UNIT-4

SEMANTIC WEB APPLICATIONS AND SERVICES

SEMANTIC WEB APPLICATIONS

Semantic Web applications are those web-based applications that take advantage of semantic content: content that includes not only information, but also metadata, or information about information. The Semantic Web can be used for more effective discovery, automation, integration, and reuse across various applications.

The Semantic Web will provide an infrastructure not just for Web pages, but databases, services, programs, sensors, and personal devices. Software agents can use this information to search, filter, and repackage information. The ontology and logic languages will make information machine readable and power a new generation of tools.

Web technologies can link information easily and seamlessly. The majority of network systems now have Web servers, and the Web interfaces make them seem part of the same world of information. Despite this, transferring content between Web applications is still difficult.

The Semantic Web can address and improve the linking of databases, sharing content between applications, and discovery and combination of Web Services. Under the current Web architecture, linkages between dissimilar systems are provided by costly, tailored software. Again and again, special purpose interfaces must be written to bring data from one systems into another. Applications that run in a given company involve a huge number of ways they can be linked together.

That linking requires a lot of custom code. Use of XML can help, but the problem of effectively exchanging data remains. For every pair of applications someone has to create an ―XML to XML bridge.‖

The problem is that different databases are built using different database schemas, but these schemas are not made explicit.

The use of Resource Description Framework (RDF) in addition to XML can be appropriate when information from two sources need to be merged or interchanged.

The Semantic Web is bringing to the Web a number of capabilities, such as allowing applications to work together in a decentralized system without a human having to custom handcraft every connection.

Some opportunities for Semantic Web applications include Semantic Web Services, Semantic Search, e-Learning, Semantic Web and Bio-Informatics; Semantics- based Enterprise Application and Data Integration, and Knowledge Base.

SEMANTIC WEB SERVICES

Semantic Web Services can bring programs and data together. Just as databases cannot be easily integrated on the current Web without RDF, the same applies to programs.

Consider the case of a company that wishes to purchase parts from a vendor, arrange shipping from a large freight company, and have those parts delivered to one of several manufacturing locations based on which plant has the most capacity at the time of the delivery. Further, they would like this deal to be brokered on the Web with the minimum amount of human interaction. These programs that execute this brokering may be running on special purpose machines and/or behind security and firewall protections.

Web Services are self-contained, self-described, component applications invoked across the Web to perform complex business processes. Once a Web Service is deployed, other applications can discover and invoke the service.

At present, Web Services require human interaction in order to identify and implement.

Tim Berners-Lee has suggested that the integration of Web Services and the Semantic Web could be done in such a way as to combine the business logic of Web Services with the Semantic Web‘s meaningful content.

The vision for Semantic Web Services is to automate the discovery, invocation, composition, and monitoring of Web Services through the use of machine processing. Web sites will be able to use a set of classes and properties by declaring and describing an ontology of services. Web Ontology Language for Services (called OWL-S) has been designed to meet this goal.

SEMANTIC SEARCH

Semantic search methods can augment and improve traditional search results by using, not just words, but concepts and logical relationships. ***There are two approaches to improving search results through semantic methods: (1) the direct use of Semantic Web metadata and (2) Latent Semantic Indexing (LSI).***

The Semantic Web will provide more meaningful metadata about content, through the use of RDF and OWL documents that will help to form the Web into a semantic network. In a semantic network, the meaning of content is better represented and logical connections are formed between related information.

However, most semantic-based search engines suffer increasingly difficult performance problems because of the large and rapidly growing scale of the Web. In order for semantic search to be effective in finding responsive results, the network must contain a great deal of relevant information. At the same time, a large network creates difficulties in processing the many possible paths to a relevant solution.

e-LEARNING

E-learning is the technology to enable people to learn any time any where.E-learning involves the use of a computer (or) any other electronic devices.The big question in the area of educational systems is what is the next step in the evolution of e- learning? Are we finally moving from scattered applications to a coherent collaborative

environment? How close we are to the vision of the Educational Semantic Web and what do we need to do in order to realize it?
On the one hand, we wish to achieve interoperability among educational systems and on the other hand, to have automated, structured, and unified authoring.

The Semantic Web is the key to enabling the interoperability by capitalizing on ***(1) semantic conceptualization and ontologies, (2) common standardized communication syntax, and (3) large-scale integration of educational content and usage.***

The RDF describes objects and their relationships. It allows easy reuse of information for different devices, such as mobile phones and PDAs, and for presentation to people with different capabilities, such as those with cognitive or visual impairments.

By tailored restructuring of information, future systems will be able to deliver content to the end-user in a form applicable to them, taking into account users‘ needs, preferences, and prior knowledge. Much of this work relies on vast online databases and thesauri, such as wordnet, which categorize synonyms into distinct lexical concepts. Developing large multimedia database systems makes materials as useful as possible for distinct user groups, from schoolchildren to university lecturers. Students might, therefore, search databases using a simple term, while a lecturer might use a more scientific term thus reflecting scaling in complexity.

The educational sector can also use the **Internet Relay Chat (IRC) (http://www.irc.org/) a tool** that can be used by the Semantic Web. The IRC is a chat protocol where people can meet on channels and talk to each other.

The IRC and related tools could work well within education, for project discussion, remote working, and collaborative document creation. Video-conferencing at schools is increasingly becoming useful in widening the boundaries for students.

SEMANTIC BIOINFORMATICS

The World Wide Web Consortium recently announced the formation of the Semantic Web Health Care and Life Sciences Interest Group **(HCLSIG)** aimed to help life scientists tap the potential benefits of using Semantic Web technology by developing use cases and applying standard Semantic Web specifications to healthcare and life sciences problems.

The initial foundation and early growth of theWeb was based in great part on its adoption by the high-energy physics community when six high-energy physics Web sites collaborated allowing their participating physicists to interact on this new network of networks. A similar critical mass in life sciences could occur if a half dozen ontologies for drug discovery were to become available on the Semantic Web.

Life science is a particularly suitable field for pioneering the Semantic Web.The biological information to nrepresent in the form of data (or) information that stored digitally in this we are having biology statics and computer science that combined together is known as bioinformatics.in 1960 two dutch scientists is named biological information using computational technology is used and collect anylise and store.

KNOWLEDGE BASE

In a number of parallel efforts, knowledge systems are being developed to provide semantic-based and context-aware systems for the acquisition, organization, processing, sharing and use of

the knowledge embedded in multimedia content.

Ongoing research aims to maximize automation of the complete knowledge lifecycle and to achieve semantic interoperability between Web resources and services.

In one particularly interesting application, **Cycorp** (http://www.cyc.com/) intends to sell products and services using its inference engine, which has been designed to work with the Cyc Knowledge. Cycorp provides a reference Cyc Server executable for Intel-based Linux and for Windows 2000.

OpenCyc is the open source version of the Cyc technology, the world's largest and most complete general knowledge base and common sense reasoning engine. OpenCyc can be used as the basis for a wide variety of intelligent applications, such as speech understanding, database integration and consistency-checking, rapid development of an ontology, and email prioritizing, routing, summarizing, and annotating.

XML-BASED WEB SERVICES

Web Services provide a standard means of interoperating between different software applications running on a variety of platforms. The XML provides the extensibility and language neutrality that is the key for standard-based interoperability of Web Services.

Web Service discovery and composition is led by ***Universal Description Discovery and Integration (UDDI) developed by IBM and Microsoft. Well accepted standards like Web Services Description Language (WSDL) for binding and Simple Object Access Protocol (SOAP)*** for messaging make it possible to dynamically invoke Web services.

Web Service Architecture requires discrete software agents that must work together to implement functionality. In XML-based Web Services, an agent sends and receives messages based upon their architectural roles.

If a requester wishes to make use of a provider's Web Service, he uses a requester agent to exchange messages with the provider agent. In order for this message exchange to be successful, the requester and the provider must first agree on both the semantics and the mechanics of the message exchange.

The message exchange mechanics are documented using WSDL. The service description is a specification that can be processed by a machine using message formats, data types, and protocols that are exchanged between the requester and provider.

CREATING AN OWL-S ONTOLOGY FOR WEB SERVICES

Creating an OWL-S based Ontology for a Web Service requires five steps:

1. Describe individual programs: describe the individual programs that comprise the service. The process model provides a declarative description of a program's properties.

2. Describe the grounding for each atomic process: relate each atomic process to its grounding.

3. Describe compositions of the atomic processes: describe the composite process that is a composition of its atomic processes.

4. Describe a simple process: describe a simple process for the service (optional).

5. Profile description: provide a declarative advertisement for the service. It is partially populated by the process model.

SEMANTIC SEARCH TECHNOLOGY

As Web ontology becomes more advanced, using RDF and OWL tags will offer semantic opportunities for search.

Searching Techniques

Semantic search deals with concepts and logical relationships.

Inference can be viewed as a sequence of logical deductions chained together. At each point along the way, there might be different ways to reach a new deduction. So, in effect, there is a branching set of possibilities for how to reach a correct solution. This branching set can spread out in novel ways.

For example, you might want to try to determine ―Whom does Kevin Bacon know?‖ based on information about his family relationships, his movies, or his business contacts.

It is possible to start at the top of the tree, the root, or with the branches. Taking the top of the tree, the query can be asked, Whom does Kevin Bacon know? Each step down from parent-to-child nodes in this tree can be viewed as one potential logical deduction that moves toward trying to assess the original query using this logical deductive step.

Imagine that each node in this tree represents a statement or fact to prove. Each link from a parent node to a child node represents one logical statement. Now the problem is that we have a big tree of possibilities and this could result in any search being limited to incomplete results.

The Halting Problem is a decision problem that can be informally stated as follows:

Given a description of an algorithm and a description of its initial arguments, determine whether the algorithm, when executed with these arguments, ever halts (the alternative is that it runs forever without halting). Alan Turing proved in 1936 that there is no general method or algorithm that can solve the halting problem for all possible inputs.

The importance of the Halting Problem lies in the fact that it was the first problem to be proved undecidable. Subsequently, many other such problems have been described; The Halting Problem implies that certain algorithms will never end in a definite answer.

You run into incompleteness because the search tree is too large. So our approach must be to search only portions of the tree. There are well-known strategies for how one addresses search problems like this. **One strategy is to search the tree in a depth-first fashion**.

A depth-first search would start at the top of the tree and go as deeply as possible down some path, expanding nodes as you go, until you find a dead end.

A dead end is either a goal (success) or a node where you are unable to produce new children. So the system cannot prove anything beyond that point. Let us walk through a depth-first search and traverse the tree.

Start at the top node and go as deeply as possible:

1. Start at the highest node.

2. Go as deeply as possible down one path.

3. When you run into a dead-end (i.e., a false statement), back-up to the last node that you turned away from. If there is a path there that you have not tried, go down it. Follow this option until you reach a dead-end or a goal (a true statement with no child nodes).

4. If this path leads to another dead-end, go back up a node and try the other branches.

5. This path leads to a goal. In other words, this final node is a positive result to the query. So you have one answer. Keep searching for other answers by going up a couple more nodes and then down a path you have not tried.

6. Continue until you reach more dead-ends and have exhausted search possibilities.

The **advantage of depth-first search** is that it is a very algorithmically efficient way to search trees in one format. It limits the amount of space that you have to keep for remembering the things you have not looked at yet.

Another strategy for searching is a breadth-first search.

Here you search layer by layer. First, you try to do all of the zero-step proofs, and then you try to do all of the one-step proofs, and so on. The advantage of breadth-first search is that you are guaranteed to get the simplest proofs before you get anything that is strictly more complicated.

The **disadvantage of breadth-first search** becomes apparent when you encounter huge deep trees. We also have huge bushy trees where you could have thousands, or tens of thousands, of child nodes.

Another disadvantage of breadth-first searching is the amount of space you have to use to store what you have not examined as yet.

So, if the third layer is explosively large, you would have to store all of the third level results before you could even look at them. With a breadth-first search, the deeper you go into the tree, the more space you will need. So, you find that each of the two traditional algorithms for search, depth-first and breadth-first, are going to run into problems with large systems.

WEB SEARCH AGENTS

While Web search engines are powerful and important to the future of the Web, there is another form of search that is also critical: Web search agents. A Web search agent will not perform like a commercial search engine. Search engines use database lookups from a knowledge base.

In the case of the Web search agent, the Web itself is searched and the computer provides the interface with the user. The agent‘s percepts are documents connected through the Web utilizing HTTP. The agent‘s actions are to determine if its goal of seeking a Web site containing a specified target (e.g., keyword or phrase), has been met and if not, find other locations to visit.

What makes the agent intelligent is its ability to make a rational decision when given a choice. In other words, given a goal, it will make decisions to follow the course of actions that would lead it to that goal in a timely manner.

An agent can usually generate all of the possible outcomes of an event, but then it will need to search through those outcomes to find the desired goal and execute the path (sequence of steps) starting at the initial or current state, to get to the desired goal state.

Building an intelligent Web search agent requires mechanisms for multiple and combinational keyword searches, exclusion handling, and the ability to self-seed when it exhausts a search space.

The search agent needs to know the target (i.e., keyword or phrase), where to start, how many iterations of the target to find how long to look (time constraint), and what methods should determine criteria for choosing paths (search methods). These issues are addressed in the software.

Implementation requires some knowledge of general programming, working with sockets, the HTTP, HTML, sorting, and searches. There are many languages with Web-based utilities, advanced application programming interfaces (APIs), and superior text parsing capabilities that can be used to write a Web search agent. Using a more advanced, efficient sorting algorithm will help improve the performance of the Web search agent.

The Web search agent design consists of four main phases: initialization, perception, action, and effect.

Initialization phase: The Web search agent should set up all variables, structures, and arrays. It should also get the base information it will need to conduct the hunt for the target, the goal, a place to start, and the method of searching.

Perception phase: It is centered on using the knowledge provided to contact a site and retrieve the information from that location. It should identify if the target is present and should identify paths to other Universal Resource Locator (URL) locations.

Action phase: It takes all of the information that the system knows and determines if the goal has been met (the target has been found and the hunt is over).

The Web search agent moves from the initialize phase to a loop consisting of the perception, action, and effect phases until the goal is achieved or cannot be achieved.

SEMANTIC METHODS

There are two approaches to improving search results through semantic methods: (1) LSI and (2) Semantic Web documents.

LATENT SEMANTIC INDEX SEARCH

So far, we have reviewed search technology in general, and identified today‘s search limitations. Now, future technologies based upon the semantics will be explored. First, we will discuss implementing LSI, which may improve today‘s search capabilities without the extreme limitations of searching large semantic networks.

Building on the criteria of precision, ranking, and recall requires more than brute force. Assigning descriptors and classifiers to a text provides an important advantage, by returning relevant documents that do not necessarily contain a verbatim match to our search query. Fully described data sets can also provide an image of the scope and distribution of the document collection as a whole.

A serious drawback to this approach to categorizing data is the problem inherent in any kind of taxonomy: The world sometimes resists categorization.

Latent semantic indexing adds an important step to the document indexing process. In addition to recording which keywords a document contains, the method examines the document collection as a whole, to see which other documents contain some of those same words.

When you search an LSI-indexed database, the search engine looks at similarity values it has calculated for every content word, and returns the documents that it thinks best fit the query. Because two documents may be semantically very close even if they do not share a particular keyword, LSI does not require an exact match to return useful results. Where a plain keyword search will fail if there is no exact match, LSI will often return relevant documents that do not contain the keyword at all.

SEMANTIC WEB DOCUMENTS

A Semantic Web Document is a document in RDF or OWL that is accessible to software agents.

Two kinds of SWDs create Semantic Web ontologies (SWOs) and Semantic Web databases (SWDBs). A document is an SWO when its statements define new classes and properties or by adding new properties. A document is considered as a SWDB when it does not define or extend terms. An SWDB can introduce individuals and make assertions about them.