SEMANTIC WEB AND SOCIAL NETWORKS

IVYearB.Tech. –IISem

SEMANTICWEBANDSOCIALNETWORKS

UNIT–I:WebIntelligence

Thinking and Intelligent Web Applications, The Information Age ,The World Wide Web, Limitations of Today's Web, The Next Generation Web, Machine Intelligence, Artificial Intelligence, Ontology, Inference engines, Software Agents, Berners-Lee www, Semantic Road Map, Logic on the semantic Web.

UNIT-II:KnowledgeRepresentationfortheSemanticWeb

Ontologies and their role in the semantic web, Ontologies Languages for the Semantic Web – Resource Description Framework(RDF) / RDF Schema, Ontology Web Language(OWL), UML, XML/XML Schema.

UNIT-III:OntologyEngineering

Ontology Engineering, Constructing Ontology, Ontology Development Tools, Ontology Methods, Ontology Sharing and Merging, Ontology Libraries and Ontology Mapping, Logic, Rule and Inference Engines.

UNIT-IV:SemanticWebApplications,ServicesandTechnology

SemanticWebapplicationsand services,SemanticSearch,e-learning,Semantic Bioinformatics, Knowledge Base ,XML Based Web Services, Creating an OWL-S Ontology for Web Services, Semantic Search Technology, Web Search Agents and Semantic Methods

UNIT-V:.SocialNetworkAnalysisandsemanticweb

What is social Networks analysis, Development of the social networks analysis, ElectronicSourcesforNetworkAnalysis –ElectronicDiscussionnetworks,Blogs and Online Communities, Web Based Networks. Building Semantic Web Applications with social network features.

TEXTBOOKS:

1. ThinkingontheWeb-BernersLee,GodelandTuring,Wileyinterscience,2008.
2. SocialNetworksandtheSemanticWeb,PeterMika,Springer,2007.

REFERENCEBOOKS:

1. SemanticWebTechnologies,TrendsandResearchinOntologyBased Systems, J.Davies, R.Studer, P.Warren, John Wiley & Sons.
2. SemanticWebandSemanticWebServices-LiyangLuChapman and Hall/CRC Publishers,(Taylor & Francis Group)
3. InformationSharing on the semantic Web – Heiner Stuckenschmidt; FrankVan Harmelen, Springer Publications.
4. ProgrammingtheSemanticWeb,T.Segaran,C.Evans,J.Taylor,O'Reilly,SPD.

CourseObjectives&Outcomes

CourseObjectives

- TolearnWebIntelligence
- TolearnKnowledgeRepresentationfortheSemanticWeb
- TolearnOntologyEngineering
- TolearnSemanticWebApplications,ServicesandTechnology
- TolearnSocialNetworkAnalysisandsemanticweb
- Tounderstandtheroleofontologyandinferenceenginesinsemanticweb
- ToexplaintheanalysisofthesocialWebandthedesignofanewclassof applications that combinehumanintelligencewithmachineprocessing.
- TodescribehowtheSemanticWebprovidesthekeyinaggregating information across heterogeneous sources.
- TounderstandthebenefitsofSemanticWebbyincorporatinguser-generated metadata and other clues left behind by users.

CourseOutcomes

AfterCompletionofthiscourseStudentswillbeableto

- Abilitytounderstandandknowledgerepresentationforthesemanticweb
- Abilitytocreateontology
- Abilitytobuildablogsandsocialnetworks
- UnderstandthebasicsofSemanticWebandSocialNetworks.
- UnderstandElectronicsourcesfornetworkanalysisanddifferentOntologylanguages.
- Modelingandaggregatingsocialnetworkdata.
- Developsocial-semanticapplications.
- EvaluateWeb-basedsocialnetworkandOntology.

INDEX

LectureNotes

UNIT-I

Thinking and Intelligent Web Applications, The Information Age ,The World Wide Web, Limitations of Today's Web, The Next Generation Web, Machine Intelligence, Artificial Intelligence, Ontology, Inference engines, Software Agents, Berners-Lee www, Semantic Road Map, Logic on the semantic Web.

THINKINGANDINTELLIGENTWEBAPPLICATIONS

Themeaningoftheterm―thinking‖mustbeprovidedinthecontextofintelligentapplicationson the World Wide Web as it is frequently loosely defined and ambiguously applied.

In general, thinking can be a complex process that uses concepts, their interrelationships, and inference or deduction, to produce new knowledge. However, thinking is often used to describe such disparate acts as memory recall, arithmetic calculations, creating stories, decision making, puzzle solving, and so on.

A person or an individual is considered as an intelligent if he posses qualities like accurate memory recall, the ability to apply valid and correct logic, and the capability to expand their knowledge through learning and deduction.

The term ―intelligence‖ can be applied to nonhuman entities as we do in the field of Artificial Intelligence (AI). But frequently we mean something somewhat different than in the case of humanintelligence.For example,apersonwhoperformsdifficultarithmeticcalculationsquickly and accurately would be considered as intelligent. But, a computer that could perform the same calculations faster and with greater accuracy would not be considered as an intelligent.

Humanthinkinginvolvescomplicatedinteractionswithinthebiologicalcomponentsofthebrain, and that the process of learning is also an important element of human intelligence.

Softwareapplications perform tasks thataresufficientlycomplexandhuman-likethat the term ―intelligent‖ may be appropriate. But, Artificial Intelligent (AI) is the science of machines simulating intelligent behavior. The concept of intelligent application on the World Wide Web takes the advantages of AI technologies in order to enhance applications and make them to behave in more intelligent ways.

Here, question arises regarding Web intelligence or intelligent software applications on theWorld Wide Web. The World Wide Web can be described as an interconnected network of networks. The present day Web consists not only of the interconnected networks, servers, and clients, but also the multimedia hypertext representation of vast quantities of information distributed over an immense global collection of electronic devices with software services being provided over the Web.

The current Web consists of static data representations that are designed for direct human access and use.

THEINFORMATIONAGE:

We are accustomed to living in a world that is rapidly changing. This is true in all aspects of our society and culture, but is especially true in the field of information technology. It is common to observe such rapid change and comment simply that ―things change.‖

Over the past decades, humanbeings have experienced two global revolutionary changes: the Agricultural Revolution and the Industrial Revolution. Each revolutionary change not only enhanced the access of humanresources but also freed the individuals to achieve higher level cultural and social goals.

In addition, over the past half century, the technological inventions of the Information Age may in fact be of such scope as to represent a third revolutionary change i.e., **the Information Revolution**.

The issue that the ―rapidly changing world of the Information Age be considered a global revolutionarychange on the scale of earlier revolutions‖ can be solved bycomparing the changes associated with the Agricultural Revolution with the Industrial Revolution.

Before the agriculture revolution, human beings move to warmer regions in the winter seasonand back to colder regions in the summer seasons. Human beings were able to migrate to all locations on the earth as they have the flexibility of human species and the capability to create adaptable human cultures.

The adaptable human‘s cultures survived and thrived in every environmental niche on the planet by fishing, herding and foroging.

Human beings lived to stay permanently in a single location as soon as they discovered the possibility of cultivating crops. The major implementation of a non migratory life style is that smallportionofland couldbeexploitedintensivelyforlongperiodsoftime.Anotherimplication is the agricultural communities concentrated the activities into one or two cucleperiodsassociated with growing and harvesting the crops. This new life style allowed individuals to save their resources and spend on their other activities. In additions it created a great focus on the primary necessity of planting, nurturing and harvesting the crops. The individual becomevery conscious of time. A part from these, they become reliant on the following:

1. Specialskillsandknowledge associatedwithagricultural production.
2. Storageandprotectionoffood supplies.
3. Distributionofproductswithin the communitytoensureadequatesubstenance.
4. Sufficientseedforthenearlifecycle‘splanning.Thislifestyleisdifferentfromhunter- gatherer life styles.

The agricultural revolution slowly moved across villages and regions introducing landcultivation as well as a new way of life.

During agricultural revolution human and animal muscle were used to produce the energy required to run the economy. As soon as the French revolution came into existence millions of horses and oxen produced the power required to run the economy.

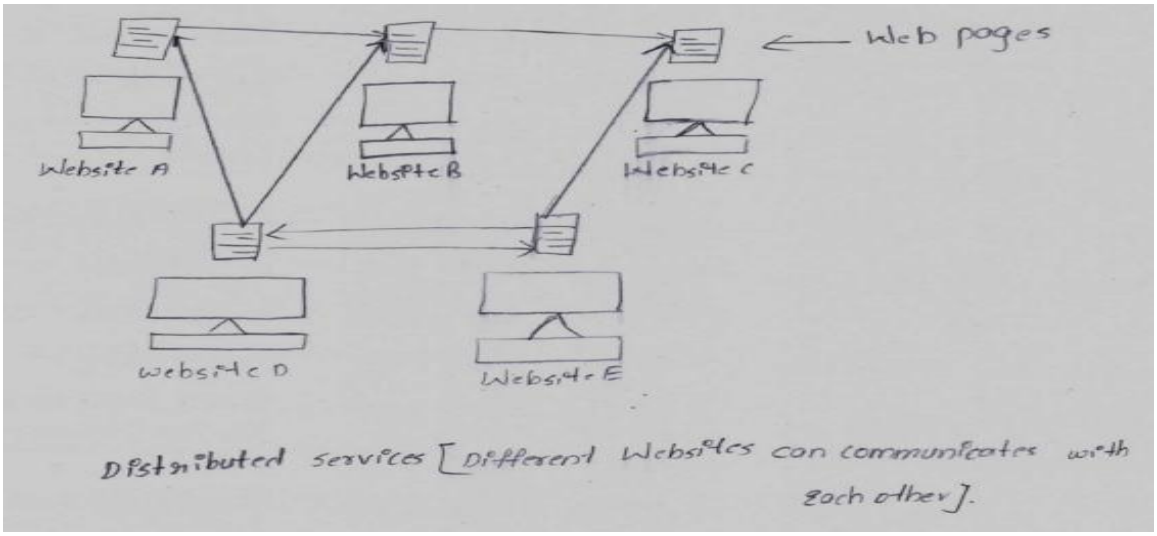
TheWorldWideWeb (WWW):

The World Wide Web (WWW) or the Web is a repository of information spread all over the world and linked to gether. The WWW has a unique combination of flexibility, portability and user friendly features that distinguish if forms other services provided by the internet.

The WWW project was initiated by CERN (European laboratory for particle physics) to create a system to handle distributed resources necessary for Scientific Research. The WWW today is a distributedclient-serverservice,inwhich aclient usingabrowser canbe accessaserviceusinga server. However, the service provided is distributed over many locations called Websites.

The web consists of many web pages that incorporate text, graphics, sound, animation and other multimedia components. These web pages are connected to one another by hypertext. In a hypertext environment the information is stored using the concept of pointers. WWW uses a concept of HTTP which allows communicate between a web browser and web server. The web page can be created byusing a HTML. This language has some commands while which are used to inform the web browser about the way of displaying the text, graphics and multimediafiles. HTML also has some commands through which we can give links to the webpages.

The WWW todayis a distributed client-server, in which a client using a web browser can access a service using a server.



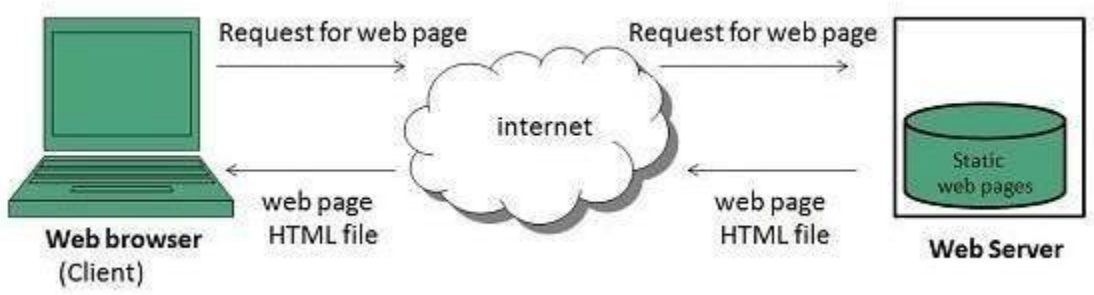
Workingofa web:

Web page is a document available on World Wide Web. Web Pages are stored on web server and can be viewed using a web browser.

WWWworksonclient-serverapproach.Followingstepsexplainshowthewebworks:

1. UserenterstheURL(say,**http://www.mrcet.ac.in**)ofthewebpageintheaddressbarof web browser.
2. ThenbrowserrequeststheDomainNameServerfortheIPaddresscorrespondingto www.mrcet.ac.in.

3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for therequested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser‘s window.



ARPANET

Licklider, a psychologist and computer scientist put out the idea in 1960 of a network of computers connected together by "wide-band communication lines" through which they could share data and information storage.

Licklider was hired as the head of computer research by the Defense Advanced ResearchProjects Agency (DARPA), and his small idea took off.

The first ARPANET link was made on October 29, 1969, between the University of California and the Stanford Research Institute. Only two letters were sent before the system crashed, butthat was all the encouragement the computer researchers needed. The **ARPANET** became ahigh-speed digital postoffice as people used it to collaborate on research projects. It was a distributed system of ―many-to-many‖ connections.

Robert Kahn of DARPA and Vinton Cerf of Stanford University worked together on a solution, and in 1977, the **internet protocol suite** was used to seamlessly link three different networks.

Transmission Control Protocol/Internet Protocol (**TCP/IP**), a suite of network communications protocols used to connect hosts on the Internet was developed to connect separate networks intoa **“network of networks”** (e.g., the Internet). These protocols specified the framework for a few basic services that everyone would need (file transfer, electronic mail, and remote logon) acrossa very large number of client and server systems. Several computers linked in a local networkcan use TCP/IP (along with other protocols) within the local network just as they can use the protocols to provide services throughout the Internet.

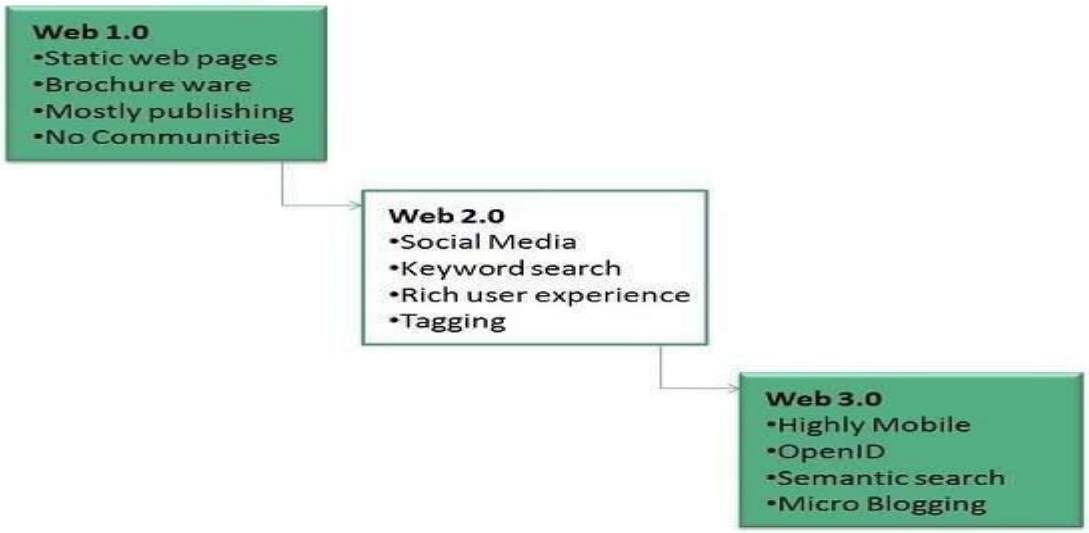
The mid-1980s marked a boom in the personal computer and superminicomputer industries. The combination of inexpensive desktop machines and powerful, network-ready servers allowed many companies to join the Internet for the first time.Corporations began to use the Internet to communicate with each other and with their customers.

By1990,theARPANET wasdecommissioned,leavingonlythevastnetwork-of-networkscalled the Internet with over 300,000 hosts. The stage was set for the final step to move beyond the Internet, as **three major events** and forces converged, accelerating the development of information technology.

These three events were the introduction of the World Wide Web, the widespread availability of the graphical browser, and the unleashing of commercialization.

In1991,**WorldWideWeb** wascreatedby**TimothyBernersLee**in1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

Thefollowingdiagram brieflydefinesevolution ofWorld Wide Web:



The Web combined words, pictures, and sounds on Internet pages and programmers saw the potential forpublishinginformation in awaythat could beas easyas using aword processor,but with the richness of multimedia.

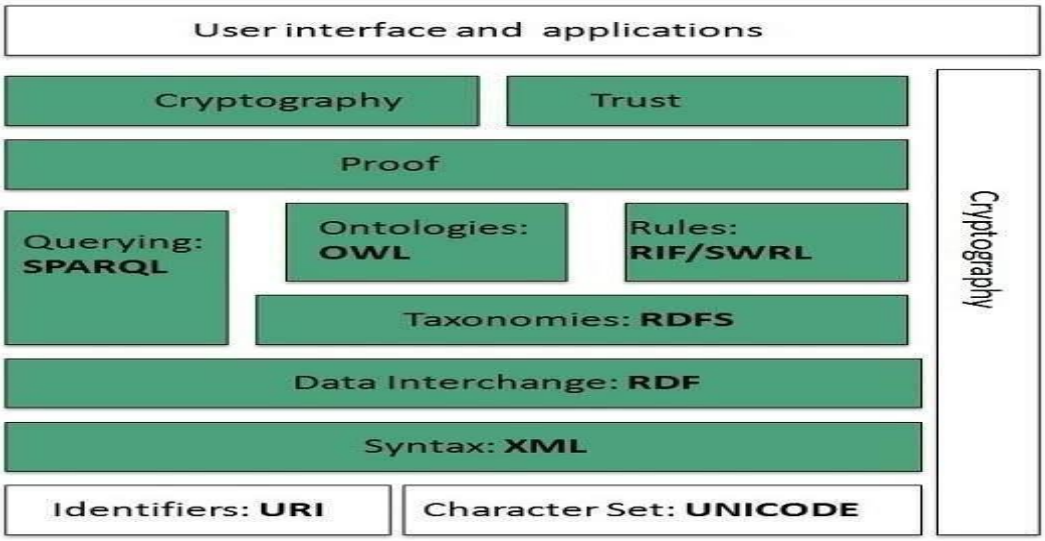
Berners-Lee and his collaborators laid the groundwork for the open standardsof theWeb. Their efforts included the Hypertext Transfer Protocol (HTTP) linking Web documents, the Hypertext Markup Language (HTML) for formatting Web documents, and the Universal Resource Locator (URL) system for addressing Web documents.

The primary language for formatting Web pages is HTML. With HTML the author describes what a page should look like, what types of fonts to use, what color the text should be, where paragraph marks come, and many more aspects of the document. All HTML documents are created by using tags.

In 1993, Marc Andreesen and a group of student programmers at NCSA (the National Center for Supercomputing Applications located on the campus of University of Illinois at Urbana Champaign) developed a graphical browser for the World Wide Web called Mosaic, which he later reinvented commercially as Netscape Navigator.

WWW Architecture

WWWarchitectureisdivided intoseveral layersasshown inthefollowingdiagram:



IDENTIFIERSANDCHARACTERSET

Uniform Resource Identifier (URI)isusedtouniquelyidentifyresourcesontheweb and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

SYNTAX

XML(ExtensibleMarkupLanguage)helpstodefinecommonsyntaxinsemanticweb.

DATAINTERCHANGE

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

TAXONOMIES

RDF Schema (RDFS) allows more standardized description of**taxonomies**and other **ontological** constructs.

ONTOLOGIES

WebOntologyLanguage(OWL)offersmoreconstructsoverRDFS.Itcomesinfollowing three versions:

- OWLLite for taxonomies and simpleconstraints.
- OWLDLfor full description logicsupport.
- OWLformoresyntactic freedomofRDF

RULES

RIF and **SWRL** offersrulesbeyondtheconstructsthatareavailable from **RDFs** and **OWL.** Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

PROOF

Allsemanticandrulesthatareexecutedatlayersbelow Proofandtheirresultwillbeusedto prove deductions.

CRYPTOGRAPHY

Cryptographymeanssuchasdigitalsignaturefor verificationoftheoriginofsourcesisused.

USERINTERFACEAND APPLICATIONS

OnthetopoflayerUserinterfaceandApplicationslayer isbuiltforuser interaction.

LIMITATIONSOFTODAY'S WEB:

1. TheweboftodaystillreliesonHTML,whichisresponsibilityfordescribinghow information is to be displayed and laid out on a web.
2. The web today donot have the ability of machine understanding and processing of web-based information.
3. The web is characterized by textual data augmented web services as it involves human assistance and relies on the intevoperation and inefficient exchange of the two competing propritery server frameworks.
4. Thewebischaracterizedbytextualdataaugmentedbypictorialandaudio-visual addition.
5. The web todau is limited to manual keyboard searches as HTML do not have the abilityto exploit by information retrieval techniques.
6. Webbrowsersarelimitedtoaccessexistinginformationinastandard form.
7. Onweb,developmentof complexnetworkswithmeaningfulcontentisdifficult.
8. Today'swebisrestrictedtosearch,database,support,intelligent,businesslogic, automation, security and trust.

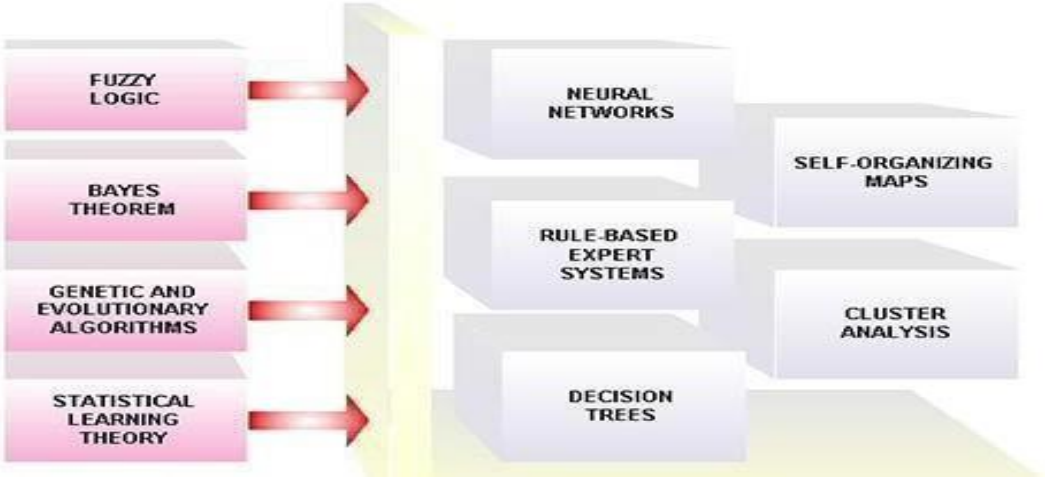
THENEXTGENERATIONWEB

A new Web architecture called the Semantic Web offers users the ability to work on shared knowledge by constructing new meaningful representations on the Web. Semantic Web research has developed from the traditions of AI and ontology languages. It offers automated processing through machine-understandable metadata.

Semantic Web agents could utilize metadata, ontologies, and logic to carry out its tasks. Agents are pieces of software that work autonomously and proactively on the Web to perform certain tasks. In most cases, agents will simply collect and organize information. Agents on theSemantic Web will receive some tasks to perform and seek information from Web resources, while communicating with other Web agents, in order to fulfill its task.

MACHINEINTELLIGENCE(Alsocalledartificialorcomputationalintelligence)

Combinesawidevarietyofadvancedtechnologiestogive **machinestheabilitytolearn,adapt, make decisions, and display behaviors not explicitly programmed into their original capabilities**. Some of machine intelligence capabilities, such as **neural networks, expert systems, and self-organizing maps, are plug-in components** – they learn and manageprocesses at a very high level. Other capabilities, such as fuzzy logic, Bayes Theorem, and genetic algorithms are building blocks – they often provide advanced reasoning and analysis capabilities that are used by other machine reasoning components.
Machine Intelligence capabilities add powerful analytical, self-tuning, self-healing, and adaptive behavior to client applications. They also comprise the core technologies for many of advanced data mining and knowledge discovery services.



ARTIFICIALINTELLIGENCE

Artificial intelligence (**AI**) is the intelligence of machines and the branch of computer science that aims to create it. AI textbooks define the field as "**the study and design of intelligent agents"**where an **intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success**. John McCarthy, who coined the term in 1955, defines it as **"the science and engineering of making intelligent machines."**

Intelligent agent:

Programs, used extensively on the Web, that perform tasks such as **retrieving and delivering information and automating repetition** More than 50 companies are currently developing intelligent agent software or services, including **Firefly** and **WiseWire.**

Agents are designed to make computing easier. Currently they are used as Web browsers, news retrieval mechanisms, and shopping assistants. By specifying certain parameters, agents will "search" the Internet and return the results directly back to your PC.

BranchesofAI

Here'salist, butsomebranches aresurelymissing, because no-onehas identified themyet.

Logical AI

What a program knows about the world in general the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language.

Theprogramdecideswhattodobyinferringthatcertainactionsareappropriateforachievingits goals.

Search
AI programs often examine large numbers of possibilities, e.g. moves in a chess game or inferences byatheorem provingprogram. Discoveries arecontinuallymadeabout howto do this more efficiently in various domains.

Patternrecognition
When a program makes observations of some kind, it is often programmed to compare what it sees with a pattern. For example, a vision program maytryto match a pattern of eyes and a nose in a scene in order to find a face.

Representation
Factsabouttheworldhavetoberepresentedinsomeway.Usuallylanguagesofmathematical logic are used.

Inference
From some facts, others can be inferred. Mathematical logical deduction is adequate for some purposes, but new methods of *non-monotonic* inference have been added to logic since the1970s. The simplest kind of non-monotonic reasoning is default reasoning in which a conclusion is to be inferred by default, but the conclusion can be withdrawn if there is evidence to the contrary

Commonsenseknowledgeandreasoning
This is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s. While there has been considerable progress, e.g. in developing systems of *non-monotonic reasoning* and theories of action, yet more new ideas are needed.

Learningfromexperience
Programs do that. Programs can only learn what facts or behaviors their formalisms can represent, and unfortunately learning systems are almost all based on very limited abilities to represent information.

Planning
Planning programs start with general facts about the world (especially facts about the effects of actions), facts about the particular situation and a statement of a goal. From these, they generatea strategy for achieving the goal. In the most common cases, the strategy is just a sequence of actions.

Epistemology
This is a studyof thekinds of knowledge that are required for solvingproblems in theworld.

Ontology
Ontology is the study of the kinds of things that exist. In AI, the programs and sentences deal with various kinds of objects, and we study what these kinds are and what their basic properties are. Emphasis on ontology begins in the 1990s.

Geneticprogramming
GeneticprogrammingisatechniqueforgettingprogramstosolveataskbymatingrandomLisp programs and selecting fittest in millions of generations.

ApplicationsofAI

Gameplaying
You can buy machines that can play master level chess for a few hundred dollars. There is some AIinthem,buttheyplaywellagainstpeoplemainlythroughbruteforcecomputation--lookingat hundreds of thousands of positions. To beat a world champion bybrute force and known reliable heuristics requires being able to look at 200 million positions per second.

Speechrecognition
In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and citynames. It is quiteconvenient. On the theother hand, while it is possible to instruct some computers using speech, most users have gone back to thekeyboard and the mouse as still more convenient.

Understandingnatural language
Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

Computervision
The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are onlylimited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

Expert systems
A "knowledge engineer" interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results.

One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expertsystems depends on their users having common sense.

Heuristicclassification
Oneofthemostfeasiblekindsofexpertsystemgiventhepresent knowledgeofAIistoputsome information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and aboutthe establishment from which he is buyingit (e.g., about whether there have been previous credit card frauds at this establishment).

ONTOLOGY

Ontologies are considered one of the pillars of the *Semantic Web*, although they do not have a universally accepted definition. A (Semantic Web) **vocabulary** can be considered as a special form of (usually light-weight) ontology.

"Ontologyisaformalspecificationofashared conceptualization"

Inthecontextofcomputer&informationsciences,ontologydefinesasetofrepresentational primitiveswithwhichtomodeladomainofknowledgeordiscourse.Therepresentational primitives are typically classes (or sets), attributes (or properties), and relationships (or relationsamong class members).

The definitions of the representational primitives include information about their meaning andconstraints on their logically consistent application. In the context of database systems, ontologycan be viewed as a level of abstraction of data models, analogous to hierarchical and relationalmodels, but intended for modeling knowledge about individuals, their attributes, and theirrelationships to other individuals.

Ontologies are typically specified in languages that allow abstraction away from data structuresand implementation strategies;

In practice, the languages of ontologies are closer in expressive power to first-order logic thanlanguages used to model databases.For this reason, ontologies are said to be at the ―semantic‖level, whereas database schema are models of data at the ―logical‖ or ―physical‖ level.Due totheir independence from lower level data models, ontologies are used for integratingheterogeneous databases, enabling interoperability among disparate systems, and specifyinginterfaces to independent, knowledge-based services.

In the technology stack of the Semantic Web standards, ontologies are called out as an explicitlayer. There are now standard languages and a variety of commercial and open source tools forcreating and working with ontologies.

- Ontologydefines(specifies)theconcepts,relationships,andotherdistinctionsthatarerelevant for modeling a domain.
- The specification takes the form of the definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use.

KEY APPLICATIONS

Ontologies are part of the W3C standards stack for the Semantic Web, in which they are used to specify standard conceptual vocabularies in which to exchange data among systems, provide services for answering queries, publish reusable knowledge bases, and offer services to facilitate interoperability across multiple, heterogeneous systems and databases.

The key role of ontologies with respect to database systems is to specify a data modeling representation at a level of abstraction above specific database designs (logical or physical), so that data can be exported, translated, queried, and unified across independently developed systems and services. Successful applications to date include database interoperability, cross database search, and the integration of web services.

INFERENCEENGINE

Inferencemeans **Aconclusion reachedonthebasisofevidenceandreasoning**.

In computer science, and specifically the branches of knowledge engineering and artificial intelligence, an **inference engine** is a **"computer programthat tries to derive answers froma knowledge base"**. It is the ***"brain"***that expert systems use to reason about the information inthe knowledge base for the ultimate purpose of formulating new conclusions. Inference engines are considered to be a special case of reasoning engines, which can use more general methods of reasoning.

Architecture

Theseparationofinferenceenginesasadistinctsoftwarecomponentstemsfromthetypical production system architecture. This architecture relies on a data store,

1. An**interpreter**.Theinterpreterexecutesthechosenagendaitemsbyapplyingthecorresponding base rules.
2. A**scheduler**.Theschedulermaintainscontrolovertheagendabyestimatingtheeffectsof applying inference rules in light of item priorities or other criteria on the agenda.
3. A**consistencyenforcer.**Theconsistencyenforcerattemptstomaintainaconsistent representation of the emerging solution.

Logic:

In logic, a **rule of inference**, **inference rule**, or **transformation rule** is the act of drawing a conclusion based on the form of premises interpreted as a function which takes premises, analyses their syntax, and returns a conclusion (or conclusions). For example, the rule of inference modus ponens takes two premises, one in the form of "If p then q" and another in the form of "p" and returns the conclusion "q". Popular rules of inference include modus ponens, modus tollens from propositional logic and contraposition.

Expert System

In artificial intelligence, an **expert system** is a **computer system that emulates the decision-making ability of a human expert**. Expert systems are designed to solve complex problems by reasoning about knowledge, like an expert, and not by following the procedure of a developer as is the case in conventional programming.

SOFTWAREAGENT

In computer science, a **software agent** is a **software program that acts for a user or other programin a relationship of agency**, which derives from the Latin *agere* (to do): an agreement to act on one's behalf.

Thebasic attributes ofa software agent are that

- Agentsare**notstrictlyinvokedforatask,butactivatethemselves,**
- Agentsmay**resideinwaitstatus onahost, perceivingcontext,**
- Agentsmay**get to runstatusona hostuponstarting conditions,**
- Agentsdonot**requireinteractionof user,**
- Agentsmay**invokeothertasks includingcommunication.**

Variousauthorshaveproposeddifferentdefinitionsofagents;thesecommonlyincludeconcepts such as

- ✓ ***Persistence***(codeisnotexecutedondemandbutrunscontinuouslyanddecidesforitself when it should perform some activity)
- ✓ ***Autonomy***(agentshavecapabilitiesoftaskselection,prioritization,goal-directedbehavior, decision-making without human intervention)
- ✓ ***Socialability***(agentsareabletoengageothercomponentsthroughsomesortof communication and coordination, they may collaborate on a task)
- ✓ ***Reactivity***(agentsperceive thecontext inwhich theyoperateandreact toitappropriately).

Distinguishingagentsfromprograms
Related and derived concepts include ***Intelligent agents*** (in particular exhibiting some aspect of Artificial Intelligence, such as learning and reasoning), ***autonomous agents*** (capable of modifyingthewayin whichtheyachievetheirobjectives), ***distributedagents*** (beingexecutedon physically distinct computers), ***multi-agent systems*** (distributed agents that do not have the capabilities to achieve an objective alone and thus must communicate), and ***mobile agents*** (agents that can relocate their execution onto different processors).

Examplesof intelligentsoftware agents
Haag(2006) suggests thattherearеonlyfouressential typesof intelligent software agents:

1. Buyeragentsorshoppingbots
2. Userorpersonal agents
3. Monitoring-and-surveillanceagents
4. DataMining agents

Buyeragents(shoppingbots)

Buyeragents travel aroundanetwork(i.e.the internet)retrievinginformation aboutgoods and services. These agents, also known as 'shopping bots', work very efficiently for commodity products such as CDs, books, electronic components, and other one-size-fits-all products.

Useragents(personalagents)
User agents, or personal agents, are **intelligent agents that take action on your behalf**. In this category belong those intelligent agents that already perform, or will shortly perform, the following tasks:

- ✓ ***Checkyoure-mail***,sortitaccordingtotheuser'sorderofpreference,andalertyouwhen important emails arrive.
- ✓ ***Playcomputergames*** asyouropponent orpatrolgameareasforyou.
- ✓ ***Assemblecustomizednewsreports***for you.
- ✓ ***Findinformation*** foryouonthesubjectofyourchoice.
- ✓ ***Filloutforms***ontheWebautomaticallyforyou,storingyourinformationforfuture reference
- ✓ ***ScanWebpages***lookingforandhighlightingtextthatconstitutesthe "important"partofthe information there
- ✓ ***"Discuss"topics***withyourangingfromyourdeepestfearstosports
- ✓ ***Facilitatewithonlinejobsearchduties*** byscanningknownjobboardsandsendingthe resume to opportunities who meet the desired criteria.
- ✓ ***Profilesynchronization***acrossheterogeneoussocial networks

Monitoring-and-surveillance(predictive)agents

Monitoring and Surveillance Agents are **used to observe and report on equipment, usually computer systems.** The agents may keep track of company inventory levels, observe competitors' prices and relay them back to the company, watch stock manipulation by insider trading and rumors, etc.
For example, **NASA's Jet Propulsion Laboratory** has an agent that monitors inventory, planning, and scheduling equipment ordering to keep costs down, as well as food storage facilities. These agents usually monitor complex computer networks that can keep track of the configuration of each computer connected to the network.

Dataminingagents

This agent uses **information technology to find trends and patterns in an abundance of information from many different sources**. The user can sort through this information in order to find whatever information they are seeking.

A data mining agent operates in a data warehouse discovering information. A 'data warehouse' brings together information from lots of different sources. "Data mining" is the process of looking through the data warehouse to find information that you can use to take action, such as ways to increase sales or keep customers who are considering defecting.

'**Classification**'is one of the most common types of data mining, which finds patterns in information and categorizes them into different classes.

TIMBERNERS-LEEWWW:

When Tim Berners-Lee was developing the key elements of the World Wide Web, he showed great insight in providing Hypertext Markup Language (HTML) as a simple easy-to-use Web development language.

The continuing evolution of the Web into a resource with intelligent features, however, presents many new challenges. The solution of the World Wide Web Consortium (W3C) is to provide a new Web architecture that uses additional layers of markup languages that can directly apply logic. However, the addition of ontologies, logic, and rule systems for markup languages means consideration of extremely difficult mathematic and logic consequences, such as paradox, recursion, undecidability, and computational complexity on a global scale.

The impact of adding formal logic to Web architecture and present the new markup languages leading to the future Web architecture: the Semantic Web. It concludes with a presentation of complexity theory and rulebased inference engines followed by a discussion of what is solvable on the Web.

THEWORLD WIDEWEB
By 1991, three major events converged to accelerate the development of the Information Revolution. These three events were the introduction of the World Wide Web, the widespread availability of the graphical browser, and the unleashing of commercialization on the Internet. The essential power of the World Wide Web turned out to be its universality though the use of **HTML**.Theconceptprovidedtheabilitytocombinewords,pictures,andsounds(i.e.,toprovide multimedia content) on Internet pages.

This excited many computer programmers who saw the potential for publishing information on the Internet with the ease of using a word processor, but with the richness of multimedia forms.

Berners-Lee and his collaborators laid the groundwork for the open standards of the Web. Their efforts included inventing and refining the **Hypertext Transfer Protocol (HTTP)** for linking Web documents, the **HTML** for formatting Web documents and the **Universal Resource Locator (URL)** system for addressing Web documents.

TIMBERNERS-LEE

Tim Berners-Lee was born in London, England, in 1955. His parents were computer scientists whometwhileworkingontheFerrantiMark I,theworld'sfirstcommerciallysoldcomputer.He soon developed his parents' interest in computers, and at Queen's College, at Oxford University, he built his first computer from an old television set and a leftover processor.

Berners-Lee studied physics at Oxford, graduated in 1976. Between 1976and 1980, he workedat Plessey Telecommunications Ltd. followed by D. G. Nash Ltd. In 1980, he was a software engineer at CERN, the ***European Particle Physics Laboratory, in Geneva, Switzerland*** wherehe learned the laboratory's complicated information system. He wrote a computer program to store information and use random associations that he called, ***"Enquire-Within-Upon- Everything," or "Enquire."*** This system provided links between documents.

In 1989, Berners-Lee with a team of colleagues developed ***HTML***, an easy-to-learn document coding system that allows users to click onto a link in a document's text and connect to another document. He also created an addressing plan that allowed each Web page to have a specific location known as a ***URL***. Finally, he completed ***HTTP*** a system for linking these documents across the Internet. He also wrote the software for the first server and the first Web clientbrowser that would allow any computer user to view and navigate Web pages, as well as create and post their own Web documents.

In the following years, Berners-Lee improved the specifications of URLs, HTTP, and HTML as the technology spread across the Internet.

HyperText Markup Language is the primarylanguage for formatting Web pages. The author of a web page uses HTML to describe the attributes of the documents such as,

- whattheweb pageshould looklike
- whattypesoffontstouse
- whatcolortext shouldbe
- whereparagraphsbegin

HypertextTransferProtocol

HyperText Transfer Protocol is the network protocol used to deliver files and data on the Web including: HTML files, image files, query results, or anything else. Usually, HTTP takes place through TCP/IP sockets. Socket is the term for the package of subroutines that provide an access path for data transfer through the network.

HTTP uses the client–server model: An HTTP client opens a connection and sends a request message to an HTTP server; the server then returns a response message, usually containing the resource that was requested. After delivering the response, the server closes the connection.

The result of an implementation of XMLis referred to as SOAP. ***Simple Object Access Protocol (SOAP)*** is an implementation of XML that represents one common set of rules about how data and commands are represented and extended.

Itconsistsofthree parts:

1. **Anenvelope**(aframework fordescribingwhatisina messageand howto processit)
2. **Setofencodingrules** (forexpressinginstancesofapplication-defineddatatypes)
3. **Aconvention** (Itisused foridentifyingremoteprocedurecallsandresponses.)

SEMANTICROADMAP:

Tim Berners - Lee, and his World Wide Web constortium (W3C) team are working collaboratively to develop, extend, and standardize the Web‘s markup languages and tools. In addition, what they are designing is the next generation Web architecture: the Semantic Web.

The goal of the Semantic Web architecture is to provide a knowledge representation of linked data in order to allow machine processingon a global scale. This involves movingthe Web from a repository of data without logic to a level where it is possible to express logic through knowledgerepresentation systems.

The vision of the Semantic Web is to expand or increased the existing Web with resources more easily interpreted by programs and intelligent agents.

Theexistingwebinvolves *two methods* togaininformationregardingdocuments.

Thefirstis tousea directory,orportal site
The directory is constructed manually by searching the Web and then categorizing pages and links. The problem with this approach is that directories take a tremendous effort to maintain. Finding new links, updating old ones, and maintaining the database technology, all add to a portal‘s administrative burden and operating costs.

ThesecondmethodusesautomaticWebcrawlingandindexingsystems.

The future semantic web approaches can produce effective results by using a system that combines the reasoning engine as well as search engine. It will be able to reach out to indexes that contain verycompletelists ofall occurrences ofa giventerm, and then uselogicto weed out all the terms of items that can be used to solve a given problem.

Hence, if the Semantic Web can produce such a structure and meaningful content to the Web, then an environment is created where software agents can perform sophisticated tasks for users.

LogiconthesemanticWeb

The goal ofthe Semantic Web is different from most systems oflogic.Th eSemanticWeb‘s goal is to create a unifying system where a subset is constrained to provide the tractability and efficiency necessary for real applications. However, the Semantic Web itself does not actually define a reasoning engine, but rather follows a proof of a theorem.

This mimics an important comparison between conventional hypertext systems and the original Web design. The original Web design dropped link consistency in favor of expressive flexibility and scalability. The result allowed individual Web sites to have a strict hierarchical order or matrix structure, but it did not require it of the Web as a whole.

Asaresult,aSemantic Web wouldactuallybea proof validatorratherthanatheorem prover.

In other words, the Semantic Web cannot find answers, it cannot even check that an answer is correct, but it can follow a simple explanation that an answer is correct. The Semantic Web as a source of data would permit many kinds of automated reasoning systems to function, but itwould not be a reasoning system itself.

The objective of the Semantic Web therefore, is to provide a framework that expresses both data and rules for reasoning for Web-based knowledge representation. Adding logic to the Webmeans using rules to make inferences, choose courses of action, and answering questions. A combination of mathematical and engineering issues complicates this task. The logic must be powerful enough to describe complex properties of objects, but not so powerful that agents can be tricked by being asked to consider a paradox.

ThelogicoftheSemanticWeb is proceedingin astep-by-stepapproach buildingonelayeron top of another. Three important technologies for developing the Semantic Web are,

1)ResourceDescription Framework2)Ontology 3)WebOntology Language

1. ResourceDescription Framework

Resource Description Framework is a model of statements made about resources and associated URI. Its statements have a uniform structure of three parts: subject, predicate, and object.

Using RDF, the statements can be formulated in a structured manner. This allows softwareagents to read as well as act on such statements. The set of statements can be expressed as a graph; a series of (subject, predicate, object) triples, or even in XML forms.

- Thefirstformisthe mostconvenient forcommunicationbetweenpeople
- Thesecondforefficientprocessing
- Thethirdoneallowsasflexiblecommunicationwithagentsoftware.

2. Ontology

Ontologyisanagreementbetweensoftwareagentsthatexchangeinformation.Thus,therequired information is obtained by such an agreement in order to interpret the structure as well as understand the exchanged data and a vocabulary that is used in the exchanges.
Usingontology,agentscanexchangenewinformationcanbeinferredbyapplyingandextending the logical rules present in the ontology.

An ontology that is complex enough to the useful for complex exchanges of information will suffer from the possibility of logical inconsistencies. This is considered as a basic consequenceof the insights of Godel‘s incompleteness theorem.

3. WebOntologyLanguage[OWL]

This language is a vocabulary extension of RDF and is currently evolving into the semantic markup language for publishing and sharing ontologies on the World Wide Web. Web Ontology Language facilitates greater machine readability of Web content than that supported by XML, RDF, and RDFS by providing additional vocabulary along with formal semantics.

OWLcan be expressed in three sublanguages: OWL Lite, OWLDL,andOWLFull.

UNIT-II

KnowledgeRepresentationfortheSemanticWeb
Ontologies and their role in the semantic web, Ontologies Languages for the Semantic Web –Resource Description Framework(RDF) / RDF Schema, Ontology Web Language(OWL), UML, XML/XML Schema.

ONTOLOGIESANDTHEIRROLEINTHESEMANTICWEB

OntologybasedKnowledgeRepresentation:

The ontology and ontology languages may be viewed as one of the important technology in semantic web. Ontology can be described as formal allotment of conceptualization band to domain. This implies it provides the description of concepts and their corresponding relationships. Inparticular,theontologiesaredesigned as domainmodelsthatbroadlyappeaseor satisfy two special characteristics signifying the semantics. They are,

1. Ontologiesareexpressed informallanguageswithawell-definedsemantics.

The first point underlines that ontology needs to be modelled using languages with a formal semantics such languages include RDF and OWL. These languages are treated as mostfrequently used languages in semantic web. These languages contain those models which are prefferred by term ontology.

2. Ontologiesbuild uponashared understandingwithin a*community*.

This understanding represents an agreement among members of the community over theconcepts and relationships that are present in a domain and their usage.

RDF and OWL, the ontology languages, have standardized syntaxes and logic-based formal semantics. RDF and OWL are the languages most commonly used on the Semantic Web, and in fact when using theterm ontologymanypractitionersrefer to domain models described in one of these two languages. The second point reminds as that there is no such thing as a ―personal ontology‖. For example, the schema of a database or a UML class diagram that we have created for the design of our own application is not an ontology.It is a conceptual model of a domain, but it is not shared: there is no commitment toward this schema from anyone else but us.

The simplest structures are **glossaries** or controlled vocabularies, in essence an agreement on the meaning of a set of terms.

Semanticnetworks areessentiallygraphsthatshow alsohowtermsarerelatedtoeach other**.**

Thesauri are richer structures in that they describe a hierarchy between concepts and typically also allow describing related terms and aliases. Thesauri are also the simplest structures where logic-based reasoning can be applied: the broadernarrower relationships of these hierarchies are transitive, in that an item that belongs to anarrower categoryalso belongs to its direct parent and all of its ancestors.

Folksonomy structures are regarded as weaker models that do not contain any explicit hierarchies, often compries extra corresponding to the social context of tags, i.e. the set of users who have been using them. These structures again success in extracting hierarchies and also relationship among the tags.

The term ***lightweight ontology*** is typically applied to ontologies that make a distinction between classes, instances and properties, but contain minimal descriptions of them.

On the other hand, ***heavyweight ontologies*** allow to describe more precisely how classes are composed of other classes, and provide a richer set of constructs to constrain how properties can be applied to classes. At the far end of the spectrum are complex knowledge bases that use the full expressivity of **first order logic (FOL)** to define to a great detail the kind of instances a concept may have and in which cases two instances may be related using a certain relationship. The more constrained the descriptions of concepts are, the less likely that their meaning will be misinterpreted by human readers.

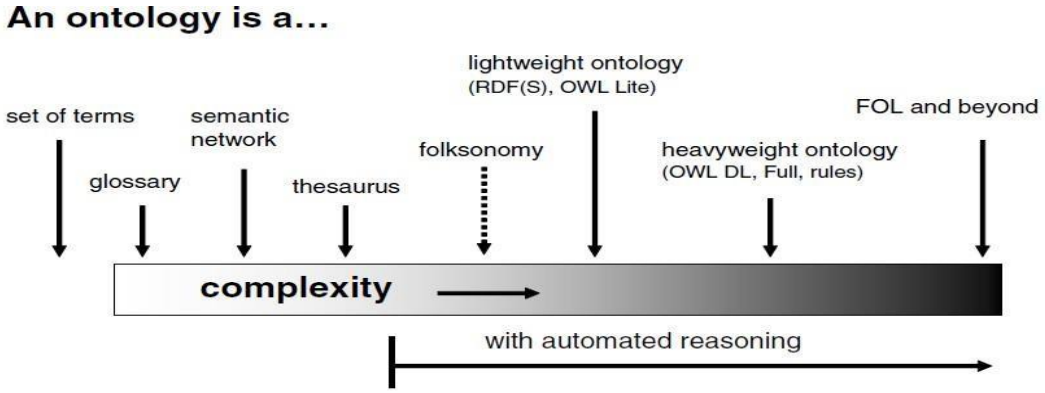


Figure 4.1. Ontologies can be organized according to complexity (informally, the level of semantics).

In practice, the most commonWeb ontologies are all lightweight ontologies due to the need of serving the needs of many applications with divergent goals. Widely shared Web ontologies also tend to be small as they contain only the terms that are agreed on by a broad user base. Large, heavyweight ontologies are more commonly found in targeted expert systems used in focused domains with a tradition of formalized processes and vocabularies such as the area of life sciences and engineering.

OntologiesandontologylanguagesfortheSemanticWeb:

Althoughthenotionofontologiesisindependent oftheWeb,ontologiesplayaspecial roleinthe architecture of the Semantic Web.

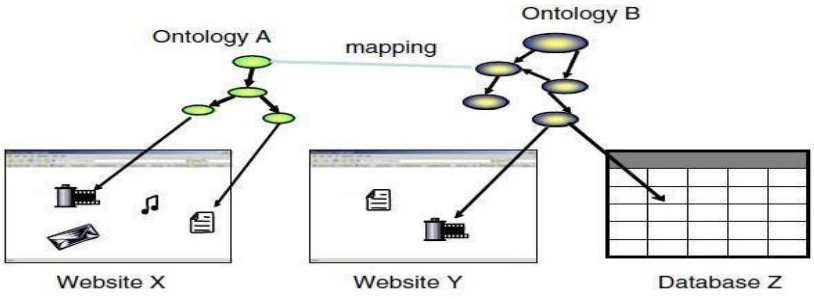


Figure 4.2. The Semantic Web in envisioned as a network of ontologies that adds machine-processable semantics to existing web content, including Web resources and databases.

This architecture provides the main motivation for the design of ontology languages for the SemanticWeb: RDF and OWLare both prepared for the distributed and open environment of the Web.

The Semantic Web will be realized by annotating existing Web resources with ontology-based metadata and by exposing the content of databases by publishing the data and the schema in one of the standard ontology languages.

Ontology languages designed for the SemanticWeb provide themeans to identify concepts in ontologies using globally unique identifiers (URIs). These identifiers can be used in data sources to point toaconcept orrelationship from an external, publicontology. Similarto creatingHTML pages and linking them to existing ones, anyone can create and publish an ontology, which may reference the concepts in remote ontologies. Much like the hyperlinks among web pages, it is expected that these references will form a contiguousweb linking all knowledge sources across theWeb.

As URLs are also URIs, ontologies can also reference existing Web resources and describe their characteristics.

Semantic Web applications collect or query such data sources, aggregate and reason with the results. Information described according to a single schema that is known by the application developer can be directly consumed: by committing to ontology the parties involved havealready implicitly agreed on the meaning of the information.

When annotating Web content or exposing the content of a database, one may choose to create ontology from scratch, or reuse an existing ontology while possibly extending it. However, as there is no coordination of any kind in reusing ontologies, it may happen that two communities develop ontologies that cover the same or overlapping domains.

OntologiesLanguagesfortheSemantic Web:

WeintroducetheontologylanguagesRDFandOWL,whichhavebeenstandardizedinrecent years by the World Wide Web Consortium.

TheResourceDescriptionFramework(RDF)/RDFSchema:

The Resource Description Framework (RDF) was originally created to describe resources on the World Wide Web. In reality, RDF is domain-independent and can be used to model both real world objects and information resources. RDFitself is a veryprimitive modelinglanguage, but it is the basis of more complex languages such as OWL.

Thereare twokindsofprimitivesinRDF:*resourcesand literals*

The definition of a resource is intentionally vague; in general everything is modelled as a resource that can be (potentially) identified and described. Resources are either identified by a URI or left blank. Blank resources (*blank nodes*) are the existential quantifiers of the language: they are resources with an identity, but whose identifier is not known or irrelevant.

Literalsarestrings (characterliterals) withoptionallanguageanddatatypeidentifiers.

Expressions are formed bymaking statements (*triples*) of the form (subject, predicate, andobject). The subject of a statement must be a resource (blank or with a URI), the predicate must be a URI and the object can be either kind of resource or a literal. Literals are thus only allowed at the end of a statement.

The eXtensible Markup Language (XML) is a universal meta-language for defining markup. It provides a uniform framework for exchanging data between applications. It builds upon the originalandmostbasiclayeroftheWeb,HypertextMarkupLanguage(HTML).However,XML does not provide a mechanism to deal with the semantics (the meaning) of data.

Resource Description Framework (RDF) was developed by the World Wide Web Consortium (W3C) for Web-based metadata in order to build and extend XML. The goal of RDF is to make work easier for autonomous agents and automated services by supplying a rudimentary semantic capability.

The RDF is a format for data that uses a simple relational model that allows structured and semistructured data to be mixed, exported, and shared across different applications. It is a data model for objects and relationships between them and is constructed with an object-attribute-valuetriplecalledastatement.WhileXMLprovidesinteroperabilitywithinoneapplication(e.g., producingandexchangingbankstatements)usingagivenschema,RDFprovidesinteroperability *across* applications (e.g., importing bank statements into a tax calculating program).

HTMLLANGUAGE

In 1990, when Tim Berners-Lee laid the foundation for the World Wide Web, he included three primary components: HTTP (Hypertext Transfer Protocol), URLs (Universal ResourceLocators), and HTML (Hypertext Markup Language).

These three components represented the essential ingredients leading to the explosive growth of the World Wide Web. The original idea behind HTML was a modest one. Browsers, such as Internet Explorer or Netscape Navigator, could view information on Web pages written in HTML. The HTML program can be written to a simple text file that is recognized by a browser application and can also be called embedded script programming.

The following listing of HTML markup tags is a HTML ―Hello World‖ example consisting of root tags (<HTML>), head tags (<HEAD>), and body tags (<BODY>) with the displayed information wedged in between the appropriate tags:

```
<HTML>
<HEAD>
<TITLE>MyTitle</TITLE>
</HEAD>
<BODY>
Hello World
</BODY>
</HTML>
```

In particular, Web applications, such as Web Services, required a means to explicitlymanipulate data. This motivated the development of XML.

XML LANGUAGE

The HTML program is not extensible. That is, it has specifically designed tags that require universalagreementbeforechangescanbemade.Althoughovertheyears,Microsoftwasableto addtagsthatworkonlyin InternetExplorer,and Netscape wasabletoadd tagsthatworkonlyin Navigator, Web site developers had no way of adding their own tags. The solution was XML. Proposed in late 1996 by the W3C, it offered developers a way to identify and manipulate their own structured data.

TheXMLdocument simplified the processof definingand usingmetadata.

XML is not a replacement, but rather a complementary technology to HTML. While XML is already widely used across the Web today, it is still a relatively new technology. The XML is a meta language, which means it is a language used to create other languages. It can provide abasic structure and set of rules for developing other markup languages.

TheXMLdocumentletsyounamethetagsanythingyouwant,unlikeHTML,which limits you to predefined tag names. You can choose element names that make sense in the context of the document. Tag names are case-sensitive, although either case may be used as long as the opening and closing tag names are consistent.

The text between the tags is the content of the document, raw information that may be the body of a message, a title, or a field of data. In its simplest form, an XML document is comprised of one or more named elements organized into a nested hierarchy. An element consists of an opening tag, some data, and a closing tag. For any given element, the name of the opening tag must match that of the closing tag. A closing tag is identical to an opening tag except that the less-than symbol (<) is immediately followed by a forward-slash (/). Keeping this simple view, we can construct the major portions of the XML document to include the following six ingredients: (*1*) XML declaration (required), (*2*) Document Type Definition (or XMLSchema), (*3*)elements(required),(*4*)attributes,(*5*)entity,and(*6*) notations.

AnexampleofawelI-formedXMLdeclarationis
<?xmlversion=—1.0‖encoding=—iso-8859-1‖standalone="yes"?>

Following the XML declaration is a document type declaration that links to a DTD in a separate file.This is followed byaset ofdeclarations. Theseparts together comprisetheprolog. Asimple XML ―Hello World‖ example follows:

```
<?xml version="1.0"?>
<!DOCTYPEmessage[
<!ELEMENTmessage(#PCDATA)>
]>
<message>H
elloWorld!
</message>
```

In addition, XMLis both a powerful and essential language for Web Services. It is the open standardthatallowsdatatobeexchangedbetweenapplicationsanddatabasesovertheWeb.

XMLdoesnotoffersemanticsandlogiccapabilities.Thenextstepupthemarkuplanguage pyramid is RDF, which begins to establish a basis for semantics on the Web.

RDF LANGUAGE
The XML tags can often add meaning to data, however, actually understanding the tags is meaningfulonlytohumans.Forexample,giventhefollowingsegmentofXMLmarkuptags:

```
<book>
<title>G¨odel,Escher,Bach:AnEternalGoldenBraid<title>
</book>
```

A human might infer that: ―The book has the title *G¨odel, Escher, Bach: An Eternal Golden Braid*.‖ This simple grammatical sentence is understood to contain **three basic parts: a subject** [The book], **a predicate** [has title], and **an object** [*G¨odel, Escher, Bach: An Eternal Golden Braid*].

Regardless, the computer would not take action based upon this string (e.g., checking to see related titles, prices, availability, etc.) without additional explicit programming.

For machines to do more automatically, it is necessary to go beyond the notion of the HTML display model, or XML data model, toward a ―meaning‖ model. This is where RDF and metadata can provide new machine-processing capabilities built upon XML technology.

What is metadata? It is information about other data. Building upon XML, the W3C developed the RDF metadata standard. The goal was to add semantics defined on top of XML.

While RDF is actually built upon a very simple model and it can support very large-scale information processing. An RDF document can delineate precise relationships between vocabulary items by constructing a grammatical representation.

RDF Triple

The RDF model is based on statements made about resources that can be anything with an associated URI (Universal Resource Identifier). The ***basic RDF model produces a triple***, where**a resource** (the subject) is linked through an arc labeled with a property (the predicate) to avalue (the object).

The RDF statements can be represented as - ***A resource[subject] has a property[predicate] with a specific value[object].***

Thiscanbereducedtoatriple:(subject,predicate,object)
Subject,predicate,andobject canbedefined intermsofresources, properties,andvalueas:

Subject: The resource (a person, place, or thing) that the statement describes. A RDF resource canbeanythinginthedatamodel(document,user,product,etc) andisuniquelyidentifiedbya URI. A URI can be a URL (Universal Resource Locator).

Predicate: The property (name, city, title, color, shape, characteristic) of the subject (person, place, or thing) and is uniquely identified by a URI.

Object: Thevalue(DouglasR.Hofstadter,SanJose,―*G¨odel,Escher,Bach:AnEternalGolden Braid*,‖ blue, circle, strong) can be specified for the property (name, city, title, color, shape, characteristic),whichdescribesthesubject(person,place,orthing).Thisvaluecanbeanyvalid RDF data type. (RDF supports all of the XML data types.)

This simple model of the triple with URIs used by RDF to describe information has many advantages. One of the most important is that any data model can be reduced to a common storage format based on a triple.

ThismakesRDFidealforaggregatingdisparatedatamodelsbecauseallthedatafromallmodels can be treated the same. This means that information can be combined from many sources and processed as if it came from a single source.

The RDF relationships can be between two resources or between a resource and a literal. These relationships form arcs. The RDF arc can be graphically represented where the subject is shown as an oval, the predicate as a connecting arc or line, and the object as an oval. Graphs are easy to read and the directed arc removes any possibility of confusion over what are the subject and the objects.

Let us examine a very simple statement and identify the components that comprise an RDF model:

Ex:ConsiderthissentenceasanRDFStatement
―Thebookhasthetitle*G¨odel,Escher,Bach:AnEternalGoldenBraid*.‖
―Thebook[subject]hasthetitle[predicate]*G¨odel,Escher,Bach:AnEternalGolden Braid* [object].‖
Thiscan berepresented asthetriple:
(Thebookhas thetitle,*G¨odel, Escher,Bach:AnEternalGolden Braid*).

It is a directed graph with labeled nodes and labeled arcs. The arc is directed from the resource (the subject) to the value (the object), and this kind of graph is recognized in the AI community as a semantic net.

Wecanthinkofthetriple(*x*,P,*y*)asalogical formulaP(*x*,*y*)wherethebinarypredicatePrelates the object *x* to the object *y*.

Applyingthis toourtriple:
(Thebook,hasthe title,*G¨odel,Escher,Bach:AnEternalGolden Braid*)

Producesalogical formula:
_hasthetitle‘(Thebook,*G¨odel, Escher,Bach:AnEternalGolden Braid*)

Wherethebinary predicate(P): _hasthe title‘

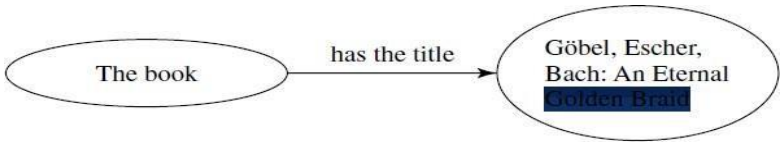


Fig:GraphicalrepresentationoftheRDFstatement

relatestheobject (*x*):Thebook
totheobject(*y*):*G¨odel,Escher,Bach:AnEternalGolden Braid.*

Think of a collection of interrelated RDF statements represented as a graph of interconnected nodes. The nodes are connected via various relationships. For example, let us say each node represents a person. Each person might be related to another person because they are siblings, parents, spouses, friends, or employees.

Eachinterconnectionislabeledwiththerelationshipname.

The RDF is used in this manner to describe these relationships. It does not actually include the nodes directly, but it does indirectly since the relationships point to the nodes. At any time, we could introduce a new node, such as a newborn child, and all that is needed is for us to add the appropriate relationship for the two parents.

BASIC ELEMENTS
MostoftheelementsofRDFconcernclasses, properties,and instancesofclasses.

Syntax
BothRDFandRDFSchema (RDFS)useXML-based syntax.

The RDF system provides a means of describing the relationships among resources in terms of named properties and values. Since RDF and XML were developed about the same time, RDF was defined as an excellent complement to XML. Encoding RDF triples in XML makes an

objectportableacrossplatformsandinteroperableamongapplications.BecauseRDFdatacanbe expressed using XML syntax, it can be passed over the Web as a document and parsed using existing XML-based software. This combination of RDF and XML enables individuals or programs to locate, retrieve, process, store, or manage the information objects that comprise a Semantic Web site.

Header

An RDF Document looks very much like all XML documents in terms of elements, tags, and namespaces. An RDFdocument starts with a header includingthe root element as an ***"rdf:RDF"*** element that also specifies a number of namespaces. It then defines properties and classes.

Document Parts	RDF Document
Header–XML Syntax declaration	`<?xml version="1.0"?>`
Root element tag	`<rdf:RDF`
XML namespaces for rdf and dc, as well as, the URLs where they are defined.	`xmlns:rdf=http://www.w3.org/1999/02/ 22-rdf-syntax-ns#xmlns:dc= "http://purl.org/dc/elements/1.1/">`
Inserting the Triple (subject, predicate, object) within the code.	`<rdf:Description rdf:about="SUBJECT"> <dc:PREDICATE>"OBJECT"</dc:PREDICATE> </rdf:Description>`
End of root element indicates end of RDF document.	`</rdf:RDF>`

Table: RDF document parts (header, XML syntax, root element, namespace, the RDF triple, and the end element)

Namespaces

The namespace mechanism of XML is also used in RDF. However, in XML, namespaces are only used to remove ambiguities. In RDF, external namespaces are expected to be RDF documents defining resources, which are used to import RDF documents.

To add a namespace to an RDF document, a namespace attribute can be added anywhere in the document, but is usually added to the RDF tag itself. The namespace declaration for RDF vocabularies usually points to a URI of the RDF Schema document for the vocabulary. We can add a namespace as:

<rdf:RDFxmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

Note,theprefixfortheRDFSyntaxisgivenas—rdf,‖theRDFSchemaisgivenas—rdfs,‖andthe DublinCoreschema(aspecialpublicationontology)isgivenas—dc.‖DCisawell- established RDF vocabulary for publications.

Description

The **"rdf:about"** attribute of the element ―rdf:Description‖ is equivalent to that of an ID attribute, but is often used to suggest the object may be defined elsewhere. A set of RDF statementsforma large graphrelatingthingstootherthings throughtheirproperties.Thecontent of **"rdf:Description"** elements are called property elements. The **"rdf:resource"** attribute and the **"rdf:type"** element introduces structure to the rdf document.

While RDF is required to be well formed, it does not require XML-style validation. The RDF parsers do not use Document Type Definitions (DTDs) or XML Schema to ensure that the RDF is valid.

DataTypes
Sometimes it is useful to be able to identify what kind of thing a resource is, much like how object-oriented systems use classes. The RDF system uses a type for this purpose. While thereare two very general types, ***a resource and a literal***, every resource may be given a precisetype.

Forexample,theresource—John‖mightbegivenatypeof—Person.‖Thevalueofthetype shouldbeanotherresourcethatwouldmeanthatmoreinformationcouldbeassociatedwiththe type itself.

Aswith otherproperties, typescanbespecifiedwitha triple:
<http://www.web-iq.com/people/John>,
rdf:type,http://xmlns.com/wordnet/1.6/Person

EX:—Thebookhasthetitle
G¨odel,Escher,Bach:AnEternalGoldenBraid,‖as:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Descriptionrdf:about=—http://www.amazon.com/books‖>
<dc:title>G¨odel,Escher,Bach:AnEternalGolden Braid</dc:title>
</rdf:Description>
</rdf:RDF>
```

Vocabularies
Anykindofbusinessresourcevocabulariescanbeusedtomodelbusinessresourcesusingthe syntax of RDF.

BecauseRDFcreatesdomain-specificvocabulariesthatarethenusedtomodelresources,we can use RDF to model business-specific resources. The only limitation is the need for industry cooperation in developing an interoperable vocabulary. We can consider RDF as a way of recording information about resources.

TheRDFrecordedinXMLisapowerfultool.ByusingXMLwehaveaccesstoagreatnumber of existing XML applications, such as parsers and APIs.

ClassesandProperties
TheRDFandRDFSchema(RDFS)classesandpropertiescanbefoundat: ***RDFW3C specifications***

RDFModel and Syntax Specification:
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

RDFS Specification:
http://www.w3.org/TR/2003/WD-rdf-schema-20030123/

Collections
Acollectionisconsideredtobeafinite groupingofitemswithagiven terminator.

WithinRDF,acollectionisdefinedthroughtheuseof**rdf:parseType**="Collection"and through listing the collected resources within the collection block.

Reification

TheRDFallowsustomakestatementsaboutstatementsusingareificationmechanism.Thisis particularly useful to describe belief or trust in other statements.

Thefollowingexamplediscussestheinterpretationofmultiplestatementsinrelationshipto RDF statements.

InterpretingMultipleSentencesasRDFStatement:

Letusstart withfivesimple factsthatwewishtorepresent asRDF triplets.

1. Thename ofthisURI(mailto: Hofstadter@yahoo.com)isDouglas R.Hofstadter.(Itisthename)
2. Thetypeofthis URI(mailto: Hofstadter@yahoo.com)is atypeofperson.
3. TheauthorofthisURI(mailto:Hofstadter@yahoo.com)isanauthor of isbn:0465026567.
4. Theid ofthis URI(isbn:0465026567) istheidentityof abook.
5. ThetitleofthisURI(isbn:0465026567)hasthetitleof *G¨odel,Escher, Bach:AnEternal Golden Braid*.

Subject	Predicate	Object
mailto:Hofstadter@yahoo.com	name	Douglas R. Hofstadter
mailto:Hofstadter@yahoo.com	type	Person
mailto:Hofstadter@yahoo.com	author-of	isbn: 0465026567
isbn:0465026567	type	book
isbn:0465026567	title	*Gödel, Escher, Bach: An Eternal Golden Braid*

RDFTripletData Table

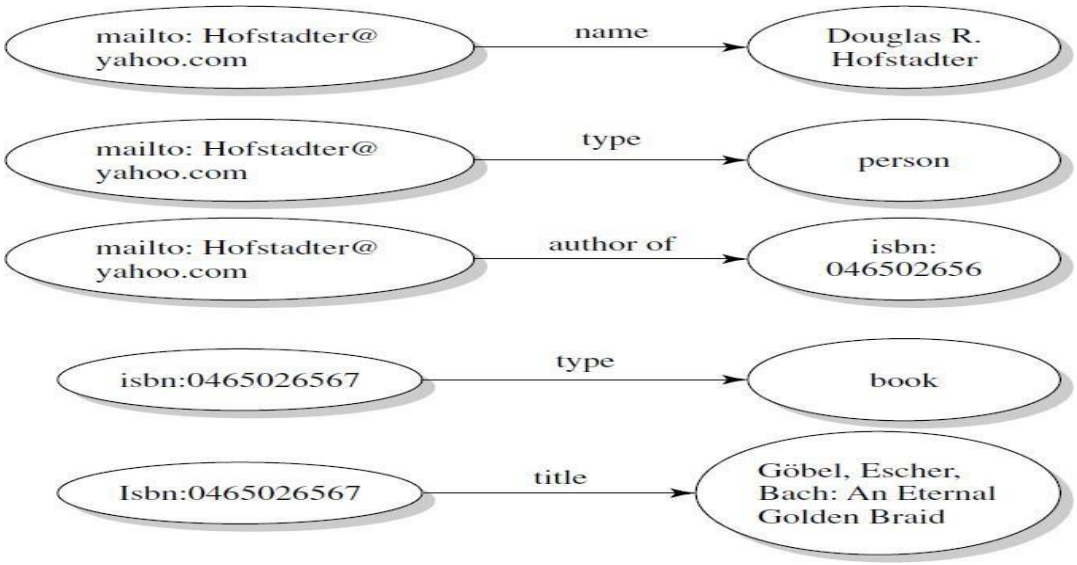
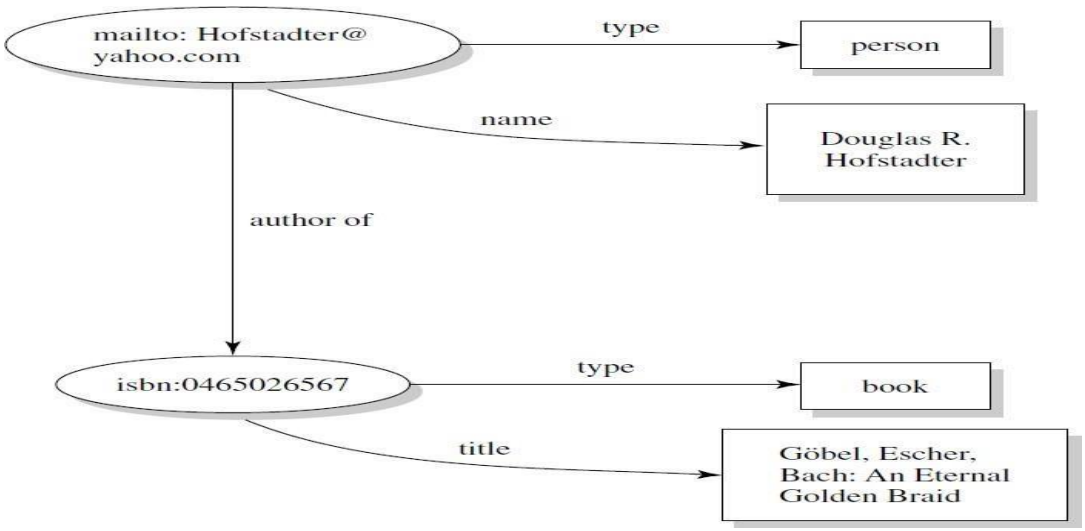


Fig:Individual graphsforeach tripletstatementof aboveExample



MergedRDFgraph

Theserializedformof theRDFdocumentforthisexamplecanbewrittenas:

SerializationofRDFStatementas
<?xml version="1.0"?>
<Classrdf:ID="book"
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#xmlns="uri">
<title>G¨odel,Escher,Bach:AnEternalGoldenBraid</title>
. . .
</Class>

In anyRDF graph, a subgraph would be asubset ofthetriples contained in thegraph. Each triple is its own unique RDF graph. The union of two or more graphs is a new graph called a merged graph.

RDF SCHEMA

The RDF provides a simple yet powerful model for describing information including a basic directed graph, but the semantics (meaning of the information) is described using RDFS. The purpose of RDFS is to provide an XML vocabulary that can express classes and their (subclass) relationships, as well as to define properties associated with classes. This Schema is actually a primitive ontology language.

Classes and Properties

Todescribeaspecificdomain,wespecifythe—things‖wewanttotalkabout.Wecantalkabout either individual objects (resources) or classes that define types of objects.

A class can be considered as a set of elements. Individual objects that belong to a class are instances of the class. The relationship between instances and classes in RDF is expressed by **"rdf:type."**

ThethreemostimportantRDFconceptsare"Resource"(rdfs:Resource),"Class"(rdfs:Class), and "Property" (rdf:Property).

Theseareall—classes.‖Classisintherdfsnamespace.Propertyisintherdfnamespace.

Wejustusetherdf:typeproperty,todeclarethatsomethingisa—type‖ofsomethingelseas following:

rdfs:Resourcerdf:typerdfs:Class.
rdfs:Class rdf:type rdfs:Class.
rdf:Property rdf:type rdfs:Class.
rdf:type rdf:type rdf:Property.

Thismeansthat—ResourceisatypeofClass,ClassisatypeofClass,PropertyisatypeofClass, and type is a type of Property.‖

Forexample,the***rdf:ID***providesanameforthe classwhiletheconjunction(AND)oftwo subClassOf statements is a subset of the intersection of the classes:

<rdfs:Classrdf:ID="Set1ANDSet2">
<rdfs:subClassOfrdf:resource="#Set1"/>
<rdfs:subClassOfrdf:resource="#Set2"/>
</rdfs:Class>

RDFandRDFSClasses

Class Name	Comment
rdfs:Resource	Class of all resources
rdfs:Literal	Class of all literals (strings)
rdfs:XMLLiteral	The class of XML literals
rdfs:Class	Class of all classes
rdf:Property	Class of all properties
rdfs:Datatype	Class of datatypes
rdf:Statement	Class of all reified RDF statements
rdf:Bag	An unordered collection
rdf:Seq	An ordered collection
rdf:Alt	A collection of alternatives
rdfs:Container	This represents the set Containers
rdfs:ContainerMembershipProperty	The container membership properties, rdf:1, rdf:2, . . ., all of which are subproperties of 'member'
rdf:List	The class of RDF Lists

RDFandRDFSProperties

Property Name	Comment
rdf:type	Related a resource to its class
rdfs:subClassOf	Indicates membership of a class
rdfs:subPropertyOf	Indicates specialization of properties
rdfs:domain	A domain class for a property type
rdfs:range	A range class for a property type
rdfs:label	Provides a human-readable version of a resource name.
rdfs:comment	Use this for descriptions.
rdfs:member	A member of a container.
rdf:first	The first item in an RDF list. Also often called the head.
rdf:rest	The rest of an RDF list after the first item, called the tail.
rdfs:seeAlso	A resource that provides information about the subject resource
rdfs:isDefinedBy	Indicates the namespace of a resource.
rdf:value	Identifies the principal value (usually a string) of a property when the property value is a structured resource.
rdf:subject	The subject of an RDF statement.
rdf:predicate	The predicate of an RDF statement.
rdf:object	The object of an RDF statement.

1. quadrilaterals(X)→polygons(X)
2. polygons(X)→shapes(X)
3. quadrilaterals(squares)

Andnowfrom this knowledgethefollowingconclusions can bededuced:
1. polygons(squares)
2. shapes (squares)
3. quadrilateral(X)→shapes(X)

Thehierarchyrelationship of classesis shown in Figure

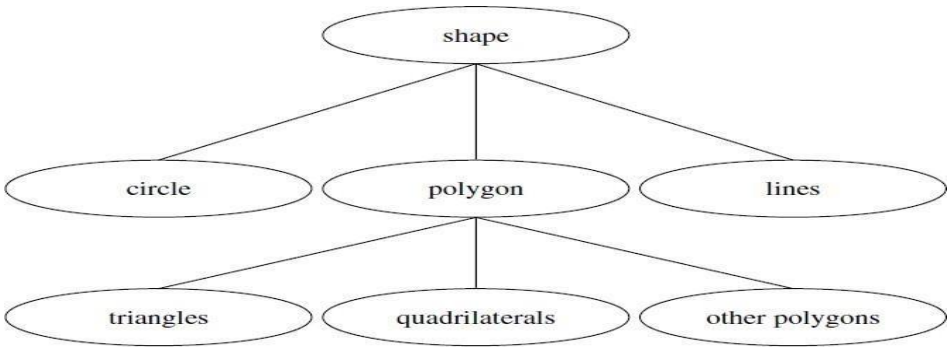


Fig:Hierarchyof Classes

OntologyWeb Language(OWL)

For machines to perform useful automatic reasoning tasks on Web documents, the language machines use must go beyond the basic semantics of XMLSchema and RDF Schema. They will require a more expressive and reasoning ontology language; as a result, the World Wide Web Consortium (W3C) has defined Web Ontology Language (called OWL).

Web Ontology Language enhances RDF with more vocabulary for describing properties and classes, including relations between classes (e.g., disjointedness), cardinality (e.g., exactly one), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes.

Ontologies are usually expressed in a logic-based language, so that accurate, consistent, and meaningful distinctions can bemade amongthe classes,properties,and relations. Someontology tools can perform automated reasoning using the ontologies, and thus provide advanced services to intelligent applications, such as conceptual (semantic) search and retrieval, software agents, speech understanding, knowledge management, intelligent databases, and e-commerce.

OWL was developed in 2003, when the W3C began final unification of the disparate international ontology efforts into a standardized ontology. Web Ontology Language is designed to express a wide variety of knowledge, as well as provide for an efficient means to reason withit in order to express the most important kinds of knowledge. Using an ontology with a rule- based system, we can reach logic inferences about Web information.

OWL can be used to describe the classes and relations between classes that are inherent in Web documents and applications.

A set of XMLstatements by itself does not allow us to reach a conclusion about any other XML statements. To employ XML to generate new data, we need knowledge embedded in some

proprietary procedural code that exists as a server page on a remote server. However, a set of OWL statements by itself can allow us to reach a conclusion about another OWL statement.

OWL ontology documents are designed to be modular and independent. They can be combined dynamically to provide additional meaning if required.

Web Ontology Language ontology documents have a logical consistency to them. They provide machine-based systems with the ability to interpret the declared relationships within them. More importantly, they also allow mathematical techniques to be applied that can interpret and calculate the relationships that are implied within the logical formulations. These inferencesmaketheuseofOWLontologiestractableandrealisticfororganizations,drasticallyreducingthe amount of information that has to be modeled, encoded, or worked around by systems engineers and integrators.

COMPATIBILITYOFOWLAND RDF/RDFS

The layered architecture of the Semantic Web would suggest that one way to develop the necessaryontologylanguage is to extend RDFSchema byusingthe RDF meaningof classes and properties (rdfs:classes, etc.) and adding primitives to support richer expressiveness.

TheW3ChasdefinedOWLtoincludethreedifferentsublanguages(OWLFull,OWLDL,OWL Lite) in order to offer different balances of expressive power and efficient reasoning.

OWLFull

The entire language is called OWL Full and it uses all the primitives and allows their combination with RDF and RDFS. The OWL Full supports maximum expressiveness and the syntactic freedom of RDF, but has no computational guarantees. For example, in OWL Full, a class can be treated simultaneously as a collection of individuals and as an individual in its own right.

The advantage of OWL Full is that it is fully compatible with RDF syntax and semantics. Any legal RDF document is also a legal OWL Full document. Any valid RDF–RDFS conclusion is also a valid OWL Full conclusion. The disadvantage of OWL Full is that the language is undecidable, and therefore cannot provide complete (or efficient) reasoning support.

OWL DL

Web Ontology Language DL (Descriptive Logic) is a sublanguage of OWL Full that restricts how the constructors from OWL and RDF can be used. This ensures that the language is related to description logic. Description Logics are a decidable fragment of First-Order Logic (FOL).

The OWL DL supports strong expressiveness while retaining computational completeness and decidability. It is therefore possible to automatically compute the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL DL.

The advantage of this sublanguage is efficient reasoning support. The disadvantage is the loss of full compatibility with RDF. However, every legal OWL DL document is a legal RDFdocument.

OWL Lite

Further restrictingOWLDLproduces a subset of the language called OWL Lite, which excludes enumerated classes, disjointness statements, and arbitrary cardinality. The OWL Lite supports a classification hierarchy and simple constraints.

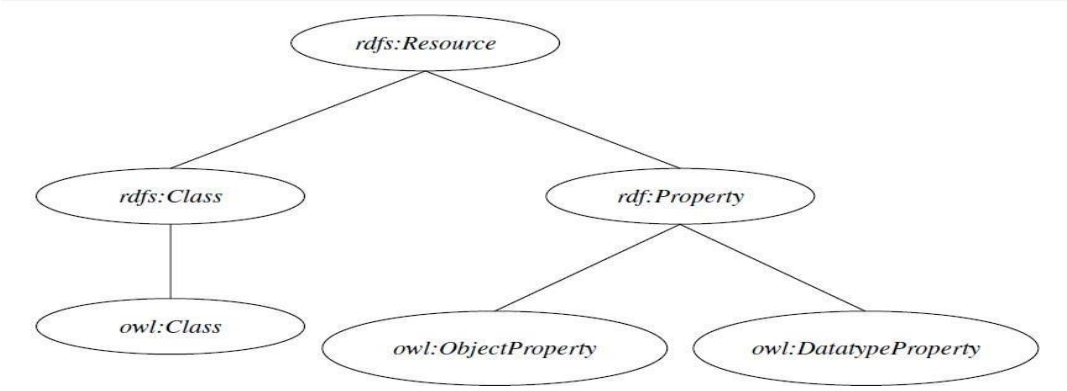


Fig:TheOWLandRDF-RDFSsubclassrelationships An

OWL document identifies:

- **Classhierarchy:**Definesclass–subclassrelationships.
- **Synonym:**Identifiesequivalentclassesandequivalentproperties.
- **Classassociation:**Mapsoneormoreclassestooneormoreclasses,throughtheuseofa property (i.e., domain/range).
- **Propertymetadata:**Containsmetadataforproperties.
- **Classdefinition:**Specifies thecomposition ofclasses.

TheOWLSyntaxSpecification http://www.w3.org/TR/owl-features/

WebOntologyLanguagedefinestheclassesandproperties,aswellastheirrelationship toeach other in the document; consequently, they are extremely similar to RDF Schema.

Unlike RDF, the OWL vocabulary is quite large. Like RDF, OWL makes use of elements from RDFS. However, OWL has several concepts unique to it, such as Boolean combination of class expressions and property restrictions, which add a layer of reasoning to applications. Both the RDFS and OWL are compatible.

BASIC ELEMENTS

MostoftheelementsofanOWLontologyconcernclasses,properties,instancesofclasses,and relationships between these instances.

Syntax

TheOWLbuilds on RDF andRDFS and uses RDFs XML-based syntax.

OWLDocumentParts

Document Parts	OWL Document
Header	
XML Syntax	<?xml version="1.0" encoding="UTF-8"?>
Root element	<rdf:RDF
Namespace	xmlns:iq = “http://www.web-iq.com”> xmlns:owl = “http://www.w3.org/2002/07/owl#” xmlns:rdf = “http://www.w3.org/1999/02/22-rdf-syntax-ns#” xmlns:rdfs = “http://www.w3.org/1999/02/22-rdf-schema#” xmlns:dc = “http://purl.org/dc/elements/1.1/” xmlns:xsd = “http://www.w3.org/2000/1/XMLSchema#”>
OWL properties and classes	<owl:Ontology rdf:about = “http://www.amazon.com”> <owl:versionInfo> $ID: Overview.html </owl:versionInfo> <dc:creator> Douglas R. Hofstadter </dc:creator> <dc:title> Gödel, Escher, Bach: An Eternal Golden Braid </dc:title>
End of OWL	</owl:Ontology>
End of RDF	</rdf:RDF>

Header

An OWL document contains an OWL ontology and is an RDF document with elements, tags,and namespaces. An OWL document starts with a header that identifies the root element as an ***rdf:RDF element***, which also specifies a number of namespaces.

Class Elements

Classes are defined using an owl:Class element.An example of anOWL class―computer‖ isdefined with a subclass ―laptop‖ as

<owl:Classrdf:ID="Computer">
<rdfs:subClassOfrdf:resource="#laptop"/>
</owl:Class>

Equivalenceofclassesisdefinedwith**owl:equivelentClass**.

Property

ApropertyinRDFprovidesinformationabouttheentityitisdescribing.Propertycharacteristics increase our ability to understand the inferred information within the data.

Thefollowingspecialidentifierscanbeusedtoprovideinformationconcerningproperties and their values:

• **inverseOf:**Onepropertymaybestatedto be theinverseofanotherproperty.
• **TransitiveProperty:**Properties maybestated to betransitive.
• **SymmetricProperty:**Propertiesmaybestated tobesymmetric.
• **FunctionalProperty:**Properties maybestatedtohaveauniquevalue.
• **InverseFunctionalProperty:**Propertiesmaybestatedto beinversefunctional.

TheOWLLiteallowsrestrictionstobeplacedonhowpropertiescanbeusedbyinstancesofa class.

• **allValuesFrom:**TherestrictionallValuesFrom isstatedon apropertywith respecttoaclass.
• **someValuesFrom:** The restriction someValuesFrom is stated on a property with respect to a class. A particular class may have a restriction on a property that at least one value for that property is of a certain type.
• **minCardinality:** Cardinality is stated on a property with respect to a particular class. If a minCardinality of 1 is stated on a property with respect to a class, then any instance of that class will be related to at least one individual by that property.
• **maxCardinality:** Cardinality is stated on a property with respect to a particular class. If a maxCardinalityof 1 is stated on a property with respect to a class, then any instance of that class will be related to at most one.
• **cardinality:** Cardinality is provided as a convenience when it is useful to state that a property on a class has both minCardinality 0 and maxCardinality 0 or both minCardinality 1 and maxCardinality1.
• **intersectionOf:**OWLLiteallowsintersectionsof namedclassesand restrictions.

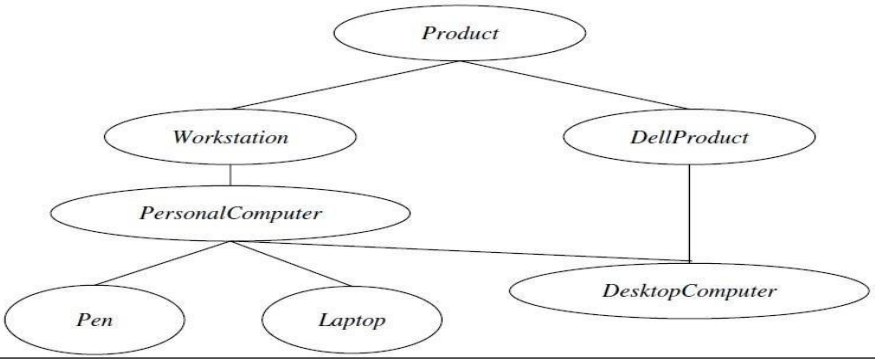


Fig:Classesandsubclassesofthe computer ontology

OWLEXAMPLE:COMPUTE ONTOLOGY

Theserialization forthecomputer ontology is

```
<[DOCTYPEowl [
<!ENTITYxsd—http://www.w3.org/2001/XMLSchema#‖>
]>
<rdf:RDF
xmlns:rdf=‖http://www.w3.org/1999/02/22-rdf-syntax-ns#‖
xmlns:rdfs=‖http://www.w3.org/200/01/rdf-schema#‖
xmlns:xsd=‖http://www.w3.org/2001/XMLSchema#‖
xmlns:owl=‖http://www.w3.org/2002/07/owl#‖
xmlns=‖http://www.web-iq.com/computer.owl#‖>
<owl:Ontologyrdf:about=‖‖>
<owl:versionInfo>
</owl:versionInfo>
</owl:Ontology>
<owl:Classrdf:ID=—Product‖>
</owl:Class>
<owl:Classrdf:ID=—Workstation‖>
<rdfs:label>Device</rdfs:label>
<rdfs:subClassOfrdf:resource=—#product‖/>
</owl:Class>
<owl:Classrdf:ID=—DellProducts‖>
<rdfs:label>DellDevices</rdfs:label>
<owl:intersectionOfrdf:parseType=—Collection‖>
<owl:Classrdf:about—#product‖/>
<owl:Restriction>
<owl:onPropertyrdf:resource=—#manufacturedby‖/>
<owl:hasValue rdf:datatype=—&xsd;string‖>
DELL
</owl:hasValue>
</owl:Restriction>
</owl:Intersection>
</owl:Class>
<owl:Classrdf:ID=—PersonalComputer‖>
<rdfs:subClassOfrdf:resource=—#workstation‖/>
</owl:Class>
<owl:Classrdf:ID=—Laptop‖>
<rdfs:subClassOfrdf:resource=—#personalcomputer‖/>
</owl:Class>
<owl:Classrdf:ID=—DesktopComputer‖>
<rdfs:subClassOfrdf:resource=—#personalcomputer‖/>
<rdfs:subClassOfrdf:resource=—#dellproduct‖/>
</owl:Class>
<owl:Classrdf:ID=—Pen‖>
<rdfs:subClassOfrdf:resource=—#personalcomputer‖/>
</owl:Class>
<owl:DatatypePropertyrdf:ID=—manufacturedby‖>
```

<rdf:domainrdf:resource=―#product‖/>
<rdf:rangerdf:resource=―&xsd:string‖/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID=―price‖>
<rdf:domainrdf:resource=―#product‖/>
<rdf:rangerdf:resource=‖&xsd:string‖/>
</owl:DatatypeProperty>
</rdf:RDF>

This onotologydemonstrates siblings ina hierarchymaynot bedisjoint.

OwlCapabilitiesandLimitations

TheOWLlanguageoffersthefollowingfeatures:lesschanceofmisinterpretation,understanding each other‘s data‘s semantics, and OWL uses existing XML syntax to express semantics. The OWL document can be extensible, reusable, and avoids misinterpretation.

AdditionalOWLproblemsincludenoadhocdiscoveryandexploitation,thusanapplicationmay not be able to effectively process new data when it is encountered.

ComparisontotheUnifiedModellingLanguage(UML)

UMLismostcommonlyusedintherequirementsspecificationanddesignofobjectoriented software in the middle tier of enterprise applications.

ThechiefdifferencebetweenUMLandRDF(S)/OWListheirmodellingscope:UMLcontains modelling primitives specific for a special kind of information resource, namely objects in an information system characterized by their static attributes and associations, but also their dynamic behavior. Many of the modelling primitives of UML are thus specific to objects and their role in OO systems; interfaces, functions etc.

Uniquefeaturesof RDF/OWL

– Ingeneral, the modelling ofRDF is less constrainedthanthat ofUML, whichmeans that many RDF models have no equivalent in UML. OWL DL also provides more primitives than UML such as the disjointness, union, intersection and equivalence of classes.

– OWLallowstodescribedefinedclasses,i.e.definitionsthatgivenecessaryandsufficient conditions for an instance to be considered as a member of the class.

– RDF/OWLgiveshigh prioritytoitsproperties.TheytraetthePropertiesareglobal:theydo not belong to any class, while UML attributes and associations are defined as part of the description of a certain class. In other words, the same property can be used with multiple classes.

– Propertiescanbedefinedas subpropertiesofotherproperties.

– Classescanbetreated asinstances,allowingformeta-modelling.

– RDFreificationismoreflexiblethantheassociationclassmechanismofUML.Forexample, statements concerning literal values can also be reified in RDF

– Allnon-blankRDFresourcesareidentifiedwitha URI, UMLclasses,instances,attributesetc.

– Instancescanand usuallyhavemultipletypes.

Uniquefeaturesof UML

– UMLhas the notion ofrelationship roles,which is not present inRDF/OWL.

– UML allows n-ary relations, which are not part of RDF, although they can bere-represented ina number of ways.

– TwocommontypesofpartwholerelationsareavailableinUML(aggregationandcomposition). These can be remodelled in OWL to some extent.

– UML makes a clear differentiation between attributes and associations. This is also different from the distinction between datatype and object-properties in OWL. On the one hand, attributes can have instances as values, while datatype properties can only have literal values.

Comparison to the Extensible Markup Language (XML) and XML Schema

Up to date XML is the most commonly used technology for the exchange of structured informationbetweensystemsandservices.FromalllanguagesdiscussedtheroleofXMListhus the most similar to RDF in its purpose.

Themost commonlyobserved similaritybetween XMLand RDFis asimilaritybetween the data models: a directed graph for RDF, and a directed, ordered tree for XML. In XML, the tree is defined by the nesting of elements starting with a single root node. This model originates from the predecessor of XML called SGML which was primarily used for marking up large text documents. Text documents follow the tree structure themselves as paragraphs are nested in subsections, subsections are nested in sections, sections are nested chapters etc. The ordering of the children of an element matters, which is again inherited from the text processing tradition.

Namely, schemas written in XML schema languages not only define the types of elements and their attributes but also prescribe syntax i.e. the way elements are allowed to be nested in thetree. XML documents can be validated against a schema on a purely syntacticlevel.

RDF models are based on arbitrary directed graphs. They are developed from single edges between the nodes of classes or instances.

XML has a variety of schema languages like XMLSchema and Relax NG. Schemas in XML schema language define the elements type, their attributes and the syntax validation of XML document against a schema is done syntactically. RDF Schema language does not introduce constraints directly on the graph model rather they effect the interpretations of data.

RDF for web based data exchanges has an advantage that agreement on shared XML format needs a stronger commitment than the agreement mode by using RDF. This agreement of exchanging RDF documents considering only the individual statements like a simple subject, predicate and object model.

UNIT-III

Ontology Engineering , Constructing Ontology, Ontology Development Tools, Ontology Methods, OntologySharingandMerging,OntologyLibrariesandOntologyMapping, Logic,Ruleand Inference Engines.

ONTOLOGYENGINEERING:

Ontologyis theformal specification ofterms within adomain and their relationships. Itdefines a common vocabulary for the sharing of information that can be used by both humans and computers. Ontologies can be in the form of lists of words; taxonomies, database schema, frame languages and logics. The main difference between these forms is their expressive power.

Ontology together with a set of concept instances constitutes a knowledge base. If a program is designed to compare conceptual information across two knowledge bases on the Web, it must know when any two terms are being used to mean the same thing. Ideally, the program musthave a way to discover common meanings for whatever knowledge bases it encounters. Typically, an ontology on the Web will combine a taxonomy with a set of inferencerules.

Taxonomy is defined as a set of classes of objects and their relationships. These classes, subclasses, and their relationships are important tools for manipulating information. Their relations are described by assigning properties to classes and allowing subclasses to inherit these properties. An ontology then is a taxonomy plus inference.

Ontology inference rules allow manipulation of conceptual information. The most important ontology relationship is the subsumption link (e.g., subtype and supertype link).

When a network of concepts is represented by a tree, it rigorously defines the taxonomy. While ontologycan sometimes be modularized as a set of trees, some advocate that all ontology should be taxonomic, but others favor a lattice structure.

Ontology engineering seeks a common vocabulary through a data collection process thatincludes discussions, interviews, document analysis, and questionnaires.

Existing ontologies on a subject are discovered, assessed, and reused as much as possible to avoid ―reinventing the wheel.‖ As part of this process, ontologies are designed as living objects with a maintenance cycle.

Ontology Applications:
The simplest ontology consists of a simple taxonomy with a single relation. Categories of ontology applications can be grouped as

• **Neutral Authoring:** The author of an object in a single language translates into a different format for use in alternative applications.
• **Ontology as Specification:** Ontology of a given domain is created and used as a basis for specification and development of some software. This approach allows documentation, maintenance, reliability and knowledge (re)use.
• **Common Access to Information:** Information in an inaccessible format becomes intelligible by providing a shared understanding of the terms, or by mapping between sets ofterms.
• **Ontology-BasedSearch:**Ontologyisused forsearchinganinformationrepository.

CONSTRUCTINGONTOLOGY:

Ontology permits sharing common understanding of the structure of information among people and software agents. Since there is no unique model for a particular domain, ontology development is best achieved through an iterative process. Objects and their relationships reflect the basic concepts within an ontology.

An iterative approach for building ontologies starts with a rough first pass through the main processes as follows:

• **First,** set the scope. The development of an ontology should start by defining its domain and scope.

Severalbasicquestions arehelpful atthis point:
What will the ontology cover?
Howwilltheontologybeused?
Whatquestionsdoestheontologyanswer? Who
will use and maintain the ontology?
Theanswers maychangeas weproceed, buttheyhelp limit the scopeof the model.

• **Second**, evaluate reuse. Check to see if existing ontologies can be refined and extended. Reusing existing ontologies will help to interact with other applications and vocabularies. Many knowledge-representation systems can import and export ontologies directly for reuse.

• **Third,** enumerate terms. It is useful to list all terms, what they address, & what properties they have. Initially, a comprehensive list of terms is useful without regard for overlapping concepts. Nouns can form the basis for class names & verbs can form the basis for propertynames.

• **Fourth**, define the taxonomy. There are several possible approaches in developing a class hierarchy: a top-down process starts by defining general concepts in the domain. A bottom-up development process starts with the definition of the most specific classes, the levels of the hierarchy, with subsequent grouping of these classes into more general concepts.

• **Fifth,** define properties. The classes alone will not provide enough information to answer questions. We must alsodescribe the internal structure of concepts. Whileattachingproperties to classes one should establish the domain and range. Property constraints (facets) describe or limit the set of possible values for a frame slot.
• **Sixth,** define facets. Up to this point the ontology resembles a RDFS without any primitives fromOWL. In thisstep,thepropertiesadd cardinality, values, and characteristicsthatwillenrich their definitions.
• **Seventh,** the slots can have different facets describing the value type, allowed values, the number of the values (cardinality), and other features of the values.

Slotcardinality:the numberofvaluesa slothas. Slotvaluetype: thetypeofvalues aslot has.
Minimumandmaximumvalue:arangeofvalues foranumeric slot.
Defaultvalue:thevalue aslothasunlessexplicitlyspecifiedotherwise.

• **Eighth,** define instances. The next step is to create individual instances of classes in the hierarchy.

• **Finally,** check for anomalies. The Web-Ontology Language allows the possibility of detecting inconsistencies within the ontology. Anomalies, such as incompatible domain and range definitions for transitive, symmetric, or inverse properties may occur.

ONTOLOGYDEVELOPMENTTOOLS:

Belowis alistof someofthemostcommon editorsused forbuilding ontologies:

• **DAG-Edit**providesaninterfacetobrowse,queryandeditvocabularieswithaDAGdata structure: **http://www.geneontology.org/#dagedit**
• **Protege2000** is themost widelyusedtool forcreatingontologies and knowledgebases: **http://protege.stanford.edu/index.shtml**
• **WonderTools** isan index forselectinganontology-buildingtool: **http://www.swi.psy.uva.nl/wondertools/**
• **WebOnto**isaJavaappletcoupledwithaWebserverthatallowsuserstobrowseandedit knowledge models: **http://kmi.open.ac.uk/projects/webonto/**

ONTOLOGY“SPOT” EXAMPLE

Portions of the following example for the ―spot‖ ontology were taken from http://www.charlestoncore.org/ont/example/index.html.

The spot ontology consists of ***three owl:Classes (spot, ellipse, and point)*** and ***six rdf:Properties (shape, center, x-position, y-position, x-radius, y-radius).*** Together, these vocabularies can be used to describe a spot.

Classes

ThethreeOWLclassesareSpot: Atwodimensional(2D)―spot‖definedasaclosedregionon the plane.

ExampleofSPOT Ontology

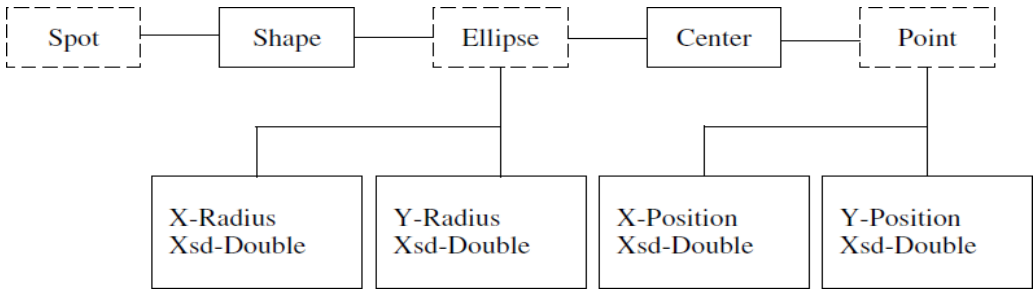


Fig:Example Ontology

Point: ApointisdefinedasalocationonaCartesianplane.Ithastwoattributes;its *x*-position and *y*-position on an implicit coordinate system of the plane.

Ellipse: Ellipse here is defined as a circle stretched along either the *x*- or *y*-axis of a coordinate system. The major and minor axes of an Ellipse parallel the coordinates of the implicitcoordinate system.

Properties

ThesixRDFpropertiesareShape: ASpotassumesashapeofanEllipse.Thereforethe domain of shape is Spot and the range of Spot is Ellipse.

Center: Thecenteristhecenterpoint oftheEllipse.It has a***rdfs:domain*** ofEllipseand a ***rdfs:range***ofPoint.

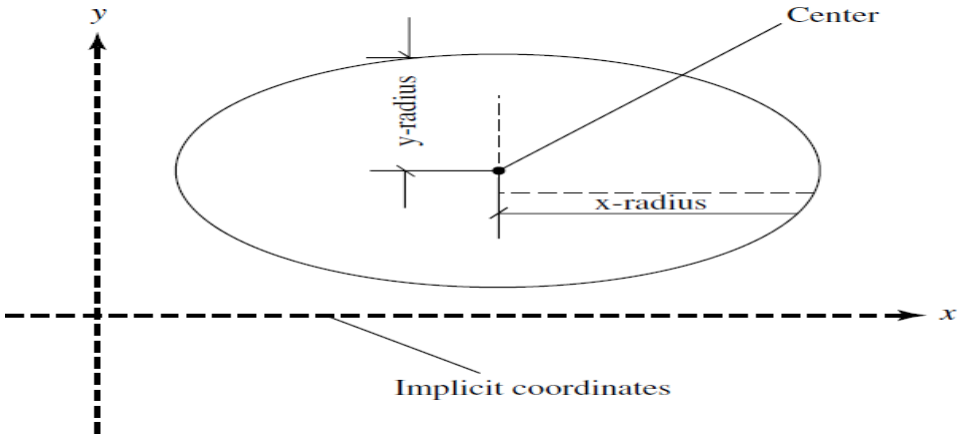


Fig:Ellipse definition

***x*-Position:** An*x*-positionisanowl:DatatypepropertythathasadomainofPoint.Itsvalue(of type xsd:double) is the distance from the origin on the *x*-axis of the coordinate system.

***y*-Position:** A*y*-positionisaowl:DatatypepropertythathasadomainofPoint.Itsvalue(oftype xsd:double) is the distance from the origin on the *y*-axis of the coordinate system.

x*-Radius:x*-radiusisaowl:Datatypepropertythathasardfs:domainofEllipse. Itistheradius parallel to the *x*-axis of the coordinate system.

***y*-Radius:** A*y*-radiusisaowl:Datatypepropertythathasardfs:domainofEllipse. Itistheradius parallel to the *y*-axis of the coordinate system.

TheOWLfileforthisontologyexample
(http://www.charlestoncore.org/ont/example/index.html)isasfollows:

<?xmlversion="1.0"encoding="iso-8859-1"?>
<!DOCTYPErdf:RDF(...)>
<rdf:RDF xmlns="http:// example#"
xmlns:example="http:// example#" xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/"xml:base="
http:// /example">
<owl:Ontologyrdf:about="">
<rdfs:isDefinedByrdf:resource="**http://example/**" />
<dc:author>**Smith**</dc:author>
<dc:title>**ExampleOntology**</dc:title>
<rdfs:comment>**Thisfiledefinesapartialontologyin OWL**</rdfs:comment>
<owl:versionInfo>**2005**</owl:versionInfo>
</owl:Ontology>
<owl:Classrdf:ID="**Spot**"/>
<owl:Classrdf:ID="**Ellipse**"/>

```
<owl:Classrdf:ID="Point"/>
<owl:ObjectPropertyrdf:ID="shape">
<rdfs:domainrdf:resource="#Spot" />
<rdfs:rangerdf:resource="#Ellipse"/>
</owl:ObjectProperty>
<owl:ObjectPropertyrdf:ID="center">
<rdfs:domainrdf:resource="#Ellipse"/>
<rdfs:rangerdf:resource="#Point"/>
</owl:ObjectProperty>
<owl:DatatypePropertyrdf:ID="x-radius">
<rdfs:domainrdf:resource="#Ellipse"/>
<rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID="y-radius">
<rdfs:domainrdf:resource="#Ellipse"/>

<rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID="x-position">
<rdfs:domainrdf:resource="#Point" />
<rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
<owl:DatatypePropertyrdf:ID="y-position">
<rdfs:domainrdf:resource="#Point" />
<rdfs:rangerdf:resource="http://www.w3.org/2001/XMLSchema#double"/>
</owl:DatatypeProperty>
</rdf:RDF>
```

ONTOLOGYMETHODS

Several approaches for developing ontologies have been attempted in the last two decades. In 1990, Lenat and Guha proposed the general process steps. In 1995, the first guidelines were proposed on the basis of the Enterprise Ontology and the TOVE (TOronto Virtual Enterprise) project. A few years later, the On-To-Knowledge methodology was developed.

The Cyc Knowledge Base (see http://www.cyc.com/) was designed to accommodate all ofhuman knowledge and contains about 100,000 concept types used in the rules and facts encoded in its knowledge base. ***The method used to build the Cyc consisted of threephases.***

The**firstphase**manuallycodifiedarticlesandpiecesofknowledgecontainingcommonsense knowledge implicit in different sources.

The**secondandthirdphase**consistedofacquiringnewcommonseknowledgeusingnatural language or machine learning tools.

The **Electronic Dictionary Research (ERD) project** in Japan has developed a dictionary with over 400,000 concepts, with their mappings to both English and Japanese words. Although the EDR project has many more concepts than Cyc, it does not provide as much detail for each one (see http://www.iijnet.or.jp/edr/).

WordNet is a hierarchy of 166,000 word form and sense pairs. WordNet does not have as much detail as Cyc or as broad coverage as EDR, but it is the most widely used ontology for natural language processing, largely because it has long been easily accessible over the Internet (see http://www.cogsci.princeton.edu/~wn/).

Cyc has the most detailed axioms and definitions; it is an example of an axiomatized or formal ontology. Both EDR and WordNet are usually considered terminological ontologies. The difference between a terminological ontology and a formal ontology is one of degree: as more axioms are added to a terminological ontology, it may evolve into a formal or axiomatized ontology.

The main concepts in the ontology development include: a top-down approach, in which the most abstract concepts are identified first, and then, specialized into more specific concepts; a bottom-up approach, in which the most specific. Concepts are identified first and then generalized into more abstract concepts; and a middle-out approach, in whichthemost important concepts are identified first and then generalized and specialized into other concepts.

Methontology was created in the Artificial Intelligence Lab from the Technical University of Madrid (UPM). It was designed to build ontologies either from scratch, reusing other ontologies as they are, or by a process of reengineering them. The Methontology framework enables the construction of ontologies at the knowledge level. It includes the identification of the ontology development process, a life cycle based on evolving prototypes, and particular techniques tocarry out each activity. The ontology development process identifies which tasks should be performed when building ontologies (scheduling, control, quality assurance, specification, knowledgeacquisition,conceptualization,integration,formalization,implementation,evaluation, maintenance, documentation, and configuration management).

ThemainphaseintheontologydevelopmentprocessusingtheMethontologyapproachis the conceptualization phase.

By comparison, the On-To-Knowledge methodology includes the identification of goals that should be achieved by knowledge management tools and is based on an analysis of usage scenarios.

The steps proposed by the methodology are **kickoff:** where ontology requirements are captured and specified, competency questions are identified, potentially reusable ontologies are studied, and a first draft version of the ontology is built; **refinement:** where a mature and application oriented ontology is produced; **evaluation:** where the requirements and competency questionsare checked, and the ontology is tested in the application environment; and finally ontology maintenance.

ONTOLOGYSHARINGANDMERGING

Knowledgerepresentationistheapplicationoflogicandontologytothetaskofconstructing automated models.

Eachofthefollowingthreefieldscontributestoknowledge representation:

• **Logic:** Different implementations support different subsets and variations of logic. Sharing information between implementations can usually be done automatically if the information can be expressed a common subset.

• **Ontology:**Differentsystemsmayusedifferentnamesforthesamekindsofobjects;orthey may use the same names for different kinds.

• **Computation:** Even when the names and definitions are identical, computational or implementation side effects may produce different behaviors indifferent systems. In some implementations, the order of entering rules may have inferences that impact computations. Sometimes, the side effects may cause an endless loop.

Although these three aspects of knowledge representation pose different kinds of problems, they are interdependent. Standardizing the terminology used to classify and find the information is important.

Forartificialintelligence,wheretheemphasisisoncomputerprocessing,efforthasbeendirected to precise axioms suitable for extended computation and deduction.

ONTOLOGYLIBRARIES

Scientistsshouldbeabletoaccessaglobal,distributedknowledgebaseofscientificdatathat appears to be integrated, locally available, and is easy to search.

Data is obtained by multiple instruments, using various protocols in differing vocabularies using assumptions that maybe inconsistent, incomplete, evolving, and distributed. Currently, ***there are existing ontology libraries including***

• DAMLontologylibrary(www.daml.org/ontologies).
• Ontolinguaontologylibrary(www.ksl.stanford.edu/software/ontolingua/).
• Prot´eg´eontologylibrary(protege.stanford.edu/plugins.html).

Availableupperontologiesinclude
• IEEEStandardUpperOntology(suo.ieee.org).
• Cyc(www.cyc.com).

Availablegeneralontologies include
• (www.dmoz.org).
• WordNet(www.cogsci.princeton.edu/∼wn/).
• Domain-specificontologies.
• UMLSSemanticNet.
• GO(GeneOntology)(www.geneontology.org).
• ChemicalMarkupLanguage,CML.

ONTOLOGYMATCHING

Ontology provides a vocabulary and specification of the meaning of objects that encompasses several conceptual models: including classifications, databases, and axiom theories. However, in the open Semantic Web environment different ontologies may be defined.

Ontology matching finds correspondences between ontology objects. These include ontology merging, query answering, and data translation. Thus, ontology matching enables data interoperate.

Todayontologymatchingisstilllargelylabor-intensiveanderror-prone. Asaresult,manual matching has become a key bottleneck.

StringMatching

String matching can help in processing ontology matching. String matching is used in text processing,informationretrieval,andpatternmatching.Therearemanystringmatchingmethods including ―edit distance‖ for measuring the similarities of two strings.

Letus considertwostrings;S1andS2.

If we use limited steps of character edit operations (insertions, deletions, and substitutions), S1 can be transformed into S2 in an edit sequence. The edit distance defines the weight of an edit sequence.

The existing ontology files on the Web (e.g., http://www.daml.org/ontologies) show that people usually use similar elements to build ontologies, although the complexity and terminology may be different. This is because there are established names and properties to describe a concept. The value of string matching lies in its utility to estimate the lexical similarity.

However,wealsoneedtoconsidertherealmeaningofthewordsandthecontext.

Inaddition,therearesomewordsthataresimilarinalphabetformwhiletheyhavedifferent meaningsuchas,―too‖and―to.‖Hence,itisnotenoughtouseonlystringmatching.

ONTOLOGYMAPPING

Ontologymappingenablesinteroperabilityamongdifferentsourcesin theSemanticWeb.Itis required for combing distributed and heterogeneous ontologies.

Ontology mapping transforms the source ontology into the target ontology based on semantic relations. There are three mapping approaches for combing distributed and heterogeneous ontologies:

1. Mappingbetweenlocalontologies.
2. Mappingbetween integrated global ontologyand localontologies.
3. Mappingfor ontologymerging,integration, oralignment.

Ontologymerge,integration,andalignmentcan beconsideredasontologyreuse processes.

Ontologymergeistheprocessofgeneratingasingle,coherentontologyfromtwoormore existing and different ontologies on the same subject.

Ontologyintegrationistheprocessofgeneratingasingleontologyfromtwoormorediffering ontologies in different subjects. Ontology alignment creates links between two original ontologies.

ONTOLOGYMAPPINGTOOLS

Thereare threetypesofontologymappingtoolsandprovideanexampleofeach:

For ontology mapping between local ontologies, an example mapping tool is **GLUE**. GLUE is a system that semiautomatically creates ontology mapping using machine learning techniques. Given two ontologies, GLUE finds the most similar concept in the other ontology.

For similarity measurement between two concepts, GLUE calculates the joint probability distribution of the concepts. The GLUE uses a multistrategy learning approach for finding joint probability distribution.

Forontologymappingsbetweensourceontologyandintegrated globalontology,anexampletool is **Learning Source Description (LSD).** In LSD, Schema can be viewed as ontologies with restricted relationship types. This process can be considered as ontology mapping between information sources and a global ontology.

For ontology mapping in ontology merging, alignment, and integration, an example tool is **OntoMorph.** OntoMorph provides a powerful rule language for specifying mappings, and facilitates ontology merging and the rapid generation of knowledge base translators. It combines two syntactic rewriting and semantic rewriting. Syntactic rewriting is done through pattern-directed rewrite rules for sentence-level transformation based on pattern matching. Semantic rewriting is done through semantic models and logical inference.

LOGIC:

Logic is the study of the principles of reasoning. As such, it constructs formal languages for expressing knowledge, semantics, and automatic reasoners to deduce (infer) conclusions.

Logic forms the foundation of Knowledge-Representation (KR), which has been applied to Artificial Intelligence (AI) in general and the World Wide Web in particular. Logic provides a high-level language for expressing knowledge and has high expressive power. In addition, KR has a well-understood formal semantics for assigning unambiguous meaning to logic statements.

Predicate (or first-order) logic, as a mathematical construct, offers a complete proof system with consequences. Predicate logic is formulated as a set of axioms and rules that can be used to derive a complete set of true statements (or proofs).

As a result, with predicate logic we can track proofs to reach their consequences and also logically analyze hypothetical answers or statements of truth to determine their validity.

Proof systems can be used to automaticallyderive statements syntacticallyfrom premises. Given a set of premises, such systems can analyze the logical consequences that arise within thesystem.

Both RDF and OWL (DL and Lite) incorporate capabilities to express predicate logic that provide a syntax that fits well with Web languages.

RULE:

Inference Rules

In logic, a rule is a scheme for constructing valid inferences. These schemes establish syntactic relations between a set of formulas called premises and an assertion called a conclusion. New true assertions can be reached from already known ones.

Therearetwoformsofdeductivelyvalidargument:

1. modus ponens(Latinfor―the affirmingmode‖)
2. modustollens(thedenyingmode).

Forfirst-orderpredicate logic, rules ofinference areneeded todealwithlogical quantifiers.

Related proof systems are formed from a set of rules, which can be chained together to form proofs, or derivations. If premises are left unsatisfied in the derivation, then the derivation is a proof of a conditional statement: ―*if* the premises hold, *then* the conclusion holds.‖

Inferencerules mayalso bestatedin thisform:
(*1*)somepremises; (*2*)aturnstilesymbol,whichmeans ―infers,‖―proves,‖or ―concludes‖;and (*3*)a conclusion.

Theturnstilesymbolizestheexecutive power.
Theimplicationsymbol → indicates*potential* inferenceanditisalogicaloperator.

For the Semantic Web, logic can be used by software agents to make decisions and select a path of action. For example, a shopping agent may approve a discount for a customer because of the rule:

RepeatCustomer*(*X*)*→discount*(*25%*)*

whererepeatcustomers areidentified fromthe companydatabase.

Thisinvolvesrulesoftheform―IF(condition),THEN(conclusion).‖Withonlyafinitenumber of comparisons, we are required to reach a conclusion.

Axioms of a theory are assertions that are assumed to be true without proof. In terms of semantics, axioms are valid assertions. Axioms are usually regarded as starting points for applying rules of inference and generating a set of conclusions.

Rules of inference, or *transformation rules*, are rules that one can use to infer a conclusion froma premise to create an argument. A set of rules can be used to infer any valid conclusion if it is complete, while never inferring an invalid conclusion, if it is sound.

Rules can be either conditional or biconditional. **Conditional rules**, or *rules of inference*, are rules that one can use to infer the first type of statement from the second, but where the second cannot be inferred from the first. With **biconditional rules**, in contrast, both inference directions are valid.

ConditionalTransformationRules
Wewill useletters *p*,*q*, *r*,*s*, etc.as propositional variables.

AnargumentisModusponensifithasthefollowingform(P1referstothefirstpremise;P2to the second premise: C to the conclusion):
(P1) if*p* then*q*
(P2)*p*
(C)*q*

Example:
(P1)IfSocratesishumanthenSocratesismortal. (P2)
Socrates is human.
(C)Socratesismortal.

WhichcanberepresentedasModus ponens:

[*(p*→*q)*∧*p*]→ [*q*]
AnargumentisModustollensifithasthefollowingform: (P1)
if *p* then *q*
(P2)not-*q*
(C)not-*p*

Example:
(P1)IfSocratesishumanthenSocratesismortal. (P2)
Socrates is not mortal.
(C)Socratesisnot human.

Inbothcases,theorderofthepremisesisimmaterial(e.g., inmodustollens ―not-*q*‖couldcome firstinsteadof―if*p*then*q*‖).

Modustollens [*(p→ q)*∧ ¬*q*] → [¬*p*]
Anargumentisadisjunctivesyllogismifithaseitherofthefollowingforms: (P1) *p* or *q* (P1) *p* or *q*
(P2)not-*p*(P2)not-*q*
(C)*q* (C)*p*

Theorderofthepremisesisimmaterial(e.g.,―not-*q*‖couldcomefirstinsteadof―*p*or*q*‖).

This argument form derives its name from the fact that its major premise is a ―disjunction,‖ that is, a proposition of the form ―*p* or *q*.‖ The propositions *p* and *q* are called the ―disjuncts‖ of the disjunction ―*p* or *q*.‖

Inlogic,thedisjunction―*p*or*q*‖isinterpretedas theclaimthatnotboth*p*and*q*arefalse;thatis, that at least one of them is true. Thus a disjunction is held to be true even when both its disjuncts are true.

BiconditionalTransformationRules
Biconditionalrules,or*rulesofreplacement*,arerulesthatonecanusetoinferthefirsttypeof statement from the second, or vice versa.

Doublenegativeeliminationisrepresentedas

[¬ ¬*p*] ↔ [*p*]

Tautologyis representedas
[*p*]↔[*p* ∨ *p*]

MONOTONICANDNONMONOTONIC RULES
If a conclusion remains valid after new information becomes available within predicate logic, then we refer to this case as a monotonic rule. If, however, the conclusion may become invalid with the introduction of new knowledge, then the case is called a nonmonotonic rule.

Nonmonotonic rules are useful where information is unavailable. These rules can be overridden by contrary evidence presented by other rules. Priorities are helpful to resolve some conflicts between nonmonotonic rules. The XML-based languages can be used to represent rules.

DESCRIPTIVE LOGIC
Descriptive logic is a family of logic based on knowledge-representation formalisms that is a descendant of semantic networks. It can describe the domain in terms of concepts (classes), roles (properties, relationships), and individuals.

InferenceandClasses
We can make inferences about relationships between classes, in particular subsumption between classes. Recall that A subsumes B when it is the case that any instance of B must necessarily be an instance of A.

INFERENCEENGINES

Anexpertsystem has threelevels oforganization: aworkingmemory, an inference engine,and a knowledge base. The inference engine is the control of the execution of reasoning rules. This means that it can be used to deduce new knowledge from existing information.

The inference engine is the core of an expert system and acts as the generic control mechanism that applies the axiomatic knowledge from the knowledge base to the task-specific data to reach some conclusion.

Twotechniques fordrawinginferences aregeneral logic-based inferenceengines and specialized algorithms. Many realistic Web applications will operate agent-to-agent without human intervention to spot glitches in reasoning. Therefore developers will need to have complete confidence in reasoner otherwise they will cease to trust the results.

HowtheInferenceEngineWorks

Insimplerule-basedsystems,therearetwokindsofinference,forwardandbackwardchaining.

ForwardChaining

In forward chaining, the data is put into working memory. This triggers rules whose conditions match the new data. These rules then perform their actions. The actions may add new data to memory, thus triggering more rules, and so on.

This is also called data-directed inference, because inference is triggered by the arrival of new data in working memory. Consider iterating continuously though the following set of rules until you reach a conclusion:

Rule1:IFAandCTHENF
Rule2:IFAandETHENG
Rule 3: IF B THEN E
Rule4:IFGTHEN D

To provethat D is true,given that A and Baretrue, westart with Rule 1 and go on down the list untilarulethat ―fires‖isfound. Inthiscase,Rule3istheonlyonethatfiresinthefirstiteration. At the end of the first iteration, it can be concluded that A, B, and E are true. This information is used in the second iteration.

In the second iteration, Rule 2 fires adding the information that G is true. This extra information causes Rule 4 to fire, proving that D is true. This is the method of forward chaining, where one proceedsfroma givensituationtowardadesired goal,addingnewassertionsalongtheway.This strategy is appropriate in situations where data are expensive to collect and few are available.

BackwardChaining

In backward chaining the system needs to know the value of a piece of data. Itsearches for rules whose conclusions mention this data. Before it can use the rules, it must test their conditions. This may entail discovering the value of more pieces of data, and so on. This is also called goal-directed inference, or hypothesis driven, because inferences are not performed until the system is made to prove a particular goal.

In backward chaining, we start with the desired goal and then attempt to find evidence for proving the goal. Using the forward chaining example, the strategy to prove that D is true would be the following.

First, findtherulethat proves D.This is Rule4.Thesubgoal is thento provethat Gis true. Rule 2 meets the subgoal, and as it is already known that A is true, therefore the next subgoal is to show that E is true. Rule 3 provides the next subgoal of proving that Bis true. But the fact that B is true is one of the given assertions. Therefore, E is true, which implies that G is true, which in turn implies that D is true.

Backward chaining is useful in situations where the amount of data is large and where a specific characteristic of the system is of interest.

TreeSearches

A knowledge base can be represented as a branching network or tree. There is a large number of tree searching algorithms available in the existing literature. However, the two basic approaches are depth-first search and breadth-first search.

The depth-first search algorithm begins at a node that represents either the given data (forward chaining) or the desired goal (backward chaining). It then checks to see if the left-most (or first) node beneath the initial node (call this node A) is a terminal node (i.e., it is proven or a goal). If not, it establishes node A on a list of subgoals outstanding. It then starts with node A and looks at the first node below it, and so on. If there are no more lower level nodes, and a terminal node has not been reached, it starts from the last node on the outstanding list and takes the next route of descent to the right.

Breadth-first search starts by expanding all the nodes one level below the first node. Then it systematically expands each of these nodes until a solution is reached or else the tree is completely expanded. This process finds the shortest path from the initial assertion to a solution. However, such a search in large solution spaces can lead to huge computational costs due to an explosion in the number of nodes at a low level in the tree.

FullFirst-OrderLogicInference Engines

Using full first-order logic for specifying axioms requires a full-fledged automated theorem prover. First-order logic is semidecidable and inferencingis computationallyintractable for large amounts of data and axioms.

ClosedWorld Machine

The Closed World Machine (CWM) (www.w3.org/2000/10/swap/doc/cwm.html) inference engine written in Python by Tim Berners-Lee and Dan Connolly is a popular Semantic Web program. It is a general-purpose data processor for the Semantic Web and is a forward-chaining reasonerthatcanbeused forquerying,checking,transforming,andfilteringinformation. Itscore language is RDF, extended to include rules.

RDFINFERENCEENGINE

RDFis asystem meant forstatingmeta-informationthrough triples composedof a subject, a property, and an object. The subject and object can be either a designation like a URL or a set of another triple. Triples form a simple directed graph.

The first triple says that Smith owns a computer and the second says that there is a computer made by Apple. The third drawing, however, is composed of two triples, and it says that Smith owns a computer made by Apple.

Supposethesetripleswereplacedinadatabase.

In the first query, the question is who owns a computer? The answer is ―Smith.‖ In the second query, the question is What make of computer are defined in the database? The third query, however asks who owns a computer and what is the make of that computer?

The query is a graph containing variables that can be matched with thegraph. Should the graph inthedatabasebemoreextended,itwouldhavetobematched withasubgraph.So, generallyfor executingan RDFquerywhat has to be done is called ―subgraph matching.‖

Following the data model for RDF the two queries are in fact equal because a sequence of statements is implicitly a conjunction.

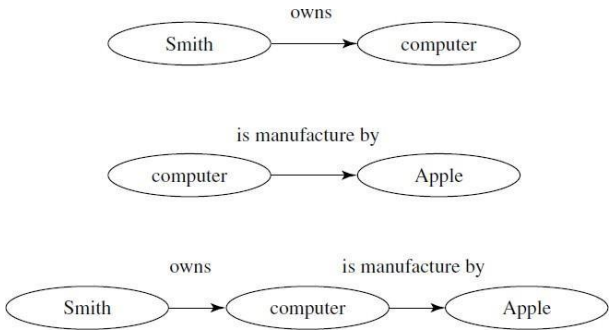


Fig: RDF Statements

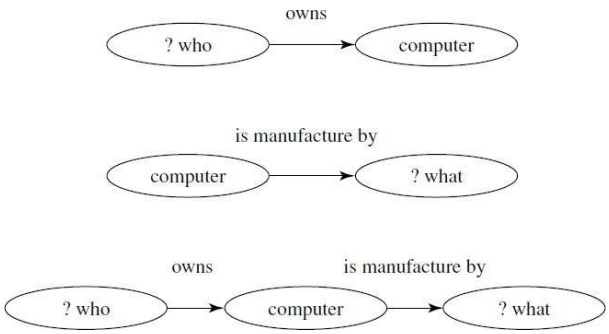


Fig:RDFQueries

Let us make a rule: If X owns a computer, then X must buy software.

Howdowerepresentsucha rule?

The nodes of the rule form a triple set. Here there is one antecedent, but there could be more. There is only one consequent. (Rules with more than one consequent can be reduced to ruleswith one consequent.)

ThedesiredanswerisJohnmustbuysoftware.Thequeryismatched withtheconsequentofthe rule. Now an action has to be taken: The antecedents of the rule have to be added to the database with the variables replaced with the necessary values (substitution). Then the query has to be continued with the antecedent of the rule.

:

ThequestionnowisWhoownsacomputer?Thisisequaltoaquerydescribedearlier.Arule subgraphistreateddifferentlyfromnonrule subgraphs.

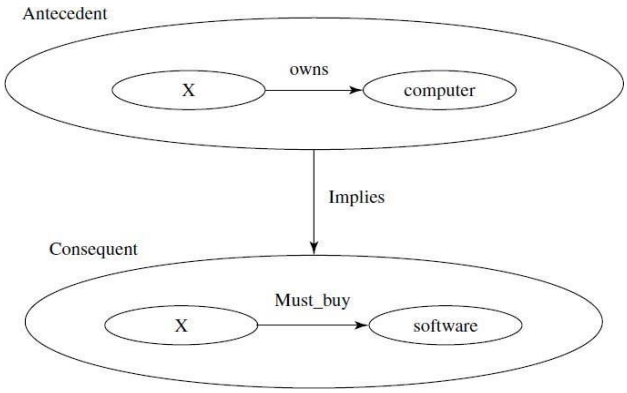


Fig:Graphrepresentationofa rule

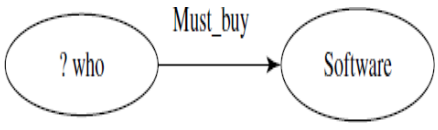


Fig:Querythat matcheswitharule

Atriplecanbemodeledasapredicate:triple(subject,property,object).Asetoftriplesequals a list of triples and a connected graph is decomposed into a set of triples.

For our example this gives
Triple(John, owns, computer).
Triple(computer,make,Apple).

Thissequenceisequivalentto:
[Triple(John,owns,computer).Triple(computer,make, Apple).]

From**FigureRDF Queries** the triplesareTriple(?who,owns, computer).

Triple(computer,make,?what).

Thissequenceisequivalentto:
[Triple(?who,owns,computer).Triple(computer,make,?what).]

From **Figure Graph representation of a rule** the triple is

Triple([Triple(X,owns,computer)],implies,[Triple(X,mustbuy,software)]).

From **Figure Query that matches with a rule** the triple is

Triple(?who,mustbuy, software).

Aunification algorithmforRDFcanhandle subgraph matchingand embedded rulesbythe term ***"subgraphmatchingwithrules."***

The unification algorithm divides the sequence of RDF statements into sets where each set constitutes a connected subgraph. This is called a **tripleset** that is done for the database and for the query. Then the algorithm matches each tripleset of the query with each tripleset of the database.

Agents
Agents are pieces of software that work autonomously and proactively. In most cases, an agent will simply collect and organize information. Agents on the Semantic Web will receive some tasks to perform and seek information from Web resources, while communicating with other Web agents, in order to fulfill its task. Semantic Web agents will utilize metadata, ontologies,and logic to carry out its tasks.

UNIT-IV

Semantic Web applications and services, Semantic Search, e-learning, Semantic Bioinformatics, Knowledge Base ,XML Based Web Services, Creating an OWL-S Ontology for Web Services, Semantic Search Technology, Web Search Agents and Semantic Methods.

SEMANTICWEBAPPLICATIONSANDSERVICES

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 exchangingdataremains.Foreverypairofapplicationssomeonehastocreatean―XMLtoXML bridge.‖

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

TheuseofResourceDescriptionFramework(RDF)inadditiontoXMLcanbeappropriatewhen information from two sources need to be merged or interchanged.

TheSemanticWebisbringingtotheWebanumberofcapabilities,suchasallowingapplications 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 EnterpriseApplication and Data Integration, and Knowledge Base.

SEMANTICWEBSERVICES

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.

WebServices areself-contained,self-described,componentapplicationsinvokedacrossthe Web to perform complex business processes. Once a Web Service is deployed, other applications can discover and invoke the service.

Atpresent,WebServicesrequirehumaninteraction inordertoidentifyandimplement.

Tim Berners-Lee has suggested that the integration of Web Services and the Semantic Webcould bedonein such a wayas to combine thebusiness logicof Web Services with theSemantic Web‘s meaningful content.

ThevisionforSemantic WebServicesistoautomatethediscovery,invocation,composition,and monitoringofWeb Servicesthrough theuseofmachineprocessing. Websites willbeableto use a set of classes and properties by declaring and describing an ontology of services. WebOntology Language for Services (called OWL-S) has been designed to meet this goal.

SEMANTICSEARCH

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 RDFand OWLdocuments that will help to formtheWeb into a semantic network. In a semantic network, the meaning of content is better represented and logical connections are formedbetween related information.

However, most semantic-based search engines suffer increasingly difficult performanceproblemsbecauseofthelargeand rapidlygrowingscaleoftheWeb. In orderforsemanticsearch 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

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 theother 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.

Bytailored restructuring ofinformation, futuresystems will beableto delivercontent to theend-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 alecturermight use amorescientificterm thusreflectingscalingincomplexity.

Theeducationalsectorcanalsousethe **InternetRelay Chat(IRC)(http://www.irc.org/) atool** 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.

SEMANTICBIOINFORMATICS

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 bythe 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.

Lifescienceis a particularlysuitablefield forpioneeringthe SemanticWeb.

KNOWLEDGE BASE

In a number of parallel efforts, knowledge systems are being developed to provide semantic-basedandcontext-awaresystemsforthe acquisition,organization,processing,sharinganduseof 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 sellproducts 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 reasoningengine. 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-BASEDWEBSERVICES

WebServicesprovideastandardmeansofinteroperatingbetweendifferent softwareapplications 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 toimplement 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.

CREATINGANOWL-SONTOLOGYFORWEBSERVICES

CreatinganOWL-SbasedOntologyforaWebServicerequiresfive steps:

1. Describeindividual programs: describethe individual programs that comprise theservice. The process model provides a declarative description of a program‘s properties.

2. Describethe groundingforeachatomicprocess:relateeachatomicprocesstoitsgrounding.

3. Describecompositionsoftheatomicprocesses:describethecompositeprocessthatisa composition of its atomic processes.

4. Describeasimpleprocess:describeasimpleprocessfortheservice (optional).

5. Profiledescription:provideadeclarativeadvertisementfortheservice. Itispartiallypopulated by the process model.

SEMANTICSEARCHTECHNOLOGY

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

Searching Techniques

Semanticsearchdealswithconceptsandlogicalrelationships.

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.

Forexample,youmightwanttotrytodetermine―WhomdoesKevinBaconknow?‖basedon 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-childnodesinthistreecanbeviewedasonepotentiallogicaldeductionthat movestowardtrying 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 incompleteresults.

TheHaltingProblemisa decisionproblemthat can beinformallystatedas 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 oralgorithm 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 HaltingProblem 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 somepath, expanding nodes as you go, until you find a dead end.

Adeadendiseithera goal(success)or anodewhere youareunabletoproducenewchildren.So thesystem cannot prove anythingbeyond that point. Let us walk through a depth-first search and traverse the tree.

Startatthe top nodeandgo asdeeplyas possible:

1. Startatthehighest node.

2. Go as deeplyas possibledown onepath.

3. When you runinto adead-end (i.e., a falsestatement), back-up to thelast nodethat 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. Ifthis pathleads toanother dead-end,go back upanodeand trythe otherbranches.

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. Continueuntil youreach moredead-endsandhaveexhaustedsearchpossibilities.

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.

Anotherstrategyforsearchingis abreadth-first search.

Here you search layerbylayer. First, you tryto do all ofthezero-stepproofs, and then you tryto 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.

WEBSEARCHAGENTS

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.

Whatmakestheagentintelligentisitsabilitytomakearationaldecisionwhengivena choice. In otherwords,givenagoal,itwillmakedecisionstofollowthecourseofactionsthatwouldleadit 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. Usinga more advanced, efficient sortingalgorithm 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 goalhas 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.

SEMANTICMETHODS

Therearetwoapproachestoimprovingsearch resultsthroughsemanticmethods:(1)LSIand (2)SemanticWeb documents.

LATENTSEMANTICINDEX SEARCH

So far, we have reviewed search technologyin 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.

Latentsemanticindexingaddsanimportantsteptothedocumentindexingprocess. Inadditionto recording which keywords a document contains, the method examines the document collectionas 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 similarityvalues it has calculated for every content word, and returns the documents that it thinks best fit the query. Because two documents maybe semanticallyveryclose even if theydo not share a particular keyword, LSIdoesnotrequireanexactmatchtoreturnusefulresults.Whereaplainkeyword searchwillfailifthereisnoexactmatch,LSIwilloftenreturnrelevantdocumentsthatdonot contain the keyword at all.

SEMANTICWEBDOCUMENTS

ASemanticWeb Document isa document in RDFor OWLthat isaccessible 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.

UNIT-V

What is social Networks analysis, development of the social networks analysis, Electronic Sources for Network Analysis – Electronic Discussion networks, Blogs and Online Communities, Web Based Networks. Building Semantic Web Applications with social network features.

WhatissocialNetworksAnalysis?

Social Network Analysis (SNA) is the study of social relations among a set of actors. The key difference between network analysis and other approaches to social science is the focus on relationships between actors rather than the attributes of individual actors.

Network analysis takes a global view on social structures based on the belief that types and patterns of relationships emerge from individual connectivity and that the presence (or absence) of such types and patterns have substantial effects on the network and its constituents. In particular,the network structureprovidesopportunitiesandimposesconstraintsontheindividual actors by determining the transfer or flow of resources (material or immaterial) across the network.

The focus on relationships as opposed to actors can be easily understood by an example. When trying to predict the performance of individuals in a scientific communitybysome measure (say, number of publications), a traditional social science approach would dictate to look at the attributes of the researchers such as the amount of grants they attract, their age, the size of the team they belong to etc. A statistical analysis would then proceed by trying to relate these attributes to the outcome variable, i.e. the number of publications.

In the same context, a network analysis study would focus on the interdependencies within the research community. For example, one would look at the patterns of relationships that scientists have and the potential benefits or constraints such relationships may impose on their work.

The patterns of relationships maynot only be used to explain individual performance but also to hypothesize their impact on the network itself (network evolution). Attributes typically play a secondary role in network studies as control variables.

SNA is thus a different approach to social phenomena and therefore requires a new set of concepts and new methods for data collection and analysis. Network analysis provides a vocabulary for describing social structures, provides formal models that capture the common properties of all (social) networks and a set of methods applicable to the analysis of networks in general.

It is interesting to note that the formalization of network analysis has brought much of the same advantages that the formalization of knowledge on the Web (the SemanticWeb) is expected to bring to many application domains.

The methods of data collection in network analysis are aimed at collecting relational data in a reliable manner. Data collection is typically carried out using standard questionnaires and observation techniques that aim to ensure the correctness and completeness of network data.

DevelopmentofSocialNetwork Analysis

The field of Social Network Analysis todayis the result of the convergence of several streams of applied research in sociology, social psychology and anthropology.

Many of the concepts of network analysis have been developed independently by various researchers often through empirical studies of various social settings. **For example,** many social psychologists of the 1940s found a formal description of social groups useful in depicting communication channels in the group when trying to explain processes of group communication. Already in the mid-1950s anthropologists have found network representations useful in generalizing actual field observations, **For example** when comparing the level of reciprocity in marriage and other social exchanges across different cultures.

Some of the concepts of network analysis have come naturally from social studies. In an influential early study at the Hawthorne works in Chicago, researchers from Harvard looked at theworkgroupbehavior(e.g. communication,friendships,helping,controversy)ataspecificpart of the factory, the bank wiring room [May33].

Theinvestigatorsnoticedthatworkersthemselvesusedspecifictermstodescribewhoisin ―our group‖. The researchers tried to understand how such terms arise by reproducing in a visual way the group structure of the organization as it emerged from the individual relationships of the factory workers (see below Figure).

Despitethe various efforts, each of the earlystudies useda different set of concepts and different methods of representation and analysis of social networks. However, from the 1950s network analysis began to converge around the unique world view that distinguishes network analysis from other approaches to sociological research. (Theterm ―social network‖ has been introduced by Barnes in 1954.)

This convergence was facilitated by the adoption of a graph representation of social networks usually credited to Moreno. What Moreno called a *sociogram* was a visual representation of social networks as a set of nodes connected by directed links.

The nodes represented individuals in Moreno‘s work, while the edges stood for personal relations. However, similar representations can be used to depict a set of relationships between any kind of social unit such as groups, organizations, nations etc.

While 2D and 3D visual modelling is still an important technique of network analysis, the sociogram is honored mostly for opening the way to a formal treatment of network analysisbasedongraphtheory.

The following decades have seen a tremendous increase in the capabilities of network analysis mostlythrough newapplications. SNA gains its relevancefromapplications and these settings in turn provide the theories to be tested and greatly influence the development of the methods and the interpretation of the outcomes.

Forexample,oneoftherelativelynewareasofnetworkanalysisistheanalysisof networksinentrepreneurship,an activeareaofresearch thatbuildsand contributes to organization and management science. The vocabulary, models and methods of network analysis also expand continuouslythrough applications that require to handle ever more complex data sets.

The increasingvarietyof applications and related advances in methodologycan be best observed at the yearly Sunbelt Social Networks Conference series, which started in 1980. The field of Social Network Analysis also has a journal of the same name since 1978, dedicated largely to methodological issues.

While the field of network analysis has been growing steadily from the beginning, there have been ***two developments*** in the last two decades that led to an explosion in network literature. **First,** advances in information technology brought a wealth of electronic data and significantly increased analytical power. **Second,** the methods of SNA are increasingly applied to networks other than social networks such as the hyperlink structure on the Web or the electric grid.

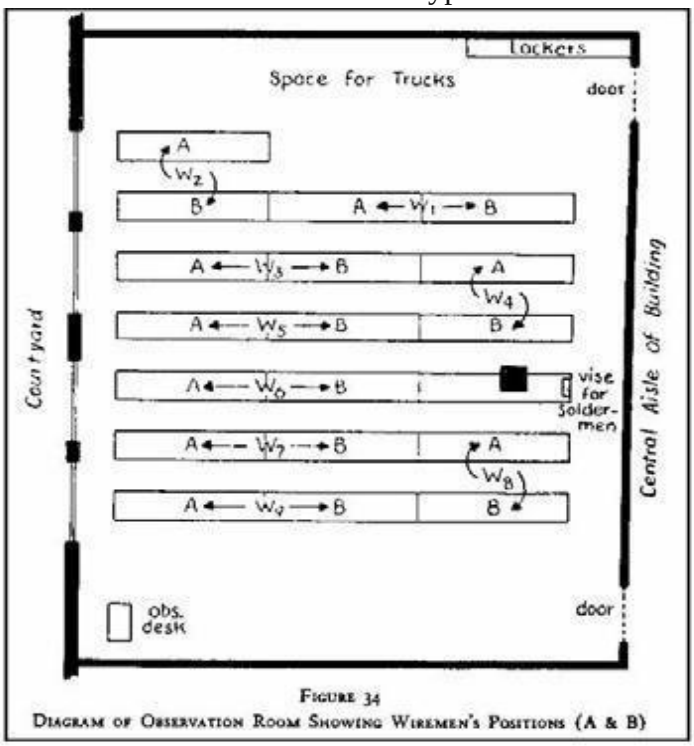


FIGURE 34
DIAGRAM OF OBSERVATION ROOM SHOWING WIREMEN'S POSITIONS (A & B)

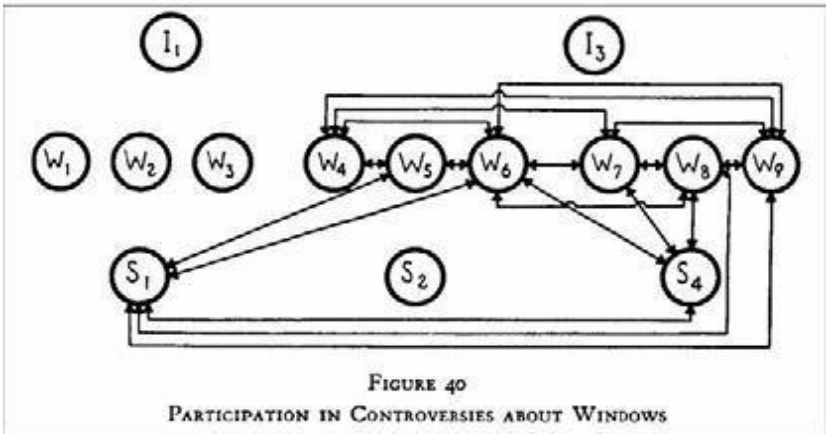


FIGURE 40
PARTICIPATION IN CONTROVERSIES ABOUT WINDOWS

IllustrationsfromanearlysocialnetworkstudyattheHawthorneworksofWesternElectric in Chicago. The upper part shows the location of the workers in the wiring room, while the lower part is a network image of fights about the windows between workers (W), solderers (S)andinspectors (I).

Electronicsourcesfornetworkanalysis

Electronicdiscussion networks

Oneoftheforemost studies to illustratetheversatilityof electronicdatais aseries ofworksfrom the Information Dynamics Labs of Hewlett-Packard.

Tyler, Wilkinson and Huberman analyze communication among employees of their own lab by using the corporate email archive. They recreate the actual discussion networks in the organization by drawing a tie between two individuals if theyhad exchanged at least a minimum number of total emails in a given period, filtering out one-way relationships. Tyler et al. find the study of the email network useful in identifying leadership roles within the organization and finding formal as well as informal communities. (Formal communities are the ones dictated by the organizational structure of the organization, while informal communities are those that develop across organizational boundaries.)

The authors verifythis finding through a set of interviews where theyfeed back the results to the employees of the Lab.

Wu, Huberman, Adamic and Tyler use this data set to verify a formal model of information flow in social networks based on epidemic models [WHAT04]. In yet another study, Adamic andAdar revisits one of the oldest problems of network research, namely the question of *local search*: how do people find short paths in social networks based on only local information about their immediate contacts?

Theirfindingssupport earlierresultsthat additionalknowledgeon contactssuchastheirphysical location and position in the organization allows employees to conduct their search much more efficiently than using the simple strategy of always passing the message to the most connected neighbor. Despite the versatility of such data, the studies of electronic communication networks based on email data are limited by privacy concerns.

Public forums and mailinglists can be analyzed without similar concerns. Startingfrom the mid-nineties, Marc Smith and colleagues have published a series of papers on the visualization and analysis of USENET newsgroups, which predate Web-based discussion forums

In the work of Peter Gloor and colleagues, the source of these data for analysis is the archive of the mailing lists of a standard setting organization, the World Wide Web Consortium (W3C)

The W3C —which is also the organization responsible for the standardization of Semantic Web technologies—is unique amongstandardization bodies in its commitment to transparencytoward the general public of the Internet and part of this commitment is the openness of the discussions within the working groups.

Group communication and collective decision taking in various settings are traditionally studied using much more limited written information such as transcripts and records of attendance and voting, see e.g. As in the case with emails Gloor uses the headers of messages to automatically re-create the discussion networks of the working group.1 The main technical contribution of Gloor is a dynamic visualization of the discussion network that allows to quickly identify the moments when key discussions take place that activate the entire group and not just a few select members.

Gloor also performs a comparative study across the various groups based on the structures that emerge over time.

Blogsandonlinecommunities

Content analysis has also been the most commonly used tool in the computer-aided analysis of blogs (web logs), primarily with the intention of trend analysis for thepurposes of marketing.2 While blogs are often considered as ***"personal publishing" or a "digital diary",*** **bloggers** themselves know that blogs are much more than that: ***modern blogging tools*** allow to easily comment and react to the comments of other bloggers, resulting in websof communication among bloggers. These ***discussion networks*** also lead to the establishment of dynamic communities, which often manifest themselves through syndicated blogs (aggregated blogs that collect posts from a set of authors blogging on similar topics), blog rolls (lists of discussion partners on a personal blog) and even result in real world meetings such as the Blog Walk series of meetings.

A slight difference is that unlike with personal emails messages to a mailing list are read by everyone on the list. Nevertheless individuals interactions can be partly recovered by looking at ***To:*** and ***CC:*** fields of email headers as well as the Reply ***Figure 3.1*** shows some of the features of blogs that have been used in various studies to establish the networks of bloggers.

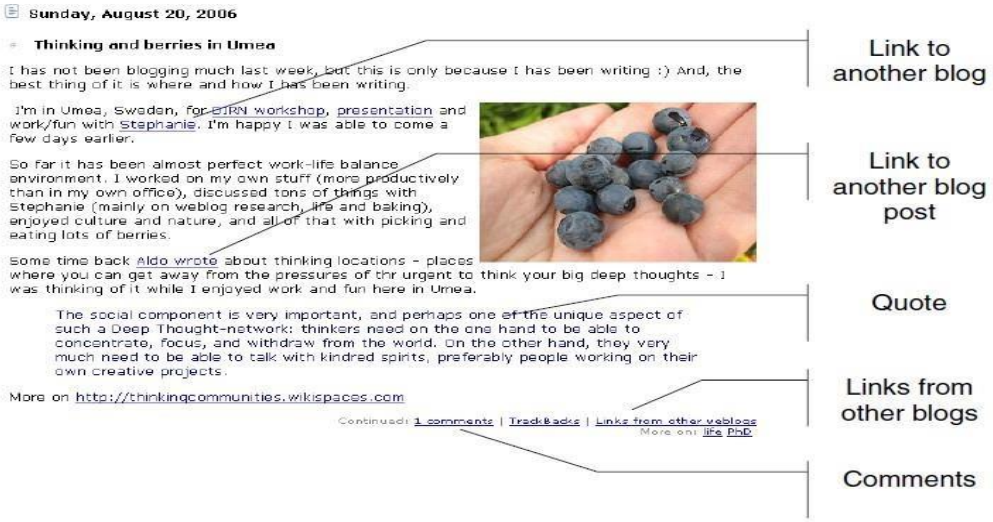


Figure 3.1. Features of blogs that can be used for social network extraction. Note also that —unlike web pages in general— blog entries are timestamped, which allows to study network dynamics, e.g. the spread of information in online communities.

Blogs make a particularly appealing research target due to the availability of structuredelectronic data in the form of RSS (Rich Site Summary) feeds. **RSS feeds** contain the text of the blog posts as well as valuable metadata such as the timestamp of posts, which is the basis of dynamic analysis.

The 2004 US election campaign represented a turning point in blog research as it has been the first major electoral contest where blogs have been exploited as a method of building networks among individual activists and supporters. Blog analysis has suddenly shed its image as relevant only to marketers interested in understanding product choices of young demographics;

Online community spaces and social networking services such as MySpace, LiveJournal caterto socialization even more directly than blogs with features such as social networking (maintaining lists of friends, joining groups), messaging and photo sharing.

As they are typically used by a much younger demographic they offer an excellent opportunity for studying changes in youth culture. Paolillo, Mercure and Wright offer a characterization of the LiveJournal community based on the electronic data that the website exposes about the interests and social networks of its users.

Backstrom et al. also study the LiveJournal data in order to answer questions regarding the influence of certain structural properties on community formation and community growth, while also examining how changes in the membership of communities relates to (changes in) the underlying discussion topics. These studies are good examples of how directly available electronic data enables the longitudinal analysis of large communities (more than 10,000 users).

Most online social networking services (Friendster, Orkut, LinkedIn and their sakes) closely guard their data even from their own users.

A technological alternative to these centralized services is the FOAF network (see also Chapter 5). FOAF profiles are stored on the web site of the users and linked together using hyperlinks. Thedrawback of FOAFis that at themoment thereis alack oftools forcreatingand maintaining profiles as well as useful services for exploiting this network. Nevertheless, a few preliminary studies have already established that the FOAF network exhibits similar characteristics to other online social networks.

Web-basednetworks

The content of Web pages is the most inexhaustible source of information for social network analysis. This content is not onlyvast, diverse and free to access but also in many cases more up to date than any specialized database.

There are two features of web pages that are considered as the basis of extracting social relations: links and co-occurrences (see Figure 3.2). The linking structure of the Web is considered asproxyforrealworldrelationshipsaslinksarechosenbytheauthorofthepageand connect to other information sources that are considered authoritative and relevant enough to be mentioned. The biggest drawback of this approach is that such direct links between personal pages are very sparse: due to the increasing size of the Web searching has taken over browsing as the primary mode of navigation on the Web. As a result, most individuals put little effort in creating new links and updating link targets or have given up linking to other personal pages altogether.

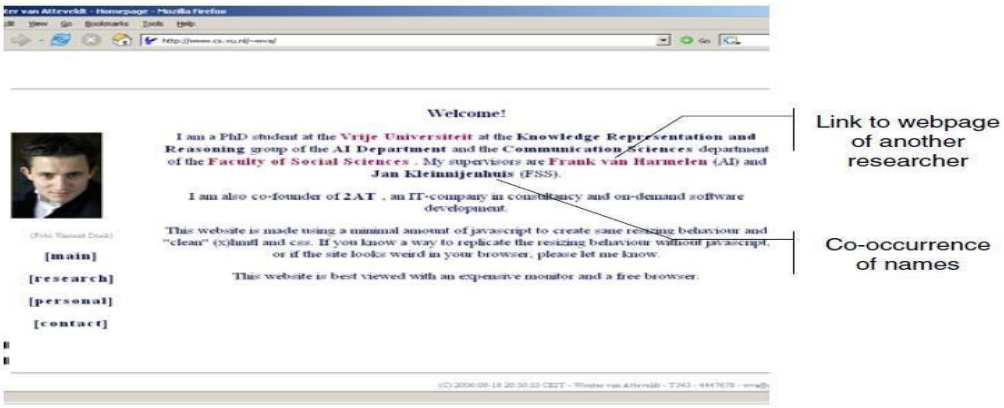


Figure 3.2. Features in web pages that can be used for social network extraction.

For this reason most social studies based on the linking structure of the web are looking at relationships at higher levels of aggregation. Unfortunately, search engines such as Google or Yahoo! typically limit the number of queries that can be issued a day.

The data for this analysis comes from bibliographic records, project databases and hyperlink networks. The connections for the latter are collected by crawling the websites of the institutions involved. In principle it could be possible to extract more fine-grained networks from the homepages of the individual researchers.

Co-occurrences of names in web pages can also be taken as evidence of relationships and are a more frequent phenomenon. On the other hand, extracting relationships based on co-occurrence of the names of individuals or institutions requires web mining as names are typically embedded in the natural text of web pages. (Web mining is the application of text mining to the content of web pages.)

Web mining has been first tested for social network extraction from the Web in the work of Kautz el al. on the ReferralWeb project in the mid-90s [KSS97]. Thegoal ofKautz et al. was not to perform sociological experiments but to build a tool for automating what he calls *referral chaining*: looking for experts with a given expertise, who are close to the user of the system, i.e. experts who can be accessed through a chain of referrals.

Tie strength was calculated by dividing the number of co-occurrences with the number of pages returned for the two names individually (see Figure 3.3). Also known as the *Jaccard-coefficient*, this is ***basically the ratio of the sizes of two sets:*** the intersection of the sets of pages and their union [Sal89]. The resulting value of tie strength is a number between zero (no co-occurrences) and one (no separate mentioning, only co-occurrences). If this number has exceeded a certain fixed threshold it was taken as evidence for the existence of a tie.

Although Kautz makes no mention of it we can assume that he also filtered ties also based on support, i.e. the number of pages that can be found for the given individuals or combination of individuals. The reason is that the Jaccard-coefficient is a relative measure of co-occurrence and it does not take into account the absolute sizes of the sets. In case the absolute sizes are verylow we can easily get spurious results:

for example, if two names only occur once on the Web and they occur on the same page, theirco-efficient will be one. However, in this case the absolute sizes are too low to take this as an evidence for a tie.

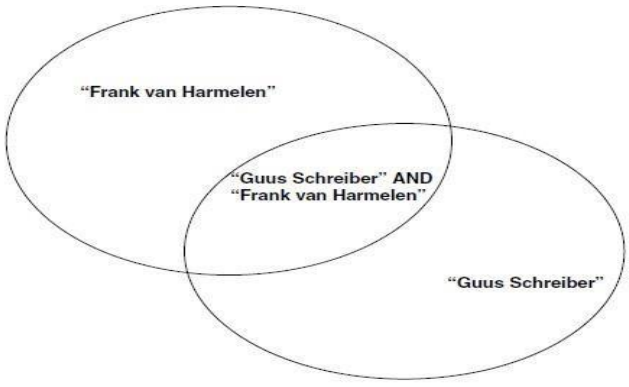


Figure 3.3. The Jaccard-coefficient is the ratio of the intersection and the union of two sets. In the case of co-occurrence analysis the two sets contain the pages where the individual names occur. The intersection is formed by the pages where both names appear.

Thenetwork inthesystemhasgrowntwo ways:

Firstly, the documents from the Web were searched for new names using *proper name extraction,* a fairly reliable NLP technique. These nameswere then used to extract new names, a process that was repeated two or three times. (Note that this is similar to the *snowballing technique of* network analysis where the network under investigation is growing through new names generated by participants in the study.)

Second,usersofthesystemwerealsoallowedtoregisterthemselves.

Kautz neverevaluatedhis system in thesenseofaskingwhetherthenetworks heextractedare an accurate reflection of real world networks. He notes that the system as a recommender system performed well on both the research domain and in the corporate setting, although ―the recommendations made by (any) recommender system tend to be either astonishingly accurate,or absolutely ridiculous [, which is] true for any AI-complete problem‖.However, he suggestthat the system is able to keep the trust of the user provided that it is made transparent. For example, the system can show the evidence on which the recommendation is based and indicate the level of confidence in its decisions.

A disadvantage of the Jaccard-coefficient is that it penalizes ties between an individual whose name often occurs on theWeb and less popular individuals (see Figure 3.4). In the sciencedomain this makes it hard to detect, for example, the ties between famous professors and their PhDstudents. Inthis casewhilethenameoftheprofessoris likelyto occur on alargepercentage of the pages of where the name of the PhD student occurs but not vice versa. For this reason we use an asymmetric variant of the coefficient.

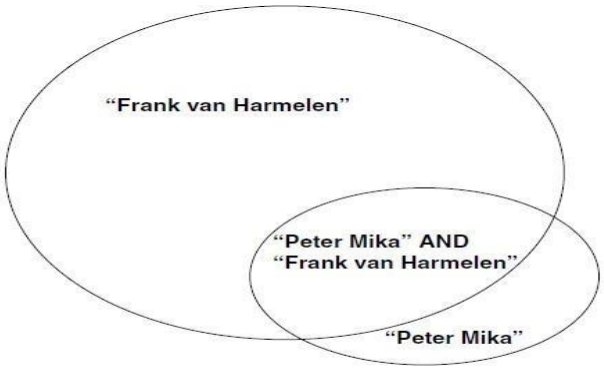


Figure 3.4. The Jaccard-coefficient does not show a correlation in cases where there is a significant difference in the sizes of the two sets such as in the case of a student and a professor.

Kautz already notes that the biggest technical challenge in social network mining is the disambiguation of person names. Person‘s names exhibit the same problems of polysemy and synonymy that we have seen in the general case of web search.

In our work in extracting information about the Semantic Web community we also add a disambiguation term our queries. We use a fixed disambiguation term (*Semantic Web OR ontology*) instead of a different disambiguation term for every 'name. This is a safe (and even desirable) limitation of the query as we are only interested in relations in the Semantic Web context.

We also experimentwith a second method based on the concept of *average precision*. When computingthe weight of adirected link between two personswe consider an ordered list of pages

for the first person and a set of pages for the second (the relevant set) as shown in Figure 3.5. In practice, we ask the search engine for the top *N* pages for both persons but in the case of the second person the order is irrelevant for the computation. Let‘s define *rel*(*n*) as the relevance at position*n*, where*rel*(*n*) is1 ifthedocumentat position*n* istherelevantset andzero otherwise(1 $\leq n \leq N$).

Let*P*(*n*) denotetheprecisionat position*n* (alsoknown as*p@n*):

$$P(n) = \frac{\sum_{r=1}^{n} rel(r)}{n}$$

Average precision is defined as the average of the precision at all relevant positions:

$$P_{ave} = \frac{\sum_{r=1}^{N} P(r) * rel(r)}{N}$$

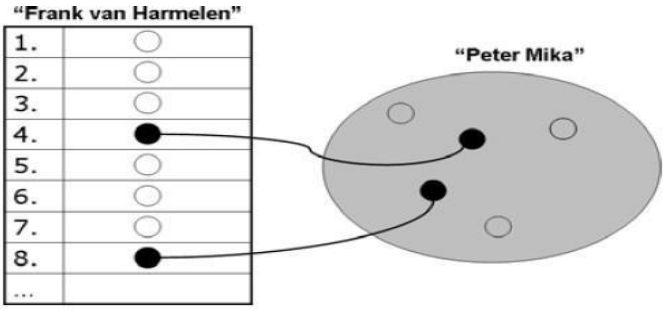


Figure 3.5. The average precision method considers also the position of the pages related to a second person in the list of results for the first person.

The average precision method is more sophisticated in that it takes into account the order in which the search engine returns document for a person:

It assumes that names of other persons that occur closer to the top of the list represent more important contacts than names that occur in pages at the bottom of the list. The method is also more scalable as it requires only downloading the list of top ranking pages once for each author.

Thedrawback ofthis method is that most search engineslimit thenumber ofpages returned to at most a thousand. In case a person and his contacts have significantly more pages than that it is likely that some of the pages for some the alters will not occur among the top ranking pages.

Lastly,wewouldnotethatonemayreasonablyargue against theabovemethodsonthebasisthat a single link or co-occurrence is hardly evidence for any relationship. In fact, not all links are equally important nor every co-occurrence is intended.

For example, it may very well happen that two names co-occur on a web page without much meaning to it (for example, theyare on the same page of the corporate phone book or appear in a list of citations).

BuildingSemanticWebapplicationswithsocialnetworkfeatures

Sesame is a general database for the storing and queryingRDF data. Along with Jena, Redland and the commercial offerings of Oracle, Sesame is one of the most popular triple stores among developers, appreciated in particular for its performance. Sesame has been developed by Aduna (formerly Aidministrator), but available as open source (currently under LGPL license).

Next, we describe the **Elmo API**, a general purpose ontology API for **Sesame**. **Elmo** allows to manipulateRDF/OWLdataatthelevelofdomainconcepts,withspecifictoolsforcollectingand aggregating RDF data from distributed, heterogeneous information sources. Elmo has been developed in part by the author and is available under the same conditions as Sesame, using the same website.

Lastly, we introduce a simple utility called GraphUtil which facilitates reading FOAF data into the graph object model of the Java UniversalNetwork Graph (JUNG) API. GraphUtil is open source and available as part of Flink.

ThegenericarchitectureofSemanticWeb applications

Semantic Web applications have been developed in the past years in a wide range of domains from cultural heritage to medicine, from music retrieval to e-science. Yet, almost all share a generic architecture as shown in Figure 6.1. By the definition above, all Semantic Web applications are mashups in that they build on a number of heterogeneous data sources and services under diverse ownership or control.

Before external, heterogeneous data sources can be reused, they need to be normalized syntactically as well as semantically. The first refers to transforming data into RDF syntax such as RDF/XML, while the latter means that the ontologies (schemas and instances) of the data sources need to be reconciled.

Needless to say; the first step can be skipped if the data is exposed as an RDF or OWLdocument, or can be queried dynamically using the SPARQL query language andprotocol.

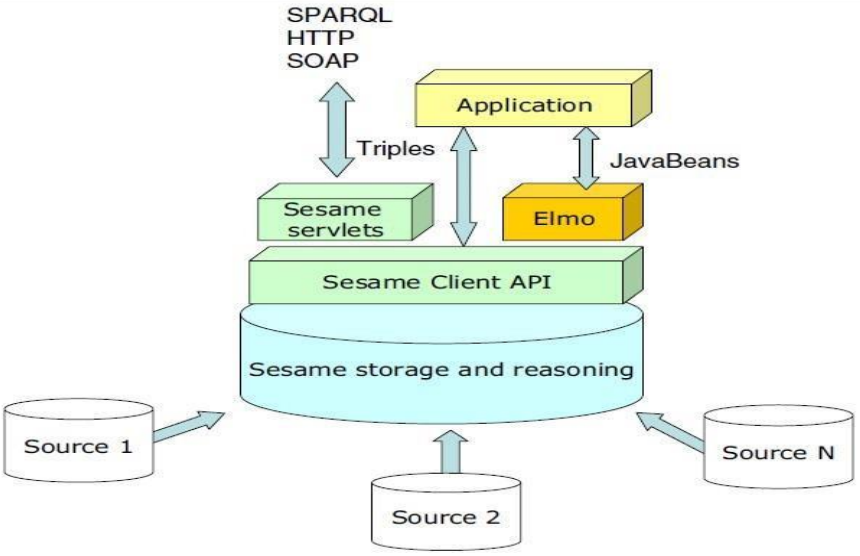


Figure 6.1. The generic design of Semantic Web applications using Sesame and Elmo. Developing with other triple stores results in similar architectures, but in general application code is not portable among triple stores due to proprietary APIs.

Most current Semantic Web applications are based on a fixed, small number of data sources selected by the application developer. In this case, the schemas of the data sources are known in advance and their mapping can be performed manually. In the future, it is expected that SemanticWeb applications will be able to discover and select new data sources and map them automatically.

Semantic Web applications persist information in ontology stores (also known as triple stores), databases specifically designed for the storage, manipulation and querying of RDF/OWL data. Ontology stores are almost always equipped with a reasoner or can be connected to an external reasoning service. Reasoning is used to infer new information based on the asserted facts or to check the existing information for consistency.

Some triple stores also allow to define customrules that are evaluated by the reasoner along with the rules prescribed by the ontology language itself. Reasoning can take place either when the data is added to a repository (forward-chaining) or at query time (backward-chaining).

Most Semantic Web applications have a web interface for querying and visualization and thus considered by all as web applications. However, this is not a requirement:

SemanticWebapplicationsmayhave a rich clientinterface (desktopapplications)orotherforms of access.

Queries are given as a combination of triple patterns and return a table (a set of variablebindings) as a result. This is similar to accessing a relational database. the above mentioned SPARQL protocol, which provides limited, read-only access and is only suitable for accessing remote data sources.

In other words, what is lacking is an equivalent of the ODBC and JDBC protocols for relational databases. This means that without additional abstraction layers (such as the one provided by Elmo), all application code is specific to a particular triple store.

Further, in most cases it is desirable to access a triple store on an ontological level, i.e. at thelevel of classes, instances and their properties. This is also the natural level of manipulating data in object-oriented frameworks.

The Elmo library to be introduced facilitates this by providing access to the data in the triplestore through Java classes that map the ontological data in the triple store. Setting and reading attributes on the instances of these classes result in adding and removing the corresponding triples in the data store.

Elmo is a set of interfaces that have been implemented for the specific case of working with data in Sesame triple stores. Sesame is one of the most popular RDF triple stores and it is to be introduced next. We note that the Elmo interfaces can be implemented for other, Java-based triples stores such as Jena.

Sesame

Sesame is a triple store implemented using Java technology. Much like a database for RDF data, Sesame allows creating repositories and specifying access privileges, storing RDF data in a repository and querying the data using any of the supported query languages. In the case of Sesame, these include Sesame‘s own SeRQL language and SPARQL.

Thedatain therepositorycanbemanipulatedonthelevelof triples:

Individualstatementscanbeadded andremovedfromthe repository.

RDF data can be addedor extracted in any of the supported RDF representations including theRDF/XML and Turtle languages.

Sesame can persistently store and retrieve the data from a variety of back-ends: data canpersist in memory, on the disk or in a relational database. As most RDF repositories, Sesame is not only a data store but also integrates reasoning. Sesame has a built-in inferencer for applying the RDF(S) inference rules.

While Sesame does not support OWL semantics, it does have a rule language that allows to capture most of the semantics of OWL, including the notion of inverse-functional properties and the semantics of the *owl*: *sameAs* relationship.

An important, recently added feature of Sesame is the ability to store and retrieve context information. In distributed scenarios, it is often necessary to capture metadata about statements. For example, in the case of collecting FOAFprofiles from the Web, we might want to keep track of where the information came from (the URL of the profile) and the time it was last crawled.

Context information is important even for centralized sites with user contributed content. Every triple becomes a *quad*, with the last attribute identifying the context. Contexts are identified by resources, which can be used in statements as all other resources. Contexts (named graphs) can alsobedirectlyqueriedusingtheSPARQLquerylanguagesupportedbythisversionofSesame. The above mentioned functionalities of ***Sesame can be accessed in three ways.***

First,SesameprovidesanHTMLinterfacethatcan beaccessed throughabrowser.

Second,asetofservletsexposesfunctionalityforremoteaccessthroughHTTP,SOAPand RMI.

Lastly, Sesame provides a Java client library for developers which exposes all the above mentioned functionality of a Sesame repository using method calls on a Java object called *SesameRepository*.

This object can provide access to both local Sesame servers (running in the same Java Virtual Machine as the application) or and remote servers (running in a different JVM as the application or on a remote machine.

Working with the Sesame client API is relatively straightforward. Queries, for example, can be executed by calling the *evaluateTableQuery* method of this class, passing on the query itself and the identifier of the query language.

Elmo

Elmo is a development toolkit consisting of two main components. The first one is the ElmoAPI, providing the above mentioned interface between a set of JavaBeans representing ontologicalclassesandtheunderlyingtriplestore containingthedatathatismanipulatedthrough the JavaBeans.

The API also includes the tool for generating JavaBeans from ontologies and vice versa. The second main component consists of a set of tools for working with RDF data, including an RDF crawler and a framework of smushers.

TheElmoAPI

The core of the Elmo API is the *ElmoManager* a JavaBean pool implementation that is responsible for creating, loading, renaming and removing ElmoBeans. ElmoBeans are a composition of *concepts* and *behaviors*.

Concepts are Java interfaces that correspond one-to-one to a particular ontological class and provide getter and setter methods corresponding to the properties of the ontological class. (The mapping is maintained using annotations on the interface.) The inheritance hierarchy of the ontological classes is mapped directly to the inheritance hierarchy of concepts. Elmo conceptsare typically generated using a code-generator.

AninstanceofElmoBeanscorrespondstoinstancesofthedataset.As resourcesinontologymay have multiple types, ElmoBeans themselves need to be composed of multiple concepts. ElmoBeans implement particular combinations of concept interfaces.

ElmoBeans can be generated runtime as the types of resources may change during the run-timeof the application. ElmoBeans mayalso implement behaviors. Behaviors are concrete or abstract classes that can be used to give particular implementations of the methods of a concept (in case the behavior should differ from the default behavior), but can also be used to add additional functionality. Behaviours can be mixed-in to ElmoBeans the same way that additional types can be added runtime.

The separation of concepts and behaviors, and the ability to compose them at will support the distributed application development, which is the typical scenario in case of Web applications.

Lastly, Elmo helps developers to design applications that arerobust against incorrect data, which is a common problem when designing for the Web.

In general,SemanticWebapplicationsprocessingexternaldatatypicallyhavefew guaranteesfor the correctness of the input. In particular, many of the RDF documents on the Web —especially documents written by hand—, are either syntactically incorrect, semantically inconsistent or violate some of the assumptions about the usage of the vocabularies involved. Most of these problems result from human error.

Syntax can be easily checked by syntax validators such as the online RDF validation service of theW3C1112 Inconsistency can be checked by OWL DL reasoners. Elmo provides solutions for performing checks that can only be carried out programmatically.

Elmotools

Elmo also contains anumberoftools to work with RDFdata. TheElmo *scutter* is a genericRDF crawler that follows ***rdfs:seeAlso*** links in RDF documents, which typically point to otherrelevant RDF sources on the web.

Severaladvancedfeaturesareprovidedtosupportthisscenario:

- **Blacklisting:** Sites that produce FOAF profiles in large quantities are automatically placed on a blacklist. This is to avoid collecting large amounts of uninteresting FOAF data produced by social networking and blogging services or other dynamic sources.
- **Whitelisting:**the crawlercan belimitedto adomain (defined byaURLpattern).
- **Metadata:** the crawler can optionallystore metadata about the collected statements. This metadata currently includes provenance (what URL was the information coming from) and timestamp (time of collection)

- **Filtering:** incoming statements can be filtered individually. This is useful to remove unnecessary information, such as statements from unknown namespaces.
- **Persistence:** when the scutter is stopped, it saves its state to the disk. This allows to continue scuttering from the point where it left off. Also, when starting the scutter it tries to load back the list of visited URLs from the repository (this requires the saving of metadata to be turned on).
- **Preloading from Google:** the scutter queue can be preloaded by searching for FOAF files using Google
- **Logging:**TheScutterusesSimpleLoggingFacadeforJava(SLF4J)toprovideadetailed logging of the crawler.

The smushers report the results (the matching instances) by calling methods on registered listeners. We provide several implementations of the listener interface, for example to write out the results in HTML, orto represent matches usingthe ***owl:sameAs***relationship and upload such statements to a Sesame repository.

Smusherscanalsobeusedasa *wrapper*.Thedifferencebetweenawrapper andasmusheristhat a smusher finds equivalent instances in a single repository, while a wrapper compares instances in a source repository to instances in a target repository.

If a match is found, the results are lifted (copied) from the source repository to the target repository. This component is typically useful when importing information into a specific repository about a certain set of instances from a much larger, general store.

GraphUtil

GraphUtil is asimpleutilitythat facilitatesreading FOAFdatainto thegraph object model ofthe Java Universal Network Graph (JUNG) API. GraphUtil can be configured by providing two different queries that define the nodes and edges in the RDF data.

These queries thus specify how to read a graph from the data. For FOAF data, the first query is typically one that returns the ***foaf :Person*** instances in the repository, while the second one returns ***foaf :knows*** relations between them. However, any other graph structure that can be defined through queries (views on the data) can be read into a graph.

JUNG16 is a Java library(API) that provides an object-oriented representation of different types of graphs (sparse, dense, directed, undirected, k-partite etc.) JUNG also containsimplementations for the most well known graph algorithms such as Dijkstra‘s shortest path.

We extended this framework with a new type of ranker called ***PermanentNodeRanker*** which makes it possible to store and retrieve node rankings in an RDF store.

Lastly, JUNG provides a customizable visualization framework for displaying graphs.Most importantly, the framework let‘s the developer choose the kind of layout algorithm to be usedand allows for defining interaction with the graph visualization (clicking nodes and edges, drag- and-drop etc.)Thevisualization component can beusedalso in applets as is thecasein Flink and openacademia.