learning, and accomplishment of any task by machine are possible only if knowledge is represented as a unit. The bottleneck problem of the machine is the requirement of devising a generalized knowledge representation scheme which can also facilitate general procedures for designing of multilingual, context-sensitive user interface, reasoning (top-down and bottom-up) and all possible learning methods. Representing knowledge as a unit is an attempt to develop a realistic ontology with the aim of enabling artificial intelligence (AI) applications to comply with human-like reasoning. This storage provides benefit to three dimensions (representation, learning, and reasoning) of the AI system. Figure 3 depicts a unit of knowledge in the domain of emergency.

Every concept carries its complete knowledge. If some event has properties sudden nature, property damaged and human harmed then it is an emergency situation. This emergency situation can have characteristic properties- noOf_deaths, noOf_injured, no warning and some location

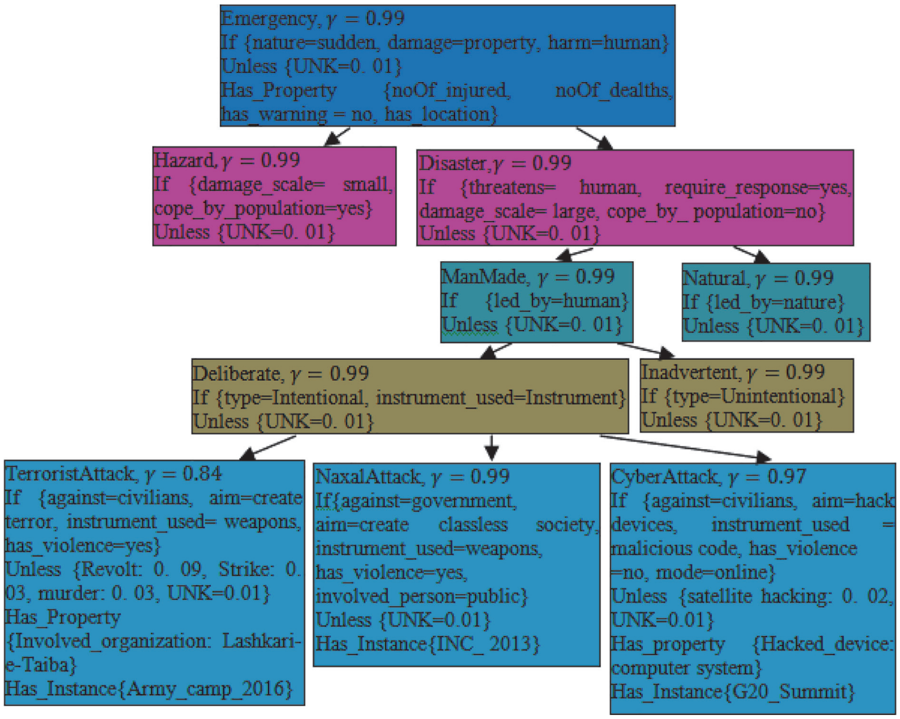

Emergency, $\gamma = 0.99$
If {nature=sudden, damage=property, harm=human}
Unless {UNK=0. 01}
Has_Property {noOf_injured, noOf_dealths, has_warning = no, has_location}
Hazard, $\gamma = 0.99$
If {damage_scale= small, cope_by_population=yes}
Unless {UNK=0. 01}
Disaster, $\gamma = 0.99$
If {threatens= human, require_response=yes, damage_scale= large, cope_by_ population=no}
Unless {UNK=0. 01}
ManMade, $\gamma = 0.99$
If {led_by=human}
Unless {UNK=0. 01}
Natural, $\gamma = 0.99$
If {led_by=nature}
Unless {UNK=0. 01}
Deliberate, $\gamma = 0.99$
If {type=Intentional, instrument_used=Instrument}
Unless {UNK=0. 01}
Inadvertent, $\gamma = 0.99$
If {type=Unintentional}
Unless {UNK=0. 01}
TerroristAttack, $\gamma = 0.84$
If {against=civilians, aim=create terror, instrument_used= weapons, has_violence=yes}
Unless {Revolt: 0. 09, Strike: 0. 03, murder: 0. 03, UNK=0.01}
Has_Property {Involved_organization: Lashkari-e-Taiba}
Has_Instance{Army_camp_2016}
NaxalAttack, $\gamma = 0.99$
If{against=government, aim=create classless society, instrument_used=weapons, has_violence=yes, involved_person=public}
Unless {UNK=0.01}
Has_Instance{INC_ 2013}
CyberAttack, $\gamma = 0.97$
If {against=civilians, aim=hack devices, instrument_used = malicious code, has_violence =no, mode=online}
Unless {satellite hacking: 0. 02, UNK=0.01}
Has_property {Hacked_device: computer system}
Has_Instance{G20_Summit}

Figure 3 Domain of emergency with complete knowledge in each concept.

mentioned. The precision with 0-level strength of rule is 0.99 and there is no foreseen exception to the rule. To supplement for possible incomplete exception list, an exception, namely UNK is there each node with δ value as 0.01. In the taxonomical representation, 'Emergency' is further classified into two subclasses namely Hazard and Disaster. The set of defining properties of Hazard {damage_scale=small, cope_by_population=yes} and Disaster {threatens=human, require_response=yes, damage_scale=large, cope_by_population=no} are mutually exclusive. Disaster can be Man-Made and Natural. The ManMade disaster can be happened deliberately or inadvertent. 'TerroristAttack', 'NaxalAttack' and 'CyberAttack' are three subclasses of deliberate Man-Made disasters. The characteristic property Involved_organization for 'Terrorist Attack' has a default value Lashkar-e-Taiba that has been taken from the constraints list{Al-Qaeda, Hizbul_Mujahideen, Jaish-e-Mohammed, Lashkar-e-Taiba}. The characteristic property Hacked_device for 'Cyber Attack' has a default value computer system that has been taken from the constraints list {computer system, network, technology dependent enterprise}

3.2.1 Knowledge Base of OWL2

The knowledge base of OWL2 has Terminological Box (T-Box) and Assertion Box (A-Box). T-Box contains conceptual knowledge and properties of the concept and refers to the concept base, whereas A-Box contains factual knowledge and instances and refers to the instance base. According to a unit of knowledge, now knowledge base contains the following information (Figure 4) into T-Box and A-Box. Temporal, spatial details and confidence values of all DPs of defining properties will be changed when an instance created. All DPs for any instance act as future case base. This will be helpful in answering future queries regarding any type of attribute of at instances.

OWL2 is developed for the use in applications that need to process or transfer the content of information instead of only presenting information to a human. OWL2 provides greater machine interpretability of web content by defining the relationship between vocabularies along with formal semantics. A concrete syntax is required in order to store OWL2 ontologies. OWL2 language supports various syntax such as RDF/XML (RDF triples with XML serialization), Trutle (provides easy way of writing and reading RDF triples), Functional-style syntax (closely resembles OWL2 syntax and specify semantics, profiles and mapping to exchange syntax), OWL/XML (provides an XML serialization for ontologies of OWL2) and Manchester (human readable

T-Box

Name of the concept with all encryption, Defining properties whose DP values will be determined at run time, Characteristic Physical Property, Characteristic Abstract Property, Unless operators with their δ values, multimodal encryption Precision with 0–degree of strength (γ), Precision with 1–degree of strength (δ) determined at run time, Precision with 2–degree of strength (ω) determined at run time, Super class, List of Subclass

A-Box

Instances, Temporal Details, Spatial Details, characteristic attributes specific to instance

Knowledge Base = T-Box + A-Box

Figure 4 Knowledge base.

```
<owl:Class rdf:about="http://www.semanticweb.org/lab/ontologies/2019/0/untitled-ontology-515#Emergency_ γ
=0.99"><rdfs:commentxml:lang="en">Emergency has sudden nature</rdfs:comment></owl:Class></rdf:RDF>
```

Figure 5 RDF/XML representation of concept 'Emergency'.

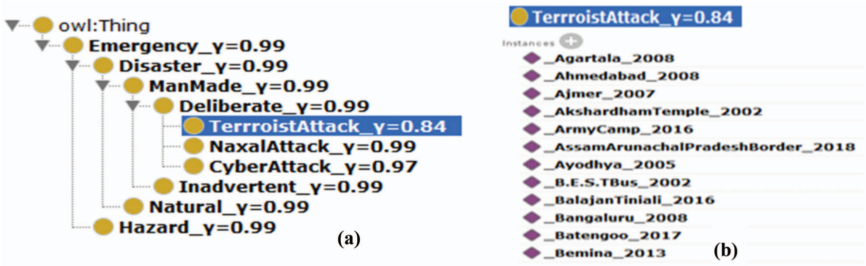
and compact syntax with a style close to frame language). The primary exchange syntax for OWL2 is RDF/XML.

3.2.2 Encoding Knowledge in OWL2

We have encoded the taxonomic structure of Figure 3 in RDF/XML format of ontology that is depicted in Figure 6. The top node of any concept called the general concept and a bottom node of that concept is called a specific concept. Figure 5 shows encoding of 'Emergency' concept with the precision of 0-degree strength and annotated via a comment in the English language into RDF/XML format.

Figure 6(a) shows that 'Emergency' is a general concept for 'Disaster' and 'Hazard' whereas 'ManMade' and 'Natural' are specific concepts of disaster. The precision with 0–degree strength of each concept is stored along with concept name. All stored instances of 'TerroristAttack' are shown in part b of the Figure 6.

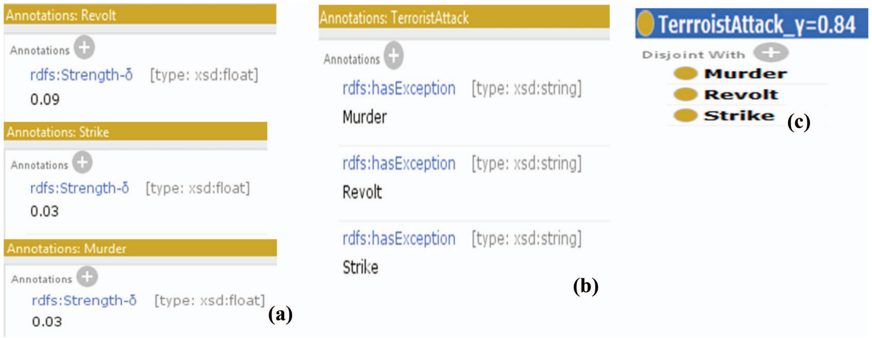
Exceptions to the rule are handled by Unless operator. All censors are stored below the 'Thing' class with an estimated probability value of censors. Censors (like Revolt, Strike, Murder) have all defining properties of the class (like TerroristAttack) with some new defining properties (like



owl:Thing
Emergency_γ=0.99
Disaster_γ=0.99
ManMade_γ=0.99
Deliberate_γ=0.99
TerrroistAttack_γ=0.84
NaxalAttack_γ=0.99
CyberAttack_γ=0.97
Inadvertent_γ=0.99
Natural_γ=0.99
Hazard_γ=0.99

TerrroistAttack_γ=0.84
Instances
_Agartala_2008
_Ahmedabad_2008
_Ajmer_2007
_AkshardhamTemple_2002
_ArmyCamp_2016
_AssamArunachalPradeshBorder_2018
_Ayodhya_2005
_B.E.S.TBus_2002
_BalajanTiniali_2016
_Bangaluru_2008
_Batengoo_2017
_Bemina_2013

Figure 6 (a) Concept with "0-degree strength" (b) Instances of TerroristAttack.



Annotations: Revolt
Annotations
rdfs:Strength-δ [type: xsd:float]
0.09

Annotations: Strike
Annotations
rdfs:Strength-δ [type: xsd:float]
0.03

Annotations: Murder
Annotations
rdfs:Strength-δ [type: xsd:float]
0.03

Annotations: TerroristAttack
Annotations
rdfs:hasException [type: xsd:string]
Murder
rdfs:hasException [type: xsd:string]
Revolt
rdfs:hasException [type: xsd:string]
Strike

TerrroistAttack_γ=0.84
Disjoint With
Murder
Revolt
Strike

Figure 7 (a) Censors with their δ values (b) All exceptions of TerroristAttack (c) Disjoint Relationship.

involved_person=public). We have assigned a disjoint relationship between classes and its censors. Figure 7 depicts all exceptions of TerroristAttack with their estimated δ values and disjoint relationships.

Ontology provides only data and object properties where data properties assign data to the instances and object properties link two classes according to domain and range. Defining and characteristic properties can both be data or object property. We have encoded set of defining properties by using rules in the ontology. Whenever an instance is created, the rule will be fired and if rule is satisfied then desired action will be displayed. For characteristic property, we simply use data and object property. Equation (5) shows the rule for defining properties of the concept 'Emergency'. Whenever an instance I of class 'Emergency' is created then it has to satisfy this rule.

$$nature(?I, ?sudden) \land damage\,(?I, ?property) \land harm\,(?I, ?human)$$
$$\implies has\_Instance(?Emergency,\ ?I) \tag{5}$$

```
<owl:DatatypePropertyrdf:ID="has_warning"><rdfs:range><owl:DataRange><owl:oneOf><rdf:List><r
df:firstrdf:datatype="&xsd;string">yes</rdf:first><rdf:rest><rdf:List><rdf:firstrdf:datatype="&xsd;string
">no</rdf:first><rdf:restrdf:resource="&rdf;nil"/></rdf:List></rdf:rest></rdf:List></owl:oneOf></owl:D
ataRange></rdfs:range></owl:DatatypeProperty>
```

(a)

```
<noOf_injured rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</noOf_injured>
```

(b)

Figure 8 (a) Constraint list of property has_warning (b) Storage of value noOf_injured.

```
<owl:Classrdf:about="http://www.semanticweb.org/lab/ontologies/2019/0/untitled-ontology
512#TerroristAttack"><rdfs:labelxml:lang="es">ataqueterrorista</rdfs:label><rdfs:labelxml:lang="de">Terro
ranschlag</rdfs:label>
</owl:Class></rdf:RDF>
<has_picturerdf:datatype=http://www.w3.org/2001/XMLSchema#string>C:\Users\Archana\Jan2019\TA.jpg
</has_picture>
<has_audiordf:datatype=http://www.w3.org/2001/XMLSchema#string>C:\Users\Archana\Jan2019\TA1.mp3
</has_audio>
<has_videordf:datatype=http://www.w3.org/2001/XMLSchema#string>C:\Users\Archana\Jan2019\TA2.mp4
</has_audio>
</TerroristAttack>
```

Figure 9 Representation of multilingual and multimodal encryptions of concept "Terrorist-Attack".

OWL2 provides owl: oneOf construct for storage of constraints list. Figure 8 (a) shows RDF/XML representation of constraints list of has_warning {yes, no}. Ontology does not directly support the concept of default value but we can store default value of properties by using data and object properties with the help of annotation. The part (b) of Figure 8 shows the storage of data property noOf_injured equal to 100.

For multilingual encryption of concept, we used 'label' in RDF/XML format. Figure 9 depicts the storage of Terrorist-Attack in Spanish (es) and German (de) languages. Multimodal encryption like audio, video, picture for a concept is stored by using data property. We have created has_audio, has_video, has_picture data properties for audio, video, picture respectively. Figure 9 shows RDF/XML representation of multimodal and multilingual encryption of concept "Terrorist-Attack".

4 Managerial Implications

Building a knowledge base is a never-ending process. Representing knowledge as a unit manages the balance between the precision value of decision and computational efficiency of deriving them. It also incorporates the balance between certainty vs. specificity. Each node/unit acts as a knowledge packet similar to long term memory of human and an instance base similar to the facts of the production system. Various applications of semantic technologies like large scale ontology mapping, matching, merging, partitioning, context-sensitive reasoning, temporal analysis, spatial analysis, question answering, and ontology learning are well performed under this representation. In this section, we briefly explain three applications where the unit of knowledge plays a significant role to overcome the existing problems.

4.1 Ontology Merging

Suppose one company acquires another company, then a knowledge base or ontology of these companies should be merged. Ontology matching (OM) is a process which provides a coherent knowledge base after merging of two or more ontologies. At the time of OM, we check only defining properties of the concept because if these properties are true then only decision will be taken. Figure 10 shows the merging of O_1 and O_2 ontologies where O'_{12} and O_{12} are intermediate and final ontology respectively. a_1, a_2... denote the defining properties of their respective concepts. Both ontologies are not preserved in their original form in final or resultant ontology but the hierarchy of both ontologies is maintained. We can see that both ontologies are merged in the correct position and the new tree is consistent. Partition of properties into two parts reduce search and conquer complexity of ontology matching as compared to traditional ontology matching where each and every aspect of one ontology is mapped with respect to each and every aspect of another ontology.

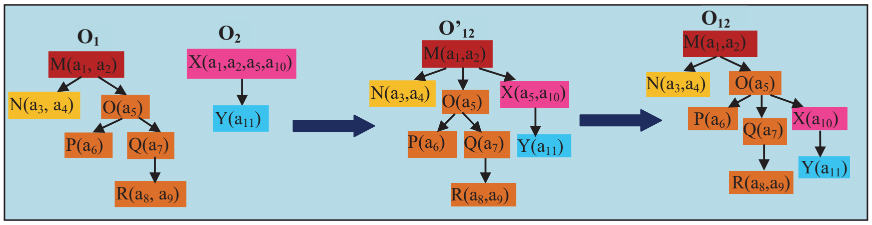


Figure 10 The process of ontology merging.

4.2 Ontology Partitioning

Nowadays we deal with large scale ontologies. It is required to partition the large search space of ontology into smaller parts so that we can perform various tasks like mapping and matching in parallel, with different processors. Ontology partitioning is a very tricky task because isolation of one node can generate inconsistency in clusters. The selection of partitioning algorithms depends on two factors namely number of clusters and type of information (incomplete, noisy and complete). During the development of ontology, user do not put knowledge as a unit therefore, most partitioning algorithms produce incomplete partitions or clusters and slow down the partitioning process hence eroding the advantages of partitioning. Our suggested storage of the unit of knowledge always produce consistent and complete information inside the clusters. Each node in the cluster itself contains a full description with exceptions and precision value. Clustering algorithms only use the properties of node for partitioning of ontology. However during partitioning, unless operator plays a very important role because more censors in one cluster can increase the computational complexity of the clusters matching.

We always put entities into the clusters in such a way so that each cluster has an average number of censors. Figure 11 shows partitioning of ontology into four clusters and every cluster has an average number of censors (shown by a colorful bubble). Now a selection of clustering algorithm depends only on the selection of a number of clusters rather than the type of information.

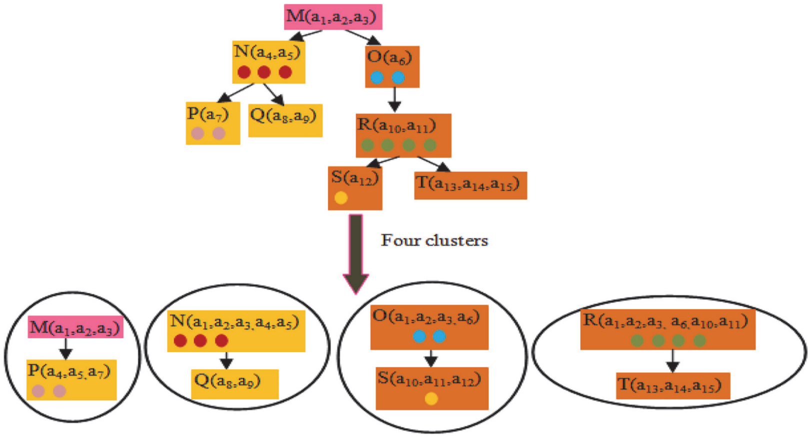


Figure 11 Partitioning of ontology.

4.3 Context Sensitive Reasoning

Context sensitive reasoning can be effectively accomplished by representing knowledge as a unit. This reasoning is helpful for answering queries of the type considering the domain of emergencies and I an instance: “What is I_1”, “Is I_1 emergency”, “Is I_1 disaster”, “Is I_1 man-made disaster”, “Is I_1 terrorist attack”. If an exception is true at any level then it would block the entire decision of that level. Suppose a user has a query “What is Bombay_26/11/2008”. Then according to context and type of users, the answer of this query will be varied. Suppose we have five types of users namely ‘very low priority user’, ‘low priority user’, ‘middle priority user’, ‘high priority user’, and ‘very high priority user’. We have classified users according to the availability of resources like time and data. Very low priority user has high resource constraint whereas very high priority user has no constraints on resources. If the user has high resource constraints then only top-level node will be considered and if the user has no resource constraints, then answer will be the last level of the taxonomy.

- Very low priority user: only the first level will be examined
  Bombay_26/11/2008 is an Emergency
- Low priority user: two levels (Emergency and Disaster) of the taxonomy will be examined
  Bombay_26/11/2008 is a disaster
- Middle priority user: three levels (Emergency, Disaster, and Man-Made) of the taxonomy will be examined
  Bombay_26/11/2008 is a man-made disaster
- High priority user: four levels (Emergency, Disaster, Man-Made and Deliberate) of the taxonomy will be examined
  Bombay_26/11/2008 is a deliberated man-made disaster
- Very high priority user: all levels (Emergency, Disaster, Man-Made, Deliberate and Terrorist Attack) of the taxonomy will be examined
  Bombay_26/11/2008 is a man-made Terrorist Attack which is a deliberate man-made disaster

5 Conclusion

The knowledge base of intelligent web should be as general as possible because the nature of information is dynamic. It should be open for expansion horizontally as well as vertically. Representing knowledge as a unit is an attempt to reduce the problem of classical ontology with the goal of enabling

AI applications to perform human-like behaviour (memorization, thinking, and imagination). All components of knowledge are stored in a hierarchical structure and this knowledge unit is able to acquire new knowledge without affecting already stored knowledge. It has the capability to continuously grow with the new added concept. It has been shown that how various operators of knowledge can be encoded in the RDF/XML format of ontology. Certain significant implications have been depicted taking domain ontology of emergency situations as the case study. Representing knowledge as a unit enriches representation scheme and provides worldwide acceptance of the web that is central to all the applications. As a major benefit, various problems of the semantic web like incomplete and uncertain information, large scale data integration, redundancy, ambiguity and vague behaviour of word, knowledge acquisition and mapping can be easily and efficiently tackled. In the future, we aim to provide different ways to store a unit of knowledge in the ontology by using existing constructs. To the best of our knowledge, it is first attempt to put all required attributes of an entity under one umbrella.

References

[1] Jain, S., and Jain, N. K. (2008). A generalized knowledge representation system for context sensitive reasoning: Generalized HCPRs System. *Artificial Intelligence Review*, *30*(1–4), 39.

[2] Shandilya S.K., Opinion Extraction and Classification of Reviews from Web Documents (2009). IEEE International Advance Computing Conference (IACC 2009).

[3] A. Patel, and S. Jain, "Present and future of semantic web technologies: a research statement", Int. J. Comput. Appl., 2019.

[4] Patel, A., and Jain, S. (2018). Formalisms of Representing Knowledge. *Procedia Computer Science*, *125*, 542–549.

[5] Michalski, R. S., and Winston, P. H. (1986). Variable precision logic. *Artificial intelligence*, *29*(2), 121–146.

[6] Bharadwaj, K. K., and Jain, N. K. (1992). Hierarchical censored production rules (HCPRs) system. *Data and knowledge engineering*, *8*(1), 19–34.

[7] Jain, N. K., Bharadwaj, K. K., and Marranghello, N. (1999). Extended hierarchical censored production rules (EHCPRs) system: An approach toward generalized knowledge representation. *Journal of Intelligent Systems*, *9*(3-4), 259–295.

[8] Jain, N. K., and Jain, S. (2013). Live multilingual thinking machine. *Journal of Experimental & Theoretical Artificial Intelligence*, *25*(4), 575–587.

[9] Jain, S., and Jain, N. K. (2014, March). A Globalized Intelligent System. In *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on* (pp. 425-431). IEEE.

[10] Horrocks, I. (2005, October). Owl: A description logic based ontology language. In *International Conference on Principles and Practice of Constraint Programming* (pp. 5–8). Springer, Berlin, Heidelberg.

[11] Bobillo, F., and Straccia, U. (2011). Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, *52*(7), 1073–1094.

[12] A. Patel, and S. Jain, "An intelligent resource manager over terrorism knowledge base", *Recent Patents on Computer Science*, 2019.

[13] Patel, A. and Jain, S. (2019). "A Partition Based Approach for large Scale Ontology Matching", Recent Patents on Engineering,

[14] Jain, S., and Jain, N. K. (2010). Acquiring knowledge in extended hierarchical censored production rules (EHCPRS) system. International Journal of Artificial Life Research (IJALR), 1(4), 10–28.

[15] Sowa, J. F. (2006). Semantic networks. *Encyclopedia of Cognitive Science*.

[16] Marchetti, A., Ronzano, F., Tesconi, M., and Minutoli, M. (2008). Formalizing Knowledge by Ontologies: OWL and KIF. *Relatórioapresenta-doL'Istituto di Informatica e Telematica (IIT).ConsiglioNazionaledelle Ricerche (CNR). Italia*.

[17] Ian Horrocks, Peter F. Patel-Schneider, Knowledge Representation and Reasoning on the Semantic Web : OWL, Published 2010.

[18] World Wide Web Consortium. (2012). OWL 2 web ontology language document overview.

[19] Taye, M. M. (2011). Web-Based Ontology Languages and its Based Description Logics. *The Research Bulletin of Jordan ACM*, *2*, 1–9.

[20] D.L. McGuinness, R. Fikes, L.A. Stein, and J.A. Hendler, "DAML-ONT: An Ontology Language for the Semantic Web", In Proceedings of Spinning the Semantic Web, 2003, pp. 65–93.

[21] McGuinness, D. L., Fikes, R., Hendler, J., and Stein, L. A. (2002). DAML+ OIL: an ontology language for the Semantic Web. IEEE Intelligent Systems, 17(5), 72–80.

[22] Jain, V., and Singh, M. (2013). Ontology based information retrieval in semantic web: A survey. *International Journal of Information Technology and Computer Science (IJITCS)*, *5*(10), 62.

[23] Stoilos, G., Simou, N., Stamou, G., and Kollias, S. (2006). Uncertainty and the semantic web. *IEEE Intelligent Systems*, *21*(5), 84–87.

[24] Bobillo, F., and Straccia, U. (2011). Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, *52*(7), 1073–1094.

[25] Milea, V., Frasincar, F., and Kaymak, U. (2012). tOWL: a temporal web ontology language. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(1), 268–281.

[26] Thomas, E., Pan, J. Z., and Ren, Y. (2010, May). TrOWL: Tractable OWL 2 reasoning infrastructure. In *Extended Semantic Web Conference* (pp. 431–435). Springer, Berlin, Heidelberg.

[27] Lee, C. S., Wang, M. H., and Hagras, H. (2010). A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. *IEEE Transactions on Fuzzy Systems*, *18*(2), 374–395.

[28] Ausín, D., Castanedo, F., and López-de-Ipina, D. (2012, December). TURAMBAR: An approach to deal with uncertainty in semantic environments. In *International Workshop on Ambient Assisted Living* (pp. 329–337). Springer, Berlin, Heidelberg.

[29] Bobillo, F., andStraccia, U. (2011). Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, *52*(7), 1073–1094.

[30] Mallik, A., Ghosh, H., Chaudhury, S., and Harit, G. (2013). MOWL: An ontology representation language for web-based multimedia applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *10*(1), 8.

[31] Ding, Z., and Peng, Y. (2004, January). A probabilistic extension to ontology language OWL. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii international conference on* System Sciences.

[32] Stoilos, G., Stamou, G., and Pan, J. Z. (2010). Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics. International Journal of Approximate Reasoning, 51(6), 656–679.

[33] Lera, I., Juiz, C., and Puigjaner, R. (2010, February). Owl-m extension for semantic representations of ontology alignments. In *2010 International Conference on Complex, Intelligent and Software Intensive Systems* (pp. 956–961). IEEE.

[34] Hewahi, N. M. (2018). A Security Technique for Censor Production Rules-based Systems. *KnE Engineering*, *3*(7), 1–10.

[35] Hewahi, N. M. (2018). Concept Based Censor Production Rules. *International Journal of Decision Support System Technology (IJDSST)*, *10*(1), 59–67.

[36] Jain, S., and Jain, N. K. (2012). Learning techniques in extended hierarchical censored production rules (EHCPRs) system. *Artificial Intelligence Review*, *38*(2), 97–117.

[37] Jain, S., Jain, N. K., and Goel, C. K. (2009). Reasoning in EHCPRs system. *Int. J. Open Problems Compt. Math*, *2*(2).

[38] Jain, S., and Jain, N. K. (2010). Representation of defaults and constraints in EHCPRs system: an implementation. *International Journal of Adaptive and Innovative Systems*, *1*(2), 105–120.

Biographies

Archana Patel is a Ph.D. student at National Institute of Technology, Kurukshetra, India since 2017. She received the Master of Computer Applications degree from National Institute of Technology Kurukshetra, India in 2016. She has two years experience as JRF in Defence Research and Development Organization (DRDO) funded research project. Her research interests include Semantic Web, Big Data, Ontological Engineering and Knowledge Warehouse.

Sarika Jain graduated from Jawaharlal Nehru University (India) in 2001. Her doctorate is in the field of Knowledge Representation in Artificial Intelligence which was awarded in 2011. She has served in the field of education for over 18 years and is currently in service at the National Institute of Technology, Kurukshetra. Her research interests are in the area of Intelligent Systems, Ontological Engineering, Semantic Web Technologies, and Linked Open Data Cloud with an aim to make people understand the importance of semantic web over the traditional web. Dr. Sarika is currently working toward solving the interoperability problem generated by IoT, Big Data and Cloud Computing initiatives. She has authored over 82 publications and five text books including "Information System" and "Mobile Computing". Dr. Sarika has just completed a research project sponsored by DRDO, India worth Rs 40 lakhs. She has constantly been supervising DAAD interns from different universities of Germany and many interns from India every summer. She is a member of IEEE and ACM and a Life Member of Computer Society of India.

Shishir K. Shandilya, Division Head of Cyber Security & Digital Forensics at VIT, is a renowned academician and active researcher with proven record of teaching and research. He is Cambridge University Certified Professional Teacher and Trainer, Senior Member of IEEE-USA and also elected as an executive member of IEEE Industry-Outreach Committee-India. Dr. Shandilya has received "IDA Teaching Excellence Award" for distinctive use of Technology in Teaching by Indian Didactics Association, Bangalore and "Young Scientist Award" for consecutive two years (2005 & 2006) by Indian Science Congress & MP Council of Science & Technology. He has written seven books of international-fame (published in USA, Denmark and India) and published quality research papers. He is an active member of over 20 international professional bodies. He is also an excellent programmer and credited various software projects in his account.