SEMANTIC WEB

PRAMEESH P.P
Assistant Professor, Department of Computer Science
SNDPYSS College, Perinthalmanna
(Affiliated to Calicut University)
Kerala

ELDHO PETER
Assistant Professor, Department of Computer Application
Nirmala Arts and Science College, Mulanthuruthy,
(Affiliated to Mahatma Gandhi University)
Ernakulam, Kerala- 682314

SINUMON T G
Assistant Professor, Department of Commerce
Nirmala Arts and Science College, Mulanthuruthy,
(Affiliated to Mahatma Gandhi University)
Ernakulam, Kerala- 682314

Abstract- The Semantic Web is a major research initiative of the World Wide Web Consortium (W3C) to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed (HTML) or syntactically (XML), but also by the meaning of the metadata. The main intent of semantic web is to give machines better access to information resources so that they can be information intermediaries in support of humans. The idea is to build a network of content stored on the web and making it possible for machines to understand data and to satisfy requests from people and other machines. Inorder to carry out their tasks intelligent agents must communicate and understand meaning. The agent based method for semantic analysis enables computers to understand documents written in natural language. To realize the vision of semantic analysis we create markup of web services that makes them machine understandable and use-apparent. Also agent technology is developed that exploits this semantic markup to support automated web service composition and interoperability. Currently, a human must perform all the tasks in the web. With semantic markup of services, we can specify the information necessary for web service discovery as computer interpretable semantic markup at websites, and search engine can automatically locate appropriate services. The major semantic web services are automatic web service discovery, automatic web service execution, and automatic web service composition and interoperability.

Keywords – Semantic web, web services, HTML, XML

INTRODUCTION

The Semantic Web is a major research initiative of the World Wide Web Consortium (W3C) to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed (HTML) or syntactically (XML), but also by the meaning of the metadata. It is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The idea is to build a network of content stored on the web – the Semantic Web – making it possible for machines to understand meaning of data and to satisfy requests from people and machines to use the web content. Semantics is the study of meaning in communication. Semantic Web is a group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web. Its implementation requires adding semantic data. This allows machines process data based on semantic information so that computers can make inferences, understand resource descriptions and relations.

The purpose of semantic web is that Humans depend on web to carry out tasks. But web pages are designed to be read by humans not machines. So, computers cannot accomplish tasks without human direction. Semantic web is a vision of information that is understandable by computers, so computers can perform more of the tedious work involved in finding, combining, and acting upon information on the web. The main intent of the Semantic Web is to give machines much better access to information resources so they can be information intermediaries in support of humans.

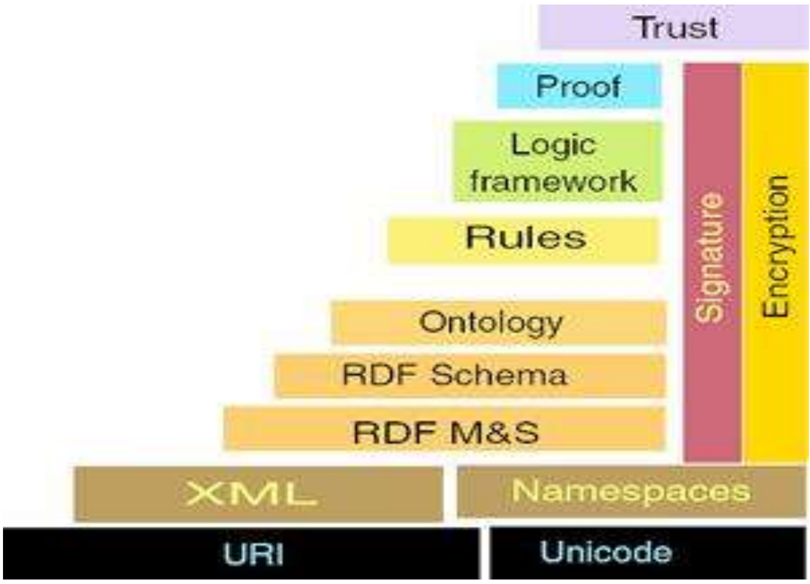
The main idea of the new approach is that a software agent is assigned to each word of the text under consideration. Agents have access to a comprehensive repository of knowledge about possible meanings of words in the text and engage into negotiation with each other until a consensus is reached on meanings of each word and each sentence.

The agent-based method for defining semantics enables computers to understand contents of documents written in a natural language such as English.Possible applications of semantic analysis are numerous and include:

- Written communication between people and computers
- Written communication among computers
- Software translators
- Text referencing engines

- Semantic search engines
- Auto-abstracting engines
- Annotation and classification systems
- Semantic document-flow management systems

SEMANTIC WEB ARCHITECTURE



The Semantic Web layers are arranged following an increasing level of complexity from bottom to top. Higher layers functionality depends on lower ones. This design approach facilitates scalability and encourages using the simpler tools for the purpose at hand. All the layers are detailed in the next subsections.

URI and UNICODE

The two technologies that conform this layer are directly taken from the World Wide Web. URI provides global identifiers and UNICODE is a character-encoding standard that supports international characters. In few words, this layer provides the global perspective, already present in the WWW, for the Semantic Web.

XML and Namespaces

The Semantic Web should smoothly integrate with the Web. Therefore, it must be interwoven with Web documents. HTML is not enough to capture all that is going to be expressible in the Semantic Web. XML is a superset of HTML that can be used the serialisation syntax for the Semantic Web. XML was initially tried but more recently other possibilities have been developed. They are presented and compared in the next section. Namespaces where added to XML to increase its modularisation and the reuse of XML vocabularies in conjunction with XML Schemas. They are also used in the Semantic Web for the same purpose.

XML (eXtensible Markup Language) has paved the road by adding some metadata in the form of human-readable tags that describe data. In addition, XML documents can include information about the author of a Web page, relevant keywords for search engine optimization, and the software tools used to create the XML file, for example.

Before XML, data was stored in flat file and database formats, where most data was proprietary to an application. XML came along and made data interoperable within a single domain, i.e., within the domain defined by a schema or a set of related schemas. By itself, XML provides syntactic interoperability only when both parties know and understand the element names used. If I label an element <price>12.00</price> and someone else labels it <cost>12.00<cost>, there‗s no way for a machine to know that those are the same thing without the aid of a separate, highly customized application to map between the elements. Semantic Web technologies help address this problem by making tags understandable not just to humans – but to machines as well.

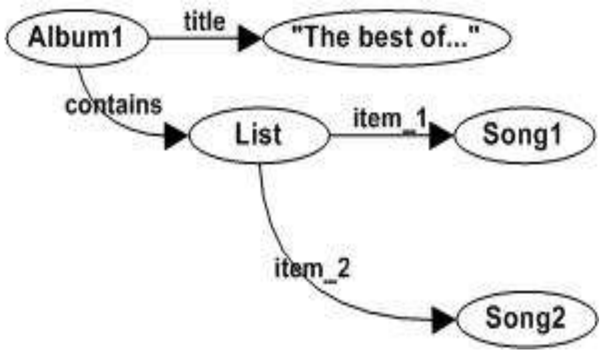
The first step required for machines to understand data is to get that data into a uniform format, where, for instance, a field labeled ―street‖ always has the same format and contains the same type of information, and so on. This type of functionality can be found today on Web sites that use forms that allow users to enter information and run a query, such as airline Web sites that allow visitors to search for and book flights based on a variety of criteria. However, considering the amount and variety of data available from different sources today, this method of data typing does not scale beyond very specific applications.

The next step towards the Semantic Web requires that data from multiple domains is classified based on its properties and its relationship with other data. This is where Semantic

Web technologies such as RDF, RDFS, and OWL come in.

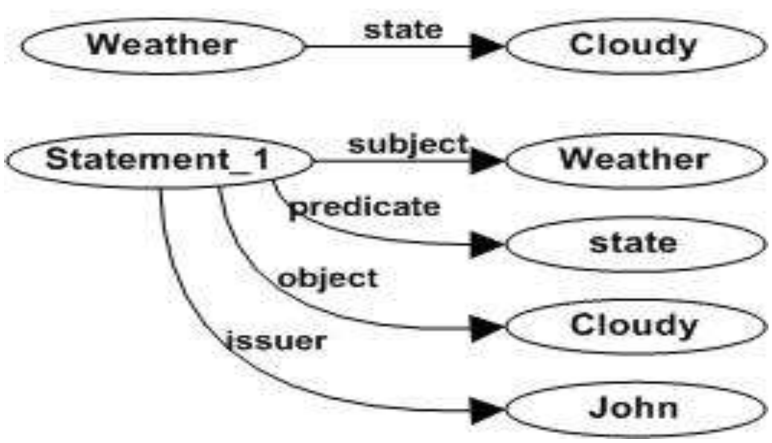
RDF Model and Syntax

The RDF Model and Syntax specification [Becket04] defines the building blocks to realise the Semantic Web. This is the first layer that was specifically developed for it. This specification defines the RDF graph model and the RDF abstract syntax. The RDF graph model defines a structure composed of nodes and directed edges between nodes. The structure of nodes and edges conform directed graphs that model the network of terms and relations between terms of the Semantic Web. The nodes and relations are called resources and are identified by URIs. Each node has its own URI and there are different types of relations that also have an URI, they are called properties. Figure shows and example of RDF graph model.



RDF Graph Model example

Particular edges are identified by the triad composed by the origin node, the property and the destination node. Triads are called triples ore RDF statements and they are the RDF abstract syntax. Graphs can be serialised as a set of triples, one for each edge in the graph. Both representations are equivalent so the graph model can be reconstructed from the set of triples. Triples can also be assigned an explicit identifier, i.e. an URI. This process is called reification. A new node is created that represents the triple and it is associated to three nodes for the three triple components. The origin node is associated using the "subject" property, the property with the "predicate" property and the destination node with the "object" property. Reification is useful to say things about RDF statements. For an example of use, see Figure.



Triple reification example

Abstract triples are the common model to which diverse data structures can be mapped. For instance, relational tables can be translated to a set of triples. Notwithstanding, triples are abstract entities. They are realised for communication using serialisation syntaxes.

The XML syntax has already been introduced in the previous section, it facilitates integrating Semantic Web documents in the current HTML/XML web. The other possibilities are N-Triples and Notation 3 syntax, http://www.w3.org/DesignIssues/Notation3.html. The former is the nearest to the abstract form, a series of triples with subject, predicate and object identified by their URI. The latter uses many syntactic tricks to improve human readability and make serialisations more compact. It is the more human aware syntax and, like XML serialisation, it uses namespaces for modularisation.

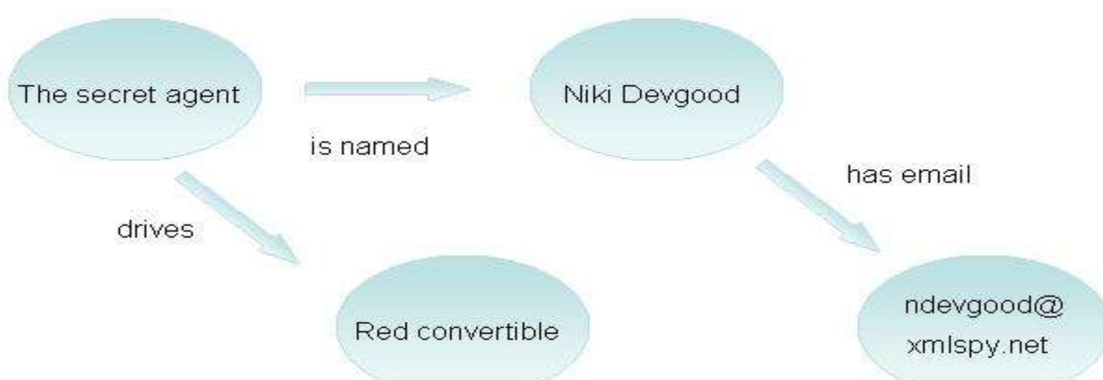
An official W3C recommendation, RDF is an XML-based standard for describing resources that exist on the Web, intranets, and extranets. RDF builds on existing XML and URI (Uniform Resource Identifier) technologies, using a URI to identify every resource, and using URIs to make statements about resources. RDF statements describe a resource

(identified by a URI), the resource‘s properties, and the values of those properties. RDF statements are often referred to as ―triples‖ that consist of a subject, predicate, and object, which correspond to a resource (subject) a property (predicate), and a property value (object). Below is an example of an RDF statement in plain English:

_[resource] __[property] __[value] __
_The secret agent__ Is __Niki Devgood__
_[subject] __[predicate]__[object] __

RDF triples can be written with XML tags, and they are often conceptualized graphically as shown below:

After creating this triple, we can go on to create other triples to associate the agent with an email address, image, etc.



RDF Schema

Simple RDF provides the tools to construct semantic networks. They are a knowledge representation technology presented in the Semantic Networks section. Nonetheless, there is still a lack of many semantic network facilities not available with RDF. There are no defined taxonomical relations. They are defined in the RDF Schema specification [Brickley04]. Taxonomical relations leverage RDF to a knowledge representation language with capabilities similar to semantic networks. This enables taxonomical reasoning about the resources and the properties that relate them.

RDF Schema specification provides some primitives from semantic networks to define metadata vocabularies. RDF Schemas implement metadata vocabularies in a modular way, like XML Schemas. Schema primitives are also similar to Object Orientation constructs they also evolved from the semantic networks tradition. The more relevant ones are:

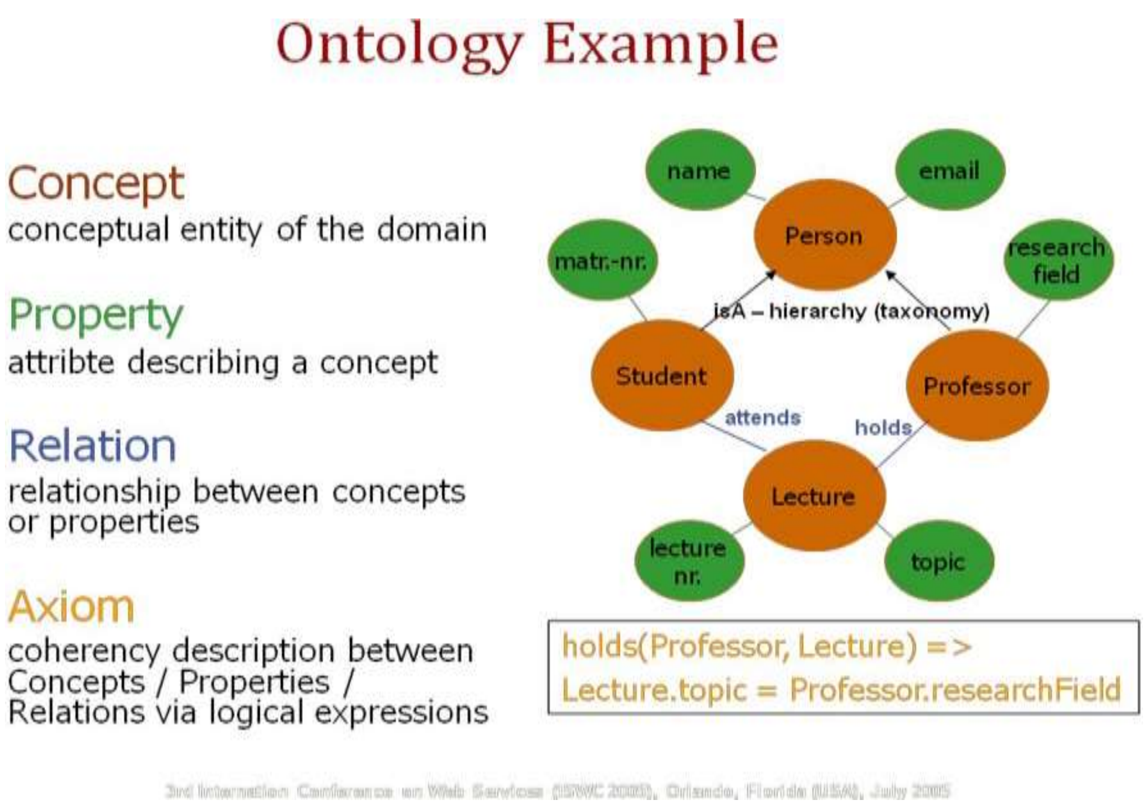
- type: it is a property that relates a resource to a Class to which it pertains. The resource is categorised as a member of this Class and thus it possesses its characteristics.
- Class: it is a set of things that share some characteristics; they have a common conceptual abstraction. A class models the concepts present at the referential semantic level.
- subClassOf: this property holds the taxonomical relations between classes. If class B is a subclass of class A, then class B has all the typical characteristics of class A plus some specific ones that can distinguish it from A.

Ontology

Ontologies are necessary when the expressiveness achieved with semantic network-like tools is not enough. Metadata vocabularies defined by RDF Schemas can be considered simplified ontologies. The tools included in this layer rise the developed vocabularies to the category of ontologies.

Ontologies, which were defined in the Knowledge Representation Ontology section, are specially suited to formalise domain specific knowledge. Once it is formalised, it can be easily interconnected with other formalisations. This facilitates the interoperability among independent communities and thus ontologies are one of the fundamental building blocks of the Semantic Web. Description Logics are particularly suited for ontology creation. They were introduced in the corresponding Knowledge Representation subsection. The World Wide Web Consortium is currently developing a language for web ontologies, OWL. It is based on Description Logics and expressible in RDF so it integrates smoothly in the current Semantic Web initiative.

Description Logic makes possible to develop ontologies that are more expressible than RDF Schemas. Moreover, the particular computational properties of description logics reasoners make possible efficient classification and subsumption inferences.



Rules

The rules layer allows proof without full logic machinery. Similar rules are those used by the production systems presented in the corresponding Knowledge Representation subsection. They capture dynamic knowledge as a set of conditions that must be fulfilled in order to achieve the set of consequences of the rule. The Semantic Web technology for this layer is the Semantic Web Rule Language (SWRL) [Horrocks04]. It is based on a previous initiative called Rule Modelling Language (RuleML) [Boley01]. As RuleML, SWRL covers the entire rule spectrum, from derivation and transformation rules to reaction rules. It can thus specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviours of workflows, services, and agents.

Logic

The purpose of this layer is to provide the features of FOL. First Order Logic was described as the most significant type of logic in the Logic types section. With FOL support, the Semantic Web has all the capabilities of logic available at a reasonable computation cost as shown in the Deduction section. There are some initiatives in this layer. One of the first alternatives was RDFLogic [Berners-Lee03]. It provides some extensions to basic RDF to represent important FOL constructs, for instance the universal (∀) and existential (∃) quantifiers. These extensions are supported by the CWM [Berners-Lee05] inference engine. Another more recent initiative is SWRL FOL [Patel-Schneider04], an extension of the rule language SWRL in order to cope with FOL features.

Proof

The use of inference engines in the Semantic Web makes it open, contrary to computer programs that apply the black-box principle. An inference engine can be asked why it has arrived to a conclusion, i.e. it gives proofs of their conclusions. There is also another important motivation for proofs. Inference engines problems are open questions that may require great or even infinite answer time. This is worse as the reasoning medium moves from simple taxonomical knowledge to full FOL. When possible, this problem can be reduced by providing reasoning engines pre-build demonstrations, proofs, that can be easily checked. Therefore, the idea is to write down the proofs when the problem is faced and it is easier to solve as the reasoning context is more constrained. Further, proofs are used whenever the problem is newly faced as a clue that facilitates reasoning on a wider content. Many inference engines specialised in particular subsets of logic have been presented so far. For instance:

- Prolog for logic programming.
- The production system Jess .
- The FaCT implementation of Description Logics reasoners.
- The CWM inference engine presented in the previous section.

Trust

This is the top layer of the Semantic Web architecture. Agents that want to work with the full-featured Semantic Web will be placed over it. They will conform the Web of Trust. The trust layer makes use of all the Semantic Web layers below. However, they do not provide the required functionality to trustily bind statements with their responsible parts. This is achieved with some additional technologies that are shown in the right part of the Semantic Web stack Figure. The used tools are digital signature and encryption. Thus, the trust web will make intensive use of Public Key Infrastructures. They are already present in the Web, for instance as digital certificates identifying parties that sign digital contracts. Notwithstanding, there is not a widespread use of them. The premise is that Public Key Infrastructure is not of extended use because it is not a decentralised web structure. It is hierarchical and therefore rigid. What the Semantic Web might contribute here is a less constraining substrate of use. The web of trust is based on the graph structure of the Web. Moreover, it supports the dynamic construction of this graph. These features might enable the common use of Public Key Infrastructure in the future Web. To conclude, the final Semantic Web picture contains reasoning engines complemented with digital signatures to construct trust-engines. Then, a Trust Web can be developed with rules about which signed assertions are trusted depending on signer.

SEMANTIC WEB SERVICES

To realize our vision of Semantic Web services we are creating semantic markup of Web services that makes them machine understandable and use-apparent. We are also developing agent technology that exploits this semantic markup to support automated Web service composition and interoperability. Driving\ the development of our markup and agent technology are the automation tasks that semantic markup of Web services will enable —in particular, service discovery, execution, and composition and interoperation.

1 Automatic Web service discovery

It involves automatically locating Web services that provide a particular service and that adhere to requested properties. A user might say, for example, ―Find a service that sells airline tickets between San Francisco and Toronto and that accepts payment by Diner‗s Club credit card.‖ Currently, a human must perform this task, first using a search engine to find a service and then either reading the Web page associated with that service or executing the service to see whether it adheres to the requested properties. With semantic markup of services, we can specify the information necessary for Web service discovery as computer-interpretable semantic markup at the service Web sites, and a service registry or (ontology-enhanced) search engine can automatically locate appropriate services.

2 Automatic Web service execution

It involves a computer program or agent automatically executing an identified Web service. A user could request, ―Buy me an airline ticket from www.acmetravel.com on UAL Flight 1234 from San Francisco to Toronto on 3 March.‖ To execute a particular service on today‗s Web, such as buying an airline ticket, a user generally must go to the Web site offering that service, fill out a form, and click a button to execute the service. Alternately, the user might send an http request directly to the service URL with the appropriate parameters encoded. Either case requires a human to understand what information is required to execute the service and to interpret the information the service returns. Semantic markup of Web services provides a declarative, computer-interpretable API for executing services. The markup tells the agent what input is necessary, what information will be returned, and how to execute—and potentially interact with—the service automatically.

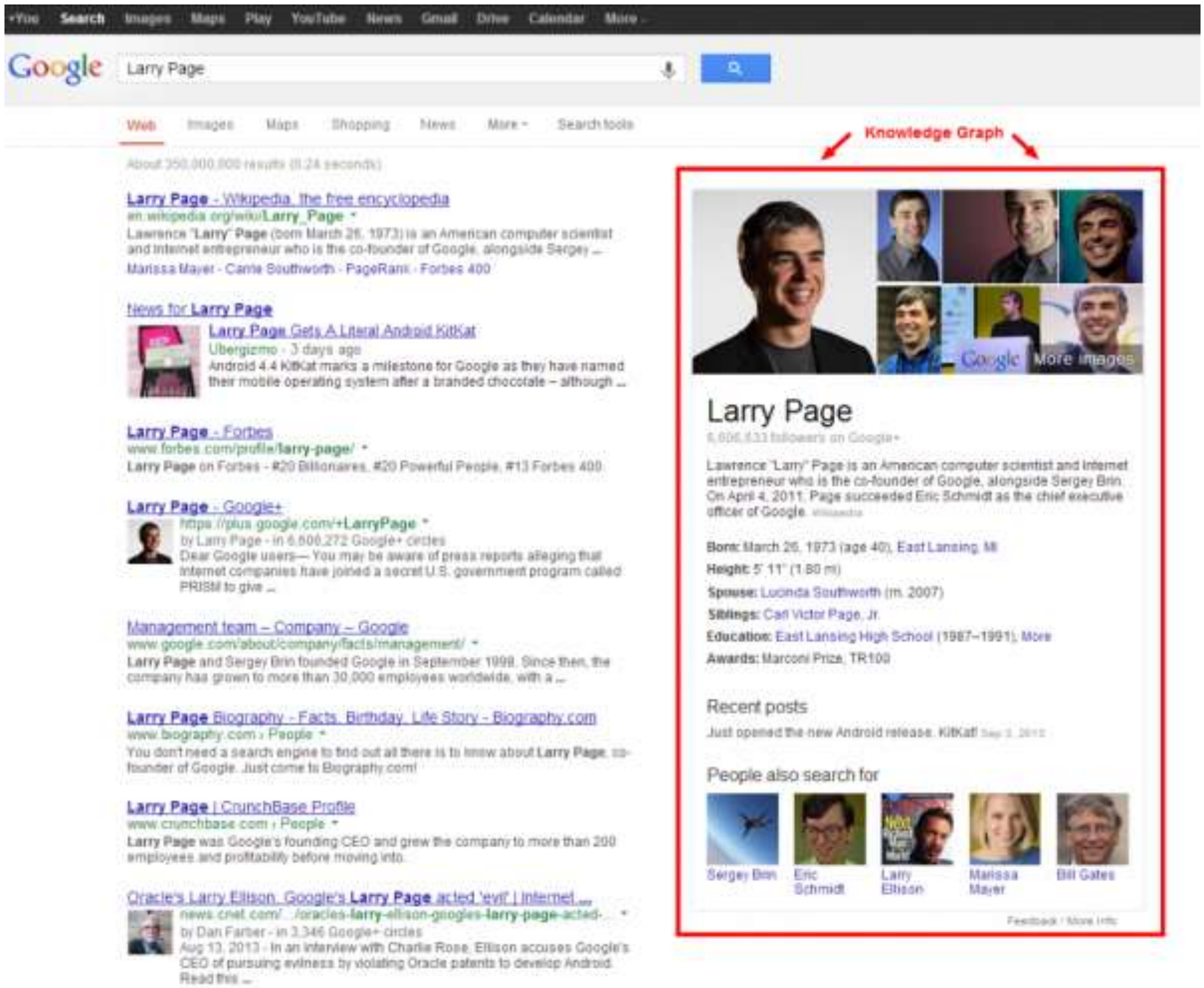
3 Automatic Web service composition and interoperation

It involves the automatic selection, composition, and interoperation of appropriate Web services to perform some task, given a high-level description of the task's objective. A user might say, ―Make the travel arrangements for my IJCAI 2001 conference trip.‖Currently, if some task requires a composition of Web services that must interoperate, then the user must select the Web services, manually specify the composition, ensure that any software for interoperation is custom-created, and provide the input at choice points (for example, selecting a flight from among several options). With semantic markup of Web services, the information necessary to select, compose, and respond to service is encoded at the service Web sites. We can write software to manipulate this markup, together with a specification of the task's objectives, to achieve the task automatically. Service composition and interoperation leverage automatic discovery and execution. Of these three tasks, none is entirely realizable with today's Web, primarily because of a lack of content markup and a suitable markup language. Academic research on Web service discovery is growing out of agent matchmaking research such as the Lark system, 6 which proposes a representation for annotating agent capabilities so that they can be located and brokered. Recent industrial efforts have focused primarily on improving Web service discovery and aspects of service execution through initiatives such as the Universal Description, Discovery, and Integration (UDDI) standard service registry; the XMLbased Web Service Description Language (WSDL), released in September 2000 as a framework-independent Web service description language; and ebXML, an initiative of the United Nations and OASIS to standardize a framework for trading partner interchange.

APPLICATIONS

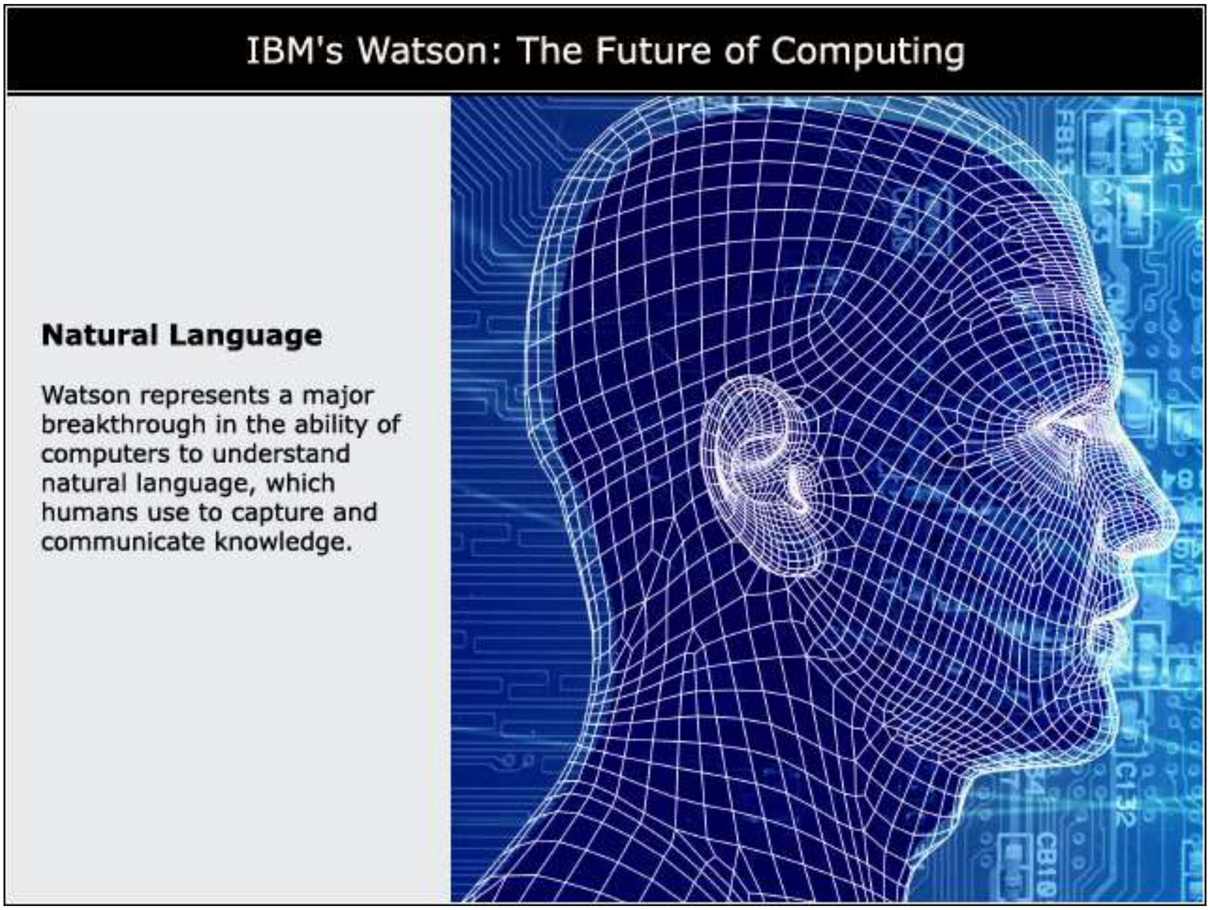
Google Knowledge Graph

The Google Knowledge Graph is a system that **Google** launched in May 2012 that understands facts about people, places and things and how these entities are all connected. **Google** Launches **Knowledge Graph** To Provide Answers, Not Just Links is our introductory article about the system from when it launched.



IBM WATSON

Watson is an IBM supercomputer that combines artificial intelligence (AI) and sophisticated analytical software for optimal performance as a "question answering" machine. The supercomputer is named for IBM's founder, Thomas J. Watson.



APPLE SIRI

Siri (Speech Interpretation and Recognition Interface) /ˈsɪri/ is a part of Apple Inc.'s iOS, watchOS, and tvOS operating systems. Siri is a computer program that works as an intelligent personal assistant and knowledge navigator. The feature uses a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Web services

CONCLUSION

The integration of agent technology and ontologies has made significant impact on the use of web services .This gives the ability to extend programs to more efficiently perform tasks for users with less human intervention. Unifying these research areas and bringing to fruition a web teeming with complex, "intelligent" agents is both possible and practical. Although a number of research challenges still remain. The pieces are coming together, and thus, the semantic web of agents is no longer a science fiction future. It is a practical application on which to focus current efforts.

BIBLIOGRAPHY

[1] Terry R. P., Rahul Singh, Katya Sikara, RCal: ―A Case Study on Semantic Web Agents‖, Carnegie Mellon University

[2] http://www.semanticuniverse.com/articles-agent-based-semantic-web.html accessed on 20th Oct.

[3] http://rhizomik.net/html/~roberto/thesis.html accessed on 16th Oct.

[4]Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng, ―Semantic Web Services‖, Stanford University

[5]http://articles.sitepoint.com/article/semantic-web-services/1 accessed on 16th Oct.

[6] http://semanticweb.com