Artificial Intelligence of the Web through Domain Ontologies

K. C. Gouda[1], Nagaraj Bhat[2], K. Deepan Siddarthan[3]

[1, 2] CSIR Centre for Mathematical Modelling and Computer Simulation, Wind tunnel Road, Bangalore-37 Karnataka, India
[3]Dept of IT, Jeppiaar Engineering College, Anna University, Chennai, 600 025, India

Abstract: ***The increased potential of the ontologies to reduce the human interference has wide range of applications. This paper identifies requirements for an ontology development platform to innovate artificially intelligent web. To facilitate this process, RDF and OWL have been developed as standard formats for the sharing and integration of data and knowledge. The knowledge in the form of rich conceptual schemas called ontologies. Based on the framework, an architectural paradigm is put forward in view of ontology engineering and development of ontology applications and a development portal designed to support ontology engineering, content authoring and application development with a view to maximal scalability in size and complexity of semantic knowledge and flexible reuse of ontology models and ontology application processes in a distributed and collaborative engineering environment.***

Keywords: Artificial Intelligence, Webthrough, Ontologies, Semantics

1. Introduction

Ontologies are defined independently from the actual data and reflect a common understanding of the semantics of the domain of discourse. Ontology in general is an explicitly specification of a representational vocabulary for a domain; definitions of classes, relations, functions, constraints etc. and other objects in philosophy and Ontology studies the nature of being and existence. The term ‘ontology’ is derived from the Greek words “onto”, which means being, and “logia”, which means written or spoken discourse. Smith reviewed the studies on the metaphysical aspect of ontologies since Aristotle’s time, and summarized the essence of ontology as follows: “provide a definitive and exhaustive classification of entities in all spheres of being”. In contrast to these studies, Quine’s ontological commitment1 [1] drove ontology research towards formal theories in the conceptual world.

Computer scientists further extended Quine’s work into a new interpretation of ontology as “a specification of a conceptualization” [2]. In computer science and information Technology, knowledge reuse is facilitated by the use of explicit ontology, as opposed to implicit ontology, i.e., knowledge encoded into software systems [3]. Hence, appropriate ontology languages are needed to realize explicit ontologies with respect to three important aspects

Conceptualization. The language should choose an appropriate reference model, such as *entity-relationship model* and *object-oriented model*, and provide corresponding ontology constructs to represent factual knowledge, such as defining the entities and their relations in a domain, and also asserting relations among entities.

Vocabulary. Apart from the semantics, the language should also need to cover the syntax such as symbol assignment (i.e., assigning symbols to concepts) and grammars (i.e., serializing the conceptualism into explicit representation).

Axiomatization. In order to capture the semantics for inference, rules and constraints are needed in addition to factual knowledge. For example, we can use rules to generate new facts from existing knowledge, and to validate the consistency of knowledge. In order to share knowledge across different communities or domains, mainly three requirements should be considered when developing explicit ontologies

Extensibility. In the context of the Web, ontology engineers should be able to develop ontologies in an incremental manner: reusing as many existing popular concepts as possible before creating a new concept from scratch. For example, the concept “woman” can be defined as a *sub-class* of an existing concept “person” in WordNet[2] vocabulary. This requirement demands an expressive common reference model as well as distributed symbol resolution mechanisms.

Visibility. Merely publishing knowledge on the Web does not guarantee that it can be readily understood by machines or human users. To make knowledge to be visible on the Web, some of the additional common ontological ground on syntax and semantics is required between consumers and information publishers. This requirement is especially critical to machines since they are not capable of understanding knowledge written in an unfamiliar language.

Inferenceability. Ontology not only serves the purpose of representation, i.e. enumerating factual domain knowledge, but also serves the computation purpose, i.e., enabling logical inference on facts through axiomatization. Hence, ontologies on the Web should provide constructs for effective binding with logical inference primitives and options to support a variety of expressiveness and computational complexity requirements. The Semantic Web inherits the power of representation from existing conceptualisms, such as Semantic Networks [4], and enhances interoperability at both syntactic and semantic levels. It can also function as a distributed database or a collaborative knowledge base according to application requirements. In particular, the extensibility is offered not only by the underlying URI based vocabulary but also by the simple graph data model of Resource Description Framework (RDF) [5].

2. Ontologies

Ontologies play an important role in fulfilling semantic interoperability. W3C has standardized a layered stack of ontology languages that possess the advantages of both knowledge representation (KR) formalisms and conceptual modeling methods for databases. Standardization encouraged creating new ontologies and porting existing ontologies into the Semantic Web. In the Semantic Web layer cake (Figure 1), the semantic part is enabled by a stack of evolving languages: Resource Description Framework (RDF) [6] offers a simple graph reference model; RDF Schema (RDFS) [7] offers a simple vocabulary and axioms for object-oriented modeling; and Web Ontology Language (OWL) [8] offers additional knowledge base oriented ontology constructs and axioms

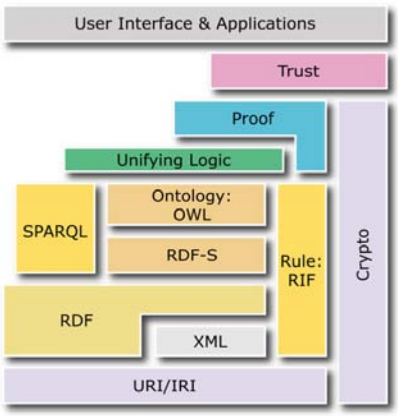

Figure 1: Shematic of Symantic Web

2.1 RDF

RDF offers a simple graph model which consists of nodes (i.e. resources or literals) and binary relations (i.e. statements). It is a type of Semantic Network and is very similar to the *Relational Model.* Such a simple model embodies a small amount of built-in semantics and offers great freedom in creating customized extensions; however, an extended or specialized semantic network is usually required in practice. John Sowa identifies six categories of semantic networks based on relation semantics:

(i) *Definitional networks*, which build taxonomies for conceptualisms with inheritance(subclass) and membership (instance) relations; (ii) *Assertional networks*, which represent cognitive assertions about the world with modal operators; (iii) *Implicational networks*, which focus on implication relations, e.g. belief network; (iv) *Executable networks*, which focus on temporal dependence relations, e.g. flowchart, PetriNet; (v)*Learning networks*, which focus on causal relations encoded in numerical value, e.g. neural network; (vi) *Hybrid networks*, which generally combine features of previous types. In the Semantic Web, most ontologies are defined using RDF(S)/OWL and thus fall in the first category; the second category (assertional networks) emerges in the context of sharing instance data and evaluating trustworthiness of such data and the third category (implicational networks) gains interests in ontology mapping study. A variation of definitional networks is natural language encyclopedia such as dictionaries and thesaurus which uses a different set of relations rather than class-property relation. WordNet 5 and Simple Knowledge Organization System (SKOS) 6 are their representative Semantic Web versions respectively.

2.2 RDFS

Under the influence of *Frame Systems* and the *Object Oriented Model*, RDFS has been used to augment RDF to provide better support for definition and classification .These models organize knowledge in a concept-centric way with descriptive ontology constructs (such as frame, facet and slot) and built-in inheritance axioms. Frame Systems enable users to represent the world at different levels of abstraction with the emphasis on entities, but this aspect also makes it quite different from the planar graph model, offered by most semantic networks (Figure 2). In addition to inheriting basic features from Frame Systems (FS), RDFS usually provides ontology constructs, that make relations less dependent on concepts: users can define relations as an instance of *rdf:Property*, describe inheritance relations between relations using *rdfs:subPropertyOf*, and then associate defined relations with classes using *rdfs:domain*or *rdfs:range*. Figure 2 presents the *rdfs* schematically.

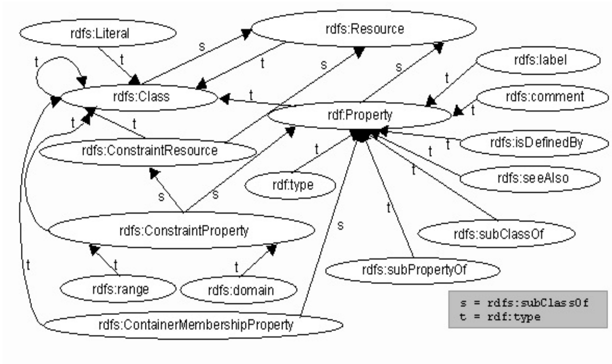

Figure 2: Representation of *rdfs*

2.3 DAML+OIL and OWL

DAML+OIL and OWL extend RDFS and emphasize support for richer logical inference. Besides inheriting advantages from Frame Systems, these ontology languages provide a rich set of constructs based on Model theoretic Semantics. Three variants of OWL trade off computational complexity and the expressiveness of ontology constructs.

OWL-Lite is the simplest variant for building a basic frame system (or an object oriented database) in terms of class, property, subclass relation, and restrictions.
OWL-Lite never uses the entire OWL vocabulary and some OWL terms are used under certain restrictions.

OWL-DL is grounded on Description Logics, and focuses on common formal semantics and inference decidability. Description logics offer additional ontology constructs (such as negation, conjunction and disjunction) besides class and relation, and have two important inference mechanisms: subsumption and consistency. The strong Set Theory background makes Description Logics suitable for capturing knowledge about a domain in which instances can be grouped into classes and relationships among classes are binary. OWL-DL uses all OWL ontology constructs with some restrictions.

OWL-Full is the most expressive version of OWL but it does not guarantee decidability. The biggest difference between OWL-DL and OWL-Full is that class space and instance space are disjointed in OWL-DL but not in OWL-Full. That is, a class can be interpreted simultaneously as a set of individuals and as an individual belonging to another class in OWL-Full. The entire OWL vocabulary can be used in without any restrictions in OWL-Full.

3. Architectural Paradigm

Ontology engineering is facilitated through wide variety of architectural paradigm. These include certain mechanism through which the ontologies are being distinguished from others. Each Framework requires different architecture for the development of Ontologies. This may vary depending upon the domain in which it is implemented.

3.1 Dogma Framework

The DOGMA framework [9] serves the purpose of ontology representation. It will be the basic constructs of ontology development architecture. The DOGMA framework is comprised of the following

Lexon is a quintuple $<\gamma, t1, r1, t2, r2>$ where $\gamma \in \Gamma$ is a *context* identifier, $t1 \in T$ and $t2 \in T$ are *terms* on and over alphabet A, $r1 \in R$ and $r2 \in R$ are roles in the semantic relationship. Γ, T and R are strings over an alphabet, $A+$.

The semantic reference of t in a lexon is generally of two kinds: concept and label types:
$C = \mathrm{ct}(\gamma, t) \mid \gamma \in \Gamma, t \in N, N \subset T$
$L = \mathrm{lt}(\gamma, t) \mid \gamma \in \Gamma, t \in \Lambda, \Lambda \subset T$
Where $N \cap \Lambda = 0$.

It is worth distinguishing the 'lexical' terms from the rest, the 'no lexical terms', for its special semantic operations. The lexical term refers to semantic symbols which themselves are strings over the same alphabet, A. Its signification, S, is
$S = \Gamma \times \Lambda \times P \mid \Gamma \subset A+, \Lambda \subset T, P \subset T, \Lambda \cap P = 0$.

Given a context $\gamma \in \Gamma$, the label, $l \in \Lambda$, yields the reference, $p \in P$ as the semantic interpretation. Lexical terms, can be used for the human interpretation of non-lexical terms in a natural language.
We can distinguish two types of lexons:
$i = <\gamma, t1, r1, r2, t2>$ where $\gamma \in \Gamma$, $r1 \in R$, $r2 \in R$, $t1 \in C$, $t2 \in C$
$b = <\gamma, t1, r1, r2, t2>$ where $\gamma \in \Gamma$, $r1 \in R$, $r2 \in R$, $t1 \in L$, $t2 \in C$

One scenario that both types of lexons will be in use is the ontology meta model about ontology models. It assumed that with a context identifier $\gamma \in \Gamma$ and $t \in T$, its concept can be uniquely identified. Context is a set of sources referring to some documents and these documents are interpreted in an abstract sense, similar to *text* in semiotics. It is one or more expressions put a different language. Intuitively, a given document source $\gamma \in \Gamma$ contextualizes and validates the relationship between two concepts. The semantic validity of the relationship is established through the developer/user agreement on the sources. Now several studies on the contextualization of lexons, contextual reasoning and modeling contexts are going on. Based on the discussed formulation, a lexon base, Ω, is a set of expressions composed from an ordered

Pair $<A, I>$ where
A is the alphabet.
$I = \Gamma \times C \times R \times C \times R$ is the set of lexons and expressing the facts as relationship
Γ, R, C are sets of strings over $A+$

3.2 MCRU ARCHITECTURE

In the database terminology M stands for *model* as. It is a subset of ontology base, defined above as $<A, I>$. It is a best way of structuring the lexon population. C is *commitment* to the lexon models with respect to the tasks. R means *rendition* of lexon models, commitments and application semantics. U is *use* for system-specific configuration of renditions for a given application. It is an architectural paradigms, which is based on DOGMA ontology representation framework presented in figure 3.

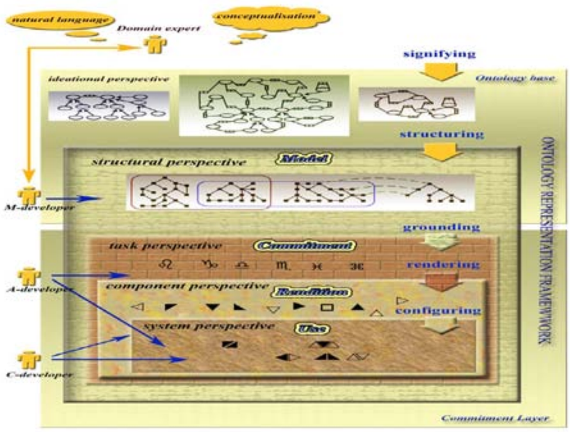

Figure 3: Framework of Ontology Representation.

4. Ontology Tools

Generally the effective enabling tools are needed for implementing Semantic Web that facilitates the creation of artificially intelligent web. Accordingly, some of the tool classes are categorized based on their features. They are depicted as described in table 1.

Table 1: List of available Ontology tools

	Import format	Export format	Graph view	Consistency check	Multi-user	Web support	Merging
Protégé 2000	XML, RDF(S), XML Schema	XML, RDF(S), XML Schema, FLogic, CLIPS, Java html	Via plug-ins like GraphViz and Jambalaya	Via plug-ins like PAL and FaCT	Limited (Multi-user capability added to it in 2.0 version)	Via Protégé-OWL plug-in	Via Anchor-PROMPT plug-in
OilEd	RDF(S), OIL, DAML+OIL	RDF(S), OIL, DAML+OIL, SHIQ, dotty, html	No	Via FaCT	No	Very limited namespaces	No
Apollo	OCML, CLOS	OCML, CLOS	No	Yes	No	No	No
RDFEdt	RDF(S), OIL, DAML, SHOE	RDF(S), OIL, DAML, SHOE	No	Only checks writing mistakes	No	Via RSS (RDF Site Summary)	?
OntoLingua	IDL, KIF	KIF, CLIPS, IDL, OKBC syntax, Prolog syntax	No	Via Chimaera	Via write-only locking,user access levels	Yes	?
OntoEdit (Free version)	XML, RDF(S), FLogic and DAML+OIL	XML, RDF(S), FLogic and DAML+OIL	Yes	Yes	No	Yes	?
WebODE	RDF(S), UML, DAML+OIL and OWL	RDF(S), UML, DAML+OIL,OWL, PROLOG, X-CARIN, Java/Jess	Form based graphical user interface	Yes	By synchronization, authentication and access restriction	Yes	Via ODEmerge

5. Conclusions

Artificially intelligent Web can be created by implying semantics in each and every operations of the web. The implementation of semantics can enhance the ability of the machines to process the web pages. This can be achieved by the use of ontologies. Ontologies serve as a knowledge representation language, similar to the natural language that we humans use, to pave way for the machines to efficiently deliver the expected outcome or result. Ontologies can be created through several tools by certain ontological engineering paradigm such as DOGMA. The DOGMA approach advocates the independence of semantic models from information systems with a layer that mediates between ontology and applications. This drives the need for decoupling general semantics from specific application logic as development methodology, thus enabling the efficient use, reuse as well as sharing of semantic models and processes across systems, applications and developments. Thus Ontology facilitates the implantation of artificially intelligent web

References

[1] Quine WVO. On What There Is. Review of Metaphysics 1948; p. 21–38.

[2] Gruber ThomasR. A Translation Approach to Portable Ontology Specifications. KnowlAcquis 1993; 5(2):199–220.

[3] Uschold Mike, Gr¨uninger Michael. Ontologies: principles, methods, and applications. Knowledge Engineering Review 1996;11(2):93–155

[4] Sowa John, editor. Principles of Semantic Networks: Explorations in the Representation of Knowledge. San Mateo: Kaufmann; 1991.

[5] Klyne Graham, Carroll JeremyJ. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/; 2004.

[6] Brickley Dan, Guha RV. RDF Vocabulary Description Language1.0: RDFSchema. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/; 2004.

[7] Dean Mike, Schreiber Guus. OWL Web Ontology Language Reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/; 2004.

[8] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American2001; 284(5):35–43.

[9] Spyns, Peter, et al. "OntoWeb-a semantic Web community portal." Practical Aspects of Knowledge Management. Springer Berlin Heidelberg, 2002. 189-200.

Author Profile

K C Gouda is currently working as a Scientist at CSIR Centre for Mathematical Modeling and Computer Simulation (CSIR C-MMACS). His research and professional career spans about twelve years of research and capacity building in modeling and computer simulation, satellite data processing, numerical modeling, Data mining, Data assimilation, cloud computing knowledge engineering and related subjects. His expertise is primarily in the domains of Software development for modeling and simulation. He is presently involved in several international and national projects related to HPC enabled modeling for weather and climate forecasting and analysis. He has published about 75 peer-reviewed papers as journal articles, book chapters, Technical reports and contributions to conference proceedings. He is a member of IMS, IEEE, AGU, AOGS and EGU. He is also a member in the board of studies of Department of Computer Science in the Jain University and Dayananda Sagar Institutes, Bangalore. He obtained his M.Sc., M.Phil, from Berhampur University, MCA from IGNOU, New Delhi and completed PhD from Mangalore University. He has Guided 18 M.Tech., 55 Masters and 45 B.E students for their academic project.

Nagaraj Bhat is currently working as a Project Scientist at CSIR Centre for Mathematical Modeling and Computer Simulation (CSIR C-MMACS), Bangalore, India. His research interest includes satellite remote sensing, Cyclone studies, software development for remote sensing data analysis and Geographical Information System modeling. He obtained B.E (Computer Sc.) during 2009 and M. Tech degree in Geo-informatics in 2011 from Visvesvaraya Technological University (VTU), Karnataka. He has published 12 technical papers in journals, international and national conference proceedings.

K.Deepan siddarthan presently pursuing his B. Tech (Information Science) at the Department of ISE, Jeppiar Engineering College, Anna University, Chennai, 600 025 His Research interests includes Semantic web, Image Processing, Big data analysis, Simulation and Computer Networks. He is working for his B. Tech final semester technical project at CSIR C-MMACS under SPARK programme under the guidance of K C Gouda. He has presented two papers in the International conferences.