Information Technology Journal 3 (2): 163-167, 2004
ISSN 1682-6027

A Journey from Information to Knowledge: Knowledge Representation and Reasoning on the Web

Qazi Mudassar Ilyas, Yang Zongkai and Muhammad Adeel Talib
Department of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan, People's Republic of China

Abstract: World Wide Web contains tons of information that is increasing everyday. Search engines and catalogues have been trying hard to make the life of users easier by indexing this ever growing information. They, however, suffer from the fact that web is not machine-understandable that makes the information retrieval very difficult. Moreover, the demands of the users are also increasing. When performing a task, e.g., an individual planning a vacation trip or an organization finding and integrating with another organization, the users do not want to find and fix all the pieces of the jigsaw by themselves. They want computers to handle the details automatically. Semantic Web comes forward to embrace this challenge by annotating the web with knowledge and performing reasoning on this knowledge hence entering into the age of knowledge from the present age of information. Semantic Web makes the web machine-understandable that makes the use of intelligent agents possible that can accomplish complex tasks automatically with minimum human intervention. Knowledge representation and reasoning on this knowledge are the two main components of Semantic and they are the focus of this paper.

Key words: Inference engine, knowledge representation, metadata, ontology, Semantic Web, web technologies

INTRODUCTION

World Wide Web has grown enormous since its appearance on the globe in 1991. The main reason for this growth was the distributed and decentralized nature of the web. This decentralized nature also created the problems for the users. Due to the huge size of the web, finding right information has become a nightmare for the users. Moreover, in addition to finding the information, individuals and organizations also want to perform some tasks on the web e.g., a user might want to plan a vacation trip or an organization might wish to integrate with a new organization on the fly. These tasks can be performed on the web today manually but require visiting a series of web pages, understanding their content, establishing their relationships and taking some decisions. The web of today is unable to do this because of the limitation that the web was basically designed for human beings and cannot be processed automatically by automated agents. But the question is can we ever achieve these goals?

Yes, we can. The answer to all the questions is Semantic Web. Semantic Web achieves these goals by precisely defining the concepts used in the web pages that makes the web machine-understandable thus allowing the existence of automated agents on the web.

Semantic Web: the web of tomorrow: Tim Berners-Lee, inventor of the web, defines the Semantic Web as follows[1]:

"The Semantic Web is not a separate web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

The Semantic Web makes it possible for machine-readable annotations to be added, linked to each other and used for organizing and accessing web content. Thus, the Semantic Web offers new capabilities, made possible by the addition of documents that encode the "knowledge" about a web page, photo, or database, in a publicly accessible, machine-readable form. Driving the Semantic Web is the organization of content into specialized vocabularies, called ontologies that can be used by web tools to provide new capabilities.

Improved search is only one of the many potential benefits from the Semantic Web. Internet agents, that are autonomous program that interact with the Internet, can

Corresponding Author: Qazi Mudassar Ilyas, Room No. 107, Foreign Students Building, Huazhong University of Science and Technology, Wuhan–430074, Hubei, People's Republic of China
Tel: 00-86-27-87461562 E-mail: mudassir_ilyas@hotmail.com, qmudassir@yahoo.com

also benefit from the Semantic Web. These agents will be capable of performing complete tasks like given above and even much more complicated. The Semantic Web enables users to locate, select, employ, compose and monitor web services automatically[2]. Push systems can also be implemented by using Semantic Web techniques, which push the documents toward the users according to the criteria given by them instead of requiring the users to find the content on the web.

Tim Berners Lee[3] has proposed a layered model of Semantic Web (Fig. 1).

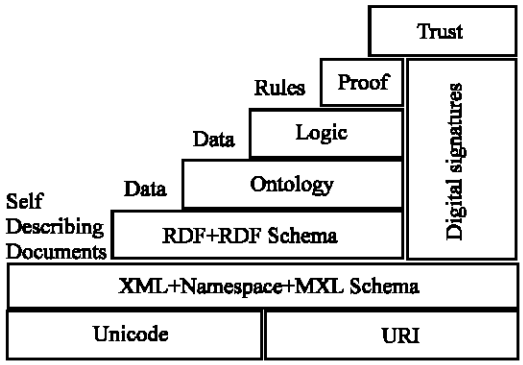

Fig 1: Layered model of Semantic Web proposed by Lee[2]

Starting from bottom, the first two layers are the basis of Internet. The Semantic Web starts from the third layer, RDF and RDF Schema. Their main goal is the description of data discussed in detail in the next sections.

The fourth layer, ontology, provides more expressivity because the definitions of RDF Schema are not enough. This is provided by ontology languages like OWL.

Logic layer is needed for expressing rules. We need ways of writing logic into documents to allow the deduction of new knowledge by reasoning on the existing knowledge Inference engine is responsible for performing logical decisions.

In the vision of Tim Berners Lee the production of proofs is not the part of Semantic Web. The reason is that the production of proofs is still a very active area of research and it is by no means possible to make a standardization of this. A Semantic Web engine should only be able to verify proofs.

Without trust, Semantic Web is unthinkable. If individuals and companies receive information but are not sure of its origin or integrity, then there remains nothing else to do with this information but to throw it away. The cryptography is necessary so that everybody can be sure that their communication partners are who they claim to be and what they send really originates from them. This explains the column "Digital Signature" in Fig. 1.

There is also one more layer of Semantic Web, which is not shown in the figure. This is the application layer that makes use of the underlying seven layers. An example might be two companies A and B exchanging information where company A is placing an order with company B.

Knowledge representation on web: The Semantic Web is basically about representing knowledge on the web. The architecture of web, however, poses some challenges for knowledge representation on the web. The main hindrances arise from the facts that web is massive, distributed, dynamic and an open world without any checks on anybody.

Ontologies form the heart of knowledge representation on the web so first we discuss what ontology is and how does it fit into Semantic Web.

Ontology: In Artificial Intelligence domain, ontology is defined as "a formal, explicit specification of a shared conceptualization"[4].

Formal refers to the fact that the ontology should be machine-understandable. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Shared reflects that an ontology should capture consensual knowledge accepted by the communities. Conceptualization refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena.

Purpose of ontology: Ontologies provide a common base to build semantics on. They are considered as a powerful tool to lift ambiguity.

Information retrieval suffers mainly from two problems: Polysemy (one word having different meanings) and synonymy (different words having same meaning). Ontologies can help in resolving both the problems by explicitly defining the context in which a word is used and creating a mapping between synonyms.

If we study human communication, it reveals that we have two tools for disambiguation. First is identification of objects and actions performed on these objects We know that book is an object and reading is an action that is performed on this object. Second is specialization and generalization of the hierarchies of concepts. We know that novel is a book and a book is a document so novel is also a document. We learn this structure of categories through education and socio-cultural interaction but the current information systems relying only on terms and plain text lack this background knowledge. And this is

exactly the purpose of ontologies: to capture the semantics and relations of the notion we use, make them explicit and eventually code them in a symbolic system so that they can be manipulated and exchanged.

Second, ontologies enable knowledge sharing. Suppose we perform an analysis and arrive at a satisfactory set of conceptualizations and their representative terms for some area of knowledge. The resulting ontology will include domain-specific terms, general terms and terms that describe behavior. The ontology captures the intrinsic structure of the domain. In order to build a knowledge representation language based on the analysis, we need to associate terms with the concepts and relation in the ontology and devise a syntax for encoding knowledge in terms of concepts and relations. The other people in this domain can benefit from ontology created by us, thereby eliminating the need for replicating the knowledge analysis process.

Semantic Web languages: Semantic Web languages have been developed in an evolutionary way, layering each new language on the base of the previous languages. This section provides a brief survey of these languages giving advantages and disadvantages of each.

RDF and RDF Schema: Resource Description Framework[5] was developed by World Wide Web Consortium (W3C) to represent information about resources on the World Wide Web It is layered on top of XML. XML gives only rules for how the byte strings should be cobbled together to form a coherent whole that can be used by a widely spread set of computer programs. XML does not say anything about the information itself, only the way it is structured. RDF determines how the information is interpreted. It is a way to express relations between objects, something XML does not allow to do.

RDF, however, provides no mechanism for declaring the properties and defining relationship between properties and other resources. This is the role of RDF Schema that allows users to create schemas of standard classes and properties that have specific semantics.

A significant weakness of RDF is that it does not possess any mechanism for defining general axioms that are used in logic to constrain the possible meaning of a term and thus provide stronger semantics. Axioms can be used to infer additional information that was not explicitly stated and perhaps more importantly for distributed systems such as web, axioms can be used to map between different representations of the same concept.

Another problem with RDF is its poor mechanism of schema revision. Essentially, each new schema is given its own URI and thus can be thought of as a distinct schema in and of itself.

Ontology Inference Layer (OIL): The semantics of OIL are based on description logic but its syntax is layered on RDF[6]. There are three layers of OIL, with each subsequent layer adding functionality to the previous one. Core OIL is basically RDFS without reification (statements about statements), which was omitted because it can be problematic. Instance OIL adds capability to model instances, essentially using RDF to describe the instances. Finally heavy OIL is an undefined layer that will include future extensions to the language. OIL is much more expressive than RDFS and adds standard Boolean operations (oil:AND, oil:OR and oil:NOT), slot constraints (oil:HasValue, oil: ValueType) and cardinality constraints (oil:MaxCaridinality, oil:MinCardinality) etc.

The main advantage of OIL is its rich modeling constructs. OIL weaknesses are inherited from RDF. It has no explicit import mechanism and inadequate support for ontology evolution. OIL cannot specify many of the common kinds of articulations mappings needed to integrate ontologies.

DAML+OIL: DARPA Agent Markup Language+OIL (hereafter referred to as DAML) provides features like data types, expression for enumerations, properties of properties (unique property, inverse property), disjoint classes, restrictions and cardinality constraints[7].

DAML is built upon RDF and has a description logic basis. DAML allows expressions to be a single class, a list of instances that comprise a class, a property restriction, or a Boolean combination of class expressions. It also provides conjunction, disjunction and negation of class expressions. DAML has an explicit feature for including ontologies, providing a means of handling synonymous terms DAML has a daml:equivalent To property that is used to state that two resources are identical.

Since DAML is based on OIL, it has all advantages of OIL. However, since it is also based on RDF, it has many disadvantages of that language.

Web Ontology Language (OWL): OWL[8] is a recommendation by W3C. The language is very close to DAML with some minor changes. OWL provides three increasingly expressive sub-languages–OWL Lite, OWL DL and OWL Full–designed for use by specific communities of implementers and users.

OWL Lite is the simplest of all and is made for users who need only simple classification hierarchy. For example, while it supports cardinality, it only permits cardinality values of 0 and 1.

OWL DL provides all language constructs but they can be used under some restrictions. For example, while a

class may be a subclass of many classes, a class cannot be an instance of another class. Thus it combines maximum expressiveness with computational completeness and decidability making sure that all conclusions will be completed within finite time.

OWL Full lifts all the constraints thus providing maximum expressiveness and syntactic freedom e.g., a class can be treated simultaneously as a collection of individuals and an individual in its own right. But this freedom comes at a cost; computations are not guaranteed to finish in finite time. According to W3C OWL working group, it is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Inference engine: The purpose of Inference engine is to infer new knowledge by performing reasoning on the available knowledge Some experimental inference engines have been developed by different individuals and research groups but they only exist in research laboratories and are unable to perform search on the web due to the fact that only a very limited number of web pages are annotated with metadata required for the engines to perform reasoning.

Closed World Machine (CWM): CWM was created by Tim Berners Lee and Dan Connolly from October 2000 onwards[9]. It is a general-purpose data processor for the Semantic Web It is forward chaining reasoner that can be used for querying, checking, transforming and filtering information. Its core language is RDF, extended to include rules and it uses RDF/XML or RDF/N3 serialization as required. It is written in Python and is far slower than is should be.

The author of CWM has the following opinion about it:

"Deploying CWM on large scale was never on the cards, although lately it appears to be outgrowing its 'play/demonstration code' status".

A CWM Clone has been written by Bijan Parsia, which is actually a translation of CWM in prolog.

Euler: Euler[10] is an inference engine supporting logic based proofs of test cases. It is backward-chaining reasoner enhanced with Euler path detection and tells you whether given set of facts and rules supports a given solution. It is implemented in Java. Another version is implemented in CSharp. Like CWM, Euler was also developed for testing purpose and lacks performance capabilities. The other main drawback is that it does not support any knowledge base and can only perform reasoning on a limited set of rules.

Simple HTML Ontology Extension (SHOE) A semantic search and reasoning engine was developed by University of Maryland in 2001[11]. It is based on ontologies created in SHOE; a language to create ontologies developed by the same university. The engine was implemented in XSB and Parka. XSB is an open source, logic-programming system that can be used as a deductive database engine. Its syntax is similar to prolog. The main problem with XSB is that it is a single-user system which makes it unsuitable for inference on any web based application where a number of users are supposed to access the engine concurrently XSB thus cannot scale to the sized needed for the Semantic Web knowledge bases.

Parka is a high-performance representation system whose roots lie in semantic networks and frame systems. It is capable of performing complex queries over very large knowledge bases in a short time. Parka knowledge base can be updated automatically which makes it suitable for a very dynamic environment like web. The main drawback of Parka is that it does not provide ability to partition its knowledge base, which makes it suitable for system that uses only one ontology. This is impossible in a real life semantic search engine.

Recommendations: After analyzing the existing inference engines, the authors give following recommendations for the future inference engine for Semantic Web.

- The engine should be able to perform reasoning on the knowledge base that is constructed from ontologies in the latest Semantic Web language like DAML or OWL.
- The engine should be implemented, preferably, in logic languages like Prolog for maximum performance.
- Relational Database Management System is the best option for storing knowledge base. They offer high performance and allow dynamic changes to the knowledge base. Moreover, they do not impose a limit for the number of ontologies in knowledge base.

Semantic Web is the web of tomorrow that will not only enable machine-understandability but where all electronics devices will be able to communicate with each other. Semantic Web, however, is suffering from the same chicken-egg problem that the web suffered in its early days. The users do not find easy to use tools and any incentive to annotate their pages with metadata. For large business organizations, the cost of updating their web

sites prevents them from annotating their web sites. The developers are not developing application for Semantic Web because there is no metadata on web. The need of the hour is to motivate the people and organizations to annotate their content on the web with metadata. The duty of developers is two-fold in this task. First they need to build intuitive tools for annotation and then develop tools to benefit from the immense power of Semantic Web.

REFERENCES

1. Lee, T.B., J. Hendler and O. Lassila, 2001. The Semantic Web, Scientific American.
2. Ankolekar A., M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara and H. Zeng, 2001. DAML-S: Semantic Markup for Web Services, in Proceedings of the International Semantic Web Workshop.
3. Lee, T.B., 2000. Semantic Web, Presentation at XML. http://www.w3c.org/2000/Talks/1206-xml2k-tbl/slide10-0.html (WWW document) (accessed 2nd March 2004).
4. Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition, 5: 199–220.
5. Fensel, D., 2000. The Semantic Web and its languages. IEEE Intelligent Systems, pp: 67-73.
6. Fensel, D., F.V. Harmelen, I. Horrocks, D.L. McGuinness and P.F.P. Scheneider, 2001. OIL: An ontology infrastructure for the Semantic Web, IEEE Intelligent Systems, pp: 38-45.
7. McGuinness, D.L., R Fikes, J. Hendler and L.A. Stein, 2002. DAML+OIL: An ontology language for the Semantic Web. IEEE Intelligent Systems, pp: 72-80.
8. McGuinness, D.L. and F. V. Harmelen 2003. OWL Web Ontology Language Overview, W3C Proposed Recommendation. http://www.w3.org/TR/2003/PR-owl-features-20031215/ (WWW document) (accessed 2nd March 2004).
9. Lee, T.B. and D. Connolly, 2001. Closed World Machine. http://www.infomesh.net/2001/cwm/ (WWW document) (accessed 2nd March 2004).
10. Roo, J.D., Euler Proof Mechanism, 2004. http://www.agfa.com/w3c/euler/ (WWW document) (accessed 2nd March 2004)
11. Heflin, J.D., 2001. Towards the Semantic Web: knowledge representation in a dynamic, distributed environment. Ph.D. Thesis, University of Maryland, USA.