View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 70 (2015) 483 – 490
Procedia Computer Science

4th International Conference on Eco-friendly Computing and Communication Systems

A Realist Framework for Ontologies and the Semantic Web

Sanju Mishra[a*], Sonika Malik[b], N. K. Jain[c], Sarika Jain[d]

[a]*TeerthankarMahaveer University, Moradabad, U.P, INDIA,*
[b, d]*National Institute of Technolog, Kurukshetra, Haryana, INDIA,*
[c]Zakir Husain Delhi College,*University of Delhi,INDIA*

Abstract

The realist approach in Ontology Engineering speaks about representing concepts in reality that science wants to uncover. An intelligent system is highly compatible with all, if it is largely consistent, high degree integrated and having minimum redundancy to represent the stored data and knowledge. This paper discusses the representation of elements of realist ontologies by the syntactical categories of an ontology description language, namely the Extended Hierarchical Censored Production Rule (EHCPRs) framework. The EHCPR specification is quite close to the representation of entities in the real world. The representational units in the EHCPRs ontology map one by one to the realist framework. After providing the suitability of EHCPRs Ontology Language for representing realist ontologies, a mapping is provided from the widely-used Web Ontology Language (OWL) into the EHCPRs Ontology Language. This will prove as an important milestone towards constructing instrumental knowledge bases among the Semantic Web communities.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015

*Keywords:*Realist Ontology; Mapping; EHCPRs; Intelligent Systems.

1. Introduction

Various knowledge representation languages facilitate to make statements about knowledge, which are captured in multiple means. These languages are able to express semantics for both humans and computers but varying in expressivity and computational activities. They also provide multiple querying and reasoning mechanisms to interact with the knowledge base. Ontology is a novel approach for representing concepts and relationship between concepts. It is the main origin of semantic web information and web services for information exchanging in humans

*Sanju Mishra. Tel.: 08081634089.
*E-mail address:*sanju.tiwari.2007@gmail.com

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ICECCS 2015
doi:10.1016/j.procs.2015.10.087

and computers and also helpful to make intelligent systems. The web information and web services are growing incredibly and need an effective realist framework for representing the web information to both computers as well as humans. A realist framework explains the variations and similarities of ontologies[22]. Ontologies are used by concurrent technologies such as Data Integration Cloud Computing [34], Software Agents[21] and also many more.

For representing knowledge in web, languages are available such as: XML, RDF, RDFs, DAML+OIL and OWL. Although web ontology language (OWL) has high expressivity and powerful reasoning services but lacking in some specific area such as OWL does not have explicit declaration of concepts and their syntax is also not fully context-free[2]. It is required to capture knowledge with high expressivity and translate it unambiguously using a well-defined semantics as well as knowledge representation language. The growing web information makes a heavy computational load on the system which has to be designed to translate, merge, integrate, access, manipulate and maintain the web information[5]. Research in knowledge representation aims to share the information among systems, intelligent systems, web services and applications[19] hence required a consistent and comprehensive framework for information exchange such as EHCPR systems[15]. EHCPR is a unit of knowledge item which is treated as a knowledge item for facing the real world problems and hence denoted as intelligent systems. Mapping is required for binding heterogeneous and distributed ontologies. Earlier representation schemes were ER-model, UML-model but they are not able to represent complete and complex information hence these models are transformed into some other representation schemes such as XML, RDF, RDFs, DAML+OIL, OWL. But there is a need of unique representation for representing ontologies. The EHCPRs Framework provides one such ontology language.

Mapping is a solution to resolve the incompatibilities and terminological ambiguities between knowledge based systems. A mapping of ontologies in other languages is a current trend to encode the information among intelligent systems or multi-agent systems[31] and generated by various mapping systems which are concerned on accuracy and efficiency. Existing mapping systems are Falcon[7], FOAM[4], Lily[35], COMA++[6], Clio[25], Chimaera[24], Prompt[29], H-Match[3], and MAFRA[26]. But these tools have some limitations such as they consume a lot of time during the mapping of large knowledge-base such as Google Classification and Wiki Classification etc. which results in unreliable mapping of information sharing. For dynamic ontologies it is required to re-create mappings. The OBO format is user-friendly and useful in GUI-based tools like OBO-Edit. OBO formats are merged with Gene Ontology[30]. Khattak uses a unique approach[20] of Change History Log (CHL) to reduce the mapping time as compared to other existing systems. Arnold and Rahm represent a new approach STROMA[1] for defining more expressive ontology mappings which support various constraints such as IS-A and Part-Of relations between ontologies. The constraints which are used to map the ontologies are available in EHCPRs and perform efficient reasoning, learning and representations[15, 17]. EHCPRs as knowledge representation scheme[9, 10, 11, 12, 13, 14, 17] is an attempt to develop a generalized knowledge representation, reasoning and learning system better than other existing systems which also exhibits capabilities of other formalisms such as rules, frames, semantic net and logic. EHCPRs have all properties and constraints of a concept as ontology to represent the knowledge in intelligent systems. An EHCPR system promoted the reasoning and representation by using defaults and constraints for minimizing redundancy and inconsistency[27].

2. An Intelligent Framework for the Semantic Web

The term EHCPR was coined by N.K. Jain in 1999[16] as an extension to HCPR for representing knowledge as a unit in Artificial Intelligence systems. Through years, it has proved its acceptance by a series of publications[9, 10, 11, 12, 13, 14, 17]. The EHCPR structure is more natural and comprehensible than any other logically equivalent forms. The EHCPRs System exhibits variable precision logic. Reasoning with imprecise and incomplete knowledge and constraints on resources is only possible with Variable Precision Logic. The EHCPRs logic is related to Non-monotonic reasoning which investigates the problems of making revisions in decisions. The EHCPRs structure is suitable for representing defaults. A default is a rule that can be used unless it is overridden by an exception. If an exception to an EHCPR at any level of specificity is found to be true then it would block all the decisions derived or derivable from the EHCPRs-tree with the blocked EHCPR as its root. In this way, the reasoning in the EHCPRs system is related to the non-monotonic reasoning. The EHCPRs System provides support for representing and reasoning with uncertainty. The knowledge is structured in its most general form with minimum level of redundancy and inconsistency. Total inheritance without reversal or change in preconditions and Inheritance with or without reversal of Has_Part and Has_Property operators from general classes is a very important feature of the EHCPRs system.

Several extensions/generalizations of the EHCPRs System have been proposed and recently efforts are being made to establish it as an intelligent framework for representing knowledge in the Semantic Web. In 2012, it was envisioned that the representation units of an EHCPR are just parallel to the representational units of ontology[18, 27, 28]. We present the various operators in the definition of an EHCPR in table 1.

Table 1: Structure of EHCPR

Operator	Description
A	Decision / Concept / Head of the Rule
B (B1, B2, ……Bm)	Preconditions for the decision A to be true
C (C1, C2, ……Cn)	Censored condition or exceptions for If-Then-Unless rule.
G	Generality of the Concept.
S (S1, S2,……Sp)	Specificity of the Concept.
Has_Part	Has_Part operator relegates the specific physical characteristics of Concept.
Has_Property	Has_Property operator relegates specific properties of Concept.
Has_Instance	Has_Instance operator relegates specific individuals, instances, example of Concept.
(γ)	0 – level strength of implication
(δ)	1 – level strength of implication

Table2: Structure of Instance

Operator	Description
Head	Represents the name of Instance
Instance_Of	Represents the name of EHCPR of which Head is an instance.
Has_Part	Overrides Has_Part attributes of instance.
Has_Property	Overrides Has_Property attributes of instance.

There are two types of attributes for an entity to be represented as an EHCPR: The attributes which are necessary and sufficient for an individual to be termed as an instance of this entity are listed with the operator B as conjunction of B1, B2, B3, ... Bm and are called the defining properties of the entity. These defining properties are implicitly inherited by all the subclasses of this entity and also by all the instances of this entity. The attributes which are cancellable for the subclasses of an entity and also for the instances are called the characteristic properties and are listed with the Has_Part and the Has_Property operators.

The EHCPRs structure exhibits variable precision and provides mechanism for handling incomplete information and resource constraints. Two aspects of precision are specificity of conclusions and certainty of belief in them. The specificity of conclusions refers to the degree of detail of a description and is handled by the operators G and S. The operator G refers to the general EHCPR (super class) of the EHCPR mentioned. If the general EHCPR is known to be false then the decision ‘A’ can never be true. The operator S as a mutually exclusive set of specific information S1, S2, ….Sp is a clue of the next set of concepts which are most likely to be satisfied after successful execution of this EHCPR. The certainty refers to degree of belief in a statement and is dealt with the operator C as a disjunction of C1, C2,….,Cn. When resources are tight and there is little or no information about the censor, the decision A may be taken. Later on when it is known that some censor Cx is true with some certainty δx, the decision A may be retraced with higher certainty by calculating the value of δ and hence the certainty factor (γ) of the decision A. Similarly if resources permit then more specific answer may be given based on the specificity information. The “IF B THEN A” part of the EHCPR represents causal relationship and the UNLESS operator acts as a switch that changes the decision A to ~A when C holds. The censors also make exceptions explicit and hence facilitating the rule repair. All is in motion. Our knowledge is always incomplete and subject to modifications so the representation scheme should allow for easy and natural modifications. When a rule which has worked well in past encounters a situation in which it fails, it is possible to: continue to use the rule as exceptions are rare but not a good option; invalidate the rule, but not a good option because it has worked well in many situations; modify the rule or rewrite the rule from scratch but both options require time and effort; remember the exception conditions is a good option as it preserves the usefulness of the rule and prevents us from making mistakes in the situation of exceptions.

The Has_Part and the Has_Property operators accredits the characteristic features, which normally holds true, but they are allowed to be false for an item or individual which is an instance of that particular concept in an extraordinary situation. The Has_Instance operator represents the list of distinct individuals, examples, or instances, of that particular concept. Parameter γ is a numeric measure of the strength of IF relationship between A and B. Every censor (c1, c2…,cn) is associated with an estimate of its likelihood (δ1,…,δn) which is also called the certainty factor of that censor.

$\delta = \gamma +$ *summation of all* δ's. δ is referred to as the 1-level strength of implication (which is the maximum). The value of δ should be greater than 0.5 and less than equal to 1for a substantial and hence suitable implication.

Each data item i.e. instance has a uniform structure in database and a general form as in table2.

All concepts are stored in the knowledge base with various instances. For example Chini and Chiku are instances of concept Animal and to be more precise they are instances of 'Sheep' and 'Giraffe'. In table 2, Head is the name of instance or data item. Instance_of is the name of concept (EHCPR) for which it is instantiated. All the attributes, say it be defining or characteristic are inherited through the Instance_Of operator. The Has_Part and Has_Property operators hold the override or peculiar attributes of instances. Table 3 shows a sample knowledge base and a sample database.

Table 3: Example of EHCPRs and Instances

Knowledge Base		
Concept: Animal If("Can,t make their food, Can move independently") UNLESS (Fish, Insect) GEN: (Living Being) SPEC: Vegeterian, Non-Vegetarian Has_Part: {Eyes: 2{0-4}, Legs: 4 {0-36},Tail: 1{0, 1}, Teeth: Yes {Yes, No},} Has_Property: { Eats: Grass{Straw, Grass, Leaves, Flesh}, Habitat: Earth, Hear: Yes {Yes, No}, Colour: Brown {Red, Yellow , Green, Black, White}, Sight: Good {Good, Bad}} Has_Instance:	**Concept:** Vegetarian If("Does not eat meat, Dependent on plant food") UNLESS (nil) GEN: (Animal) SPEC: Sheep, Cow, Girrafe Has_Part: Has_Property: Eats: Grass Has_Instance:	**Concept:** Non-Vegetarian If("Eats meat, Dependent on Fleshy food") UNLESS (...) GEN: (Animal) SPEC: Lion, Tiger, Dog Has_Part: Has_Property: Eats: Flesh Has_Instance:
Concept : Sheep IF("Human domesticated, Comfortable in flocks, Source of wool") UNLESS (nil) GEN: (Vegetarian) SPEC: Merino, Texel, Dorper Has_Part: Has_Property: Has_Instance: Chini	**Concept :** Giraffe IF ("Have a long Neck, Dependent only on tree leaves and grass") UNLESS (nil) GEN: (Vegetarian) SPEC: Has_Part:Neck:Long{Long, Short, Medium} Has_Property: Eats: Leaves{ Leaves, grass} Has_Instance: Chiku	**Concept** : Lion IF ("Having Big Paws for hunting, Contains impressive mane, Spiky tongue") UNLESS (nil) GEN: (Non-Vegetarian) SPEC: Has_Part: Has_Property: Eats:Flesh{ Flesh} Has_Instance: Jack
DataBase		
Chini Instance_Of (Sheep) Has_Part: Has_Property:	Chiku Instance_Of (Girrafe) Has_Part: Has_Property:	Jack Instance_Of (Lion) Has_Part: Has_Property:

Ontological realism is a philosophical viewpoint which provides an annotational framework for various ontologies. There are two types of entities[22] Universals and Particulars. Universals are entities which can be instantiated and are represented uniquely by an EHCPR (Table 1) in the EHCPRs framework. Particulars or Individuals are ones which cannot be repeated or instantiated and are represented uniquely by an Instance (Table 2) in the EHCPRs framework. To achieve useful results i.e., meaningful reasoning that correspond to reality; in a formal ontology,

- A universal is specified by means of necessary properties. For particulars to be instances of a universal, it must satisfy all the necessary properties. For example, Consciousness attributed to a Living Being.
- A universal must not list accidental properties in its list of properties. Accidental properties are the properties of a universal which its instances may or may not have. For example "possess teeth" and "possess stomach" are accidental properties and not necessary conditions of "Living Beings". Since a Sea Horse is a Living Being but it has neither teeth nor a stomach. Also an old man having all broken teeth does not have teeth but is still a Living Being.

The definition of an EHCPR contains two sets of attributes: Defining and Characteristic. Defining attributes are essential pre-conditions for any concept and characteristic attributes are the attributes which explicitly defines the concept. For example, in the EHCPR of Bird, "If Bipedal, Lay Eggs" preconditions that is an essential property of any Bird while "Wings, Legs, Feathers, Beak etc. and Fly, Habitat, Voice, Colour etc" are the characteristic properties of Bird EHCPR which characterize the features of Bird. As is required by the realist ontology for meaningful reasoning, the EHCPRs framework have all the necessary properties mentioned in the defining part i.e. with the operator B and all the cancellable properties or the accidental properties mentioned with the characteristic attributes i.e., with the Has_Part and the Has_Property operators.

In ontological realism, instances are absolutely determinate. Meaning, instead of property 'Having Mass', an instance should have a property 'Weighs one Kg exactly'. As depicted in Table 2, an instance of an EHCPR contains the Has_Part and Has_Property lists which contains overridden specific default values for each attribute. For example, Titu is an instance of EHCPR Bird. So Titu inherits 2 legs from the EHCPR of Bird via the Instance_Of operator. But Titu has one leg broken. So in the Has_Property list of Titu, 2 legs are overridden by 1 leg as specific default.

3. Mapping of OWL Ontologies and EHCPR

EHCPRs systems are more consistent and efficient than any existing systems due to their various properties such as more realistic, close to human intelligence and practically acceptable. Ontologies use the UML model to represent the concept classes and their association. These UML classes are then converted into OWL ontologies and attributes of UML class are converted into properties in OWL[33]. The uniqueness of OWL is represented by cardinality where max_cardinality maps to upper bound and min_cardinality maps to lower bound while unlimited bound is mapped by 'SomeValue'[32]. In this section, OWL ontologies are mapped into EHCPR ontology which facilitates semantic integration and searching and maps it to a general model. The EHCPRs framework has been proved to be closer to the ontological realism and is a better solution for representation, reasoning and learning in intelligent systems. They are well suited ontologies due to some enhanced features and are capable to build an intelligent system. Ontology Mappings are useful for evolution of ontologies and their integration. The compatibility of OWL ontologies and EHCPRs can be mapped by the table 4.

Table 4: Compatibility of OWL and EHCPRs

OWL	**EHCPRs**
OWL syntax is not fully context-free.	EHCPRs contain plenteous information to correctly disambiguate the axioms.
OWL does not facilitate for explicit declaration of concepts	EHCPR s structure declares all the concepts explicitly.
Due to absence of explicit declaration of concepts OWL is unable to check consistency of ontologies.	High efficiency and effectiveness achieved in EHCPRs due to their explicitly declared structure.
OWL abstract syntax is structurally different from OWL RDF syntax.	EHCPRs have a unique structure for all types of representation.

Relationship between EHCPRs can be defined using the hierarchical concepts of domain. A defined relationship is like a binary predicate and consists of a subject, a relationship type and an object. EHCPRs systems have link lists. EHCPRs have search & growth learning algorithms. EHCPR concepts are mapped with OWL classes such as (owl: class) where child classes are described with subclass relationships such as (rdf :SubClassOf). Relationships between EHCPR ontology and OWL are encoded by the relationship tag at the concept level. In EHCPRs framework, the general and the specific classes are represented by the generality and the specificity operators while in OWL ontology it is explained by SubClassOf relationship. In order to get the absolute semantics intended in the EHCPR ontology over OWL, the relationships are translated in form of Has_Part, Has_Property, Has_Instance operator in EHCPR. In EHCPR, each concept has its own parts, properties and also their instances. All parts of the concept are described by has_part operator, properties and instances of concept described by Has_Properties and Has_Instance operator of EHCPRs. In OWL the properties are expressed by object properties and data properties. As mentioned in table 5, all data types of concept use data properties while attributes use object properties.

Table 5. OWL and EHCPRs Structure

OWL		**EHCPRs**	
OWL Class		EHCPR Class	
Property	Data Type Property	Property	Defining Property (Preconditions)
	Object Type Property		Characterstics Property(Has_Part and Has_Property)
Instance		Has_Instance and Instance_of Operator	
Cardinalty(Max, Min)		Default and Constraints manage the cardinality.	

EHCPR concepts are mapped with OWL classes such as (owl: class) where child classes are described with subclass relationships such as (rdf :SubClassOf). The following table displays the mappings of EHCPR and OWL.

Table 6. Mapping of OWL Ontologies into EHCPRs

OWL Class(People.owl)	EHCPRs
<owl:Classrdf:about="http://owl.man.ac.uk/2005/07/sssw/people#animal"> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty> <owl:ObjectProperty rdf:about="http://owl.man.ac.uk/2005/07/sssw/people#eats"/> </owl:onProperty> <owl:someValuesFromrdf:resource="http://www.w3.org/2002/07/owl#Thing"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:labelrdf:datatype="http://www.w3.org/2001/XMLSchema#string" >animal</rdfs:label> </owl:Class>	■ EHCPR of Animal Concept: Animal If(b1, b2....) UNLESS(c1,c2....) GEN: (Thing) SPEC: Has_Part: Has_Property: Eats {} Has_Instance:
<!-- http://owl.man.ac.uk/2006/07/sssw/people#sheep --> <owl:Classrdf:about="#sheep"> <rdfs:labelrdf:datatype="&xsd;string">sheep</rdfs:label> <rdfs:subClassOfrdf:resource="#animal"/> <rdfs:subClassOf> <owl:Restriction> <owl:onPropertyrdf:resource="#eats"/> <owl:allValuesFromrdf:resource="#grass"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:commentrdf:datatype="&xsd;string"></rdfs:comment> </owl:Class>	■ EHCPR of Sheep Concept : Sheep IF ( ) UNLESS ( ) GEN: (Animal) SPEC: Has_Part: Has_Property:Eats: Grass{Grass} Has_Instance:
<!-- http://owl.man.ac.uk/2006/07/sssw/people#cow --> <owl:Classrdf:about="#cow"> <rdfs:labelrdf:datatype="&xsd;string">cow</rdfs:label> <rdfs:subClassOfrdf:resource="#vegetarian"/> <rdfs:commentrdf:datatype="&xsd;string">Cows are naturally vegetarians.</rdfs:comment> </owl:Class> <!-- http://owl.man.ac.uk/2006/07/sssw/people#giraffe-->	■ EHCPR of Cow Concept : Cow IF ( ) UNLESS ( ) GEN: Vegetarian SPEC: Has_Part: Has_Property: Has_Instance:

<owl:Classrdf:about="#giraffe"> <rdfs:labelrdf:datatype="&xsd;string">giraffe</rdfs:label> <rdfs:subClassOfrdf:resource="#animal"/> <rdfs:subClassOf> <owl:Restriction> <owl:onPropertyrdf:resource="#eats"/> <owl:allValuesFromrdf:resource="#leaf"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:commentrdf:datatype="&xsd;string"></rdfs:comment> </owl:Class>	■ EHCPR of Giraffe Concept : Giraffe IF ( ) UNLESS ( ) GEN: (Animal) SPEC: Has_Part: Has_Property: Eats:Leaf{ Leaf, grass} Has_Instance:

In this table People.owl file is considered for mapping in EHCPRs ontology. In this ontology various concepts are described with their attributes and properties. The main class of owl is ‘Thing’ under which all classes are declared. In people.owl *Animal* is sub class of *Thing* which has only *eats* object property so EHCPR of animal has only *eats* property. Other properties are not declared of *Animal* class in people.owl so other operators of Animal EHCPR are blank at present in this mapping table. Later on these operators will also be declared according to requirement by human counterpart for all EHCPRs. Sheep, Vegetarian and Giraffe classes are subclasses of Animal class.

Conclusions

Key research issues of any research groups are interoperability and information exchange. The proposed scheme uses the concept of EHCPR as a realist framework for ontologies. This realist framework presented a mapping of OWL ontology with EHCPRs which facilitated the interoperability and information exchange between knowledge based systems. The OWL ontology can be translated into EHCPRs without loss of knowledge. Our future work will focus to implement Knowledge Base and Databases of EHCPRs and also reduce the consumed time.

References

1. Arnold P., Rahm E. Enriching ontology mappings with semantic relations, Data & Knowledge Engineerin 2014; 93: 1–18.
2. Motik B., Horrocks I. Problems with OWL syntax. In *OWLED2006 Second Workshop on OWL Experiences and Directions* 2006; 216.
3. Castano S., Ferrara A., Montanelli S. Matching ontologies in open networked systems. Tech. Appl. *J. Data Semantics* (JoDS) 2006; 25–63.
4. Ehrig M., Sure Y. Foam – framework for ontology alignment and mapping; results of the ontology alignment initiative.*In: Proc. of the Workshop on Integrating Ontologies*; 2005.
5. Flouris G., Manakanatas D., Kondylakis H. Plexousakis D. Antoniou G. Ontology change: classification and survey, *Knowl. Eng. Rev. (KER*) 2008; **23**(2): 117–152.
6. H.-H. Do. Schema Matching and Mapping-based Data Integration. *PhD Thesis.*Department of Computer Science. Universitat Leipzig; 2006.
7. Hu W., Qu. Y. Falcon-AO: a practical ontology matching system.*J. Web Semantics 2008;* 6 (3): 237–239.
8. Jain S. and Jain N. K.A Globalized Intelligent System, Int. *Conf. on -Computing for Sustainable Global Development*; 2014. p. 425-431.
9. Jain NK, Jain S, Goel C. K. A generalized knowledge representation system and its implementation: an extended hierarchical censored production rules system (EHCPRs). MERI, *J Manag IT. 2007;* 1:88–101.
10. Jain S, Jain NK, Goel C. K. Implementing general control scheme in EHCPRs system. Proceedingsof second national conference on mathematical techniques: emerging paradigms for electronics and ITindustries; 2008a. p. 347–353.
11. Jain S, Jain NK, Goel C. K. EHCPRs system and management. *JIPEM*2008b; **3**(1):1–17.
12. Jain S, Jain NK, Goel C. K. Reasoning in EHCPRs system. Int*J Open Problems Comput Math 2009a;* **2**(2): 173–193.
13. Jain S, Jain N.K. Representation of defaults and constraints in EHCPRs system: an implementation *Int. J Adapt InnovSyst* 2009b*;* **1**(2): 105-120.
14. S. Jain, Jain N. K. A generalized knowledge representation system for context sensitive reasoning: generalized HCPRs system. *in Artificial Intelligence Review, (Springer*) 2009; **30**: 39– 42.
15. S. Jain, Jain N. K. Acquiring knowledge in extended hierarchical censored production rules (EHCPRs) system. *International Journal of Artificial Life Research* 2010; **1**(4): 10–28
16. Jain, N. K., Bharadwaj, K. K., Marranghello Norian. Extended Hierarchical Censored Production Rules (EHCPRs) system: An approach toward generalized knowledge representation. *Journal of Intelligent Systems*. 1999; **9**(3- 4): 259-295.
17. Jain S., Jain N. K. Learning techniques in Extended Hierarchical Censored Production Rules (EHCPRs) System. *ArtifIntell Rev* 2012; **38**: 97-117.
18. Jain S., Mishra S. Article: Knowledge Representation with Ontology. *IJCA Proceedings on International Conference on Advances in Computer Engineering and Applications ICACEA* 2014; **6**: 1-5.
19. Khattak A.M. et al. Mapping evolution of dynamic web ontologies, *Inform. Sci*. 2015; **303**: 101–119.

20. Khattak A. M., Latif K., Lee S. Y. Change management in evolving web ontologies. *Knowl.-Based Sy;* 2013. 37. p. 1–18.
21. Kless D., Milton, S., Kazmierczak E. Relationships and Relata in Ontologies and Thesauri: Differences and Similarities. *In: Applied Ontology* 2012;7:4. p. 401–428.
22. Kless D., Janson L. How fit is OWL to represent realist ontologies? The semantics of representational units in realist ontologies and the Web Ontology Language; 2014.
23. Khan W. A., Bilal M., Khattak A. M., Hussain M., Afzal M., Lee S. Y., Kim E. S. Object oriented and ontology alignment patterns based expressive mediation bridge ontology (MBO). *J. Inf. Sci.* 2014; **41**(3): 1-19.
24. McGuinness D. L., Fikes R., Rice J. and Wilder S. The chimaera ontology environment. *In Proceedings of AAAI 2000.* p. 1123–1124.
25. Miller R. J., Haas L. M., and Hernandez M. A. Schema mapping as query discovery. *In VLDB* 2000. p. 77–88.
26. Maedche A., Motik B., Silva N., Volz R. MAFRA – a MappingFRAmework for distributed ontologies, in: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. London; 2002.p. 235–250.
27. Mishra S., Jain S. A Study of Various Approaches and Tools on Ontology, *IEEE conference CICT 2015* in ABES College of Engineering (Ghaziabad); 2015. p. 57-61.
28. Mishra S., Jain S. Automatic Ontology Acquisition and Learning, IJRET proceedings of *3rd International Conference on System Modeling & Advancement in Research Trends (SMART)* 2014; **3**(4): 38-43.
29. Noy N., Musen M. The PROMPT suite: interactive tools for ontology merging and mapping.*Int. J. Hum. Comput. Stud* 2003; **59** (6): 983–1024.
30. OBO Foundry Homepage: The Open Biological and Biomedical Ontologies, http://www.obofoundry.org/. 2012.
31. Otero-Cerdeira L., Rodrıguez-Martınez F. J., and Gomez-Rodriguez A. Ontology matching: A literature review. *Expert Systems with Application 2015;*42(2): 949–971.
32. Park D. and Park S. Ontology Mapping for Enhanced Interoperability of S-100 Geographic Information Registers.*International Journal of Software Engineering and Its Applications*2014;8(1): 225-234.
33. Pabitha P., Priya A. S. and Rajaram M. An Approach for Detecting and Resolving Inconsistency using DL Rules for OWL Generation from UML Models. *European Journal of Scientific Research* 2012; 72(3): 404-413.
34. Tao F., Cheng Y., Xu L. D., Zhang L. Li B. H. CCIoT-CMfg: Cloud Computing and Internet of Things Based Cloud Manufacturing Service System. *IEEE Trans. Ind. Inform*2014; 10: 1435-1442.
35. Wang P., Xu B. Lily: ontology alignment results for oaei. Ontology Matching (OM). *in Proc. 4th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*; 2009. p. 186–192.