ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Semantic Web–A Conceptual Model for Heterogeneous Ontology Environment

Vasavi Bande *
Associate Professor, Computer Science and Engineering,
Hyderabad Institute of Technology and Management,
Hyderabad, (A P), India
vasavi.bande@yahoo.co.in

Keerthi Manchikanti
Assistant.Professor,Information Technology
Hyderabad Institute of Technology and Management,
Hyderabad, A P, India
keerthi_1212@yahoo.com

M.V.N Bhargava
Computer Science and Engineering,
Hyderabad Institute of Technology and Management,
Hyderabad, (A P), India
mvn_bhargava@yahoo.co.in

Abstract: The web has become a vital space to search globe of information, thus it has become a huge and tedious task to handle copious wanted and unwanted information. Semantic web is an emerging as well as welcome resource discovery tool, which enhances the search facilities. The conventional search framework emphasizes on HTML found to be insufficient for representing multifarious data structures. Subsequently XML has developed within w3c and was used for the development of all new formats. XML Schemas are used to describe the properties of data and describes the syntax but not the semantic aspect. RDF is an XML application, which make use of a formal model which is the basis for the Semantic Web and it also uses Ontology to define relationship between fields into a rational and a meaningful association. This paper focuses on how semantic web has evolved with the help of RDF and ontology and also discusses how the drawbacks of existing syntactic web could be eradicated. Semantic web is an extension of the current web in which the information is given well defined meaning and thus enabling computers and people to work in cooperation. This paper describes the importance and conceptual usage of the semantic web services, Web Services that are seen as the technology of choice for implementing service oriented architecture systems, while they also provide state of the art data exchange platform for diverse environments.

Keywords: Extensible markup language (XML), Resource Description Framework (RDF), Web Ontology, Web Ontology Language (OWL), Service Oriented Architecture (SOA).

I. INTRODUCTION

The term Semantic Web barely refers to technology that speeds up response rates, rather it refers to a still emerging body of software tools whose overall goal is to automate the collection and incorporation of information garnered from the websites [1]. The idea is to free the Google/Yahoo user from painfully interactive, highly repetitive keyword searches where we continue to sharpen up our queries until we seem to be finding the exact stuff. Semantic web is a maturing field of technology that continues to be the emphasis of much focused research. This text introduces the standardized knowledge representation and languages for modeling ontology's operating at the core of the semantic web. To support our paper we bring up here about each language, syntax and underlying intuition through examples, with separate treatment of the underlying formal semantics using RDF schema [6], ontology web language (OWL) [12], rules and query languages, such as SPARQL.

In the semantic web data itself becomes part of the web and is able to be processed independent of the application, platform or domain. This is in disparity to the World Wide Web as we know it today, which contains virtually boundless information in the form of documents. We can use computers to search for these documents, but they still have to be read and to be deduced by users before any useful information can be extrapolated. Computers can present you with information but can't understand what the information reveals in a given circumstance. The Semantic Web, on the other hand, is about having data as well as documents on the Web so that machines can process, transform, assemble, and even proceed on the data in useful ways.

Semantic Web technology includes namespaces, which try to put more brilliance in websites by having data tagged with widely shared, standardized set of tags. And things like XML Schema and XQuery can be employed to influence namespace technology to support high-volume, set-oriented queries of data stored on web servers. These are very analogous to the kind of queries that can today be coded in SQL and run on single database servers running database management systems like Oracle, SQL Server, DB2, MySQL and PostgreSQL. Essentially, XML-based technology takes the capability of a relational database schema to help us interpret data, and broadens it to the entire web.

The Semantic Web is interconnection of information linked up in such a way to be easily processable by machines, on a universal scale as shown in figure 1. It is an efficient way of representing data on the World Wide Web or a worldwide linked database. It is a group of methods and technologies to allow machines to understand the meaning or semantic of information on the World Wide Web.

It provides a standard framework that allows data to be integrated and reused across application, enterprise, and societal boundaries. It is a web of data linked up in such a way as to be easily process able by machines, on a global scale. It is about general formats for interchange of data and the language for recording how the data recounts to real world objects. It is a collaborative effort led by the World Wide Web Consortium or W3C with participation from a large number of researchers and industrial partners. The term "Semantic Web" was coined by World Wide Web Consortium (W3C) director Tim Berners-Lee. According to the original vision, the availability of machine-readable metadata would facilitate automated agents and other software to access the Web more shrewdly. The agents would be able to perform tasks automatically and locate the related information for the user.

Data that is generally hidden in the HTML files is often useful in some contexts, but not in others. The problem with the greater part of data on the Web is that it is difficult to use on a large scale, as there is no global system for publishing data in such a way as it can be easily processed.

For example, just think of information about local sports events, weather information, plane times, Major League Baseball statistics, and television guides all of this information is presented by numerous sites, but all in HTML. The problem here is , in some contexts, it is difficult to use this data in the ways that one might want to.

In this paper we attempt to say that it becomes easier to publish data in a purposeful form, so more people will want to publish data and there will be a domino effect. We may find that a large number of Semantic Web applications can be used for a variety of different tasks, increasing the modularity of applications on the Web. But adequate subjective reasoning is required to accomplish this.

The Semantic Web is generally built on syntaxes which use URIs to represent data, usually in triples based structures, i.e. many triples of URI data that can be held in databases or interchanged on the WWW using a set of particular syntaxes developed especially for the task. These syntaxes are called Resource Description Framework (RDF) syntaxes.

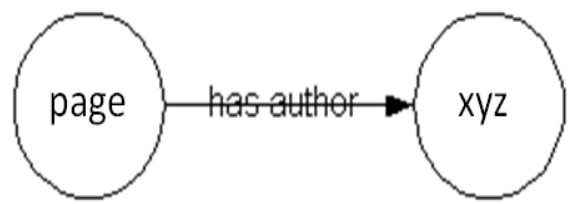
For example, instead of describing digital resources which themselves describe entities of interest such as database records, wiki pages and files, we should focus on describing those entities of interest directly without taking a deviation through describing database entries and other artifacts of the pre-Semantic Web era. We can use the identifier tag: example.org:Eiffel_Tower to refer to the one single Eiffel Tower itself, and not to some abstract conception, description or database entry about it.

The Semantic Web technologies allow us to directly map the structure of reality itself [2]. This seems like a very fine, almost purely philosophical distinction, but it has major practical implications also. RDF/OWL is not only a syntactically more flexible alternative to current database systems but it also enables a whole new philosophy of how information can be organized. If we want to demonstrate all of the advantages of the Semantic Web, we need to be bold enough to break with existing patterns of philosophy. The most widely developed space at the moment within the Semantic Web is in information management, i.e. the organization and discovery of information. This is the primary drive behind the Semantic Web's development, but people are taking a assortment of approaches in developing tools to extend the current Web into a factual Semantic Web. These tools typically take an existing Web element what we are familiar with, such as browsers, servers and search engines, and enhance them with the power to process the semantic annotations associated with web pages. Semantic Web Browsers, for example, extend the notion of the Web browser into the Semantic Web by allowing the RDF annotations of resources to be read and presented in a structured manner.

A. How RDF model is unlike from the XML model

Why should we use RDF and why not just XML. This has been a question which has been around ever since RDF Started. At the W3C Query Language workshop, there was a clear difference of view between those who wanted to query documents and those who wanted to extract the "meaning" in some form and query that. It is a frustrated attempt to explain what the RDF model was for those who though in terms of the XML model. It doesn't try to map one directly onto the other; it expresses the RDF model using XML.

For example a single RDF assertion as shown below. The author of the *page* is "xyz". This is traditional. In RDF this is a triple triple (author, page, xyz) which you can think of as represented by the diagram.



How this information would be typically be represented in XML?

<author>
<uri>page</uri>
<name>xyz</name>
</author>
or maybe
<document href="page">
<author>xyz</author>
</document>
or maybe
<document>
<details>
<uri>href="page"</uri>
<author>
<name>xyz</name>
</author>
</details>
</document>
or maybe
<document>
<author>
<uri>href="page"</uri>
<details>
<name>xyz</name>
</details>
</author>
</document>

B. The XML Graph

These documents sound good in XML and to a person reading then they mean the same thing. To a machine parsing them, they produce different XML trees. Suppose you look at the XML tree:

```
<v>
<x>
<y> a="ppppp"</y>
<z>
<w>qqqqq</w>
</z>
</x>
</v>
```

It's not so obvious what to make of it. The element names were a big hint for a human reader. Without looking at the schema, you know things about the document structure, but nothing else. You can't tell what to deduce. You don't know whether *ppppp* is a *y* of *qqqqq*, or *qqqqq* is a *z* of *ppppp* or what. You can't even really tell what real questions can be asked. The xyz has emerged as a source of confusion and there are lots of questions we *can* ask. They are questions like:

[a] Is there a w element within a detailed element?
[b] What is the content of the w element within the first x element?
[c] What is the content of the w element following the first y element which contains an x element whose attribute is "pppp"?

These are all questions about the *document*. If you know the document schema (a big if) , and if that schema only gives you a limited number of ways of expressing the same thing (another big if) , then asking these questions can be in fact equivalent to asking questions like, who is the author of *page*?

It is possible because there is a mapping from XML documents to semantic graphs [7]. In brief, it is tedious because:

[a] The mapping is many to one.
[b] You need a schema to know about the mapping.
[c] The expression you need for querying something in terms of the XML tree is necessarily more complicated than the expression you need for querying something in terms of the RDF tree.

If we try to write down the expression for the author of a document where the information is in some arbitrary XML schema, you can probably do it though it may or may not be very appealing. If you try to combine more than one property into a combined expression for example, give me a list of books by the same author, for XML it gets too clumsy to consider.

Think of trying to define the addition of numbers by regular expression operations on the strings. It is possible for addition. When you get to multiplication it gets ridiculous to solve the problem you would end up reinventing numbers as a separate type.

Looking at the simple XML encoding above,

```
<author>
<uri>page</uri>
<name>xyz</name>
</author>
```

The complexity of querying the XML tree is because the large number of ways in which the XML maps onto the logical tree, and the query you write has to be independent of the choice of them. So much of the query is an attempt to basically convert the set of all possible representations of a fact into one statement [11]. This is just what RDF does. It gives you more standard ways of writing statements so that however it occurs in a document, they produce the same effect in RDF terms. The same RDF tree results from many XML trees.

It would be nice if we could label our XML so that when the parser read it, it could find the assertions (triples) and distinguish their subjects and objects, so as to just deduce the logical assertions without needing RDFas indicated in figure 1.

C. The RDF Graph

In fact RDF is very flexible, it can represent this triple in many ways in XML [6], so as to be able to fit in with particular applications thus we could write the above as :

```
<Description about="http://www.w3.org/test-tam/page"
Author ="xyz" />
```

In fact as anyone can create or own the verbs, subjects and objects in a distributed Web, any term has to be identified by a URI somehow. This can be depicted with namespaces as follows.. This actual real example works out to in real life more like

```
<?xml version="1.0"?>
<Description
xmlns="http://www.w3.org/TR/WD-rdf-syntax#"
xmlns:s="http://docs.r.us.com/bibliography-info/"
about="http://www.w3.org/test-tam/page"
s:Author =http://www.w3.org/staff-hitam/xyz
/>
```

You can think that the "description" RDF element gives the clue to the parser as to how to find the subjects, objects and verbs in what follows.

This is pretty much the most shorthand way of using the base RDF in XML. There are others which are longer, but more efficient when you have, for instance, sets of many properties of the same object. The useful thing is that of course they all convey the same triple.

It is a mess when you use questions about a document to try to ask questions about what the document is trying to convey. But flagging the grammar explicitly (RDF syntax is a way of doing this) is better.

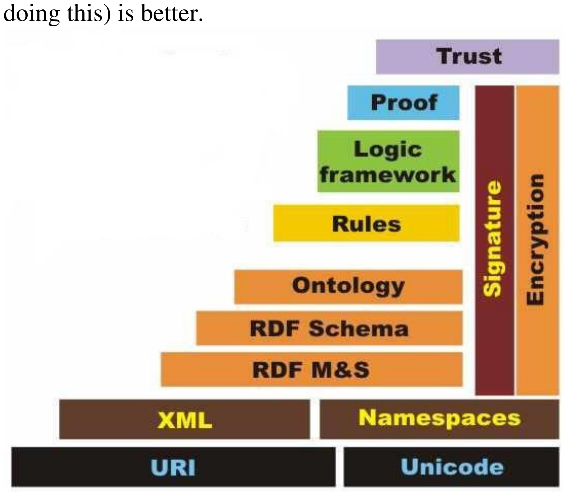


Figure 1. Semantic Web Layer Cake.

Things you can do with RDF which you can't do with XML include are that you can parse the semantic tree, which ends up giving you a set of (possibly mutually referential) triples and then you can use the one what and disregarding the one you don't want.

D. Problems in understanding the structure include

Without having gone to the trouble of getting the schema, or having an application hand-programmed to recognize a particular document type, you can't pick up any semantic information from a document [9]. When an XML schema changes, it could typically introduce new intermediate elements like “details” in the tree above or “div" is HTML. These may or may not invalidate any query which has been based on the structure of the document.

Semantic Web technologies in fundamentally shows novel ways, instead of just adding a thin cover of “semantic layer” on top of traditional software systems. Without hesitation, such semantic layer can make many traditional applications much more flavorsome and efficient, but the potential of Semantic Web technologies is much greater than that. This potential can be realized when we use these technologies as the foundations of our systems. The other most widely used tools on the Web, as far as a user’s experience is concerned, are search engines i.e. with Google today being the most popular. Semantic Web search engines such as Swoogle are under development. Swoogle can use ontologies to purify the search, and has harvested the existing ontologies and RDF [6] data available on the Web. As yet there is a long way to go to make such tools sensitive to the general user, but in the future we can reasonably expect powerful extensions to general search engines.

E. Simple Ontologies in RDF and RDF Schema

The Resource Description Framework (RDF) is a formal language for describing structured information. The goal of RDF is to enable applications to exchange data on the Web while still preserving their original meaning. As opposed to HTML and XML, the main intention now is not to display documents correctly, but rather to allow for further processing and re-combination of the information contained in them. RDF consequently is often viewed as the basic representation format for developing the Semantic Web.

F. Scaling the semantic web

Everyone seemed to agree that manually tagging documents is a fragile exercise. Vladimir Zelevinsky from Endeca had first suggested putting a parser on each machine for efficiency. He said that since you type slower than one sentence per second that at the moment of creation, semantics could be injected into the document. Of course, it is a bit more complex than this, but this was an interesting concept. Kathleen Dahlgren from Cognition said that NLP at scale was the wave of the future. NLP is complex but deeply distributed. Computers are getting faster and cheaper, and this can make it fast and scalable.

Is it practical? There is a huge amount of data out there and it keeps changing. There is also a lot of duplicate information on the web. Is it economically viable to think about parsing the web? Ron Kaplan said he had done a back of the envelope calculation using the following assumptions: “The simple order-of-magnitude calculation goes as follows: There are roughly 2.5M seconds in a month, so an 8-core machine gives you 20M cpu seconds. If it takes one second on the average to process a sentence (an upper bound), then you can do 20M sentences per month. If a web page has on the average 20 sentences, you get 1M pages per month per machine. So, 1000 machines can do a billion pages per month. More if one second over estimates, less if 20 sentence/document underestimates.”

II. SYNTATCTIC WEB APPROACH

Syntactic web is used to describe the current, mostly HTML-based World Wide Web [4], in order to distinguish it from the Semantic Web. Semantic web is a concept in which web pages carry information that can be read and understood by machines in a systematic way. The term is derived from the word “syntax”, which is the mechanics of a language used to convey information, while “semantics” is the actual meaning of that information. On a syntactic web page, any document on the web that does not contain special tagging to communicate meaning, which is difficult to be parsed by a computer program. An example is imagine a site giving the information of weather for any city in the world in HTML form[7]. Even though the site offers dynamic, database-driven information, it is presented in a purely syntactic way. One could imagine a computer program that tried to retrieve this weather information through text parsing or "web scraping”. Though it would be possible to do, if the creators of the site ever decide to change around the layout or HTML of the site, the computer program would most likely need to be rewritten in some way. In contrast, if the weather site published its data semantically, the program could retrieve that semantic data, and the site's creators could change the look and feel of the site without affecting that retrieval ability.

A. Syntactic Clustering of Web: The Web has undergone exponential growth since its birth, and this expansion has generated a number of problems. In this paper we address following issues such as:

The instability of URLs: The basis of our approach is a mechanism for discovering when two documents are "roughly the same" that is, for discovering when they have the same content except for modifications such as formatting, minor corrections, Webmaster signature, or logo. Similarly, we can learn when a document is "roughly contained" in another [3]. Applying this mechanism to the entire collection of documents found by the AltaVista spider yields a grouping of the documents into clusters of closely related items. As explained below, this clustering can help solve the problems of document duplication and URL instability. The duplication problem arises in two ways: First, there are documents that are found in multiple places in a like form. Some examples are FAQ (Frequently Asked Questions) or RFC (Request for Comments) documents, the on-line documentation for popular programs, documents stored in several mirror sites and legal documents [13]. Second, there are documents that are found in almost identical incarnations because they are different versions of the same document, the same document with different formatting [8]. The same document with site-specific links, customizations or contact information shared with other source material to form a larger document, split into smaller documents.

The instability problem arises when a particular URL becomes undesirable because:

A. The associated document is temporarily unavailable or has moved.
B. The URL refers to an old version and the user wants the current version.
C. The URL is slow to access and the user wants an identical or similar document that will retrieve faster. In all these cases, the ability to find documents that are syntactically similar to a given document allows the user

to find other, acceptable versions of the desired item hence introducing duplication.

III. SEMANTIC WEB PROCESS LIFECYCLE

Semantic Web services [5] will allow the semi-automatic and automatic annotation, advertisement discovery; selection, composition, and execution of inter-organization business logic, making the Internet become a global common platform where organizations and individuals communicate among each other to carry out various commercial activities and to provide value-added services [10].

In order to fully tie up the power of web services, their functionality must be combined to create web processes. Web processes allow representing complex interactions among organizations, representing the progress of workflow technology. Semantics can play an vital role in all stages of web process lifecycle. The main stages of the web process lifecycle are illustrated in Figure 2. As mentioned by J. Cardoso and A. Sheth in "Introduction to Semantic Web Services and Web Process Composition, springer, 2005" narrating Semantic Web Process as a powering next generation of processes with Semantics and Web Services.

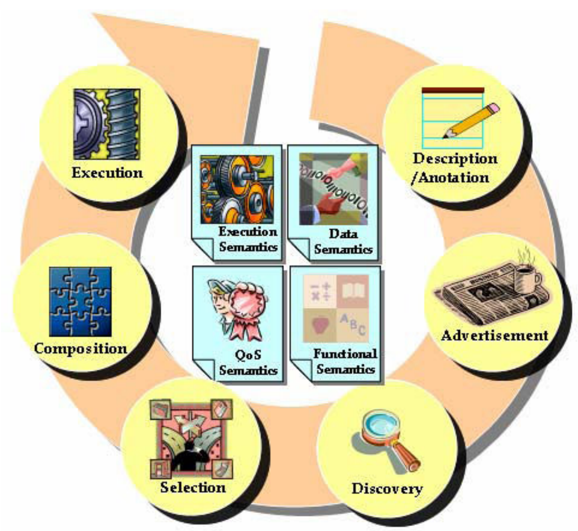


Figure 2. Web process lifecycle and semantics

The lifecycle of semantic web processes includes the description/annotation, the advertisement, the discovery, the selection, the composition of web services that makeup Web processes, and the execution of Web processes. All these stages are significant for the Web process lifecycle and their success.

A. *Semantics and Ontologies*

There is a growing consent that web services alone will not be adequate to develop valuable and complicated Web processes due the degree of heterogeneity, autonomy, and distribution of the web. Several researchers agree that it is essential for web services to be machine understandable in order to allow the full exploitation of efficient solutions supporting all the phases of the lifecycle of web processes.

The idea and vision of the "Semantic Web" catches on and researchers as well as companies have already realized the benefits of this great idea. Ontologies [11] are considered as the basic building block of the Semantic Web as they allow machine supported data interpretation reducing the human involvement in data and process integration.

An ontology "is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of phenomenon in the world by having identified the significant concepts of those phenomena. *Explicit* means that the type of concepts used, and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine readable. Shared reflects that ontology should capture consensual knowledge accepted by the communities" [4].

When the knowledge about a domain is represented in a declarative language, the set of objects that can be represented is called the universe of discourse. We can de-scribe the ontology of a program by defining a set of representational terms. Definitions relate the names of entities in the universe of discourse (e.g. classes, relations, functions or other objects) with human-readable text describing what the names mean and formal axioms that constrain the interpretation and well-formed use of these terms. A set of web services that share the same ontology will be able to communicate about a domain of discourse. We say that a Web service commits to ontology if its observable actions are consistent with the definitions in the ontology.

Example-Benefits of ontologies for the travel industry. The web has permanently changed the manner travel packages can be created as shown in figure 3. Consumers can now acquire packages from a diversity of web sites including online agencies and airlines [12]. With the spread of web travel, a new technology has surfaced for the leisure travel industry i.e. dynamic packaging. For the development of dynamic packaging solutions it is necessary to look in detail at the technology components needed to enhance the online vacation planning experience. By migrating from a third-party service in most markets, dynamic packaging engines can better tailor its package offerings, pricing and merchandising to consumer demand. Currently, the travel industry has concentrated their efforts on developing open specifications messages, based on Extensible Markup Language (XML), to ensure that messages can flow between industry segments as easily as possible. For example, the Open Travel Alliance (OTA) [5] is an organization pioneering the development and use of specifications that support e-business among all segments of the travel industry. The collective effort of various teams, individuals, associations, companies, and international organizations, including air, car, cruise, rail, hotel, travel agencies, tour operators and technology providers, has produced a fairly inclusive set of XML-based specifications for the travel industry (more than 140 XML specification files exist). The current development of open specifications messages based on XML, such as the OTA (Open Travel Alliance) schema, to ensure the interoperability between trading partners and working groups is not sufficiently expressive to guaranty an automatic exchange and processing of information. The development of a suitable ontology for the tourism industry is requisite and will serve as a common language for travel-related terminology and a mechanism for promoting the flawless exchange of information across all travel industry segments.

The development of such ontology can be used to bring together autonomous and heterogeneous web services, web processes, applications, data, and components residing in distributed environments. Semantics allow rich descriptions of Web ser-vices and Web processes that can be used by computers for automatic processing in various tourism related applications. The deployment of ontologies help in articulating a well-defined set of common data elements or vocabulary that can support communication across the multiple channels, speed up the flow of information, and meet travel industry and customer needs. For the travel industry, the simplest form to construct an ontology is to retrieve rich semantic interrelationships from the data and terminology present in the XML-based OTA specifications already implemented and available to organizations. This procedure is illustrated in Figure 3.

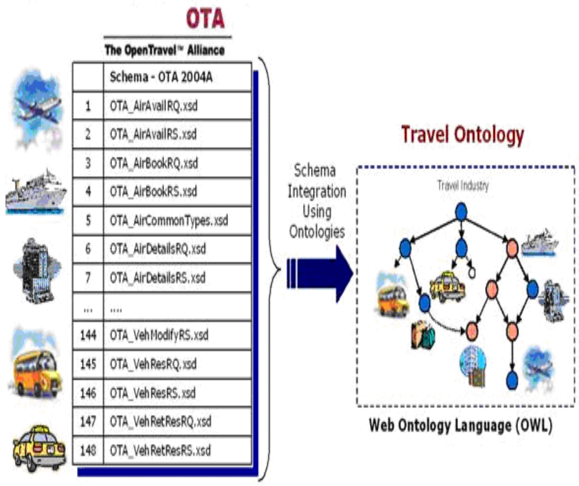


Figure 3. Ontology for Travel Industry

One possible language to construct such an ontology is using the Web Ontology Language (OWL) [6] designed by the World Wide Web Consortium (W3C). The OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of web content by providing additional vocabulary along with a formal semantics. It can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms.

OWL [12] is appropriate to develop an ontology for the travel industry since it is in-tended to be used when the information used by web services needs to be processed by applications, as opposed to situations where in the content only needs to be presented to humans[15].

The development of such an ontology lead to the spearhead and cultivate the cross-industry consensus needed to establish and maintain the most effective and widely used specifications designed to electronically exchange business data and information among all sectors of the travel industry. This effort represents what can be accomplished by the symbiotic synthesis of two of the hottest R&D and technology application areas: Web services and the semantic Web, as recognized at the Thirteenth International World Wide Conference (2004) and in the industry press.

B. Semantics for Web Services

In Web services domain, semantics can be classified into the following types illustrated in Figure 1 and Figure 2:

[a] Functional Semantics.
[b] Data Semantics .
[c] QoS Semantics .
[d] Execution Semantics.

These different types of semantics can be used to represent the capabilities, requirements, effects and execution of a web service. In this section we describe the nature of web services and the need for different kind of semantics for them.

a] Functional Semantics**:** The power of web services can be realized only when appropriate services are discovered based on the functional requirements. It has been assumed in several semantic web service discovery algorithms that the functionality of the services is characterized by their inputs and outputs. Hence, these algorithms look for semantic matching between inputs and outputs of the services and requirements. This kind of semantic matching may not always retrieve an appropriate set of services that satisfy functional requirements. Though semantic matching of inputs and outputs are required, they are not sufficient for discovering relevant services. For example, two services can have the same input/output signature even if they perform entirely different functions. A simple mathematical service that performs addition of two numbers taking the numbers as input and produce the sum as output will have the same semantic signature as that of another service that performs subtraction of two numbers that are provided as input and gives out their difference value as output. Hence, matching the semantics of the service signature may result in huge recall and low accuracy. As a step towards representing the functionality of the service for better discovery and selection, the web services can be annotated with functional semantics. It can be done by having an ontology called Functional Ontology in which each concept/class represents a well-defined functionality [14]. The intended functionality of each service can be represented as annotations using this ontology.

b] Data Semantics**:** All the Web services take a set of inputs and produce a set of out-puts. These are represented in the signature of the operations in a specification file. However, the signature of an operation provides only the syntactic and structural details of the input/output data. These details (like data types, schema of a XML complex type) are used for service invocation. To effectively perform discovery of services, semantics of the input/output data has to be taken into account. Hence, if the data involved in web service operation is annotated using ontology, then the added data semantics can be used in matching the semantics data. Semantic discovery algorithm proposed is used to deal the semantics of the operational data.

c] QoS Semantics: After discovering Web services whose semantics match the semantics of the requirements, the next step is to select the most suitable service. Each ser-vice can have different quality aspect and hence service selection involves locating the service that provides the best quality criteria match. Service selection is also an important activity in web service composition [10]. This demands management of QoS (Quality of Service) metrics for web services. Web services in different domains can have different quality aspects. For organizations, being able to characterize web processes based on QoS has several advantages: a) It allows organizations to transform their vision into their business processes more efficiently, since web processes can be designed according to the QoS metrics. b) It allows the selection and execution of web processes based on their QoS to fulfill customer expectations better. c) It makes possible to monitor the web processes based on QoS, and d) Also allows for the evaluation of alternative strategies when Web process adaptation becomes necessary.

d] Execution Semantics: Execution semantics of a web service encompasses the ideas of message sequence, conversation pattern of web service execution, flow of actions, preconditions and effects of web service invocation etc. Some of these details may not be meant for sharing and some may be, depending on the organization and the application that is exposed as a web service [14]. In any case, the execution semantics of these services are not the same for all services and hence before executing or invoking a service, the execution semantics or requirements of the service should be verified.

Some of the issues and solutions with regard to execution semantics are inherited from traditional workflow technologies. However, the globalization of web services and processes result in additional issues. In e-commerce, using execution semantics can help in dynamically finding partners that will match not only the functional requirements, but also the operational requirements like long running interactions and complex conversations. A proper model for execution semantics will help in coordinating activities in transactions that involve multiple parties.

IV. DIFFERENCES BETWEEN CLASSICAL AND NEW

Table I: Differences between Semantic and Syntactic Web

Semantic Web	Syntactic web
The semantics can cover any ordered or non-structured data and applications, such as Web sites, web services, devices, flow of data, databases.	Meaningful search is not exactly possible.
More useful retrieval with in fewer hits to get more than contextual data.	Retrieval Time would be more.
Can be applied to specify the terminology of heterogeneous systems and semantic integration.	It is possible to retrieve homogeneous system information only.
Could ease the messaging between applications and foster software component reuses and B2B automation.	It is not suitable for B2B environment.
Enabling standardized, searchable and intelligent agents on the Web.	No intelligent agents.

V. APPLICATIONS OF SEMANTIC WEB

A. *Real-Life Applications*

Several real life applications have been implemented during the course of the OTK project to fulfill two requirements to identify the real life requirement for the design of the tools and another way around to secure the usability of the tools for tackling problems to convert the classical date to the RDF formats as shown in Figure 4.

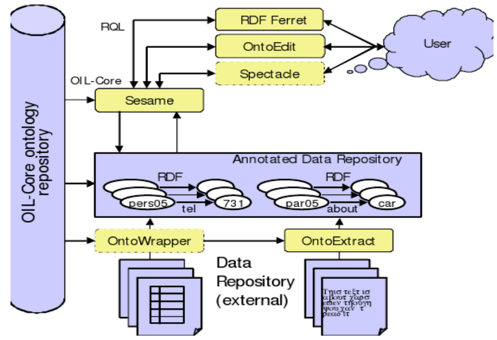


Figure 4. OIL-Semantic Web ConvertToRDF Schema

B. *British Telecom Call Center*

Call centers are the platform for companies to communicate with their customers and the market is growing by 20% each year, with millions being spent on improving customer relationships. Current call center technology lacks the support of the operator in solving incoming requests. The investment in call center technology can offer great rewards including enhanced customer service, lower overheads, lower operational costs and increased staff profitability. In the BT case study, a system for supporting Intranet-based virtual communities of practice is being developed, allowing the automatic sharing of information. The system, *OntoShare*, allows the storage of best practice information in ontology and the automatic dissemination of new best practice information to relevant call center agents. In addition, call center agents can browse or search the ontology to find the information of most relevance to the problem they are dealing with at any given time. The ontology help in orientating new agents and acts as a store for key learning's and best practices accumulated through experience. It provides a sharable structure for the knowledge base and a common language for communication between call center agents.

VI. CONCLUSION

The Semantic Web entails novel tools that can be used in new ways. One important use will be the semantic web portal, letting people to enthusiastically create and use Semantic Web information. Building such an application will need a number of new technologies, we have tried to portray some tools intended to provide this foundation. Thus, the tools illustrated in this paper are the instances of some of the basic technologies that are needed to create this innovative portal knowledge. These include, tools for generating Semantic Web, instances from structured sources (ConvertToRDF), from HTML pages i.e. RDF Screen Scraper, a tool for creating marked up pages (RDF Editor) effortlessly, tool for creating instance data simply, chiefly for non-text sources and a back-end ontology management tool. The intelligent combination of Web services and the semantic Web can start off a technological revolution with the development of semantic Web processes. These technological advances can ultimately lead to a new breed of Web based applications

VII. REFERENCES

[1] Paolucci, M., et al. Importing the Semantic Web in UDDI. in Proceedings Web Services, E-Business and

Semantic Web Workshop, CAiSE 2002. 2002. Toronto, Canada.

[2] Cardoso, J. and A. Sheth, Semantic e-Workflow Composition. Journal of Intelligent Information Systems (JIIS). 2003. 21(3): p. 191-225.

[3] Graham, S., et al., Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI. 2002: SAMS. [4]Berners-Lee, T. and M. Fischetti, Weaving the Web: The Original Design and Ultimate Destiny of the WWW by its Inventor, Harper, San Francisco, 1999.

[4] Berners-Lee, T., Hendler, J. and Lassila, O. “The Semantic Web,” Scientific American, May, 2001

[5] Brickley, D and R.V. Guha, “Resource Description Framework (RDF) Model and Syntax Specification”, W3C Recommendation submitted 22 February 1999, (current May2002).

[6] Sheth, A.P., W.v.d. Aalst, and I.B. Arpinar, Processes Driving the Networked Economy. IEEE Concurrency, 1999. 7(3): p. 18-31.

[7] W3C, W3C Semantic Web Activity. http://www.w3.org/2001/sw/. 2004.

[8] Uschold, M. and M. Gruninger, Ontologies: Principles, methods and applications. Knowledge Engineering Review, 1996. 11(2): p. 93-155.

[9] Introduction to Semantic Web Services and Web Process Composition Jorge Cardoso1 and Amit Sheth[2] Funchal,LSDIS Lab,GA, USA .

[10] Gruber, T.R., toward principles for the design of ontologies used for knowledge sharing. International Journal of Human- Computer Studies, 1995. 43(5-6): p. 907-928.

[11] OpenTravelAlliancewww.opentravel.org, 2004

[12] OWL,Web Ontology Language(OWL). http://www.w3.org/2004/OWL/, 2004, World Wide Web Consortium (W3C).

[13] Cardoso, J. and A.P. Sheth. Semantic Web Processes: Semantics Enabled Annotation, Dis-covery, Composition and Orchestration of Web Scale Processes, in Fourth International Conference on Web Information Systems Engineering (WISE'03). 2003. Roma, Italy.

[14] Sivashanmugam, K., et al., Metadata and Semantics for Web Services and Processes, in Datenbanken and Informationssysteme,Festschrift zum 60.Geburtstag von Gunter Schlageter, W. Benn, et al., Editors. 2003: Hagen, Ger-many. p. 245-271. Connolly, D, van Harmelen, F., Horrocks, I, McGuinness, D., Patel- Schneider, P., and Stein, L. DAML+OIL (March 2001).

[15] Christensen, E., et al., W3C Web Services Description Language(WSDL).http://www.w3c.org/TR/wsdl, 2001

AUTHORS

Vasavi Bande, M.Tech in Computer Science from Jawaharlal Nehru Technological University, Kakinada, A.P., India. She has vast experience in Computer Engineering area pertaining to academics and industry related real time projects. She is presently Associate Professor in Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, A.P, India**.** She has presented many papers to her credit at National and National conferences. She has presided over as judge to many Paper Presentations and Technical Quizzes. She has authored 3 research papers to date and are published in reputed and indexed International Computer Science Journals. She has guided 15 Students of Master degree in Computer Science and Engineering in their major projects. She is bestowed with the Editorial Member on two International Journals Boards and is nominated as Reviewer to three International Journals. Her area of research include Cloud computing, Network Security, Image Processing, Data Mining, Web Technologies and Emerging Technologies. She can be reached at: vasavi.bande@yahoo.co.in

Bhargava Mandapaka, pursuing B.Tech final year from Hyderabad Institute of Technology and Management, gowdawalli, RR dist.**,** Jawaharlal Nehru Technological University , Hyderabad, A.P, India. He has organized national level tech-cultural festivals. He can be reached at: mvn_bhargava@yahoo.co.in

Keerthi.Manchikanti, B.Tech from Jawaharlal Nehru Technological University, Hyderabad, A.P, India. She has worked as Project engineer in WIPRO Technologies and has total experience of 2 years in teaching and currently working for Hyderabad Institute of Technology and Management as Assistant Professor in the department of Information Technology. She has delivered many paper presentations to her credit at national level techno fests. She has received many awards in Academics at school and Graduation level**.** She can be reached at: keerthi_1212@yahoo.com

