International Journal of Web-Based Learning and Teaching Technologies
Volume 16 • Issue 3 • May-June 2021

Sup_Ont:
An Upper Ontology

Sonika Malik, Maharaja Surajmal Institute of Technology, Delhi, India & National Institute of Technology, Kurukshetra, India
Sarika Jain, National Institute of Technology, Kurukshetra, India
https://orcid.org/0000-0002-7432-8506

ABSTRACT

A domain-independent conceptual model that aims to be highly reusable across specific domain applications is provided by upper-level ontologies which usually describe abstract concepts. In this paper, the authors proposed Sup_Ont, a fundamental upper ontology. In this ontology, the structure of the universe shows the concept of reality that is defined to have an existence which is known as truth. The devised super ontology and hence the domain ontologies can be reused across applications because of the generalized representation scheme used that is an EHCPR. An extended hierarchical censored production rules (EHCPRs) system is a knowledge representation system for reasoning with real-life problems and a step towards a generalized representation system. An EHCPR is a unit of knowledge resulting in a knowledge base that shows modularity and hierarchy. Extended hierarchical censored production rules (EHCPRs) have been used to represent the knowledge in intelligent systems.

KEYWORDS

EHCPR, Matter, Space, Sup_Ont, Time, Upper Ontology

INTRODUCTION

The technology around us is growing at a very fast pace as so as the web. We started with static web and in today's world, we are talking about the semantic web. The inventor of the WWW, Tim Berners Lee, came up with the idea of the semantic web. He provided the means for annotation of Web resources with machine process-able metadata providing them with background meaning and knowledge. The Semantic Web can be considered to be made of the Knowledge Treasure (Jain & Jain, 2014). The Knowledge Treasure incorporates both procedural and declarative knowledge. The Knowledge Base and the database together form the declarative Knowledge. A hierarchical network of multi-encrypted concepts forms the Knowledge Base whereas the database is the set of multi-encrypted instances. The procedural knowledge includes many features of visualization, navigation, knowledge discovery, Online database integration, user interface that is context-sensitive and multilingual in nature, querying, reasoning, decision support, and the system management tools within it. For the sake of decidability, an ontology language does not provide the expressiveness we

DOI: 10.4018/IJWLTT.20210501.oa6

want (e.g. constructor for composite properties), therefore it is essential to include rules as well in procedural knowledge (Jain & Jain, 2012).

Knowledge Treasure = Declarative Knowledge + Procedural Knowledge
Declarative Knowledge = Knowledge Base +Database
Procedural Knowledge = Procedures + Rules.

In Philosophical terms, studying various kinds of things that exist can be considered as ontology. Ontology is a representation of vocabulary which is often specialized to some domain. If we go deep, it is not the words or vocabulary that qualifies as ontology, but the concept that the vocabulary intends to capture. So any translation of terms in a particular ontology from one language to another, for example from German to French, will not change the conceptual meaning. In engineering design, ontology is discussed with the domain of electronic-devices that constitute terms that describe conceptual elements like transistors, operational amplifiers, and voltages and the relationship between these elements. Operational amplifiers are a type of electronic device, and transistors are a component-of operational amplifiers. In order to identify such vocabulary and the associated conceptualizations an analysis of the kinds of relations and objects that are present within the domain is generally required. The body of knowledge describing some domain is often referred to as ontology (Chandrasekaran et al., 1999).

Analysis of various ontologies clarifies knowledge structure. Ontology is the heart of any system of knowledge representation for any given domain. Without any ontology or conceptualizations, no ontology can represent the knowledge. Thus, performing an ontological analysis of the domain effectively, an effective knowledge representation system and vocabulary need to be defined.

Ontologies are grouped into three broad categories of upper, mid-level, and domain ontologies. The Upper ontologies are also known as a top-level ontology that defines the universal concepts which are the same through all knowledge domains.

BACKGROUND INFORMATION

This section briefly discusses the existing upper ontologies, ontology languages available and their comparisons. The comparisons are given in Table 1 and Table 2 respectively.

Upper Ontology

The definition of upper ontology, as given by (Semy & Pulvermacher, 2004), is domain-independent ontology and is of a high level, which gives a general description of the concepts. Domain-specific ontologies can be constructed with the upper ontologies. Common sense objects are characterized by the upper ontologies, i.e. ones which are basic for humans and their understanding of the world (Kiryakov et al., 2001). Therefore, an upper ontology is often restricted to Meta, generic, abstract and philosophical concepts (Standard Upper Ontology Working Group Website, 2001). Foundational ontologies or Universal ontologies can be used instead of the term Upper ontology.

The usage of upper ontologies can be described in two ways: bottom-up and top-down. In a top-down approach, upper ontology is used as the basis for deriving concepts in the domain ontology. Here, the designer of the domain ontology utilizes knowledge and experience of the upper ontology. However, in a bottom-up approach, the new or existing domain ontology is mapped to the upper ontology. This approach also capitalizes on the knowledge built into the upper ontology but one would expect the mapping to be more challenging, as inconsistencies may exist between the domain and upper ontology.

Domain ontologies are used to state-specific information about domains, or their situation. We can represent knowledge of propositional attitudes (such as hypothesize, believe, expect, hope, desire, and fear) once a basis for proposition representation is obtained (Chandrasekaran et al., 1999). Thus,

Table 1. Comparison of existing upper ontologies

Criteria/ Names	Year	Developers	Availability	Dimensions	Languages	Applications
CYC	Initial Release 1984 Stable Release: 2013	CYCORP	Open Source: Research Cyc and open cyc	about 300.000 concepts, 3.000.000 assertions (facts and rules), 15.000 relations (these numbers include microtheories)	CycL, OWL	Natural language Processing, Network Risk Management
GFO	1999	Onto-Med Research Group	Licensed	Classes:79 Subclass Relation :97 Properties: 67	FOL, KIF,OWL	Bio-Medical Domain
Sowa	1999	J.F.Sowa	Open Source	Classes:30 Subclass Relation :5 Axioms: 30	FOL, KIF	No Application as but used for making other upper ontologies
SUMO	2000	Teknowledge Cooperation	Open Source	Terms: 20,000 Axioms: 60,000	SUO-KIF, OWL	Representation, Reasoning
BFO	2002	B.Smith, P.Grenon, H.Stenzhom, A.Spear	Open Source	Entity Class (SNAP Classes: 18, SPAN Classes:17)	OWL	Bio-Medical Domain
DOLCE	2002	N.Guarino and other researchers from LAO	Open Source	Approximately 100 concepts are defined	FOL, KIF,OWL	Multilingual Information Retrieval, E-learning
PROTON	2004	Ontotext	Open Source	Concepts: 300 Properties: 100	OWL-Lite	Knowledge Management Systems for legal and tele-communication domain, Business Data Ontology for semantic web services
COSMO	Initial Release: 2006 Latest : 2016	COSMO working group	Open Source	Classes:8000 Relations :1000 Restrictions: 3000	OWL	Natural Language front end, Graphical Interfaces

Table 2. Comparison of existing ontology languages

Name/Criteria		Developers	Standard	Paradigm Used	Web Standard Used	Expressiveness	Inference Mechanism	Constraint checking
SHOE		Sean Luke, Lee Spector, James Hendler, Jeff Jeflin, David Rager at the university of Maryland, college Park	Yes	Tags	HTML	High	Yes	No
KIF		DARPA	Yes	First Order Predicate Logic	No	High	Yes	Weak
RDF/RDFS		W3C	Yes	Object Oriented Structure	XML	Medium	No	Weak
DAML+OIL		DARPA	No	DL	XML and RDF	High	Yes	Weak
OWL	Lite	W3C	Yes	DL	RDF	Weak	Yes	Good
	DL		Yes	DL	RDF	Medium	Yes	Good
	Full		Yes	Cannot be translated to DL	RDF	High	No	Good
OWL2		W3C	Yes		RDF	More than OWL	Yes	Good

beliefs, goals, hypotheses, and predictions about a domain can be represented by ontology. Things such as plans and activities can also be represented by the ontologies because the specification of world objects and relations is also required. Activities and properties of objects such as "intentional entities" belonging to special classes are described by propositional attitude objects—for instance, agents such as humans with mental states. General terms that are restricted to specific knowledge encompass the ontologies. For example, terms like space, time, parts, and subparts which are applicable to all domains; malfunction is applicable to biological or engineering domains; while hepatitis is applicable to medicine only. Even in cases where a task might seems very domain-specific, knowledge representation requires an ontology that describes knowledge at higher levels of generality. For instance, solving problems of turbines and its domain may require knowledge expressed using terms such as flows and causality which are domain-general. Descriptive and general terms are called as the upper ontology or top-level ontology.

Existing Upper Ontologies

The existing seven upper ontologies are being described, namely Cyc, GFO, Sowa's Ontology, SUMO, BFO, DOLCE, PROTON, and COSMO. Their comparison is given in Table 1. These ontologies are quite active inside the research community.

Open Cyc Ontology was developed by Douglas at Microelectronics and Computer Technology Corporation (MCC) in 1984 (Niles & Pease, 2001). Cyc technology was applied for further development and commercialization by a company called Cycorp formed later in 1994. The Cyc knowledge base is the division of thousands of micro theories based on a particular knowledge domain, a particular level of detail, and a particular time interval. The Cyc knowledge base contains a set of terms and assertions needed to be stored in a manner so that it can be easily used, so CycL is developed.

General Formal Ontology (GFO) ontology was developed by Heinrich Herre of OntoMed Research group in Leipzig. Processes, objects, functions, time, space, properties, and relations are part of GFO. It is implemented in KIF and also OWL-DL version exists. It is essentially used in Gene ontology for the tnowledge representation of biological functions (Herre et al., 2006).

Sowa's Ontology is not based on a fixed hierarchy because hierarchy is generated automatically rather it is based on the framework of distinctions. It is implemented in First Order Logic (FOL) and version of KIF is also existed (Basic Formal Ontology and Medical Ontology, 2003; Sowa Upper Ontology, 1999).

Suggested Upper Merged Ontology (SUMO) ontology was initially developed by Adam Pease and Ian Niles, and it was released in Dec 2000. The development of SUMO was done for recovery simplification and information search, data interoperability and automated interpretation (Sowa Upper Ontology, 1999). This ontology includes particulars and universals. Elements of realism and intellectual categories are included in SUMO (Farar & Bateman, 2004; Niles & Pease, 2001). SUMO is represented by Standard Upper Ontology Knowledge Interchange Format (SUO-KIF). Lower level ontologies, Mid-level Ontology (MILO) are also connected to SUMO. Domain ontologies for government, military, terrorism and bombings are also included.

Basic Formal ontology (BFO) is an upper level ontology that is devised for information recovery support, unifying and analyzing different domains. The BFO is divided into distinct entities SNAP and SPAN. It comprises of 36 classes which are split into one top connecting class, and 17 SPAN classes and 18 SNAP classes (Basic Formal Ontology and Medical Ontology, 2003; Mascardi et al., 2007). Currently, it is implemented in Web Ontology Language (OWL).

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) was started in 2002; it was a project Wonder web. It was ended in 2004. However, DOLCE is actively being used which was developed as part of project Wonder web (Herre et al., 2006; Standard Upper Ontology Working Group Website, 2001). DOLCE aims to establish an agreement to negotiate the meaning of validating collaboration. KIF, WL and FOL are the implementation languages for DOLCE. DOLCE is the ontology of particulars, and those particulars are instances. The particulars are entities that are organized into categories and don't have instances; universals are those entities that organize the categories of particulars and can also have instances.

PROTo ONtology (PROTON) is a basic upper ontology, which gives upper-level ideas for semantic notation, indexing and recovery. OWL-Lite is used to implement PROTON. It has been used for semantic web services as a basis of Business data ontology (Basic Formal Ontology and Medical Ontology, 2003).

COmmon Semantic MOdel (COSMO) is an upper ontology developed by Ontology and Taxonomy Coordinating Working Group (ONTACWG) and its child group COSMO working group (COSMO-WG). COSMO is considered as a web or group of ontologies that considers and combines all the basic logically specified concepts like relations, classes, instances and functions. All these basic concepts can be further used in other domain ontologies (Micra, n.d.). The use of these basic concepts help in providing precise interoperability of knowledge-based systems using logical relations of their ontologies for reasoning purposes. Presently, COSMO assimilates concepts from OpenCyc and SUMO ontologies and also classes from DOLCE and BFO.

Ontology Languages

Ontology can be encoded using ontology language which is a formal language for this purpose. Ontology languages are almost always generalizations of frame languages and are declarative languages. They are commonly based on either first-order logic or on description logic (Maniraj & Sivakumar, 2010). Ontology languages can be classified either on the basis of syntax and structure.

Classification based on syntax:

Traditional syntax ontology languages
Mark-up ontology languages
Classification based on structure:
Frame-based languages
Description logic-based languages
First-order logic-based languages

Common logic, CycL, DOGMA, F-Logic, KIF, KL-ONE, KM programming language, LOOM, OCML, OKBC, and RACER are some of the traditional syntax ontology languages. Languages such as SHOE, RDF, RDFS, DAML+OIL, OIL, OWL, OWL2 etc. are Mark-Up ontology languages to encode knowledge, most commonly with XML using mark-up schemes.

The languages which are frame-based are completely or partially frame-based languages like F-Logic, OKBC, and KM. An extension of frame languages is provided by the description languages without taking a jump to first-order logic and support for arbitrary predicates. Some of them are KL-ONE , RACER, and OWL. Common predicates such as Common Logic, CycL and KIF are allowed by many of the ontology languages which support expressions in first-order logic. Some of the semantic web languages are compared in Table 2 based on several parameters like standard, the paradigm used, developers, web standard used, expressiveness, inference mechanism, and constraint checking.

GAPS OF THE EXISTING UPPER ONTOLOGIES

All the upper ontologies mentioned in 2.1 except SOWA are implemented in OWL. Although OWL is a very expressive language but it does not support some features. In OWL there is no distinction between defining features and characteristic features. In any ontology each entity has these two features. The defining features of a concept are basically necessary and sufficient conditions which cannot be changed in any case. However, the instance of a particular concept isn't allowed to be held for an individual or an item by characteristic features. Along with non-monotonic inheritance, OWL on a very basic level is not capable of representing any kind of non-monotonic reasoning. It can also represent the complex kinds of monotonic inheritance (Gupta & Bharadwaj, 2017).

OWL ontologies do not distinguish between knowledgebase and database. Exceptions are not supported by OWL. No consistent support for uncertainty representation or plausible reasoning is provided by Semantic web languages like OWL. To support all these features an EHCPR framework is defined by Jain and Jain (Jain & Jain, 2013; Jain & Jain, 2010a; Jain & Jain, 2010b). A knowledge item for solving real world problems that combine rules, exceptions, hierarchal characteristic features and instances is what is referred to as EHCPR or so-called an intelligent representation.

Defining and characteristic features are uniquely represented by EHCPR. Variable accuracy in reasoning is exhibited in EHCPR representation to show the variance of certainty in belief in a conclusion and its specificity in the reasoning process. Easy management of complexity of information, leveled views of detail, and facilitation of inference engine attention on different aspects is achieved by EHCPR by the portioning of knowledge at different levels.

There is a structure of EHCPR given by Jain & Jain which consists of different operators. The operator A defines concept or decision, which is head of the rule, operator B is the preconditions that is defined by If, C is a censor or exception part of "If-Then-else rule", which is defined by Unless, G operator gives a general concept, and S denotes the Specificity. Three more operators in EHCPR are Has_part, Has_property, Has_instance which define the structural parts, characteristic properties and instances respectively. γ, δ defines the strength of implication (Casellas et al., 2005; Jain et al., 2015; Jain & Mishra, 2014).

DOLCE is an "ontology of particulars"; it does have universals (classes and properties), but the claim is that they are only employed in the service of describing particulars. DOLCE does not include such items as a hierarchy of process types, physical objects, organisms, units and measures, and event roles. Cyc is by far the oldest ontology project; only part of it (called Open Cyc, http://opencyc.org/) is released under a free license. One of the drawbacks of SUMO is its relatively low coverage that does not allow its employment for open-domain applications. It also lacks a connection between its concepts and natural language words. These limitations have been partially overcome by connecting SUMO to the Wordnet lexicon. The difference between COSMO and Super Ontology (SUP_ONT) is in Table 3.

Table 3. Comparison of COSMO and SUP_ONT

S. No	COSMO	SUP_ONT
1.	Do not distinguish between Knowledge Base and Data Base	Differentiate between Knowledge Base and Data Base, as instances are in Data base and Classes, Properties and Relations are in knowledge base.
2.	No distinction between defining features and characteristic features. The defining features of a concept are basically necessary and sufficient conditions	In Super Ontology there is distinction between defining features and characteristic features.
3.	In COSMO the authors didn't concentrate on Abstract objects.	Abstract Objects are included in Super Ontology.
4.	Variable Precision Logic is not included in COSMO i.e. Two aspects of precision are the *specificity* of conclusions and the *certainty* of belief	Specificity and Certainty factor is included in Super Ontology
5.	COSMO is not the common FO, but is being used to demonstrate that common foundation ontology is technically feasible, if funding is available.	Super Ontology can be act as common foundation ontology.

SUPER ONTOLOGY

The super ontology as is the theme of this paper describes the structure of the universe and defines the concept of reality. A reality is defined to have an existence which is known as truth (Malik et al., 2015). All objects existing in this world come under the purview of this ontology. All entities belonging to this universe are permanent, but undergo countless changes continuously. No destruction takes place during these changes. Another form is obtained after recycling (Jainlbrary, n.d.; UMich, n.d.).

An entity suffers the changes into different shapes and forms that can be either artificial or natural. For example: different changes like childhood, youth and old are undergone by a human as he/she goes through the process of growth. Human beings undergo these changes, which are natural modifications.

The structure of the universe is illustrated in Figure 1. In this ontology, we have defined that everything in this world is an entity that can be either a concrete entity or an abstract entity. A Concrete entity is an entity which can be touched that is tangible and abstract entity is an entity which cannot be touched that is intangible. We have defined the different levels for this upper ontology. At the first level we have defined the concrete and abstract entities.

At the second level concrete entity is divided into two modules named as Living and Non-living entities. A living entity (Jiva) is an entity whose defining feature is consciousness which means the entity either has various senses or has the capacity to know, learn, understand feel pain and pleasure. For example: A human being who has all the five senses like touch, taste, smell, sight and hearing. A non-living entity (Ajiva) is an entity which does not possess life and does not have the capacity to know, understand, feel pain, and pleasure or does not have various senses.

Living entities can be divided into Mobile and Non-mobile entities. Mobile entities are those entities that can be movable and Non-mobile entities which are not movable. Having only one sense, the sense of touch, Non-mobile entities are called Jivas which can be air-bodied, fire-bodied, water-bodied, earth-bodied and plant-bodied. Earth-bodied substances are lifeless types of earth like dirt, metal, sand and coral. Water-bodied elements are apparently lifeless types of various sorts of water like dew, mist, chunk of ice and rain and etcetera. Fire-bodied substances are apparently lifeless types of various sorts of flame like blaze, lightening, woods fire and hot fiery remains and so forth. Air-bodied substances are lifeless types of air like wind, whirlwinds and cyclones (UMich, n.d.).

In a similar manner, mobile entities can be 2-sensed, 3-sensed, 4-sensed and 5-sensed. Having the sense of touches and taste organisms such as shells, worms, insects, microbes in stale food and termites are two-sensed. Along with having the sense of touch and taste organisms such as bugs, lice, and white ants are three sensed also having the sense of smell. Organisms which possess the sense of touch, taste, smell and sight such as scorpions, crickets, spiders, beetles, locusts and flies are four-sensed organisms. Humans and animals are examples of the organisms which have all the five senses of touch, taste, smell, sight and hearing and are regarded as five sensed beings (Gupta & Bharadwaj, 2017; Malik et al., 2015).

Non-living entities can be classified as Natural and Man-made entities. Natural entities are those entities in which there is no involvement of human beings and manmade entities are invented by human beings. Natural nonliving things are like sun, moon, rivers, mountains, clouds, wind and stones etc. and manmade nonliving entities are like computers, machines, chairs, balls etc.

Now we will come to abstract entities which can be divided as Based abstract entity and derived abstract entity.

The Based abstract entities are those entities which are independent and classified into mind, matter, space, time and spirits. The derived abstract entities are those entities which are dependent on some other entities and are classified as mathematical objects, facts, attributes and propositions.

Matter is a stuff out of which things are made. It is the underlying structure of changes or we can say that particularly changes of growth and of delay. It is a potential which has implicitly capacity to develop its reality. It is a kind of stuff without specific qualities and so is in-determine and contingent. Matter is divided into two parts. One is perceptible matter and other is intelligible matter (Jainlbrary, n.d.).

The limit of an encompassed body is characterized as space while that of measure of movement is characterized as time. On the off chance that there is no adjustment in the universe there would be no time. So, there is no plausibility of time in the absence of a counting mechanism.

The mind is an entity that has some decision making capability and can be classified as unconscious mind, conscious mind and su-conscious mind. Spirits are the nonphysical part of a person and can be classified as ghosts, vampires and demons. Time is divided based on its position and duration. Time duration is a time interval which can be weeks, months, hours, and years. Time position defined the time status and can be classified as full time or part time.

. A week can be divided into seven days, year can be divided into twelve months and hour can be divided into minutes. In the next level minutes can be classified into seconds and seconds can be milliseconds, picoseconds, microseconds and so on. Mathematical objects can be divided as numbers, figures, symbols and text. Numbers can be complex number, real number and imaginary number.

Attributes is an entity that defines the property of an object and can be divided into two parts. One is internal attributes and other is relational attributes. The Internal again can be classified as shape attribute like rigid, fillable and pliable etc., physical state like solid, liquid and gas and sat saturation attribute like wet and dry. Similarly Relational attribute can be divided as social role, truth value like true or false and positional attribute. The positional attribute can be classified as vertical, horizontal, left and right.

All the entities of super ontology have their defining and characteristic features. Each entity is an EHCPR. The defining features distinguish one concept from another while characteristic features provide further knowledge. We can define all the operators of EHCPR for all the entities.

At the first level there is an entity. The defining feature of an entity is Part_of _universe i.e. if anything exists in this universe that is an entity. The Generality (G) for the entity is Nil. The specificity(S) for the entity is concrete entity and abstract entity. The Has_part, Has_property and Has_instances are also Nil.

Entity
If Part_of_universe
Unless Nil
G Nil
S Concrete entity, Abstract Entity
Has_part Nil
Has_property Have some name or value
Has_instance Nil

At the second level the defining feature for concrete entity is is_tangible which can be touched. At the same level the defining feature of abstract entity is is_intangible which cannot be touched.

Concrete Entity
If Tangible
Unless Nil
G Entity
S Living, Nonliving
Has_part Nil
Has_property Has some weight
Has_instance Nil
Abstract Entity
If In-tangible

Unless Nil
G Entity
S Based Abstract Entity, Derived Abstract Entity
Has_part Nil
Has_property Weightless
Has_instance Nil

At the third level living, nonliving, based abstract entity and derived abstract entity.

Living entity
If Consciousness
Unless Nil
G Concrete Entity
S mobile, non-mobile
Has_part body, senses, speech, mind, respiration
Has_property growth, reproduction
Has_instances Nil
Non-living entity
If do not possess life
Unless Nil
G Concrete Entity
S manmade, natural
Has_part Nil
Has_property shape, size
Has_instance Nil
Based Abstract Entity
If Independent Entity definition
Unless Nil
G Abstract Entity
S Space, time, mind, matter, spirits
Has_part Nil
Has_property Nil
Has_instance Nil
Derived Abstract Entity
If Dependent entity definition
Unless Nil
G Abstract Entity
S Mathematical objects, Attributes, Facts, Propositions
Has_part Nil
Has_property Nil
Has_instance Nil

At fourth level there are some other entities like mobile, non-mobile, man-made, natural, space, time, mind, matter, spirits, propositions, facts, mathematical objects and attributes.

Mobile
If Movable
Unless Broken Leg/wing
G Living Entity

S 2-sensed, 3-sensed, 4-sensed, 5-sensed
Has_part eyes:2{0,1,2},legs:2{0,1,2,3,4},ears:2{0,1,2}
Has_property requires energy
Has_instance Nil
Non-mobile
If Not movable
Unless Nil
G Living Entity
S 1-sensed
Has_part Nil
Has_property produces energy
Has_instance Nil

Manmade
If invented by human being
Unless Nil
G Non-living
S computers, machines, chairs
Has_part Nil
Has_property usefulness: yes {yes, no}
Has_instance Nil

Natural
If No human involvement
Unless Nil
G Non-living
S sun, moon, wind, storms, mountains
Has_part Nil
Has_property Abundant: yes {{yes, no}, Renewable: yes {yes, no}
Has_instance Nil

Space
If limit of surrounding
Unless Nil
G based abstract entity
S Nil
Has_part Nil
Has_property Geometry, Topology and Dimensionality
Has_instance Nil

Time
If duration/period
Unless Nil
G based abstract entity
S time duration, time position
Has_part Nil
Has_property stoppable: no {no}, changeable: yes {yes}
Has_instance Nil

Mind
If decision-making
Unless undeceive
G based abstract entity
S unconscious, conscious, sub-conscious
Has_part cognitive, affective, conative
Has_property thinking: yes {yes, no}, feeling: yes {yes, no},
doing: yes {yes, no}
Has_instance Nil
Matter
If particular kind of stuff
Unless no stuff
G based abstract entity
S perceptible matter, intelligible matter
Has_part Nil
Has_property color, density, volume, mass
Has_instance Nil

Spirits
If non-physical
Unless Nil
G based abstract entity
S holy spirit, evil spirit
Has_part Nil
Has_property strength: yes {yes}
Has_instance Nil

Mathematical objects
If inert and unchanging
Unless Nil
G Derived abstract entity
S figures, numbers, symbols, text
Has_part integer, float
Has_property even, odd
Has_instance Nil

Attributes
If defines property of an object
Unless Nil
G derived abstract entity
S relational attribute, internal attribute
Has_part name, value
Has_property attribute identifiers
Has_instance Nil

Propositions
If assertion that express an opinion
Unless Nil
G derived abstract entity
S Nil

Has_part Nil
Has_property truthiness
Has_instance Nil

Facts
If really exists
Unless Nil
G derived abstract entity
S Nil
Has_part Nil
Has_property unchanged
Has_instance Nil

At fifth level of the tree structure, the entities are 1-sensed, 2-sensed, 3-sensed, 4-sensed, 5-sensed, relational attributes, internal attributes, figures, symbols, text and numbers.

1-sensed
If sense of touch
Unless Nil
G non-mobile
S earth bodied, water bodied, fire bodied, air bodied, plant bodied
Has_part cell-wall
Has_property feel, autotrophic
Has_instance Nil
2-sensed
If sense of touch and taste
Unless Nil
G non-mobile
S insects, worms, shells
Has_part body, mouth
Has_property Body symmetry: radial {radial, bilateral, unilateral} habitat: marine {marine, terrestrial, aquatic}
Has_instance Nil
3-sensed
If sense of touch, taste and smell
Unless nil
G non-mobile
S white-ants, bugs, lice
Has_part body, mouth, nose
Has_property body type: soft-bodied {soft-bodied, hard bodied}, Respiration: through skin {through skin and lungs}
Has_instance Nil
4-sensed
If sense of touch, taste, smell and sight
Unless Nil
G non-mobile
S spiders, scorpions
Has_part body, mouth, nose, eyes
Has_property Skeleton system: Exoskeleton {Exoskeleton, Endoskeleton}
Has_instance Nil

5-sensed
If sense of touch, taste, smell, sight and hearing
Unless Nil
G non-mobile
S humans, animals
Has_part skin, tongue, nose, eyes, ears
Has_property Body symmetry: bilateral {radial, bilateral, unilateral}
Has_instance Nil
Figures
If make an appearance
Unless Nil
G mathematical objects
S Nil
Has_part side
Has_property dimensions, angle
Has_instance circle, triangle, rectangle, polygon
Symbols
If written or drawn representation
Unless Nil
G mathematical objects
S Nil
Has_part Nil
Has_property sign, shape, color
Has_instance α, β, μ
Text
If written or printed work
Unless Nil
G mathematical objects
S Nil
Has_part cut, copy, paste
Has_property font style, visibility, size
Has_instance a...z, A...Z
Numbers
If an arithmetic value
Unless Nil
G mathematical objects
S real number, complex number, imaginary number
Has_part fractional, numerator, denominator
Has_property commutative, associative, distributive
Has_instance 1…¥

We have defined EHCPR's of five levels. Similarly, we can define all other levels. In this paper, we have designed a super ontology and defined EHCPR of each and every entity and it is implemented in Protégé.

Example

We can take the number of examples for this particular scenario. In this paper, we have defined super ontology as upper ontology. This super ontology can be further used to define many domain-specific ontologies. For Example, Biodiversity ontology published by NCBO is domain ontology for plants and animals. These plants and animal entities can be further categorized into other domain ontologies.

Figure 1. Super ontology

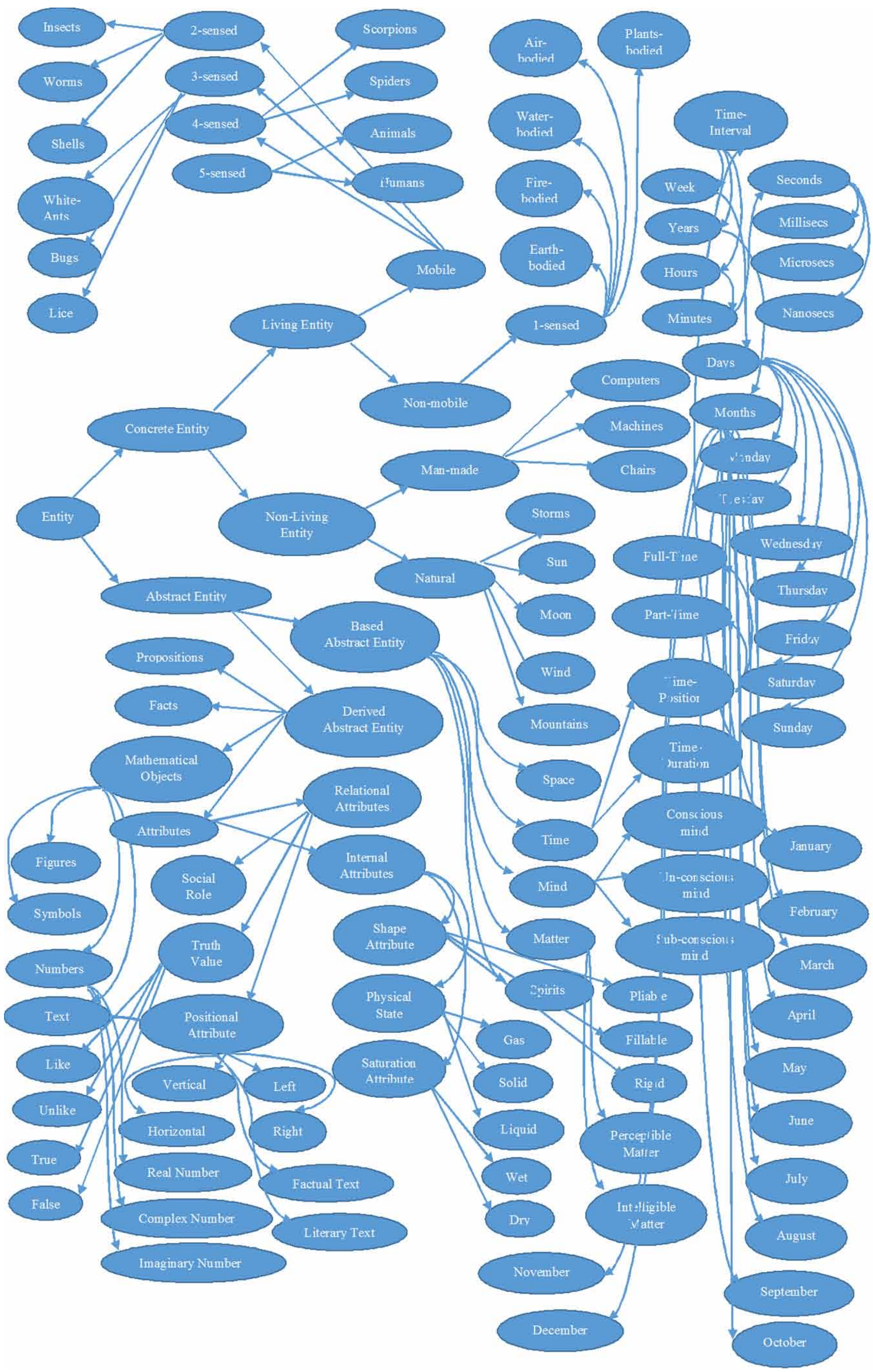
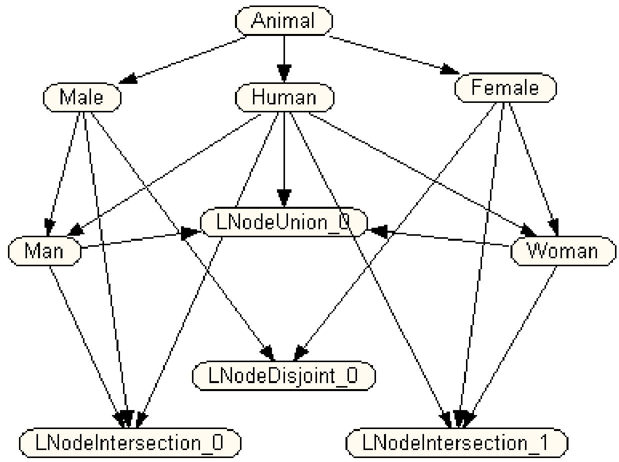


Figure 2.



In the super ontology as in Figure 1, we have started from an entity and reached to the animals and humans. We have defined nature ontology as domain ontology as in Figure 2 (Zhang et al., 2009). In this figure, ontology is started from animal and we can explore from that entity to the next level. The entity Animal can be divided into Male, female. Human can be a part of animal that depends on the perception of the individual. Similarly, we can divide entities into next levels like man and woman. We can also define different operations into it like Union, intersection and disjoint. Similarly, we can define any of the domain ontology from Super ontology.

IMPLEMENTATION IN PROTÉGÉ

Portege is a knowledge base and an ontology editor developed by Stanford University. The construction of domain ontologies and customized data entry forms are enabled by this tool. Portege defines the classes with their hierarchies, the relationship between classes, properties, variables and variable value restrictions. The visualization packages such as OntoViz come with Protégé which helps the user to visualize the ontologies. The key feature of Protégé is that at the same time it supports domain specialists, knowledge engineers and tools builders. This is the main difference with existing tools, which are typically targeted at the knowledge engineer and lack flexibility for meta-modeling. This latter feature makes it easier to adapt Portege to new requirements and/or changes in the model structure (Kapoor & Sharma, 2013).

Classes, Data Properties, Object Properties, individuals and Onto-graph can be represented in Protégé. All the classes of Super Ontology are defined in Figure 3(a) and Figure 3(b). Similarly, Data properties and Object Properties are defined in Figure 4 and Figure 5. Onto-graph for Concrete Entity and Abstract Entity classification are in Figure 6 and Figure 7.

CONCLUSION

In this paper, we have defined a super ontology that explains the structure of the universe and concepts of reality. This super ontology provides a shared knowledge space with consistent and well-defined vocabulary. It allows the reuse of Knowledge. The super ontology will contain all symbolic

Figure 3a. Classes

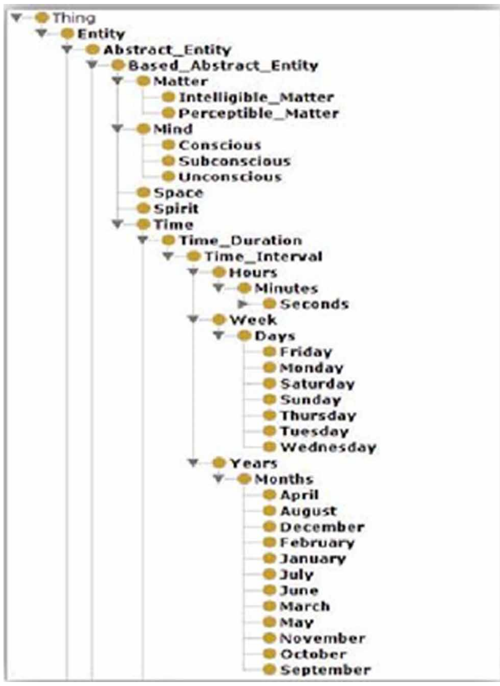


Figure 3b. Classes

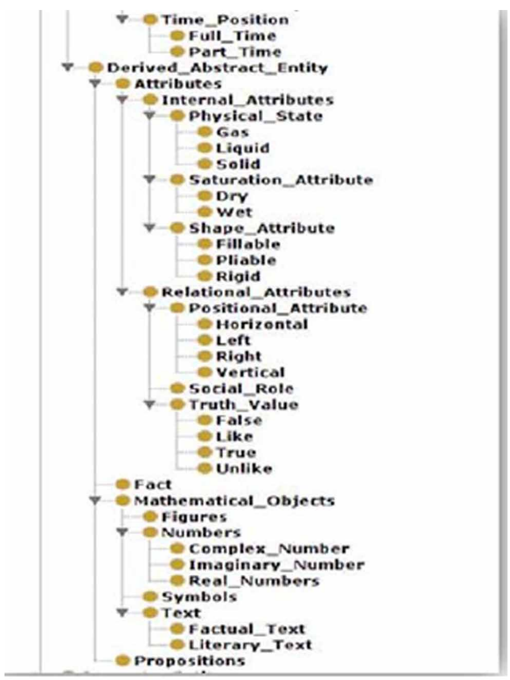


Figure 4. Object Properties

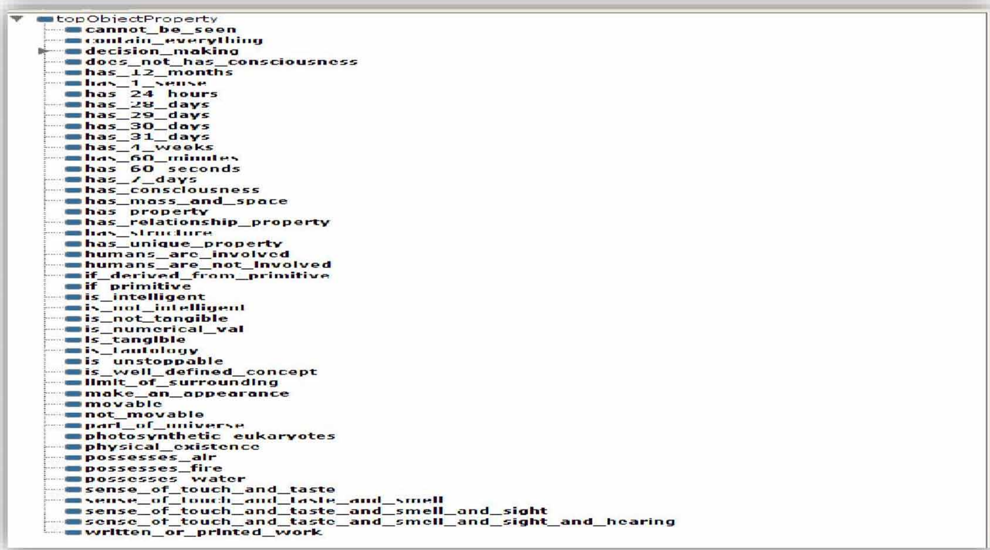


Figure 5. Data Properties

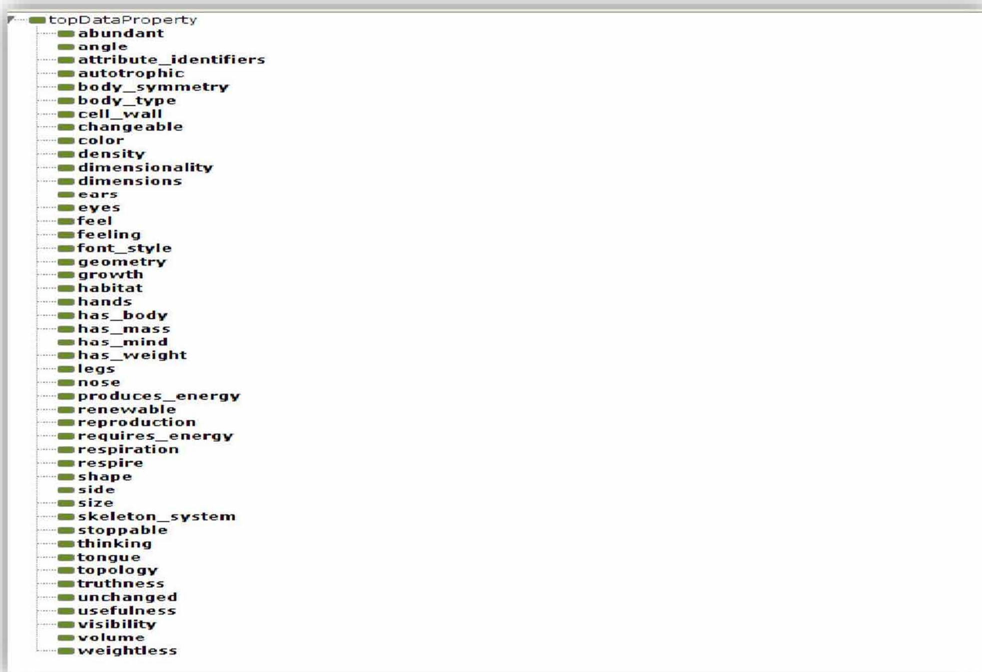


Figure 6. Onto graph for concrete entity classification

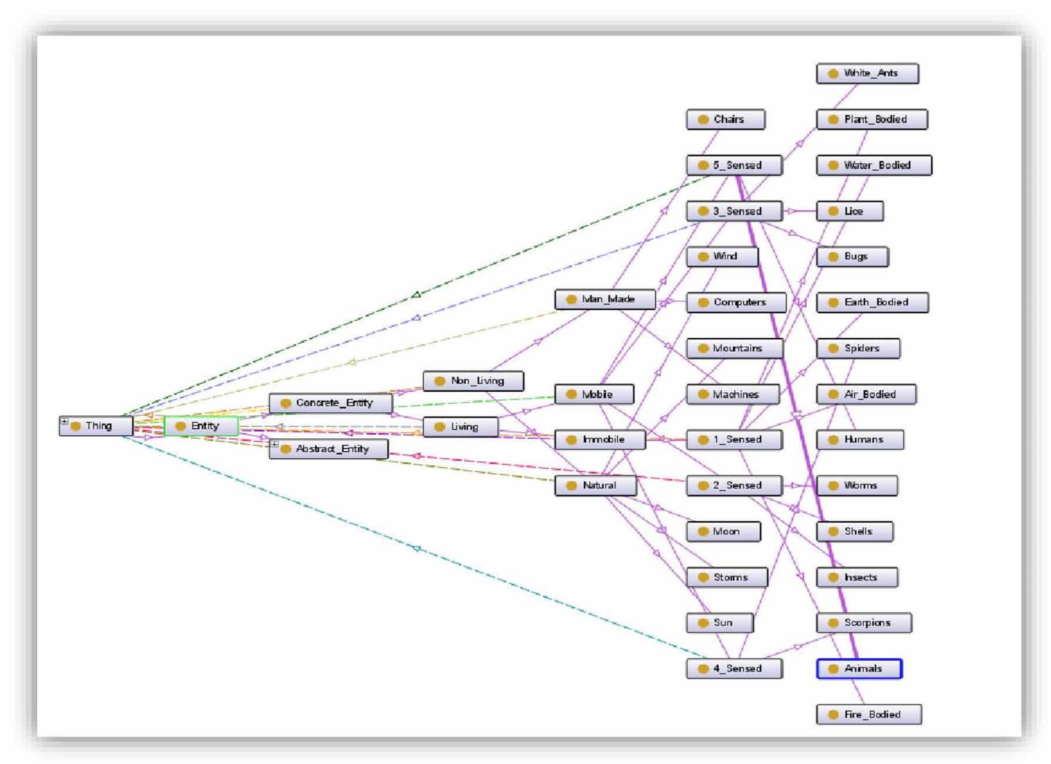
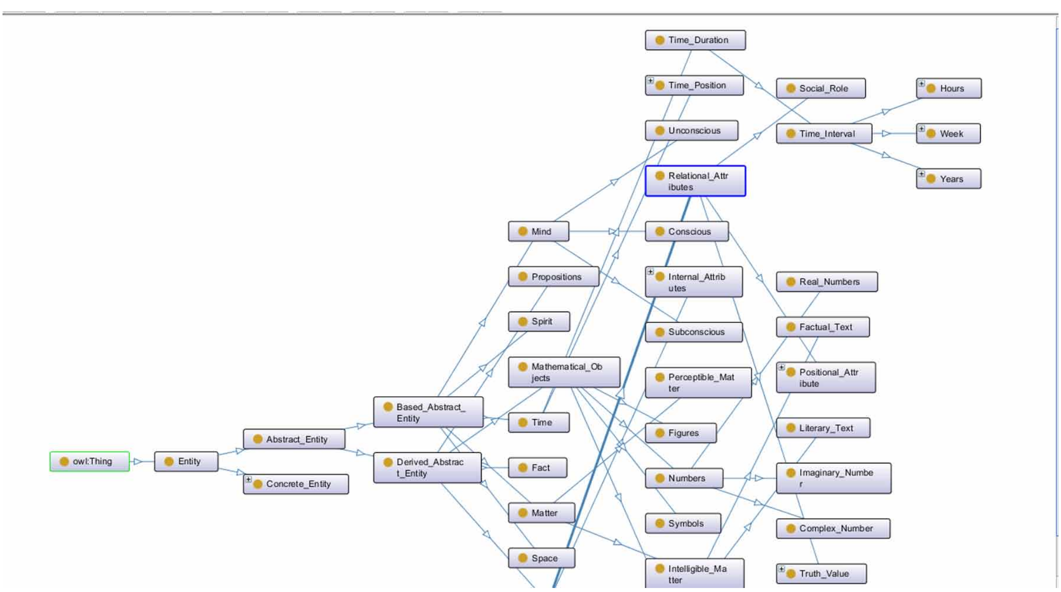


Figure 7. Onto graph for abstract entity classification



representations only once. The filling of operators is by reference and not by value. Its structure will allow extensibility and flexibility. It is easier to use in a wide range of applications. These all the features are due to the unique representational framework which is an EHCPR.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation and thanks to the reviewers for their careful and thorough critical appraisement of the paper.

REFERENCES

Basic Formal Ontology and Medical Ontology. (2003). http://ontol-ogy.buffalo.edu/bfo/BFO.html

Casellas, N., Blázquez, M., Kiryakov, A., Casanovas, P., Poblet, M., & Benjamins, R. (2005). OPJK into PROTON: Legal domain ontology integration into upper-level ontology. *Proceedings of the 3rd International Workshop on Regulatory Ontologies*, (3762), 846–855. doi:10.1007/11575863_106

Chandrasekaran, B., Josephson, R. J., & Richard, V. B. (1999). What Are Ontologies, and Why Do We Need them. *IEEE Intelligent Systems*, *1*(14), 20–26. doi:10.1109/5254.747902

Farar, S., & Bateman, J. (2004). General Ontology Baseline-[Onto Space]. *Deliverable*, *D1*, I1.

Grenon, P. (2003). *BFO in a nutshell: A bi-categorical axiomatization of BFO and comparison with DOLCE, Technical Report IFOMIS*. University of Leipzig.

Gruber, T. (1991). The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases in Principles of Knowledge Representation and Reasoning. *Proceedings of the 2nd International Conference*.

Gupta, C., & Bharadwaj, A. (2017). Research Directions under the Parasol of Ontology Based Semantic Web Structure. *Advances in Intelligent and Soft Computing,* (614).

Herre, B., Heller, P., Burek, R., Hoehndorf, F., & Michalek, H. (2006). *General formal ontology (GFO) – part I Basic principles.* Technical Report 8, Onto-Med, University of Leipzig.

Jain, J., Grover, A., Thakur, P. S., & Choudhary, S. K. (2015). Trends, problems and solutions of recommender system. *International Conference on Computing, Communication & Automation (ICCCA).* doi:10.1109/CCAA.2015.7148534

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

Jain, S., & Jain, N. K. (2010a). Acquiring Knowledge in Extended Hierarchical Censored Production Rules (EHCPRS) System. *International Journal of Artificial Life Research, IGI Global, Hershey, PA, USA*, *1*(4), 10–28. doi:10.4018/jalr.2010100102

Jain, S., & Jain, N. K. (2010b). Representation of Defaults and Constraints in EHCPRs System: An Implementation. *International Journal of Adaptive and Innovative Systems.*, *1*(2), 105–120. doi:10.1504/IJAIS.2010.032274

Jain, S., & Jain, N. K. (2012). Learning techniques in Extended Hierarchical Censored Production Rules (EHCPRs) System. *Artificial Intelligence Review*, *38*(2), 97–117. doi:10.1007/s10462-011-9242-x

Jain, S., & Jain, N. K. (2014). Globalized Intelligent System. *Int. Conf. on Computing for Sustainable Global Development*, 425-431.

Jain, S., & Mishra, S. (2014). Knowledge Representation with Ontology. *International Journal of Computers and Applications*, (6), 1–5.

Jainlbrary. (n.d.). http://www.jainlbrary.org/JAB/11_JAB_2015_Manual_Finpdf

Kapoor, A., & Sharma, S. (2013). A comparative study ontology building tools for semantic web applications. *International Journal of Web & Semantic Technology, 1*(3), 1-13.

Kiryakov, K., Ivanov, S., & Dimitrov, M. (2001). OntoMap: portal for upper-level ontologies. *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, 47–58.

Lenat, D. (2008). Voice of the Turtle: Whatever Happened to AI? *AI Magazine, 2*(29).

Malik, S., Mishra, S., Jain, N. K., & Jain, S. (2015). Devising a Super Ontology. *Procedia Computer Science*, *70*, 785–792. doi:10.1016/j.procs.2015.10.118

Maniraj, V., & Sivakumar, R. (2010). Ontology languages – A review. *International Journal of Computer Theory and Engineering.*, *2*(6), 887–891. doi:10.7763/IJCTE.2010.V2.257

Marine TLO. (2015). *A Top Level Ontology for the Marine/Biodiversity Domain*. Author.

Mascardi, V., Cordì, V., & Rosso, P. (2007). *A Comparison of Upper Ontologies*. Technical Report DISI-TR-06-2 Genova Italy.

Matuszek, J., Cabral, M., & DeOliveira, J. (2006). An introduction to the syntax and content of Cyc. *AAAI Spring Symposium*.

Micra. (n.d.). http://micra.com/COSMO/COSMO.ow

Nichols, D., & Terry, A. (2003). *User's Guide to Teknowledge Ontologies*.

Niles, A., & Pease, A. (2001). Towards a standard upper ontology. *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS)*, 2–9.

Riichiro, M. (2015). *Part 3: Advanced course of ontological engineering*. Springer.

Sehgal, S., Chaudhry, S., Biswas, P., & Jain, S. (2016). A New Genre of Recommender Systems Based on Modern Paradigms of Data Filtering. *Procedia Computer Science*, *92*, 562–567. doi:10.1016/j.procs.2016.07.383

Semy, S.K., & Pulvermacher, L.J. (2004). *Toward the use of an upper ontology for U.S. government and U.S. military domains: An evaluation*. Technical Report MTR 04B0000063, the MITRE Corporation.

Sowa Upper Ontology. (1999). http://www.jfsowa.com/ontology/toplevel.htm

Standard Upper Ontology Working Group Website. (2001). http://suo.ieee.org/

Terukazu, K. (2015). Toward knowledge structuring of sustainability science based on ontology engineering. *Sustainability Science*, (4), 99–116.

Thorp, J. (2010). *Western University, Intelligible Matter in Aristotle*. The Open Repository Binghamton.

UMich. (n.d.). http://www.umich.edu/~umjains/jainismsimplified/chapter03.html

Zhang, S., Sun, Y., Peng, Y., & Wang, X. (2009). Bayes OWL: A Prototypes System for Uncertainty in Semantic Web. *Proceedings of IC-AI*, 678-684.

Sonika Malik has done B.Tech from Kurukshetra University, India in 2004 and did her Masters from MMU in 2010. She is doing her doctorate from National Institute of technology, Kurukshetra. She has served in the field of education from last 11 years and is currently working at Maharaja Surajmal Institute of Technology, Delhi. Her current research interests are in the area of Sematic Web, Knowledge representation and Ontology Design.

Sarika Jain graduated from Kurukshetra University (India) in 1998 and did her masters (MCA) 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 17 years and is currently serving at National Institute of Technology, Kurukshetra. Her current research interests are in the area of Intelligent Systems, including Knowledge Representation, Ontological Engineering and Semantic Web Technologies. She is a member of IEEE and ACM and a Life Member of Computer Society of India.